

Deliverable

D6.4 FRACTAL engineering framework validation

Deliverable Id:	D6.4
Deliverable Name:	FRACTAL engineering framework validation
Status:	Final
Dissemination Level:	Public
Due date of deliverable:	2023 (M31)
Actual submission date:	2023 (M31)
Work Package:	WP6 CPS Communication Framework
Organization name of lead	IKERLAN
contractor for this	
deliverable:	
Author(s):	Ana Patricia Bautista, IKER
	Adrián Morán, IKER
	Luca Visconti, AKKODIS
	Pietro Abbatangelo, AKKODIS
	Enrico Ferrari, RULEX
	Nicola Alchera, RULEX
Reviewers:	Roman Obermaisser, SIEG
	Stefan Krassin, PLC2

Abstract:

D6.4 "FRACTAL engineering framework validation" is a report of the results of the validation tests of the functionalities and components of the FRACTAL Edge Node, designed and implemented in tasks T6.1 and T6.2. The tests were carried out following the validation methodology defined in deliverable D6.3, where the steps to be followed to carry out the validation process were set out.



This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 877056



Co-funded by the Horizon 2020 Programme of the European Union under grant agreement No 877056.

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Contents

1	His	tory		4
2	Sur	mma	ry	5
	2.1	Ach	nievements	5
3	Int	rodu	ction	7
4	Val	idati	on test template overview	8
5	Edg	ge No	ode architecture review	10
	5.1	FRA	ACTAL Edge Node processing architecture	10
	5.2	FRA	ACTAL Edge Node processing architecture implementation	11
6 co			on tests implementation of the Edge Node microservices functionalities	
	6.1	Tes	t case development	12
	6.2	Tes	t environment setup	17
	6.3	Tes	t execution	19
7	FRA	АСТА	L Edge Controller review	23
8	Val	idatio	on test plan and implementation of the Edge Controller	25
	8.1	Orc	hestration (Edge Controller)	25
	8.1	.1	Test planification	25
	8.1	.2	Test case development	27
	8.1	.3	Test environment setup	28
	8.1	.4	Test execution	28
	8.2	Orc	hestration (Agent Nodes Controller)	33
	8.2	.1	Test planification	33
	8.2	.2	Test case development	35
	8.2	.3	Test environment setup	36
	8.2	.4	Test execution	37
	8.3	Rur	ntime Manager	42
	8.3	.1	Test planification	42
	8.3	.2	Test case development	43
	8.3	.3	Test environment setup	50
	8.3	.4	Test execution	53
	8.4	Dat	a Ingestion	58
	8.4	.1	Test planification	58

	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1		Project	FRACTAL
	FRACTA	£	Title	FRACTAL engineering framework validation
	X		Del. Code	D6.4
	8.4.2	Te	st case dev	elopment60
	8.4.3	Te	st environr	nent setup61
	8.4.4	Te	st executio	n61
8	8.5 Fe	edera	ted Data C	ollection65
	8.5.1	Te	st planifica	tion65
	8.5.2	Te	st case dev	elopment66
	8.5.3	Te	st environr	nent setup66
	8.5.4	Te	st executio	n67
8	8.6 Lo	ow Er	nd Node	69
	8.6.1	Te	st planifica	tion69
	8.6.2	Те	st case dev	elopment70
	8.6.3	Те	st environr	nent setup78
	8.6.4	Te	st executio	n79
8	в.7 H	ardwa	are-level E	dge Controller83
	8.7.1	Te	st planifica	tion83
	8.7.2	Те	st case dev	elopment86
	8.7.3	Те	st environr	nent setup101
	8.7.4	Те	st executio	n104
9	Conclu	usions	s	
10	Bibliog	graph	y	
11	List of	figur	·es	
12	List of	table	es	
13	List of	abbr	eviations	
14	Annex	(es		
1	.4.1	Orch	estration (Edge Controller) component complete templates117
1	.4.2	Orch 122	estration (Agent Nodes Controller) component complete templates
1	.4.3	Runt	ime Manag	er component complete templates129
1	.4.4	Data	Ingestion	component complete templates134
1	.4.5	Fede	rated Data	Collection component complete templates139
1	.4.6	Low	End Node	component complete templates141
1	.4.7	Hard	lware-level	Edge Controller component complete templates 145

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

1 History

Version	Date	Modification reason	Modified by
0.0	30/05/2022	Draft	Ana Patricia Bautista
0.1	15/11/2022	Redefinition of sections	Ana Patricia Bautista
0.2	15/03/2023	Deliverable ready for review	Authors
1.0	29/03/2023	Final version	Authors

Table 1 Document history

Copyright © FRACTAL Project Consortium	4 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

2 Summary

This deliverable reports the results of the validation tests implemented in the framework of task T6.3. The inputs of this task have been the developments of tasks T6.1 and T6.2, the Edge Node design, and implementation and the Edge Controller design and implementation, respectively.

For the development of this task, deliverables D6.1 and D6.2, other code, video demos, and repositories that were delivered as results of tasks T6.1 and T6.2 were studied and considered. The methodology outlined in D6.3 has been followed to define and develop the validation tests that will ensure that the components function correctly. However, the tests for tasks T6.1 and T6.2 have been approached in different ways and are presented in different chapters. The reason is that the results have been presented differently, and some of the developments that were in the scope of task T6.1 were moved to task T6.2 as is the case for the Low End Node. Therefore, the components of Task T6.2 complement the developments in Task T6.1, so the results presented in this document are aligned with that situation.

It is important to mention that during the definition of the validation tests, developers have been consulted to gain a better understanding of how the components work. In addition, they have also received feedback and have been able to implement some improvements to the components while the validation was being carried out.

2.1 Achievements

Highlights

- 1) "Coverage" the microservices related to connectivity presented in D6.1 have been considered in this deliverable.
- 2) "Coverage" all components presented in D6.2 have been taken into account in this deliverable.
- 3) "Coverage" D6.2 presents all functionalities related to each component, but some of them are out of project scope or for future scope; in this deliverable there is a clarification on what is working at this project step and what is out of scope.
- 4) "External view" in many cases partners who led the validation activities and test cases definition were not involved in the development.
- 5) "Know-how sharing" as per point 4, partner had the possibility to work together, tester and developer.
- 6) "Quality improvement" The validation work has helped to identify some bugs in the components, and these have been fixed in the course of the validation.

Copyright © FRACTAL Project Consortium	5 of 149
--	----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

In addition, feedback has been sent to the developers with some suggestions for improvement.

Lowlights

- 1) "Coverage": some functionality presented in D6.1 and D6.2 are of future scope.
- 2) In some cases, functionalities are presented and tested in simple scenarios. More complex scenarios will be defined during Use Cases implementation.

Copyright © FRACTAL Project Consortium	6 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

3 Introduction

The main content of this deliverable is the report of the validation results of the components developed in WP6 (tasks T6.1 and T6.2). The validation tests defined in this deliverable are based on deliverables D6.1, D6.2 and other materials such as video demos, code scripts and repositories that have been delivered by the developers of tasks T6.1 "FRACTAL processing node design and implementation" and T6.2 "FRACTAL Edge Controller design and implementation".

The phases of the validation process defined in section 5.4 of deliverable D6.3 have been followed to carry out the validation tests:

- 1. Test planification: where the scope of the test is defined, and the identification data is assigned.
- 2. Test case development: where the steps to carry out the test are defined, and the necessary scripts or configurations are created.
- 3. Test environment setup: This section refers to or explains how to prepare the test environment for the test (this information is given by the developers).
- 4. Test execution: the test is carried out and the results are reported according to these guidelines.

The reporting process was documented through subchapters with the name of the phases of the validation process, and in the case of the tests performed for task T6.2 components, they were documented in a validation template which encloses these steps. An overview of the validation template and usage is given in chapter 4.

First, chapter 5 provides an overview of the architecture of the Edge Node and the input received by task T6.1, and in chapter 6 the implementation of the validation tests of the microservices to test connectivity functionalities are documented.

Then, a brief description of the FRACTAL Edge Controller is presented in chapter 7, and in chapter 8, the implementation of the validation tests of the components developed in task T6.2 are documented.

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

4 Validation test template overview

As it is mentioned in the introduction, in this chapter, the validation test template is briefly explained. It was designed as a supporting document to report the phases of the validation process based on the validation methodology approach defined in deliverable D6.3, section 5.4.

	Validation test template	רו			
Test ID	TestNumber_ComponentCode (e.g. T01_WP6T62-06)				
Test type	Functional				
Test name	Testing interconnection between tools			1. Test planifica	ation
Date	25/05/2022			I. Test plaining	
Tester's Name	Tester 1				
Test scope or obje	ctive				
The objective of th	is test is to validate if the connection between Kafka and MongoDB works propoerly.	_			
Steps		1			
Step 1				2. Test case	
Step 2				Z. Test case	
Step 3				development	
Step n		_		development	
Results/Evidence					
Success criteria					4 To at an anti-
No error messages	/All partial results are as expected				4. Test execution
Test observations				3. Test	
Test configuration			_	anvirannaat	
Test conditions			-	environment	
Remarks		-		setup	
Test result				scrup	
Passed/Failed (if f	ailed, proved explanation)			_	



The template is self-explanatory. However, here's how to fill in some important fields:

TestID: consist of the test number + component code e.g., the first test for component WP6T62-06 would be:

T01_ WP6T62-06

Test type: it depends on the kind of test to be implemented, it could be functional or nonfunctional.

- Functional requirements describe the function of the system and its components, defining the behaviour between inputs and outputs of the system.
- Nonfunctional requirements establish the standardized criteria to assess the quality of the product developed. This kind of requirements should be tested on the actual HW platform. Since in a simulated environment the data might

Copyright © FRACTAL Project Consortium	8 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

not be accurate, and the results will be meaningless for the implementation in real use cases.

Test scope or objective: it defines what is to be validated and why.

Steps: the steps for carrying out the test are defined.

Results/Evidence: this space shows evidence that the test was carried out (screenshot of the execution, message from the application, etc.).

Success criteria: describe the criteria to consider the test successful.

Test observations:

- **Test configuration:** provide the set-up configuration and requirements to carry out the test. It could be a link to the FRACTAL repository or the reference to a document.
- **Test conditions:** briefly describe the conditions of the test, e.g., the test was done remotely, three servers were used to perform the test, etc.
- **Remarks:** report the issues detected during the testing process.

Test result: passed or failed/Not passed.

Copyright © FRACTAL Project Consortium	9 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

5 Edge Node architecture review

This chapter briefly describes the Edge Node architecture design implemented in task T6.1, which developed and deployed the necessary Edge Computing infrastructure. For more information and detail on the Edge Node software design and implementation, refer to deliverable D6.1.

The Edge Node architecture was designed using mostly open-source software. It includes core functionalities, core microservices and appropriate mechanisms to support remote monitoring, resource management and dynamic reconfiguration. It also provides connection interfaces to different IoT devices and cloud platforms.

According to D6.1, the reference architecture for the development of the Edge Node was Kubernetes with Docker, which suits the interoperability approach and integrations with other systems. In addition, it is open to extensions, which provides more openness and fits into fractality design principles. The implementation was based on Kubernetes family Microk8S and k3S as it brings lightweight, fully-featured, conformant Kubernetes for IoT devices.

5.1 FRACTAL Edge Node processing architecture

The FRACTAL architecture is hierarchical, which means that upon layers are built upon, and consume data and services provided by lower layers. Each layer performs an important function throughout the architecture. The application layer, for example, contains modules to implement core functions for visualization, control, analytics, data fusion, filtering, and storage database. On the other hand, the communication and connectivity layer provide the intermediate elements required in terms of hardware and software to exchange data between the data center and the network devices. For a detailed explanation of the different layers, refer to D6.1.

The following image shows the FRACTAL Edge Node processing architecture designed in task T6.1:

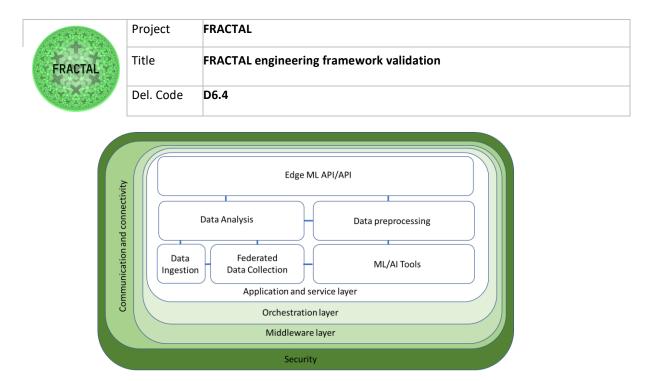


Figure 2 FRACTAL Edge Node processing architecture designed in task T6.1

5.2 FRACTAL Edge Node processing architecture implementation

The FRACTAL Edge Node processing architecture is the main input for this deliverable. The validation test plan will be designed based on it to validate the correct functionality of the Edge Node.

As it was mentioned before, the FRACTAL processing architecture was implemented using open-source applications and following an architecture based on microservices, which allows the Edge Node to have flexibility and scalability, as well as interoperability between heterogeneous devices and applications. Figure 3 below shows the FRACTAL Edge Node processing architecture implementation developed in task T6.1. For further detail, refer to deliverable D6.1.

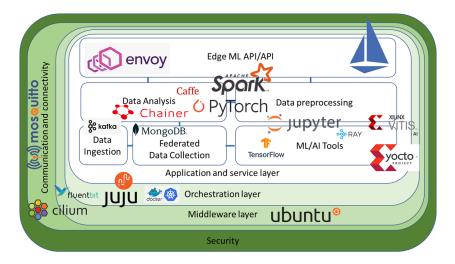


Figure 3 FRACTAL Edge Node processing architecture implementation developed in task T6.1

Copyright © FRACTAL Project Consortium	11 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

6 Validation tests implementation of the Edge Node microservices to test connectivity functionalities.

This chapter presents the results of the implementation of the tests for connectivity functionalities of the Edge Node. The first step, "Test planification" was done in the previous deliverable D6.3. It is important to mention that not all tests foreseen in the deliverable D6.3 were feasible to perform because the developments were not in the scope of task T6.1.

The code provided for testing can be founded in this link: <u>Code</u>

6.1 Test case development

In this test case, the interoperability between different data technologies has been evaluated. In order to understand the validations carried out, the whole test bed prepared must be explained. The testbed architecture is shown in Figure 4.

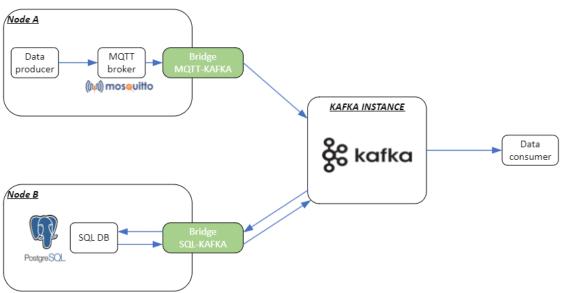


Figure 4 FRACTAL Edge Node testbed architecture

In the proposed testbed, there are three main entities that is worth to be described independently.

The so-called **Node A** has a data producer generating, in this case, the temperature of a city every 3 seconds. This data is produced randomly (with values between 1 and 30) and sent to the MQTT Broker in the same node. This node has a bridge to translate MQTT data to KAFKA. This will be one of the main entities under test.

On the other side, there is a so-called **Node B** that also has a bridge but, in this case, to translate information between KAFKA and a SQL database. This bridge is

Copyright © FRACTAL Project Consortium	12 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

bidirectional and will be the second main entity under test. This bridge has been configured to retrieve temperature data from KAFKA to store it in the SQL database and to periodically export the stored temperature value to another KAFKA topic.

Finally, there is the **KAFKA INSTANCE** that works as a central node to exchange data between nodes. Connected to this KAFKA broker, there is a data consumer in charge of reading information exchanged over the KAFKA broker to verify that the testbed is working correctly.

As aforementioned, this testbed was designed to validate that intercommunication between nodes is possible regardless of their differences in terms of technology.

The code of each entity is quite simple and is provided below for a better understanding:



Figure 5 - MQTT data producer code

As can be seen in Figure 5, the data producer entity generates, every 3 seconds, a random number between 1 and 30 and publish that value on the MQTT broker in the topic *temp_measured*.

Copyright © FRACTAL Project Consortium	13 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

from kafka import KafkaProducer
import paho.mqtt.client as mqtt
import time
The Topic Name
MQTT_TOPIC = "temp_measured"
KAFKA_TOPIC = "temp_on_kafka"
The address of Kafka server
KAFKA_HOST = "127.0.0.1:29092"
Mqtt Address
MQTT_HOST = "localhost"
MQTT Settings
<pre>mqtt_client = mqtt.Client("BridgeMQTT2Kafka")</pre>
<pre>mqtt_client.connect(MQTT_HOST, 1883)</pre>
Kafka Settings
kafka_producer = KafkaProducer(bootstrap_servers=KAFKA_HOST)
<pre>def on_message(client, userdata, message):</pre>
msg_payload = message.payload
<pre>msg_payload = msg_payload.decode()</pre>
<pre>print("Received MQTT message: ", msg_payload)</pre>
kafka_producer.send(KAFKA_TOPIC, message.payload)
<pre>print("Send the message: " + msg_payload + f" to Kafka with topic {KAFKA_TOPIC}!")</pre>
<pre>mqtt_client.loop_start()</pre>
<pre>mqtt_client.subscribe(MQTT_TOPIC)</pre>
<pre>mqtt_client.on_message = on_message</pre>
time.sleep(30000)
<pre>mqtt_client.loop_end()</pre>

Figure 6 - Bridge MQTT-KAFKA code

As can be seen in Figure 6, the MQTT-KAFKA bridge performs a connection to Node A's MQTT broker, in which it subscribes to the topic *temp_measured* (the one in which the data producer sends the generated data). On the other side, it also performs a connection with the KAFKA broker. Each time the bridge receives new data on the subscribed MQTT topic, it produces the same data of the KAFKA topic called *temp_on_data*. This way, the data produced by the data producer is available over MQTT within Node A (in MQTT's topic *temp_measured*) and also over KAFKA from other nodes (in KAFKA's *temp_on_kafka* topic).

Copyright © FRACTAL Project Consortium	14 of 149
--	-----------

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4
<pre>import psycopg2</pre>		

from kafka import KafkaConsumer
from kafka import KafkaProducer
import time
import threading
KAFKA Address
KAFKA_HOST = "127.0.0.1:29092"
PostgresSQL Settings
PS_DB_NAME = "ikerlan"
PS_USERNAME = "ikerlan"
PS_PASSWORD = "ikerlan"

PS_HOST = "localhost" PS_PORT = 5432

The Topic Name

TOPIC_WT = "postgresql"
TOPIC_RD = "temp_on_kafka"

```
def send_to_kafka():
```

```
# Kafka Settings
```

```
kafka_producer = KafkaProducer(bootstrap_servers=KAFKA_HOST)
```

```
# POSTGRESQL
```

```
connection = psycopg2.connect(f"dbname={PS_DB_NAME} user={PS_USERNAME}
password={PS_PASSWORD} host={PS_HOST} port={PS_PORT}")
```

cursor = connection.cursor()

```
while True:
```

```
cursor.execute("SELECT * FROM weather WHERE city = 'Oulu'")
data = cursor.fetchall()
```

```
Con d to date.
```

```
for d in data:
```

```
kafka_producer.send(TOPIC_WT, str(d[1]).encode())
print("Sent the " + d[0] + " temperature " + str(d[1]) + f" to topic {TOPIC_WT}")
```

```
time.sleep(1)
```

```
def fetch_from_kafka():
```

```
# Kafka Settings
```

```
kafka_consumer = KafkaConsumer(bootstrap_servers=KAFKA_HOST)
kafka_consumer.subscribe(TOPIC_RD)
```

POSTGRESQL

```
connection = psycopg2.connect(f"dbname={PS_DB_NAME} user={PS_USERNAME}
password={PS_PASSWORD} host={PS_HOST} port={PS_PORT}")
```

	Project	FRACTAL	
FRACTAL	Title	FRACTAL engineering framework validation	
	Del. Code	D6.4	
cursor = con	nection.curs	cor()	
while True:			
for msg	in kafka_con	isumer:	
data	<pre>data = msg.value.decode()</pre>		
curs	or.execute("	UPDATE weather SET temp = " + data + " WHERE city = 'Oulu'")	
conn	ection.commi	t()	
prin	t("Received	city temperature: " + data)	
ifname == "	main":		
threading.Th	read(target=	send_to_kafka).start()	
fetch_from_k	afka()		

Figure 7 - Bridge SQL-KAFKA code

As aforementioned, SQL-KAFKA bridge was designed as a bidirectional bridge. In this sense, it can be seen in the code (Figure 7) hat there are two threads, one in charge of moving data from the KAFKA broker to the SQL database and a second thread in charge of extracting data from SQL database and publishing it on KAFKA topic. The thread in charge of getting data from KAFKA reads information from KAFKA's topic temp_on_kafka, and each time it receives data, it stores the received data in the database. On the other side, the thread in charge of sending data to KAFKA reads information from the database and sends it to KAFKA's topic postgresql periodically.

At this point of the testbed's explanation, it is possible to see that the data generated by the data generator is available at these points:

- At MQTT level in Node A's scope.
- At KAFKA level in KAFKA instance's scope.
- At SQL level in Node B's scope.

Moreover, the value stored in the database is also available for reading at KAFKA's scope.

2752 P. 2.	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

from kafka import KafkaConsumer
Address of the Kafka
HOST = "127.0.0.1:29092"
LIST OF THE TOPICS
<pre>topics_list = ["temp_on_kafka", "postgresql"]</pre>
<pre>consumer = KafkaConsumer(bootstrap_servers=HOST)</pre>
<pre>consumer.subscribe(topics_list)</pre>
for i in consumer:
<pre>print("Message from topic: ", i.topic, "VALUE: ", i.value)</pre>

Figure 8 - Data consumer code

Finally, as can be seen in Figure 8, the data consumer subscribes itself on KAFKA broker to the topics *temp_on_kafka* and *postgresql*. It is a simple way to check that data exported from Node A is correct and that data imported to Node B is also successfully stored in the database and exported then to KAFKA again.

6.2 Test environment setup

For the simplicity of the test, all the testbed has been developed over a single node using Docker. The node used for this purpose is described in Table 6:

OS	Ubuntu 20.04 LTS
CPUs	2
RAM	4GB
Docker engine version	20.10.21
Python version	3.8

Table 2 – Testing node specifications

To run common entities of the testbed in an autonomous way, docker compose has been used, Figure 9 shows the configuration for the automatic launching of MQTT, KAFKA and SQL instances:

Copyright © FRACTAL Project Consortium	17 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

version: "3"
services:
mosquitto:
image: eclipse-mosquitto:1.6.12
ports:
- 1883:1883
zookeeper:
<pre>image: confluentinc/cp-zookeeper:latest</pre>
environment:
ZOOKEEPER_CLIENT_PORT: 2181
ZOOKEEPER_TICK_TIME: 2000
ports:
- 22181:2181
kafka:
<pre>image: confluentinc/cp-kafka:latest</pre>
depends_on:
- zookeeper
environment:
KAFKA_BROKER_ID: 1
KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://kafka:9092,PLAINTEXT_HOST://localhost:29092
KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: PLAINTEXT:PLAINTEXT,PLAINTEXT_HOST:PLAINTEXT
KAFKA_INTER_BROKER_LISTENER_NAME: PLAINTEXT
KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
ports: - 29092:29092
- 9092:9092
postgres:
image: postgres
environment:
POSTGRES_PASSWORD: ikerlan
POSTGRES_USER: ikerlan
POSTGRES_DB: ikerlan
ports:
- 5432:5432

Figure 9 - Automatic launching of MQTT, KAFKA and SQL instances

This docker compose configuration launches:

- MQTT broker based on Mosquitto.
- KAFKA broker (which also requires Zookeeper).
- SQL database based on PostgreSQL.

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

6.3 Test execution

The first step to execute the test, is running the docker compose configuration with basic services as it is shown in Figure 10:

	mpose -f docker-compose.yml up
[+] Running 4/0	
<pre># Container t61_demo2-:</pre>	
<pre>!! Container t61_demo2-</pre>	
<pre># Container t61_demo2-</pre>	
<pre># Container t61_demo2-</pre>	
	kafka-1, t61_demo2-mosquitto-1, t61_demo2-postgres-1, t61_demo2-zookeeper-1
t61_demo2-zookeeper-1	===> User
t61_demo2-zookeeper-1	uid=1000(appuser) gid=1000(appuser) groups=1000(appuser)
t61_demo2-zookeeper-1	===> Configuring
t61_demo2-mosquitto-1	1678435955: mosquitto version 1.6.12 starting
t61_demo2-mosquitto-1	1678435955: Config loaded from /mosquitto/config/mosquitto.conf.
t61_demo2-mosquitto-1	1678435955: Opening ipv4 listen socket on port 1883.
t61_demo2-mosquitto-1	1678435955: Opening ipv6 listen socket on port 1883.
t61_demo2-mosquitto-1	1678435955: mosquitto version 1.6.12 running
t61_demo2-postgres-1	
t61_demo2-postgres-1	PostgreSQL Database directory appears to contain a database; Skipping initialization
t61_demo2-postgres-1	
t61_demo2-postgres-1	2023-03-10 08:12:35.868 UTC [1] LOG: starting PostgreSQL 15.2 (Debian 15.2-1.pgdg110+1) on x86_64-pc-linux-gnu,
t61_demo2-postgres-1	2023-03-10 08:12:35.869 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
t61_demo2-postgres-1	2023-03-10 08:12:35.869 UTC [1] LOG: listening on IPv6 address "::", port 5432
t61_demo2-postgres-1	2023-03-10 08:12:35.873 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
t61_demo2-postgres-1	2023-03-10 08:12:35.878 UTC [29] LOG: database system was shut down at 2023-03-10 08:12:23 UTC
t61_demo2-postgres-1	2023-03-10 08:12:35.886 UTC [1] LOG: database system is ready to accept connections
t61_demo2-kafka-1	===> User
t61_demo2-kafka-1	uid=1000(appuser) gid=1000(appuser) groups=1000(appuser)
t61_demo2-kafka-1	===> Configuring
t61_demo2-zookeeper-1	===> Running preflight checks
t61_demo2-zookeeper-1	===> Check if /var/lib/zookeeper/data is writable
t61_demo2-zookeeper-1	<pre>===> Check if /var/lib/zookeeper/log is writable</pre>
t61_demo2-zookeeper-1	===> Launching
t61_demo2-zookeeper-1	===> Launching zookeeper
t61_demo2-kafka-1	===> Running preflight checks

Figure 10 - Run common services

Now that we have MQTT, KAFKA and PostgreSQL running, the next step is to run the data producer (Figure 11):

(t61) → T6.1_Demo2 python mqtt_producer.py						
Send a me	ssage to	MQTT:	29	to	topic	temp_measured
Send a me	ssage to	MQTT:	28	to	topic	temp_measured
Send a me	ssage to	MQTT:	21	to	topic	temp_measured
Send a me	ssage to	MQTT:	11	to	topic	temp_measured
Send a me	ssage to	MQTT:	17	to	topic	temp_measured
Send a me	ssage to	MQTT:	11	to	topic	temp_measured
Send a me	ssage to	MQTT:	30	to	topic	temp_measured

Figure 11 - Run data producer

Now that we have data in Node A's MQTT broker, it is time to run the bridge that exports this data to KAFKA's scope (Figure 12):

(t61) →	T6.1_Demo2 python kafka_bridge.py
Received	MQTT message: 26
Send the	message: 26 to Kafka with topic temp_on_kafka!
Received	MQTT message: 3
Send the	message: 3 to Kafka with topic temp_on_kafka!
Received	MQTT message: 29
Send the	message: 29 to Kafka with topic temp_on_kafka!

Figure 12 - Run MQTT-KAFKA bridge

Copyright © FRACTAL Project Consortium	19 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

With the data in KAFKA's scope, it is time to run the bridge (Figure 13) that imports such data into SQL scope and also exports databased stored information into KAFKA's scope:

(t61) → T6	.1_Demo2 python psql-kafka-bridge.py
Sent the Ou	lu temperature 18 to topic postgresql
Sent the Ou	lu temperature 18 to topic postgresql
Received ci	ty temperature: 21
Sent the Ou	lu temperature 21 to topic postgresql
Sent the Ou	lu temperature 21 to topic postgresql
Sent the Ou	lu temperature 21 to topic postgresql
Received ci	ty temperature: 8
Sent the Ou	lu temperature 8 to topic postgresql
Sent the Ou	lu temperature 8 to topic postgresql
Sent the Ou	lu temperature 8 to topic postgresql
Received ci	ty temperature: 3

Figure 13 - Run SQL-KAFKA bridge

At this point, the only missing entity to run is the data consumer (Figure 14), which is, in turn, the entity that allows us to validate that the data pipeline is working correctly:

(t61) →	T6.1_Demo2	python consumer.py
Message	from topic:	temp_on_kafka VALUE: b'18'
Message	from topic:	postgresql VALUE: b'18'
Message	from topic:	postgresql VALUE: b'18'
Message	from topic:	postgresql VALUE: b'18'
Message	from topic:	temp_on_kafka VALUE: b'9'
Message	from topic:	postgresql VALUE: b'9'
Message	from topic:	postgresql VALUE: b'9'
Message	from topic:	postgresql VALUE: b'9'
Message	from topic:	temp_on_kafka VALUE: b'16'
Message	from topic:	postgresql VALUE: b'16'
Message	from topic:	postgresql VALUE: b'16'

Figure 14 - Run data consumer

Now that we have all the entities involved in the test up and running, it is time to validate that the data workflow is the expected one. For this task, capture of all entities generating/moving data is presented in order to analyse it:

Copyright © FRACTAL Project Consortium	20 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Send a message to MQTT: 25 to topic temp_measured	
Send a message to MQTT: 5 to topic temp_measured	←──
Send a message to MQTT: 19 to topic temp_measured	<u> </u>
Send a message to MQTT: 7 to topic temp_measured	
Send a message to MQTT: 8 to topic temp_measured	
send a message co nerro co copie cemp_measarea	Data areducar
	Data producer
Received MOTT message: 5	▲
Send the message: 5 to Kafka with topic temp_on_kafka!	Ì
Received MQTT message: 19	
Send the message: 19 to Kafka with topic temp_on_kafka!	
Received MQTT message: 7	
Send the message: 7 to Kafka with topic temp_on_kafka!	
Received MQTT message: 8	
Send the message: 8 to Kafka with topic temp_on_kafka!	
	-KAFKA bridge
MQTT	KAI KA bridge
Received city temperature: 5	
Sent the Oulu temperature 5 to topic postgresql	←──
Sent the Oulu temperature 5 to topic postgresql	
Sent the Oulu temperature 5 to topic postgresgl	
Received city temperature: 19	_
Sent the Oulu temperature 19 to topic postgresql	
Sent the Oulu temperature 19 to topic postgresql	
Sent the Oulu temperature 19 to topic postgresql	
Received city temperature: 7	
Sent the Oulu temperature 7 to topic postgresql	
Sent the Oulu temperature 7 to topic postgresql	
Sent the Oulu temperature 7 to topic postgresql	
Received city temperature: 8	←──
Sent the Oulu temperature 8 to topic postgresql	
Sent the Oulu temperature 8 to topic postgresql	
Sent the Oulu temperature 8 to topic postgresql	
SQL	-KAFKA bridge
	4
Message from topic: temp_on_kafka VALUE: b'5'	←
Message from topic: postgresql VALUE: b'5'	
Message from topic: postgresql VALUE: b'5'	
Message from topic: postgresql VALUE: b'5'	
Message from topic: temp_on_kafka VALUE: b'19'	
Message from topic: postgresql VALUE: b'19'	
Message from topic: postgresql VALUE: b'19'	
Message from topic: postgresql VALUE: b'19'	
Message from topic: temp_on_kafka VALUE: b'7'	←
Message from topic: postgresql VALUE: b'7'	
Message from topic: postgresql VALUE: b'7'	
Message from topic: postgresql VALUE: b'7'	
Message from topic: temp_on_kafka VALUE: b'8'	←──
Message from topic: postgresql VALUE: b'8'	
Message from topic: postgresql VALUE: b'8'	
	Data consumer

Figure 15 - Testbed data analysis

As can be seen in the results above (Figure 15), each data generated by the data producer is received by MQTT-KAFKA bridge and forwarded to KAFKA. This forwarded data is received by the SQL-KAFKA bridge and stored into the SQL database. Finally, the value stored in the database is retrieved every second and sent to the KAFKA broker. At the consumer level, it is possible to validate that the value retrieved from MQTT and the one retrieve from SQL are ok.

The scripts used to perform these validation tests are not exactly the same as the received ones (from task T6.1). Some adaptations were necessary, but the fundamental idea of the tests can be carried out to validate that data exchange

Copyright © FRACTAL Project Consortium	21 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

mechanism designed in the project is viable and usable. Therefore, the tests are considered passed.

Copyright © FRACTAL Project Consortium	22 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

7 FRACTAL Edge Controller review

This chapter briefly describes the FRACTAL Edge Controller architecture implemented in task T6.2, which focuses on the development of a communication and system monitoring component to optimize the overall system resources of a group of FRACTAL nodes. Therefore, it is an open-source software component to provide the Edge platform with self-orchestration and independence mechanisms at various levels. Figure 5 shows a diagram with the architecture of the Edge Controller designed in Task T6.2. For more detailed information on the Edge Controller design and implementation, refer to the deliverable D6.2.

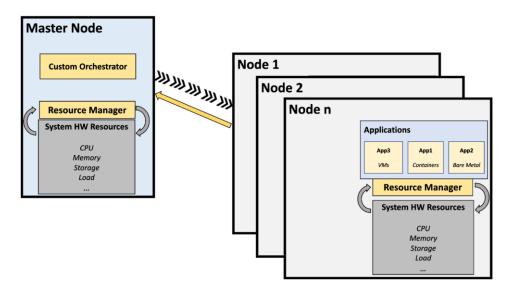


Figure 16: Multi-node Edge Controller architectural design (designed in task T6.2)

The main inputs to Task T6.2 are the following eigth components:

- 1. WP6T62-06 Orchestration (Edge Controller)
- 2. WP6T62-06 Orchestration (Agent Nodes Controller)
- 3. WP6T62-03 Run time Manager
- 4. WP6T62-01 Data Ingestion
- 5. WP6T62-0W MQTT Cloud comm. System --> Merged into Data Ingestion
- 6. WP6T62-02 Federated Data Collection
- 7. WP6T62-06 Low-end node orchestrator
- 8. WP6T62-0X Hardware Edge Controller

Through chapter 8, the validation report of each component will be presented. Here is a summary of each component:

WP6T62-06 Orchestration (Edge Controller)

The Edge Controller is an autonomous orchestrator for containers to support K8S, Docker, or orchestrator-less Fractal nodes.

Copyright © FRACTAL Project Consortium	23 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

WP6T62-06 Orchestration (Agent Nodes Controller)

The Agent Nodes Controller is part of the orchestration and is an add-on to the Edge Controller that provides Edge Nodes with the ability to orchestrate tasks and assign them to available nodes.

WP6T62-03 Run time Manager

The Runtime Manager is a component developed to coordinate and manage task scheduling and load balancing operations between modules in one or more fractal nodes at runtime. The purpose of the Runtime Manager is to enable communication and data dispatch among the various components installed on the node, and to manage the load balancing operations, when needed, by assigning the execution of the activities to a different instance of the Runtime Manager module installed on another node.

WP6T62-01 Data Ingestion

The Data Ingestion component provides data ingestion and data streaming processing tools for the High-End and Mid-End Fractal nodes.

WP6T62-0W MQTT Cloud comm. System

This component is part of the WP6T62-01 Data Ingestion component.

WP6T62-02 Federated Data Collection

The federated Data Collection component provides data storage capabilities for the Fractal High-End and Mid-End nodes.

WP6T62-06 Low-end node orchestrator

As a result of WP3, Nuttx RTOS was ported to the PULP low-end systems to offer a Posix completable application environment and reported in D3.6. In WP6 an IoT Hub was integrated into the Nuttx. In this way, nodes connect to the cloud, where they are orchestrated based on their identity.

WP6T62-0X Hardware Edge Controller

The hardware Edge Controller based on a network-on-chip (NoC) multicore architecture was developed in WP4 and reported in D4.4. It supports heterogeneous cores connected via NoC. In order to allow multiple nodes communication within WP6, one of the NoC cores was devoted to function as a hardware gateway controller. It allows communication between on-chip and off-chip network, as described in D6.2.

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8 Validation test plan and implementation of the Edge Controller

This section will be dedicated to the definition of the validation tests and their implementation for the validation of the Edge Controller and the components implemented in task T6.2. For further information on each component, refer to D6.2 and components from task T6.2 in the FRACTAL GitHub repository. In addition, to facilitate the reading of this section a complete overview of each test can be found in the Annexes (chapter 14).

8.1 Orchestration (Edge Controller)

According to D6.2, the main functionality of this component is to monitor (separately) the status of "n" number of nodes where it is deployed. It is in charge of collecting all the resources information inside each of the Fractal nodes and was designed following a modular design. This software is composed of two main modules: the metrics exporter and the resource manager. These modules are managed by the custom Edge Orchestrator, which is designed to modify and take actions on both Kubernetes and Docker nodes, as well as user-defined orchestrators, by using the information from the metrics-exporter.

It is based on the architecture illustrated in Figure 5.

8.1.1 Test planification

8.1.1.1 Define the testing scope and identify the functionality that needs to be tested

Since this component and the next one (WP6T62-06-mid-range-orchestration "Agent Nodes Controller") has the same code but were delivered as individual components in different repositories, we have differentiated their tests by adding <u>EC</u> (e.g. T01_WP6T62-06_EC) when it comes to Edge Controller and <u>ANC</u> (e.g. T01_WP6T62-06_ANC) when it comes to Agent Nodes Controller.

After careful study of this component, 4 test cases have been identified, which can be carried out.

- 1. Installation.
- 2. Validate if the master node can monitor several (2) workers' nodes.
- 3. Validate through the REST API if the "metrics exporter" is working properly.
- 4. Test how the resource manager behaves if the nodes are stressed.

Copyright © FRACTAL Project Consortium	25 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

a) T01_WP6T62-06_EC - Testing the installation of the component

Validation test			
Test ID	T01_WP6T62-06_EC		
Test type	Functional-Installation		
Test name	Testing the installation of the component		
Date	15/11/2022		
Tester's Name	Ana Bautista		
Test scope or objective			
The objective of this test is to validate if the Edge Controller Orchestrator can be installed without any issues.			

The objective of this test is to validate if the Edge Controller Orchestrator can be installed without any issues.

Table 3 - Validation Test T01_WP6T62-06_EC

b) T02_WP6T62-06_EC - Testing if the master node can monitor several (2) workers' nodes

Validation test			
Test ID	T02_WP6T62-06_EC		
Test type	Functional		
Test name	Testing if the master node can monitor several (2) workers' nodes		
Date	20/01/2023		
Tester's Name	s Name Ana Bautista		
Test scope or objective			
The objective of this test is to validate if the master node can monitor two workers' nodes.			

Table 4 - Validation Test T02_WP6T62-06_EC

c) T03_WP6T62-06_EC - Testing through the REST API if the metrics exporter is working properly

Validation test		
Test ID	T03_WP6T62-06_EC	
Test type	Functional	
Test name	Testing through the REST API if the metrics exporter is working properly	
Date	15/11/2022	
Tester's Name	Ana Bautista	
Test scope or objective		
The objective of this test is to validate through the REST API if the metrics exporter is working properly.		

Table 5 - Validation Test T03_WP6T62-06_EC

d) T04_WP6T62-06_EC - Testing how the resource manager behaves if the nodes are stressed

Validation test			
Test ID	04_WP6T62-06_EC		
Test type	Functional		
Test name	Testing how the resource manager behaves if the nodes are stressed		
Date	15/11/2022		
Tester's Name	Ana Bautista		
Test scope or objective			
The objective of this test is to observe how the resource manager behaves if the nodes are stressed.			

Table 6 - Validation Test T04_WP6T62-06_EC

Copyright © FRACTAL Project Consortium	26 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.1.2 Test case development

In this section, the steps to be followed to carry out the validation tests for the Edge Controller were identified.

8.1.2.1 T01_WP6T62-06_EC - Testing the installation of the component

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.

Table 7 - Steps for Validation Test T01_WP6T62-06_EC

8.1.2.2 T02_WP6T62-06_EC - Testing if the master node can monitor several (2) workers' nodes

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.
Step 5	Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes.
Step 6	Deploy the metrics exporter container on each of the worker nodes.
Step 7	Review the logs from the resource manager (deployed on the master node).

Table 8 - Steps for Validation Test T02_WP6T62-06_EC

8.1.2.3 T03_WP6T62-06_EC - Testing through the REST API if the metrics exporter is working properly

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.
Step 5	Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes.
Step 6	Deploy the metrics exporter container on each of the worker nodes.
Step 7	Go to: http:// <node_ip>:61208/api/3/cpu</node_ip>

Table 9 - Steps for Validation Test T03_WP6T62-06_EC

Copyright © FRACTAL Project Consortium	27 of 149

1992 P.C.	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
X-X- <i>X</i>	Del. Code	D6.4

8.1.2.4 T04_WP6T62-06_EC - Testing how the resource manager behaves if the nodes are stressed

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.
Step 5	Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes.
Step 6	Deploy the metrics exporter container on each of the worker nodes.
Step 7	Install stress-ng on one of the worker nodes.
Step 8	Execute the command stress-ngcpu 8timeout 60s which will stress the node for 60 seconds.
Step 9	Review the logs from the resource manager (deployed on the master node).
Step 10	Check the alerts and the metrics in the logs of the resource manager.

Table 10 - Steps for Validation Test T04_WP6T62-06_EC

8.1.3 Test environment setup

The steps to install and configure the component can be found in the FRACTAL project GitHub repository: <u>https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator</u>

8.1.4 Test execution

The following tables show he results of the execution of each of the tests.

Some issues that were detected during the testing process are reported in remarks and some of them were already solved by developers.

Copyright © FRACTAL Project Consortium	28 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
X X X X X	Del. Code	D6.4

8.1.4.1 T01_WP6T62-06_EC - Testing the installation of the component

Successfully built d37fc3f57648 Successfully tagged custom-orchestrator:latest Successfully tagged custom-orchestrator:latest Success criteria No error messages/All partial results are as expected Test configuration https://github.com/project-fractal/WP6762-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 Intervention Intervention Intervention Intervention Intervention Intervention Intervention Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 Intervention Intervention Intervention Intervention Intervention Intervention Intervention Intervention Intervention	Results/Evidence	
No.error messages/All partial results are as expected Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt_get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 # WEGGINERS # WEGGINERS (# # # # # # # # # # # # # # # # # # #	Successfully	
Test observations Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions Test conditions The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 ************************************	Success criteria	
Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 Image: first conditions Image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first conditions Image: first conditions image: first conditions image: first condi	No error messages/A	All partial results are as expected
Test conditions The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04	Test observations	
Test conditions (master node). Two bugs were found during installation that have been reported and corrected by the developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 """""""""""""""""""""""""""""	Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
<pre>developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 ####################################</pre>	Test conditions	The metrics exporter, resource manager and custom orchestrator containers runs in the same node (master node).
<pre>Usage: flask run (OFTIONS) Try 'flask runhelp' for help. Error: While importing 'custom_orchestrator', an ImportError was raised: Traceback (most recent call last): File "/uur/local/lib/python3.5/site-packages/flask/cli.py", line 210, in locate_app importmodule_name) File "/home/custom-orchestrator.py", line 14, in <module> from uuis.orchestrate_K05 import orchestrates k05 orchestrate File "/home/custom-orchestrator/tuis/orchestrate k85.py", line 3, in <module> from aux_func import taint_node, scale_replicas, limit_node_resources, remove_node_resource_limitations ModuleNotFoundError: No module named </module></module></pre>	Remarks	<pre>1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 ************************************</pre>
		<pre>Usage: flask run [OFTIONS] Try 'flask runhelp' for help. Error: While importing 'custom_orchestrator', an ImportError was raised: Traceback (most recent call last): File "/\usr/local/lib/python3.9/stte-packages/flask/cli.py", line 218, in locate_app moottmodule_mame) File "/home/custom-orchestrator/custom_orchestrator.py", line 14, in <module> from utils.orchestrate_k83 import orchestrate ak88_orchestrate File "/home/custom-orchestrator/utils/orchestrate_k88.py", line 3, in <module> from utils.orchestrate_k83 import orchestrate_k88.py", line 3, in <module> from utils.orchestrate_k83 import ack_areale_replicas, linut_node_resource_limitations</module></module></module></pre>
Passad	Test result	
r asseu	Passed	

Table 11 - Results of the test T01_WP6T62-06_EC

Copyright © FRACTAL Project Consortium	29 of 149

255 E245	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
X X X X X	Del. Code	D6.4

8.1.4.2 T02_WP6T62-06_EC - Testing if the master node can monitor several (2) workers' nodes

Results/Evidence	
Master node:	
CONTAINER ID 1110afe4abf6 r 27a41d81626b r	ge-controller-orchestrator git:(WP6762-06-edge-orchestrator)
Vorker node 1:	
CONTAINER ID	ge-controller-orchestrator git:(WP6T62-06-edge-orchestrator) docker ps IMAGE COMMAND CREATED STATUS PORTS NAMES metrics-exporter "/bin/sh -c 'glances" 22 hours ago Up 22 hours peaceful_gauss
Norker node 2:	
CONTAINER ID I	e-controller-orchestrator git:(WP6T62-0 <mark>6-edge-orchestrator</mark>) docker ps MAGE COMMAND CREATED STATUS PORTS NAMES etrics-exporter "/bin/sh -c 'glances" 22 hours ago Up 22 hours pedantic_antonelli
Review logs fron	n the resorce manager:
2023- 2023-	02-16 14:56:12,127 02-16 14:56:12,127 02-16 14:56:12,127 10-05ger - INFO - fractal-K830.jpd.ikerlan.es Total memory: 15.62395576 Gb 02-16 14:56:12,127 10-05ger - INFO - fractal-K830.jpd.ikerlan.es Available memory: 15.62395576 Gb 02-16 14:56:12,127 10-05ger - INFO - fractal-K830.jpd.ikerlan.es Available memory: 15.62395576 Gb 02-16 14:56:12,127 10-16 14:56:12,127 10-16 14:56:12,127 10-16 14:56:12,127 10-16 14:56:12,728 10-16 14:56:12,738 10-16 14:56:12,745 10-16 14:56:12,75 10-16 14:56:12,75 10-16 14:56:12,75 10-16 14:56:12,75 10-16 14:56:12
Success criteria	
-	es/All partial results are as expected
est observation	
Test configuratic	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node The metrics exporter runs in the two worker's nodes.
Remarks	Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected).
Test result	
Passed	

Table 12 - Results of the test T02_WP6T62-06_EC

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.1.4.3 T03_WP6T62-06_EC - Testing through the REST API if the metrics exporter is working properly

Results/Evidence	
Master node:	
< →	😋 🛦 No es seguro NODE IP :61208/арі/3/сри 🔤 🖄 🛠 🖪 🚱 🤫 🌟 🔲 祸 🗄
"softirg	: 1.1, "user": 0.7, "nice": 0.0, "system": 0.4, "idle": 98.8, "iowait": 0.1, "irq": 0.0, ": 0.0, "steal": 0.0, "guest": 0.0, "guest_nice": 0.0, "time_since_update": 1128.0130908489227, ": 8, "ctx_switches": 7384901, "interrupts": 3861801, "soft_interrupts": 838841, "syscalls": 0}
Worker node 1:	
\leftarrow \rightarrow	C 🛦 No es seguro NODE_IP 61208/api/3/cpu 🔄 🔄 🛧 🛂 🕼 😨 🧑 🌧 🔲 🖪 :
"softirq	0.8, "user": 0.7, "nice": 0.0, "system": 0.2, "idle": 99.1, "iowait": 0.0, "irq": 0.0, ': 0.0, "steal": 0.0, "guest": 0.0, "guest_nice": 0.0, "time_since_update": 3.140522003173828, : 8, "ctx_switches": 5630, "interrupts": 3113, "soft_interrupts": 1195, "syscalls": 0}
Worker node 2:	
\leftarrow \rightarrow	C 🔺 No es seguro 👔 NODE IP (61208/api/3/cpu 🖉 🖻 🚖 🛪 🦉 🚱 😵 🏞 🔲 🔕 🗄
"softirq"	0.9, "user": 0.4, "nice": 0.0, "system": 0.3, "idle": 99.3, "iowait": 0.0, "irq": 0.0, : 0.0, "steal": 0.0, "guest": 0.0, "guest_nice": 0.0, "time_since_update": 3.3217294216156006, : 8, "ctx_switches": 5666, "interrupts": 3047, "soft_interrupts": 1100, "syscalls": 0}
Success criteria	
No error messages/	All partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
Test conditions	The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes.
Remarks	The REST API exposed by the custom orchestrator is reached by the resource manager and provides
-	information about the nodes previously configured (as expected).
Test result	
Passed	

Table 13 - Results of the test T03_WP6T62-06_EC

Copyright © FRACTAL Project Consortium	31 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.1.4.4 T04_WP6T62-06_EC - Testing how the resource manager behaves if the nodes are stressed

Posults / Evidence	
Results/Evidence	
Master node: WP6T62-06-edge-control	llog optioning of (19676) AS also protochaston & deckon logs ton fristal
a wronoz-oo-euge-control	ller-orchestrator git:(WP6762-86-edge-orchestrator) 🛛 docker logs zen_feistel
Stressed worker node	: fractal-k8s0.ipd.ikerlan.es
	ontroller-orchestrator git:(WP6T62-06-edge-orchestrator) stress-ngcpu 8timeout 60s
	1051804] dispatching hogs: 8 cpu
	1051804] successful run completed in 60.07s (1 min, 0.07 secs)
Results 1:	
	: fractal-k8s0.ipd.ikerlan.es
Worker node: fractal-l	
Worker Houe. Hactar i	Kostipulikentines
2023-02-17 12:40:44,687 - logge 2023-02-17 12:40:44,687 - logge	er - INFO - fractal-k850.ipd.ikerlan.es Total CPU usage <mark>: 33.1 %</mark> er - INFO - fractal-k850.ipd.ikerlan.es Total memory: 16.783896576 Gb
2023-02-17 12:40:44,687 - logge 2023-02-17 12:40:44,687 - logge	er - 1NFO - fractal-x830.ipd.ikerlan.es Avsliable memory: 15.54231152000001 Gb er - 1NFO - fractal-x830.ipd.ikerlan.es Total memory usage: 7.4 %
2023-02-17 12:40:44,687 - logge 2023-02-17 12:40:44,687 - logge	er - 1NFO - fractal-k830.ipd.ikerlan.es Lond avg 1 min: 0.6962890625 % er - 1NFO - fractal-k830.ipd.ikerlan.es Alerts: []
2023-02-17 12:40:44,688 - logge 2023-02-17 12:40:44,792 - logge 2023-02-17 12:40:44,792 - logge	er - INFO - fractal-k8s1.ipd.ikerlan.es Total CPU usage 0.5 %
2023-02-17 12:40:44,792 - logge 2023-02-17 12:40:44,792 - logge	er – INFO – fractal-k851.ipdi.ikerlan.es Totalable memory: 15.575318528 6b er – INFO – fractal-k851.ipdi.kerlan.es Total memory usage: 7.2 %
2023-02-17 12:40:44,792 - logge 2023-02-17 12:40:44,792 - logge	er - 1NFO - fractal-k851.jpd.ikerlan.es Load avg 1 min: 0.0 % - 1NFO - fractal-k851.jpd.ikerlan.es Alerts: []
2023-02-17 12:40:59,935 - 10gge	
2023-02-17 12:40:59,933 - logge	er - 1NFO - fractal-k890.jpd.ikerlan.es Total memory: 16.763906576 Go er - 1NFO - fractal-k890.ipd.ikerlan.es Total memory: 15.541850112 Gb er - 1NFO - fractal-k890.ipd.ikerlan.es Total memory usage: 7.4 ≸
2023-02-17 12:40:59,933 - logge	er - INFO - fractal-K850.ipd.ikerian.es load avg 1 min: 2.314453125 % er - INFO - fractal-K850.ipd.ikerian.es Alerts: [[1676637569.0, -1, 'CRITICAL', 'CPU_TOTAL', 100.0, 100.0, 100.0, 100.0, 1, [], '', 'cpu_percent']]
2023-02-17 12:40:59,934 - logge 2023-02-17 12:40:59,934 - logge	er - INFO - Tainting node fractal-k8s0.ipd.ikerlan.es to avoid scheduling of any other containers
2023-02-17 12:40:59,934 - logge 2023-02-17 12:41:00.064 - logge	er - 1NFO - Tainted nodes: ('fractal-k890.jpd.ikerlan.es': 'Marchedule') er - 1NFO - Fractal-k81.jnd.ikerlan.es Total (PU)usage 0.6 %
2023-02-17 12:41:00,064 - logge 2023-02-17 12:41:00,064 - logge 2023-02-17 12:41:00,064 - logge	er - INFO - fractal-k851.jpd.ikerlan.es Total memory: 10 .705090970 Gb er - INFO - fractal-k851.jpd.ikerlan.es Available memory: 15.574115400000001 Gb
2023-02-17 12:41:00,065 - logge 2023-02-17 12:41:00,065 - logge	er - 1NFO - fractal-k851.ipdi.kerlan.es Load avg 1 min: 0.0 % pr - 1NFO - fractal-k851.ipdi.kerlan.es Alerts: []
2023-02-17 12:41:00,065 - logge	er - INFO - fractal-k8s1.ipd.ikerlan.es Alerts: [] er - INFO - Tainted modes: ('fractal-k8s8.ipd.ikerlan.es': 'NoSchedule')
Results 2:	
2023-02-17 12:41:30,417 -	logger - INFO - fractal-k8s0.ipd.ikerlan.es Total CPU usage: <mark>180.0 %)</mark> logger - INFO - fractal-k8s0.ipd.ikerlan.es Total memory: 16.78389576 Gb
2023-02-17 12:41:30,417 -	logger - INFO - fractal-k8s0.ipd.ikerlan.es Available memory: 15.539376128 Gb
2023-02-17 12:41:30,417 - 2023-02-17 12:41:30,417 -	
2023-02-1/ 12:41:30,41/ .04, 0.0, 0.0, 0.0], 'pid'	logger - INFO - fractal-k8s0.1pd.1kerlan.es Alerts: [[16/65/659.0, -1, 'CRITICAL', 'CPU_TOTAL', 100.0, 100.0, 100.0, 300.0, 3, : 1051806, 'num_threads': 1, 'memory_info': [7311360, 55607296, 3723264, 1728512, 0, 27832320, 0], 'nice': 0, 'ppid': 1051804, 't
1, 2, 3, 4, 5, 6, 7], 'nur , '', ''], 'username': 'met	m_ctx_switches': [1, 9550], 'memory_swap': None, 'tcp': None, 'udp': None, 'extended_stats': True, 'key': 'pid', 'time_since_upd: tricsexporter'}, {'cpu_percent': 99.8, 'io_counters': [0, 0, 0, 0, 0], 'name': 'stress-ng-cpu', 'status': 'R', 'cpu_times': [50.
7832320, 0], 'nice': 0, 'p; ame': 'metricsexporter'},	Logger - IntO - trattal-k80s.jpuikelan.es Loud av [Imli+4:5380-0023 % Logger - IntO - trattal-k80s.jpuikelan.es Alerts [[[L65637266, 372364, 172812, 0] 2783230, 0], nice: 0, ppid: 1051804, : 1051806, num_threads: 1, 'memory_info': [7311360, 55607296, 3723244, 172812, 0] 2783230, 0], nice: 0, ppid: 1051804, : ricesuporter }, ('cpu_percent: 9, 0, io: counters: [0, 0, 0, 6, 0], name: stress-mg-cpu, status: R, 'cpu_ters: [50, 0] : di: 1051804, memory_percent: 9, 93, io: counters: [0, 0, 0, 6, 0], name: stress-mg-cpu, status: R, 'cpu_ters: [50, 0] : di: 1051804, memory_percent: 9, 0435617555007130, gids: [1060, 1000], key: pid; 'time_pinet, verters: [50, 6, 0, 0, 0], 6, 0] : on_percent: 9.6435278349211, 'gids: [1000, 1000], key: pid; 'time_since_update: 15.1125326718571, comline: s
: 0, 'ppid': 1051804, 'memo ter'}], '', 'cpu percent']	ory_percent': 0.04392782073349211, 'gids': [1000, 1000, 1000], 'key': 'pid', 'time_since_update': 15.212553262710571, 'cmdline':
2023-02-17 12:41:30,418 - 2023-02-17 12:41:30,418 -]]]ogger - NARNING - CPU in node fractal-k8s0.ipd.ikerlan.es is over 80%: CPU = 100.0]ogger - INFO - Node fractal-k8s0.ipd.ikerlan.es is already tainted as NoSchedule
2023-02-17 12:41:30,418 - 2 2023-02-17 12:41:30,500 -	iosger - INFO - <mark>Hode fractai-K850.ipd.ikerian.es is aiready tainted as Noschedule</mark> logger - INFO - Tainted nodes: ('tractai-K850.ipd.ikerian.es <u>: Max-hadule'</u>) logger - INFO - fractai-K851.ipd.ikerian.es Totai CPU usage: (B ⊂ S
2023-02-17 12:41:30.500 -	logger - INFO - fractal-k8s1.ipd.ikerlan.es Total memory: 16.783896576 Gb
2023-02-17 12:41:30,500 -	logger - INFO - fractal-k8s1.ipd.ikerlan.es Available memory: 15.574810624000001 Gb logger - INFO - fractal-k8s1.ipd.ikerlan.es Total memory usage: 7.2 %
023-02-17 12:41:30.501 -	logger - INFO - fractal-k8s1.ipd.ikerlan.es Load avg 1 min: 0.0 % logger - INFO - fractal-k8s1.ipd.ikerlan.es Alerts: []
2023-02-17 12:41:45,597 - 1	logger - INFO - Tainted nodes: {'fractal-k8s0.ipd.ikerlan.es': 'NoSchedule'} logger - INFO - fractal-k8s0.ipd.ikerlan.es Total CPU usage: <mark>61.5 %</mark>
2023-02-17 12:41:45,598 - 2 2023-02-17 12:41:45,598 - 2	logger - INFO - fractal-k8s0.ipd.ikerlan.es Total memory: 16.783896576 Gb logger - INFO - fractal-k8s0.ipd.ikerlan.es Available memory: 15.564402688000001 Gb
023-02-17 12:41:45.598 -	logger - INFO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.3 % logger - INFO - fractal-k8s0.ipd.ikerlan.es Load avg 1 min: 4.677734375 %
nity': [0, 1, 2, 3, 4, 5, 6	logger - 1MFO - fractal-k880.jpd.ikerlan.es Alerts: [[16/6637658]0, 1676537058.0, 167853705.0, (CRIITAL', (CPU_107AL', 100.0, 10
3512, 0, 27832320, 0], 'nic	ce': 0, 'ppid': 1051804, 'memory_percent': 0.04356175556071301, 'gids': [000, 1000, 1000], 'key': 'pid', 'time_since_update': 1:
], 'nice': 0, 'ppid': 105	1804, 'memory_percent': 0.04392782073349211, 'gids': [1000, 1000], 'key': 'pid', 'time_since_update': 15.212553262710571,
ricsexporter'}], '', 'cpu 023-02-17 12:41:45,598 -	logger - INFO - Node fractal-k8s0.ipd.ikerlan.es untainted. All parameters are below critical tresholds.
Success criteria	
	I partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
	The metrics exporter, resource manager and custom orchestrator containers run in the master node.
Test conditions	The metrics exporter runs in the two worker's nodes.
	The node called fractal-k8s0.ipd.ikerlan.es is the node that was stressed.
	As it can be observed in the "Posults 1" screenshot, when the CDU usage of a node is over 200/ it is an iteration
	As it can be observed in the "Results 1" screenshot: when the CPU usage of a node is over 80% it is considered
Remarks	as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if
Remarks	as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if any of the monitored resources are above some fixed thresholds, that node is no longer able to perform any
	as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if
Remarks Test result Passed	as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if any of the monitored resources are above some fixed thresholds, that node is no longer able to perform any

Table 14 - Results of the test T04_WP6T62-06_EC

	Copyright © FRACTAL Project Consortium	32 of 149	
--	--	-----------	--

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2 Orchestration (Agent Nodes Controller)

This component is part of the Edge Controller, so it has to do with orchestration, in this case of "tasks". The name of this component is **WP6T62-06-mid-range-orchestration** and according to the developers' documentation, this software consists of three main components based on the architecture illustrated in Figure 176:

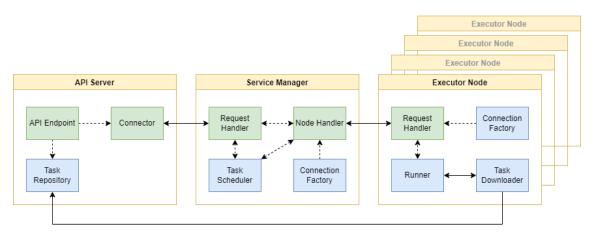


Figure 17: WP6T62-06-mid-range-orchestration architecture (designed in task T6.2)

The main functionality of this component is to orchestrate tasks execution on the available "Executor Nodes". For more details on how this component works, see the D6.2 deliverable and the GitHub repository <u>https://github.com/project-fractal/WP6T62-06-mid-range-orchestration</u>.

8.2.1 Test planification

8.2.1.1 Define the testing scope and identify the functionality that needs to be tested

After careful study of this component, 6 test cases have been identified, which can be carried out.

- 1. Installation.
- 2. Basic orchestration functionality.
- 3. Validate that a running task can be deleted.
- 4. Validate that a running task can be stopped and started again.
- 5. Validate that running multiple tasks is possible and list their state.
- 6. Validate the behaviour of the orchestrator with multiple Executor Nodes.

Copyright © FRACTAL Proj	ect Consortium 33 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

a) T01_WP6T62-06_ANC - Testing that the Agent Nodes Controller can be installed without any issues

Validation test		
Test ID	T01_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing that the Agent nodes controller can be installed without any issues	
Date	20/01/2023	
Tester's Name	ster's Name Ana Bautista, Adrian Moran	
Test scope or objective		
The objective of the test is to validate if the Agent nodes controller can be installed without any issues.		

The objective of the test is to validate if the Agent nodes controller can be installed without any issues.

Table 15 - Validation Test T01_WP6T62-06_ANC

b) T02_WP6T62-06_ANC - Testing basic orchestration functionality.

Validation test		
Test ID	T02_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing WP6T62-06-mid-range-orchestration component	
Date	09/02/2023	
Tester's Name	ster's Name Ana Bautista, Adrian Moran	
Test scope or objective		
The objective of the test is validate basic orchestration functionality.		

Table 16 - Validation Test T02_WP6T62-06_ANC

c) T03_WP6T62-06_ANC - Testing that a running task can be deleted

Validation test		
Test ID	T03_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing WP6T62-06-mid-range-orchestration component	
Date	09/02/2023	
Tester's Name	r's Name Ana Bautista, Adrian Moran	
Test scope or objective		
The objective of the test is validate that a running task can be deleted.		

Table 17 - Validation Test T03_WP6T62-06_ANC

d) T04_WP6T62-06_ANC - Testing that a running task can be stopped and started again

	Validation test	
Test ID	T04_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing WP6T62-06-mid-range-orchestration component	
Date	09/02/2023	
Tester's Name Ana Bautista, Adrian Moran		
Test scope or objective		
The objective of the	e test is validate that a running task can be stopped and started again.	

Table 18 - Validation Test T04_WP6T62-06_ANC

Copyright © FRACTAL Project Consortium	34 of 149

1992 P.C.	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

e) T05_WP6T62-06_ANC - Testing that a running multiple tasks is possible and list their state

	Validation test		
Test ID	T05_WP6T62-06_ANC		
Test type	Functional		
Test name	Testing WP6T62-06-mid-range-orchestration component		
Date	9/02/2023		
Tester's Name Ana Bautista, Adrian Moran			
Test scope or objective			
The objective of the te	est is validate that a running multiple tasks is possible and list their state		

The objective of the test is validate that a running multiple tasks is possible and list their state.

 f) T06_WP6T62-06_ANC - Testing the behaviour of the orchestrator with multiple Executor Nodes

	Validation test	
Test ID	T06_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing WP6T62-06-mid-range-orchestration component	
Date	09/02/2023	
Tester's Name Ana Bautista, Adrian Moran		
Test scope or objective		
The objective of the	e test is validate the behaviour of the orchestrator with multiple Executor Nodes.	

Table 20 - Validation Test T06_WP6T62-06_ANC

8.2.2 Test case development

In this section, the steps to be followed to carry out the validation tests for the Agent Nodes Controller were identified.

^{8.2.2.1} T01_WP6T62-06_ANC - Testing that the Agent Nodes Controller can be installed without any issues

Steps		
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).	
Step 2	Launch backend-service-manager (Service Manager).	
Step 3	Launch API Server.	
Step 4	Launch frontend-service-manager (Executor Node).	

Table 21 - Steps for Validation Test T01_WP6T62-06_ANC

8.2.2.2 T02_WP6T62-06_ANC - Testing basic orchestration functionality

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create a new task.

Table 22 - Steps for Validation Test T02_WP6T62-06_ANC

		Copyright © FRACTAL Project Consortium	35 of 149
--	--	--	-----------

Table 19 - Validation Test T05_WP6T62-06_ANC

1992 P.C.	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.2.3 T03_WP6T62-06_ANC - Testing that a running task can be deleted

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create a new task and delete it.

Table 23 - Steps for Validation Test T03_WP6T62-06_ANC

8.2.2.4 T04_WP6T62-06_ANC - Testing that a running task can be stopped and started again

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create a new task, stop and then start it.

Table 24 - Steps for Validation Test T04_WP6T62-06_ANC

8.2.2.5 T05_WP6T62-06_ANC - Testing that a running multiple tasks is possible and list their state

Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create a new tasks.

Table 25 - Steps for Validation Test T05_WP6T62-06_ANC

8.2.2.6 T06_WP6T62-06_ANC - Testing the behaviour of the orchestrator with multiple Executor Nodes

Steps		
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).	
Step 2	Launch backend-service-manager (Service Manager).	
Step 3	Launch API Server.	
Step 4	Launch two instances of frontend-service-manager (Executor Node).	
Step 5	From a different device, use REST API to create two simultaneous tasks.	

Table 26 - Steps for Validation Test T06_WP6T62-06_ANC

8.2.3 Test environment setup

The steps to install and configurate the component can be found in the FRACTAL project repository: <u>https://github.com/project-fractal/WP6T62-06-mid-range-orchestration</u>

Copyright © FRACTAL Project Consortium	36 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.4 Test execution

The following tables show the results of the execution of each of the tests.

Some issues that were detected during the testing process are reported in remarks and were also reported as feedback to developers.

8.2.4.1 T01_WP6T62-06_ANC - Testing that the Agent Nodes Controller can be installed without any issues

Results/Evidence		
{"version":	tal-qemu:-\$ curl -X GET http://localhost:5001/api/v1/tasks	
isks":[]])	
cess criteria		
No error messages/All partial results are as expected		
Test observations		
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration	
Test conditions	API Server, Service Manager and Executor Node runs in the same node.	
Remarks	All the subcomponents are up and running.	
Test result		
Passed		

Table 27 - Results of the test T01_WP6T62-06_ANC

Copyright © FRACTAL Project Consortium	37 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.4.2 T02_WP6T62-06_ANC - Testing basic orchestration functionality

Results/Evidence	
Client:	
→ task curl -X POST h {"status":"task: test	ttp://172.16.58.4:5001/api/v1/tasks/test -F file=@test_task.py -F "cmd=test_task.py" -F "rt=" created successfuly."}
API:	
	09/Feb/2023 11:13:59] "DELETE /api/v1/tasks/test HTTP/1.1" 200 - 09/Feb/2023 11:14:58] "POST /api/v1/tasks/test HTTP/1.1" 201 -
Executor Node:	
('task', 71, 'json 2023-02-09 11:14:5 2023-02-09 11:15:0 2023-02-09 11:15:0 2023-02-09 11:15:0	8,501 INFO heartbeat received ', {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 8,514 INFO running task: test 0,504 INFO heartbeat received 2,507 INFO heartbeat received 3,535 INFO task: test completed with return code: 0 4,508 INFO heartbeat received
Success criteria	
No error messages/Al	l partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration
Test conditions	API Server, Service Manager and Executor Node runs in the same node.
Remarks	Basic workflow is completed successfully.
Test result	
Passed	

Table 28 - Results of the test T02_WP6T62-06_ANC

8.2.4.3 T03_WP6T62-06_ANC - Testing that a running task can be deleted

Results/Evidence		
Client:		
	LETE http://172.16.58.4:5001/api/v1/tasks/test {"status":"ok"},"status":"task: test deleted successfuly from API host."}	
API:		
172.16.105.43	[09/Feb/2023 11:17:23] "DELETE /api/v1/tasks/test HTTP/1.1" 200 -	
Cusas a suita via		
Success criteria		
No error messages/A	II partial results are as expected	
Test observations		
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration	
Test conditions	API Server, Service Manager and Executor Node runs in the same node.	
	Task is deleted from API Server repository succesfully.	
Remarks	Deleted task is not deleted from Executor node .tasks folder, which can lead to problems	
	like lack of storage or DoS attacks.	
Test result		
Passed		

Table 29 - Results of the test T03_WP6T62-06_ANC

Copyright © FRACTAL Project Consortium	38 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.4.4 T04_WP6T62-06_ANC - Testing that a running task can be stopped and started again

Results/Evidence
Client:
<pre>- task curl -X POST http://172.16.58.4:5001/api/v1/tasks/test -F file=@test_task.py -F "cmd=test_task.py" -F "rt= {"status":"task: test created successfuly."} - task curl -X POST http://172.16.58.4:5001/api/v1/tasks/test/stop {"status":"ok"} + task curl -X POST http://172.16.58.4:5001/api/v1/tasks/test/start {"status":"ok"} + task curl -X DELETE http://172.16.58.4:5001/api/v1/tasks/test {"backend-status":{"status":"ok"} </pre>
API:
172.16.105.43 [09/Feb/2023 11:18:35] "POST /api/v1/tasks/test HTTP/1.1" 201 - 127.0.0.1 [09/Feb/2023 11:18:35] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - 172.16.105.43 [09/Feb/2023 11:18:45] "POST /api/v1/tasks/test/stop HTTP/1.1" 200 - 172.16.105.43 [09/Feb/2023 11:18:48] "POST /api/v1/tasks/test/start HTTP/1.1" 200 - 127.0.0.1 [09/Feb/2023 11:18:48] "POST /api/v1/tasks/test/download HTTP/1.1" 200 - 127.0.0.1 [09/Feb/2023 11:18:48] "GET /api/v1/tasks/test/download HTTP/1.1" 200 -
Executor Node:
<pre>2023-02-09 11:18:34,783 INFO heartbeat received ('task', 71, 'json', {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 2023-02-09 11:18:35,786 INFO running task: test 2023-02-09 11:18:36,785 INFO heartbeat received 2023-02-09 11:18:42,794 INFO heartbeat received 2023-02-09 11:18:44,797 INFO heartbeat received 2023-02-09 11:18:44,797 INFO heartbeat received 2023-02-09 11:18:44,797 INFO heartbeat received 2023-02-09 11:18:44,797 INFO heartbeat received 2023-02-09 11:18:44,786 INFO task stopped 2023-02-09 11:18:45,786 INFO task: test terminated with return code: -15 2023-02-09 11:18:48,803 INFO heartbeat received 2023-02-09 11:18:48,803 INFO heartbeat received 2023-02-09 11:18:48,803 INFO heartbeat received ('task', 71, 'json', {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 2023-02-09 11:18:48,852 INFO running task: test 2023-02-09 11:18:52,809 INFO heartbeat received 2023-02-09 11:18:54,812 INFO heartbeat received 2023-02-09 11:18:54,812 INFO heartbeat received 2023-02-09 11:18:54,812 INFO heartbeat received 2023-02-09 11:18:54,812 INFO heartbeat received 2023-02-09 11:18:55,814 INFO hask' test terminated with return code: -15</pre>
Success criteria
No error messages/All partial results are as expected
Test observations
Test configuration https://github.com/project-fractal/WP6T62-06-mid-range-orchestration
Test conditions API Server, Service Manager and Executor Node runs in the same node.
Remarks Task lifecycle in correctly handled.
Fest result
Passed

Table 30 - Results of the test T04_WP6T62-06_ANC

Copyright © FRACTAL Project Consortium	39 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.4.5 T05_WP6T62-06_ANC - Testing that a running multiple tasks is possible and list their state

Results/Evidence	
<pre>{"tasks":[]} task curl -X POST H {"status":"task: test task curl -X GET ht {"tasks":["test"]} task curl -X POST H {"status":"task: test: task curl -X GET ht {"tasks":["test","test task curl -X GET ht {"status":"ok", "task_s</pre>	:tp://172.16.58.4:5001/api/v1/tasks/test/status status":"running"} :tp://172.16.58.4:5001/api/v1/tasks/test2/status
+ task	
172.16.105.43 127.0.0.1 - [09/ from server {'stat 172.16.105.43 from server {'stat 172.16.105.43 172.16.105.43 172.16.105.43 172.16.105.43	<pre>[09/Feb/2023 11:21:41] "GET /api/v1/tasks HTTP/1.1" 404 - [09/Feb/2023 11:21:49] "POST /api/v1/tasks/test HTTP/1.1" 201 - /Feb/2023 11:21:50] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - tus': 'ok', 'tasks': ['test']} [09/Feb/2023 11:21:55] "GET /api/v1/tasks HTTP/1.1" 200 - [09/Feb/2023 11:21:55] "POST /api/v1/tasks/test2 HTTP/1.1" 201 - tus': 'ok', 'tasks': ['test', 'test2']} [09/Feb/2023 11:21:59] "GET /api/v1/tasks HTTP/1.1" 200 - [09/Feb/2023 11:22:12] "GET /api/v1/tasks/test/status HTTP/1.1" 200 - [09/Feb/2023 11:22:12] "GET /api/v1/tasks/test/status HTTP/1.1" 200 - [09/Feb/2023 11:22:16] "GET /api/v1/tasks/test2/status HTTP/1.1" 200 -</pre>
Success criteria	
• •	partial results are as expected
Test observations	https://with.com/ansight.forstal/NURCTC2.0C.mid.ansag.org/ansights/
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration
Test conditions	API Server, Service Manager and Executor Node runs in the same node. It is possible to list the tasks.
Remarks	<pre>It is possible to add multiple tasks and get their status. A bug has occured generating two task, one named "test" and other one named "test2". Executor Node computes the execution path from name, and since it does not deletes the old tasks, this leads to failure:</pre>
	This way of computing task_dir is bugged. If the same node has executed in their lifetime a tasks called "test" and "test2", there won't be any chance of running "test" task, since this piece of code will always select last "test2-timestamp" folder: drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:18 test-1675941515 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:18 test-1675941528 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:21 test-1675941528 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:21 test-1675941879 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:24 test-1675941879 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:27 test-1675942024 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:30 test-1675942222 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:30 test-1675941833 drwxrwxr-x 6 ikerlan ikerlan 4096 Feb 9 11:30 venv + .tasks git:(main) X
Test result	

Table 31 - Results of the test T05_WP6T62-06_ANC

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.2.4.6 T06_WP6T62-06_ANC - Testing the behaviour of the orchestrator with multiple Executor Nodes

Results/Evidence			
Client:			
{"status":"task: test cre → task curl -X POST http	l -X POST http://172.16.58.4:5001/api/v1/tasks/test -F file=@test_task.py -F "cmd=test_task.py" -F "rt=" 'task: test created successfuly."} 'l -X POST http://172.16.58.4:5001/api/v1/tasks/ikerlan -F file=@test_task.py -F "cmd=test_task.py" -F "rt=" 'task: ikerlan created successfuly."}		
127.0.0.1 [09/Fe 172.16.105.43 [0	172.16.105.43 [09/Feb/2023 13:22:36] "POST /api/v1/tasks/test HTTP/1.1" 201 - 127.0.0.1 - [09/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - 172.16.105.43 [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - 127.0.0.1 - [09/Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200 -		
Executor Node 1:			
('task', 71, 'json', 2023-02-09 13:22:36,	2023-02-09 13:22:34,827 INFO heartbeat received ('task', 71, 'json', {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 2023-02-09 13:22:36,821 INFO running task: test 2023-02-09 13:22:36,828 INFO heartbeat received		
Executor Node 2:			
('task', 74, 'json', 2023-02-09 13:22:42,	737 INFO heartbeat received {'task_name': 'ikerlan', 'args_to_run': 'test_task.py', 'return_type': ''}) 823 INFO running task: ikerlan 739 INFO heartbeat received		
Success criteria			
U	rtial results are as expected		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration		
Test conditions	API Server, Service Manager and Executor Node runs in the same node.		
Remarks	Multiple nodes works fine.		
Test result			
Passed			

Table 32 - Results of the test T06_WP6T62-06_ANC

Copyright © FRACTAL Project Consortium	41 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3 Runtime Manager

The Runtime Manager coordinates and manages task scheduling and load balancing operation between modules in one or more FRACTAL nodes at runtime. It performs the scheduling of various operations which are entirely configurable. In addition, it provides load balancing capabilities using the interface with the Load Balancer component, sending the task execution to a different node.

8.3.1 Test planification

In this section Runtime Manager functionalities are defined as per D6.2 and, for each function, test cases are defined. We will have three test cases for the first function and two test cases for the second one.

8.3.1.1 Define the testing scope and identify the functionality that needs to be tested

The functionality that was identified for the Runtime Manager Component that needs to be tested are related to how to distribute the computational load and the task scheduling, they were defined as follows:

- 1. The Runtime Manager mustbe able to distribute the computational load;
- 2. Execution of configured task related to the task scheduling.

For the first functioningthree test cases are defined and shown below:

a) T01_WP6T62-03 - Testing interaction between nodes with local node overloaded

Validation test		
Test ID	T01_WP6T62-03	
Test type	Functional	
Test Name	Testing interaction between nodes with local node overloded	
Date	05/12/2022	
Tester's Name	Luca Visconti (Modis Consulting SRL)	
Test scope or objective		
The objective of this test is to validate the correct interaction and data exchange between nodes		

le objective of this test is to validate the correct interaction and data exchange between node

Table 33 - Validation Test T01_WP6T62-03

b) T02_WP6T62-03 - Testing interaction in the local node when the local node can perform the computation

Validation test		
Test ID	T02_WP6T62-03	
Test type	Functional	
Test Name	Testing interaction in the local node when the local node can perform the computation	
Date	05/12/2022	
Tester's Name	Luca Visconti (Modis Consulting SRL)	
Test scope or objective		
The objective of this test is to validate the correct interaction and data exchange in the local node		

Table 34 - Validation Test T02_WP6T62-03

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

c) T03_WP6T62-03 - Testing interaction between nodes with Node 1 and Node 2 overloaded

Validation test		
Test ID	T03_WP6T62-03	
Test type	Functional	
Test Name	Testing interaction between nodes with Node 1 and Node 2 overloaded	
Date	05/12/2022	
Tester's Name	Luca Visconti (Modis Consulting SRL)	
Test scope or objective		
The objective of this test is to validate the correct interaction and data exchange between nodes		

The objective of this test is to validate the correct interaction and data exchange between nodes

Table 35 - Validation Test T03_WP6T62-03

For the second functioning there are defined two test cases that are shown below:

d) T04_WP6T62-03 - Task Scheduling on the local node

Validation test		
Test ID	T04_WP6T62-03	
Test type	Functional	
Test Name	Task Scheduling on the local node	
Date	05/12/2022	
Tester's Name	Luca Visconti (Modis Consulting SRL)	
Test scope or objective		
The objective of this test is to validate the correct execution of the Task in the local node		

Table 36 - Validation Test T04_WP6T62-03

e) T05_WP6T62-03 - Task Scheduling on the remote node

Validation test		
Test ID	T05_WP6T62-03	
Test type	Functional	
Test Name	Task Scheduling on the remote node	
Date	05/12/2022	
Tester's Name	Luca Visconti (Modis Consulting SRL)	
Test scope or objective		
The objective of this test is to validate the correct execution of the tasks in the remote node		

Table 37 - Validation Test T05_WP6T62-03

8.3.2 Test case development

All test cases are based on the same architecture. The test environment is presented in detail in section 8.3.3.

There are three nodes and a Runtime Manager on each node ("RMx" on node "Nx").

N1, the local node, is on a Xilinx board, N2 and N3, the remote nodes are virtual machines.

Copyright © FRACTAL Project Consortium	43 of 149

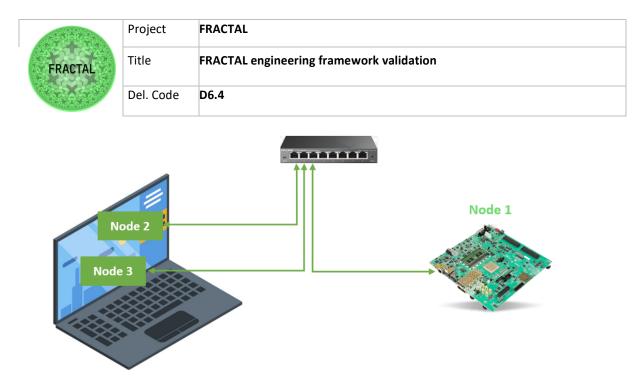


Figure 18 - Node interconnection

8.3.2.1 T01_WP6T62-03 - Testing interaction between nodes with local node overloaded

Step 1: Create the condition that overload the Node 1

For the overload condition of the N1 node, a simple "While True" cycle was executed on the python interpreter. This occupies resources at the node's processor. This procedure needs to be repeated until the processor is not overloaded (>85%). The procedure is described as follows:

- Open the command prompt of the N1 nodes and execute the following line:
 - python3 #to open the python interpreter
 - while True: #infinite cycle print(1)

As it is possible to notice in the table (using the htop command) the overload condition is verified and the node is in an overloaded state.

Step 2: Send the command of the execution flow to RM1 running "test_mqtt_published.py"

Expected Results: having generated the overload condition for the N1 node, RM1 will ask at Load Balancer instances which of the nodes can perform the workflow. The load balancer will respond with {"id_node":2}. RM1 will direct the workflows to the N2 node as shown in the figures below.

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps	
Step 1	Create the condition that overload the Node 1 1 [
Step 2	Send the command of the execution flow to RM1 running "test_mqtt_published.py" root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}

Table 38 - Steps for Validation Test T01_WP6T62-03

8.3.2.2 T02_WP6T62-03 - Testing interaction in the local node when the local node can perform the computation

Step 1: Send the command of the execution flow to RM1 running "test_mqtt_published.py"

Expected Result: Having not generated the overload condition for N1 nodes, it will be to execute the computational load locally. RM1 will ask at load balancer instance what is the node that can execute the computational load, the load balancer respond with {"id_node": none}. RM1 will execute the flows on the node N1 as shown below in the figure.

Steps	
Step 1	Send the command of the execution flow to RM1 running "test_mqtt_published.py"
	root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py
	data published
	[rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}

Table 39 – Steps for Validation Test T02_WP6T62-03

8.3.2.3 T03_WP6T62-03 - Testing interaction between nodes with Node 1 and Node 2 overloaded

Step 1: Create the condition that overload the Node 1

For the overload condition of the N1 node, it was executed some simple "while True" cycles on the python interpreter. This occupies resources at the node's processor. This procedure needs to be repeated until the processor is not overloaded. The procedure is described as follows:

- Open the command prompt of the N1 nodes and execute the following line:
 - python3 #to open the python interpreter
 - while True: #infinite cycle print(1)

As is possible to notice in the table (using the htop command), the overload condition (>85%) is verified and the node is to be in an overloaded state.

Copyright © FRACTAL Project Consortium	

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Step 2: Create the condition that overload the Node 2

In the same way as Node N1, - has been performed the overload condition on Node N2.

Step 3: Send the command of the execution flow to RM1 running "test_mqtt_published.py"

Expected Results: having generated the overload condition for the N1 and N2 nodes, RM1 will ask at Load Balancer instances which of the nodes can perform the workflow. The load balancer will respond with {"id_node":3}. RM1 will direct the workflows to the N3 node as shown in the table.

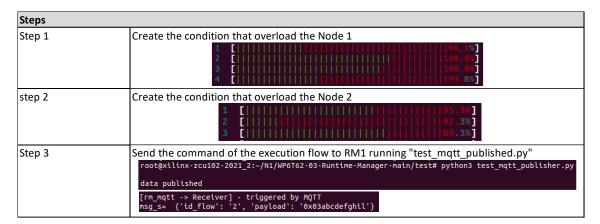


Table 40 – Steps for Validation Test T03_WP6T62-03

8.3.2.4 T04_WP6T62-03 - Task Scheduling on the local node

Step 1: Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with "id_flow=1".

Flow 1, as described in the configuration file "flows.conf", provides the following information:

1
TO component1 1
POST payload
TO component2 1
POST result1
TO component3 1
POST result1 result2

Figure 19 - Runtime Manager Flow 1

Expected Results: N1 executes correctly the flow 1 compared to configuration files. N1 receives the json message with "id_flow" and "payload" information, in particular "id_flow=1", it takes and executes the flow with the same id within the configuration file "flows.conf".

Step 2: Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id_flow=2".

Copyright © FRACTAL Project Consortium	46 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Flow 2, as described in the configuration file "flows.conf", provides the following information:



Figure 20 - Runtime Manager Flow 2

Expected Results: N1 executes correctly the flow 2 compared to configuration files. N1 receives the json message with "id_flow" and "payload" information, in particular "id_flow=2", it takes and executes the flow with the same id within the configuration file "flows.conf".

Step 3: Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id_flow=3".

Flow 3, as described in the configuration file "flows.conf", provides the following information:



Figure 21 - Runtime Manager Flow 3

Expected Results: N1 executes correctly the flow 3 compared to configuration files. N1 receives the json message with "id_flow" and "payload" information, in particular "id_flow=3", it takes and executes the flow with the same id within the configuration file "flows.conf".

Steps	
Step 1	Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with "id_flow=1". root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg s= {'id_flow': '1', 'payload': '0x03abcdefghil'}
Step 2	Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id_flow=2". root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}
Step 3	Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id_flow=3". root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '3', 'payload': '0x03abcdefghil'}

Table 41 – Steps for Validation Test T04_WP6T62-03

Copyright © FRACTAL Project Consortiu	m 47 of 149
---------------------------------------	-------------

1995200 A	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.2.5 T05_WP6T62-03 - Task Scheduling on the remote node

Step 1: Create the condition that overload the Node 1

For the overload condition of the N1 node, it was executed some simple "while True" cycles on the python interpreter. This occupies resources at the node's processor. This procedure needs to be repeated until the processor is not overloaded (>85%). The procedure is described as follows:

- Open the command prompt of the N1 nodes and execute the following line:
 - python3 #to open the python interpreter
 - while True: #infinite cycle print(1)

As is possible to notice in the table (using the htop command), the overload condition is verified and the node is to be in an overloaded state.

Step 1: Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with "id_flow=1".

Flow 1, as described in the configuration file "flows.conf", provides the following information:

1				
то	comp	onenti	۱1	
POS	т ра	yload		
то	comp	onenta	21	
POS	Т ге	onent2 sult1		
то	COMD	onent	31	
POS	Т ге	sult1	resu	ılt2

Figure 22 - Runtime Manager Flow 1

Expected Results: having generated the overload condition for the N1, RM1 will ask at Load Balancer instances which of the nodes can perform the workflow. The load balancer will respond with {"id_node": 2}. RM1 will direct the workflows to the N2 node as shown in the table. N2 executes correctly the flow 1 compared to configuration files. N2 receives the json message with "id_flow" and "payload" information, in particular "id_flow=1", it takes and executes the flow with the same id within the configuration file "flows.conf".

Step 2: Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id_flow=2".

Flow 2, as described in the configuration file "flows.conf", provides the following information:

Copyright © FRACTAL Project Consortium	48 of 149

		Project	FRACTAL
FRACTAL Title FRACTAL engineering framework validation	ing framework validation	Title	
Del. Code D6.4		Del. Code	



Figure 23- Runtime Manager Flow 2

Expected Results: having generated the overload condition for the N1, RM1 will ask at Load Balancer instances which of the nodes can perform the workflow. The load balancer will respond with {"id_node": 2}. RM1 will direct the workflows to the N2 node as shown in the table. N2 executes correctly the flow 2 compared to configuration files. N2 receives the json message with "id_flow" and "payload" information, in particular "id_flow=2", it takes and executes the flow with the same id within the configuration file "flows.conf".

Step 3: Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id_flow=3".

Flow 3, as described in the configuration file "flows.conf", provides the following information:



Figure 24 - Runtime Manager Flow 3

Expected Results: having generated the overload condition for the N1, RM1 will ask at Load Balancer instances which of the nodes can perform the workflow. The load balancer will respond with {"id_node": 2}. RM1 will direct the workflows to the N2 node as shown in the table. N2 executes correctly the flow 3 compared to configuration files. N2 receives the json message with "id_flow" and "payload" information, in particular "id_flow=3", it takes and executes the flow with the same id within the configuration file "flows.conf".

Copyright © FRACTAL Project Consortium	49 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps	
Step 1	Create the condition that overload the Node 1 1 [] [
Step 2	Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with "id_flow=1". root@xilinx-zcu102-2021_2:~/N1/WP0T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '1', 'payload': '0x03abcdefghil'}
Step 3	Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id_flow=2". root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}
Step 4	Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id_flow=3". root@xilinx-zcu102-2021_2:-/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '3', 'payload': '0x03abcdefghil'}

Table 42 - Steps for Validation Test T05_WP6T62-03

8.3.3 Test environment setup

The Test environment in the Runtime Manager Validation Test is configured using three different nodes:

- Local Node with id = 1 is the Zynq UltraScale+ ZCU102 board having Quad-Core Arm Cortex-A53 processor, CPU frequency up to 1.5GHz and 4GB of RAM Memory;
- Remote Node with id = 2 is the Virtual Machine Computer having 2 Core Intel i7-8650U processor, CPU Frequency up to 1.90 GHz and 2 GB RAM Memory;
- Remote Node with id = 3 is the Virtual Machine Computer having the same characteristics of Remote Node with id = 2.

The nodes have been interconnected using an ethernet connection. A schematic is reported in Figure 18 just presented in previous section.

To better understand the nomenclature, we have defined the following legend:

- RM $\{1, 2, 3\}$ = Runtime Manager on the node with id = 1, 2, 3;
- LB {1, 2, 3} = Load Balancer on the node with id = 1, 2, 3 related to RM {1, 2, 3}
- N {1, 2, 3} = Node with id = 1, 2, 3.

On each node were installed all the requirements specified on the GitHub repository of the Runtime Manager Component [<u>https://github.com/project-fractal/WP6T62-03-Runtime-Manager.git</u>].

Copyright © FRACTAL Project Consortium	50 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

After that, the configurations files of the Runtime Manager Component were modified as follows. In this way, a test environment was created within a local network.

The nodes N1, N2, N3 have been configured in the following way:

-N1:

RM1:

```
ip = "192.168.0.1"
ip api = "192.168.0.1"
port api = 7777
entrypoint api = "/startrm"
loadbalancer ip = "127.0.0.1"
loadbalancer port = 7776
loadbalancer endpoint = "LB/id_node"
```

LB1:

```
id = 1
ip = 192.168.0.1
ip api = 127.0.0.1
port_api = 7776
ip_broker = 192.168.0.2
```

-N2:

RM2:

```
ip = 192.168.0.2
ip api = "192.168.0.2"
port api = 8888
entrypoint = "/nodo2"
loadbalancer ip = "127.0.0.1"
loadbalancer port = 8886
loadbalancer endpoint = "LB/id_node"
```

LB2:

id = 2 ip = 192.168.0.2 ip_api = 127.0.0.1 port_api = 8887 ip_broker = "localhost"

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

-N3:

RM3:

```
ip = 192.168.0.2
ip api = "192.168.0.2"
port api = 9999
entrypoint = "/nodo3"
loadbalancer ip = "127.0.0.1"
loadbalancer port = 9998
loadbalancer endpoint = "LB/id_node"
```

LB3:

id = 3 ip = 192.168.0.2 ip_api = 127.0.0.1 port_api = 9998 ip_broker = "localhost"

As described in the github repository, to execute Runtime Manager has been used "rm_apy.py" and "rm_mqtt.py" scripts. For example, on the node N1:

- Runtime Manager listens on API REST



Figure 25 - Run "rm_api.py" script

- Runtime Manager listens on MQTT

root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/runtime_manager# python3 rm_mqtt.py

Figure 26 - Run "rm_mqtt.py" script

Furthermore, the "test_mqtt_publisher.py" script has been used to send a JSON message with the following field:

- "id_flow", describes the execution flow
- "payload", describes the data that needs to be exchanged between nodes.

Modifying the "id_flow" field with {1, 2, 3} is possible to execute different workflows.

To create the condition that overloads the nodes it is possible to execute a blocking function (ex: a few instances of a while True cycle) The results of this action is a processor overloaded.

Copyright © FRACTAL Project Consortium	52 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.4 Test execution

8.3.4.1 T01_WP6T62-03 - Testing interaction between nodes with local node overloaded

Results/Evidence				
N1:	N2:			
<pre>[ACIOn Ashager] - Calling the Lo [File_Reader] - reading file [File_Reader] - closing file [Request_Maker] - trying GET from [Request_Maker] - to ET request dom (ACIOn_Manaper] - Load Balancer + [LB_Strategy] - sending node payl (LB_Strategy] - sending node payl (LB_Strategy] - sending node payl (LB_Strategy] - to node with ID 2 [File_Reader] - reading file [File_Reader] - closing file</pre>	by mort [frm_most -> Receiver] - received POST request [Receiver] : sending nessage to Action_dispatcher [Receiver] : received %_lowd_balaching: True id_flow: 2 payload: 0x0 absolution: dBalancer for info [Hone_Executioner] - received %_lowd_balaching: True id_flow: 2 payload: 0x0 [Hone_Executioner] - received %_lowd_b %03abcdefghlt [Hone_Executioner] - received %_lo			
Success criteria				
	ata exchange between nodes have to be executed successfully			
Test observations				
	https://github.com/project-fractal/WP6T62-03-Runtime-Manager			
	$RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3$			
	$N\{1, 2, 3\}$ = Node with id = 1, 2, 3			
Test configuration	N1 is defined by Zyng UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777			
	N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888			
Test conditions	N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999 N1 is overloaded and it cannot perform other computation			
Remarks	Three bugs were foud during validation test that have been reported and corrected by the			
itemarks	developers.			
	1. Error on the configuration file "component.conf"			
	<pre>File "/home/lucs/MDDIS/MI_BMI/MPGT02-03-Runtime-Manager-main/runtime_manager/runtime_manager/core/action_dispatcher.py", line 20, in dispatch exec_type.execute() /le '/home/lucs/MDIS/MI_BMI/MPGT02-03-Runtime-Manager-main/runtime_manager/runtime_manager/core/logic/home_executioner.py", line 19, in execute sair.context.do_logic(sair.id_Tow, sair.payload) is all reference of the same same same same same same same sam</pre>			
	2. Error on the configuration file "comm.conf"			
	Ella # /hans/Hurs/WADTE/MS DNS HUDETE2. A2. Dusting Hansons main frunting annons frunting annons from lighting dispatcher out ling 20. in dispatch			
	<pre>rtsec_type_securine())ii_ni_ni_avoic2.03.nuiternamger.neu/inition_numger/inition_numger/icitystchum.pt/ file/inver/luce/NDSI/ii_INII/iNIC2.03.nuiternamger.neu/inition_numger/inition_numger/cort/logic/home_securitioner.py", line 19, in execute self.context.do.logic(celf.id_flow, self.psyload) file /home/luce/NDSI/ii_INII/iNIC2.03.nuiter.namager.nei/runtine_namager/runtine_namager/corte/logic/context.py", line 23, in do_logic self.context.do.logic(celf.id_flow).self.psyload) file /home/luce/NDSI/ii_INII/iNIC2.03.nuiter.namager.nei/runtine_namager/runtine_namager/corte/logic/home_strategy.py", line 23, in do_logic file /home/luce/NDSI/III_INII/iNIC2.03.nuiter.namager.nei/in/inition_mamager/curtine_namager/corte/logic/home_strategy.py", line 38, in do_algorithm fileptror: 'endpointi' endpointi'</pre>			
	3. Error on the POST Request			
	<pre>[rm_mqtt -> Receiver] - triggered by MqTT msg_s= ('id_Tlow': 'i', 'payload': '0x03abcdefghil') [Receiver] - sending message to Action_dispatcher [Action_dispatcher] - received is_load_balancting: None id_flow: i payload: 0x03abcdefghil [Action_floanger] - coling the Load balancer for info [File_Reseder] - received Balancer for info [Ed_Extractioner] - the ID of the node is 2 [Ed_Extractioner] - the ID of the node is 2 [Ed_Extractioner] - the ID of the node is 2 [Ed_Extractioner] - the ID of the node is 2 [Ed_Extractioner] - the ID of the node is 2 [File_Reseder] - resulting rise [File_Reseder] - rossing from node 2 is [File_Reseder] - ro</pre>			
Test Result				
Passed				

Table 43 - Results of the test T01_WP6T62-03

Copyright © FRACTAL Project Consortium	53 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.4.2 T02_WP6T62-03 - Testing interaction in the local node when the local node can perform the computation

Results/Evidence	
N1:	<pre>[rm_mqtt -> Receiver] - triggered by MQTT nsg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'] [Receiver] - sending message to Action_dispatcher [Action_Manager] - calling the Load Balancer for info [File_Reader] - reading file [File_Reader] - reading file [Request_Maker] - trying GET from http://127.0.0.1:7776/LB/id_node [Request_Maker] - trying GET from http://127.0.0.1:7776/LB/id_node [Home_Executioner] - tabl Balancer returned node ID: None [Home_Executioner] - tabl GET from http://127.0.0.1:7776/LB/id_node [File_Reader] - reading file [File_Reader] - reading file [File_Reader] - reading file [File_Reader] - reading file [Home_Executioner] - tabl Balancer returned node ID: None [Home_Executioner] - tabl Balancer returned node ID: None [Home_Executioner] - tabl Balancer for ID to be executed is 2 [Home_Executioner] - reading file [File_Reader] - reading file [File_Reader] - reading file [File_Reader] - reading file [File_Reader] - trying GET from http://127.0.0.1:6000/endpoint2 [Request_Maker] - trying GET from http://127.0.0.1:6000/endpoint2 [Request_Maker] - trying GET from http://127.0.0.1:6000/endpoint2 [Request_Maker] - trying FT form http://127.0.0.1:6000/endpoint2 [Request_Maker] - Trying FT form form form form form form form form</pre>
Success criteria	
The interaction and o	data exchange in the local nodes have to be executed successfully
Test observations	
	https://github.com/project-fractal/WP6T62-03-Runtime-Manager
	RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3
Test configuration	N{1, 2, 3} = Node with id = 1, 2, 3
Test configuration	N1 is defined by Zynq UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777
	N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888
	N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999
Test conditions	N1 is not overloaded and it can perform any computation
Remarks	
Test Result	
Passed	

Table 44 - Results of the test T02_WP6T62-03

Copyright © FRACTAL Project Consortium	54 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.4.3 T03_WP6T62-03 - Testing interaction between nodes with Node 1 and Node 2 overloaded

N1:		N3:
[Action_Manager] - calling the [File_Reader] - closing file [File_Reader] - closing file [Request_Maker] - trying GET fri [Request_Maker] - definition of the file [Is_Strategy] - performing the [Is_Strategy] - performing the [Is_Strategy] - sending mode_pay [File_Reader] - reading file [Request_Maker] - trying POST [Request_Maker] - response from ni 280 Success criteria	<pre>di: '0x33abcdefghll') Action dispatcher is_load balanctng: None (td_flow: 2 payload: 0x83abcdefghll oad Balancer for info one! </pre>	[Request_maker] - GT request doe! ['TO', 'component?', '1'] ['TO', 'component?', '1']
The interaction and c	lata exchange between nodes have to be ex	acuted successfully
	lata exchange between nodes have to be ex	ecuted successfully
	lata exchange between nodes have to be ex https://github.com/project-fractal/WP6T	·
	~	62-03-Runtime-Manager
Test observations	https://github.com/project-fractal/WP6T	62-03-Runtime-Manager
Test observations	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the no N{1, 2, 3} = Node with id = 1, 2, 3	62-03-Runtime-Manager
Test observations	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the no N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102	62-03-Runtime-Manager ode with id = 1, 2, 3
Test observations	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the nc N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 N2 is defined by a Virtual Machine on Cor	62-03-Runtime-Manager ode with id = 1, 2, 3 board with IP=192.168.0.1 and PORT=7777
Test observations	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the nc N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 N2 is defined by a Virtual Machine on Cor	62-03-Runtime-Manager bde with id = 1, 2, 3 board with IP=192.168.0.1 and PORT=7777 mputer with IP=192.168.0.2 and PORT=8888 mputer with IP=192.168.0.2 and PORT=9999
Test observations Test configuration Test conditions	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the no N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 N2 is defined by a Virtual Machine on Con N3 is defined by a Virtual Machine on Con	62-03-Runtime-Manager bde with id = 1, 2, 3 board with IP=192.168.0.1 and PORT=7777 mputer with IP=192.168.0.2 and PORT=8888 mputer with IP=192.168.0.2 and PORT=9999
The interaction and c Test observations Test configuration Test conditions Remarks Test Result	https://github.com/project-fractal/WP6T RM{1, 2, 3} = Runtime Manager on the no N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 N2 is defined by a Virtual Machine on Con N3 is defined by a Virtual Machine on Con	62-03-Runtime-Manager bde with id = 1, 2, 3 board with IP=192.168.0.1 and PORT=7777 mputer with IP=192.168.0.2 and PORT=8888 mputer with IP=192.168.0.2 and PORT=9999

Table 45 - Results of the test T03_WP6T62-03

Copyright © FRACTAL Project Consortium	55 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.4.4 T04_WP6T62-03 - Task Scheduling on the local node

Results/Evidence	
Result Step 1:	
Flow1: 1 To compone POST paylo TO compone POST resul POST resul	oad [Action_displayment] - received is loss_balancing: Nome is_Tion: 1 payload: %xd3wdcderghil ent2 1 [Action_displayment] - reddisplayment for info [Clayment] - clayment = clayme
Result Step 2:	
Flow2: 2 TO compon GET TO compon	[Action_dispatcher] - received is_load_balancing: None id_flow: 2 payload: 0x03abcdefghil
Result Step 3:	
Flow3: 3 TO compon GET TO compon GET	[Action_dispatcher] - received is load_balancing: None id_flow: 3 payload: 0x03abcdefghil [Action_dispace] - colling the load Palancer for infe
Success criteria	
	task in local node have to be executed successfully
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-03-Runtime-Manager RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3 N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777 N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888 N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999
Test conditions	N1 is not overloaded and it can perform any computation
Test conditions Remarks	N1 is not overloaded and it can perform any computation
	N1 is not overloaded and it can perform any computation

Table 46 - Results of the test T04_WP6T62-03

	Copyright © FRACTAL Project Consortium	56 of 149
--	--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.3.4.5 T05_WP6T62-03 - Task Scheduling on the remote node

Result step 2:		
Flow1: 1 TO component POST payload TO component POST result1 TO component	2 1 3 1	
<pre>NII. (m_nqt >> Receiver] - triggered nsg_ss ('id_flow': 1', 'payload' (Sectiver) - santing nessage to Ac (Action_Manager) - calling the too [File_Beader] - reading file [File_Beader] - closing file [File_Beader] - closing file [Request_Naker] - CdT request done (Action_Manager] - load Balancer fr (id_Drecutione_] - executing action section gathered</pre>	y m017 " Socialsocarfylli] (active states that active states that states and the states and sta	N2: (re_spt - # Rectiver) - resctived PDST request [more_trees] - enciding message to Action_dispatcher [action_dispatched] - rectived (s.log_dislaticity] [more_trees_trees] - enciding the Log_dislaticity of the encided (s.l. [more_trees_trees] - payload to absolated phill [more_trees_trees] - payload to absolated phill [m
transferrendigt her restricted one and	Sfith on a different node di ("payload": "endlabcdefghll", "id_flow": "1", "is_load_balan yload: "endlabcdefghll", "id_flow": "1", "is_load_balancing": t el el 2 is	<pre>[Request_maker] - POT request done ['10]; composity 'i'] [Propust_inker] : rying POST ('payload': ['(\n "endpointi_res": "something"(n)\n")] to hitp [Propust_inker] : root request done ['Request_inker] : POST request done ['Request_inker] : trying POST ('payload': ['(\n "endpointi_res": "something"(n)\n", '(\n "e utit_res": "monthing 'n\n")] to hitp://LIT.8.8.1:0000/endpointi [Request_inker] : POST request done ['Request_inker] : POST request_inker] : POST</pre>
lesult step 3:		
Flow2: 2 T0 component GET T0 component POST result: N1:	t2 1 1 payload	N2:
<pre>(id_flow: 2', 'payload' msg_s= ('id_flow: 2', 'payload' Receiver] - sending message to Ac [Actton dispatcher] - reserved 1s, Actton dispatcher] - reserved 1s, File Beader] - clasing file Telle Beader] - clasing file Request_Bhaker] - trying GET from Request_Bhaker] - GET request done [Actton Janager] - Load Balancer [Ld_Encertuner] - executing actic getting actions and setting actic getting actions and setting actic getting actions actio</pre>	by RUTABLOOFGB(1') : DorBale International Statements International Statements d Balancer for Info http://117.8.8.1:7776/J8/id_mode : : international fiferent mode code 15 2 statement fiferent mode statements s	(rm apt -> Rective?) - rectived POST request Sective? - sective? - received POST request Sective? - sective station dispatcher (Action dispatcher) - received as Load Balancing: True id_flow: 2 payload: 8x83abcdefphil Nome_Securitore? - the ID of the flow to be executed is 2 Nome_Securitore? - public is balabcdefphil Nome_Strategy? - performing the algorithm at home file_Reader? - closing file file_Reader? - closing file
<pre>[LB_Strategy] - Sending node_paylo cing": true} to node with ID 2 [File_Reader] - reading file [File_Reader] - closing file</pre>	ad: ("payload": "dw83abcdefghil", "id_flow": "2", "is_load_bala yload": "dw83abcdefghil", "id_flow": "2", "is_load_balancing": dd2 ef	[File_Reader] - closing file ['10', 'componenti', '2'] '('GEY'] [Request_Naker] - cET request done! Request_Naker] - CET request done!
260		[Request_Maker] - POST request_done! 192.168.0.1 - [14/Dec/2022 15:19:31] "POST /nodo2 HTTP/1.1" 200 -
Result step 4: Flow2: ³ TO componen GET TO componen GET		
N1: [rm_mqtt -> Receiver] - triggered	ыу жотт	N2: [rm api -> Receiver] - received POST request
[Action_Manager] - calling the Loan [File_Reader] - reading file [File_Reader] - closing file	load_balancing: None id_flow: 3 payload: 0x03abcdefghil d Balancer for info	[rn_pip-> Receiver] - received POI request [Receiver] - sending nessage to action dispatcher [Action dispatcher] - received is load balancing: True id_flow: 3 payload: 6x83abcdefgh [Nome_Executioner] - terecuting action here [Nome_Executioner] - her Do it her flow to be executed is 3 [Nome_Executioner] - her Do it her flow to be executed is 3 [Nome_Executioner] - her Do it her flow to be executed is 3 [Nome_Executioner] - her Do it he algorithm at home
Request Maker] - trying GET from I [Request Maker] - GET request done [Action Manager] - Load Balancer ri [LB_Executioner] - texe stallancer (LB_Executioner] - the ID of then n [LB_Strategy] - performing the alg [LB_Strategy] - Sending node_paylo cing': true) to node with ID 2	turned node ID: 2 eturned node ID: 2 no different node de 15 2 rithm on a different node ad: ("payload": "BxBJabcdefghll", "id_flow": "3", "is_load_bala ad: ("payload": "BxBJabcdefghll", "id_flow": "3", "is_load_bala	[File_Reader] - reading file [File_Reader] - closing file [File_Reader] - reading file
[File_Reader] - reading file [File_Reader] - closing file	yload": "0x03abcdefghil", "id_flow": "3", "is_load_balancing": do2	[Request_Maker] - trying GET fron http://127.8.8.1:6000/endpoint2 [Request_Maker] - GET request done! ['To', 'component2', '2'] ['GET']
[Request_Maker] - trying POST ("pa rue} to http://192.168.8.2:8888/noi [Request_Maker] - POST request dom [LB_Strategy] - response from node	2 ts	[Request_Maker] - trying GET from http://127.0.0.1:6000/endpoint2
rue} to http://192.168.8.2:8888/no [Request_Maker] - POST request don [LB_Strategy] - response from node	2 15	[Request_Maker] - GET request done] [Request_Maker] - GET request done] 192.168.0.1 - [14/Dec/2022 15:22:31] "POST /nodo2 HTTP/1.1" 200 -
rue} to http://192.168.0.2:8888/no [Request_Maker] - POST request don [LB_Strategy] - response from node 200	2 ls	[Request_Maker] - GET request done!
rue) to http://192.168.0.218886/no. [Request Maker] - POST request don [LB_Strategy] - response from node 200 Success criteria		[Request_Maker] - GET request done!
rue) to http://192.1868.0.2:8888/no. Request_Maker - POST request_dom [LB_Strategy] - response from node 200 Guccess criteria Fhe execution of the	task and the exchange data with remo	[Request_flaker] - CET request_done! 192.168.8.1 [14/Dec/2822_15:22:31] *POST /nodo2_HTTP/1.1* 280 - Date node have to be executed successfully
rue) to http://192.1868.0.2:3888/no. Request_Make1 - POST request_dom [LB_Strategy] - response from node 200 Success criteria Fhe execution of the	task and the exchange data with remonstrained by the second secon	Request_flaker] - CET request_done! 192.168.8.1 [14/Dec/2822 15:22:31] *POST /nodo2 HTTP/1.1* 280 - Dote node have to be executed successfully WP6T62-03-Runtime-Manager_
rue) to http://192.186.8.28883/nof (Request,Nake') roost request down (L1_Strategy) - response from node 280 Success criteria The execution of the Test observations	task and the exchange data with remo https://github.com/project-fractal/ RM{1, 2, 3} = Runtime Manager on N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ Z	Request_flaker] - GET request_done! 192.168.0.1 [14/Dec/2022 15:22:31] *POST /rodo2. HTTP/1.1* 200 - Dete node have to be executed successfully WP6T62-03-Runtime-Manager_ the node with id = 1, 2, 3 CU102 board with IP=192.168.0.1 and PORT=7777
rue) to http://192.186.8.28883/nof (Request,Nake') roost request down (L1_Strategy) - response from node 280 Success criteria The execution of the Test observations	task and the exchange data with remo https://github.com/project-fractal/ RM{1, 2, 3} = Runtime Manager on N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ Z N2 is defined by a Virtual Machine	Request_flaker] - GET request_done! 192.168.6.1 [14/Dec/2022 15:22:31] *POST /redo2. HTTP/1.1* 200 - Dote node have to be executed successfully WP6T62-03-Runtime-Manager
rue) to http://1902.168.0.23888/nor (Request,Naer / NoST request tool (14_Strategy) - response from node 200 Success criteria The execution of the Test observations Fest configuration	task and the exchange data with remo https://github.com/project-fractal/ RM{1, 2, 3} = Runtime Manager on N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ Z N2 is defined by a Virtual Machine	Request_flaker] - GET request_done! 923.468.6.1 [14/Dec/2022 15/22331] *POST /rodo2.HTTP/1.1* 200 - Dete node have to be executed successfully WP6T62-03-Runtime-Manager_ the node with id = 1, 2, 3 CU102 board with IP=192.168.0.1 and PORT=7777 on Computer with IP=192.168.0.2 and PORT=8888 on Computer with IP=192.168.0.2 and PORT=9999
rue) to http://192.186.0.23888/nor (Request,Nater) - NoST request tool (L2_strategy) - response from node 286 Success criteria The execution of the Test observations Test configuration Fest conditions	task and the exchange data with remo https://github.com/project-fractal/ RM{1, 2, 3} = Runtime Manager on N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ Z N2 is defined by a Virtual Machine N3 is defined by a Virtual Machine	Request_faker] - GET request done! 92.168.0.1 ··· [4/Per/2022 15:22:31] *Post /nodo2 HTTP/1.1* 200 · obte node have to be executed successfully WP6T62-03-Runtime-Manager the node with id = 1, 2, 3 CU102 board with IP=192.168.0.1 and PORT=7777 on Computer with IP=192.168.0.2 and PORT=8888 on Computer with IP=192.168.0.2 and PORT=9999
rue) to http://192.188.0.218880,no [Bewest Maker] - POST request dom [La_strategy] - response from node 200 Success criteria	task and the exchange data with remo https://github.com/project-fractal/ RM{1, 2, 3} = Runtime Manager on N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ Z N2 is defined by a Virtual Machine N3 is defined by a Virtual Machine	Request_laker] - GET request_done! 93.168.6.1 [14/Dec/2022 152231] *POST /nodo2 HTTP/1.1* 200 - obte node have to be executed successfully WP6T62-03-Runtime-Manager. the node with id = 1, 2, 3 CU102 board with IP=192.168.0.1 and PORT=7777 on Computer with IP=192.168.0.2 and PORT=8888 on Computer with IP=192.168.0.2 and PORT=9999

Table 47 - Results of the test T05_WP6T62-03

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.4 Data Ingestion

Data may come into the Fractal Edge Node in various forms, formats, file sizes, and from various sources. For this reason, a Fractal component able to deal with all these heterogeneous types of data is needed. Data ingestion is the process of collecting, importing, and processing raw data from various sources into a data storage or analysis system. This process is a crucial step in data management and is essential to prepare data for analysis and knowledge extraction. Data ingestion involves several steps, including data collection, transformation, and loading. In D6.2 different open-source robust and reliable tools have been identified to support the data ingestion task. Each of them can be used according to the specific needs, the requirements, and the operating system where the component is installed. The tools have been selected to support the different environment and architectures used in the FRACTAL node. In particular, tools have been selected for use in ARM64, RISCV64 and RISCV32 architectures. Since they are well consolidated and open-source tools, a broad documentation can be found online for each of them.

8.4.1 Test planification

8.4.1.1 Define the testing scope and identify the functionality that needs to be tested

In this section the tools proposed in D6.2 are reviewed to check that they can be installed and used according to the guidelines that can be found here https://github.com/project-fractal/WP6T62-01-data-ingestion.

The tests that have been identified consist of installing the components and validating that they are up and running and able to perform basic tasks. More detailed validation can be found in the documentation of the different tools.

T01_WP6T62-01_DI – Testing that Apache NiFi can be installed without any issue.

Validation test		
Test ID	T01_WP6T62-01_DI - Testing Apache NiFi	
Test type	Functional-Installation	
Test name	Installation of Apache NiFi	
Date	13/01/2023	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate that Apache Nifi can be installed without any issue.		

Table 48 - Validation Test T01_WPT62-01_DI

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

T02_WP6T62-01_DI – Testing that PySpark can be installed and configured without any issue.

Validation test		
Test ID	T02_WP6T62-01_DI	
Test type	Functional	
Test name	Installation and configuration of PySpark	
Date	16/02/2023	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to ensure that PySPark can be installed and configured without any issue		

The objective of this test is to ensure that PySPark can be installed and configured without any issue.

Table 49 - Validation Test T02_WPT62-01_DI

T03_WP6T62-01_DI – Testing that Faust can be installed and configured without any issue.

Validation test		
Test ID	T03_WP6T62-01_DI	
Test type	Functional-Installation	
Test name	Installation and configuration of Faust	
Date	16/02/2023	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate the Faust installation guidelines.		

Table 50 - Validation Test T03_WPT62-01_DI

T04_WP6T62-01_DI – Testing that RedNote can be installed and configured without any issue.

Validation test		
Test ID	T04_WP6T62-01_DI	
Test type	Functional-Installation	
Test name	Installation of RedNote	
Date	16/02/2023	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate the RedNote installation guidelines.		

Table 51 - Validation Test T04_WPT62-01_DI

T05_WP6T62-01_DI – Testing that the MQTT is working properly.

Validation test		
Test ID	T05_WP6T62-01_DI	
Test type	Functional-Installation	
Test name	test of MQTT Cloud Communications Component	
Date	03/03/23	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate the MQTT Cloud Communication guidelines.		

Table 52 - Validation Test T05_WPT62-01_DI

Copyright © FRACTAL Project Consortium	59 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.4.2 Test case development

8.4.2.1 T01_WP6T62-01_DI - Testing Apache NiFi

Steps	
Step 1	Prepare a node with Ubuntu and required dependencies (Python)
Step 2	Download files and checking prerequisites
Step 3	Configure and run ApacheNifi
Step 4	Login and Open Apache Nifi in browser

Table 53 - Steps for Validation Test T01_WP6T62-01_DI

8.4.2.2 T02_WP6T62-01_DI - Testing PySpark

Steps	
Step 1	Prepare a node with Ubuntu and the required dependencies (Python 3)
Step 2	Install pySpark and open it on python3

Table 54 - Steps for Validation Test T02_WP6T62-01_DI

8.4.2.3 T03_WP6T62-01_DI – Testing Faust

Steps	
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)
Step 2	install Faust and run python3
Results/Evidence	

Table 55 - Steps for Validation Test T03_WP6T62-01_DI

8.4.2.4 T04_WP6T62-01_DI - Testing RedNote

Steps		
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)	
Step 2	Install Docker	
Step 3	Run the docker	
Step 4	Open the docker in browser	

Table 56 - Steps for Validation Test T04_WP6T62-01_DI

8.4.2.5 T05_WP6T62-01_DI - Testing MQTT Broker

Steps	
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)
Step 2	Set up a container with an MQTT broker
Step 3	Test the publish functionalities
Step 4	Test the subscribe functionalities

Table 57 - Steps for Validation Test T05_WP6T62-01_DI

Copyright © FRACTAL Project Consortium	60 of 149
	00 01 1 17

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.4.3 Test environment setup

The steps to install and configure the component can be found in the FRACTAL project repository <u>https://github.com/project-fractal/WP6T62-01-data-ingestion</u>. According to the guidelines reported there, the tests have been performed on an Ubuntu 22.04 machine.

8.4.4 Test execution

8.4.4.1 T01_WP6T62-01_DI - Testing Apache NiFi

Results/Evidence The Nifi software has been successfully opened a	nd it is ready to use						
	← C ▲ Not secure htt	ps //10.13	8.0.28:8443/nifi/				
	nifi 🖸 🕹	○ट	<u>10</u> 1 <u>10</u> 1	-9990 	2 7	//	
	🔠 0 🔲 0 / 0 bytes	0	80 8	0	▶ 0	0	
	Navigate	Θ					
	Q Q 🕻 :						
		-					
	🖒 Operate	0					
	NIFI Flow Process Group a635a848-0185-1000-b5a7-34deeb201588						
	2 7 % D D D						
		_					
Success criteria							

No error messages/	All partial results	are as expected

Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion
Test conditions	The tests have been performed on a node with Ubuntu 22.04.
Remarks	Running ./nifi.sh start you might be warned that the Java home path is empty. This seems to be a warning. It can be fixed using the following command "export JAVA_HOME="your_java_folder_path" nicola.alchera@rulex.ai@ubuntu=l=vm:=/nifi=1.16.3/bin\$./nifi.sh status nifi.sh: JAVA_HOME not set; results may vary Java home: NiFi home: /home/nicola.alchera/nifi=1.16.3 Bootstrap Config File: /home/nicola.alchera/nifi=1.16.3/conf/bootstrap.conf 2023-01-12 16:34:32,680 INFO [main] org.apache.nifi.bootstrap.Command Apache NiFi is currently running, listening to Boo
	tstrap on port 43525, PID=39872 You cannot download the tarball file from https://nifi.apache.org/download.html because the 1.18 and 1.19 release has the .zip binary files only. To download the .tar file you need to consider the archive https://archive.apache.org/dist/nifi/. The latest version with the .tar file available is the 1.16.3 and this is the one that has been used.
Test result	
Passed	

Table 58 - Results for test T01_WP6T62-01_DI

Copyright © FRACTAL Project Consortium	61 of 149
--	-----------

199 27 2 3 A	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.4.4.2 T02_WP6T62-01_DI - Testing PySpark

Results/Evidence				
pySpark has been cori	rectly installed and it	can be used as a Pytho	on3 package	
>>> pyspark. pyspark.Accumulator(pyspark.Ang(pyspark.Ang(pyspark.BarieriseTaskInfo(pyspark.BarieriseTaskInfo(pyspark.BarieriseTaskInfo(pyspark.BarieriseTaskInfo(pyspark.ClaikBerel pyspark.ClaikBerel pyspark.HiveGontext(pysp	pyspark.Optional(pyspark.Profiler(pyspark.RoOC pyspark.RoOC pyspark.RoOC pyspark.Socketar(pyspark.Socketar(pyspark.SparkConfest(pyspark.SparkConfest(pyspark.SparkConfest(pyspark.StarkStagelrof(pyspark.StarkStagelrof(pyspark.StarkStagelrof()	pyspark.T pyspark.TaskContext(pyspark.Vnon(pyspark.accumLaters pyspark.accumLaters pyspark.claudpickte pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.files pyspark.files	pyspark.inhvritable_thread_target(pyspark.java.gatemay pyspark.join pyspark.eymont_only(pyspark.eymont_only(pyspark.eymonter pyspark.resultiterable pyspark.serializers pyspark.sintfle pyspark.sl	pyspark.statcounter pyspark.status pyspark.storagelevel pyspark.storactorext pyspark.storext pyspark.over pyspark.version pyspark.version pyspark.version
>> pyspark.comulator(pyspark.accumulator(pyspark.accumulator(pyspark.acrierTashContext(pyspark.acrierTashChfd(pyspark.casicProfiler(pyspark.casicProfiler(pyspark.calable(pyspark.calable(pyspark.free;	pyspark.Optional(pyspark.Rodot pyspark.Rodot pyspark.Rodot pyspark.Rodot pyspark.Sodotext(pyspark.Soarkonf(pyspark.SparkConfect(pyspark.SparkConfect(pyspark.SparkConfect(pyspark.SparkConfect(pyspark.StarkConfect(pyspark.StarkConfect(pyspark.StarkConfect(pyspark.StarkConfect(pyspark.StarkConfect(pyspark.StarkConfect()	pyspark.T pyspark.TaskContext(pyspark.Union(pyspark.Union(pyspark.Accumulators pyspark.Insources pyspark.com/ pyspark.com/ pyspark.com/ pyspark.com/ pyspark.com/ pyspark.find pyspark.find	pyspark.inheritable_thread_target(pyspark.jou.gatemay pyspark.spin pyspark.revnorg_only(pyspark.rev1er pyspark.rev1er pyspark.resultierable pyspark.resultierable pyspark.settalizers pyspark.sitfle pyspark.site(pyspark.sql	pyspark.status pyspark.status pyspark.staragelevel pyspark.stakcontext pyspark.rescheak.utils pyspark.vest pyspark.veston pyspark.veston pyspark.weston
Success criteria				
No error messages				
Test observations	-			
Test configuration	https://github.com	/project-fractal/WP6T6	52-01-data-ingestion	
Fest conditions	The tests have been	performed on a node	with Ubuntu 22.04.	
Remarks	No issues have been	n higlighted during the	test	
Test result				
Passed				

Table 59 - Results for test T02_WP6T62-01_DI

8.4.4.3 T03_WP6T62-01_DI - Testing Faust

Results/Evidence	
Faust has been corre	ctly installed and it can be used as python3 library
Success criteria	
No error messages/A	Il partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion
Test conditions	The tests have been performed on a node with Ubuntu 22.04.
Remarks	No issues have been highlighted during the test
Test result	
Passed	

Table 60 - Results for test T03_WP6T62-01_DI

Copyright © FRACTAL Project Consortium	62 of 149

1992 P.C.	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
X-X- <i>X</i>	Del. Code	D6.4

8.4.4.4 T04_WP6T62-01_DI - Testing RedNote

	cure 10.138.6.281185(/#fox/162187d/17ca8	A1	Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø Ø	
Node-RED	You can open the tips		= Deploy •	Q.
rodes .	Piew 1	* *	i info i a a o =	•
inject	MARHING: please check you have started this container with a volume that is mounted to Jata's otherwise any flow changes are test when you reliably or upgrade the container (b g upgrade to a more recent node-red docien image), If you are using named solutions you can spone this warning		Poes Poes Poes Poes Sobflows	0 4
complete ()	Double click or see info side panel to learn how to start Node-RED in Docker to save your work		> Global Configuration Nodes	+
campun i califi status bik in () link cali () comment comment				
function ()				
selich 0			E Flow1	
change) Semplate) Gelay) Migger () Koac			Pox "Stelloritzal"	60
s crite	ria	10 - 0 +		0

Test observations		
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion	
Test conditions	The tests have been performed on a node with Ubuntu 22.04.	
	Check you have started this container with a volume mounted on /data. I checked, and it seems the container is monted on /data, as you can see in the image below.	
Remarks	<pre>Despite this, the workflow is not saved . "Mounts": [{ "Type": "volume", "Name": "node_red_data", "Source": "/var/Lib/docker/volumes/node_red_data/_data", "Destination": "/data", "Driver": "local", "Mode": "z", "RW": true, "Propagation": "" }],</pre>	
Test result		
Passed		

Table 61 - Results for test T04_WP6T62-01_DI

Copyright © FRACTAL Project Consortium	63 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.4.4.5 T05_WP6T62-01_DI - Testing MQTT Broker

tesults/Evidence	Nicolaratcheralmiter_simumtul-ums-/mP0T62-01-data-ingestion/mosquitto/test-com5 ovthon3 test-subscr
	ibe.py
Both the publisher and subscriber script works correctly.	Connected to MQTT Broker! Received `messages: 35` from `fractal/mgtt-test` topic
both the publisher and subscriber script works correctly.	Necesved messages: 35' from 'fracta/mgtt-test' topic Received 'messages: 36' from 'fracta/mgtt-test' topic
	Received 'messages: 37' from 'fractal/mgtt-test' topic
	Received 'messages: 38' from 'fractal/mqtt-test' topic
🗾 nicola.akhera@rulex.ai@ubus X 🔊 nicola.akhera@rulex.ai@ubun X 🕂 🗸	Received 'messages: 39' from 'fractal/mqtt-test' topic Received 'messages: 40' from 'fractal/mqtt-test' topic
b3d48adedc5b8edcb43437aecb611827dca	Received 'messages: 41' from 'fractal/mgt-test' topic
Stored in directory: /home/nicola.alchera/.cache/pip/wheels/6a/48/01/c895	Received 'messages: #2' from 'fractal/mqtt-test' topic
stored in directory. /nome/nicota.atchera/.tache/pip/wheets/da/40/01/cost	Received 'messages: 43' from 'fractal/mqtt-test' topic
r Successfully built paho-mott	Received 'messages: 44' from 'fractal/mqtt-test' topic Received 'messages: 45' from 'fractal/mqtt-test' topic
Installing collected packages: paho-mgtt	Received 'messages: 46' from 'fractal/mgtt-test' topic
	Received 'messages: H7' from 'fractal/mqtt-test' topic
Successfully installed paho-mqtt-1.6.1	Received 'messages: 48' from 'fractal/aptt-test' topic
<pre>nicola.alchera@rulex.ai@ubuntu-l-vm:~/WP6T62-01-data-ingestion/mosquitto/te</pre>	Received 'messages: 49' from 'fractal/mqtt-test' topic Received 'messages: 50' from 'fractal/mqtt-test' topic
nicola.alchera@rulex.ai@ubuntu-l-vm:~/WP6T62-01-data-ingestion/mosquitto/te	Received 'messages: 51' from 'fractal/mqtt-test' topic
Connected to MQTT Broker!	Received 'messages: 52' from 'fractal/mqtt-test' topic
Send `messages: 0` to topic `fractal/mqtt-test`	Received 'messages: 53' from 'fractal/mqtt-test' topic Received 'messages: 50' from 'fractal/mqtt-test' topic
Send `messages: 1` to topic `fractal/mqtt-test`	necesved messages; buf from 'fractal/mqtt-test' topic Received 'messages; 55' from 'fractal/mqtt-test' topic
Send `messages: 2` to topic `fractal/mqtt-test`	Received 'messages: 56' from 'fractal/agt-test' topic
Send `messages: 3` to topic `fractal/mqtt-test`	Received 'messages: 57' from 'fractal/mqtt-test' topic
Send `messages: 4` to topic `fractal/mqtt-test`	Received 'messages: 58' from 'fractal/mgtt-test' topic Received 'messages: 59' from 'fractal/mgtt-test' topic
Send `messages: 5` to topic `fractal/mqtt-test`	necesveo messages: ov 'rom 'rracta/mutt-test' topic Received 'messages: do 'from 'fracta/mutt-test' topic
Send `messages: 6` to topic `fractal/mqtt-test`	Received 'messages: 61' from 'fractal/mqtt-test' topic
Send 'messages: 7' to topic 'fractal/mqtt-test'	Received 'messages: 62' from 'fractal/mqtt-test' topic
Send `messages: 8` to topic `fractal/mqtt-test`	Received 'messages: 63' from 'fractal/mqtt-test' topic Received 'messages: 64' from 'fractal/mqtt-test' topic
Send 'messages: 9' to topic 'fractal/mqtt-test'	Received messages: 65 'from 'fractal/mgtt-test' topic
Send `messages: 10` to topic `fractal/mqtt-test`	Received 'messages: 66' from 'fractal/mqtt-test' topic
Send `messages: 11` to topic `fractal/mqtt-test`	Received 'messages: 67' from 'fractal/ngt-test' topic
Send `messages: 12` to topic `fractal/mgtt-test`	Received 'messages: 68' from 'fractal/mqtt-test' topic Received 'messages: 69' from 'fractal/mqtt-test' topic
Send `messages: 13` to topic `fractal/mgtt-test`	Acceived 'messages: 70' from 'fractal/mutt-test' topic
Send `messages: 14` to topic `fractal/mgtt-test`	Received 'messages: 71' from 'fractal/mqtt-test' topic
Send `messages: 15` to topic `fractal/mgtt-test`	Received 'messages: 72' from 'fractal/mqtt-test' topic Received 'messages: 73' from 'fractal/mgtt-test' topic
Send 'messages: 16' to topic 'fractal/mgtt-test'	Heceived messages: 73' from 'fracta/mott-test' topic Received messages: 74' from 'fracta/mott-test' topic
Send `messages: 17` to topic `fractal/mgtt-test`	Received 'messages: 75' from 'fractal/mqtt-test' topic
Send `messages: 18` to topic `fractal/mgtt-test`	Received 'messages: 76' from 'fractal/mqtt-test' topic
Send `messages: 19` to topic `fractal/mgtt-test`	Received 'messages: 77' from 'fractal/mott-test' topic Received 'messages: 78' from 'fractal/mott-test' topic
Send `messages: 20` to topic `fractal/mgtt-test`	Received 'messages: 79' from 'fractal/mqtt-test' topic
	Received 'messages: 80' from 'fractal/mqtt-test' topic
uccess criteria	
o error messages/All partial results are as expected	

Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion
Test conditions	The tests have been performed on a node with Ubuntu 22.04.
	The clone command "git clone https://github.com/project-fractal/WP6T62-01-data-ingestion.git"
	could not work properly. The user password authentication is no longer supported: a valid token is
	required for access. If you haven't any token, you have create a new one following the instructions
	you can fine here https://docs.github.com/en/get-started/getting-started-with-git/about-remote-
	repositories#cloning-with-https-urls
Remarks	
	The command "source run.sh" could give you an error. To avoid the error you shold give the
	permission through the following command "sudo chmod 777 /var/run/docker.sock"
	There is an error in guideline: guidlines says to test two times the same python script (test-publish.py)
	and not the "test-publish.py" and the "test-subscribe.py"
Test result	
Passed	

Table 62 - Results for test T05_WP6T62-01_DI

Copyright © FRACTAL Project Consortium	64 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.5 Federated Data Collection

Federated data collection is a process that enables organizations to collect and analyse data from multiple sources in a decentralized manner. Unlike traditional data collection methods, which involve bringing all data into a single central location for analysis, federated data collection allows organizations to analyze data in place, without moving it to a central location. In D6.2 different solutions, based on open-source tools, are proposed to handle the federated collection of the data coming from the Edge Nodes.

8.5.1 Test planification

In this section the tools proposed in D6.2 are reviewed to check that they can be installed and used according to the guidelines that can be found here https://github.com/project-fractal/WP6T62-02-federated_data_collection.

The tests that have been identified consist of installing the components and validating that they are up and running and able to perform basic tasks. More detailed validation can be found in the documentation of the different tools.

T01_WP6T62-02_FDC – Testing that CrateDB can be installed and used without any issue.

Validation test		
Test ID	T01_WP6T62-02_FDC	
Test type	Functional-Installation	
Test name	Installation of CrateDB	
Date	16/01/2023	
Tester's Name	Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate the CrateDB installation guidelines.		

Table 63 - Validation Test T01_WPT62-02_FDC

T01_WP6T62-02_FDC – Testing that MongoDB can be installed and used without any issue.

Validation test		
Test ID	T02_WP6T62-02_FDC	
Test type	Functional-Installation	
Test name	Installation of MongoDB	
Date	24/01/2023	
Tester's Name	s Name Nicola Alchera (Rulex)	
Test scope or objective		
The objective of this test is to validate the MongoDB installation guidelines.		

Table 64 - Validation Test T02_WPT62-02_FDC

Copyright © FRACTAL Project Consortium	65 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.5.2 Test case development

8.5.2.1 T01_WP6T62-02_FDC - Testing CrateDB

Steps	
Step 1	Prepare a node with Ubuntu and required dependencies
Step 2	Install CrateDB on Ubuntu
Step 3	Run the docker and open CrateDB in a web browser
Step 4	Create and view a table using SQL into Crate DB

Table 65 - Steps for Validation Test T01_WP6T62-02_FDC

8.5.2.2 T02_WP6T62-02_FDC - Testing MongoDB

Steps	
Step 1	Prepare a node with Ubuntu and required dependencies
Step 2	Install Mongo on Ubuntu
Step 3	Run the docker

Table 66 - Steps for Validation Test T02_WP6T62-02_FDC

8.5.3 Test environment setup

The steps to install and configure the component can be found in the FRACTAL project repository <u>https://github.com/project-fractal/WP6T62-02-</u> <u>Federated Data Collection</u>. According to the guidelines reported there, the tests have been performed on a Ubuntu 22.04 machine following all the instruction reported in the guidelines repository.

Copyright © FRACTAL Project Consortium	66 of 149

199 199 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.5.4 Test execution

Test result Passed

8.5.4.1 T01_WP6T62-02_FDC - Testing CrateDB

Results/Evidence					
CrateDB has been suc	cessfully opened in browser and it is	ready to	o use		
			teDB		Cluster: crate Versior: 5.1.3 Nodes: 1 Health: 0
← C ▲ Not secure 10.138.0.28420	A4.001		Console		
CrateDB	AU#0	5	Enter a SQL statement to query CrateDB		C Store console history persistently
Cluster: crate			CREATE TABLE tab2(col1 INTEGER, col2 TEXT		
Health		000			
good	100.0% 100.0%	*	SUBMIT QUERY		CREATE TABLE OK, 1 record affected (0.385 seconds)
Cluster load		6			
88 az					
*		🗮 Crate	еОВ		Cluster: crate Version: 51.3 Nodes: 1 Healt
8			Console		
0			Enter a SQL statement to query CrateDB	Format results	Store console history persistently
0.10			SELECT * FROM tab LINIT 100;	Pomat results	Store console nationy persistentry
0.05					
0.00		<u>\$8</u>			
Queries Per Second	🔵 Overali 🔮 Select 🕒 Insert 🌑 Update 🐞 Delete	*	SUBMIT CUERY		SELECT OK, 2 records returned (0.455 seconds)
24					
		0			
Success criteria					
Data can be created a	nd viewed.				
Success criteria					
Data can be created a	nd viewed.				
Test observations					
Test configuration	https://github.com/project-fracta	<u>I/WP6T6</u>	52-02-federated_data_collection		
Test conditions	The tests have been performed or	i a node	with Ubuntu 22.04.		
Remarks	No issues were highlighted during	the test	t		

Table 67 - Results for Validation Test T01_WP6T62-02_FDC

Copyright © FRACTAL Project Consortium	67 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
$\mathbf{x} = \mathbf{X}_{\mathbf{x}} = \mathbf{x}_{\mathbf{x}}$	Del. Code	D6.4

8.5.4.2 T02_WP6T62-02_FDC - Testing MongoDB

Results/Evidence	
MongoDB has been s	uccessfully installed
 mongod.servic Loaded: lo Active: ac Docs: ht Main PID: 56 Memory: 16 CGroup: /s 	<pre>rulex.ai@ubuntu-l-vm:~\$ sudo systemctl status mongod e - MongoDB Database Server aded (/Lib/systemd/system/mongod.service; disabled; vendor preset: enabled) tive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago tps://docs.mongodb.org/manual 3824 (mongod) 8.3M ystem.slice/mongod.service 563824 /usr/bin/mongodconfig /etc/mongod.conf ubuntu-l-vm systemd[1]: Started MongoDB Database Server.</pre>
Success criteria	
Data can be created a	and viewed via MongoDB
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-02-federated_data_collection
Test configuration Test conditions	https://github.com/project-fractal/WP6T62-02-federated_data_collection The tests have been performed on a node with Ubuntu 22.04.
Test conditions	The tests have been performed on a node with Ubuntu 22.04.

Table 68 - Results for Validation Test T02_WP6T62-02_FDC

Copyright © FRACTAL Project Consortium	68 of 149
	00 01 147

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.6 Low End Node

This component is based on the open-source RISC-V based PULP platform. The main features of the Low End Node are:

- Bridges between the IoT Hub and Kubernetes Cluster
- Shows the Nodes in CRD format in the Kubernetes Cluster
- Support any other MQTT broker-based connection

8.6.1 Test planification

8.6.1.1 Define the testing scope and identify the functionality that needs to be tested

The functionalities identified for the Low End Node component that need to be tested are related to how to connect and communicate with the cloud platform and the task scheduling.They were defined as follows:

- 1. The Low End Node has to be able to connect and communicate with the Cloud Platform;
- 2. The Low End Node has to be able to execute task scheduling;
- 3. The Low End Node has to be able to manage Ingestion and Storage.

For the first functioning three test cases are defined as shown below.

a) T01_WP6T62-06 – Testing the connection between the Device and the Cloud Platform.

Validation Test			
Test ID	T01_WP6T62-06		
Test type	Functional		
Test name	Testing the connection between the Device and the Cloud Platform		
Date	19/01/2022		
Tester's Name	Luca Visconti (Akkodis)		
Test scope or objective			
The objective of this test is to validate the connection between Low End Node Device and Cloud Platform.			

Table 69 - Validation Test T01_WP6T62-06

b) T02_WP6T62-06 – Testing the communication from the Device to the Cloud.

Validation Test		
Test ID	T02_WP6T62-06	
Test type	Functional	
Test name	Testing the communication from Device to the Cloud	
Date	19/01/2022	
Tester's Name	Luca Visconti (Akkodis)	
Test scope or object	tive	
The objective of this test is to validate communication between Low End Node Device and Cloud Platform.		

Table 70 - Validation Test T02_WP6T62-06

Copyright © FRACTAL Project Consortium	69 of 149
	07 01 147

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

c) T03 WP6T62-06 – Testing the communication from the Cloud Platform to the Device.

Validation Test			
Test ID	T03_WP6T62-06		
Test type	Functional		
Test name	Testing the communication from the Cloud Platform to the Device		
Date	19/01/2022		
Tester's Name	Luca Visconti (Akkodis)		
Test scope or objective			
The objective of this test is to validate the communication between Cloud Platform and Low End Node Device.			

Table 71 - Validation Test T03 WP6T62-06

For the second functioning, one test case is defined and shown below.

d) T04_WP6T62-06 – Testing the task scheduling running Nuttx on the Device.

Validation Test			
Test ID	T04_WP6T62-06		
Test type	Functional		
Test name	Testing the Tasks Scheduling running Nuttx on the Device		
Date	19/01/2022		
Tester's Name	Tester's Name Luca Visconti (Akkodis)		
Test scope or objective			
The objective of this test is to validate the task scheduling on the Low End Node			

Table 72 - Validation Test T04_WP6T62-06

Nuttx supports normal posix socket and posix file systems, where data flow control can be handled with adequate drivers. In this way, various communication and storage mediums may be added. However, driver development is out of scope of validation activities.

8.6.2 Test case development

8.6.2.1 T01_WP6T62-06 - Testing the connection between the Device and the Cloud Platform

Step 1: Power up the device.

In this step, the device is connected to a power source with 5V to start it correctly.

Expected Results: having booted correctly the device is possible to notice that the red LED switch on as expected.

Step 2: Connect the device to the Internet.

In this step, the device is connected to an Access Point using SSID and password parameters. The figures below show that the device is connected to the network.

Copyright © FRACTAL Project Consortium	70 of 149

- States	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4
	ubuntu@maste NAME	<pre>r:~\$ kubectl get lowends.fractal-cluster.eu -n low-end-ctrl AGE</pre>

ubuncu@master:~	s Rubecci gec	towends.fractal-cluster.eu -n	tow-end-ctrt
NAME	AGE		
low-end-2v2wf	3m24s		
low-end-5r2gf	3m18s		
low-end-llq4k	3m17s		
low-end-lzjrf	3m21s		
low-end-mm6pw	3m21s		
low-end-w9p27	3m14s		
low-end-z8srs	3m15s		
ubuntu@master:~	\$		

Figure 27 - The list of the provisioned devices as CRD in Kubernetes cluster with their ids

In the figure below is shown the list of IoT Devices on the Cloud side that are Connected or Disconnected. The connected device id is "fractal-node-8063XXX".

e offcode				Fra	actal Refre
IoT Hub Devices:					^
Disconnected fractal-node- 17B81F32C7C7CC9F7005FB05884F4D06	Desired	Reported	Measure 🗑 Ret	move U	⁹ a
Disconnected fractal-node- 904ADF13C9AD3DE1FC48EB3F731BD00D	Desired	Reported	Measure 🛱 Ret	move	RA
Disconnected fractal-node- A62253BE425CE8893CA6839375D6BEF1	Desired	Reported	Measure 🕯 Ren	move D	BA
Connected fractal-node- 806317DC2DFED29E09D5DF353287AFE8	Desired	Reported	Measure 😫 Ret	move 💍	6 ³
Disconnected	Desired	Reported			•

Figure 28 - The list of devices in IoT hub

The following figure shows through "kubectl" command the description of the Low End node in the Kubernetes cluster. The K8s generated id is low-end-5r2gf and the device id is "fractal-node-8063XX".

Copyright © FRACTAL Project Consortium	71 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

<pre>dustributer:= % hubert1 describ townds.fractal-cluster.eu =n low-end-ctrl low-end-Str2gf Nume:= low-end-Str2f Low-end-Str2f Aunoitation: net "spect:: ["connectionState":"Connected", "desiredState": ["lastUpdated": "2023-02-20107:33:00.05241742", "state":0], "device! devices: net "spect:: ["connectionState":"Connected", "desiredState": ["lastUpdated": "2023-02-20107:33:00.05241742", "state":0], "device! devices: net "spect:: ["connectionState":"Connected", "desiredState": ["lastUpdated": "2023-02-20107:33:00.05241742", "state":0], "device! devices: net "spect:: ["connectionState": "ConnectionState": ["lastUpdated": "2023-02-20107:33:00.05241742", "state":0], "device! devices: ["connectionState: ["connectionState": ["lastUpdated": "2023-02-20107:33:00.05241742", "state":0], "device! failed:: ["connectionState: ["connectionState": ["connectionState": ["connectionState": ["connectionState": ["connectionState": ["connectionState": ["connectionState": ["connectionState: ["connectionState: ["connectionState": ["connectionState: ["connectionStat</pre>		
<pre>Namespace: low-mod-trl Label:onnex Annotations: kopf.zalando.org/Last-handled-configuration:</pre>		
Labdiz: <pre>cnone> Annotations: {"Spec::\frommetionState":"Connected"."desizedState":"2023-02-20107:33:00.95201702","state":0),"deviceI d":"fract1-cluster.eu/v1 Kind: Unefind Mr Annotations: Unefind Mr Annotations: // Unefind Mr Annotat</pre>		
Annotations: kopf.alando.org/last-handled-configuration: ("spec:"("connected","desiredState":"("lastUpdated":"2023-02-20107:33:00.952417A2","state":0),"deviceI d":"fratal-node=80 Factal-cluster.eu/v1 Kind: Formation: Formatio		
<pre>[*spect:/ronmectionState::Connected","desiredState":["lastUpdated":"2023-02-20107:33:00.95241742","state":0),"device1 d?: *factal.oueMed Metadata: Creation :mestamp: 2023-02-20107:39:242 Final : LowEnd Metadata: Creation :mestamp: 2023-02-20107:39:242 Final :mestamp: 2023-02-20107:39:34 Final :mestamp: 2023-02-20107:39:34 Final :mestamp: 2023-02-20107:39:312 Final :mestamp: 2023-02-20107:39:00.95241742 Final :mestamp: 2023-02-20107:39:00.95241742 Final :mestamp: 2023-02-20107:39:00.95241742 Final :mestamp: 2023-02-20107:39:00.95241742 Final :mestamp: 2023-02-20107:39:00.95241742 Final :mestamp: Final :m</pre>		
<pre>d': Fractal-mode-86 AFU Version: Fractal-cluster.eu/v1 Kind: LowEnd Metadata: Creation limestamp: 203-02-20T07:39:39/24Z Franitzers: Notification: 2 Managed Fields: AFU Version: 2 Managed Fields: AFU Version: 2 Managed Fields: fieldsType: FieldsV1 FieldsType: FieldsV1 Field</pre>	Annotations:	
ADT Version: fractal-clutter.eu/v1 Ktadata: Creation Timestamp: 2023-02-20T07:39:24Z Finalizers: kopf.zalando.org/KopfFinalizerMarker Generation: 2 Mampdoffsion: 2 Mampd		
<pre>kind: LowEnd Hetadata: Creation Timestamp: 2023-02-20107:39:34Z Finalizers: Kopf:zalando.org/KopfFinalizerMarker Generate Name: low-end- Generation: # filestadata: filestadatadata: filestadatadat</pre>		
<pre>Metadata: Creation Timestamp: 2023-02-20107:39:2/UZ Finalizers: kopf.salands.org/kopfFinalizerMarker Generation: 2 Managed Fistds: API Version: fractal-cluster.eu/v1 FistdsType: FistdsU1 FistdsType: FistdsU2 FistdsType: FistdsType: FistdsU2 FistdsType: FistdsType: FistdsU2 FistdsType: FistdsType: FistdsTyp</pre>		
<pre>Greation Timestamp: 2023-02-20107:39:202 Finalizes: KopF.zalando.org/KopFfanlizes/Marker Generate Name: low-end- Generation: 2 Anagef Fickles: Anagef Fic</pre>		LowEnd
Finalizers: kopf.zalando.org/kopffinalizerMarker Generation: 2 Managed Fields:		
<pre>kopf.zlando.org/kopfinalizerMarker Generation: 2 Managed Fields: API Version: fractal-cluster.eu/vl Fields Type: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl fieldsType: FieldsVl V: Nopf.zalando.org/kopfFinalizerMarker*: Nanager: kopf Operation: Update Time: 2023-02-20107:39:312 API Version: fractal-cluster.eu/vl FieldsType: FieldsVl fieldsVl: field</pre>		
Generation: 2 Managed Fields: API Version: fractal-cluster.eu/vl FieldsType: FieldsVl fieldsVl: fieldsVl: fieldsVl: fieldsClui: finalizers: Manager: kopf Operation: Update Operation: Update Operation: Update fieldsVl:		
Generation: 2 Managed Fields: Managed Fields: Fields Type: FieldsVI FieldsType: FieldsVI fieldsUV: field		
<pre>Managed Fields: Ap: Version: fractal-cluster.eu/vl FieldsV1: fieldsV1: fieldsV1: fieldsV1: fishadtata: f.motations: i.iopf.zalando.org/last-handled-configuration: f.finlizers: v.'Wcopf.zalando.org/KopffinalizerMarker": Managed: kopf Operation: Update 2023-02-20107.39:312 API Version: fractal-cluster.eu/vl fieldsV1: f.idsV1 Version: fractal-cluster.eu/vl fieldsV1: f.idsV2: f.idsV2:</pre>		
Adž Version: fractal-cluster.eu/v1 Fields Type: FieldsV1 fieldsV1: f:montations: if:finalizers: iv:"topf.zalando.org/KopfFinalizerMarker": Manager: p0:f.ralando.org/KopfFinalizerMarker": Manager: p0:f.ralando.org/KopfFinalizerMarker": Manager: p0:f.ralando.org/KopfFinalizerMarker": Manager: p0:f.ractal-cluster:.eu/v1 Fields Type: FieldsV1 fields Type: FieldsV1 fields Type: FieldsV1 fields Type: FieldsV1 fieldsU2: i: f:connectionState: f:dsiredState: i: f:lastUpdated: f:tastUpdated: f:		
<pre>FieldsType: FieldsV1 fieldsV1: fieldsV1: fieldsV1: fieldsV2: fieldsV2:</pre>		
<pre>fieldsVL: f:menotations: fiannotations: fianlizers:</pre>		
<pre>f=metadata: f:amontations: i: i: i: f:finalizers: v:"kopf.zalando.org/Last-handled-configuration: f:finalizerMarker": Manager: kvpof.zalando.org/KopfFinalizerMarker": Manager: v:"kopf.zalando.org/KopfFinalizerMarker": Manager: dQ2=02-20107:39:31Z API Version: fpatato- tilestUpdated: f:metadata: f:metadata: f:spec: i: f:connectionState: f:connectionState: f:lastUpdated: f:state: f:lastUpdated: f:state: Manager: pykube-ng Operation: UDD: convectionState: fister Tame: 2023-02-20107:39:31Z Resource Version: S55543366 UDD: convectionState: connection State: connection State: connection State: f: fister f</pre>		
<pre>f:annotations: f:kopf.zalando.org/last-handled-configuration: f:finalizers:</pre>		
<pre>:::untrol f:finalizers: v:"kopf.zalando.org/kopfFinalizerMarker": Manager: kopf Operation: Update: f::state: f::state:</pre>		
<pre> f:kopf.zalando.org/last-handled-configuration: f:finalizers: if "kopf.zalando.org/kopfFinalizerMarker": Manager: kopf Operation: Update Z023-02-20707.39:31Z API Version: fractal-cluster.eu/vl Fields Type: Factal-cluster.eu/vl Fistate: f.generateName: f.state: f.state: f.desiredState: i: f.iastUpdated: f.istupec: i: f.iastUpdated: f.state: f.iastUpdated: f.state: f.iastUpdated: f.istate: f.iastUpdated: f.state: f.iast</pre>		otations:
<pre>f:finalizers:</pre>		
<pre>.:: "Nopf.zalando.org/KopfFinalizerMarker": Manager: k kopf Operation: Update Time: 2023-02-20T07:39:312 API Version: fractal-cluster.eu/v1 FieldsType: FieldsV1 fieldsV1: f:metadata: f:spec: f:connectionState: f:desiredState: f:tastUpdated: f:tastU</pre>		
<pre>v:"topf.zlando.org/k0pfFinalizerMarker": Manager: kopf Operation: Update Time: 2023-027070.39:31Z API Version: fractal-cluster.eu/v1 FieldsV1: fieldsV1</pre>		alizers:
<pre>Manager: kopf Operation: Update Time: 2023-02-20107:39:31Z API Version: fractal-cluster.eu/v1 FieldsV1: FieldsV1: fieldsV1: fieldsV1: fieldsV1: fispec: f:spect f:spect f:spect f:desiredState: f:desiredState: f:deviceId: fister fister</pre>		
Operation:UpdateTime:2023-02-20107:39:312API Version:fractal-cluster.eu/vlFields Type:Fields Vl:fields type:<		
<pre>Time: 2023-02-20107:39:312 API Version: fractal-cluster.eu/v1 Fields Type: Fieldsv1 fieldsv1: f:entadata: f:generateName: f:spec:</pre>		
API Version: fractal-cluter.eu/vl Fields Type: FieldsVl fieldsVl: fimetadata: figenerateName: fispec: i ficonnectionState: ficonnectionState: i filastUpdated: fistate: filastUpdated: fistate: filastUpdated: fistate: filastUpdated: fistate: filastUpdated: fistate: filastUpdated: fistate: cluteNamager: VMUMDe-ng Operation: Update Time: 2023-02-20107:39:31Z Resource Version: 5558504366 UD: connection State: Connection State: Connected Desired State: Last Updated: 2023-02-20107:33:00.95241742 State: 0 Device 1: fistate: 0 Device 2: 0 Device 3: 0 Device 4: 0 Device 3: 0 Device 4: 0 Device 4: Device 4: Device 4: Device 4: Device 4: Device 4: Device 4: Device 4: Device 4: Device		
<pre>Fields Type: FieldsV1 fieldsV1: fimetadata: f:generateName: f:spmerateName: f:spmerateName: f:spmerateName: f:spmerateName: f:spmerateName: f:spmerateName: f:state: f:st</pre>		
<pre>filedsVL¹ f:metadata: f:generateName: f:generateName: f:generateName: f:generateName: f:generateName: f:generateName: f:generateName: f:desUpdated: f:connectionState: if tastUpdated: f:state: f:desUpdated: f:tastUpdated: f:t</pre>		
<pre>f:metadata: f:generateName: f:spec: i f:connectionState: f:desiredState: i f:lastUpdated: f:state: f:device3() f:reportedState: i f:lastUpdated: f:state: f:lastUpdated: f:state: f:tastUpdated: f:state: f:state: f:state: f:state: f:astUpdated: f:state: f:state: f:state: f:astUpdated: f:state: f:state: f:state: f:state: f:state: f:astUpdated: f:state: f:stat</pre>		
<pre>f:generateName: f:spec: f:spec: f:desiredState: f:desiredState: f:lastUpdated: f:state: f:tastUpdated: f:reportedState: i f:lastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: UD: dog23-02-20107:39:31Z Resource Version: S5585U366 UID: connection State: Connection State: Connected Desired State: Last Updated: 2023-02-201707:39:29.68705272 Connection State: Connection Stat</pre>		
<pre>f:spec:</pre>		
<pre>f:connectionState: f:connectionState: f:desiredState: i f:dastUpdated: f:state: f:dastUpdated: f:reportedState: i f:lastUpdated: f:lastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:tastUpdated: f:lastU</pre>		
<pre>f:connectionState: f:desiredState: i: f:lastUpdated: f:state: f:deviceId: f:reportedState: f:teviceId: f:reportedState: f:teviceId: f</pre>		
f:desiredState: f:lastUpdated: f:state: f:deviceId: f:reportedState: f:astUpdated: f:state: f:tastUpdated: f:state: f:tastUpdated: 0peration: Update Time: 2023-02-20107:39:31Z Resource Version: 055854366 UID: connection State: Connection State: Connected Desired State: Last Updated: 2023-02-20107:33:60.95241742 State: 0 Device Id: f:rectal-node-806317DC2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20107:39:29.6870527Z		
.: f:lastUpdated: f:state: f:stat		
f:lastUpdated: f:state: f:deviceId: f:reportedState: i f:lastUpdated: f:state: Manager: pykube-ng Operation: Update Time: 2023-02-20107:39:31Z Resource Version: 555854366 UID: dc9e051f-8362-4065-8f4b-40013a4c68066 UID: dc9e051f-8362-4065-8f4b-40013a4c68066 Spec: Connection State: Connected Desired State: Last Updated: 2023-02-20107:33:60.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20107:39:29.6870527Z		iredstate:
f:state: f:reportedState: f:state: f:state: manager: pykube-ng Operation: Update Tame: 2013-02-20T07:39:31.Z Resource Version: 558694366 UID: cd>0e951f-8362-4065-8f4b-4d013a4c00d6 Spec: Connection State: Connection State: Connected Desired State: 0 Last Updated: 023-02-20T07:33:00.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED39E09DSDF353287AFE8 Reported State: Last Updated: Last Updated: 023-02-20T07:39:29.6870527Z		
f:deviceId: f:reportedState: f:lastUpdated: f:lastUpdated: f:state: Manager: pykube-ng Operation: Update Time: 2023-02-20T07:39:31Z Resource Version: 555854366 UID: dc9e051f-8362-4065-8f4b-4d013a4c80d6 Spec: Connection State: Connected Desired State: 0 Device Id: fractal-node-806317Dc2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20T07:39:29.6870527Z		
f:reportedState: f:lastUpdated: f:state: Manager: pykube-ng Operation: Update Time: 2023-02-201797:39:31Z Resource Version: 555854366 UID: d02951f-8362-4065-8f4b4d013a4c0046 Spec: Connection State: Connected Desired State: Last Updated: 2023-02-201797:33:00.95241744Z State: 0 Device Id: fractal-node-806317DC2DFED39E09DSDF353287AFE8 Reported State: Last Updated: 2023-02-201797:39:29.68705277Z		
 f:lastUpdated: f:state: Manager: pykube-ng Operation: Update Time: 2023-02-20T07:39:31Z Resource Version: 555854366 UID: dc9e051f-8362-4065-8f4b-44013a4c08d66 Spec: Connection State: Connected Desired State: Last Updated: 2023-02-20T07:33:00.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED29E09D5DF353287AFE8 Reported State: Last Updated: 2023-02-20T07:39:29.6870527Z		
f:lastUpdated: f:state: Manager: pykube-ng Operation: Update Time: 2023-02-20107:39:31Z Resource Version: 555854366 UID: cd>vdv851f-8362-4065-8F4b-40013a4c08d6 Spec: connection State: Connection State: Connected Desired State: 0 Device Id: fractal-node-806317DC2DFED39E09DSDF3S3287AFE8 Reported State: Last Updated: Last Updated: 2023-02-20107:39:29.6870527Z		
f:state: pykube-ng Operation: Update Time: 2023-02-20107:39:31Z Resource Version: 555854366 UID: dc9e051f-0362-4065-8f4b-4d013a4c00d6 Spec: Connection State: Connection State: 0 Last Updated: 2023-02-20107:33:00.9524174Z State: 0 Device Id: fractal-node-8063170C20FED29E0905DF353287AFE8 Reported State: Last Updated: Last Updated: 2023-02-20107:39:29.6870527Z		astindated
Manager: pykube-ng Operation: Update Time: 2023-02-20107:39:312 Resource Version: 555850366 UID: dc9e051f-8362-4065-8f4b-4d013a4c08d6 Spec: connection State: Connection State: connected Desired State: 0 Device Id: fractal-node-806317DC2DFED39E0905DF353287AFE8 Reported State: Last Updatedt: Last Updatedt: 2023-02-20107:39:29.6870527Z		
Operation: Update Time: 2023-02-201097:39:31Z Resource Version: 555854366 UID: dc9e951f-8362-4065-8f4b-4d013a4c08d6 Spec: Connection State: Connection State: Connected Desired State: 2023-02-20107:33:00.9524174Z State: 0 Device Id: fractal-node-8063170C2DFED29E09D5DF353287AFE8 Reported State: Last Updated: Last Updated: 2023-02-20107:39:29.6870527Z		
Time: 2023-02-20107:39:31Z Resource Version: 555584366 UID: dc9e051f-8362-4065-8f4b-4d013a4c60d6 Spec: Connection State: Connection State: Connected Desired State: 0 Device Id: fractal-node-806317DC2DFED29E0905DF353287AFE8 Reported State: Last Updated: Last Updated: 2023-02-20107:39:29.68706527Z		
Resource Version: 555854366 UD: dc9e051f-8362-4065-8f4b-4d013a4c08d6 Spec: Connection State: Connection State: Last Updated: 1287 2023-02-20107:33:00.9524174Z State: Device Device fractal-node-8063170C2DFED29E09D5DF353287AFE8 Reported State: Last Updated: Last Updated: 2023-02-20107:39:29.6870627Z		
UTD: dc9e851f-8362-4065-8f4b-44013a4c68d6 Spec: Connection State: Connected Desired State: 2023-02-20T07:33:00.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20T07:39:29.6870527Z		
Spec: Connected Desired State: Last Updated: 2023-02-20107:33:00.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED29E09D5DF353287AFE8 Reported State: 1 Last Updated: 2023-02-20107:39:29.68705277		
Connection State: Connected Desired State: Last Updated: 2023-02-20107:33:00.9524174Z State: 0 Device Id: fractal-node-8063170C2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20107:39:29.68706527Z Device Id: Fractal-node-8063170C2DFED29E0905DF353287AFE8		
Desired State: Last Updated: 2023-02-20T07:33:00.9524174Z State: 0 Device Id: fractal-node-806317DC2DFED29E0905DF353287AFE8 Reported State: Last Updated: 2023-02-20T07:39:29.6870627Z		State: Connected
Last Updated: 2023-02-20107:33:00.9524174Z State: Device Id: fractal-node-8063170C2DFED29E09D5DF353287AFE8 Reported State: Last Updated: 2023-02-20107:39:29.68706277Z		
State: 0 Device Id: fractal-node-806317DC2DFED29E09D5DF353287AFE8 Reported State: Last Updated: 2023-02-20T07:39:29.6870627Z		
Reported State: Last Updated: 2023-02-20T07:39:29.6870527Z		
Reported State: Last Updated: 2023-02-20T07:39:29.6870527Z	Device Id:	fractal-node-806317DC2DFED29E09D5DF353287AFE8
Last Updated: 2023-02-20T07:39:29.6870527Z		

Figure 29 - The description of the connected to the Kubernetes device

The figures below show the connected device and its green LED for the connection status.



Figure 30 - The connected device with its serial number: 8063

Copyright © FRACTAL Project Consortium	72 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4



Figure 31 - Device connected with the green LED

Expected Results: during the connection period green LED is dim. When the device is connected the green LED is turned on. In this way, having connected the device to the internet is possible to notice that it is accepted and connected to the Cloud Platform.

Steps	
Step 1	Power up the device
Step 2	Connect the device to internet

Table 73 - Steps for Validation Test T01_WP6T62-06

8.6.2.2 T02_WP6T62-06 - Testing the communication from the Device to the Cloud

Step 1: Power up the device.

In this step, the device is connected to a power source with 5V to start it correctly.

Expected Results: having booted correctly the device is possible to notice that the red LED switch on as expected.

Step 2: Connect the device to the Internet.

In this step, the device is connected to an Access Point using SSID and password parameters. The figure below shows that the device is connected to the network.



Figure 32 - Device connected with the green LED

Copyright © FRACTAL Project Consortium	73 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Expected Results: During the connection period green LED is dim. When the device is connected the green LED is turned on. In this way, having connected the device to the Internet is possible to notice that it is accepted and connected to the Cloud Platform.

Step 3: Change the device status

There is a button on the device that changes its desired state. To perform this step, we simply press the button on the device. The figures below show that the device status is changed, also the status LEDs are changed from red to green.



Figure 33 - The desired state is OFF before pressing the button

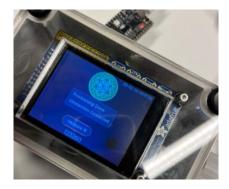


Figure 34 - The device before pressing button

The following figures are shown that the desired state and reported flags of the "fractal-node-7AEDXXXX" are changed, and its related LEDs green is switched on.



Figure 35 - The status of the device is changed in IoT hub after pressing button

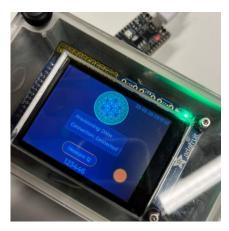


Figure 36 - The device LED is green after pressing button

Copyright © FRACTAL Project Consortium	74 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Expected Results: having connected the device on the Internet and having changed the device status is possible to notice that is reported status on the Cloud Platform side is as expected. In this way, it was tested the communication from the Device to the Cloud Platform.

Steps	
Step 1	Power up the device
Step 2	Connect the device to internet
Step 3	Change device status

Table 74 - Steps for Validation Test T02_WP6T62-06

8.6.2.3 T03_WP6T62-06 - Testing the communication from the Device to the Cloud

Step 1: Power up the device.

In this step, the device is connected to a power source with 5V to start it correctly.

Expected Results: having booted correctly the device is possible to notice that the red LED switch on as expected.

Step 2: Connect the device to the Internet.

In this step, the device is connected to an Access Point using SSID and password parameters. The figure below shows that the device is connected to the network.



Figure 37 - Device connected with the green LED

Expected Results: During the connection period green LED is dim. When the device is connected the green LED is turned on. In this way, having connected the device to the Internet is possible to notice that it is accepted and connected to the Cloud Platform.

Step 3: Change the status in Kubernetes by the "patch" method.

In this step, there is a Kubernetes method called "kubectl patch" that updates the Kubernetes object or updates the running configuration. It uses "json merge" patch type passing json message with "desired state: 1" or "desired state: 0". The structure of the command is shown below.

Copyright © FRACTAL Project Consortium 7.	5 of 149
---	----------

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

kubectl patch -n low-end-ctrl lowends.fractal-cluster.eu low-end-5r2gf --type merge --patch '{ "spec": { "desiredState": { "state": 1} }}'



Figure 38 - Running the patch command using K8s CLI (Updating the time is ignored here)

In the following figures is shown that the desired state of the Low End node is changed from 0 to 1 and its related LED green is switched on.

Spec:	
Connection State	: Connected
Desired State:	
Last Updated:	2023-02-20T07:33:00.9524174Z
State:	1
Device Id:	fractal-node-806317DC2DFED29E09D5DF353287AFE8
Reported State:	
Last Updated:	2023-02-20T07:39:29.6870527Z
State:	0

Figure 39 - Desired state of the device is changed to 1 in Kubernetes CRD (ON)

Normal Logging	27s	kopf	Updating is processed: 1 succeeded; 0 failed.
Normal Logging	27s	kopf	Handler 'update_fn' succeeded.

Figure 40 - Kubernetes log showing the invocation of update function



Figure 41 - The state in IoT hub has been switched to green

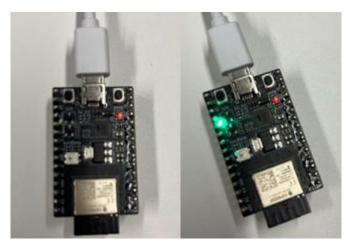


Figure 42 - The device before the patch command on the left, and after the patch command on the right

Expected Results: having connected the device on the Internet and having changed the device status on the Cloud Platform side is possible to notice that its reported status on the Device side is as expected. In this way, it was tested the communication from the Cloud Platform and the Device.

	Copyright © FRACTAL Project Consortium	76 of 149
--	--	-----------

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps	
Step 1	Power up the device
Step 2	Connect the device to internet
Step 3	Change status in Kubernetes by "patch" method
	ubuntugmaster: \$ kubectl patch -n low-end-ctrl lowends.fractal-cluster.eu low-end-5r2gftype mergepatch '{ "spec": { "desiredState": { "state": 1} }}' lowend.fractal-cluster.eu/low-end-5r2gf patched

Table 75 - Steps for Validation Test T03_WP6T62-06

8.6.2.4 T04_WP6T62-06 - Testing the tasks scheduling running Nuttx

Step 1: Power up the device.

In this step, the device is connected to a power source with 5V to start it correctly.

Expected Results: having booted correctly the device is possible to notice that the red LED switch on as expected.

Step 2: Connect the device by USB to the PC.

In this step, the device is connected to the PC via USB (that can also power the device).

At Linux PC with "dmesg" command is possible to show all the USB peripheral connected to the device:

\$dmesa [598658.253187] usb 3-2.1.4: new full-speed USB device number 16 using xhci_hcd [598658.338481] usb 3-2.1.4: New USB device found, idVendor=10c4, idProduct=ea60, bcdDevice= 1.00 [598658.338492] usb 3-2.1.4: New USB device strings: Mfr=1, Product=2, SerialNumber=3 [598658.338497] usb 3-2.1.4: Product: CP2102N USB to UART Bridge Controller [598658.338501] usb 3-2.1.4: Manufacturer: Silicon Labs [598658.338504] usb 3-2.1.4: SerialNumber: 363aaf4f44a0eb11a2a8cbacdf749906 3-2.1.4:1.0: [598658.340065] cp210x cp210x converter detected [598658.345156] usb 3-2.1.4: cp210x converter now attached to ttyUSB1

Figure 43 - dmesg command

The device is at ttyUSB1 port.

Step 3: Open the terminal connection and run the command "ps".

After connecting the device to the PC we run the process status (ps) command. It is used to get information about currently running processes and their PIDs in your system.

Using the command below is possible to access at the serial console of the device.

\$ minicom -D /dev/ttyUSB1

offcode>

Figure 44 - Command to open the terminal of device

Copyright © FRACTAL Project Consortium	77 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

The "ps" command and its related information are shown below:

offcor	de> de> ps	
	•	DLICY TYPE NPX STATE EVENT SIGMASK STACK COMMAND
0	0 0 FIFO	Kthread N Ready 0000000 004048 Idle Task
1	1 224 RR	Kthread Waiting Semaphore 00000000 004016 hpwork 0x3fc844d8
2	2 224 RR	Kthread Waiting Semaphore 00000000 004016 hpwork 0x3fc844d8
3	3 100 RR	Kthread Waiting Semaphore 00000000 004016 lpwork 0x3fc844ec
4	4 100 RR	Kthread Waiting Semaphore 00000000 004016 lpwork 0x3fc844ec
5	5 100 RR	Task Running 00000000 001968 nsh_main
6	6 223 RR	Kthread Waiting Semaphore 00000000 001968 rt_timer
7	7 253 RR	Kthread Waiting MQ empty 00000000 006592 wifi
8	8 100 RR	Task Waiting Signal 00000000 002944 NTP daemon
0.poc	l.ntp.org;1.p	ool.ntp.org;2.pool.ntp.org
10	10 100 RR	Task Waiting Signal 00000000 016320 fractal
11	10 100 RR	pthread Waiting Signal 00000000 003024 pt-0x420347ee
0x3fc	b0bd0	
12	10 100 RR	pthread Waiting Signal 00000000 003024 pt-0x420347ee
0x3fc	b6800	
offcod	le>	

Figure 45 - ps command

Expected Results: having connected the device to the PC and after running the "ps" command on the device terminal, the result is as expected. This command prints all the parallel tasks that are running on the device.

Steps	
Step 1	Power up the device
Step 2	Connect the device by usb to a pc
Step 3	Open terminal connection and run command "ps"
	<pre>\$ minicom -D /dev/ttyUSB1 offcode> offcode> ps</pre>

Table 76 - Steps for Validation Test T04_WP6T62-06

8.6.3 Test environment setup

First of all, we put in the config.yaml file the IoT hub credential that should be encoded to base64. After that, we defined a Wi-Fi configuration with parameters SSID and password for the local network.

Other configuration and information of the Low End Node Device at the GitHub Repository, in the following link <u>https://github.com/project-fractal/WP6T62-06-low-end-node-orchestrator</u>.

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.6.4 Test execution

8.6.4.1 T01_WP6T62-06 - Testing the connection between the Device and the Cloud Platform

Results/Evi	idence	
low low low low low low low uow uow		
	IoT Hub Devices: Disconnected Tractal-node- 17891F32C7C7C5F700 Disconnected Tractal-node- 904ADF13C9AD3DE1FCC Disconnected Tractal-node- A62253BE425CE8093CA Connected Tractal-node- 806317DC2DFED29E091 Disconnected	SFB05884F4D06 Period Masse From C C C 48EB3F731BD00D Period Perioted Masse From C C C 6839375D6BEF1 Perioted Masse From C C C
Success crit	teria	
Device acce	epted and dev	ice status "connected"
Test observ		
Test config	uration	Device access point config: SSID and PW
Test condit	tions	Device connected to local wifi
Remarks		
Test result		
Passed		

Table 77 - Results of the test T01_WP6T62-06

Copyright © FRACTAL Project Consortium	79 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.6.4.2 T02_WP6T62-06 – Testing the communication from the Device to the Cloud Platform

Results/I	Evidence	•							
Step 2:	Connected fractal-node- 806317DC2DFED29E0	9D5DF353287AFE8	Desired	Reported	Measure	B Remove	U	ø	
Step 3:	Connected fractal-node- 7AEDFBE5EF120C2F7BF	399D1B03E3E9F	Desired	Reported	Measure	Il Remove	0	ø	
Success of	criteria								
	d status as expe	cted							
	ervations	1							
	figuration	Device access				and P\	N		
Test con		Device connec	ted to	local	wifi				
Remarks									
Test resu	ult								
Passed									

Table 78 - Results of the test T02_WP6T62-06

Copyright © FRACTAL Project Consortium	80 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.6.4.3 T03_WP6T62-06 - Testing the communication from the Cloud Platform to the Device

Results/Ev	vidence	
Step 2:	Connected fractal-node- 806317DC2DFED2	9E09D5DF353287AFE8
Step 3:	State: Device Id: Reported S Last Upd State: Normal Loggin Normal Loggin Connected fractal-node-	<pre>sate: lated: 2023-02-20T07:33:00.9524174Z fractal-node-806317DC2DFED29E09D5DF353287AFE8 state: lated: 2023-02-20T07:39:29.6870527Z 0 g 27s kopf Updating is processed: 1 succeeded; 0 failed.</pre>
Success cr	iteria	
		vice status "connected"
Test obse	rvations	
Test confi	guration	Device access point config: SSID and PW
Test cond	itions	Device connected to local wifi
Remarks		
Test resul	t	
Passed		

Table 79 - Results of the test T03_WP6T62-06

Copyright © FRACTAL Project Consortium	81 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.6.4.4 T04_WP6T62-06 - Testing the tasks scheduling running Nuttx on the device

Results/E	vidence		
Step 2:	[598658.338481] bcdDevice= [598658.338492] SerialNumber=3 [598658.3385047] [598658.338504] [598658.338504] [598658.340065]	usb 3-2.1.4: Product: CP2102N USB to UART Bridge Controller usb 3-2.1.4: Manufacturer: Silicon Labs usb 3-2.1.4: SerialNumber: 363aaf4f44a0eb11a2a8cbacdf749906	
Step 3:	0 0 0 FIFO 1 1 224 RR 2 2224 RR 3 3 100 RR 4 4 100 RR 5 5 100 RR 6 6 223 RR 7 7 253 RR 8 8 100 RR	ICY TYPE NPX STATE EVENT SIGMASK STACK COMMAND Kthread Waiting Semaphore 0000000 004016 hpwork 0x3fc844d8 Kthread Waiting Semaphore 0000000 004016 hpwork 0x3fc844d8 Kthread Waiting Semaphore 0000000 004016 hpwork 0x3fc844d8 Kthread Waiting Semaphore 0000000 004016 hpwork 0x3fc844de Kthread Waiting Semaphore 0000000 004016 hpwork 0x3fc844ec Task Waiting Semaphore 0000000 001587 t_timer Kthread Waiting Modemphore 00000000 005592 wifi Task Waiting Signal 00000000 016320 fractal Inth.org.jc.pol.ntp.org Task Waiting Signal 00000000 00324 pt-0x420347ee	
Success c	riteria		
Print all p	arallel runnin	g task as expected	
Fest obse	ervations		
Test configuration		GitHub repository: https://github.com/project-fractal/WP6T62-06-low-end-node- orchestrator Device access point config: SSID and PW	
Test conditions		Device connected to local wifi	
Remarks			
Test resu	lt		
Passed			

Table 80 - Results of the test T04_WP6T62-06

Copyright © FRACTAL Project Consortium	82 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7 Hardware-level Edge Controller

The hardware Edge Controller controls the underlying hardware, such as communications and computations. The underlying hardware is based on a network-on-chip (NoC) multicore architecture that supports heterogeneous cores connected via NoC. The description of the underlying NoC-based multicore architecture is reported in contribution D4.4 (WP4). It is a time-triggered extension layer used in the VERSAL NoC to establish the temporal partitioning over the Chip.

In order to allow multiple nodes communication, within WP6, one of the NoC cores was devoted to function as a hardware gateway controller. It allows communication between on-chip and off-chip networks.

About the validation work in T6.3, we focused on the Network Gateway Interface that connect both on-chip and off-chip.

8.7.1 Test planification

8.7.1.1 Define the testing scope and identify the functionality that needs to be tested

In D6.2, several services are presented for this network hardware gateway architecture. Some of them are for future extensions, and they are not in this project scope.

This section will focus on following functionalities:

- 1. Message-Classification and Message-Scheduling Service;
- 2. Egress-Queuing and Ingress-Queuing Service;
- 3. Serialization Service.

For the first functioning, three test cases were defined, and they are shown below. Message scheduling refers to timing and port definition according to configuration, so we check messages from sending port to destination, seeing timing and message content correctness.

Message classification refers to the ability of the NGW to manage different types of messages. They are TT (Time Triggered), RATE (Rate Constraint), BE (Best Effort), but in the project scope, only TT messages are used.

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

a) T01_WP6T62-0X – Testing the Message-Classification and Message-Scheduling Services at different port according to scheduling configuration

	Validation Test		
Test ID	T01_WP6T62-0X		
Test type	Functional		
	Testing the Message-Classification and Message-Scheduling Services at different port		
Test name	according to scheduling configuration		
Date	03/02/2022		
Tester's Name	Luca Visconti (Akkodis)		
Test scope or objective			
The objective of this test is to validate the message-classification and message-scheduling services sending 3			
message at different port and according to the scheduling configuration.			

Figure 46 - Validation Test T02_WP6T62-0X

b) T02_WP6T62-0X – Testing the Message-Classification and Message-Scheduling Services at same port according to scheduling configuration

Validation Test			
Test ID	T02_WP6T62-0X		
Test type	Functional		
Test name	Testing the Message-Classification and Message-Scheduling Services at same port		
	according to scheduling configuration		
Date	03/02/2022		
Tester's Name	Luca Visconti (Akkodis)		
Test scope or objective			
The objective of this test is to validate the message-classification and message-scheduling services sending			
message at same port according to scheduling configuration			

Figure 47 - Validation Test T02_WP6T62-0X

c) T03_WP6T62-0X – Testing the Message-Classification and Message-Scheduling Services at same NI

Validation Test		
Test ID	T03_WP6T62-0X	
Test type	Functional	
Test name	Testing the Message-Classification and Message-Scheduling Services at same NI	
Date 03/02/2022		
Tester's Name	Luca Visconti (Akkodis)	
Test scope or objective		
The objective of this test is to validate the message-classification and message-scheduling services sending		
message at same NI when have 2 port configured		

Figure 48 - Validation Test T03_WP6T62-0X

Copyright © FRACTAL Project Consortium	84 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

For the second functioning, one test case was defined, and it is shown below. Each NI has a queue for each port, and it can be configured as ingress or egress. Here we check the capacity of these queues using different lengths.

d) T04_WP6T62-0X – Testing the Ingress and Egress-queuing Services

Validation Test		
Test ID	T04_WP6T62-0X	
Test type	Functional	
Test name	Testing the Ingress and Egress-queuing Services	
Date	03/02/2022	
Tester's Name	Luca Visconti (Akkodis)	
Test scope or objective		
The objective of this test is to validate the ingress-queuing and egress-queuing services using the max lenght of		
queue.		

Figure 49 - Validation	Test T04_WP6T62-0X
------------------------	--------------------

For the third functioning one test case was defined and it is shown below:

e) T05_WP6T62-0X – Testing the serialization services

Serializer component is a subcomponent of NGW. As, it is not possible to see what happens inside the NGW, we defined according to the developers to run a simulation in order to proof the correct behaviors.

The Scope of the serializer is to take a message from the off-chip network and to provide a protocol conversion. It also works from on-chip to off-chip network in the same way.

Validation Test		
Test ID	T05_WP6T62-0X	
Test type	Functional	
Test name	Testing the serialization services	
Date	03/02/2022	
Tester's Name Luca Visconti (Akkodis)		
Test scope or objective		
The objective of this test is to validate the serialization service on NGW out port		

Figure 50 - Validation Test T05_WP6T62-0X

Copyright © FRACTAL Project Consortium	85 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7.2 Test case development

8.7.2.1 T01_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at different port according to scheduling configuration

Step 1: Define a set of messages to send (3 messages at 3 different ports according to the scheduling configuration TTCommSched.cfg of each NI).

- One message to NI0 on port 2

Port ID	Instant	Next	Instant (microsec)
2 [NI0-NI2]	956 (7)	0	28,483

Figure 51 - Information described in "ttcommsched.cfg" on NIO

- Two messages to NI1 respectively on port 2 and 3

Port ID	Instant	Next	Instant (microsec)
2 [NI1-NI0]	1092 (8)	1	32,552
3 [NI1-NI3]	2594 (19)	0	77,311

Figure 52 - Information described in "ttcommsched.cfg" on NI1

In the "ttcommsched.cfg" on the NIO we have defined the following row:

- 0000000000000000000000000003BC020

In the "ttcommsched.cfg" on the NI1 we have defined the following rows:

The "hw.cfg" file is the same for all NIs, so it has the following rows:

- 04 1 1601010C000000000000000 1601010A00000000000000 1006 1006 1006 1006

Step 2: Send messages to NIO and to NI1.

In the architecture, we have four processing elements, and one processing element is working as a network gateway, and the rest of the processing elements are working as cores.

Copyright © FRACTAL Project Consortium	86 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

We have created this file in a C language that simply creates and send the message on a specific NI and Port. The first message is sent to NIO on port 2, the second message is sent to NI1 on port 2 and the third message is sent to NI1 on port 3.

```
// send message from NIO on port 2
for(int i=0;i<mesg_size;i++)
{
    Xil_Out32(XPAR_SKINOC_0_SO0_AXI_BASEADDR+131072,i); // Write Message on NIO, port2
}
Xil_Out32(XPAR_SKINOC_0_SO0_AXI_BASEADDR+163840,7); // Terminate the Message
// send messages from NII on port 2
for(int i=0;i<mesg_size;i++)
{
    Xil_Out32(XPAR_SKINOC_0_SO1_AXI_BASEADDR+131072,i); // Write Message on NIO, port2
}
Xil_Out32(XPAR_SKINOC_0_SO1_AXI_BASEADDR+163840,7); // Terminate the Message
// send messages from NII on port 3
for(int i=0;i<mesg_size;i++)
{
    Xil_Out32(XPAR_SKINOC_0_SO1_AXI_BASEADDR+196608,i); // Write Message on NIO, port2
}
Xil_Out32(XPAR_SKINOC_0_SO1_AXI_BASEADDR+196608,i); // Write Message on NIO, port2
}
Xil_Out32(XPAR_SKINOC_0_SO1_AXI_BASEADDR+196608,i); // Write Message on NIO, port2
}
</pre>
```

Figure 53 - Send messages scripts

Expected Results: In this case, having defined and sent messages to different NI, we expect to receive at the defined port the complete message. As depicted in the figures below is possible to see that the messages are arrived at the destination port and period according to the scheduling. The message content is also not corrupted.

Copyright © FRACTAL Project Consortium	87 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Release 2021.1 Feb 7 2023 - 07:42:21
PMU-FW is not running, certain applications may not be supported.
NIO - NI2 , recived_data=0 , original_data = 0
NIO - NI2 , recived data=1 , original data = 1
NIO - NI2 , recived data=2 , original data = 2
NIO - NI2 , recived data=3 , original data = 3
NIO - NI2 , recived_data=4 , original_data = 4
NIO - NI2 , recived_data=5 , original_data = 5
NIO - NI2 , recived_data=6 , original_data = 6
NIO - NI2 , recived_data=7 , original_data = 7
NIO - NI2 , recived_data=8 , original_data = 8
NIO - NI2 , recived data=9 , original data = 9
NIO - NI2 , recived_data=10 , original_data = 10
NIO - NI2 , recived_data=11 , original_data = 11
NIO - NI2 , recived data=12 , original data = 12
NIO - NI2 , recived_data=12 , original_data = 12 NIO - NI2 , recived_data=13 , original_data = 13
NIO - NI2 , recived_data=13 , original_data = 13 NIO - NI2 , recived_data=14 , original_data = 14
NI1 - NI0 , recived data=0 , original data = 0
NII - NIO , recived_data=0 , original_data = 0 NII - NIO , recived_data=1 , original_data = 1
NII - NIO , recived_data=1 , original_data = 1 NII - NIO , recived_data=2 , original_data = 2
NI1 - NI0 , recived_data=2 , original_data = 3
NII - NIO , recived_data=3 , original_data = 3
NII - NIO , recived_data=4 , original_data = 4 NII - NIO , recived_data=5 , original_data = 5
NII - NIO , recived data=6 , original data = 6
NII - NIO , recived_data=0 , original_data = 0 NII - NIO , recived_data=7 , original_data = 7
NII - NIO , recived data= 7 , original data = 8
NII - NIO , recived data=9 , original data = 9
NII - NIO , recived data=10 , original data = 10
NII - NIO , recived_data=10 , original_data = 10
NII - NIO , recived_data=11 , original_data = 11 NII - NIO , recived_data=12 , original_data = 12
NII - NIO , recived_data=12 , original_data = 12 NII - NIO , recived_data=13 , original_data = 13
NII - NIO , recived data=13 , original data = 13
NII - NIO , Tectved_data=14 , of tgthat_data = 14
NI1 - NI3 , recived_data=0 , original_data = 0
NII - NIS , recived data=0 , original_data = 0 NII - NIS , recived data=1 , original data = 1
NII - NIS , recived_data=1 , original_data = 1 NII - NIS , recived_data=2 , original_data = 2
NII - NIS , recived data=2 , original data = 2 NII - NIS , recived data=3 , original data = 3
NI1 - NI3 , recived_data=4 , original_data = 4 NI1 - NI3 , recived_data=5 , original_data = 5
NII - NIS , Fectived_data=5 , original_data = 5 NII - NIS , recived_data=6 , original_data = 6
NII - NIS , recived_data=0 , original_data = 0 NII - NIS , recived_data=7 , original_data = 7
NII - NIS , reclived data= 7 , or ignal data = 7
NI1 - NI3 , recived_data=8 , original_data = 8 NI1 - NI3 , recived data=9 , original data = 9
NII - NIS , recived_data=9 , original_data = 9
NII - NIS , recived_data=10 , original_data = 10 NII - NIS , recived_data=11 , original_data = 11
NII - NIS , recived_data=11 , original_data = 11 NII - NIS , recived_data=12 , original_data = 12
NII - NIS , recived_data=12 , original_data = 12 NII - NIS , recived data=13 , original data = 13
NII - NIS , recived data=15 , original data = 15 NII - NIS , recived data=14 , original data = 14
NII - NIS , Tectved_data=14 , ortgtnat_data = 14

Figure 54 - Messages received

Furthermore, as shown in the figure below, the injection time and the different time between messages are correct according to configuration files.

Time difference between m2, m1 =4 micro second
Time difference between m3, m2 =44 micro second Pass
Time difference between m3. m1 =48 micro second
Pass
Injection time of m1 =1122987 micro second
Injection time of m2 =1122991 micro second
Injection time of m3 =1123036 micro second

Figure 55 - Time different and Injection time

Copyright © FRACTAL Project Consortium	88 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps							
Step 1	Define a set of me	Define a set of messages to send (3 messages at 3 different port according to the					
	scheduling config	uration TTCommSc	held.cfg of each NI)			
	- 1 message to NI) on the port 2					
	Port ID	Instant	Next	Instant (microsec)			
	2 [NI0-NI2]	956 (7)	0	28,483			
	- 2 messages to N	I1 on the port 2 and	d 3				
	Port ID	Instant	Next	Instant (microsec)			
	2 [NI1-NI0]	1092 (8)	1	32,552			
	3 [NI1-NI3]	2594 (19)	0	77,311			
	Send a messages	to NI1:					
	// send messages from NI1 on port 2						
	<pre>for(int i=0;i<mes <="" pre=""></mes></pre>	g_size;i++)					
	Xil_Out32(XPAR_SKINOC_0_S01_AXI_BASEADDR+131072,i); // Write Message on NI0, port2						
	<pre>Xil_Out32(XPAR_SKINOC_0_S01_AXI_BASEADDR+163840,7); // Terminate the Message</pre>						
	// send messages from NI1 on port 3						
	<pre>for(int i=0;i<mesg_size;i++)< pre=""></mesg_size;i++)<></pre>						
	<pre>{ Xil_Out32(XPAR_SKINOC_0_S01_AXI_BASEADDR+196608,i); // Write Message on NI0, port2</pre>						
	<pre>} Xil Out32(XPAR SKINOC 0 S01 AXI BASEADDR+229376,7); // Terminate the Message</pre>						
	_ ,			-			

Table 81 - Steps for Validation Test T01_WP6T62-0X

8.7.2.2 T02_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at the same port according to scheduling configuration

Step 1: Define a set of messages to send (3 messages from the same port according to the scheduling configuration TTCommScheld.cfg of NI1)

Port ID	Instant	Next	Instant (microsec)
2 [NI1-NI0]	1092 (8)	1	32,552
3 [NI1-NI3]	2594 (19)	0	77,311

- Three messages to NI1 on the port 2:

Figure 56 - Information described in "ttcommsched.cfg" on NI1

In the "ttcommsched.cfg" on the NI1 we have defined the following rows:

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

The "hw.cfg" file is the same for all NIs, so it has the following rows:

- 04 1 1601010C000000000000000 1601010A00000000000000 1006 1006 1006 1006

Step 2: Send messages to NI1

We have created this file in a C language that simply create and send the message on a specific NI and Ports.

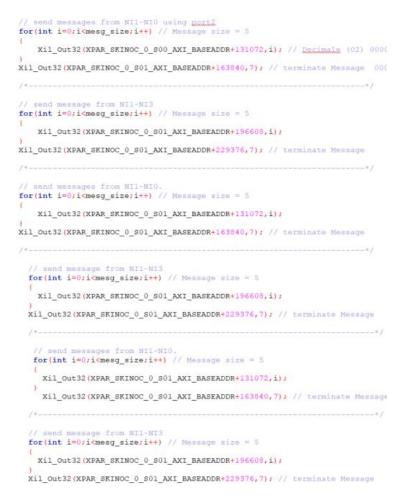


Figure 57 - Send messages scripts

Copyright © FRACTAL Project Consortium	90 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Expected Results: In this case, having defined and sending messages to the same NI, we expect to receive at the defined port the complete message. In the figures below is possible to see that the messages arrived at the destination port and period according to the scheduling. The message content is also not corrupted.

NI1 - NI3 , recived_data=0 , original_data = 0
NI1 - NI3 , recived_data=1 , original_data = 1
NI1 - NI3 , recived_data=2 , original_data = 2
NI1 - NI3 , recived_data=3 , original_data = 3
NTA NTO sectored data F sectored data F
NII - NI3 , recived_data=6 , original_data = 6 NI1 - NI3 , recived data=7 , original data = 7
NI1 - NI3 , recived data=8 , original data = 8
NI1 - NI3 , recived data=9 , original data = 9
NI1 - NI3 , recived_data=10 , original_data = 10
NI1 - NI3 , recived data=10 , original data = 10
NII - NI3 , recived_data=12 , original_data = 12 NI1 - NI3 , recived_data=13 , original_data = 13
NI1 - NI3 , recived_data=14 , original_data = 14
NI1 - NI3 , recived data=0 , original data = 0
NI1 - NI3 , recived_data=1 , original_data = 1 NI1 - NI3 , recived_data=2 , original_data = 2
NTA NTO sectored data as sectored data as
NI1 - NI3 , recived_data=9 , original_data = 9 NI1 - NI3 , recived_data=10 , original_data = 10
NII - NIS , recived data=10 , original data = 10
NII - NI3 , recived_data=12 , original_data = 12 NI1 - NI3 , recived data=13 , original data = 13
NI1 - NI3 , recived_data=14 , original_data = 14
NI1 - NI3 , recived data=0 , original data = 0
NII - NI3 , recived_data=3 , original_data = 3 NI1 - NI3 , recived_data=4 , original_data = 4
NTA NTO sectored data as sectored data as
NTA NTA control data 7
NII - NI3 , recived_data=8 , original_data = 8 NI1 - NI3 , recived_data=9 , original_data = 9
NII - NIS , recived data=9 , original data = 9 NII - NIS , recived data=10 , original data = 10
NII - NIS , recived_data=10 , original_data = 10 NII - NIS , recived_data=11 , original_data = 11
NII - NIS , recived data=11 , original data = 12
NII - NIS , recived_data=12 , original_data = 12 NII - NIS , recived_data=13 , original_data = 13
NII - NIS , recived data=13 , original data = 14
$\frac{1}{1}$ $\frac{1}$

Figure 58 - Messages received

Furthermore, as shown in the figure below, the injection time and the period between messages are correct according to configuration files. Each message is received in a different period.

Time difference between two consecutive messages m2 and m1 = 122 Time difference between two consecutive messages m3 and m2 = 122 $\,$

Figure 59 - Time different between messages

Copyright © FRACTAL Project Consortium	91 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps				
Step 1	the scheduling co	-	messages at 3 from nmScheld.cfg of NIC	n the same port according))
	Port ID	Instant	Next	Instant (microsec)
	2 [NI0-NI2]	956 (7)	0	28,483
Step 2	<pre>for(int i=0;:</pre>	Ages from NII-NIO using icmesg_size;i++) // Men 2 (XPAR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS Age from NII-NI3 icmesg_size;i++) // Men 2 (XPAR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_BAS AR_SKINOC_0_SO1_AXI_SO AR_SKINOC_0_SO_SO AR_SKINOC_0_SO A	<pre>sage size = 5 I_BASEADDR+131072,i); // te ssage size = 5 I_BASEADDR+196608,i); SEADDR+229376,7); // te ssage size = 5 I_BASEADDR+131072,i); SEADDR+163040,7); // te dessage size = 5 I_BASEADDR+196608,i); BASEADDR+196608,i); Message size = 5 KI_BASEADDR+229376,7); // Message size = 5 KI_BASEADDR+163040,7); </pre>	rminate Message */ rminate Message */ terminate Message */

Table 82 - Steps for Validation Test T02_WP6T62-0X

8.7.2.3 T03_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at the same NI

Step 1: Define a set of messages to send (3 messages at 2 ports according to the scheduling configuration TTCommScheld.cfg of NI2)

- Two messages to NI2 on the port 1
- One message to NI2 on the port 2

Port ID	Instant	Next	Instant (microsec)
2 [NI2-NI0]	2048 (15)	1	61,035
3 [NI2-NI3]	2185 (16)	0	65,104

Figure 60 - Information described in "ttcommsched.cfg" on NI2

	Copyright © FRACTAL Project Consortium	92 of 149
--	--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

In the "ttcommsched.cfg" on the NI2 we have defined the following rows:

- 000000000000000000000000000889030

The "hw.cfg" file is the same for all NIs, so it has the following rows:

```
- 04

1

1601010C000000000000000

1601010A00000000000000

1006

1006

1006

1006

1006
```

Step 2: Send messages to NI2:

We have created this file in a C language that simply creates and sends the message on a specific NI and Port. The first message is sent to NI2 on port 2, the second message is sent to NI2 on port 3 and the third message is sent to NI2 on port 2.

Figure 61 - Send messages scripts

Expected Results: In this case, having defined and sending messages to the same NI we expect to receive at the defined port the complete message. In the figures below is possible to see that the messages arrived at destination port and period according to the scheduling. The message content is also not corrupted.

Copyright © FRACTAL Project Consortium	93 of 149
--	-----------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

NTA	NTO		recived data=0 . original data = 0
NI1			······································
NI1			recived_data=1 , original_data = 1
NI1			recived_data=2 , original_data = 2
NI1			recived_data=3 , original_data = 3
NI1			recived_data=4 , original_data = 4
NI1			recived_data=5 , original_data = 5
NI1	NIO		recived_data=6 , original_data = 6
NI1	NIO		recived_data=7 , original_data = 7
NI1			recived_data=8 , original_data = 8
NI1	NIO		recived_data=9 , original_data = 9
NI1	1110		recived_data=10 , original_data = 10
NI1	NIO		recived_data=11 , original_data = 11
NI1	1110		recived_data=12 , original_data = 12
NI1			recived_data=13 , original_data = 13
NI1	NI0		recived_data=14 , original_data = 14
NI1	 NI3		recived_data=0 , original_data = 0
NI1	NI3		recived data=1 , original data = 1
NI1		,	
NI1	NI3		recived_data=2 , original_data = 2 recived_data=3 , original_data = 3
NI1			
NI1		,	recived_data=4 , original_data = 4 recived data=5 , original data = 5
NI1		,	recived_data=6 , original_data = 6
NI1		2	recived_data=7 , original_data = 7
NI1		2	recived_data=8 , original_data = 8
NI1	NI3	,	recived data=9 , original data = 9
NI1	NI3	<i>.</i>	recived_data=10 , original_data = 10
NI1	NI3	ĺ.	recived_data=11 , original_data = 11
NI1		ĺ.	recived_data=12 , original_data = 12
NI1	NI3	1	recived_data=13 , original_data = 13
NI1	NI3	,	recived_data=14 , original_data = 14
NI1	NI0		recived_data=0 , original_data = 0
NI1	NI0		recived_data=1 , original_data = 1
NI1	NI0		recived_data=2 , original_data = 2
NI1	NI0		recived_data=3 , original_data = 3
NI1	NI0		recived_data=4 , original_data = 4
NI1			recived_data=5 , original_data = 5
NI1			recived_data=6 , original_data = 6
NI1			recived_data=7 , original_data = 7
NI1			recived_data=8 , original_data = 8
NI1			recived_data=9 , original_data = 9
NI1			recived_data=10 , original_data = 10
NI1	NIO		recived_data=11 , original_data = 11
NI1	NIO		recived_data=12 , original_data = 12
NI1	NIO		recived_data=13 , original_data = 13
NI1	NIO		recived_data=14 , original_data = 14

Figure 62 - Messages received

Furthermore, as shown in the figure below, the period between messages is correct according to the configuration files. Two messages are received in the same period and one message is received in the next period.

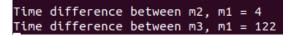


Figure 63 - Time different between messages

Copyright © FRACTAL Project Consortium	94 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps						
Step 1	Define a set of messages to send (3 messages at 2 ports according to the scheduling					
	configuration TT	CommScheld.cfg of	NI2)			
	- 2 messages to I	NI2 on the port 1				
	- 1 message to N	I2 on the port 2				
	Port ID	Instant	Next	Instant (microsec)		
	2 [NI2-NI0]	2048 (15)	1	61,035		
	3 [NI2-NI3]	2185 (16)	0	65,104		
Step 2	Send messages t	o NI2:				
	/* // send messac for(int i=0;id { Xil_Out32 } Xil_Out32(XPAN /*	R_SKINOC_0_S02_AXI ge from NI2-NI3 (mesg_size;i++) // (XPAR_SKINOC_0_S02_AXI ges from NI2-NI0 u 0;i <mesg_size;i++) (XPAR_SKINOC_0_S02_ KPAR_SKINOC_0_S02_</mesg_size;i++) 	Message size = 5 _AXI_BASEADDR+196 _BASEADDR+229376, sing port2 // Message size = _AXI_BASEADDR+131	<pre>*/ 608,i); 7); // terminate Message*/ = 5 072,i);</pre>		

Table 83 - Steps for Validation Test T03_WP6T62-0X

8.7.2.4 T04_WP6T62-0X - Testing the Ingress and Egress-Queuing Services

We have defined the same configuration of Validation Test T01. In the "ttcommsched.cfg" on the NIO we have defined the following row:

- 00000000000000000000000000003BC020

And in the "hw.cfg" file is the same for all NIs, so it has the following rows:

- 04 1 1601010C000000000000000 1601010A00000000000000 1006 1006 1006 1006

We defined a set of messages to send, in particular, three messages from the same port according to the scheduling configuration TTCommScheld.cfg of NIO with a queue length equal to 16:

• Message 1 size: 8-bit, length 10;

Copyright © FRACTAL Project Consortium	95 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

- Message 2 size: 16-bit, length 16;
- Message 3 size: 32-bit, length 50.

Step 1: Send a message with length less than queue length.

We have created this file in a C language that simply creates and sends the message to NIO on port 2. The message has a length of 10. In the configuration file "port.cfg" is defined a queue length equal to 16.

```
for(int i=0;i<mesg_size1;i++)
{
    Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+131072,message1+i);
}
Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>
```

Figure 64 - Send message

Expected Results: In this case, having defined and sent a message with length 10 we expect to receive at the defined port the complete message. In the figures below is possible to see that the message has arrived at the destination port. The message content is also not corrupted.

Message id = 0, recived_data=3 , original_data = 3
Message id = 1, recived_data=4 , original_data = 4
Message id = 2, recived_data=5 , original_data = 5
Message id = 3, recived_data=6 , original_data = 6
Message id = 4, recived_data=7 , original_data = 7
Message id = 5, recived_data=8 , original_data = 8
Message id = 6, recived_data=9 , original_data = 9
Message id = 7, recived_data=10 , original_data = 10
Message id = 8, recived_data=11 , original_data = 11
Message id = 9, recived_data=12 , original_data = 12
Time stamp=37196732
**

Figure 65 - Message received

Step 2: Send a message with length equal than queue length

We have created this file in a C language that simply creates and sends the message to NIO on port 2. The message has a length of 16. In the configuration file "ort.cfg" is defined a queue length equal to 16.

```
for(int i=0;i<mesg_size2;i++)
{
  Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+131072,message2+i);
}
Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>
```

Figure 66 - Send message

Expected Results: In this case, having defined and sent a message with length 16 we expect to receive at the defined port the complete message. In the figures below is possible to see that the message has arrived at the destination port. The message content is also not corrupted.

Copyright © FRACTAL Project Consortium	96 of 149
--	-----------

1.2 10 - 07 - 1 - 1	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

hessage to - 1, received_data-s , or tgthat_data - s
Message id = 2, recived_data=6 , original_data = 6
Message id = 3, recived_data=7 , original_data = 7
Message id = 4, recived_data=8 , original_data = 8
Message id = 5, recived_data=9 , original_data = 9
Message id = 6, recived_data=10 , original_data = 10
Message id = 7, recived_data=11 , original_data = 11
Message id = 8, recived_data=12 , original_data = 12
Message id = 9, recived_data=13 , original_data = 13
Message id = 10, recived_data=14 , original_data = 14
Message id = 11, recived_data=15 , original_data = 15
Message id = 12, recived_data=16 , original_data = 16
Message id = 13, recived_data=17 , original_data = 17
Message id = 14, recived_data=18 , original_data = 18
Time stamp=72389564

Figure 67 - Message received

Step 3: Send a message with length longer than queue length

We have created this file in a C language that simply creates and sends the message to NIO on port 2. The message has a length of 32. In the configuration file "port.cfg" is defined a queue length equal to 16.

for(int i=0;i<mesg_size3;i++)
{
 Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+131072,message3+i);
}
Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>



Expected Results: In this case, having defined and sent a message with length 32, we expect to receive at the defined port the only first 16 rows and to have a queue overflow. In the figures below is possible to see that the message has arrived at the destination port, but the message content is corrupted after the 16 rows.

Copyright © FRACTAL Project Consortium	97 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Message id = 0, recived data=5 , original data = 5
Message id = 1, recived_data=6 , original_data = 6
Message id = 2, recived_data=7 , original_data = 7
Message id = 3, recived_data=8 , original_data = 8
Message id = 4, recived_data=9 , original_data = 9
Message id = 5, recived_data=10 , original_data = 10
Message id = 6, recived_data=11 , original_data = 11
Message id = 7, recived_data=12 , original_data = 12
Message id = 8, recived_data=13 , original_data = 13
Message id = 9, recived_data=14 , original_data = 14
Message id = 10, recived_data=15 , original_data = 15
Message id = 11, recived_data=16 , original_data = 16
Message id = 12, recived_data=17 , original_data = 17
Message id = 13, recived_data=18, original_data = 18
Message id = 14, recived data=19, original data = 19
Message id = 15, recived_data=20 , original_data = 20
Message id = 16, recived_data=20 , original_data = 21
Message id = 17, recived_data=20 , original_data = 22
Message id = 18, recived_data=20 , original_data = 23
Message id = 19, recived_data=20, original_data = 24
Message id = 20, recived data=20, original data = 25
Message id = 21, recived_data=20 , original_data = 26
Message id = 22, recived_data=20 , original_data = 27
Message id = 23, recived_data=20 , original_data = 28
Message id = 24, recived_data=20 , original_data = 29
Message id = 25, recived_data=20 , original_data = 30
Message id = 26, recived data=20, original data = 31
Message id = 27, recived_data=20 , original_data = 32
Message id = 28, recived_data=20 , original_data = 33
Message id = 29, recived_data=20 , original_data = 34
Message id = 30, recived_data=20 , original_data = 35
Message id = 31, recived_data=20 , original_data = 36
Message id = 32, recived_data=20 , original_data = 37
Message id = 33, recived_data=20 , original_data = 38
<pre>Message id = 34, recived_data=20 , original_data = 39</pre>
Message id = 35, recived_data=20 , original_data = 40
<pre>Message id = 36, recived_data=20 , original_data = 41</pre>
Message id = 37, recived_data=20 , original_data = 42
Message id = 38, recived_data=20 , original_data = 43
Message id = 39, recived_data=20 , original_data = 44
Message id = 40, recived_data=20 , original_data = 45
<pre>Message id = 41, recived_data=20 , original_data = 46</pre>
Message id = 42, recived_data=20 , original_data = 47
Message id = 43, recived_data=20 , original_data = 48
Message id = 44, recived_data=20 , original_data = 49
<pre>Message id = 45, recived_data=20 , original_data = 50</pre>
<pre>Message id = 46, recived_data=20 , original_data = 51</pre>
Message id = 47, recived_data=20 , original_data = 52
Message id = 48, recived_data=20 , original_data = 53
Message id = 49, recived_data=20 , original_data = 54
Time stamp=20
d. d.

Figure 69 - Messages received

Copyright © FRACTAL Project Consortium	98 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Steps	
Step 1	Send a message to NIO on port 1 with a length less than the queue length
	<pre>for(int i=0;i<mesg_size1;i++)< pre=""></mesg_size1;i++)<></pre>
	<pre>Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+131072,message1+i);</pre>
	<pre>} Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>
Step 2	<pre>Send a message to NIO on port 1 with a length equal to the queue length for (int i=0;i<mesg_size2;i++) 0="" <="" axi="" baseaddr+131072,message2+i);="" out32(xpar="" pre="" s00="" skinoc="" xil="" {=""></mesg_size2;i++)></pre>
	<pre>Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7); Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>
Step 3	Send a message to NIO on port 1 with a length bigger than the queue length
	<pre>for(int i=0;i<mesg_size3;i++)< pre=""></mesg_size3;i++)<></pre>
	<pre>Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+131072,message3+i);</pre>
	<pre>} Xil_Out32(XPAR_SKINOC_0_S00_AXI_BASEADDR+163840,7);</pre>

Table 84 - Steps for Validation Test T04_WP6T62-0X

8.7.2.5 T05_WP6T62-0X - Testing the Serialization service

Step 1: Define a set of messages to send (1 message at 1 port according to the scheduling configuration TTCommScheld.cfg of NI3)

- One message to NI3 on port 2

Port ID	Instant	Next	Instant (microsec)
2[NI3-NI2]	1502 (11)	0	44,759

Figure 70 - Information described in "ttcommsched.cfg" on NI3

In the "ttcommsched.cfg" on the NI3 we have defined the following rows:

The "hw.cfg" file is the same for all NIs, so it has the following rows:

_

Step 2: Send a message to NI3 on port 2.

We have created this file in a C language that simply creates and sends the message to NI3 on port 2.

Copyright © FRACTAL Project Consortium	99 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4
		<pre>/ send message from NI3 , port2 INIT_AXI_TXN=1; PORT_ID_WR =2; INPUT_WDATA=0; #48 INIT_AXI_TXN=0; r(integer j=0;j<16;j=j+1)//f begin #8;</pre>

Figure 71 - Send message

INPUT_WDATA =j;

end

Expected Results: In this case, we expect to see the serialized messages in the NI sender, after messages pass through the serialize component.

In this way, we check the message from the NI sender on port 2. We can see the message on the "buffer_inst" field, it is a queue with a 005 "msglen".

Name	Value
🕌 clk	1
🕌 reset_n	1
🕌 enq	0
> 👹 din[31:0]	0000001c
🏭 full	0
🕌 deq	0
> 🗑 dout[31:0]	000000
> 🗑 msglen[9:0]	005
🏭 empty	0
> 😻 buffer_inst[0:511][31:0]	00000000,00000001,00000002,00000003,00000004,UUUUUUUUUUUUUUUUUUUUU
> 😻 wrpntr[9:0]	005
> 😻 rdpntr[9:0]	000
👪 full_loc	0
🜡 empty_loc	0
> 😻 nqd[9:0]	005
🜡 AddrWidth	9
🜡 WordWidth	32
> V C_MAX_NUM_ELEMENTS[9:0]	200

Figure 72 - Message on NI sender

In order to see the message in a waveform we wrote it in memory. We can see the message before serialization and refer to the clock.

NI Sender		[
🕌 clk	0								
🕌 reset_n	1								
Message before Serialized									
🖬 message before ser	00000002		XXXXXXXXXX	00000000 0	000001 🗙 0000	0002	00000003	60006604	
Message after Serialized									

Figure 73 - Message before serialization

Copyright © FRACTAL Project Consortium	100 of 149
--	------------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Here we see the serialize packetized message and added all information needed to forward the message to the NoC (head filt that contains the message path, timestamp, destination port, etc.)



Figure 74 - Message after serialization

Steps						
Step 1	configuration TT	Define a set of messages to send (1 message at 1 port according to the scheduling configuration TTCommScheld.cfg of NI3) - One message to NI3 on port 2				
	Port ID	Instant	Next	Instant (microsec)		
	2[NI3-NI2]	1502 (11)	0	44,759		
Step 2	INIT_AX PORT_ID INPUT_W #48 INIT for(integ begin #8;	message from NI3 , p I_TXN=1; _WR =2; DATA=0; _AXI_TXN=0; er j=0;j<16;j=j+1)//				

Table 85 - Steps for Validation Test T05_WP6T62-0X

8.7.3 Test environment setup

The Hardware Edge Controller component is mainly configured through the following configuration and scheduling files:

- "hw.cfg", contains the overall information about the NoC;
- "port.cfg" contains the port configuration for each NI;
- "ttcommsched.mem", contains the time-triggered communication schedule.

The "hw.cfg" represents the configuration of the whole NoC and serves as the basis for the instantiation of the NIs and their internal building blocks. The file is composed as described in the following figure:

Copyright © FRACTAL Project Consortium	101 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Name	Range	Description	
NR_TILES	1 ("01") to 255 ("FF")	Number of tiles on the chip	
NR_PORTS	0 ("00") to 255 ("FF")	Number of ports at each NI	
TIMELY_BLOCK	0 or 1	0 if timely block is deactivated	
TIMELI_BLOOK	0 01 1	and 1 if activated	
PHASE_WIDTH	$1 ((01^{2})) + (0) ((01^{2}))$	Width of the phase slice in the	
PHASE_WIDTH	1 ("01") to 63 ("3F")	time format	
NR_PERIODS	1 ("01") to 63 ("3F")	Number of supported periods	
PERIOD_DELTA	$1 ((01^{2})) + (0) ((2E^{2}))$	Bit distance between period bits	
FERIOD_DELIA	1 ("01") to $63 ("3F")$	of two consecutive periods	
		Positional index of the bit of	
MSB_PERIODBIT	1 ("01") to 63 ("3F")	the greatest period in the global	
MOD_FERIODDI1		time base (greater than the	
		MACROTICK_BIT	
	"0000000000000000"	Hex representation of bit-wise	
PERIOD_ENABLE	"FFFFFFFFFFFFFFFFF" to	period enable; 1 for activation.	
	to 255 ("FF")	period enable; 1 for activation.	

Figure 75 - "hw.cfg" file

The "port.cfg" contains the configuration parameters of the ports for each NI. All configuration parameters are represented by hexadecimal digits, which are generated based on the parameters. The file is composed as described in the following figure:

Name	bits	location	Hex digits	Values	Description
Reserved	3 bits	127-125	1		
Enable	1 bit	124	1	Enable: "1" Disable: "0"	Shows whether the port is activated or not
Туре	2 bits	123-122		TT: "00" RC1: "01" RC2: "10" BE: "11"	Represents the type of the port
Direction	1 bit	121	1	OUT: "0" IN: "1"	Represents the direction of the port
Semantics	1 bit	120		State: "0" Event: "1"	Represents the semantics of the port
<mark>Buffer Size</mark>	12 bits	119-108	3	0 to 4095	The size of the buffer in words
<mark>Queue</mark> length	12 bits	<mark>107-96</mark>	3	<mark>0 to 4095</mark>	This fields is valid for event ports. in case of state ports, this field should be 1
Cluster ID	<mark>8 bits</mark>	<mark>95-88</mark>	<mark>2</mark>	<mark>0 to 255</mark>	
Node ID	8 bits	<mark>87-80</mark>	2	0 to 255	Represent the Physical address
Tile ID	<mark>8 bits</mark>	<mark>79-72</mark>	<mark>2</mark>	<mark>0 to 255</mark>	of destination
Port ID	<mark>8 bits</mark>	<mark>71-64</mark>	2	<mark>0 to 255</mark>	
MINT	64 bits	63-0	8	Global Time format	In case of an RC port, this field represent the value for MINT

Figure 76 - "hw.cfg" file

The "ttcommsched.cfg" is conceptually made up of a circular linked list. Each list is associated with a period. The information included in this file is shown below:

- "next", this field is the next pointer. It points to the next entry of the circular linked list, and it has a width of 5 bits.
- "instant", this field denotes the phase of the injection of the message which is located at the given port (in PortId). It has a width of 8 bits.
- "PortId", this field represents the ID of the port, whose message is injected into the NoC at the time given by Instant. It has a width of 4 bits.

Copyright © FRACTAL Project Consortium	102 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

In particular, in those configuration files we have defined:

- period of the clock (122us)
- Clock granularity (28,483ns)
- Injection time for messages (Instant)

We also installed ATTNoC on the FPGA board from Xilinx. The configuration has been applied with the Vitis Unified Development Environment. The simulation and visualization of the off-chip network was driven with the Vivado Design Suite.

All the configuration and information of the Hardware Edge Control at the GitHub Repository, in the following link https://github.com/project-fractal/WP6T62-HW-Edge-Controller.

Copyright © FRACTAL Project Consortium	103 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7.4 Test execution

8.7.4.1 T01_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at different port

Results/Evidence	
Step 2:	
Release 2021.1 Feb 7 2023 - 07:42:21 PMU-FW is not running, certain applications may not be supported. NIO - NIZ , recived_data=0 , original_data = 0 NIO - NIZ , recived_data=1 , original_data = 1 NIO - NIZ , recived_data=2 , original_data = 2 NIO - NIZ , recived_data=3 , original_data = 3 NIO - NIZ , recived_data=3 , original_data = 3 NIO - NIZ , recived_data=4 , original_data = 5 NIO - NIZ , recived_data=7 , original_data = 5 NIO - NIZ , recived_data=9 , original_data = 9 NIO - NIZ , recived_data=1 , original_data = 10 NIO - NIZ , recived_data=1 , original_data = 10 NIO - NIZ , recived_data=1 , original_data = 10 NIO - NIZ , recived_data=1 , original_data = 13 NIO - NIZ , recived_data=2 , original_data = 13 NIO - NIZ , recived_data=2 , original_data = 13 NIO - NIZ , recived_data=0 , original_data = 13 NIO - NIZ , recived_data=0 , original_data = 1 NII - NIO , recived_data=2 , original_data = 1 NII - NIO , recived_data=0 , original_data = 1 NII - NIO , recived_data=2 , original_data = 3 NII - NIO , recived_data=3 , original_data = 5 NII - NIO , recived_data=3 , original_data = 5 NII - NIO , recived_data=3 , original_data = 10 NII - NIO , recived_data=1 , original_data = 13 NII - NIO , recived_data=1 , original_data = 10 NII - NIO , recived_data=1 , original_data = 13 NII - NIO , recived_data=1 , original_data = 14 	Time difference between m2, m1 =4 micro second Pass Time difference between m3, m2 =44 micro second Pass Time difference between m3, m1 =48 micro second Pass Injection time of m1 =1122987 micro second Injection time of m2 =1122991 micro second Injection time of m3 =1123036 micro second
Success criteria	
Check: - destination port as per configuration "port.cfg" - message content (not corrupted) - period as per configuration "hw.cfg" - Is it possible to check the "instant" (Time Unit)?	
Test observations	
Test configuration Config: "ttcommsched.cfg", "po	ort.cfg", "hw.cfg"
C	TNoC running on the board, all configuration files at project-fractal/WP6T62-HW-Edge-Controller
Remarks	· · · · · · ·
Test result	
Passed	

Table 86 - Results of the test T01_WP6T62-0X

Copyright © FRACTAL Project Consortium	104 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7.4.2 T02_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at the same port

Results/Evide	nce
NII -	<pre>NI3 ; rectved_data=2 ; original_data = 2 NI3 ; rectved_data=4 ; original_data = 3 NI3 ; rectved_data=5 ; original_data = 5 NI3 ; rectved_data=6 ; original_data = 6 NI3 ; rectved_data=9 ; original_data = 7 NI3 ; rectved_data=9 ; original_data = 10 NI3 ; rectved_data=11 ; original_data = 11 NI3 ; rectved_data=12 ; original_data = 11 NI3 ; rectved_data=12 ; original_data = 12 NI3 ; rectved_data=14 ; original_data = 13 NI3 ; rectved_data=14 ; original_data = 14 NI3 ; rectved_data=1 ; original_data = 6 NI3 ; rectved_data=2 ; original_data = 14 NI3 ; rectved_data=2 ; original_data = 6 NI3 ; rectved_data=2 ; original_data = 14 NI3 ; rectved_data=2 ; original_data = 6 NI3 ; rectved_data=2 ; original_data = 6 NI3 ; rectved_data=3 ; original_data = 6 NI3 ; rectved_data=4 ; original_data = 6 NI3 ; rectved_data=5 ; original_data = 7 NI3 ; rectved_data=5 ; original_data = 8</pre>
Success criter	a
Check:	port (and NI) as "port.cfg"
	content (not corrupted)
-	xpect three different period)
Test observat	
Test configura	
Test condition	
	github link https://github.com/project-fractal/WP6T62-HW-Edge-Controller
Remarks	
Test result	

Table 87 - Results of the test T02_WP6T62-0X

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7.4.3 T03_WP6T62-0X - Testing the Message-Classification and Message-Scheduling Services at the same NI

Results/Evidence	·
Step 2: NII - NI0 , rect NII - NI3 , rect NII - NI0 , rect NII - NI0 , rect <	<pre>ved data=0 , ortginal_data = 0 ved data=1 , ortginal_data = 1 ved data=2 , ortginal_data = 1 ved data=3 , ortginal_data = 3 ved data=5 , ortginal_data = 4 ved data=5 , ortginal_data = 6 ved data=6 , ortginal_data = 10 ved data=9 , ortginal_data = 10 ved data=10 , ortginal_data = 10 ved data=10 , ortginal_data = 11 ved data=10 , ortginal_data = 11 ved data=10 , ortginal_data = 12 ved data=2 , ortginal_data = 14</pre>
Success criteria	
Check:	
- destination port (and	
- the message content (
- period ?? (Two differe	nt periods)
Test observations	
Test configuration	Config: "ttcommsched.cfg", "port.cfg", "hw.cfg"
Test conditions	Vitis for FPGA configuration, ATTNoC running on the board, all configuration files at
	github link https://github.com/project-fractal/WP6T62-HW-Edge-Controller
Remarks	
Test result	
Passed	

Table 88 - Results of the test T03_WP6T62-0X

Copyright © FRACTAL Project Consortium	106 of 149
	100 01 1.17

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

8.7.4.4 T04_WP6T62-0X - Testing the Ingress and Egress-Queuing Services

Step 1:		Step 3:	
Message id = 0, recti Message id = 1, reci Message id = 1, reci Message id = 2, reci Message id = 3, reci Message id = 4, rect Message id = 0, rect Message id = 0, reci Message id = 0, reci Time stamp=37196732 Step 2: Message id = 0, reci Message id = 1, reci	<pre>ved_data=5 , original_data = 5 ved_data=6 , original_data = 6 ved_data=8 , original_data = 7 ved_data=9 , original_data = 10 ved_data=11 , original_data = 11 ved_data=12 , original_data = 11 ved_data=12 , original_data = 12 </pre>	Message id = 0, recived_data=5, original_data = 5 Message id = 2, recived_data=6, original_data = 6 Message id = 3, recived_data=7, original_data = 7 Message id = 3, recived_data=9, original_data = 9 Message id = 5, recived_data=10, original_data = 10 Message id = 5, recived_data=11, original_data = 11 Message id = 7, recived_data=12, original_data = 11 Message id = 7, recived_data=12, original_data = 13 Message id = 9, recived_data=13, original_data = 13 Message id = 9, recived_data=13, original_data = 13 Message id = 10, recived_data=14, original_data = 15 Message id = 11, recived_data=17, original_data = 16 Message id = 11, recived_data=16, original_data = 16 Message id = 12, recived_data=14, original_data = 18 Message id = 13, recived_data=14, original_data = 18 Message id = 14, recived_data=10, original_data = 18 Message id = 14, recived_data=20, original_data = 23 Message id = 16, recived_data=20, original_data = 23 Message id = 16, recived_data=20, original_data = 23 Message id = 19, recived_data=20, original_data = 23 Message id = 19, recived_data=20, original_data = 23 Message id = 20, recived_data=20, original_data = 23 Message id = 21, recived_data=20, original_data = 23 Message id = 22, recived_data=20, original_data = 23 Message id = 23, recived_data=20, original_data = 23 Message id = 24, recived_data=20, original_data = 33 Message id = 26, recived_data=20, original_data = 33 Message id = 33, recived_data=20, original_data = 33 Message id = 33, recived_data=20, original_data = 34 Message id = 33, recived_data=20, original_data = 33 Message id = 33, recived_data=20, original_data = 34 Message id = 33, recived_data=20, original_data = 34 Message id = 34, recived_data=20, original_data = 44 Message id = 37, recived_data=20, original_data = 44 Message id = 37, recived_data=20, original_data = 43 Message id = 37, recived_data=20, original_data =	
Success criteria			
Check: - to receive all messag - all messages on senc			
Test observations			
Test configuration	Config: "ttcommsched.cfg", '	'port.cfg", "hw.cfg"	
Test conditions	Vitis for FPGA configuration, ATTNoC running on the board, all configuration files at github link https://github.com/project-fractal/WP6T62-HW-Edge-Controller, max		
	queue length for a port = 16		
Remarks	queue length for a port = 16		

Table 89 - Results of the test T04_WP6T62-0X

8.7.4.5 T05_WP6T62-0X - Testing the Serialization service

In this case, as introduced in test planification session we defined to perform a simulation. We used Vivado to see the waveform in subcomponents referred to clock signal, so we can observe the behaviour with a time reference.

The scope is to compare the message before and after the serializer in order to validate the serializer subcomponent that is in charge of the serialization service as described in D6.2.

Copyright © FRACTAL Project Consortium	107 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Serializer subcomponent is very important in order to have at the end, a complete message packet to send through the network, from on-chip to off-chip network and vice versa.

Results/Evidence					
Step 2:					
Name		Value			
🕌 clk		1			
₩ reset_n		1			
🕌 enq		0			
> 🗑 din[31:0]		0000001c			
li full		0			
🕌 deq		0			
> 📽 dout[31:0]		000000			
> 📽 msglen[9:0]		005			
≟ empty		0			
> 😻 buffer_inst[0:	511][31:0]	0000000,0000001,0000002,0000003,0000004,UUUUUUUUUUUUUUUUUUUUUUU			
> 😻 wrpntr[9:0]		005			
> 😻 rdpntr[9:0]		000			
14 full_loc		0			
18 empty_loc		0 005			
> 😻 nqd[9:0]		9			
•		32			
↓ WordWidth > W C_MAX_NUM_E		200			
Message after Serialized Imessage before ser Message after Serialized	00000002 00000002 00000002	XXXXXXXX C0000000 C00000002 C0000003 C0000004 XXXXXXXX C0000000 C00000002 C0000003 C0000004 XXXXXXXX C00000003 C00000003 C00000003 C00000004 XXXXXXXX C00000003 C00000003 C00000003 C00000004			
Success criteria					
Check:					
	NU	and the ment D			
 the message from the 					
 the serialized message 	e (flits in o	rdered)			
Test observations					
Test configuration	Config: "ttcommsched.cfg", "port.cfg", "hw.cfg"				
Test conditions Vitis for FPGA configuration, ATTNoC running on the board, all configur					
		ik https://github.com/project-fractal/WP6T62-HW-Edge-Controller			
Remarks	0				
Test result					
Passed					
rasseu					

Table 90 - Results of the test T05_WP6T62-0X

Copyright © FRACTAL Project Consortium	108 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

9 Conclusions

In conclusion, this deliverable is focused on identifying the functionalities of the FRACTAL Edge Node Architecture implemented in task T6.1 and task T6.2, and on defining, implementing, and documenting the results of the validation tests that have been performed to validate the FRACTAL Edge Node proper operation at a component level.

Validation was carried out starting with the microservices regarding to connectivity, then the orchestration components, the runtime manager component, the Data ingestion component and Federated data collection component for the High-end nodes and Mid-range nodes. For the Low End Node there were a series of tests to validate the following functionalities:

- connection and communication with the Cloud Platform
- execution of task scheduling
- management of ingestion and storage

Finally, the Hardware-level Edge Controller was also validated, focused on the Network Gateway Interface that connects both on chip and off chip networks.

Some of the tests were performed on real HW, that is to say, on the HW platforms that are part of the project, and others in a virtual environment. In the case of some components, the functionalities could be tested on any HW/virtual environment, since by design they were valid for both High-end nodes and Mid-range nodes. In the case of the Low End Node, it was crucial to test the functionalities on the real HW.

Almost all tests passed validation. Except for one: T05_WP6T62-06_ANC. This failed test was communicated to the developers in order to improve the quality of the component.

Copyright © FRACTAL Project Consortium	109 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

10 Bibliography

- Richardson, C. (2021). Microservices Pattern: Microservice Architecture pattern.Retrieved12April2022,fromhttps://microservices.io/patterns/microservices.html
- Vaid, A., Maria, M., & Udupa, N. (2020). A Framework-driven Approach for Verification and Validation (V&V) of IoT Systems. Retrieved 5 April 2022, from https://www.wipro.com/content/dam/nexus/en/service-lines/productengineering/latest-thinking/a-framework-driven-approach-for-verificationand-validation-v-and-v-of-iot-systems.pdf

Copyright © FRACTAL Project Consortium	110 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

11 List of figures

Figure 1: Validation test template
Figure 2 FRACTAL Edge Node processing architecture designed in task T6.111
Figure 3 FRACTAL Edge Node processing architecture implementation developed in
task T6.111
Figure 4 FRACTAL Edge Node testbed architecture
Figure 5 - MQTT data producer code13
Figure 6 - Bridge MQTT-KAFKA code14
Figure 7 - Bridge SQL-KAFKA code16
Figure 8 - Data consumer code17
Figure 9 - Automatic launching of MQTT, KAFKA and SQL instances
Figure 10 - Run common services
Figure 11 - Run data producer19
Figure 12 - Run MQTT-KAFKA bridge19
Figure 13 - Run SQL-KAFKA bridge20
Figure 14 - Run data consumer20
Figure 15 - Testbed data analysis21
Figure 16: Multi-node Edge Controller architectural design (designed in task T6.2)
Figure 17: WP6T62-06-mid-range-orchestration architecture (designed in task T6.2)
Figure 18 - Node interconnection
Figure 19 - Runtime Manager Flow 146
Figure 20 - Runtime Manager Flow 247
Figure 21 - Runtime Manager Flow 347
Figure 22 - Runtime Manager Flow 148
Figure 23- Runtime Manager Flow 249
Figure 24 - Runtime Manager Flow 349
Figure 15 - Run "rm_api.py" script52
Figure 16 - Run "rm_mqtt.py" script52
Figure 17 - The list of the provisioned devices as CRD in Kubernetes cluster with their
ids71
Figure 18 - The list of devices in IoT hub71
Figure 19 - The description of the connected to the Kubernetes device
Figure 20 - The connected device with its serial number: 806372
Figure 21 - Device connected with the green LED73
Figure 22 - Device connected with the green LED73
Figure 23 - The desired state is OFF before pressing the button74
Figure 24 - The device before pressing button74
Figure 25 - The status of the device is changed in IoT hub after pressing button74
Figure 26 - The device LED is green after pressing button74
Figure 27 - Device connected with the green LED
Figure 28 - Running the patch command using K8s CLI (Updating the time is ignored
here)76

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Figure 29 - Desired state of the device is changed to 1 in Kubernetes CRD (ON)76
Figure 30 - Kubernetes log showing the invocation of update function
Figure 31 - The state in IoT hub has been switched to green
Figure 32 - The device before the patch command on the left, and after the patch
command on the right
Figure 33 - dmesg command
Figure 34 - Command to open the terminal of device
Figure 35 - ps command
Figure 36 - Validation Test T02_WP6T62-0X84
Figure 37 - Validation Test T02_WP6T62-0X
Figure 38 - Validation Test T03_WP6T62-0X
Figure 39 - Validation Test T04_WP6T62-0X
Figure 40 - Validation Test T05_WP6T62-0X
Figure 41 - Information described in "ttcommsched.cfg" on NIO
Figure 42 - Information described in "ttcommsched.cfg" on NI1
Figure 43 - Send messages scripts
Figure 44 - Messages received
Figure 45 - Time different and Injection time
Figure 46 - Information described in "ttcommsched.cfg" on NI1
Figure 47 - Send messages scripts90
Figure 48 - Messages received91
Figure 49 - Time different between messages91
Figure 50 - Information described in "ttcommsched.cfg" on NI292
Figure 51 - Send messages scripts93
Figure 52 - Messages received94
Figure 53 - Time different between messages94
Figure 54 - Send message96
Figure 55 - Message received96
Figure 56 - Send message96
Figure 57 - Message received97
Figure 58 - Send message97
Figure 59 - Messages received98
Figure 60 - Information described in "ttcommsched.cfg" on NI3
Figure 61 - Send message100
Figure 62 - Message on NI sender100
Figure 63 - Message before serialization100
Figure 64 - Message after serialization101
Figure 65 – "hw.cfg" file
Figure 66 – "hw.cfg" file

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

12 List of tables

Table 1 Darward bistory	
Table 1 Document history 2 Table 2 Table 2	+
Table 2 – Testing node specifications 17 Table 2 – Validation Test To1, WDCTC2, 00, 50 20	
Table 3 - Validation Test T01_WP6T62-06_EC 26 Table 4 - Validation Test T02_WP6T62-06_EC 26	
Table 4 - Validation Test T02_WP6T62-06_EC 26 Table 5 - Validation Test T02_WP6T62-06_EC 26	
Table 5 - Validation Test T03_WP6T62-06_EC 26 Table 6 - Validation Test T03_WP6T62-06_EC 26	
Table 6 - Validation Test T04_WP6T62-06_EC 26	
Table 7 - Steps for Validation Test T01_WP6T62-06_EC	
Table 8 - Steps for Validation Test T02_WP6T62-06_EC	
Table 9 - Steps for Validation Test T03_WP6T62-06_EC	
Table 10 - Steps for Validation Test T04_WP6T62-06_EC28	
Table 11 - Results of the test T01_WP6T62-06_EC 29	
Table 12 - Results of the test T02_WP6T62-06_EC 30	
Table 13 - Results of the test T03_WP6T62-06_EC 31	
Table 14 - Results of the test T04_WP6T62-06_EC 32	
Table 15 - Validation Test T01_WP6T62-06_ANC 34	
Table 16 - Validation Test T02_WP6T62-06_ANC 34	
Table 17 - Validation Test T03_WP6T62-06_ANC 34	
Table 18 - Validation Test T04_WP6T62-06_ANC 34	
Table 19 - Validation Test T05_WP6T62-06_ANC 35	
Table 20 - Validation Test T06_WP6T62-06_ANC35	
Table 21 - Steps for Validation Test T01_WP6T62-06_ANC 35	
Table 22 - Steps for Validation Test T02_WP6T62-06_ANC 35	
Table 23 - Steps for Validation Test T03_WP6T62-06_ANC 36	
Table 24 - Steps for Validation Test T04_WP6T62-06_ANC 36	
Table 25 - Steps for Validation Test T05_WP6T62-06_ANC 36	
Table 26 - Steps for Validation Test T06_WP6T62-06_ANC 36	
Table 27 - Results of the test T01_WP6T62-06_ANC 37	
Table 28 - Results of the test T02_WP6T62-06_ANC 38	
Table 29 - Results of the test T03_WP6T62-06_ANC	
Table 30 - Results of the test T04_WP6T62-06_ANC 39	
Table 31 - Results of the test T05_WP6T62-06_ANC40	
Table 32 - Results of the test T06_WP6T62-06_ANC41	1
Table 33 - Validation Test T01_WP6T62-0342	
Table 34 - Validation Test T02_WP6T62-0342	
Table 35 - Validation Test T03_WP6T62-03 43	
Table 36 - Validation Test T04_WP6T62-0343	
Table 37 - Validation Test T05_WP6T62-03 43	
Table 38 - Steps for Validation Test T01_WP6T62-03 45	
Table 39 – Steps for Validation Test T02_WP6T62-0345	
Table 40 – Steps for Validation Test T03_WP6T62-0346	
Table 41 – Steps for Validation Test T04_WP6T62-0347	
Table 42 - Steps for Validation Test T05_WP6T62-03 50	
Table 43 - Results of the test T01_WP6T62-03 53	3

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

2431242	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Table 90 - Results of the test T05_WP6T62-0X108
Table 91 - Validation Test T01_WP6T62-06_EC (overview)
Table 92 - Validation Test T02_WP6T62-06_EC (overview)
Table 93 - Validation Test T03_WP6T62-06_EC (overview)
Table 94 - Validation Test T04_WP6T62-06_EC (overview) 121
Table 95 - Validation Test T01_WP6T62-06_ANC (overview) 122
Table 96 - Validation Test T02_WP6T62-06_ANC (overview) 123
Table 97 - Validation Test T03_WP6T62-06_ANC (overview) 124
Table 98 - Validation Test T04_WP6T62-06_ANC (overview) 125
Table 99 - Validation Test T05_WP6T62-06_ANC (overview) 127
Table 100 - Validation Test T06_WP6T62-06_ANC (overview)128
Table 101 - Validation Test T01_WP6T62-03 (overview)
Table 102 - Validation Test T02_WP6T62-03 (overview)
Table 103 - Validation Test T03_WP6T62-03 (overview)
Table 104 - Validation Test T04_WP6T62-03 (overview) 132
Table 105 - Validation Test T05_WP6T62-03 (overview)133
Table 106: Validation Test T01_WP6T62-01_DI (overview)
Table 107: Validation Test T02_WP6T62-01_DI (overview)135
Table 108: Validation Test T03 WP6T62-01_DI (overview)
Table 109: Validation Test T04_WP6T62-01_DI (overview)137
Table 110: Validation Test T05_WP6T62-01_DI (overview)138
Table 111: Validation Test T01_WP6T62-02_FDC (overview)
Table 112: Validation Test T02_WP6T62-02_FDC (overview)140
Table 108 - Validation Test T01_WP6T62-06 (overview) 141
Table 109 - Validation Test T02_WP6T62-06 (overview) 142
Table 110 - Validation Test T03_WP6T62-06 (overview) 143
Table 111 - Validation Test T04_WP6T62-06 (overview) 144
Table 112 - Validation Test T01_WP6T62-0X (overview)
Table 113 - Validation Test T02_WP6T62-0X (overview) 146
Table 114 - Validation Test T03_WP6T62-0X (overview)147
Table 115 - Validation Test T04_WP6T62-0X (overview)148
Table 116 - Validation Test T05_WP6T62-0X (overview) 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

13 List of abbreviations

API ATTNoC CPU CRD DB FPGA HW ID IoT LED MQTT NGW NI NGW NI NOC PC PW RAM REST SQL SSID USB	Application Programming Interface Adaptable Time Triggered Network on Chip Central Processing Unit Custom Resource Definition Database Field Programmable Gate Array Hardware Identifier Internet of Things Light Emitting Diode Message Queue Telemetry Transport Network Gateway Network Interface Network on Chip Personal Computer Password Random Access Memory Representational State Transfer Structured Query Language Service Set Identifier Universal Serial Bus
USB	Universal Serial Bus

Copyright © FRACTAL Project	Consortium 116 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14 Annexes

14.1 Orchestration (Edge Controller) component complete templates

Test ID	Validation test
Test ID	T01_WP6T62-06_EC
Test type	Functional-Installation
Test name	Testing the installation of the component
Date	15/11/2022
Tester's Name	Ana Bautista
Test scope or object	ive
The objective of this	s test is to validate if the Edge Controller Orchestrator can be installed without any issues.
Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.
Results/Evidence	
	built d37fc3f57648 tagged custom-orchestrator:latest
	All partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
Test comiguration	The metrics exporter, resource manager and custom orchestrator containers runs in the same node
Test conditions	(master node). Two bugs were found during installation that have been reported and corrected by the
	developers. 1. apt-get update no longer works on containers with Ubuntu21.10 so we need to use Ubuntu22.04 * WFFFE-06-edge-cmtroller-erchaetrator git: (WFFE6-06-edge-orchestrator) docker build -t metrics-exporter metrics-exporter Ending build context to Docker deepen 23.048 (************************************
Remarks	<pre></pre>
Remarks	<pre>Ign:1 http://active.uburtu.com/uburtu impish Finelease Ign:2 http://active.uburtu.com/uburtu impish Finelease Ign:3 http://active.uburtu.com/uburtu impish-security Finelease Ign:3 http://active.uburtu.com/uburtu Ignish-security Finelease Ign:3 http://actity=Kinelease Ign:3 http://active.uburtu.com/uburtu Ignish-</pre>
Remarks Test result	<pre>[]pr:1 http://schive.ubunte.com/ubunts implish-Sciences []pr:2 http://schive.ubunte.com/ubunts implish-Sciences []pr:2 http://schive.ubunts.com/ubunts implish-Sciences []pr:3 http://schive.ubunts.com/ubunts.c</pre>

Table 91 - Validation Test T01_WP6T62-06_EC (overview)

Copyright © FRACTAL Project Consortium 117 of 14	19
--	----

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Test Up TO2_WFGT2-OG_EC Test stage Functional Test mane Test string if the master node can monitor several [2] workers' nodes: Date 20/01/2023 Test strings of behaviors Ana Bautista Test strings of behaviors Ana Bautista Test strings of behaviors Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node: Step 1 Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node: Step 2 Deploy the metrics exporter container. Step 3 Deploy the metrics exporter container. Step 4 Deploy the metrics exporter container or each of the worker nodes. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container or each of the worker nodes. Results/video Reside controller or ordestration of classification or each of the worker nodes. Results/string of controller or ordestration of classification or each of the worker nodes. Results/string of the master node in the second or each of the worker nodes. Results/string of controller or ordestration of classification or each of the worker nodes. Results/string of the master node in the second or each of the		Validation test
Test spee Functional Enst spee 20/01/2023 Test speed 20/01/2023 Test speed 20/01/2023 Test speed 20/01/2023 Test speed 20/01/2023 Test speed speed 20/01/2023 Test speed speed 20/01/2023 Test speed speed 20/01/2023 Test speed speed Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node. Step 3 Deploy the resource numager container. Step 4 Deploy the resource numager container. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's node. Step 7 Review the logs from the resource numager (deployed on the master node). Results/fidence Master node: Step 5 Interview the logs from the resource numager (deployed on the master node). Step 5 Interview the logs from the resource numager (deployed on the master node). Step 5 Interview the logs from the resource numager (deployed on the master node). Step 5 <td< th=""><th>Test ID</th><th></th></td<>	Test ID	
Test mane Testing if the master node can monitor several (2) workers' nodes. Date 20/01/2021 Test scope ar objective Ana Bautista Test scope ar objective Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node. Step 1 Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node. Step 2 Deploy the retrics exporter container. Step 3 Deploy the metrics exporter container. Step 4 Deploy the metrics exporter container. Step 5 Prepare to modes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 5 Prepare to mode swith Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 5 Prepare to mode swith Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 5 Deploy the metrics exporter container. Step 5 Step 6 Deploy the metrics exporter container. Step 6 Step 6 Deploy the metrics exporter container. Step 6 Step 6 Deploy the metrics exporter container. Step 6 Step 6 Deploy the metrics exporter container. Step 6 Step 6		
Date 20/01/2023 Test scope or abjective freat scope or abjective of this test is to validate if the master node. Step 5 Prepare to validate if the master node and incret. Step 5 Step 4 Deploy the metrics exporter container. Step 5 Prepare to validate if the worker nodes. Step 7 Step 5 Prepare to validate if from the resource manager (deployed on the master node). Step 7 Review the log from the resource manager (deployed on the master node). Step 7 Review the log from the resource manager (deployed on the master node). Step 7 Review the log from the resource manager (deployed on the master node). Step 7 Review the log from the resource manager (deployed on the master node). Step 7 Review the log from the resource manager (deployed on the master node). Status protocol is status		
Tet scope or objective "Interpreter and with Uburtu and all needed dependencies (Python). This will be the master node. Step 1 Prepare and with Uburtu and all needed dependencies (Python). This will be the master node. Step 2 Deploy the metrics exporter container. Step 3 Deploy the custom constanter container. Step 5 Prepare tow nodes with Uburtu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager container. Step 7 Review the logs from the resource manager container on each of the worker nodes. Review the logs from the resource manager container on each of the worker nodes. Review logs from the resource manager container on each of the worker nodes. Review logs from the resource manager container of the worker nodes in the master node. Review logs from the resource manager container of the resource node in the master node in the resource manager of the resource manager container of the worker ps Review logs from the resource manager in the resource manager of the resource node in the resource manager in the resource node in the resource manager in the resource manager in the resource ma	Date	
The objective of this test is to validate if the master node can monitor two worker? Indes. Step 1 Prepare anode with Ubuntu and all needed dependencies (Python). This will be the master node. Step 2 Deploy the metrics exporter container. Step 3 Deploy the metrics exporter container. Step 4 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker modes. Step 5 Deploy the metrics exporter container on each of the worker step 5 Deploy the metrics exporter container. Step 5 Deploy the metrics export container on each of the worker step 5 Deploy the metric export	Tester's Name	Ana Bautista
Steps Control Step 1 Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node. Step 2 Deploy the resource manager container. Step 3 Deploy the resource manager container. Step 4 Deploy the resource manager container. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 7 Beview the logs from the resource manager (deployed on the master node). Results/Evidence Master node: Waffe? de node controller-orchestrator will (WMAD to dev vacable) of 8 docker ps Step 5 CONTAINE 10 FMAGE CONNA CRAFTD STATUS Step 7 Review the logs from the resource manager (deployed on the master node). Edit Status Step 7 Review the logs from the resource manager (deployed on the master node). CONTAINE 10 FMAGE CONTAINE 10 Status	Test scope or objectiv	e
Step 1 Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node. Step 2 Deploy the metrics exporter container. Step 3 Deploy the custom orchestrator container. Step 4 Deploy the metrics exporter container. Step 5 Prepare to node with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 5 Deploy the metrics exporter container. Step 7 Review the logs from the resource manager (deployed on the master node). Results/Vide COMMO Matter node: COMMO Step 5 COMMO COMMO COMMO Step 6 Deploy the metrics exporter container. Step 7 Review the logs from the resource manager (deployed on the master node). Step16 COMMO Step17 Review the logs from the resource manager. Step17 Review the logs from the resource manager. Step17 Review the logs from the resource manager. Step17 Review togs from the resource manager. Step17 Review togs from the resource manager. Step17 Review togs from the resource manager. <	The objective of this t	est is to validate if the master node can monitor two workers' nodes.
Step 2 Deploy the metrics exporter container. Step 3 Deploy the metrics exporter container. Step 4 Deploy the custom orchestrator container. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Results/Kuldence Master node: Waster node: CGRATED STATUS PORTS Waster node 1: UNFORCE decide: controller: orchestrator if (CGRADED 2000) STATUS PORTS Worker node 2: UNFORCE decide: controller: orchestrator if (CGRADED 2000) STATUS PORTS NAMES Status 10: TAMES CGRATED 2000 STATUS PORTS NAMES Status 10: TAMES CGRATED 2000	Steps	1
Step 3 Deploy the resource manager container. Step 4 Deploy the custom orchestrator container. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review tog the set of the set o	Step 1	
Step 4 Deploy the custom orchestrator container. Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Results/Fvidence Master node: Visito2-de-edge-controller-orchestrator it: it: if: if: if: if: if: if: if: if: if: if	Step 2	
Step 5 Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Results/Evidence Market node: Market node: Common container on each of the worker nodes. Variable node: Common container on each of the worker nodes. Market node: Common container on each of the worker ps Variable node: Common container on each of the worker ps Variable node: Common container on each of the worker ps Variable node: Common container on each of the worker ps Variable node: Common container on each of the worker ps Variable node: Common container on each of the worker ps Variable node: Variable node node node node node node node nod		
Step 5 Dodes. Step 6 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review the logs from the resource manager (deployed on the master node). Review logs from the resorce manager.	Step 4	
Step 5 Deploy the metrics exporter container on each of the worker nodes. Step 7 Review the logs from the resource manager (deployed on the master node). Results/Evidence Review the logs from the resource manager (deployed on the master node). Results/Evidence COMMAN COMMAN Commentation Image of the master node. Image of the master node: Command Command Deploy the metrics exponence Points Image of the master node: Command Command Image of the master node. Review the logs from the resource manager (deployed on the master node). Image of the master node: Command Command Image of the master node. Review the logs from the resource manager (deployed on the master node). Image of the master node in the normal in the master node in the master	Step 5	
Step 7 Review the logs from the resource manager (deployed on the master node). Results/Evidence Master node: ImigES: 46 edge controller-orchestration #1 (deployed on the master node). Status / Deployed Status / Deploy	Stop 6	
Results/Evidence Control Master node: Status Status Ubidits/Bodde_controller-orchestrator "/bin/ch - c "pritons." 20 hours go Up 20 hours go		
Mester node: INFECT 2006-odge: controller-orchestrator::::::::::::::::::::::::::::::::::		
UMPGE2-06-edge-controller-orchestrator CHARLE CREATED STATUS PORTS NAMES CONTAINER 10 TMAGE CONTAINER 10 TMAGE CONTAINER 10 PORTS NAMES CONTAINER 10 TMAGE CONTAINER 10 Up 20 hours 3 Up 20 hours 3 Len_reistat CONTAINER 10 CONTAINER 10 CONTAINER 10 Up 20 hours 3 Status Status Status Status Status PORTS NAMES CONTAINER 10 TMAGE CONTAINER 10 CONTAINER 10 CONTAINER 10 CONTAINER 10 PORTS NAMES VMORE-GO-edge-controller-orchestrator T/bin/sh - c 'glances' 22 hours ago Up 22 hours PORTS NAMES VMORE-GO-edge-controller-orchestrator T/bin/sh - c 'glances'' 22 hours ago Up 22 hours PORTS NAMES VMORE-GO-edge-controller-orchestrator T/bin/sh - c 'glances'' 22 hours ago Up 22 hours PORTS NAMES VMORE-GO-edge-controller-orchestrator T/bin/sh - c 'glances'' 22 hours ago Up 22 hours PORTS NAMES		
CONTAINER 10 IMAGE CONMAND CREATED STATUS PORTS MARES 27461080600 metrics-exporter "/bin/sh - c 'pin/sh - c '		
WP6162-06-edge-controller-orchestrator Image: ContAlker 10 PARTS NAMES CONTAINER 10 TAAGE COMMAND CREATED STATUS PORTS NAMES Dis20ebb949 metrics-exporter "/bin/sh - c'glances." Usersage Dis20ebb949 Dis20eb949 Dis20eb949 Dis20eb949 Dis20eb949 Dis20eb949<	CONTAINER ID IMAGE d110afe4abf6 resour 27a41d81626b metric	COMMAND CREATED STATUS PORTS NAMES rce-manager "/bin/sh -c 'python3" 20 hours ago Up 20 hours zen_feistel cs-exporter "/bin/sh -c 'glances" 20 hours ago Up 20 hours quirky_curran
CONTAINER ID IVAGE CONVAND CREATED STATUS PORTS NAMES Did20b0937 metrics-exporter "/bin/sh -c'glances" 22 hours ago Up 22 hours peaceful gauss Worker node 2: IVATUS PORTS NAMES peaceful gauss CONTAINER ID IVAGE COMMAND CREATED STATUS PORTS NAMES Professor Marcial Status Ports NAMES Ports NAMES Professor Ports Ports NAMES Ports NAMES Professor Ports Ports NAMES Ports NAMES Ports Ports Ports	Worker node 1:	
Worker node 2: UP6162-06-edge-controller-orchestrator (UP6162-06-edge-controller-orchestrator STATUS PORTS NAMES P0700270000 CREATED STATUS PORTS NAMES P07002700000 CREATED STATUS PORTS NAMES P07002700000000000000000000000000000000	CONTAINER ID IMAGE	COMMAND CREATED STATUS PORTS NAMES
CONTAINER ID THATUS PORTS NAMES 4570c097cacc metrics-exporter "/bin/sh -c 'glances" 22 hours ago Up 22 hours pedantic_antonelli Review logs from the resorce manager: Review logs from the resorce manager: Note that the shift of the shift o	Worker node 2:	
Review logs from the resorce manager:	CONTAINER ID IMAGE	COMMAND CREATED STATUS PORTS NAMES
solid of a labor information of the solid of increments of a labor information of a labor information of the solid of increments of a labor information of the solid of informatio of the solid of information of the solid of		
ENDERGY Provided HTTP: Fainted modes: () Success criteria No error messages/All partial results are as expected Test observations Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result	2023-02.16 2023-02.16	<pre>14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k880.jpd.ikerjan.es Total memory: 15.783896576 Gb 14:56:12,187 - logger - 1WFO - fractal-k881.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:12,187 - logger - 1WFO - fractal-k81.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:12,378 - logger - 1WFO - fractal-k81.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:12,378 - logger - 1WFO - fractal-k81.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:12,378 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:12,378 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total CFU usage: 0.5 % 14:56:12,378 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total memory: 15.78389576 Gb 14:56:17,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total CFU usage: 0.5 % 14:56:17,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory: 15.78389576 Gb 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory: 15.63896576 Gb 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory: 15.63896576 Gb 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory: 15.78389575 Gb 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory: 15.78389575 Gb 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory usage: 0.5 % 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory usage: 0.5 % 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory usage: 0.5 % 14:56:27,369 - logger - 1WFO - fractal-k80.jpd.ikerjan.es Total Memory usage: 0.5 % 14:56:27,355 - logger - 1WFO - fractal-k80.jpd.iker</pre>
No error messages/All partial results are as expected Test observations Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result	2023-02-16	14:56:42,653 - logger - INFO - Tainted modes: ()
Test observations Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result	Success criteria	
Test configuration https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator Test conditions The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result Test result		I partial results are as expected
Test conditions The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result		
Test conditions The metrics exporter runs in the two worker's nodes. Remarks Reviewing the logs from the resource manager it can be observed that the information from the worker's nodes is given in the right way (as expected). Test result	Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
Remarks worker's nodes is given in the right way (as expected). Test result Image: Comparison of the right way (as expected).	Test conditions	
Test result	Remarks	
	Test result	

Table 92 - Validation Test T02_WP6T62-06_EC (overview)

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test		
Test ID	T03_WP6T62-06_EC		
Test type	Functional		
Test name	Testing through the REST API if the metrics exporter is working properly		
Date	15/11/2022		
Tester's Name	Ana Bautista		
Test scope or objective	/e		
The objective of this	test is to validate through the REST API if the metrics exporter is working properly.		
Steps			
Step 1 Prepare a node with Ubuntu and all needed dependencies (Python). This will be the ma			
Step 2	Deploy the metrics exporter container.		
Step 3	Deploy the resource manager container.		
Step 4	Deploy the custom orchestrator container.		
Step 5	Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes.		
Step 6	Deploy the metrics exporter container on each of the worker nodes.		
Step 7	Go to: http:// <node ip="">:61208/api/3/cpu</node>		
Results/Evidence			
Master node:			
"softiq": "cpucore": Worker node 2: ← → {"total": "softirq":	C No es seguro NODE_IP 61208/api/3/cpu Image: constraints Image: const		
Success criteria	Il partial results are as expected		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator		
Test conditions	The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes.		
Remarks	The REST API exposed by the custom orchestrator is reached by the resource manager and provides information about the nodes previously configured (as expected).		
Test result			

Table 93 - Validation Test T03_WP6T62-06_EC (overview)

Copyright © FRACTAL Project Consortium	110 of 140
	119 01 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Test ID	Validation test
restin	T04_WP6T62-06_EC
Test type	Functional
Test name	Testing how the resource manager behaves if the nodes are stressed
Date	15/11/2022
Tester's Name	Ana Bautista
Test scope or obje	ctive
The objective of t	nis test is to observe how the resource manager behaves if the nodes are stressed.
Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python). This will be the master node.
Step 2	Deploy the metrics exporter container.
Step 3	Deploy the resource manager container.
Step 4	Deploy the custom orchestrator container.
Step 5	Prepare tow nodes with Ubuntu and all needed dependencies (Python). These will be the worker's nodes.
Step 6	Deploy the metrics exporter container on each of the worker nodes.
Step 7	Install stress-ng o n one of the worker nodes.
Step 8	Execute the command stress-ngcpu 8timeout 60s which will stress the node for 60 seconds.
Step 9	Review the logs from the resource manager (deployed on the master node).
Step 10	Check the alerts and the metrics in the logs of the resource manager.
Results/Evidence	
Master node:	
-	
	e-controller-orchestrator git:(UP6762-06-edge-orchestrator) stress-ngcpu 8timeout 60s [1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs)
stress-ng: info: Results 1:	[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs)
stress-ng: info: Results 1:	[1051804] dispatching hogs: 8 cpu
stress-ng: info: Results 1: Stressed worker n	[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs)
stress-ng: info: Results 1: Stressed worker n Worker node: frac 1013-02-17 12:40:44,687 1003-02:71 21:40:44,687 1003-02:71 21:40:44,687	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es logger - 1WFO - fractal-k80.ipd.ikerlan.es Total remov: 15.748800506 Gb logger - 1WFO - fractal-k80.ipd.ikerlan.es Total memory: 15.748800506 Gb logger - 1WFO - fractal-k80.ipd.ikerlan.es Total memory: 15.748800506 Gb logger - 1WFO - fractal-k80.ipd.ikerlan.es Total memory: 15.748800506 Gb</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 1023-02-17 12:40:44,687 1023-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:400-487 1033-02-17 12:400-487 1033-02-17 12:400-487 1033-02-1	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es logger - NHO - fractal-k8s0.ipd.ikerlan.es Total CPU usage 33.1 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory: 16.768805076 Gb logger - NHO - fractal-k8s0.ipd.ikerlan.es Mailable memory: 15.54231152000001 Gb logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Mailable memory: 15.54231152000001 Gb logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory usage: 7.4 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory is 10.7 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory is 10.7 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory is 10.7 % logger - NHO - fractal-k8s0.ipd.ikerlan.es Total memory is 10.7 % logger -</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,688 2023-02-17 12:40:44,688 2023-02-17 12:40:44,688 2023-02-17 12:40:44,688	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 10.78200576 06 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 06 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 06 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 06 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 06 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 05 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory: 15.74320576 05 logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total emory usage: 7.4% logger - NHO - fractal-k8s0.ipd.ikerlan.es Total</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,678 2023-02-17 12:40:44,678 2023-02-17 12:40:44,678 2023-02-17 12:40:44,678 2023-02-17 12:40:44,678 2023-02-17 12:40:44,788 2023-02-17 12:40:44,788 2023-02-17 12:40:44,798 2023-02-17 12:40:44,798 202	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es total memory: 15.04330570 Gb logger - INFO - fractal-k80.ipd.ikerlan.es Total memory: 15.04330575 Gb logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.5 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.5 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.5 % logger - INFO - fractal-k81.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k81.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k81.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total GPU usage 5.7 % logger - INFO - fractal-k80.ipd.ikerlan.es Total Memory 15.75318528 Gb</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 0033-00-17 12:40-44,667 0033-00-17 12:40-44,667 0033-00-17 12:40-44,667 0033-00-17 12:40-44,667 0033-00-17 12:40-44,667 0033-00-17 12:40-44,67 0033-00-17 12:40-44,67 0033-00-17 12:40-44,79 0033-00-17 12:40-44,79 003-00-17	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es total emory: 15.743808076 Gb logger - INFO - fractal-k80.ipd.ikerlan.es total memory: 15.743231152000001 Gb logger - INFO - fractal-k80.ipd.ikerlan.es total memory: 15.7318526 Gb logger - INFO - fractal-k80.ipd.ikerlan.es Total memory: 15.7318528 Gb logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.8 logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal-k80.ipd.ikerlan.es total memory usage: 7.2 % logger - INFO - fractal</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,607 0033-02-17 12:40:44,702 0033-02-17 12:40:40,702 0033-02-17 12:40:40,702 003	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 003-02-17 12:40:44/687 -003-02-17 12:40:44/687 -003-02-17 12:40:44/687 -003-02-17 12:40:44/687 -003-02-17 12:40:44/687 -003-02-17 12:40:44/687 -003-02-17 12:40:44/787 -003-02-17 12:40:44/782 -003-02-17 12:40:45/9,033 -003-02-17 12:40:59,033 -003-02-17 12:40:59,03 -003-02-17 12:40:59,03 -003	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es total emenory: 16.78309076 GD logger - 1WFO - fractal-k880.ipd.ikerlan.es Total memory: 15.542321152000001 GD logger - 1WFO - fractal-k880.ipd.ikerlan.es Total memory: 15.543232152000001 GD logger - 1WFO - fractal-k881.ipd.ikerlan.es Total memory: 15.575318528 GD logger - 1WFO - fractal-k881.ipd.ikerlan.es Total memory: 15.575318528 GD logger - 1WFO - fractal-k881.ipd.ikerlan.es Total memory: 15.755318528 GD logger - 1WFO - fractal-k881.ipd.ikerlan.es Total CWU usage <u>0.5 %</u> logger - 1WFO - fractal-k881.ipd.ikerlan.es Total MEMORY usage: 7.7 %</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,687 003-02-17 12:40:44,787 003-02-17 12:40:44,792 003-02-17 12:40:45,933 003-02-17 12:40:59,933 003-02-17 12:40:59,933	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es Total CPU usage 83.1 % logger - NHO - fractal-k80.ipd.ikerlan.es Total memory: 15.943231152000001 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total nemory: 15.943231152000001 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total nemory: 15.943231152000001 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total nemory: 15.953231152000001 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total memory: 15.95323152000001 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy: 15.95316528 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy: 15.975316528 Gb logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Total Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.2 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NHO - fractal-k80.ipd.ikerlan.es Iotal Remoy usage: 7.4 % logger - NH</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,792 2023-02-17 12:40:45,993 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,792 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,934 2023-02-17 12:40:59,934 202	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1033-02-17 12:40:44,687 1032-02-17 12:40:44,687 1032-02-17 12:40:44,687 1032-02-17 12:40:44,687 1032-02-17 12:40:44,792 1033-02-17 12:40:45,9033 1033-02-17 12:40:59,933 1033-02-17 12:40:59,933 10	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,687 2023-02-17 12:40:44,787 2023-02-17 12:40:44,792 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:45,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,933 2023-02-17 12:40:59,934 2023-02-17 12:40:59,934 202	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 003-02-17 12:40:44,687 -003-02-17 12:40:44,687 -003-02-17 12:40:44,687 -003-02-17 12:40:44,687 -003-02-17 12:40:44,687 -003-02-17 12:40:44,687 -003-02-17 12:40:44,787 -003-02-17 12:40:44,787 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,99 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,792 -003-02-17 12:40:44,93 -003-02-17 12:40:45,93 -003-02-17 12:40:45,93 -003-02-17 12:40:45,93 -003-02-17 12:40:45,93 -003-02-17 12:40:59,933 -003-02-17 12:40:59,933 -003-02-17 12:40:59,933 -003-02-17 12:40:59,934 -003-02-17 12:40:59,94 -003-02-17	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.</pre>
stress-ng: info: Results 1: Stressed worker n Worker node: frac 0033 00-11 12:40:44,667 0033 00-11 12:40:44,667 0033 00-11 12:40:44,667 0033 00-11 12:40:44,667 0033 00-11 12:40:44,667 0033 00-11 12:40:44,67 0033 00-11 12:40:44,67 0033 00-11 12:40:44,79 0033 00-11 12:40:45,9,933 0033 00-11 12:40:59,933 0033 00-11 12:40:59,933 0033 00-11 12:40:59,933 0033 00-11 12:40:59,934 0033 00-11 12:40:59,934 003 00-11 12:40:59,934 0033 00-11 12:40:	<pre>[1051804] dispatching hogs: 8 cpu [1051804] successful run completed in 60.07s (1 min, 0.07 secs) ode: fractal-k8s0.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es tal-k8s1.ipd.ikerlan.es Total memory: 15.453015206 logger - INFO - fractal-k8s0.ipd.ikerlan.es Total memory: 15.75318528 Go logger - INFO - fractal-k8s1.ipd.ikerlan.es Total memory: 15.75318528 Go logger - INFO - fractal-k8s1.ipd.ikerlan.es Total memory: 15.75318528 Go logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal PU usage 10.0 S logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal memory: 15.75318528 Go logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal memory: 15.75318528 Go logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal memory: 15.7783855576 Gb logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal CFU usage 100.0 S logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal memory: 15.7783855576 Gb logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal memory: 15.7783855576 Gb logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal Memory: 15.7783855576 Gb logger - INFO - fractal-k8s1.ipd.ikerlan.es Iotal CFU usage 100.0 S logger - INFO - fractal-k8s0.ipd.ikerlan.es Iotal Memory: 15.7783855576 Gb logger - INFO - fractal-k8s0.ipd.ikerlan.es Iotal Memory: 15.7783855576 Gb logger - INFO - fractal-k8s0.ipd.ikerlan.es Iotal Memory: 15.7745985576 Gb logger - INFO -</pre>

(continues)

Copyright © FRACTAL Project Consortium	120 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

(continued)

2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 2023-02-17 12:41:30,417 783320,01, 'nice':0,	<pre>logger - WARNING - CPU in node fractal-k880.ipd.ikerlan.es is orver 80%: CPU = 100.0 logger - 1NFO - Node fractal-k880.ipd.ikerlan.es is aircedy teinted as NoSchedule logger - 1NFO - fractal-k881.ipd.ikerlan.es is aircedy teinted as NoSchedule logger - 1NFO - fractal-k881.ipd.ikerlan.es total CPU usage: 0.6 % logger - 1NFO - fractal-k881.ipd.ikerlan.es total memory: 15.7481062400001 Gb logger - 1NFO - fractal-k881.ipd.ikerlan.es valiable memory: 15.57481062400001 Gb logger - 1NFO - fractal-k881.ipd.ikerlan.es valiable memory: 15.7481062400001 Gb logger - 1NFO - fractal-k880.ipd.ikerlan.es valiable memory: 15.74820580001 Gb logger - 1NFO - fractal-k880.ipd.ikerlan.es total CPU usage: 7.3 % logger - 1NFO - fractal-k880.ipd.ikerlan.es valiable memory: 15.74842058080001 Gb logger - 1NFO - fractal-k880.ipd.ikerlan.es valiable memory: 15.74842058080001 Gb logger - 1NFO - fractal-k880.ipd.ikerlan.es Alerts: [1676637659.0, 1676637765.0, 'CRITICAL', 'CPU_TOTAL', 100.0, 100.0, 100.0, [0.0], 'pid': 1051806, 'num_threads': 1, 'memory_info': [7311360, 5560726, 3723264, 1728512, 0, 27832320, 0], 'nice': 0, 'ppid': 1 6.7], 'num_tx_xwitchex': [1, 9550], 'memory_smap': None, 'top': None, 'udp': None, 'extended_stats': True, 'key': 'pid', 'time rrame': 'metricsexporter'), (cpu_percent: 9.0.4550755670181, 'gids': [1000, 1000], 'noe'; 'pid', 'status': 'R', 'cpu_tit ice': 0, 'ppid': 1051806, 'memory_percent: '9.0.45501755670181, 'gids': [1000, 1000], 'noe'; 'pid', 'status': 'R', 'cpu_tit ice': 0, 'ppid': 1051804, 'memory_percent: '9.0.6, 0, 0, 0, 0], 'name': 'stress.mg-cpu', 'status': 'R', 'cpu_tit ice': 0, 'ppid': 1051804, 'mem</pre>
Success criteria	
No error messages/A	All partial results are as expected
No error messages/A Test observations	All partial results are as expected
No error messages/A	
No error messages/# Test observations Test configuration	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node.
No error messages/A Test observations	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator
No error messages/# Test observations Test configuration	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node.
No error messages/# Test observations Test configuration	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes.
No error messages/ <i>I</i> Test observations Test configuration Test conditions	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. The node called fractal-k8s0.ipd.ikerlan.es is the node that was stressed.
No error messages/# Test observations Test configuration	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. The node called fractal-k8s0.ipd.ikerlan.es is the node that was stressed. As it can be observed in the "Results 1" screenshot: when the CPU usage of a node is over 80% it is considered
No error messages/ <i>I</i> Test observations Test configuration Test conditions	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. The node called fractal-k8s0.ipd.ikerlan.es is the node that was stressed. As it can be observed in the "Results 1" screenshot: when the CPU usage of a node is over 80% it is considered as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if
No error messages/ <i>I</i> Test observations Test configuration Test conditions	All partial results are as expected https://github.com/project-fractal/WP6T62-06-edge-controller-orchestrator The metrics exporter, resource manager and custom orchestrator containers run in the master node. The metrics exporter runs in the two worker's nodes. The node called fractal-k8s0.ipd.ikerlan.es is the node that was stressed. As it can be observed in the "Results 1" screenshot: when the CPU usage of a node is over 80% it is considered as tainted (low on resources and restricted) as NoSchedule. According to the component documentation, if any of the monitored resources are above some fixed thresholds, that node is no longer able to perform any

Table 94 - Validation Test T04_WP6T62-06_EC (overview)

Copyright © FRACTAL Project Consortium	121 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.2 Orchestration (Agent Nodes Controller) component complete templates

Validation test			
Test ID	T01_WP6T62-06_ANC		
Test type	Functional		
Test name	Testing that the Agent nodes controller can be installed without any issues		
Date	20/01/2023		
Tester's Name	Ana Bautista, Adrian Moran		
Test scope or objective			
The objective of the te	est is to validate if the Agent nodes controller can be installed without any issues.		
Steps			
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).		
Step 2	Launch backend-service-manager (Service Manager).		
Step 3	Launch API Server.		
Step 4	Launch frontend-service-manager (Executor Node).		
Results/Evidence			
{"version":":	tal-qemu:~\$ curl -X GET http://localhost:5001/api/v1/version 1.0"} tal-qemu:~\$ curl -X GET http://localhost:5001/api/v1/tasks		
Success criteria	Success criteria		
No error messages/All	partial results are as expected		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration		
Test conditions	API Server, Service Manager and Executor Node runs in the same node.		
Remarks	All the subcomponents are up and running.		
Test result			
Passed			

Table 95 - Validation Test T01_WP6T62-06_ANC (overview)

Copyright © FRACTAL Project Consortium	122 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test		
Test ID	T02_WP6T62-06_ANC		
Test type	Functional		
Test name	Testing basic orchestration functionality		
Date	09/02/2023		
Tester's Name	Ana Bautista, Adrian Moran		
Test scope or objectiv	/e		
The objective of the t	est is to validate basic orchestration functionality.		
Steps			
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).		
Step 2	Launch backend-service-manager (Service Manager).		
Step 3	Launch API Server.		
Step 4	Launch frontend-service-manager (Executor Node).		
Step 5	From a different device, use REST API to create a new task.		
Results/Evidence			
172.16.105.43 Executor Node: 2023-02-09 11:14:8 ('task', 71, 'json 2023-02-09 11:14:8 2023-02-09 11:15:6 2023-02-09 11:15:6	[09/Feb/2023 11:13:59] "DELETE /api/v1/tasks/test HTTP/1.1" 200 - [09/Feb/2023 11:14:58] "POST /api/v1/tasks/test HTTP/1.1" 201 - 58,501 INFO heartbeat received 1', {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 58,514 INFO running task: test 10,504 INFO heartbeat received 12,507 INFO heartbeat received 13,535 INFO task: test completed with return code: 0 04,508 INFO heartbeat received		
Success criteria			
	Il partial results are as expected		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration		
Test conditions	API Server, Service Manager and Executor Node runs in the same node.		
Remarks	Basic workflow is completed successfully.		
Test result			
Passed			

Table 96 - Validation Test T02_WP6T62-06_ANC (overview)

Copyright © FRACTAL Project Consortium	123 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test	
Test ID	T03_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing that a running task can be deleted	
Date	09/02/2023	
Tester's Name	Ana Bautista, Adrian Moran	
Test scope or object	ive	
The objective of the	test is to validate that a running task can be deleted.	
Steps		
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).	
Step 2	Launch backend-service-manager (Service Manager).	
Step 3	Launch API Server.	
Step 4	Launch frontend-service-manager (Executor Node).	
Step 5	From a different device, use REST API to create a new task and delete it.	
Results/Evidence		
	ELETE http://172.16.58.4:5001/api/v1/tasks/test :{"status":"ok"},"status":"task: test deleted successfuly from API host."}	
172.16.105.43	[09/Feb/2023 11:17:23] "DELETE /api/v1/tasks/test HTTP/1.1" 200 -	
Success criteria		
No error messages/	All partial results are as expected	
Test observations		
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration	
Test conditions	API Server, Service Manager and Executor Node runs in the same node.	
Remarks	Task is deleted from API Server repository succesfully.	
	Deleted task is not deleted from Executor node .tasks folder, which can lead to problems	
	like lack of storage or DoS attacks.	
Test result	, , , , , , , , , , , , , , , , , , ,	

Table 97 - Validation Test T03_WP6T62-06_ANC (overview)

Copyright © FRACTAL Project Consortium	124 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Test ID	T04 WP6T62-06 ANC
Test type	Functional
Test name	Testing that a running task can be stopped and started again
Date	09/02/2023
Tester's Name	Ana Bautista, Adrian Moran
Test scope or objective	
	est is to validate that a running task can be stopped and started again.
Steps	est is to valuate that a running task can be stopped and started again.
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create a new task, stop and then start it.
Results/Evidence	
Client:	
{"status":"task: test → task curl -X POST h {"status":"ok"} → task curl -X POST h {"status":"ok"} → task curl -X DELETE	ttp://172.16.58.4:5001/api/v1/tasks/test -F file=@test_task.py -F "cmd=test_task.py" -F "rt=" created successfuly."} ttp://172.16.58.4:5001/api/v1/tasks/test/stop ttp://172.16.58.4:5001/api/v1/tasks/test/start ttp://172.16.58.4:5001/api/v1/tasks/test atus":"ok"},"status":"task: test deleted successfuly from API host."}
127.0.0.1 [09/Feb/ 172.16.105.43 [09/ 172.16.105.43 [09/ 127.0.0.1 [09/Feb/ from server {'status':	Feb/2023 11:18:35] "POST /api/v1/tasks/test HTTP/1.1" 201 - 2023 11:18:35] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - Feb/2023 11:18:45] "POST /api/v1/tasks/test/stop HTTP/1.1" 200 - Feb/2023 11:18:48] "POST /api/v1/tasks/test/start HTTP/1.1" 200 - 2023 11:18:48] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - 'ok'} Feb/2023 11:18:58] "DELETE /api/v1/tasks/test HTTP/1.1" 200 -
('task', 71, 'json', {' 2023-02-09 11:18:38,786 2023-02-09 11:18:36,788 2023-02-09 11:18:36,788 2023-02-09 11:18:40,799 2023-02-09 11:18:44,799 2023-02-09 11:18:45,786 2023-02-09 11:18:45,786 2023-02-09 11:18:46,800 2023-02-09 11:18:46,800 ('task', 71, 'json', {' 2023-02-09 11:18:56,860 2023-02-09 11:18:56,860 2023-02-09 11:18:54,812 2023-02-09 11:18:56,814	i INFO task: test terminated with return code: -15 INFO heartbeat received INFO heartbeat received task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) INFO running task: test INFO heartbeat received INFO heartbeat received INFO heartbeat received INFO heartbeat received INFO heartbeat received
Success criteria	
	partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration
Test conditions	API Server, Service Manager and Executor Node runs in the same node.
Remarks	Task lifecycle in correctly handled.
Test result	
Passed	

Table 98 - Validation Test T04_WP6T62-06_ANC (overview)

	Copyright © FRACTAL Project Consortium	125 of 149
--	--	------------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test	
Test ID	T05_WP6T62-06_ANC	
Test type	Functional	
Test name	Testing that a running multiple tasks is possible and list their state	
Date	09/02/2023	
Tester's Name	Ana Bautista, Adrian Moran	
Test scope or object	ive	
The objective of the	test is to validate that a running multiple tasks is possible and list their state.	
Steps		
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).	
Step 2	Launch backend-service-manager (Service Manager).	
Step 3	Launch API Server.	
Step 4	Launch frontend-service-manager (Executor Node).	
Step 5	From a different device, use REST API to create a new tasks.	
Results/Evidence		
→ task curl -X GET { {"tasks":["test","test → task curl -X GET { {"status":"ok","task	nttp://172.16.58.4:5001/api/v1/tasks/test/status _status":"running"} http://172.16.58.4:5001/api/v1/tasks/test2/status	
172.16.105.43 127.0.0.1 - [09 from server {'sta 172.16.105.43 172.16.105.43 from server {'sta 172.16.105.43 172.16.105.43	<pre>- [09/Feb/2023 11:21:41] "GET /api/v1/tasks HTTP/1.1" 404 - - [09/Feb/2023 11:21:49] "POST /api/v1/tasks/test HTTP/1.1" 201 - //Feb/2023 11:21:50] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - stus': 'ok', 'tasks': ['test']} - [09/Feb/2023 11:21:51] "GET /api/v1/tasks HTTP/1.1" 200 - - [09/Feb/2023 11:21:55] "POST /api/v1/tasks/test2 HTTP/1.1" 201 - stus': 'ok', 'tasks': ['test', 'test2']} - [09/Feb/2023 11:21:59] "GET /api/v1/tasks HTTP/1.1" 200 - - [09/Feb/2023 11:21:59] "GET /api/v1/tasks/test/status HTTP/1.1" 200 - - [09/Feb/2023 11:22:12] "GET /api/v1/tasks/test/status HTTP/1.1" 200 - - [09/Feb/2023 11:22:16] "GET /api/v1/tasks/test/status HTTP/1.1" 200 -</pre>	

(continues)

Copyright © FRACTAL Project Consortium	126 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

(continued)

Success criteria	
No error messages/	All partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-06-mid-range-orchestration
Test conditions	API Server, Service Manager and Executor Node runs in the same node.
	It is possible to list the tasks.
	It is possible to add multiple tasks and get their status.
	A bug has occured generating two task, one named "test" and other one named "test2".
	Executor Node computes the execution path from name, and since it does not deletes the
	old tasks, this leads to failure:
Remarks	<pre>58 logging.info(f*running task: {task_name}") 59 selict.cond("task-cond"task-cond", {task_name}", task_name"; task_name]) 60 task_dir = [dir 61 for dir in os.listdir(os.getcwd() + "/.tasks/") if (task_name in dir and "venv" != dir) 62] 63 task_dir.sort() 64 task_dir = os.getcwd() + "/.tasks/" + task_dir[-1] 65 python_dir = os.getcwd() + "/.tasks/" + "venv/bin" 66 task_args = task_args.split(" ") 67 try: 68 PROCESS = await asyncio.subprocess.create_subprocess_exec(</pre>
	This way of computing task_dir is bugged. If the same node has executed in their lifetime a
	tasks called "test" and "test2", there won't be any chance of running "test" task, since this
	piece of code will always select last "test2-timestamp" folder:
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:18 test-1675941515
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:18 test-1675941528
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:21 test-1675941710
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:24 test-1675941879
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:27 test-1675942024 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:30 test-1675942222
	drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:30 test-1675942222 drwxrwxr-x 2 ikerlan ikerlan 4096 Feb 9 11:23 test2-1675941833
	drwxrwxr-x 6 ikerlan ikerlan 4096 Feb 9 11:25 test2 10/0941035
	\rightarrow .tasks git:(main) ×
Test result	
Not passed	

Table 99 - Validation Test T05_WP6T62-06_ANC (overview)

Copyright © FRACTAL Project Consortium	127 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test
Test ID	T06 WP6T62-06 ANC
Test type	Functional
Test name	Testing the behavior of the orchestrator with multiple Executor Nodes
Date	09/02/2023
Tester's Name	Ana Bautista, Adrian Moran
Test scope or objectiv	ve
The objective of the [†]	test is to validate the behaviour of the orchestrator with multiple Executor Nodes.
Steps	
Step 1	Prepare a node with Ubuntu and all needed dependencies (Python).
Step 2	Launch backend-service-manager (Service Manager).
Step 3	Launch API Server.
Step 4	Launch two instances of frontend-service-manager (Executor Node).
Step 5	From a different device, use REST API to create two simultaneous tasks.
Results/Evidence	
API:	
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, 127.0.0.1 [09, 127.0.0.1 [09, 127.0.0.1 [09, 127.0.0.1 [09, 13:22:3 ('task', 71, 'json 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'json 2023-02-09 13:22:4	<pre>[09/Feb/2023 13:22:36] "POST /api/v1/tasks/test HTTP/1.1" 201 - /Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200 - //Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" //Feb/2023 13:22:42] TFO running task: ikerlan //Feb/2023 INFO heartbeat received // #GE0 HEG/Feb/2023 INFO heartbeat received //</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, Executor Node 1: 2023-02-09 13:22:3 ('task', 71, 'jsor 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'jsor 2023-02-09 13:22:4 2023-02-09 13:22:4 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 2023-02-04 203-04 203-04 203-04 203-04 203-04	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, Executor Node 1: 2023-02-09 13:22:3 ('task', 71, 'jsor 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'jsor 2023-02-09 13:22:4 Success criteria No error messages/A	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200 - 84,827 INFO heartbeat received 34,827 INFO heartbeat received 34,827 INFO heartbeat received 42,737 INFO heartbeat received 42,737 INFO heartbeat received 1, {'task_name': 'ikerlan', 'args_to_run': 'test_task.py', 'return_type': ''}) 42,823 INFO running task: ikerlan</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, Executor Node 1: 2023-02-09 13:22:3 ('task', 71, 'jsor 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'jsor 2023-02-09 13:22:4 Constant of the second sec	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200 - #4,827 INFO heartbeat received *4,827 INFO heartbeat received *4,827 INFO nunning task: test *66,828 INFO heartbeat received *42,737 INFO heartbeat received *42,737 INFO heartbeat received *44,737 INFO heartbeat received *44,827 INFO heartbeat received *44,827 INFO heartbeat received *42,737 INFO heartbeat received *42,737 INFO heartbeat received *44,737 INFO heartbeat received *44,739 INFO heartbeat received ************************************</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, 2023-02-09 13:22:3 ('task', 71, 'json 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'json 2023-02-09 13:22:4 2023-02-09 13:22:4 Success criteria No error messages/A Test observations Test configuration	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, 2023-02-09 13:22:3 ('task', 71, 'jsor 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'jsor 2023-02-09 13:22:4 2023-02-09 13:22:4 2023-02-09 13:22:4 Success criteria No error messages/A Test observations Test configuration Test conditions	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200 - 24,827 INFO heartbeat received, {'task_name': 'test', 'args_to_run': 'test_task.py', 'return_type': ''}) 66,821 INFO running task: test 42,737 INFO heartbeat received, {'task_name': 'ikerlan', 'args_to_run': 'test_task.py', 'return_type': ''}) 42,823 INFO running task: ikerlan 44,739 INFO heartbeat received, {'task_name': 'ikerlan', 'args_to_run': 'test_task.py', 'return_type': ''}) 42,823 INFO running task: ikerlan 44,739 INFO heartbeat received, in the same as expected, in the same node.</pre>
127.0.0.1 [09, 172.16.105.43 127.0.0.1 [09, 2023-02-09 13:22:3 ('task', 71, 'json 2023-02-09 13:22:3 2023-02-09 13:22:3 Executor Node 2: 2023-02-09 13:22:4 ('task', 74, 'json 2023-02-09 13:22:4 2023-02-09 13:22:4 Success criteria No error messages/A Test observations Test configuration	<pre>/Feb/2023 13:22:36] "GET /api/v1/tasks/test/download HTTP/1.1" 200 - [09/Feb/2023 13:22:42] "POST /api/v1/tasks/ikerlan HTTP/1.1" 201 - /Feb/2023 13:22:42] "GET /api/v1/tasks/ikerlan/download HTTP/1.1" 200</pre>

Table 100 - Validation Test T06_WP6T62-06_ANC (overview)

Copyright © FRACTAL Project Consortium	128 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.3 Runtime Manager component complete templates

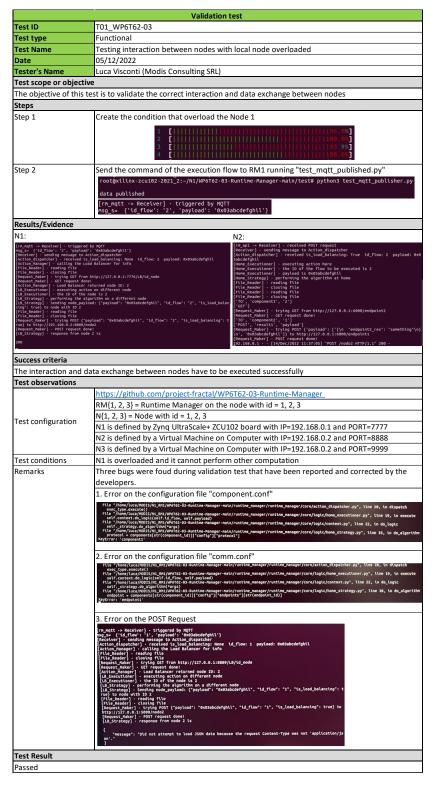


Table 101 - Validation Test T01_WP6T62-03 (overview)

Copyright © FRACTAL Project Consortium	129 of 149
--	------------

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test		
Test ID	T02_WP6T62-03		
Test type	Functional		
Test Name	Testing interaction in the local node when the local node can perform the computation		
Date	05/12/2022		
Tester's Name	Luca Visconti (Modis Consulting SRL)		
Test scope or object	ive		
The objective of this	test is to validate the correct interaction and data exchange in the local node		
Steps			
Step 1	Send the command of the execution flow to RM1 running "test_mqtt_published.py"		
	root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py data published [rm_mqtt -> Receiver] - triggered by MQTT msg s= {'id flow': '2', 'payload': '0x03abcdefghil'}		
Results/Evidence	isg_se (to_rtow . 2 , paytood . oxosobcatignet)		
N1:	[rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}		
	<pre>[Receiver] - Sending message to Action_dispatcher [Action_Manager] - calling the Load Balancer for info [File_Reader] - reading file [File_Reader] - reading file [Request_Maker] - trying GET from http://127.0.0.1:7776/LB/id_node [Request_Maker] - Load Balancer returned node ID: None [Mome_Executioner] - the ID of the flow to be executed is 2 [Home_Executioner] - the ID of the flow to be executed is 2 [Home_Executioner] - payload is &0x83abcdefghil [Home_Executioner] - load Balancer returned node ID: None [Home_Executioner] - the ID of the flow to be executed is 2 [Home_Executioner] - the ID of the flow to be executed is 2 [Home_Executioner] - load Balancer returned node [File_Reader] - reading file [File_Reader] - reading file [File_Reader] - colsing file</pre>		
Success criteria			
The interaction and	data exchange in the local nodes have to be executed successfully		
Test observations			
	https://github.com/project-fractal/WP6T62-03-Runtime-Manager RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3		
Test configuration	N{1, 2, 3} = Node with id = 1, 2, 3 N1 is defined by Zynq UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777		
	N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888 N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999		
Test conditions	N1 is not overloaded and it can perform any computation		
Remarks			
Test Result			
· · · · · ·			

Table 102 - Validation Test T02_WP6T62-03 (overview)

Copyright © FRACTAL Project Consortium	130 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test			
Test ID	T03 WP6T62-03			
Test type	Functional			
Test Name		Testing interaction between nodes with Node 1 and Node 2 overloaded		
Date	05/12/2022			
Tester's Name	Luca Visconti (Modis Consulting SRL)			
Test scope or objectiv				
		re hetween noder		
-	test is to validate the correct interaction and data exchang	ge between nodes		
Steps	Constants the second this state are also at the Newley 4			
Step 1	Create the condition that overload the Node 1 1 [
step 2	Create the condition that overload the Node 2 1 [95.3%] 92.3%] 89.3%]		
Step 3	Send the command of the execution flow to RM1 runni root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Managerid data published [rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}			
Results/Evidence				
N1:	N3:			
[Action_Manager] - calling the L [File_Reader] - reading file [File_Reader] - closing file [Request_Maker] - trying GCT from [Request_Maker] - CGT request do [Action_Manager] - Load Balancer [LB_Executioner] - executing acti [LB_Executioner] - executing action [LB_Executioner] - executing action [LB_Execution] - file [File_Reader] - cooling file	<pre>d': "083abcdefghll'] d': "083abcdefghll'] d': "083abcdefghll determined and and and and and and and and and an</pre>	ng file ng file 2'] Ing GET from http://127.0.0.1:6000/endpoint2		
Success criteria				
	ata exchange between nodes have to be executed succes	stully		
Test observations				
	https://github.com/project-fractal/WP6T62-03-Runtim			
	$RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3$			
Test configuration	$N{1, 2, 3} = Node with id = 1, 2, 3$			
	N1 is defined by Zynq UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777			
	N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888			
	N3 is defined by a Virtual Machine on Computer with IF			
Test conditions	N1 and N2 is overloaded and it cannot perform other c	omputation		
Remarks				
Test Result				

Table 103 - Validation Test T03_WP6T62-03 (overview)

Copyright © FRACTAL Project Consortium	131 of 149

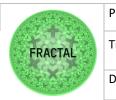


Project	FRACTAL
Title	FRACTAL engineering framework validation
Del. Code	D6.4

	and the second				
Test ID	Validation test T04 WP6T62-03				
Test type	Functional				
Test Name	Task Scheduling on the local node				
Date	05/12/2022				
Tester's Name	Luca Visconti (Modis Consulting SRL)				
Test scope or objective					
	est is to validate the correct execution of the Task in the local node				
Steps	Cond the command of the everytion flow "1" to DM1 marks which work with the "				
Step 1	Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with "id flow=1".				
	root@xilinx-zcu102-2021_2:-/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py				
	data published				
	[rm_mqt -> Receiver] - triggered by MQTT nsg_s= ('id_flow': '1', 'payload': '0x03abcdefghil')				
Store 2					
Step 2	Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id flow=2".				
	root@xilinx-zcu102-2021_2:-/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py				
	data published				
	<pre>[rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}</pre>				
Char 2					
Step 3	Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id flow=3".				
	root@xilinx-zcu102-2021_2:~/N1/WP6T62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py				
	data published				
	[rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '3', 'payload': '0x03abcdefghil'}				
D 4 (5 · · ·					
Results/Evidence					
Result Step 1: Flow1:	[rm_mott> Receiver] - triggered by MpTT				
TO component	<pre>Int 1 IN: [rm_sett &scotare] . tiggered by mpt at 1 IN: [rm_sett &scotare] . tiggered by mpt at 1 IN: [rm_sett &scotare] . tiggered by mpt at 1 IN: [rm_sett &scotare] . tiggered by mpt at 1 IN: [rm_sett &scotare] . tiggered by mpt if [rm_sett &scotare] . tigge</pre>				
POST paylo TO component	au [Action_Messaper] - calling the Load Balancer for info nt2 1 [File_Reader] - reading File [File_Reader] - closing File				
POST resul TO componen	<pre>L1 [Request_Paker] - trying GCT from http://i27.0.0.1:7776/LB/Ld_node nt3 1 [Request_Paker] - GCT request done!</pre>				
POST resul	t1 result2 [How_fxecutioner] - executing action here [None_fxecutioner] - the 10 of the flow to be executed is 1 [None_fxecutioner] - payload is doublederhall				
	[blue_Strategy] - performing the algorithm at home [File_Reader] - reading file [File_Reader] - indexing file				
	(File_Reader) - consing file [File_Reader] - closing file [File_Reader] - closing file				
	<pre>(100; 'components', '1') ('NOT', 'paptand') (Baganti, note', 'paptand') (Baganti, note', 'paptand') ('south, note', 'paptand') ('south, 'papt</pre>				
	['903T', 'resulti']				
	<pre>[Request_Maker] - trying POST {'payload': {'lin "endpointi_res': "something"\n}\n']) to http://12 7.0.0.10000/endpointi [Request_Maker] - POST request done!</pre>				
	[Bewert_Beics] - trying HOT ('gayLass': ['[n 'modestst_res': 'sseething'ho]us']) to http://12 /f.d., index/pointening /f.d., i				
	nt; rest "something (n)() to http://ii/.6.ii6009/endpoint; [Request_maker] - POST request done:				
Result Step 2:					
Flow2: 2	[rm_ngtt -> Receiver] - triggered by MQTT N1: Hsg.s= ('id.flow': '2', 'payload': '0x03abcdefghli')				
TO COMPONE GET	<pre>sht1 2 int 2 if (n, ngtt - n Marinery). triggered by GTI int 2 if (n, ngtt - n Marinery). triggered b</pre>				
TO compone	ent2 1 [Action_Manager] - calling the Load Balancer for info [File_Reader] - reading file [It1 payload [File_Reader] - closing file				
1051 1050	[Request_Maker] - Closing Tile [Request_Maker] - trying GET from http://127.0.0.1:7776/LB/ld_node				
	[Action Manager] - Load Balancer returned node ID: None [Nome_Executioner] - executing action here				
	[Home_Executioner] - the ID of the flow to be executed is 2. [Home_Executioner] - payload is #x03abcdefghll [Home_Executionul - performing the algorithm at home				
	[File_Reader] - reading file [File_Reader] - closing file				
	[File_Reader] - reading file [File_Reader] - closing file				
	[No , tongonent; , 2] ['6ET'] [Reguest Maker] - trying GET from http://127.8.0.1:0000/endpoint2				
	[Request_Naker] - trying EET (row http://127.0.0.110080/endpoint2 [Request_Naker] - EET request does ['10] - 'component2', '11] ['Request_Naker] - trying Fof['] ['Request_Naker] - trying FOT ('paylaad': ['(\n "endpoint2_res"; "something"\n)\n", 'ButBabcderfg [']] ['Bequest_Naker] - FOT request_done!				
	[POST , 'resulti', 'payload'] [Request_Maker] - trying POST ('payload': ['(\n "endpoint2_res": "something"\n)\n', '0x03abcdefgl [11]] to http://j27.0.n_1.is600/cendmint1				
	[Request_Naker] - POST request done!				
Result Step 3:					
Flow3: 3 TO compone	<pre>Int 2 If Restive? - triggered by April If April 2 If</pre>				
GET TO compone	[Action_dispatcher] - received is_load_balancer for info [Action_dispatcher] - received is_load_balancer for info [Action_dispatcher] - calling the load Balancer for info				
TO COMPONE GET	[File_Reader] - reading file [File_Reader] - closing file				
	[Request_Maker] - trying GET from http://127.6.0.1:7776/LB/id_mode [Request_Maker] - GET request_done!				
	<pre>[sequers_twist"] : trying Git From http://127.6.3.12776/git/d.gode [Sequers_March - Git request done: /Actum_Manager] - Load Balancer removate [mem_Executioner] - the Die of the Flow to be executed is 3 [mem_Executioner] - the Die of the Flow to be executed is 3 [mem_Executioner] - projoad is detablood foll [mem_Executioner] - projoad is detablood foll [mem_Executioner] - the Die of the Flow to be executed [Flow_Header] - classing the algorithm at howe [Flow_meder] - classing file [Flow_meder] - classing file [Flow_meder] - reaction file [Flow_meder] - classing file [Flow_meder] - classing file</pre>				
	[Hone_Structure] - payload is oxfactored for the forme_Structure is a [Hone_Structure] - performing the algorithm at home				
	[File_Reader] - reading file [File_Reader] - closing file				
	[File Reader] - reading file [File Reader] - closing file [700] :readement1 = 131				
	('GE') ['GE'] [Request Maker] - trying GET from http://127.0.0.1:6000/endpoint2				
	[Request_Maker] - trying GET from http://127.0.0.1:6000/endpoint2 [Request_Maker] - GET request done! ['10', 'component', 'z']				
	['eEY'] [Request_Haker] - trying GET from http://127.0.0.1:6000/endpoint2 [Request_Haker] - GET request done!				
	Incluest_haver j - GET request_Gome)				
Success criteria					
	ask in local node have to be executed successfully				
Test observations	https://github.com/project-fractal/WP6T62-03-Runtime-Manager				
	RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3				
	N(1, 2, 3) = Node with id = 1, 2, 3				
Test configuration N1, 2, 3, - Node With Id = 1, 2, 3 N1 is defined by Zyng UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777					
	N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888				
	N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999				
Test conditions	N1 is not overloaded and it can perform any computation				
Remarks					
Test Result					
Passed					

Table 104 - Validation Test T04_WP6T62-03 (overview)

Copyright © FR	ACTAL Project Consortium	132 of 149
----------------	--------------------------	------------



Project	FRACTAL
Title	FRACTAL engineering framework validation
Del. Code	D6.4

T	Validation test
Test ID	T05_WP6T62-03
Test type	Functional
Test Name Date	Task Scheduling on the remote node 05/12/2022
Tester's Name	Luca Visconti (Modis Consulting SRL)
Test scope or objecti	
	test is to validate the correct execution of the tasks in the remote node
Steps	
Step 1	Create the condition that overload the Node 1
	1 [
Step 2	Send the command of the execution flow "1" to RM1 running "test_mqtt_published.py" with
	"id_flow=1". root&ilinx-zcu102-2021_2:-/N1/NPGT62-03-Runtime-Manager-main/test# python3 test_mqtt_publisher.py
	data published [rm.mqt -> Receiver] - triggered by MQTT msg.s= ('id_flow': '1', 'payload': '0x03abcdefghil')
Step 3	Send the command of the execution flow "2" to RM1 running "test_mqtt_published.py" with "id flow=2". rootgatline: rootgatline: data published
	<pre>[rm_mqtt -> Receiver] - triggered by MQTT msg_s= {'id_flow': '2', 'payload': '0x03abcdefghil'}</pre>
Step 4	
Step 4	Send the command of the execution flow "3" to RM1 running "test_mqtt_published.py" with "id_flow=3". root&tltkx=zcu02-2021_2:-/N1/NPG102-03-Runtime-Hanager-Main/test# python3 test_mqtt_publisher.py data_published [rm_mqtt -> Receiver] - triggered by MOTT mg_s= ("id_flow: '3', "psyload': "0x03abcdefghl1')
Results/Fuidence	to the test of the second
Results/Evidence Result step 2:	
Flow1: 1 TO component POST payload TO component POST result1 TO component	21
magging (C) (C) The M 1 are support denoted () and the sensitive to a functional standard () and the sensitive to the control standard () and the sensitive to the sensitive to the sensitive to the functional standard () and the sensitive f	PERSURPT N22 Visition (intermediation) (intermediation) (intermediation) Visition (intermediation) (intermediation) (intermediation) (intermediation) Visition (intermediatio
Result step 3: Flow2: 2 TO componen GET TO componen POST result N1:	t2 1 i payload
In acts - Restore() - tragent Restore() - Restore() - tragent Restore() - Restore() - reserved b Restore() - Restore() - reserved b Restore() - Restore() - Re	by spit. The spit. The spit and spit of the spit and
Result step 4: Flow2: 3 TO componen GET TO componen GET N1:	t1 2 t2 2
10.113	<pre>W M2 W2 W2 W2 W2 W2 W2 W2 W2 W2 W</pre>
Success criteria	
	task and the exchange data with remote node have to be executed successfully
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-03-Runtime-Manager RM{1, 2, 3} = Runtime Manager on the node with id = 1, 2, 3 N{1, 2, 3} = Node with id = 1, 2, 3 NI is defined by Zynq UltraScale+ ZCU102 board with IP=192.168.0.1 and PORT=7777 N2 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=8888
	N3 is defined by a Virtual Machine on Computer with IP=192.168.0.2 and PORT=9999
Test conditions	N1 is overloaded and it cannot perform other computation
Remarks	
Test Result	
Passed	

Table 105 - Validation Test T05_WP6T62-03 (overview)

Copyright © FRACTAL Project Consortium	133 of 149
--	------------

199 <u>1</u> 200	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.4 Data Ingestion component complete templates

	Validation test			
Test ID	T01_WP6T62-01_DI - Testing Apache NiFi			
Test type	Functional-Installation			
Test name	Installation of Apache NiFi			
Date	13/01/2023			
Tester's Name	Nicola Alchera (Rulex)			
Test scope or objectiv				
	s test is to validate that Apache Nifi can be installed without any issue.			
Steps				
Step 1	Prepare a node with Ubuntu and required dependencies (Python)			
Step 2	Download files and checking prerequisites			
Step 3	Configure and run ApacheNifi			
Step 4	Login and Open Apache Nifi in browser			
Results/Evidence				
	s been successfully opened and it is ready to use			
the second second has	C A Not secure Mayes/10.138.028/041/ml/			
	nifi 🗅 🖓 💀 🖻 🖻 🌹 🔭			
	🙆 Navigate 😑			
	e e 🖸 1:1			
	Operate E			
	Process Broup #53/6444-0105-1000-65-3/346ee6201588			
	0 * % > = 2-2-			
	C 🗈 🖉 🖉 🖿 antre			
Success criteria				
	/All partial results are as expected			
Test observations				
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion			
Test conditions	The tests have been performed on a node with Ubuntu 22.04.			
	Running ./nifi.sh start you might be warned that the Java home path is empty. This seems to			
	be a warning. It can be fixed using the following command "export			
	JAVA_HOME="your_java_folder_path"			
	micola.alchera@rulex.ai@ubuntu-l-vm:-/nifi-1.16.3/bin\$./nifi.sh status			
	mifi.sh: JAVA_MOME not set; results may vary			
	Java home:			
. .	Nifi home: /home/nicola.alchera/nifi-1.16.3			
Remarks	Bootstrap Config File: /home/nicola.alchera/nifi-1.16.3/conf/bootstrap.conf			
	2023-01-12 16:54:32,680 INFO [main] org.apache.nifi.bootstrap.Command Apache NiFi is currently running, listening to Boo tstrap on port 43525, PID=39872			
	You cannot download the tarball file from https://nifi.apache.org/download.html because the			
	1.18 and 1.19 release has the .zip binary files only. To download the .tar file you need to			
	consider the archive https://archive.apache.org/dist/nifi/ . The latest version with the .tar file			
	available is the 1.16.3 and this is the one that has been used.			
Fest result				
Passed				

Table 106: Validation Test T01_WP6T62-01_DI (overview)

Copyright © FRACTAL Project Consortium	134 of 149
--	------------

27 ¹ 272	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

		Validation	n test		
Test ID	T02_WP6T62-01_C				
Test type	Functional				
Test name	Installation and co	Installation and configuration of PySpark			
Date	16/02/2023				
Tester's Name	Nicola Alchera (Rul	ex)			
Test scope or objectiv	/e				
The objective of this t	est is to ensure that	PySPark can be instal	led and configured withou	ut any issue.	
Steps					
Step 1	Prepare a node wit	h Ubuntu and the rec	quired dependencies (Pyth	ion 3)	
Step 2	Install pySpark and	open it on python3			
Results/Evidence					
pyspark.BarrierTaskInfo(nyspark_BasicProfiler(pyspark.types	
pyspark.BaicProfiler(pyspark.Collable(pyspark.Collable(pyspark.cllable(pyspark.cllable(pyspark.nivecontext(pyspark.nivecontext(pyspark.Arxshalserializer(pyspark.Arxshalserializer(pyspark.Asy(pyspark.BarrierTaskIcOntext(pyspark.BarrierTaskIcOntext(pyspark.BaicProfiler(pyspark.Celkalserializer(pyspark.sQLContext(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkContext(pyspark.sparkContext(pyspark.sparkStaptConfect pyspark.statusTracker(pyspark.statusTracker(pyspark.statusTracker(pyspark.spot(context(pyspark.sparkConfect(pyspark.sparkConfect(pyspark.sparkConfect()	pyspark.broadcast pyspark.cast(pyspark.cast(pyspark.comf pyspark.comf pyspark.comf pyspark.files pyspark.files pyspark.files pyspark.rakicontext(pyspark.rakicontext(pyspark.union(pyspark.accumulators pyspark.accumulators pyspark.cloudpickie	<pre>pyspark.rdd pyspark.rddsamplex pyspark.rddsamplex pyspark.resultiterable pyspark.seralizers pyspark.sine(pyspark.sine(pyspark.sine(pyspark.sine(pyspark.sine(pyspark.sine(pyspark.sine(pyspark.sine(pyspark.rdd) pyspark.rdd pyspark.rdd pyspark.rdd pyspark.resource</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.taskcontext pyspark.taskcontext pyspark.taskcontext pyspark.urtis	
pyspark.BaicProfiler(pyspark.Croidcast(pyspark.Croidcast(pyspark.Callable(pyspark.AuxContext(pyspark.AuxContext(pyspark.HarshalSerializer(pyspark.Accumulator(pyspark.Accumulator(pyspark.Accumulator(pyspark.Accumulator(pyspark.BarierTaskContext(pyspark.BarierTaskContext(pyspark.BarierTaskContext(pyspark.BarierTaskContext(pyspark.sparkconf(pyspark.sparkcontext(pyspark.sparkcites(pyspark.sparkcites(pyspark.statisTracter(pyspark.st	pyspark.cast(pyspark.cludpickle pyspark.comf pyspark.comf pyspark.comf pyspark.files pyspark.files pyspark.files pyspark.faketorst(pyspark.tpskort pyspark.tpskort pyspark.tpskort pyspark.cast(pyspark.cast(<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.setalizers pyspark.shuffle pyspark.since(pyspark.since(pyspark.since(Lable.thread_target</pre>	pyspark.util pyspark.util pyspark.uraps(pyspark.status pyspark.tatus pyspark.taskcontext pyspark.taskcontext pyspark.taskcontext pyspark.taskcontext	
<pre>pyspark.BaicProfiler(pyspark.Collecter(pyspark.Collecter(pyspark.Collecter(pyspark.Collecter(pyspark.NiveContext(pyspark.NarchalSerialIzer(pyspark.AscshalSerialIzer(pyspark.AscshalSerialIzer(pyspark.Ascrimetator(pyspark.BarrierTaskContext(pyspark.BarrierTaskContext(pyspark.BarrierTaskContext(pyspark.BarrierTaskContext(pyspark.BarrierTaskContext(pyspark.CollectesrialIzer(pyspark.Staberief(pyspark.Staberief)</pre>	pyspark.sparkconf(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkcontext(pyspark.sparkcontext(pys	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.finds pyspark.finds pyspark.finds pyspark.taskContext(pyspark.tpsevar(pyspark.comf pys	<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.sput pyspark.sput pyspark.redsampler pyspark.redsampler pyspark.redsampler pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.taskcontext pyspark.taskcontext pyspark.taskcontext pyspark.urtis	
pyspark. BaicProfiler(pyspark. Froadcast(pyspark. Childlesrializer(pyspark. Childlesrializer(pyspark. Callable(pyspark. Callable(pyspark. TaheritableThread(pyspark. TaheritableThread(pyspark. Accumulator(pyspark. Accumulator(pyspark. AccumulatorParam(pyspark. AccumulatorParam(pyspark. AccumulatorParam(pyspark. BarrierTaskinfo(pyspark. BarrierTaskinfo(pyspark. Callable(pyspark. Callable(pyspark. FiveritableThread(pyspark. FiveritableThread(pyspark. FiveritableThread(pyspark. FiveritableThread(pyspark. FiveritableThread(pyspark.sparkconf(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkcontext(pyspark.sparkcontext(pys	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.finds pyspark.finds pyspark.finds pyspark.taskContext(pyspark.tpsevar(pyspark.comf pys	<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.sput pyspark.sput pyspark.redsampler pyspark.redsampler pyspark.redsampler pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.taskcontext pyspark.taskcontext pyspark.taskcontext pyspark.urtis	
pyspark. BaicProfiler(pyspark. Evadcast(pyspark. CPicitiserializer(pyspark. CPicitiserializer(pyspark. Hyterializerializer(pyspark. Harshalserializer(pyspark. Harshalserializer(pyspark. Accumulator(pyspark. Accumulator(pyspark. Accumulator(pyspark. Accumulator(pyspark. HasicProfiler(pyspark. HasicProfiler(pyspark. HasicProfiler(pyspark. CPicitiserializer(pyspark. cliable(pyspark. cliable(pyspark. cliablef(pyspark. harshalserializer(pyspark. harshalserializer(pyspark.sparkconf(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.sparkcontext(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkstaptof(pyspark.starkcontext(pyspark.sparkcontext(pys	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.finds pyspark.finds pyspark.finds pyspark.taskContext(pyspark.tpsevar(pyspark.comf pys	<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.sput pyspark.sput pyspark.redsampler pyspark.redsampler pyspark.redsampler pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.istontext pyspark.istontext pyspark.istontext pyspark.urtis	
pyspark. BaicProfiler(pyspark. Creditiserializer(pyspark. Creditiserializer(pyspark. Creditiserializer(pyspark. Creditiserializer(pyspark. Accumulator(pyspark. Accumulatorializer(pyspark. Accumulatorializer(pyspark. Accumulatorializer(pyspark. Accumulatoric(pyspark. Bariefrahistoric(pyspark. Bariefrahistoric(pyspark. Bariefrahistoric(pyspark. Harishtiserializer(pyspark. Haniefrahistoric(pyspark. Haniefrahistoric(pyspark. Haniefrahistoric(pyspark. Haniefrahistoric(pyspark. Haniefrahistoricer(pyspark. Haniefrahistoric(pyspark. Haniefrahi	pyspark.sparkConf(pyspark.sparkContext(pyspark.sparkContext(pyspark.sparkContext(pyspark.scatkContext(pys	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.flas pyspark.flas pyspark.flas pyspark.flas pyspark.spark.bome pyspark.spark.comf pyspark.spark.comf pyspark.actumulators pyspark.comf pyspa	<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.shtfle pyspark.sput pyspark.sput pyspark.redsampler pyspark.redsampler pyspark.redsampler pyspark.setalizers pyspark.shtfle pyspark.shtfle pyspark.shtfle</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.istontext pyspark.istontext pyspark.istontext pyspark.urtis	
pyspark. BasicProfiler(pyspark. Evadcast(pyspark. Cricklesprializer(pyspark. Cricklesprializer(pyspark. Cricklesprializer(pyspark. Lineritabiethermad(pyspark. Harshalserializer(pyspark. Ascendulator(pyspark. Accumulator/arma(pyspark. Accumulator/arma(pyspark. Accumulator/arma(pyspark. Asciellator(pyspark. MarierTasicContext(pyspark. BasicProfiler(pyspark. BasicProfiler(pyspark. Cricklesprializer(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Taberitabiethermad(pyspark. Rarshalserializer(Contexpark. BasicProfiler(pyspark. Rarshalserializer(pyspark. Rarshalserializer(pyspark. Rarshalserializer(Contexpark. Rarsha	pyspark.sparkConf(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkConfect pyspark.sparkConfect pyspark.scalasteret(pyspark.scalasteret(pyspark.scalasteret(pyspark.scalaster(p	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.flad pyspark.flad pyspark.tpark.bose pyspark.somf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.flad pyspark.flad pyspark.flad pyspark.flad	<pre>pyspark.redsample: pyspark.resource pyspark.resource pyspark.selalizers pyspark.selalizers pyspark.selalizers pyspark.sel pyspark.selalizers pyspark.selalizers pyspark.redsampler pyspark.resource pyspark.selalizers pyspark.selalizers pyspark.since(pyspark.since(pyspark.since)</pre>	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.istontext pyspark.istontext pyspark.istontext pyspark.urtis	
pyspark. BasicProfiler(pyspark. Evadcast(pyspark. Kilable(pyspark. Kilable(pyspark. Kilable(pyspark. HiverGontext(pyspark. Ascshalserializer(pyspark. Ascshalserializer(pyspark. Ascshalserializer(pyspark. BasicProfiler(pyspark. BasicProfiler(pyspark. BasicProfiler(pyspark. BasicProfiler(pyspark. BasicProfiler(pyspark. SasicProfiler(pyspark. SasicProf	pyspark.sparkConf(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkConfect pyspark.sparkConfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sparkConfect pyspark.sparkConf(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkConfect pyspark.sparkConfect pyspark.starkSparkConfect pyspa	pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.flad pyspark.flad pyspark.tpark.bose pyspark.somf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.comf pyspark.flad pyspark.flad pyspark.flad pyspark.flad	pyspark.reduiterable pyspark.resultiterable pyspark.selatizerable pyspark.selatizerable pyspark.selatizerable pyspark.selatizerable pyspark.john pyspark.john pyspark.selatizerable pyspark.reduiterable pyspark.resultiterable pyspark.selatizers pyspark.selatizer	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.istontext pyspark.istontext pyspark.istontext pyspark.urtis	
pyspark. BasicProfiler(pyspark. Evalcast(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. ilable(pyspark. for the ilable ilable) pyspark. Arcimilator(pyspark. accimilator(pyspark. accimilator(pyspark. accimilator(pyspark. accimilator(pyspark. accimilator(pyspark. accimilator(pyspark. ispection) pyspark. ispection) pyspark. serierTaskContext(pyspark. ispection) pyspark. serierTaskContext(pyspark. ispection) pyspark. serierTaskContext(pyspark. ispection) pyspark. serierTaskContext(pyspark. seri	pyspark.sparkConf(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkConfect pyspark.sparkConfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sconfect pyspark.sparkConfect pyspark.sparkConf(pyspark.sparkConf(pyspark.sparkConfect pyspark.sparkConfect pyspark.sparkConfect pyspark.starkSparkConfect pyspa	pyspark.cast(pyspark.cludpickle pyspark.context pyspark.context pyspark.context(pyspark.find_spark_home pyspark.find_spark_home pyspark.staticontext(pyspark.staticontext(pyspark.context) pyspark.context pyspark.files pyspark.f	pyspark.reduiterable pyspark.resultiterable pyspark.selatizerable pyspark.selatizerable pyspark.selatizerable pyspark.selatizerable pyspark.john pyspark.john pyspark.selatizerable pyspark.reduiterable pyspark.resultiterable pyspark.selatizers pyspark.selatizer	pyspark.urti pyspark.ursion pyspark.uraps(pyspark.status pyspark.istatus pyspark.istontext pyspark.istontext pyspark.istontext pyspark.urtis	

Table 107: Validation Test T02_WP6T62-01_DI (overview)

Copyright © FRACTAL Project Consortium	135 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Validation test			
Test ID	T03_WP6T62-01_DI		
Test type	Functional-Installation		
Test name	Installation and configuration of Faust		
Date	16/02/2023		
Tester's Name	Nicola Alchera (Rulex)		
Test scope or object	tive		
The objective of this	s test is to validate the Faust installation guidelines.		
Steps	Steps		
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)		
Step 2	install Faust and run python3		
Results/Evidence			
Faust has been con	rectly installed and it can be used as python3 library		
Success criteria			
No error messages/	All partial results are as expected		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion		
Test configuration Test conditions	https://github.com/project-fractal/WP6T62-01-data-ingestion The tests have been performed on a node with Ubuntu 22.04.		
<u> </u>			
Test conditions	The tests have been performed on a node with Ubuntu 22.04.		

Table 108: Validation Test T03 WP6T62-01_DI (overview)

Copyright © FRACTAL Project Consortium	136 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test		
Test ID	T04_WP6T62-01_DI		
Test type	Functional-Installation		
Test name	Installation of RedNote		
Date	16/02/2023		
Tester's Name	Nicola Alchera (Rulex)		
Test scope or objectiv	/e		
The objective of thi	s test is to validate the RedNote installation guidelines.		
Steps			
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)		
Step 2	Install Docker		
Step 3	Run the docker		
Step 4	Open the docker in browser		
Results/Evidence			
	ccessfully opened and it is ready to use		
Fight an party latest			
	installed without major errors		
	······································		
Test observations			
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion		
Test conditions	The tests have been performed on a node with Ubuntu 22.04.		
Remarks	Check you have started this container with a volume mounted on /data. I checked, and it seems the container is monted on /data, as you can see in the image below. Despite this, the workflow is not saved . "Mounts": [{ "Type": "volume", "Name": "node_red_data", "Source": "/var/lib/docker/volumes/node_red_data/_data", "Destination": "/data", "Destination": "/data", "Driver": "local", "Mode": "z", "RW": true, "Propagation": ""],		
Tost result			
Test result			
Passed			

Table 109: Validation Test T04_WP6T62-01_DI (overview)

Copyright © FRACTAL Project Consortium	137 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Validation test			
Test ID	T05_WP6T62-01_DI		
Test type	Functional-Installation		
Test name	test of MQTT Cloud Communications Component		
Date	03/03/23		
Tester's Name	Nicola Alchera (Rulex)		
Test scope or objective			
The objective of this test is to validate the MQTT Cloud Communication guidelines.			
Steps			
Step 1	Prepare a node with Ubuntu and required dependencies (python 3)		
Step 2	Set up a container with an MQTT broker		
Step 3	Test the publish functionalities		
Step 4	Test the subscribe functionalities		
Results/Evidence			

Both the publisher and subscriber script works correctly.

	and a state of the
icolaalchere@niexal@ubur X 🗴 nicolaalchera@niexal@ubun X + 🗵	<pre>nicola.alchera@rulex.ai@ubuntu-l-vm:-/WP6T62-01-data-ingestion/mosquitto/test- ibe.py</pre>
WadedcSb@edcb43437aecb611027dca ored in directory: /home/nicola.althera/.cache/pip/wheels/6a/48/01/cB90 Hssfully built caho-mutt	Connected to MOTT Broker! Received messages: 35° from 'fractal/mott-test' topic Received 'messages: 35° from 'fractal/mott-test' topic Received 'messages: 37° from 'fractal/mott-test' topic
saruty built panomete lling collected packaps: paho-mqtt ssfully installed paho-mqtt-1.6.1 a.alcheragruler.aigubuntu-1-wn=/MP6762-01-data-ingestion/mosquitto/te a.alcheragruler.aigubuntu-1-wn=/MP6762-01-data-ingestion/mosquitto/te	Received 'messages: 38' from 'fractal/entit-test' topic Received 'messages: 30' from 'fractal/entit-test' topic Received 'messages: 40' from 'fractal/entit-test' topic Received 'messages: 40' from 'fractal/entit-test' topic
<pre>icted to MQTT Broker! 'messages: 0' to topic 'fractal/mqtt-test' 'messages: 1' to topic 'fractal/mqtt-test' 'messages: 2' to topic 'fractal/mqtt-test' 'messages: 2' to topic 'fractal/mqtt-test'</pre>	Received 'messages: 42' from 'fractal/eqtt-test' topic Received 'messages: 43' from 'fractal/eqtt-test' topic Received 'messages: 44' from 'fractal/eqtt-test' topic Received 'messages: 55' from 'fractal/eqtt-test' topic
<pre>mosages: 0 to topic 'ractal/mgtt-test' messages: 0' to topic 'ractal/mgtt-test' </pre>	Received messages: 40° from 'fractal/mptt-test' topic Received messages: 40° from 'fractal/mptt-test' topic
TRECEMENT A TO LODIC TREETATIONTOLIST	neczived "essages: 30 from "fractal/mqtt-test" topic Reczived "essages: 51 from "fractal/mqtt-test" topic Reczived "essages: 52 from "fractal/mqtt-test" topic Reczived "essages: 53 from "fractal/mqtt-test" topic

Success criteria	
No error messages/	All partial results are as expected
Test observations	
Test configuration	https://github.com/project-fractal/WP6T62-01-data-ingestion
Test conditions	The tests have been performed on a node with Ubuntu 22.04.
	The clone command "git clone https://github.com/project-fractal/WP6T62-01-data-ingestion.git"
	could not work properly. The user password authentication is no longer supported: a valid token
	is required for access. If you haven't any token, you have create a new one following the
	instructions you can fine here https://docs.github.com/en/get-started/getting-started-with-
	git/about-remote-repositories#cloning-with-https-urls
Remarks	
	The command "source run.sh" could give you an error. To avoid the error you shold give the
	permission through the following command "sudo chmod 777 /var/run/docker.sock"
	There is an error in guideline: guidlines says to test two times the same python script (test-
	publish.py) and not the "test-publish.py" and the "test-subscribe.py"
Test result	
Passed	

Table 110: Validation Test T05_WP6T62-01_DI (overview)

Copyright © FRACTAL Project Consortium	138 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.5 Federated Data Collection component complete templates

Validation test				
Test ID	T01_WP6T62-02_FDC			
Test type	Functional-Installation			
Test name	nstallation of CrateDB			
Date	16/01/2023			
Tester's Name	Nicola Alchera (Rulex)			
Test scope or objective	e			
The objective of this te	est is to validate the CrateDB installation guidelines.			
Steps				
Step 1	Prepare a node with Ubuntu and required dependencies			
Step 2	Install CrateDB on Ubuntu			
Step 3	Run the docker and open CrateDB in a web browser			
Step 4	Create and view a table using SQL into Crate DB			
Results/Evidence				
CrateDB has been succ	cessfully opened in browser and it is ready to use			
CreteO Cluster: crate Cluster: crate Cluster load Cluste	V fact to SQL definement to party Catality If hower to sails If the sail to SQL definement to party Catality National data National data If the sail to SQL definement to party Catality If the sail to SQL definement to party Catality National data National data If the sail to SQL definement to party Catality If the sail to SQL definement to party Catality National data National data National data If the sail to SQL definement to party Catality National data National data National data If the sail to SQL definement to party Catality National data National data If the sail to SQL definement to party Catality If the sail to SQL definement to party Catality National data National data If the sail to SQL definement to party Catality If the sail to SQL definement to party Catality National data National data National data If the sail to SQL definement to party Catality National data National data National data If the sail to SQL definement to party Catality National data National data National data If the sail to SQL definement to party National data National data If the sail to SQL definement			
Success criteria Data can be created and viewed.				
Test observations	u vieweu.			
	https://github.com/project-fractal/WP6T62-02-federated_data_collection			
Test configuration				
Test conditions				
Remarks No issues were highlighted during the test				

Table 111: Validation Test T01_WP6T62-02_FDC (overview)

Test result Passed

Copyright © FRACTAL Project Consortium	139 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

	Validation test		
Test ID	T02_WP6T62-02_FDC		
Test type	Functional-Installation		
Test name	Installation of MongoDB		
Date	24/01/2023		
Tester's Name	Nicola Alchera (Rulex)		
Test scope or object	ctive		
The objective of thi	s test is to validate the MongoDB installation guidelines.		
Steps			
Step 1	Prepare a node with Ubuntu and required dependencies		
Step 2	Install Mongo on Ubuntu		
Step 3	Run the docker		
Results/Evidence			
Loaded: l Active: a	ce - MongoDB Database Server oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: /	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod)		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: /	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service 563824 /usr/bin/mongodconfig /etc/mongod.conf		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service 563824 /usr/bin/mongodconfig /etc/mongod.conf		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2 Success criteria Data can be created	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service -563824 /usr/bin/mongodconfig /etc/mongod.conf 1 ubuntu-l-vm systemd[1]: Started MongoDB Database Server.		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2 Success criteria Data can be created Test observations	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service -563824 /usr/bin/mongodconfig /etc/mongod.conf 1 ubuntu-l-vm systemd[1]: Started MongoDB Database Server.		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2 Success criteria Data can be created Test observations Test configuration	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service -563824 /usr/bin/mongodconfig /etc/mongod.conf 1 ubuntu-l-vm systemd[1]: Started MongoDB Database Server.		
Loaded: 1 Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2 Success criteria Data can be created Test observations Test configuration Test conditions	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service -563824 /usr/bin/mongodconfig /etc/mongod.conf 1 ubuntu-l-vm systemd[1]: Started MongoDB Database Server. d and viewed via MongoDB <u>https://github.com/project-fractal/WP6T62-02-federated data collection</u>		
Loaded: l Active: a Docs: h Main PID: 5 Memory: 1 CGroup: / Jan 24 11:18:2	oaded (/lib/systemd/system/mongod.service; disabled; vendor preset: enabled) ctive (running) since Tue 2023-01-24 11:18:21 UTC; 10s ago ttps://docs.mongodb.org/manual 63824 (mongod) 68.3M system.slice/mongod.service 563824 /usr/bin/mongodconfig /etc/mongod.conf 1 ubuntu-l-vm systemd[1]: Started MongoDB Database Server. d and viewed via MongoDB <u>https://github.com/project-fractal/WP6T62-02-federated data collection</u> The tests have been performed on a node with Ubuntu 22.04.		

Table 112: Validation Test T02_WP6T62-02_FDC (overview)

Copyright © FRACTAL Project Consortium	140 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.6 Low End Node component complete templates

	Validation Test T01 WP6T62-06		
LOL LVDC	Functional		
	Testing the connection between the Device and the Cloud Platform		
ester's Name Luca Visconti (Akkodis)			
Test scope or objective			
The objective of this test	is to validate the connection between Low End Node Device and Cloud Platform.		
Steps			
Step 1	Power up the device		
Step 2	Connect the device to internet		
Results/Evidence			
low-end-lzjrf 3n21s low-end-ms6pw 3n21s low-end-ms6py 3n21s low-end-ms6py 3n1s ubuntu@naster:-\$	FFactal SFB05884F4D06 Degrée Peopted Nexce O C ABEB3F731BD00D		
fractal-node- A62253BE425CE8893CA Connected fractal-node- 806317DC2DFED29E09D Disconnected	Depind Reported Masure Elemone D @		
Success criteria Device accepted and dev	vice status "connected"		
Test observations			
Test configuration	Device access point config: SSID and PW		
-	Device connected to local wifi		
Remarks			
Test result			

Table 113 - Validation Test T01_WP6T62-06 (overview)

Copyright © FRACTAL Project Consortium	141 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

		Validation Test		
Test ID		T02_WP6T62-06		
Test typ	e	 Functional		
Test nar		Testing the communication from Device to the Cloud		
Date		19/01/2022		
Tester's	Name	Luca Visconti (Akkodis)		
Test sco	pe or objective			
The obje	ective of this test	is to validate communication between Low End Node Device and Cloud Platform.		
Steps				
Step 1		Power up the device		
Step 2		Connect the device to internet		
Step 3		Change device status by pressing button		
Results/	Evidence			
Step 2: Step 3:	Connected fractal-node- 806317DC2DFED29E03 Connected fractal-node- 7AEDFBE5EF120C2F7BF	Desired Reported Messure @ Remove 20 67		
Success Reporte	criteria d status as expe	cted		
	ervations			
	figuration	Device access point config: SSID and PW		
Test con	-	Device access point coming. SSID and PW Device connected to local wifi		
Remarks				
	Remarks Test result			
Passed				
rassed				

Table 114 - Validation Test T02_WP6T62-06 (overview)

Copyright © FRACTAL Project Consortium	142 of 149

FRACTAL	Project	FRACTAL
	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

Validation Test Test ID T03_WP6T62-06	
Test type Functional	
Test name Testing the communication from the Cloud Platfo	rm to the Device
Date 19/01/2022	
Tester's Name Luca Visconti (Akkodis)	
Test scope or objective	
The objective of this test is to validate the communication between Cloud	Platform and Low End Node Device.
Steps	
Step 1 Power up the device	
Step 2 Connect the device to internet	
Step 3 Change status in Kubernetes by "patch" method	
ubunt@master:~\$ ubunt@master:~\$ v{ "spec": { "desiredState": { "state": 1} }}' lowend.fractal-cluster.eu/low-end-5r2gf patched	-cluster.eu low-end-5r2gftype mergepatch
Results/Evidence	
Step 2: Connected fractal-node- 806317DC2DFED29E09D5DF353287AFE8 Desired Period Reported Period Remove D B Step 3: Spec: Connection Connected Desired State: Last Updated: 2023-02-20T07:33:00.9524174Z State: Last Updated: 1 Device Id: fractal-node- 806317DC2DFED29E09D5DF353287AFE8 Fractal-node-806317DC2DFED29E09D5DF353287AFE8 Reported State: 0 0 Normal 2023-02-20T07:39:29.6870527Z State: 0 Normal Logging 275 kopf Updatedir 'update_fn' succeeded; 0 failed. Connected fractal-node- 806317DC2DFED29E09D5DF353287AFE8 Period Reported To connected Connected fractal-node- 806317DC2DFED29E09D5DF353287AFE8 Reported Reported To connected Connected	
Success criteria	
Device accepted and device status "connected"	
Test observations	
Test configuration Device access point config: SSID and PW	
Test conditions Device connected to local wifi	
Remarks	
Test result	
Passed	

Table 115 - Validation Test T03_WP6T62-06 (overview)

Copyright © FRACTAL Project Consortium	143 of 149

1999292	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

		Validation Test
Test ID		T04 WP6T62-06
Test type		Functional
Test name	2	Testing the Tasks Scheduling running Nuttx on the Device
Date	-	19/01/2022
Tester's N	ame	Luca Visconti (Akkodis)
Test scope	e or objective	
		t is to validate the task scheduling on the Low End Node
Steps		
Step 1		Power up the device
Step 2		Connect the device by usb to a pc
Step 3		Open terminal connection and run command "ps"
		\$ minicom -D /dev/ttyUSB1
		offcode>
		offcode> ps
Results/Ev	vidence	
	[598658.338481] bcdDevice= [598658.338492] SerialNumber=3 [598658.338501] [598658.338501] [598658.338501] [598658.345056] [598658.345156] offcode> offcode> ps	usb 3-2.1.4: SerialNumber: 363aaf4f44a0eb11a2a8cbacdf749906
	0 0 0 0 FIFO K 1 1 224 RR 4 2 224 RR 4 3 3 100 RR 4 4 4 100 RR 4 5 5 100 RR 7 6 6 223 RR 6 8 8 100 RR 7 0.pool.ntp.org;1.pool. 10 10 100 RR 11 10 100 RR 0x3fcb0bd0	Chi The INPASIATE EVENT SIGNASS STACK COMMAND Kitnead V Ready 0000000 004048 Idle Task Kitnead V Waiting Semaphore 00000000 004016 hpwork 0x3fc844d8 Kitnead Waiting Semaphore 0000000 004016 lpwork 0x3fc844dc Task Running 0000000 001968 nsh_main Kitnead Waiting Semaphore 0000000 001968 nsh_main Kitnead Waiting Semaphore 0000000 001968 nsh_main Kitnead Waiting Signal 00000000 001968 nsh_main Kitnead Waiting Signal 00000000 001968 nsh_main Task Runing 0000000 001968 nsh_main Kitnead Waiting Signal 00000000 001968 nsh_main Task Waiting Signal 00000000 001968 nsh_main Task Waiting Signal 00000000 0016320 fractal pthread Waiting Signal 00000000 016320 fractal pthread Waiting Signal 00000000 003024 pt-0x420347ee
Success cr	iteria	
Print all pa	arallel running	task as expected
Test obser	rvations	
Test config	guration	GitHub repository: https://github.com/project-fractal/WP6T62-06-low-end-node- orchestrator
Test "		Device access point config: SSID and PW
Test condi	itions	Device connected to local wifi
Remarks	•	
Test result	τ	
Passed		

Table 116 - Validation Test T04_WP6T62-06 (overview)

Copyright © FRACTAL Project Consortium	144 of 149

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
	Del. Code	D6.4

14.7 Hardware-level Edge Controller component complete templates

Fest ID	T01_WP6T62-0X	Validation Te		
est type	Functional			
		e-Classification an	d Message-Schedu	ling Services at different port
est name		luling configuration		o
Date	03/02/2022			
ester's Name	Luca Visconti (Akko	odis)		
est scope or object				
		nessage-classificati	on and message-sc	cheduling services sending 3
	ent port and according t			- 01
iteps				
itep 1				rent port according to the
		iration TTCommSch	neld.cfg of each NI)	
	- 1 message to NIO			
	Port ID	Instant	Next	Instant (microsec)
	2 [NI0-NI2]	956 (7)	0	28,483
	2	1	1.2	
		1 on the port 2 and		Instant (missons)
	Port ID	Instant	Next	Instant (microsec)
	2 [NI1-NI0]	1092 (8)	-	32,552
	3 [NI1-NI3]	2594 (19)	0	77,311
tep 2	Send a message to			
	<pre>// send message fr for(int i=0;i<mesg< pre=""></mesg<></pre>	com NIO on port 2		
	(// Marker Manager on MTA
)			// Write Message on NIO, <u>port2</u>
	Xil_Out32 (XPAR_SKI	NOC_0_S00_AXI_BASEA	DDR+163840,7); // T	erminate the Message
	Send messages to			
		from NI1 on port 2		
	<pre>for(int i=0;i<mes pre="" {<=""></mes></pre>	g_size;i++)		
	Xil_Out32 (XPA	R_SKINOC_0_S01_AXI_	BASEADDR+131072, i);	// Write Message on NIO, <u>port</u>
	Xil_Out32 (XPAR_SK	INOC_0_S01_AXI_BASE	EADDR+163840,7); //	/ Terminate the Message
	// send messages	from NI1 on port 3		
	<pre>for(int i=0;i<mes< pre=""></mes<></pre>			
	(
		P SKINOC 0 PO1 P		
	}			// Write Message on NIO, <u>port</u>
	}			; // Write Message on NIO, <u>port</u> / Terminate the Message
Rosults / Evidence	}			
	}			
Results/Evidence Step 2: KELEASE 2021.1 Feb) Xil_Out32(XPAR_SK	NNOC_0_S01_AXI_BASE	EADDR+229376,7); //	/ Terminate the Message
Step 2: Release 2021.1 Feb) Xil_Out32(XPAR_SK	INOC_0_S01_AXI_BASE	e difference betweer	/ Terminate the Message
Step 2: Release 2021.1 Feb PMU-FW is not running NIO - NI2 , recived NIO - NI2 , recived) Xil_Out32(XPAR_SK 7/2023 - 07:42:21 , certain applications nay no data=0, original_data = 0 data=1, original_data = 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass	EADDR+229376,7); // e difference betweer e difference betweer	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second
Release 2021.1 Feb PHU-FW is not running NIO - NI2 , recived NIO - NI2 , recived) Xil_Out32(XPAR_SK 7/2023 - 07:42:21 , certain applications nay no data=0, original_data = 0 data=1, original_data = 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass	EADDR+229376,7); // e difference betweer e difference betweer	/ Terminate the Message
Release 2021.1 Feb PHU-FW is not running NIO - NI2 , recived NIO - NI2 , recived) Xil_Out32(XPAR_SK 7/2023 - 07:42:21 , certain applications nay no data=0, original_data = 0 data=1, original_data = 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time Pass	addR+229376,7); // e difference between e difference between difference between	/ Terminate the Message m m2, m1 =4 micro second m m3, m2 =44 micro second m m3, m1 =48 micro second
Step 2: Release 20211 Feb PULFR LS not running For the state of	2023 - 0/14721 2023 - 0/14721 2023 - 0/14721 2024 - 0rightal_data = 0 data-1, orightal_data = 1 data-2, orightal_data = 1 data-4, orightal_data = 1 data-4, orightal_data = 4 data-4, orightal_data = 4 data-4, orightal_data = 5 data-4, orightal_data = 7 data-6, orightal_data = 7 d	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Release 20211 Feb PULFR LS not running For the state of) 1. Out32 (XPAR_SK 1. Out32 (INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message m m2, m1 =4 micro second m m3, m2 =44 micro second m m3, m1 =48 micro second
Step 2: Release 2021.1 PHU-FNL EN ort running NIO - NIZ , rectived) 1. Out32 (XPAR_SK 1. Out32 (INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Xeless 2021.1 Feb WH FW 15 15 not running Feb WH FW 15 not running) 11_outJ2 (XPAF_SK 2601 - 0/40214 10_000 - 0/51014 dets #0 0/4000 - 0/5100 - 0/51000 - 0/51000 - 0/510000000 - 0/510000000000000000000000000000	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Xeless 2021.1 Feb WH FW 15 15 not running Feb WH FW 15 not running) 11_outJ2 (XPAF_SK 2601 - 0/40214 10_000 - 0/51014 dets #0 0/4000 - 0/5100 - 0/51000 - 0/51000 - 0/510000000 - 0/510000000000000000000000000000	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: Release 2011.1 Feb PHUFM 15 and running WH 5M 15 and running WH 5M 12, restwa N10 N12, restwa) 11_outJ2 (XPAF_SK 2601 - 0/40214 10_000 - 0/51014 dets #0 0/4000 - 0/5100 - 0/51000 - 0/51000 - 0/510000000 - 0/510000000000000000000000000000	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: rest file 5 001.1 rest file 5 001.1 rest rest 001.1	Jil_outJ2 (XPAF_SK Jil_outJ2 (XP	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: respect 620111 feb respective 620111) 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: respect 620111 feb respective 620111) 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: respect 620111 feb respective 620111) 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: respect 620111 feb respective 620111) 1	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: Plane 4011 Teb Plane 4011 Teb	, 211_0122(XPAG_SR , 2233 - 074212 , 2233 - 074212 , 2234 - 074212 , 2334 - 074212 , 3	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
tep 2: Plane 4011 Teb Plane 4011 Teb	, 211_0122(XPAG_SR , 2233 - 074212 , 2233 - 074212 , 2234 - 074212 , 2334 - 074212 , 3	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: PREFACE \$021.1	Jil_outJ2 (KPAL_SR J. cristal, asplications are detail, orginal, data = 0 datasi, orginal, data	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: PREFRE 00111 Feb prefre 00111	Jil_outJ2 (KPAL_SR J. cristal, asplications are detail, orginal, data = 0 datasi, orginal, data	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle 691.1	1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle 4 to 7 rounds	Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jilou	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle 4 to 7 rounds	Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jilou	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle 4 to 7 rounds	1 11_0ut32 (KPAL_SR 1, crs1a ap31c1utss symo deasa , original, deas = 0 deasa , original, deas = 1 deasa , ori	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Participation of the foot running for the foot running foot of the foot running foot of the foo	Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR JiloutJ2(KPAL_SR Jilout	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Store 2: Particle 401.1 February 100 Partin 401.1 Febr	Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR Jil_outJ2(KPAL_SR JiloutJ2(KPAL_SR Jilout	INOC_0_S01_AXI_BASE t be supported. Time Pass Time Pass Time Pass Time To In	e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
<pre>step 2:</pre>	111_OUT22 (KPAK_SR 7262) - 0/14211 Anten - original data = 0 data: - ori	INC_0_SOI_AXI_BASE	cadDR+229376,7); /, e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
<pre>step 2:</pre>	1, 2013 - 07-02-01 1, 2014 - 07-02-01 1, 201	INC_0_SOI_AXI_BASE	cadDR+229376,7); /, e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particular (2011) Feb 1011) Partin (2011) Feb	1 11_0ut32 (KPAL_SR 1, cort32 (KPAL_SR 1, cort32 (KPAL_SR 1, cort34 , co	INC_0_SOI_AXI_BASE	cadDR+229376,7); /, e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle 2011 Feb 1 Particle 2011 Feb 1 Particle 2011	111_out32 (KPAL_SR 7203) - 014212 fatter of original data and data of original data and data of original data and data of original data and data of original data of data of original data of da	INC_0_SOI_AXI_BASE	cadDR+229376,7); /, e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Paters (921.1 Feb mit 5 Ni2 C (2011) Paters (921.1 Feb mit 5 Ni2 C (2014) Paters (921.1 Feb mit 5 Ni2 C (2014) Ni 5 Ni2 C (2014)	1 11_0ut32 (KPAL_SR 1, cort32 (KPAL_SR 1, cort32 (KPAL_SR 1, cort34 , co	INC_0_SOI_AXI_BASE	cadDR+229376,7); /, e difference between e difference between e difference between e difference between sction time of m =	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Step 2: Particle (2011) Feb 1 Particle (2011)	Jill_out32 (KPAL_SR J. Cot3 - 015121 J. Cot3 - 01512 J. Cot3 -	t be supported.	2ADDR+223376,7); /,	/ Terminate the Message n m2, m1 =4 micro second n m3, m2 =44 micro second n m3, m1 =48 micro second
Site 2: Particle 2021.1 Feb particle 2021.1 Particle 2021.2 Feb particle 2021.1 Particle 2021.1	Jil_out32 (KPAL_SR Jil_out32 (KPAL_SR Jiliout32 (KPAL_SR Jiliout33 (KPAL_SR Jiliou	t be supported. The support of the s	<pre>2ADDR+229376,7); /, e difference between difference between difference between colon time of n1 = colon time of n3 = 2 colon time of n3 = 2</pre>	/ Torminate the Message
Step 2: Paters (921.1 Feb mit 5 Ni2 C (2011) Paters (921.1 Feb mit 5 Ni2 C (2014) Paters (921.1 Feb mit 5 Ni2 C (2014) Ni 5 Ni2 C (2014)	1 211_0ut32 (KRAL_SR 1 213_0 0.00121 1 2013_0 0.00121 1 2013_0 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 <td>t be supported. The Past Take The Take Take Take Take Take Take Take Tak</td> <td><pre>zaDOR+229376,7); // c difference betweer difference betweer difference betweer ction time of n1 = ction time of n3 = ction time of n3 = ction time of n3 = </pre></td> <td>/ Terminate the Message</td>	t be supported. The Past Take The Take Take Take Take Take Take Take Tak	<pre>zaDOR+229376,7); // c difference betweer difference betweer difference betweer ction time of n1 = ction time of n3 = ction time of n3 = ction time of n3 = </pre>	/ Terminate the Message
Site 2: Particle 2021.1 Feb particle 2021.1 Particle 2021.2 Feb particle 2021.1 Particle 2021.1	1 211_0ut32 (KRAL_SR 1 213_0 0.00121 1 2013_0 0.00121 1 2013_0 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 2 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 2 0.00121 0.00121 <td>t be supported. The Past Take The Take Take Take Take Take Take Take Tak</td> <td><pre>zaDOR+229376,7); // c difference betweer difference betweer difference betweer ction time of n1 = ction time of n3 = ction time of n3 = ction time of n3 = </pre></td> <td>/ Terminate the Message</td>	t be supported. The Past Take The Take Take Take Take Take Take Take Tak	<pre>zaDOR+229376,7); // c difference betweer difference betweer difference betweer ction time of n1 = ction time of n3 = ction time of n3 = ction time of n3 = </pre>	/ Terminate the Message

Table 117 - Validation Test T01_WP6T62-0X (overview)

Copyright © FRACTAL Project Consortium	145 of 149
--	------------



Project	FRACTAL
Title	FRACTAL engineering framework validation

Del. Code D6.4

Test ID	T02_WP6T62-0X	Validation Tes		
Test type	Functional			
Test name		Classification and	Message-Schedul	ing Services at same port
	according to schedul			
Date	03/02/2022			
Tester's Name	Luca Visconti (Akkod	is)		
Test scope or objective The objective of this te		ssage-classificatio	n and message.cot	eduling services sending
	according to scheduling		and message-Su	cooming services serioring
Steps				
Step 1	Define a set of messa	ages to send (3 m	essages at 3 from t	he same port according to
	the scheduling config		Scheld.cfg of NIO)	
	- 3 messages to NIO o			
	Port ID	Instant	Next	Instant (microsec)
	2 [NI0-NI2]	956 (7)	0	28,483
Step 2	Send messages to NI // send messages	1: from NII-NIO using pg g_size;i++) // Messag	at2	
	(seaddr+131072,i); // p	ecimale (02) 0000
	1		DR+163840,7); // termi	
	/*			*/
	<pre>// send message f for(int i=0;i<mes< pre=""></mes<></pre>	rom NI1-NI3 g_size;i++) // Messag	e size = 5	
	I more presented and the second se	r_skinoc_0_s01_axi_ba		
	Xil_Out32 (XPAR_SK	INOC_0_S01_AXI_BASEAD	DR+229376,7); // termi	nate Message
	/*			*/
	<pre>// send messages for(int i=0;i<mes i="0;i<mes)</pre" int=""></mes></pre>	from NII-NIO. g_size;i++) // Messag	e size = S	
	1	R_SKINOC_0_S01_AXI_BA		
	Xil_Out32 (XPAR_SK	INOC_0_S01_AXI_BASEAD	DR+163840,7); // termi	nate Message
	/*	from NI1-NI3		
	for(int i=0;i <m< td=""><td>esg_size;i++) // Moss</td><td></td><td></td></m<>	esg_size;i++) // Moss		
	}	R_SKINOC_0_S01_AXI_BA	SEADDR+196608,i); ADDR+229376,7); // ter	minate Messarr
	X11_0UE32 (XPAR_	SKINGC_0_S01_AXI_BASE	ADDR+229376,7)1 77 ter	minate Message
	// send messag	es from NI1-NI0.		
	. (mesg_size;i++) // Mes AR_SKINOC_0_S01_AXI_E		
	}			
		AR_SKINOC_0_S01_AXI_E	ASEADDR+163840,7); //	terminate Message
	/*	AR_SKINOC_0_S01_AXI_E	ASEADDR+163840,7); //	terminate Message +/
	/*	from NI1-NI3		terminate Message */
	for(int i=0;i <m< th=""><th></th><th>age size = 5</th><th>terminate Message •/</th></m<>		age size = 5	terminate Message •/
	for (int i=0/i <m (xil_Out32 (XPA</m 	from NII-NI3 esg_size;i++) // Mess R_SKINOC_0_S01_AXI_BA	age size = 5	•/
Results/Evidence Step 2:	for(int i=0/icm (from NII-NI3 esg_size/i++) // Mess R_SKINOC_0_S01_AXI_BAS SKINOC_0_S01_AXI_BASE	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	minate Message
Step 2: NII NI3, rect NII NI3, rect	ed data-0 , original data ed data-1 , original data	from NII-NI3 esg_size/i++) // Mess R_SKINOC_0_S01_AXI_BAS SKINOC_0_S01_AXI_BASE	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	•/
Step 2: 	for(int_i=0)idm xil_out32(XFAM xil_out32(XFAM xil_out32(XFAM cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data cd_data; ortginal_data	from B11-WE2 ge_size(i=) // Hose B_BETHOC_0_B01_AXT_BAR BETHOC_0_B01_AXT_BAR BETHOC_0_B01_AXT_BAR The differ 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: 11 N3, rect 11 N3, re	ed data-0 , original, data ed data-0 , original,	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: 111 N33, rect 111 N33, rect	reg (data) or (gana) data wil out 3 (SAN) ki out 3 (SAN)	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2:	red data a program da	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: 11 N3 , rect 11 N3 ,	for (int int) is into its into	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: II NI3 rect NI = NI3 rect NI = NI3 re	ddtame o riginal data ddtame o riginal data	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: 11 N13 , rect N13 N143 , rect N15 rect N13 N13 , rect N13 rect N13 rect N13 N13 , rect N13 rect	ddtame o riginal data ddtame o riginal data	from H11-H12 ge_size(i+) // Hease R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR R_HTHOC_0_601_AXT_BAR 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	age size = 5 SEADDR+196608,i); ADDR+229376,7); // ter	*/
Step 2: III NI3 rect NII NII NI3 rect NII NI3 rect NII NII NI3 rect NII NI3 rect NII NII NI3 rect NII NI3 rect NII NI3 NII NI3 rect NII NI3 rect	d data - o'iginal data - o'igi	from B11-813 g_size14-91 // Hesse m_string	ngo mino = 5 HERADRA-16600,1); ABDRA-120976,7); // ter	*/
Step 2: III NI3 rect NI = NI3 rect NI = NI3 r	for that levins xii_out32 (KBA xii_out32 (KB	from B11-M12 grant and a set of the set of	ngo mino = 5 HERADOR+10600, £); ADDR+210976, 7); // ter Proce between two contexe mee between two contexe ince between two contexe "hww.cfg"	*/
Step 2: III NI3 rect NII NII NI3 rect NII NI3 rect NII NII NI3 rect NII NI3 rect NII NI3 NII NII NII NIII NIII NIII NIII NIII NIII NIIII NIIII NIIIII NIII NIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	for that levins xii_out32 (KBA xii_out32 (KB	from B11-912 g_stize14*) R_BETHOD_0_B01_AKT_BAR RETHOD_0_B01_AKT_BAR RETHOD_0_B01_AKT_BAR RETHOD_0_B01_AKT_BAR RETHOD_0_B01_AKT_BAR 0 0 0 0 0 0 0 0 0 0 0 0 0	nge size = 5 stratore.iscon.iy; Addre.izoifd.rj; // ter ence between two consec more between two consec "hw.cfg" unning on the boa	<pre>xiLasto Message xuttive messages m2 and m1 = 122 xuttive messages m2 and m2 = 122 xuttive mess</pre>

Table 118 - Validation Test T02_WP6T62-0X (overview)

Copyright © FRACTAL Project Consortium	146 of 149



Del. Code D6.4

		Validation T	est	
Fest ID	T03_WP6T62-0X	(
Test type	Functional			
Test name	Testing the Mes	sage-Classification a	nd Message-Sched	uling Services at same NI
Date	03/02/2022		-	
Tester's Name	Luca Visconti (A	kkodis)		
Test scope or objec				
		e message-classifica	tion and message-	scheduling services sending 3
	ort when 2 port config		5	Ū.
Steps				
Step 1	Define a set of r	nessages to send (3	messages at 2 por	ts according to the scheduling
	configuration T	CommScheld.cfg of	NI2)	
	- 2 messages to	NI2 on the port 1		
	- 1 message to N	VI2 on the port 2		
	Port ID	Instant	Next	Instant (microsec)
	2 [NI2-NI0]	2048 (15)	1	61.035
	3 [NI2-NI3]	2185 (16)	0	65,104
	[5 [142-1415]	2105 (10)	0	05,104
Step 2	Send messages			
	// send messa	ges from NI2-NIO u	sing port2	
	{	<mesg_size;i++) <="" td=""><td>nessage size = 5</td><td></td></mesg_size;i++)>	nessage size = 5	
	Xil_Out32	(XPAR_SKINOC_0_S02	AXI_BASEADDR+131	072,i);
	Xil Out32(XPA	R_SKINOC_0_S02_AXI	BASEADDR+163840.	7);
	-			
	/*			*/
		ge from NI2-NI3		
	<pre>for(int i=0;i {</pre>	<mesg_size;i++) <="" td=""><td>message size = 5</td><td></td></mesg_size;i++)>	message size = 5	
	Xil_Out32	(XPAR_SKINOC_0_S02	AXI_BASEADDR+196	608,i);
	Xil Out32(XPA	R SKINOC 0 S02 AXI	BASEADDR+229376.	7); // terminate Message
	/* // send messa	ges from NI2-NIO u	sing port2	*/
	for (int i=	0;i <mesg_size;i++)< td=""><td>// Message size =</td><td>= 5</td></mesg_size;i++)<>	// Message size =	= 5
	(Vil 0:+22	(XPAR_SKINOC_0_S02	AXT BASEADDD+121	072.1):
	ATT_OUE22		DRJERUDKT131	·····
	,			
	Xil_Out32(XPAR_SKINOC_0_S02_	AXI_BASEADDR+16384	40,7);
	Xil_Out32(XPAR_SKINOC_0_S02_	AXI_BASEADDR+16384	40,7);
	/*	XPAR_SKINOC_0_S02_	AXI_BASEADDR+16384	40,7);
Results/Evidence Step 2: NI1 - NI0 NI1 - NI0	/*	_data = 0		*/
Step 2: NI1 - NI0	/*	_data = 0	difference betw	*/
Step 2: NI1 - NI0	/*	_data = 0	difference betw	een m2, m1 = 4
Step 2: N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10	/*	data = 0 data = 1 Time data = 2 data = 3 data = 4 data = 4 data = 6 data = 6	difference betw	een m2, m1 = 4
Step 2: N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10	/*	data = 0 data = 1 Time data = 2 data = 3 data = 4 data = 4 data = 6 data = 6	difference betw	een m2, m1 = 4
Step 2: N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10	/*	data = 0 data = 1 Time data = 2 data = 3 data = 4 data = 4 data = 6 data = 6	difference betw	een m2, m1 = 4
Step 2: N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10	/*	data = 0 data = 1 Time data = 2 data = 3 data = 4 data = 4 data = 6 data = 6	difference betw	een m2, m1 = 4
Step 2: N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10 N11 - N10	/*	data = 0 data = 1 Time data = 2 data = 3 data = 4 data = 4 data = 6 data = 6	difference betw	een m2, m1 = 4
Step 2: Ni1 - Ni6 Ni - Ni6 Ni - Ni6	/* rectived_data=0, original rectived_data=2, original rectived_data=2, original rectived_data=3, original rectived_data=5, original rectived_data=5, original rectived_data=6, original rectived_data=10, original rectived_data=10, original rectived_data=11, original rectived_data=12, original rectived_data=13, original rectived_data=13, original rectived_data=4, original	data = 0 data = 1 data = 2 data = 2 data = 2 data = 5 data = 6 data = 6 data = 0 data = 0 data = 0 data = 0 data = 10 data = 11 data = 13 data = 13 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni8 Ni = Ni8 Ni = Ni8	/* rectived_data=0, original rectived_data=2, original rectived_data=3, original rectived_data=4, original rectived_data=0, original rectived_data=0, original rectived_data=3, original rectived_data=1, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=13, original rectived_data=2, original rectived_data=2, original rectived_data=2, original	data = 0 data = 1 data = 2 fata = 2 fata = 3 data = 5 data = 6 data = 7 data = 0 data = 0 data = 10 data = 11 data = 12 data = 13 data = 12 data = 1 data = 1 d	difference betw	een m2, m1 = 4
Step 2: Ni = Ni6 Ni = Ni6 Ni = Ni6 Ni = Ni8 Ni = Ni3 Ni = Ni3 Ni = Ni3 </td <td>/*</td> <td>data = 0 Time data = 1 Time data = 2 Time data = 6 data = 6 data = 6 data = 6 data = 0 Idata = 10 data = 1 data = 10 data = 1 data = 2 data = 1 data = 2 data = 0 data = 0 data = 0 data = 1 data = 1 data = 0 data = 0 data = 1 data = 0 Idata = 1 data = 1 Idata =</td> <td>difference betw</td> <td>een m2, m1 = 4</td>	/*	data = 0 Time data = 1 Time data = 2 Time data = 6 data = 6 data = 6 data = 6 data = 0 Idata = 10 data = 1 data = 10 data = 1 data = 2 data = 1 data = 2 data = 0 data = 0 data = 0 data = 1 data = 1 data = 0 data = 0 data = 1 data = 0 Idata = 1 data = 1 Idata =	difference betw	een m2, m1 = 4
Step 2: N1 = N16 N1 = N16	/* rectved_data=0, original rectved_data=2, original rectved_data=2, original rectved_data=2, original rectved_data=4, original rectved_data=4, original rectved_data=3, original rectved_data=4, or	data = 0 Time gata = 1 Time gata = 4 Time gata = 6 Time gata = 6 Time gata = 6 Time gata = 6 Time gata = 7 Time gata = 7 Time gata = 6 Time gata = 7 Time gata = 6 Time gata = 7 Time gata = 6 Time gata = 6 Time gata = 7 Time gata = 6 Time gata = 7 Time gata = 7 Time gata = 8 Time gata = 1 Time Gata = 10 Time Gata = 11 Time Gata = 12 Time Gata = 13 Time	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=2, original rectived_data=2, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Niö Nii = Niö Nii = Niö	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=2, original rectived_data=2, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nii = Nič Nii = Nič Nii = Nič Nii = Niš Nii = Niš Nii = Nič Nii = Nič Nii = Nič Nič	/* rectved_data=0, original rectved_data=2, original rectved_data=2, original rectved_data=2, original rectved_data=3, original rectved_data=4, original rectved_data=3, original rectved_data=3, original rectved_data=3, original rectved_data=3, original rectved_data=3, original rectved_data=3, original rectved_data=4, or	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nil = Nile Nil = N	/* rectived_data=0, original rectived_data=1, original rectived_data=1, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=3, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=1, original rectived_data=2, original rectived_data=3, original rectived_data=0, original rectived_data=0, original rectived_data=1, original rectived_data=2, original rectived_data=2, original rectived_data=3, original rectived_data=3	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nil = Nile Nil = N	/*************************************	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nil = Nilo Nil = N	/*************************************	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: N1 N1 N10 N1 N10 N10 N10 N1 N10 N11 N10 N11 N11 N11 N11 N11 N13 N11 N13 N11 N10 N10	/*************************************	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nil = Nilo Nil = N	/*************************************	data = 0 Time data = 1 Time data = 1 Time data = 5 data = 1 data = 6 data = 1 data = 8 data = 1 data = 1 data = 2 data = 2 data = 2 data = 2 data = 2 data = 2 data = 3	difference betw	een m2, m1 = 4
Step 2: Nil = Nilo Nil = N	/*	data = 0 gata = 1 gata = 1 gata = 2 gata = 4 gata = 5 gata = 6 gata = 8 gata = 8 gata = 12 gata = 12 gata = 12 gata = 0 gata = 12 gata = 0 gata = 10 gata = 12 gata = 12	dlfference betw	een m2, m1 = 4
Step 2: NII NIII NIII NIII NIII NIII NIII NIII NIIII NIII NIII NIII NIIII NIII NIII NIII NIIII NIIII NIIII NIII NIIII NIIII NIIIII NIIII NIIII NIIIIIIIIIIIIIII	/*************************************		difference betw difference betw g", "hw.cfg"	een m2, m1 = 4 een m3, m1 = 122
Step 2: NII NIII NIII NIII NIII NIII NIIII NIII NIIII NIIIII NIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	/*************************************		dlfference betw dlfference betw g", "hw.cfg" C running on the b	een m2, m1 = 4 een m3, m1 = 122
Step 2: N1 N1 N10 N1 N10 N10 N10 N1 N10 N11 N10 N1 N10 N11 N10 N1 N10 N11 N10 N1 N11 N11 N11 N1 N11 N13 N11 N1 N13 N11 N13 N1 N10 N11 N10	/*************************************		dlfference betw dlfference betw g", "hw.cfg" C running on the b	een m2, m1 = 4 een m3, m1 = 122
Step 2: NII NIII NIII NIII NIII NIII NIII NIII NIIII NIII NIII NIII NIII NIIII NIIII NIII NIIII NIIII NIIII NIII NIIII NIIII NIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	/*************************************		dlfference betw dlfference betw g", "hw.cfg" C running on the b	een m2, m1 = 4 een m3, m1 = 122

Table 119 - Validation Test T03_WP6T62-0X (overview)

Copyright © FRACTAL Project Consortium	147 of 149
--	------------

Title

	Validatio	Diffest
Test ID	T04 WP6T62-0X	
Test type	Functional	
Test name	Testing the Ingress and Egress-o	
Date	03/02/2022	
Tester's Name	Luca Visconti (Akkodis)	
Test scope or objec		
		g and egress-queuing services using the max lenght of
•		is and egress-queuing services using the max length (
queue.		
Steps	Cond a massage to NIO on parts	1 with a length loss than the guove length
Step 1	•	1 with a length less than the queue length
	<pre>for(int i=0;i<mesg_size1;< pre=""></mesg_size1;<></pre>	<u>i++)</u>
	Xil_Out32 (XPAR_SKINOC	<pre>0_S00_AXI_BASEADDR+131072,message1+i);</pre>
	<pre>} Xil_Out32(XPAR_SKINOC_0_S(</pre>	0 AVI BASEADDP+163840 7) -
Step 2		1 with a length equal to the queue length
	<pre>for(int i=0;i<mesg_size2 pre="" {<=""></mesg_size2></pre>	(1++)
	Xil_Out32 (XPAR_SKINOC_0_	<pre>S00_AXI_BASEADDR+131072,message2+i);</pre>
	Xil Out32 (XPAR SKINOC 0 S	00_AXI_BASEADDR+163840,7);
Stop 2		
Step 3	5 1	1 with a length bigger than the queue length
	<pre>for(int i=0;i<mesg_size3 pre="" {<=""></mesg_size3></pre>	;1++)
	Xil_Out32 (XPAR_SKINOC_0	_S00_AXI_BASEADDR+131072,message3+i);
	Xil Out32 (XPAR SKINGC 0 S	00 AXI BASEADDR+163840,7);
/		
Results/Evidence		
Message id = 7, r Message id = 8, r Message id = 9, r	ecived_data=9 , original_data = 9 ecived_data=10 , original_data = 10 ecived_data=11 , original_data = 11 ecived_data=12 , original_data = 12 23	<pre>Message to = 0, rectived_data=>, original_data = 5 Message to = 2, rectived_data=, original_data = 6 Message to = 2, rectived_data=>, original_data = 7 Message to = 3, rectived_data=0, original_data = 8 Message to = 4, rectived_data=10, original_data = 10 Message to = 6, rectived_data=10, original_data = 11 Message to = 6, rectived_data=12, original_data = 11 Message to = 6, rectived_data=12, original_data = 12 Message to = 8, rectived_data=13, original_data = 13 Message to = 8, rectived_data=14, original_data = 13 Message to = 8, rectived_data=14, original_data = 13</pre>
Time stamp=371967 Message id = 0, rr Message id = 1, rr Message id = 2, rr Message id = 2, rr Message id = 3, rr Message id = 6, rr Message id = 6, rr Message id = 10, r Message id = 10, r Message id = 12, r Message id = 14, r Time stamp=72369	<pre>trived data=4 , original data = 4 trived data=5 , original data = 5 trived data=6 , original data = 7 trived data=7 , original data = 7 trived data=8 , original data = 8 trived data=10 , original data = 10 trived data=11 , original data = 12 trived data=14 , original data = 13 trived data=15 , original data = 14 ercured data=16 , original data = 16 ercured data=16 , original data = 16 ercured data=18 , original data = 17 ercured data=18 , original data = 17 ercured data=18 , original data = 18 </pre>	Message (d = 5, rectved_data=10, original_data = 10 Message (d = 6, rectved_data=10, original_data = 11 Message (d = 7, rectved_data=11, original_data = 13 Message (d = 10, rectved_data=14, original_data = 14 Message (d = 10, rectved_data=14, original_data = 15 Message (d = 11, rectved_data=10, original_data = 16 Nessage (d = 11, rectved_data=10, original_data = 17 Message (d = 11, rectved_data=10, original_data = 17 Message (d = 11, rectved_data=10, original_data = 17 Nessage (d = 11, rectved_data=10, original_data = 17 Message (d = 11, rectved_data=10, original_data = 17 Message (d = 11, rectved_data=20, original_data = 17 Message (d = 11, rectved_data=20, original_data = 12 Message (d = 10, rectved_data=20, original_data = 21 Message (d = 10, rectved_data=20, original_data = 23 Message (d = 10, rectved_data=20, original_data = 23 Message (d = 21, rectved_data=20, original_data = 24 Message (d = 22, rectved_data=20, original_data = 26 Message (d = 22, rectved_data=20, original_data = 27 Message (d = 22, rectved_data=20, original_data = 28 Message (d = 20, rectved_data=20, original_data = 33 Message (d = 20, rectved_data=20, original_data = 33 Message (d = 20, rectved_data=20, original_data = 33 Message (d = 30, rectved_data=20, original_data = 34 Message (d = 30, rectved_data=20, original_data = 34 Message (d = 30, rectved_data=20, original_data = 44 Message (d = 43, rectved_data=20, original_data = 44 Message (d = 43, rectved_data=20, original_data = 44 Message (d = 44, rectved_data=20, original_data = 4
Step 2: Message id = 0, rr Message id = 1, rr Message id = 2, rr Message id = 3, rr Message id = 4, rr Message id = 6, rr Message id = 6, rr Message id = 6, rr Message id = 10, rr Message id = 11, Message id = 13, Message id = 13,	<pre>trived data=4 , original data = 4 trived data=5 , original data = 5 trived data=6 , original data = 7 trived data=7 , original data = 7 trived data=8 , original data = 8 trived data=10 , original data = 10 trived data=11 , original data = 12 trived data=14 , original data = 13 trived data=15 , original data = 14 ercured data=16 , original data = 16 ercured data=16 , original data = 16 ercured data=18 , original data = 17 ercured data=18 , original data = 17 ercured data=18 , original data = 18 </pre>	Message (d = 15, rectved_data-20, original_data = 20 Message (d = 17, rectved_data-20, original_data = 21 Nessage (d = 17, rectved_data-20, original_data = 22 Message (d = 18, rectved_data-20, original_data = 22 Message (d = 20, rectved_data-20, original_data = 24 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 27 Message (d = 22, rectved_data-20, original_data = 37 Message (d = 20, rectved_data-20, original_data = 37 Message (d = 28, rectved_data-20, original_data = 33 Message (d = 30, rectved_data-20, original_data = 33 Message (d = 33, rectved_data-20, original_data = 37 Message (d = 37, rectved_data-20, original_data = 44 Message (d = 43, rectved_data-20, original_data = 44 Message (d = 44, rectved_data-20, original_data = 45 Message (d = 44, rectved_data-20
Step 2: Message td = 0, rr Message td = 1, rr Message td = 1, rr Message td = 1, rr Message td = 1, rr Message td = 0, rr Message td = 0, rr Message td = 10, rr Message td = 12, rr	<pre>ctived_data=4 , original_data = 4 ctived_data=5 , original_data = 5 ctived_data=6 , original_data = 5 ctived_data=9 , original_data = 1 ctived_data=9 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=12 , original_data = 12 ctived_data=14 , original_data = 13 ctived_data=15 , original_data = 14 ctived_data=16 , original_data = 16 ctived_data=17 , original_data = 17 cetived_data=18 , original_data = 18 64</pre>	Message (d = 15, rectved_data-20, original_data = 20 Message (d = 17, rectved_data-20, original_data = 21 Nessage (d = 17, rectved_data-20, original_data = 22 Message (d = 18, rectved_data-20, original_data = 22 Message (d = 20, rectved_data-20, original_data = 24 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 27 Message (d = 22, rectved_data-20, original_data = 37 Message (d = 20, rectved_data-20, original_data = 37 Message (d = 28, rectved_data-20, original_data = 33 Message (d = 30, rectved_data-20, original_data = 33 Message (d = 33, rectved_data-20, original_data = 37 Message (d = 37, rectved_data-20, original_data = 44 Message (d = 43, rectved_data-20, original_data = 44 Message (d = 44, rectved_data-20, original_data = 45 Message (d = 44, rectved_data-20
Step 2: Message td = 0, rr Message td = 2, rr Message td = 2, rr Message td = 3, rr Message td = 4, rr Message td = 0, rr Message td = 0, rr Message td = 12, r Message td = 12, r Message td = 12, r Message td = 14, r Message td = 14, r Time stamp=72389 Success criteria Check: - to receive all mess	ctived_data=4 , original_data = 4 ctived_data=5 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=9 , original_data = 7 ctived_data=9 , original_data = 9 ctived_data=0 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=11 , original_data = 12 ctived_data=12 , original_data = 13 ecived_data=14 , original_data = 14 ecived_data=15 , original_data = 16 ecived_data=18 , original_data = 18 64 nages on destination NI	Message (d = 15, rectved_data-20, original_data = 20 Message (d = 17, rectved_data-20, original_data = 21 Nessage (d = 17, rectved_data-20, original_data = 22 Message (d = 18, rectved_data-20, original_data = 22 Message (d = 20, rectved_data-20, original_data = 24 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 27 Message (d = 22, rectved_data-20, original_data = 37 Message (d = 20, rectved_data-20, original_data = 37 Message (d = 28, rectved_data-20, original_data = 33 Message (d = 30, rectved_data-20, original_data = 33 Message (d = 33, rectved_data-20, original_data = 37 Message (d = 37, rectved_data-20, original_data = 44 Message (d = 43, rectved_data-20, original_data = 44 Message (d = 44, rectved_data-20, original_data = 45 Message (d = 44, rectved_data-20
Step 2: Message td = 0, rr Message td = 1, rr Message td = 1, rr Message td = 1, rr Message td = 1, rr Message td = 0, rr Message td = 0, rr Message td = 10, rr Message td = 11, r Message td = 11, r Message td = 14, r The stamp=72369 Success criteria Check: - to receive all message on se	ctived_data=4 , original_data = 4 ctived_data=5 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=9 , original_data = 7 ctived_data=9 , original_data = 9 ctived_data=0 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=11 , original_data = 12 ctived_data=12 , original_data = 13 ecived_data=14 , original_data = 14 ecived_data=15 , original_data = 16 ecived_data=18 , original_data = 18 64 nages on destination NI	Message (d = 15, rectved_data-20, original_data = 20 Message (d = 17, rectved_data-20, original_data = 21 Nessage (d = 17, rectved_data-20, original_data = 22 Message (d = 18, rectved_data-20, original_data = 22 Message (d = 20, rectved_data-20, original_data = 24 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 26 Message (d = 22, rectved_data-20, original_data = 27 Message (d = 22, rectved_data-20, original_data = 37 Message (d = 20, rectved_data-20, original_data = 37 Message (d = 28, rectved_data-20, original_data = 33 Message (d = 30, rectved_data-20, original_data = 33 Message (d = 33, rectved_data-20, original_data = 37 Message (d = 37, rectved_data-20, original_data = 44 Message (d = 43, rectved_data-20, original_data = 44 Message (d = 44, rectved_data-20, original_data = 45 Message (d = 44, rectved_data-20
Step 2: Message td = 0, rr Message td = 2, rr Message td = 2, rr Message td = 3, rr Message td = 5, rr Message td = 0, rr Message td = 0, rr Message td = 10, r Message td = 11, r Message td = 14, r Message td = 14, r Time stamp=72389 Success criteria Check: • to receive all messages on se Test observations	rectived_data=4 , original_data = 4 rectived_data=5 , original_data = 5 rectived_data=5 , original_data = 7 rectived_data=7 , original_data = 7 rectived_data=9 , original_data = 10 rectived_data=10 , original_data = 11 rectived_data=12 , original_data = 12 rectived_data=14 , original_data = 13 rectived_data=15 , original_data = 14 rectived_data=26 , original_data = 16 rectived_data=26 , original_data = 16 rectived_data=30 , original	<pre>Message (d = 15, rectved_dta=20, original_dta = 20 Message (d = 17, rectved_dta=20, original_dta = 22 Message (d = 18, rectved_dta=20, original_dta = 22 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 19, rectved_dta=20, original_dta = 24 Message (d = 19, rectved_dta=20, original_dta = 24 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 37 Message (d = 22, rectved_dta=20, original_dta = 37 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 20, rectved_dta=20, original_dta = 33 Message (d = 20, rectved_dta=20, original_dta = 33 Message (d = 30, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 36 Message (d = 33, rectved_dta=20, original_dta = 38 Message (d = 33, rectved_dta=20, original_dta = 38 Message (d = 33, rectved_dta=20, original_dta = 43 Message (d = 33, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 45 Message (d = 44, rectved_dta=20, original_dta = 54 Time_stamp=20</pre>
Step 2: Message td = 0, rr Message td = 2, rr Message td = 2, rr Message td = 5, rr Message td = 6, rr Message td = 6, rr Message td = 0, rr Message td = 10, rr Message td = 11, r Message td = 11, r Message td = 11, r Message td = 12, r Message td = 12, r Message td = 12, r Message td = 13, r Message td = 14, r Message td = 12,	ctived_data=4 , original_data = 4 ctived_data=5 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=9 , original_data = 10 ctived_data=10 original_data = 10 ctived_data=10 original_data = 11 ctived_data=12 , original_data = 12 ctived_data=13 , original_data = 13 ecived_data=15 , original_data = 14 ecived_data=16 , original_data = 16 ecived_data=17 , original_data = 10 ecived_data=18 , original_data = 10 data=10 data=10 config: "ttcommsched.cfg", "po	<pre>Message (d = 15, rectved_dta=20, original_dta = 20 Message (d = 17, rectved_dta=20, original_dta = 22 Message (d = 18, rectved_dta=20, original_dta = 22 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 19, rectved_dta=20, original_dta = 24 Message (d = 23, rectved_dta=20, original_dta = 25 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 23, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 34 Message (d = 33, rectved_dta=20, original_dta = 43 Message (d = 33, rectved_dta=20, original_dta = 44 Message (d = 33, rectved_dta=20, original_dta = 45 Message (d = 43, rectved_dta=20, original_dta = 45 Message (d = 44, rectved_dta=20, original_dta = 55 Message (d = 44, rectved_dta=20, original_dta = 55 Message (d = 44, rectved_dta=20, original_dta = 54 Time_stamp=20</pre>
Step 2: Message td = 0, rr Message td = 2, rr Message td = 2, rr Message td = 5, rr Message td = 6, rr Message td = 6, rr Message td = 0, rr Message td = 10, rr Message td = 11, r Message td = 11, r Message td = 11, r Message td = 12, r Message td = 12, r Message td = 12, r Message td = 13, r Message td = 14, r Message td = 12,	ctived_data=4 , original_data = 4 ctived_data=5 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=6 , original_data = 1 ctived_data=10 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=12 , original_data = 12 ctived_data=13 , original_data = 13 ecived_data=15 , original_data = 14 ecived_data=16 , original_data = 16 ecived_data=17 , original_data = 10 ecived_data=18 , original_data = 10 ecived_data=18 , original_data = 10 ddd	<pre>Message (d = 15, rectved_dta=20, original_dta = 20 Message (d = 17, rectved_dta=20, original_dta = 22 Message (d = 18, rectved_dta=20, original_dta = 22 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 19, rectved_dta=20, original_dta = 24 Message (d = 19, rectved_dta=20, original_dta = 24 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 27 Message (d = 22, rectved_dta=20, original_dta = 37 Message (d = 22, rectved_dta=20, original_dta = 37 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 20, rectved_dta=20, original_dta = 33 Message (d = 20, rectved_dta=20, original_dta = 33 Message (d = 30, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 36 Message (d = 33, rectved_dta=20, original_dta = 38 Message (d = 33, rectved_dta=20, original_dta = 38 Message (d = 33, rectved_dta=20, original_dta = 43 Message (d = 33, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 44 Message (d = 40, rectved_dta=20, original_dta = 45 Message (d = 44, rectved_dta=20, original_dta = 54 Time_stamp=20</pre>
Step 2: Message tid = 0, rr Message tid = 2, rr Message tid = 2, rr Message tid = 3, rr Message tid = 3, rr Message tid = 1, rr	ctived_data=4 , original_data = 4 ctived_data=6 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=7 , original_data = 7 ctived_data=10 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=12 , original_data = 12 ctived_data=13 , original_data = 13 ecived_data=16 , original_data = 14 ecived_data=17 , original_data = 16 ecived_data=17 , original_data = 10 ecived_data=18 , original_data = 10 ecived_data=18 , original_data = 10 ddd	<pre>Message (d = 15, rectved_dta=20, original_dta = 20 Message (d = 17, rectved_dta=20, original_dta = 22 Message (d = 18, rectved_dta=20, original_dta = 22 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 23, rectved_dta=20, original_dta = 33 Message (d = 23, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 34 Message (d = 34, rectved_dta=20, original_dta = 43 Message (d = 44, rectved_dta=20, original_dta = 45 Message (d = 44, rectved_dta=20, original_dta = 54 Time_stamp=20</pre>
Step 2: Message td = 0, rr Message td = 2, rr Message td = 2, rr Message td = 3, rr Message td = 5, rr Message td = 0, rr Message td = 0, rr Message td = 12, r Message td = 12, r Message td = 12, r Message td = 14, r Time stamp=72389 Success criteria Check: • to receive all mess	ctived_data=4 , original_data = 4 ctived_data=6 , original_data = 5 ctived_data=6 , original_data = 7 ctived_data=7 , original_data = 1 ctived_data=1 , original_data = 10 ctived_data=10 , original_data = 11 ctived_data=12 , original_data = 12 ctived_data=13 , original_data = 13 ecived_data=16 , original_data = 14 ecived_data=17 , original_data = 16 ecived_data=17 , original_data = 10 ecived_data=18 , original_data = 10 ecived_data=18 , original_data = 10 ddd	<pre>Message (d = 15, rectved_dta=20, original_dta = 20 Message (d = 17, rectved_dta=20, original_dta = 22 Message (d = 18, rectved_dta=20, original_dta = 22 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 19, rectved_dta=20, original_dta = 23 Message (d = 22, rectved_dta=20, original_dta = 33 Message (d = 23, rectved_dta=20, original_dta = 33 Message (d = 23, rectved_dta=20, original_dta = 33 Message (d = 33, rectved_dta=20, original_dta = 34 Message (d = 34, rectved_dta=20, original_dta = 43 Message (d = 44, rectved_dta=20, original_dta = 45 Message (d = 44, rectved_dta=20, original_dta = 54 Time_stamp=20</pre>

Table 120 - Validation Test T04_WP6T62-0X (overview)

Copyright © FRACTAL Project Consortium	148 of 149
--	------------

	Project	FRACTAL
FRACTAL	Title	FRACTAL engineering framework validation
- *-	Del. Code	D6.4

			Validation Te	st		
Test ID	T05 WP6	5T62-0X				
Test type	Functional					
Test name	Testing the serialization services					
Date						
Tester's Name	03/02/2022 Luca Visconti (Akkodis)					
			5)			
Test scope or objective			·			
The objective of this test	t is to valid	date the seria	alization service	on NGW out port		
Steps	-					
Step 1				• ·	ccording to the scheduling	
	-		mScheld.cfg of	NI3)		
		ssage to NI3	on port 2			
	Port ID		Instant	Next	Instant (microsec)	
	2[NI3-N	[12]	1502 (11)	0	44,759	
Step 2	Sond mo	scago at ono	port			
Step 2		ssage at one	-			
		<pre>// send mess INIT_AXI_TX</pre>	age from NI3 , p N=1:	ort2		
		PORT_ID_WR				
		#48				
	4	INIT_AXI		£		
	Т	begin	=0;j<16;j=j+1)//	Т		
		#8;				
		INPUT_WDA end	TA =];			
Results/Evidence						
Step 2: Name		Value				
L clk		1				
₩ reset_n ₩ enq		0				
> 🗑 din[31:0]		0000001c				
🕌 full 🕌 deq		0				
> 🗑 dout[31:0]						
> 📽 msglen[9:0]		005				
¥ empty > ⊮ buffer_inst[0:	5111[31:0]	0	0001.00000002.000000	3,00000004,UUUUUUUUUUUUU	00000.001	
> 😻 wrpntr[9:0]		005				
> 😻 rdpntr[9:0]		000				
lå full_loc lå empty_loc		0				
> 😻 nqd[9:0]		005				
18 AddrWidth 18 WordWidth		9 32				
> W C_MAX_NUM_E	ELEMENTS[9:0]	200				
NI Sender	0					
	1					
Message before Serialized						
Message before ser (Message after Serialized	0000002	200	000000	CCCCCCCC X CCCCCCCC X C	0000002 0000003 00000004	
message alter senaized						
> 🖬 message before ser	0000002	X0000	0000	000000 × 00000001 × 0000000	2 00000003 00000004	
Message after Serialized				^		
> 👹 message after ser	00000002	00000000 13000410				
Success criteria						
Check:						
 the message from the 			2			
 the serialized message 	e (flits in o	rdered)				
Test observations						
Test configuration			d.cfg", "port.cfg			
Test conditions					ard, all configuration files at	
					HW-Edge-Controller	
Remarks	-		, · · · ·	· · ·		
Test result	1					
Passed						

Table 121 - Validation Test T05_WP6T62-0X (overview)

Copyright © FRACTAL Project Consortium	149 of 149	
--	------------	--