MASTERPIECE -

Multidisciplinary Approaches and Software Technologies for Engagement, Recruitment and Participation in Innovative Energy Communities in Europe

Deliverable 5.1

Holistic proof-of-concept and pilot implementation roadmaps

Title	Holistic proof-of-concept and pilot implementation roadmaps
Document description	The planning activities to accelerate the technical developments at the planned pilot sites will be reported in the current report, which will consider the planning of proof of concept, integration, verification and monitoring activities - guided by the needs of businesses, owners and citizens/residents - as well as the creation of prototypes and adaptations of the pilot. Two versions of the deliverable are planned, the current one for month 18 and the planned update for month 33. This continuity will ensure the communication of transparent and solid results at different but very important stages of the different pilot demonstration tasks 5.1, 5.2, 5.3, 5.4 and 5.5.
Nature	Type and dissemination level: DEM - PU
Task	T5.1
Status	FINAL
WP	WP5. Integration, demonstration, and evaluation
Lead Partner	ODINS
Partners Involved	ODINS, UMU
Date	28/06/2024

Revision history	Author	Delivery date	Summary of changes and comments
Version 0.1	ODINS	27/05/2024	ToC
Version 0.2	UMU	03/06/2024	Contributions from the PoC
Version 0.3	ODINS	07/06/2024	Main contributions and general revision
Version 0.4	EXP + AMU	21/06/2024	Contributions of the reviewers
Version 1.0	ODINS	28/06/2024	Final version

Disclaimer:

TABLE OF CONTENTS

LIST OF FIGURES

1. EXECUTIVE SUMMARY

This document is the first part of a series of deliverables that will describe the implementation of the integration, demonstration and evaluation activities planned in the four pilot scenarios: Italy, Turkey, France and Sweden, in addition to the Proof of Concept (PoC) hosted by UMU in Spain, combining social innovations and digital tools identified in the MASTERPIECE project.

It represents the first version of the "*Holistic proof-of-concept and pilot implementation roadmaps*", which will be updated in month 33. It is a technical document that demonstrates the chosen mechanisms for defining a systematic roadmap. This roadmap explains how to correctly integrate, install, configure, and monitor the performance of the technological elements planned in the MASTERPIECE project. The aim is to increase the acceptance of companies and end users in the effective creation of new energy community schemes or the expansion of existing ones.

Divided into several sections, the document presents a detailed global integration/configuration guide for the MASTERPIECE solution, providing a highlighted route followed by the PoC as a reference pilot, which explains the tasks covered in the different phases (definition: devices; definition: use cases; installation, configuration and integration; testing and validation; evaluation), and includes tailored guides for different stakeholders. In addition, this document also covers the current integration/verification status of the pilots and the performance monitoring initial analysis of the implemented technological elements.

Finally, some lessons learnt from the PoC will be presented, which will serve as a reference to speed up technical developments in other pilots and facilitate the implementation of the solution in other scenarios.

2. Introduction

2.1. Background and context

The ambitious MASTERPIECE project aims to rethink the dynamics of Energy Communities and promote their creation and development throughout Europe in response to the urgent demand for transformative solutions in the European energy environment. The fundamental goal of MASTERPIECE is to provide a digital space for collaboration and coordination that stands out for being designed with direct participation of multiple stakeholders in mind. Using participatory approaches and user-centric solutions, the program hopes to inspire citizens to get active in the energy industry. This strategy adds something unique to the energy ecosystem in Europe, trying to make the whole sector work in a more distributed and scalable way.

The integration, demonstration and evaluation activities foreseen in the pilot scenarios will allow for the integration of social innovations and appropriate digital tools designed in other Work Packages, WP3 and WP4 respectively. This is one of the main tasks of this Work Package.

In addition, the implementation of the pilots will include a Proof of Concept (PoC) and early integration of the system in a pilot scenario to validate the main tools of the MASTERPIECE solution, the installation and operational demonstration of all the interventions in the pilot studies, and the monitoring of post-installation data and multi-dimensional impact assessment.

By using the PoC to holistically analyse the steps taken to help define a systematic roadmap for the integration of the pilots, the MASTERPIECE solution provides valuable information that can be used as a reference and replicated for future pilots. This widespread knowledge of how to correctly integrate, install, configure and monitor the performance of the technological elements envisaged in the project aims to increase the acceptance of companies and end users in the effective creation of new community energy schemes (or the expansion of existing ones).

2.2. Relationship with other WPs, Deliverables, and Tasks

The current deliverable contributes to a cohesive vision of the MASTERPIECE project's implementation program in coordination with other deliverables and tasks across several Work Packages. It is essential to highlight these links to promote a comprehensive understanding of the planning of the proof-of-concept, integration, verification and monitoring activities and the overall objective of the MASTERPIECE project.

The main interconnections with other WPs are:

- WP2 "Energy community requirements at national and EU levels for different stakeholders and shareholders".
- WP4 "Digital platforms and tools for energy communities".

2.2.1. Rationale for relationship with WP2

WP2 includes all the planning activities aimed at speeding up the technical developments at the sites planned for the pilots. As such, the results of T2.3 'Pilot surveys, validation scenarios analysis and deployment definition', T2.4 'Evaluation planning and KPIs definition' and T2.5 'Social innovations specifications and architectural blueprint' are important outputs for the construction of this deliverable.

In this regard, D5.1 is directly connected to the results generated in the different tasks from WP2 related to use case definition and evaluation, two of the main blocks described in the ***Execution Plan*** introduced in this document.

Another relevant aspect introduced in this document connected to evaluation is included in the ***Monitoring and Verification*** section, since monitoring is an important activity to be carried out in multiple contexts, such as pure maintenance, detection of data losses, etc.

2.2.2. Rationale for relationship with WP4

Since D5.1 is a demonstrator-type deliverable, part of its content is related to integration and deployment. As a result, there is a direct connection with WP4 as a technical Work Package, especially when it comes to describe some of the decisions made during the integration of data sources in the PoC.

More specifically, this alignment is related to the way to represent the information described in D4.3 *First implementation of the digital platform WP4 and tools*.

2.2.3. Alignment with other WP5 Deliverables

In month 15, the first deliverable of this Work Package was submitted, D5.3 *Intervention Program: All pilots WP5 implementation*. There are several references to this deliverable in the description of the PoC, linking to the definition of use cases and KPIs, but also in the section describing the ***Integration*** of data sources, since progress in this area is reported in the current document.

2.3. Document Structure Overview

D5.1 is divided in several sections associated with the PoC, guided by the needs of companies, owners and citizens/occupants, and also includes specific roadmaps for each stakeholder. The document is structured as follows:

- ***Proof of Concept***: This section presents the general information about the PoC will be presented, including the campus map, indicating which buildings will provide information, a description of the environment of the PoC, a description of the systems that will be controlled in each building and their relationship to the use cases. It also includes technical information related to the integration of data sources.
- ***Execution Plan***: This section provides the *Execution Plan* together with the description of each task/activity of the MASTERPIECE project related to WP5, highlighting the activities carried out so far in the PoC, as a reference for the other pilots.
- Sections ***Roadmap for Component Developers***, ***Roadmap for Pilot partners*** and ***Roadmap for Integrators*** present diagrams highlighting the activities applicable to each stakeholder in the *Execution Plan* in order to facilitate understanding and speed up execution of the entire process.
- ***Monitoring and Verification***: This section describes the methodology for monitoring and verification of performance and availability of the technological elements through the use of Jenkins [11].
- ***Lessons learnt from the Proof of Concept***: This section presents the lessons learnt from the planning, integration, verification and monitoring activities carried out so far in the PoC in order to provide important input for improving future integrations, with the aim of achieving trouble-free prototyping, starting with the other pilots of the project.

3. Proof of Concept

3.1. Global overview

Within the University of Murcia's main campus, several buildings stand as pillars of academic excellence, each one with its own unique purpose and contribution to the university community. Some of the University of Murcia facilities serve as the Proof of Concept (PoC) for the project, being the base example for the subsequent pilot projects, and are mostly located on the Espinardo Campus. Within these buildings, various operational devices are housed, in an independent mode or integrated into existing Building Management Systems (BMS).

In the figure below, multiple buildings and solar installations which are part of the Campus of Espinardo in University of Murcia are presented. The blue flags and sun symbols in pink represent the buildings and PV installations respectively, which will form part of the different use cases of the PoC (they were preliminary defined in D5.3), which are monitored and/or controllable depending on the case.

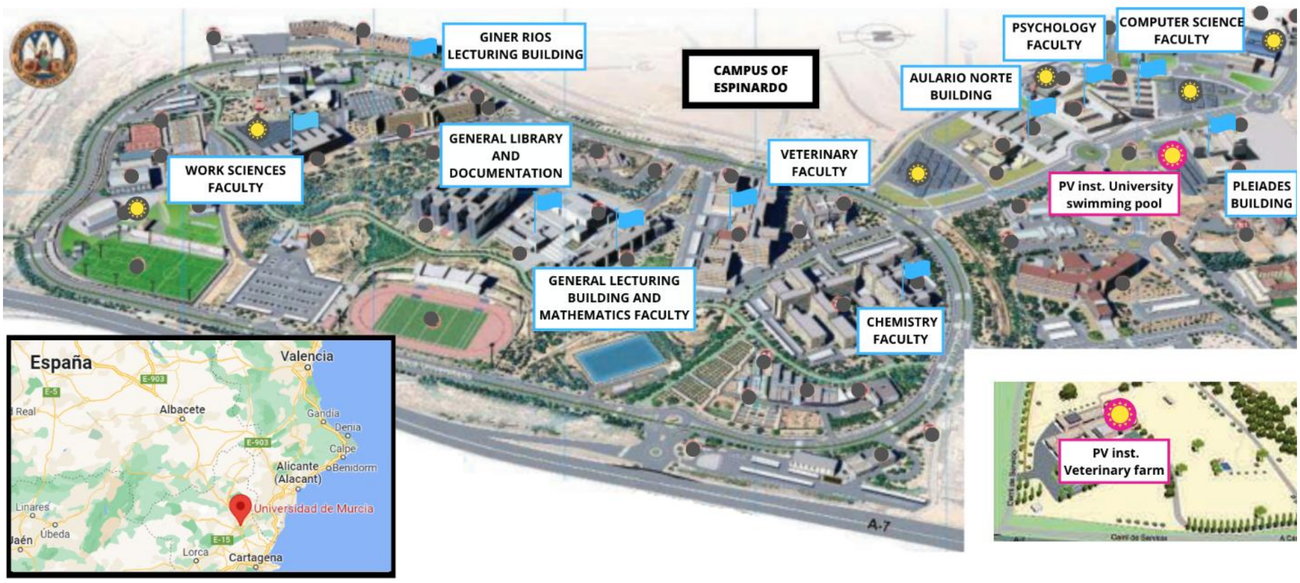

Figure 1: PoC

It is important to point out that in the PoC there is a differentiation between two types of participants within the digital EC, individuals using university facilities and the university buildings themselves. This distinction allows for the implementation of varied use case strategies tailored to meet the needs of both individual users and the broader university infrastructure.

3.2. Selected buildings

The subset of buildings chosen regarding monitoring and controllability, among others aspects for the PoC and its use cases, are the following ones:

- **Pleiades Building**

Standing as a beacon of multidisciplinary research, the Pleiades Building is home to a range of scientific endeavours. Housing eight floors, it accommodates the Interdisciplinary Experimentation Platform in Earth and Life Sciences (PLEIADES), alongside the administrative offices of University of Murcia Editions and a comprehensive Learning and Research Resource Centre Library.

Figure 2: Pleiades Building

The Pleiades Building includes general building consumption, global HVAC consumption, and ventilation system and DHW (domestic hot water) consumption. Additionally, data on temperature, CO_2, humidity, presence, etc. is collected at the room level. People counting in the library is planned to be included as well. HVAC monitoring covers setpoint settings, modes, fan speed and ON/OFF at the room level, while total HVAC control includes setpoint and ON/OFF at the room level.

- **Chemistry Faculty**

This faculty was originally established to modernize scientific education. This complex now serves as the nucleus for Chemistry and related disciplines. With six floors and a distinctive architectural design, it houses departments focusing on Chemistry, Biochemistry, Chemical Engineering and Physics. The main building features expansive windows that invite natural light, fostering an environment conducive to learning and research.

Figure 3: Chemistry Faculty

The Chemistry Faculty monitors general building consumption and global HVAC consumption. People counting is conducted at the building level. HVAC monitoring includes setpoint settings, modes, fan speed and ON/OFF at the room level, with total HVAC control covering setpoint, and ON/OFF at the room level.

- **Veterinary Faculty**

The faculty was recognised for its excellence in veterinary education. This facility is affiliated with the European Association of Establishment for Veterinary Education (EAEVE). Spread across seven floors, it hosts classrooms and crucially controlled laboratories essential for conducting temperature-sensitive experiments and research in the field of veterinary sciences.


Figure 4: Veterinary Faculty

The Veterinary Faculty monitors general building consumption and global HVAC consumption. People counting is conducted at the building level. HVAC monitoring includes setpoint settings, modes, fan speed and ON/OFF at the room level, while total HVAC control covers setpoint, and ON/OFF at the room level.

- **Work Sciences Faculty**

Established to meet the educational needs of the modern workforce, this faculty offers programs in Labor Relations, Human Resources and related fields. Spread over five floors, it provides specialized instruction and resources to prepare students for careers in various industries

Figure 5: Work Sciences Faculty

The Work Sciences Faculty includes general building consumption and global HVAC consumption. People counting is conducted at the building level. HVAC control includes ON/OFF at the block/building level.

- **Computer Science Faculty**

Housed within a modern facility, it offers a dynamic learning environment equipped with cutting-edge resources and state-of-the-art laboratories. Here, students engage in a diverse range of programs and research initiatives spanning computer science, information technology, and related fields. The faculty is committed to nurturing the next generation of computer scientists and professionals.


Figure 6: Computer Science Faculty

The Computer Science Faculty includes general building consumption and global HVAC consumption. People counting is conducted at the building level.

- **Psychology Faculty**

Dedicated to the study of the mind and behaviour, the Psychology Department spans five floors within its modern facility. Here, students pursue degrees at various levels, and researchers explore a myriad of psychological phenomena through state-of-the-art laboratories and academic programs.

Figure 7: Psychology Faculty

The Psychology Faculty monitors general building consumption and global HVAC consumption. People counting is conducted at the building level. HVAC control includes ON/OFF at the block/building level.

- **Mathematics Faculty**

This building serves as a centralized location for academic activities, hosting lectures and seminars across disciplines including Mathematics, Chemistry and Communication. The building is comprised the Mathematics Faculty and the General Lecture room.

Figure 8: Mathematics faculty

The Mathematics Faculty includes general building consumption and global HVAC consumption. Noise levels are measured, and people counting is conducted at the building level. HVAC control includes ON/OFF at the block/building level.

- **General Library**

Comprising both a library and a faculty, this complex is dedicated to the dissemination and preservation of knowledge. The library offers a space for study and research, while the Documentation Faculty focuses on degrees related to communication and information science, providing students with comprehensive training in these field.

Figure 9: General Library

The General Library includes general building consumption and global HVAC consumption. People counting is conducted at the building level, and HVAC control includes ON/OFF at the block/building level.

- **Giner de los Ríos Lecturing Building**

The Faculty of Education teaches in this lecture building. Its fosters intellectual growth and academic discourse among students and faculty alike.

Figure 10: Giner de los Ríos Lecturing Building

The Giner de los Ríos Lecturing Building monitors general building consumption and global HVAC consumption. Data on temperature, CO_2, humidity, presence, etc., is also collected at the room level. People counting is conducted at the building level.

- **Aulario Norte Building**

The building is shared between different degrees such as Computer Science, Psychology and Nursing, and it offers a big canteen.

Figure 11: Aulario Norte Building

The Aulario Norte Building includes general building consumption and global HVAC consumption. Data on temperature, CO_2, humidity, presence, etc. is also collected at the room level. People counting is conducted at the building level.

- **PV Installations**

The PV installations include monitoring of the Veterinary Farm PV and Pool Parking PV. The monitoring parameters include active power, apparent power, current, active energy, and voltage.

Figure 12: Veterinary Farm PV (left) and Pool Parking PV (right)

As detailed previously, numerous buildings offer a variety of energy consumption monitoring capabilities, both at a global level and specifically for HVAC systems. Additionally, environmental data such as temperature, CO_2 levels, humidity as well as people presence are also included. This comprehensive array of data allows for the development of sophisticated components within the MASTERPIECE digital tools ecosystem. These components go from demand modification for consumption optimization and flexibility to the efficient use of renewable energy. For example, real-time monitoring of general building consumption and HVAC-specific data enables dynamic adjustments to energy usage based on grid requirements and electricity tariffs. Environmental sensors provide crucial insights that ensure HVAC operations maintain occupant thermal comfort while optimizing energy use. This integrated approach not only enhances energy savings, but also supports sustainable practices by maximizing self-consumption of photovoltaic energy while reducing the reliance on the grid.

With this, the use cases defined for the PoC can be correctly executed and subsequently evaluated, allowing for continuous monitoring of their status throughout the project's development. The following sections will detail the involvement of the buildings in the corresponding use cases and the impact of their capabilities and monitoring systems on the correct execution and evaluation of these use cases:

UC2 - Promoting Collaboration and Self-Consumption from PV

Involved buildings

- Pleiades Building. To optimise the use of the library of Pleiades as a climate shelter during periods of high photovoltaic generation, a comprehensive strategy to gather students in this space is designed trough the MASTERPIECE solution. During hours when solar energy production is at its peak, we encourage students to concentrate in the library, thereby enjoying a comfortable and climate-controlled environment without overloading the HVAC system in other areas of the campus. This measure not only maximizes the efficient use of renewable energy, but also prevents the dispersion of students in semi-empty areas, contributing to a more controlled and safer environment. The monitoring of people counting will be used for evaluating the effectiveness of this strategy, providing valuable data to ensure optimal occupancy and resource management.
- Giner de los Ríos Lecturing Building and Aulario Norte Building. These buildings are also used by students for study and work. The strategy for this use case starts from the mobilisation of students towards the climate shelter (library of Pleiades Building), therefore, the occupancy data of these buildings are useful to evaluate the participation of the users and the effectiveness of the mobilisation towards the proposed climate shelter.

Additional data sources involved

- Photovoltaic Installations. Data on renewable generation is used to study periods of maximum renewable generation and determine the times during which the use of the library as a climate refuge will be promoted.

KPI Evaluation: Analysis of occupancy and energy consumption allows us to measure the impact of the strategy implemented in this use case.

UC3 - Management of Demand and Fair Pricing

Involved buildings

- 7 Buildings (Pleiades Building, Chemistry Faculty, Veterinary Faculty, Work Sciences Faculty, Psychology Faculty, Mathematics Faculty and General Library). These buildings with the capability to control HVAC systems will be utilized in this use case. The automated control of these systems will enable modifications in energy demand, managed by the DR-FLEX component from MASTERPIECE.

KPI Evaluation

The energy and environmental monitoring of these buildings allow the assessment of the impact of this strategy, enabling the calculation of KPIs related to demand modification and energy cost savings.

UC 4 - Achieving Sustainable Energy Transition and Equitable Access

Involved buildings

- All the buildings. All the buildings are part of the strategy to carry out this use case. As noted above, the buildings have different equipment characteristics, with some being better prepared than others to implement energy efficiency measures and demand modification, making some of them more energy efficient. Energy consumption data, along with environmental data, allows for the analysis of the potential of each building and the design of a strategy to achieve a sustainable energy transition and equitable access.

Additional data sources involved

- Photovoltaic Installations. Data on renewable generation is used to analysing the renewable generation potential of the community and define the strategies for an appropriate distribution of the generation between the altruistic buildings and the energy poorer ones.

KPI Evaluation

Energy consumption data, photovoltaic generation, and other metrics allow us to measure the impact of the strategy by evaluating the shift in grid independence and community sustainability when comparing equitable and altruistic distribution scenarios.

3.3. Additional data sources

There are certain special data sources involved in the PoC, including different BMS and Scadas, other portals, IoT Gateways, etc.

In terms of BMS, there are 2 main platforms available:

- *OpenData*. This Scada / Multi-BMS manages a significant number of systems in multiple campuses of UMU. Out of the information provided by it, the most relevant for the PoC is the access to individual HVAC units in multiple buildings.
- *Pleiades BMS*. This is a BMS for the Pleiades building introduced in the previous section.

Focused on energy consumption monitoring, another Scada provides metering data at a building level for almost all the buildings in the different campuses.

Designed to cover a wider range of datasets, the *Campus as Living Lab* portal [1] has integrated data from different types of data sources, such as building occupancy (people counting), air quality monitoring, energy generation in certain solar plants, etc.

In the context of other separate projects that were executed in the past and are sort of isolated, additional information is available through IoT Gateways and will be integrated as well.

An external data source is also going to participate in the PoC, namely ENTSO-E [2], a public portal which provides access in Europe at a country level to data related to energy (day-ahead) prices, loads, generation forecasts, etc.

3.4. Integration

3.4.1. Architecture

The integration of data sources in the PoC has been designed so that it's not just modular but can also be reused by other consumers/producers now and in the future, when the project is finished. In this regard, an MQTT broker has been deployed in an external machine, where all the PoC-only integrations are deployed, and will be interconnected with the DLT Secure Platform through the FIWARE MQTT/JSON Agent.

An overview of the architecture can be seen in the following figure.

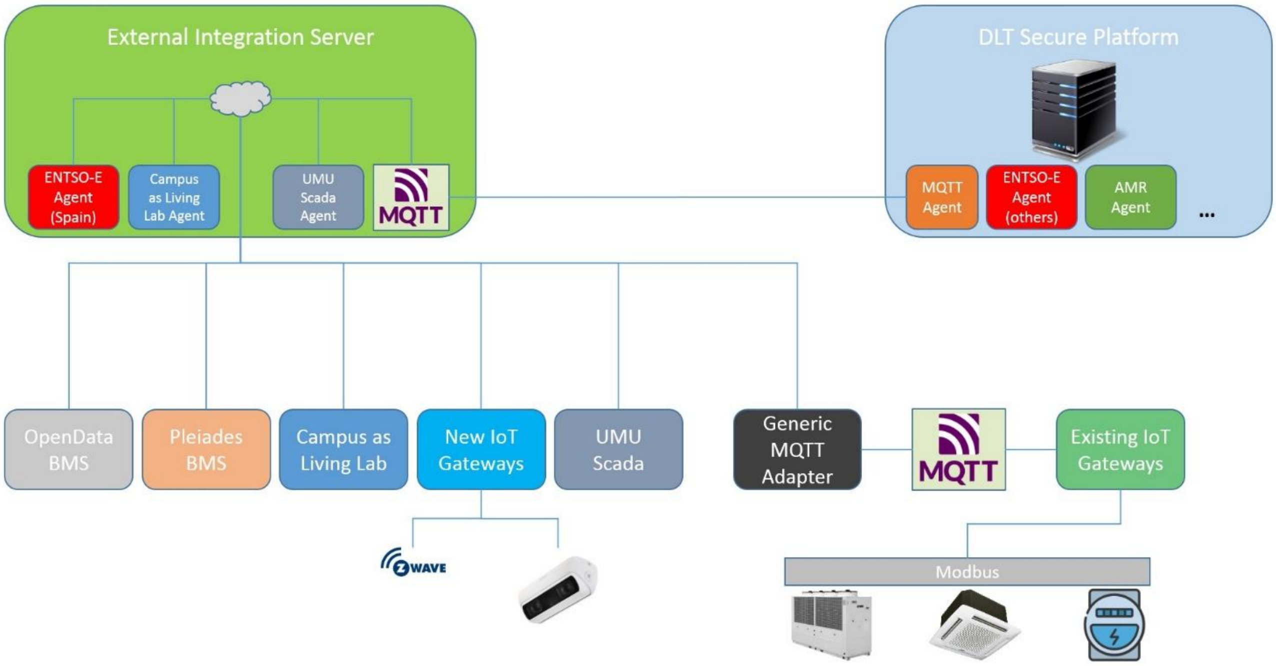

Figure 13: Integration architecture

Since FIWARE is the underlying technology used in multiple projects in UMU and is also used by the DLT Secure Platform, all the data exchanges at MQTT level have been formatted to make them compatible with the set of topics used by the FIWARE MQTT/JSON Agent, which follows this topic pattern:

/apikey/deviceid/topicsuffix

As can be seen in the previous example, each topic includes 3 items:

- apikey → This is merely a group key.
- deviceid → A unique identifier is used for each device.
- topicsuffix → Depending on the type of message sent, there are 3 possible suffixes:
  - cmd → Commands sent to devices.
  - cmdexe → Confirmation of reception of commands.
  - attrs → Values sent by the device.

In order to simplify the management of the external MQTT broker, the two first parts of the topic have been encoded in a well-known format:

- apikey → It's divided in *CAMPUS-BUILDING* (i.e. **Espinardo-Pleiades**).
- deviceid → It's divided in *SOURCE_Device-property* (i.e. **PleiadesBMS_PMGeneral-totalActiveEnergyImport**).

With this, a MQTT publication could be:

- **Topic**

  /Espinardo-Pleiades/PleiadesBMS_PMGeneral-totalActiveEnergyImport/attrs

- **Payload**

  {“value”:1.234,”dateObserved”:”2024-05-06T01:00:00.000Z”}

3.4.2. Integration through IoT Gateways

There are 2 scenarios that involve IoT Gateways: The first one includes some existing equipment that was already producing readings in MQTT format and provided access mostly to Modbus devices (read/write support). The way to provide access to this information is by using an *MQTT ↔ MQTT Adapter* which converts the original data into the unified format. This adapter is based on the one described in D5.3 (see section 5.3.3, implemented on top of *Node-RED* [6], which runs on top of *Node.js* [7]), which updated directly the FIWARE IoT Platform and also included the conversion from global to partial counters when necessary.

In this case, the flow can be seen in the following figure.

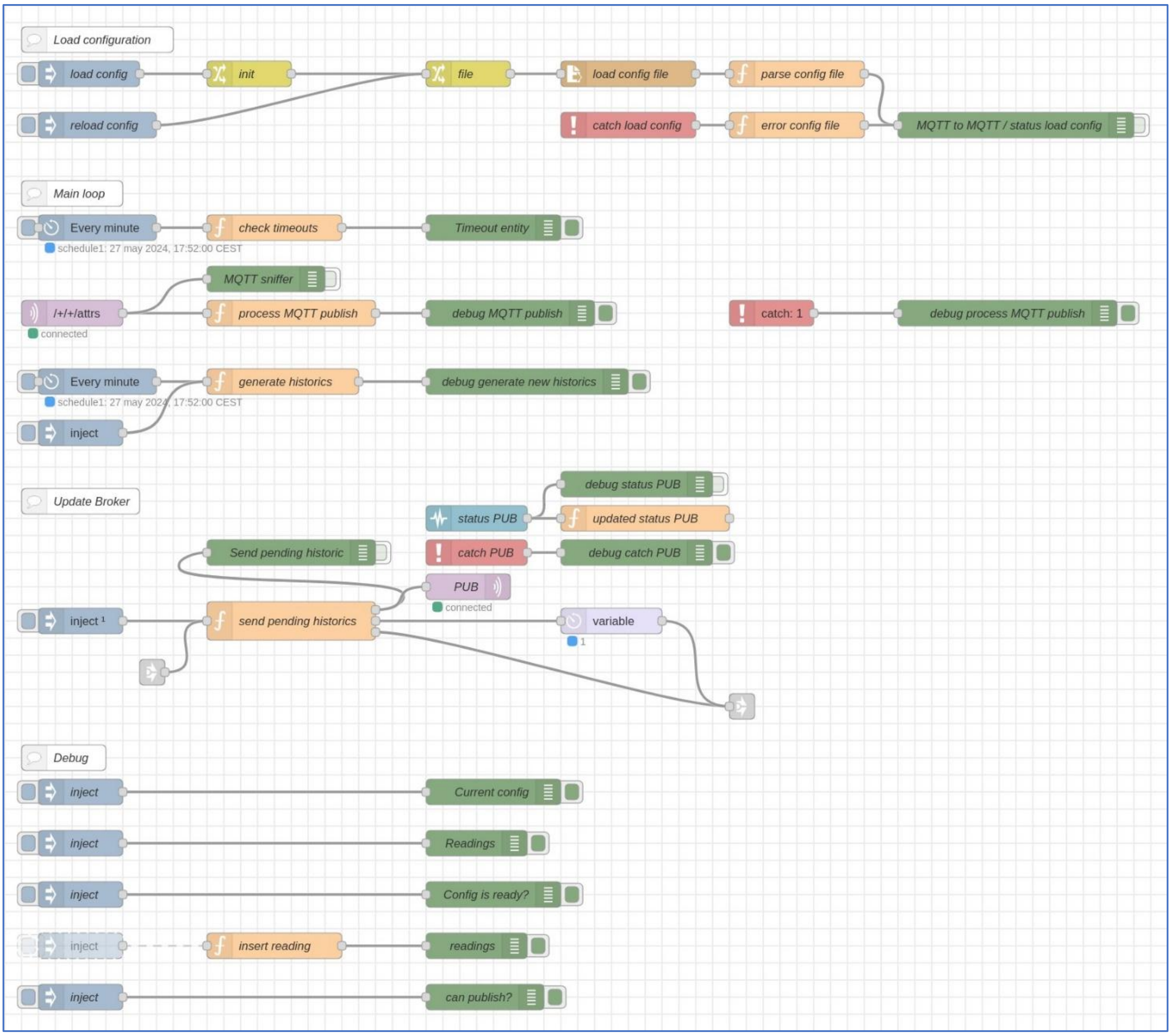

Figure 14: Integration of generic MQTT data sources (into another MQTT broker)

And here is an example configuration file for the previous flow.

```
{
    "valueAttribute":"numValue",
    "timestampAttribute":"dateObserved",
    "includeAdditionalTimestamp":false,
    "additionalTimestampAttribute":"TimeInstant",
    "data": [
        {
            "topic": "/odins/IQ0101G19014600033_3f01ea/attrs",
            "valueSourcePath": [
                "numValue",
                "value"
```

```
            ],
            "timestampSourceType": "attributeInPayload",
            "timestampSourceFormat": "ISO-8601-UTC",
            "timestampSourcePath": [
                "numValue",
                "observedAt"
            ],
            "reportPeriodicityInMinutes": 0,
            "reportTimeoutInMinutes": 0,
            "keepAliveTopic": "",
            "keepAlivePaths": [],
            "outputTopic":"/Espinardo-FacultadInformatica/IQ0101G19014600033_PM1-activeEnergyImport/attrs"
        }
    ]
}
```


On the other hand, new IoT Gateways have been installed to provide access to Z-Wave [4] devices and to integrate people-counting cameras, needed in one of the use cases. While the version of the Z-Wave integration reported in D5.3 was still in progress (it converted the data directly from Z-Wave to NGSI-LD), right now it is fully operational and is also compatible with MQTT.

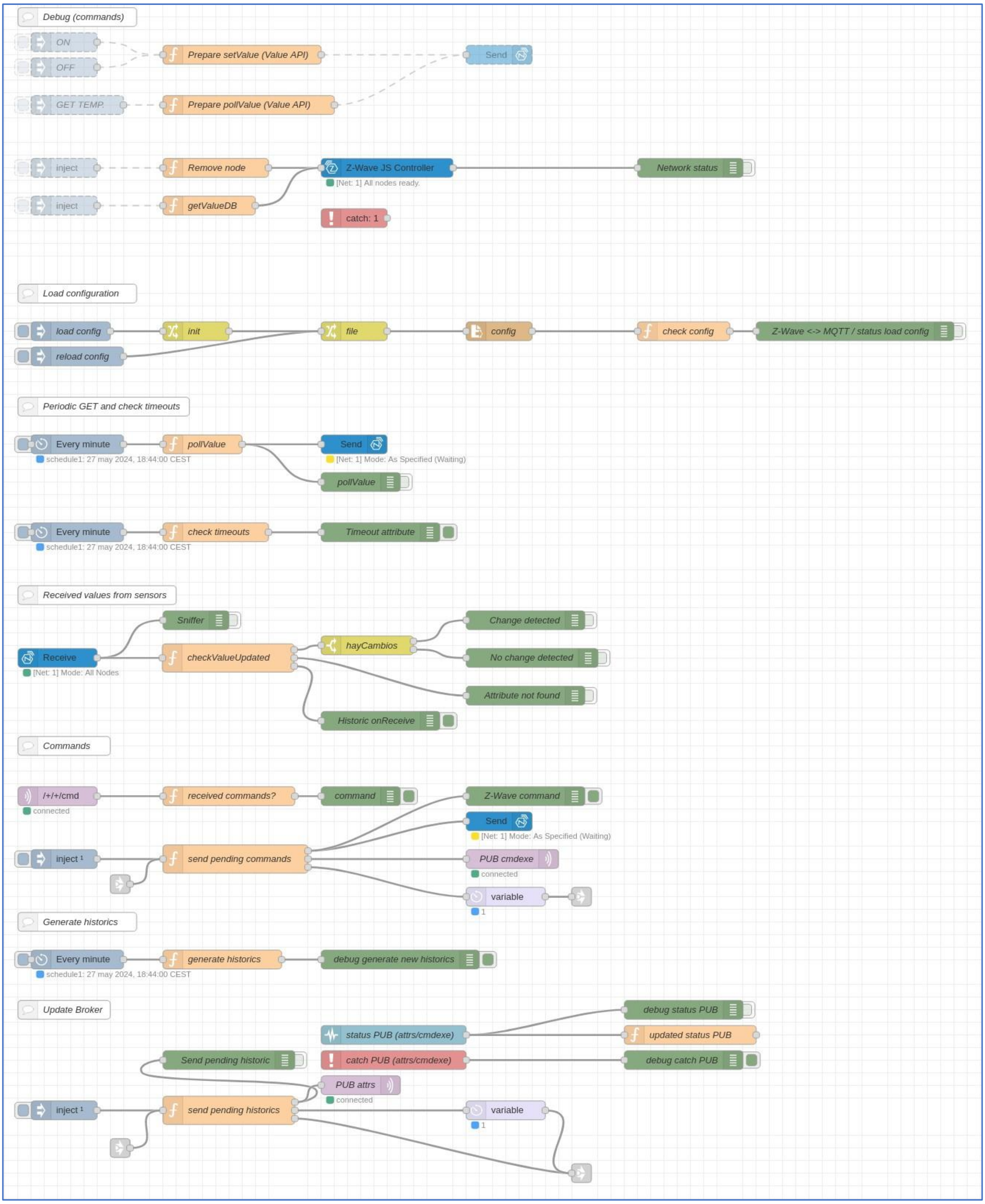

Figure 15: Integration of Z-Wave devices (final MQTT version, bidirectional)

As for the configuration file of this flow, all the required data is extracted using the *Z-Wave JS Control Panel* as described in section 5.3.4 of D5.3 and this information is used later on the template for generation of entities described in section 4.6.2 of D4.3.


```
{
    "valueAttribute":"numValue",
    "timestampAttribute":"dateObserved",
    "includeAdditionalTimestamp":false,
    "additionalTimestampAttribute":"TimeInstant",
    "commands": [
        {
            "apikey":"Espinardo-Pleiades",
            "deviceid":"IQ0101G19014600030_SmartSwitch3",
            "name": "deviceState",
            "nodeid": 3,
            "cmdclass": 37,
            "endpoint": 0,
            "property": "targetValue",
            "type": "boolean"
        }
    ],
    "attributes": [
        {
            "topic": "/Espinardo-Pleiades/IQ0101G19014600030_MCOHome4Library-temperature/attrs",
            "nodeid": 4,
            "cmdclass": 49,
            "endpoint": 0,
            "property": "Air temperature",
            "type": "number",
            "converttype": "None",
            "numdecimals": 1,
            "reportPeriodicityInMinutes": 15,
            "reportTimeoutInMinutes": 30,
            "mapping":"temperature"
        },
        {
            "topic": "/Espinardo-Pleiades/IQ0101G19014600030_MCOHome4Library-humidity/attrs",
            "nodeid": 4,
            "cmdclass": 49,
            "endpoint": 0,
            "property": "Humidity",
            "type": "number",
            "converttype": "None",
            "numdecimals": 1,
            "reportPeriodicityInMinutes": 15,
            "reportTimeoutInMinutes": 30,
```

```
            "mapping":"humidity"
        },
        {
            "topic": "/Espinardo-Pleiades/IQ0101G19014600030_MCOHome4Library-co2/attrs",
            "nodeid": 4,
            "cmdclass": 49,
            "endpoint": 0,
            "property": "Carbon dioxide (CO₂) level",
            "type": "number",
            "converttype": "None",
            "numdecimals": 0,
            "reportPeriodicityInMinutes": 15,
            "reportTimeoutInMinutes": 30,
            "mapping":"co2"
        }
    ]
}
```


Regarding people counting, the selected camera offers a REST API which needs to be queried periodically (no real-time support). As a result, the occupancy level will not be 100% accurate and therefore it is recommended to read the current values without too much delay. However, since this value is not critical, the module will be configured to read values every 5 minutes.

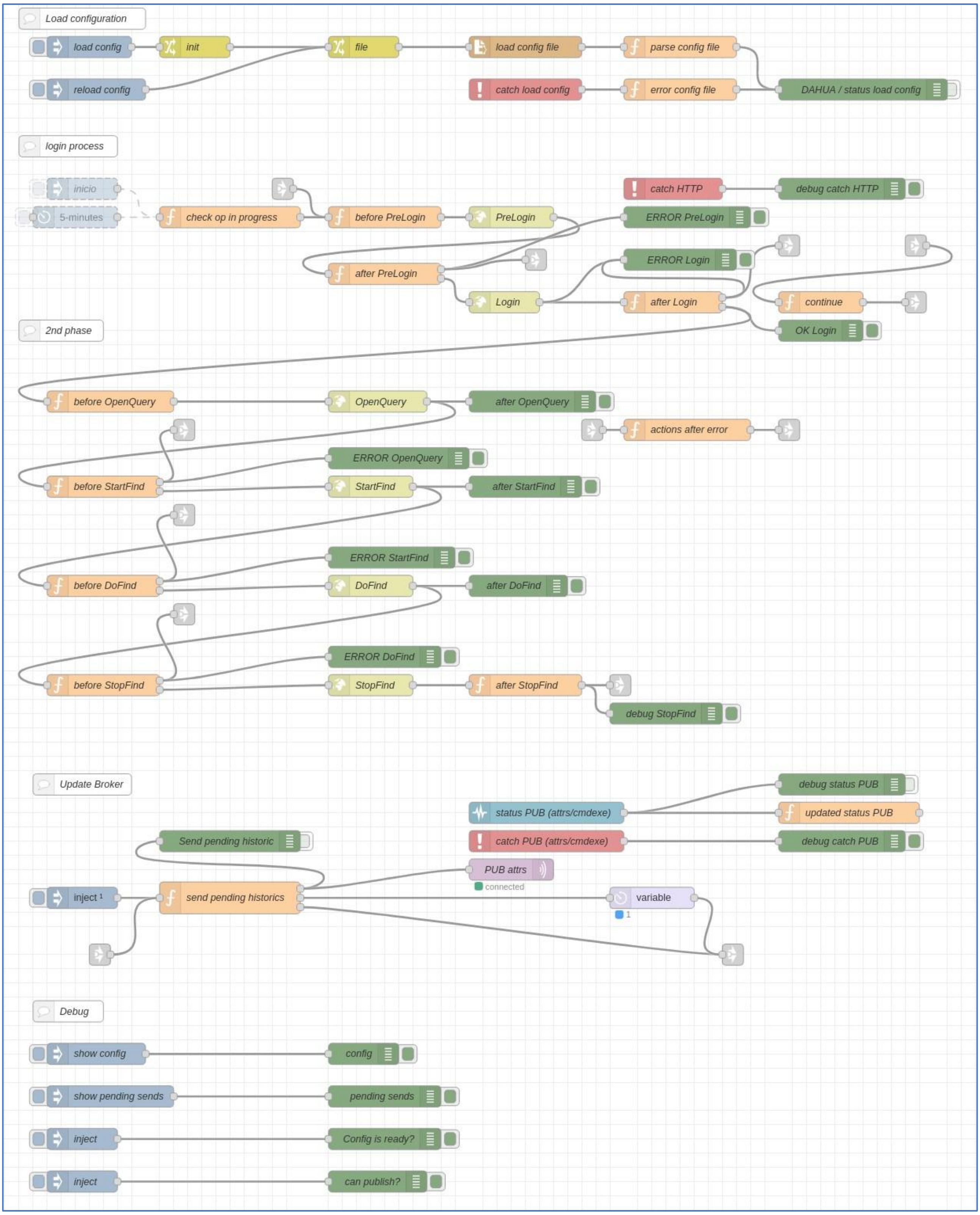

Figure 16: Integration of people-counting cameras

And it also needs a configuration file.


```
{
    "valueAttribute":"numValue",
    "timestampAttribute":"dateObserved",
    "includeAdditionalTimestamp":false,
    "additionalTimestampAttribute":"TimeInstant",
    "data": [
        {
            "topicIn": "/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-in/attrs",
            "topicOut": "/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-out/attrs",
            "topicInStatus": "/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-inStatus/attrs",
            "topicOutStatus": "/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-outStatus/attrs",
            "host": "http://hostname.camera:8080",
            "user": "admin",
            "password": "password",
            "mustExchangeValues":false
        }
    ]
}
```


3.4.3. Integration of BMS, Scadas and other portals

In the case of *OpenData* and *Pleiades BMS*, since we have direct access to the internal databases used by them, custom adapters have been implemented to integrate them in the MQTT ecosystem. They are both deployed in the machines which host the systems. As a result, both reading and writing are supported in a transparent way even though none of them is actually compatible with MQTT.

For the Scada that provides metering information at a building level, the integration accesses the data through a REST API (read only).

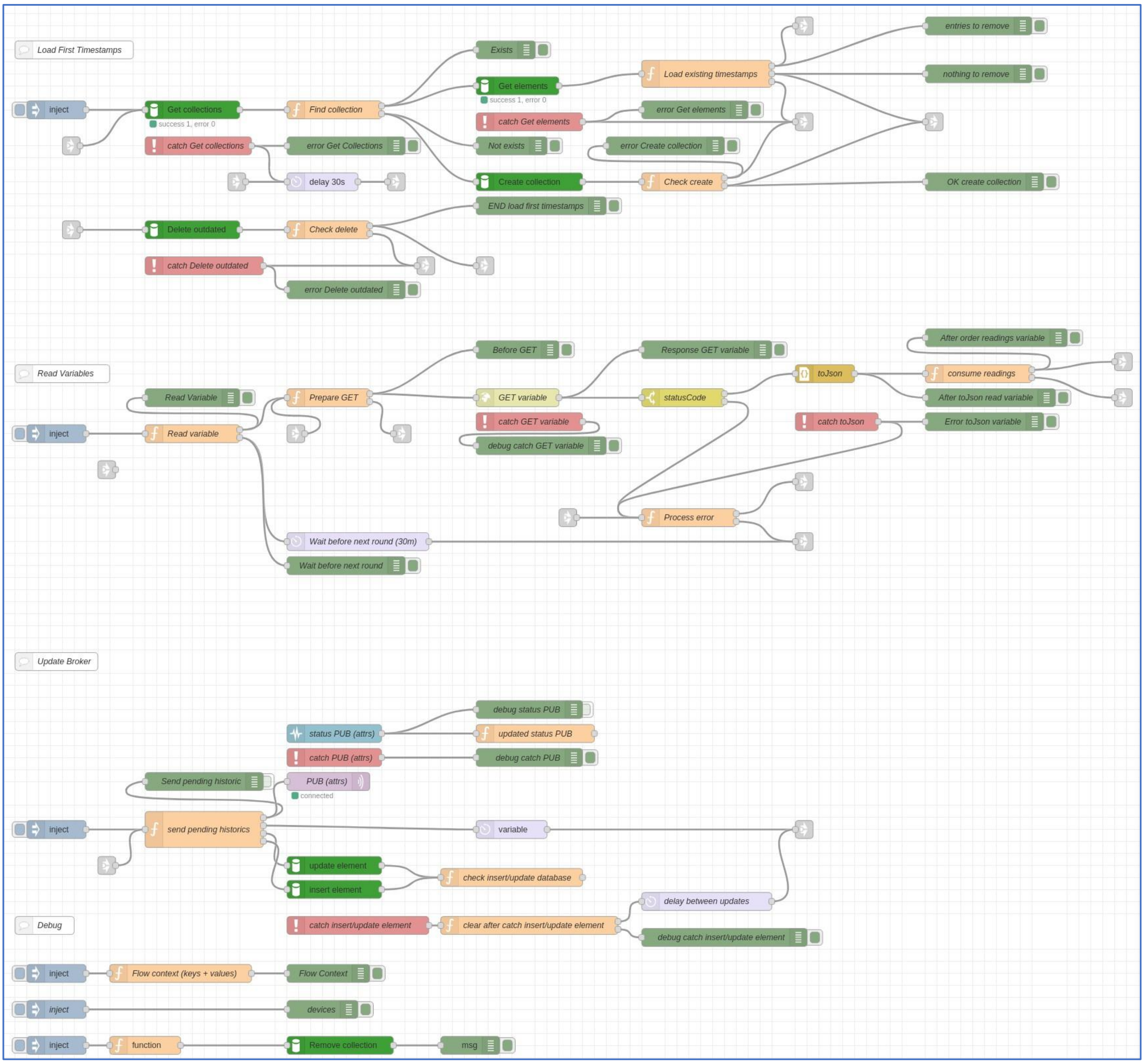

Figure 17: Integration of UMU Scada (power metering at building level)

The last integration is still in progress and will act as a bridge between the external MQTT broker and the *Campus as Living Lab* portal, which works on top of a NGSI-v2 Context Broker, namely *Orion* [3]. Since data from different types of sources is stored in the same portal, a number of groups have been defined using the ***fiware-service*** mechanism offered by *Orion* and one subscription will be created per *fiware-service* to receive notifications every time the devices are updated.

On the other hand, when it comes to send data from MQTT to NGSI-v2, this parameter will have to be defined in addition to the identifier of the destination entity and the property to be updated. An example of configuration file can be seen in the following box.

```
{
    "upload": [
        {
            "topic":"/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-in/attrs",
            "attributeValue":"numValue",
            "attributeTimestamp":"dateObserved",
            "fiware-service":"peoplecounting",
            "fiware-servicepath":"/#",
            "id":"deviceGroup-pleiades-library",
            "property":"peopleEntered",
            "type":"number"
        },
        {
            "topic":"/Espinardo-Pleiades/IQ0101G19014600033_Camera1Library-out/attrs",
            "attributeValue":"numValue",
            "attributeTimestamp":"dateObserved",
            "fiware-service":"peoplecounting",
            "fiware-servicepath":"/#",
            "id":"deviceGroup-pleiades-library",
            "property":"peopleExited",
            "type":"number"
        }
    ],
    "download": [
        {
            "fiware-service":"solarpanelmonitoring",
            "fiware-servicepath":"/#",
            "description-suscription":"external-mqtt-broker-subscription-solarpanelmonitoring",
            "subject": {
                "entities": [
                    {
                        "idPattern": "device-ID0102E18005200066",
                        "type": "Device"
                    },
                    {
                        "idPattern": "device-ID0102E18005200065",
                        "type": "Device"
                    }
                ],
                "condition": {
                    "attrs": []
                }
            },
            "notification": {
                "http": {
                    "url": "http://host:1880/endpoint"
```

```
                }
            },
            "entities": [
                {
                    "id":"device-ID0102E18005200066",
                    "properties": [
                        {
                            "id":"totalActiveEnergy",
                            "topic":"/Espinardo-Pool/CampusAsLivingLab_PoolPM-totalActiveEnergyExport/attrs",
                            "type":"number",
                            "attributeMqtt":"numValue",
                            "description":"Total active energy generated by the solar plant next to the pool"
                        }
                    ]
                },
                {
                    "id":"device-ID0102E18005200065",
                    "properties": [
                        {
                            "id":"totalActiveEnergy",
                            "topic":"/Espinardo-GranjaVeterinaria/CampusAsLivingLab_GranjaVeterinariaPM-totalActiveEnergyExport/attrs",
                            "type":"number",
                            "attributeMqtt":"numValue",
                            "description":"Total active energy generated by the solar plant installed in Veterinary Farm"
                        }
                    ]
                }
            ]
        }
    ],
    "includeAdditionalTimestamp":false,
    "additionalTimestampAttribute":"TimeInstant"
}
```

To summarize, the idea is to *upload* values received in MQTT publications to certain topics, and with them update a property of an entity inside a specific *fiware-service*. On the other hand, notifications coming from certain selected entities inside one or more *fiware-service* will be consumed and will produce MQTT publications.

3.4.4. Integration of additional data sources

The integration of ENTSO-E has been designed in a modular level so that it is easy to choose the countries to get data from. With this approach, and knowing that this data will be used in the PoC after the project is finished for Spain, the integration will be deployed in two separate locations:

- In the *External Integration Server* for Spain.
- In the MASTERPIECE platform for the other countries.

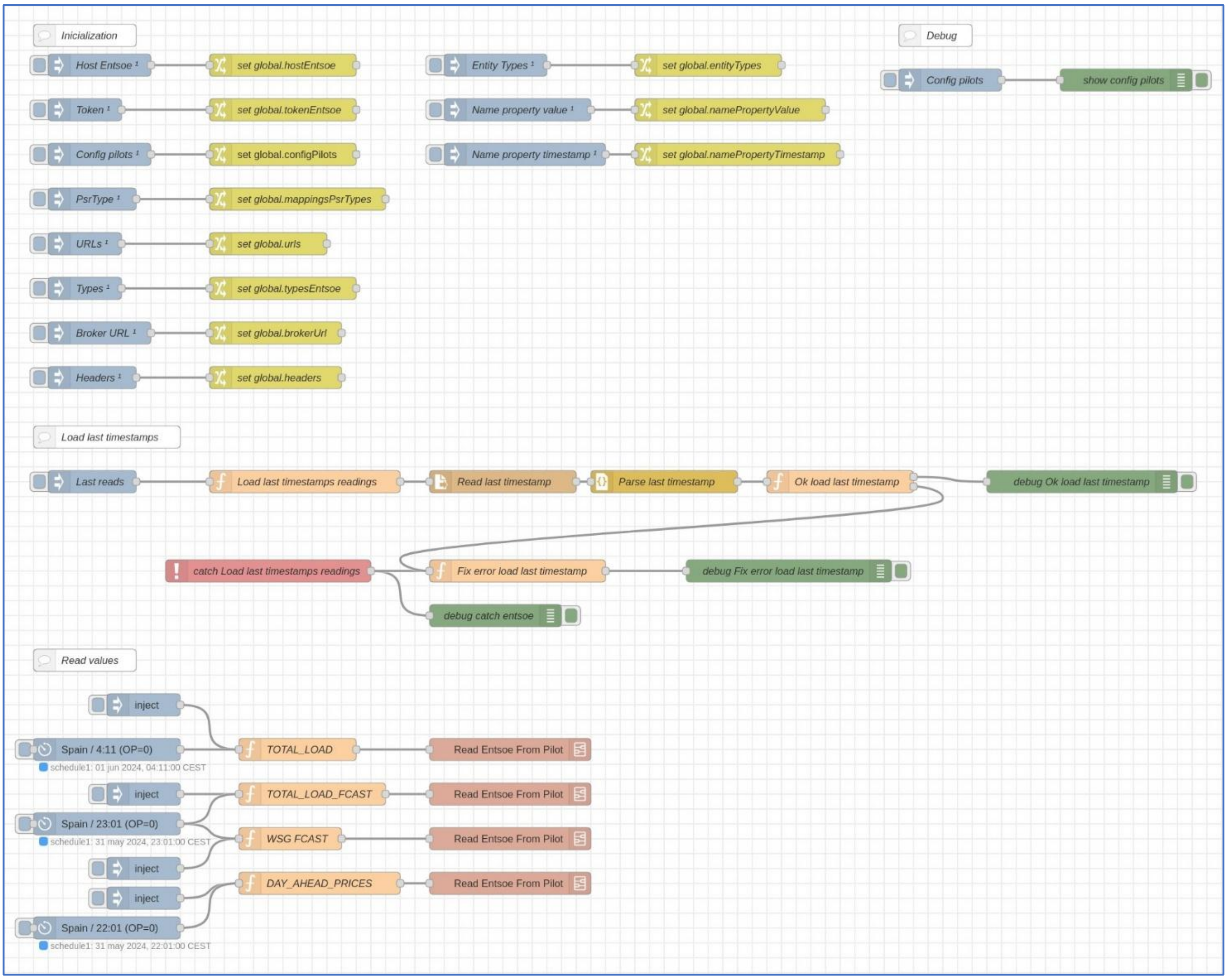

Figure 18: ENTSO-E integration

When it comes to add more countries, the block of nodes from the bottom-left of the previous figure must be cloned (one block per country) and the configuration must include the additional information, but overall is very simple to extend it.

4. EXECUTION PLAN

The *Execution Plan* is a fundamental tool to ensure continuity in the implementation of MASTERPIECE project interventions and pilot tasks, as described in D5.3.

One of the key components of this plan is the creation of a strategy for early execution of the tasks in the PoC. The idea is to use the information obtained from the PoC as reference for the integration and configuration of other pilots. This approach is expected to standardise and optimise the practices adopted, facilitating the replication and adaptation of implementations in different contexts.

This guide not only provides a comprehensive overview of the planned tasks and their interdependencies, but also describes metrics for evaluating performance and ensuring that project objectives are effectively achieved. By implementing rigorous monitoring, you can quickly identify any deviations from the original plan and make the necessary adjustments to ensure quality and sustainability of the process.

This *Execution Plan* therefore is not just an operational tool, but a strategic document that guides all phases of the project related to integration, demonstration and evaluation, ensuring the coherence and effectiveness of the proposed actions.

In this sense, this section presents the global roadmap of the MASTERPIECE project tasks connected to WP5 (*Global Roadmap of WP5*), as well as an adapted/reduced version of the same roadmap for three target stakeholders, namely component developers (*Roadmap for Component Developers*), pilot partners (*Roadmap for Pilot partners*) and integrators (*Roadmap for Integrators*). With this approach, the idea is to have a global view of the stages involved in this WP and other tasks related to it, but at the same time, to produce a set of *guides* to clarify which stages each stakeholder is involved in, like a sort of wizard. In these adapted roadmaps, the grey-task boxes are not applicable to the selected stakeholder but appear to provide an overview of the flow and dependencies.

Additionally, the flow of tasks carried out in the PoC has been highlighted using arrows in red colour. Since the PoC has a specific role in the project as a reference pilot which will be used for obtaining knowledge and experience that will be reused in other pilots, most tasks have been finished in the PoC before the expected execution month for them in the rest of the pilots.

At this stage of the project, only a portion of the *Execution Plan* is available, with the most relevant information extracted from the know-how obtained in the PoC. However, it will be extended once the components are more mature and both Component Developers and Pilot partners have a clear view of what is expected to be carried out during the trials, that is why definition of the trials is a critical task.

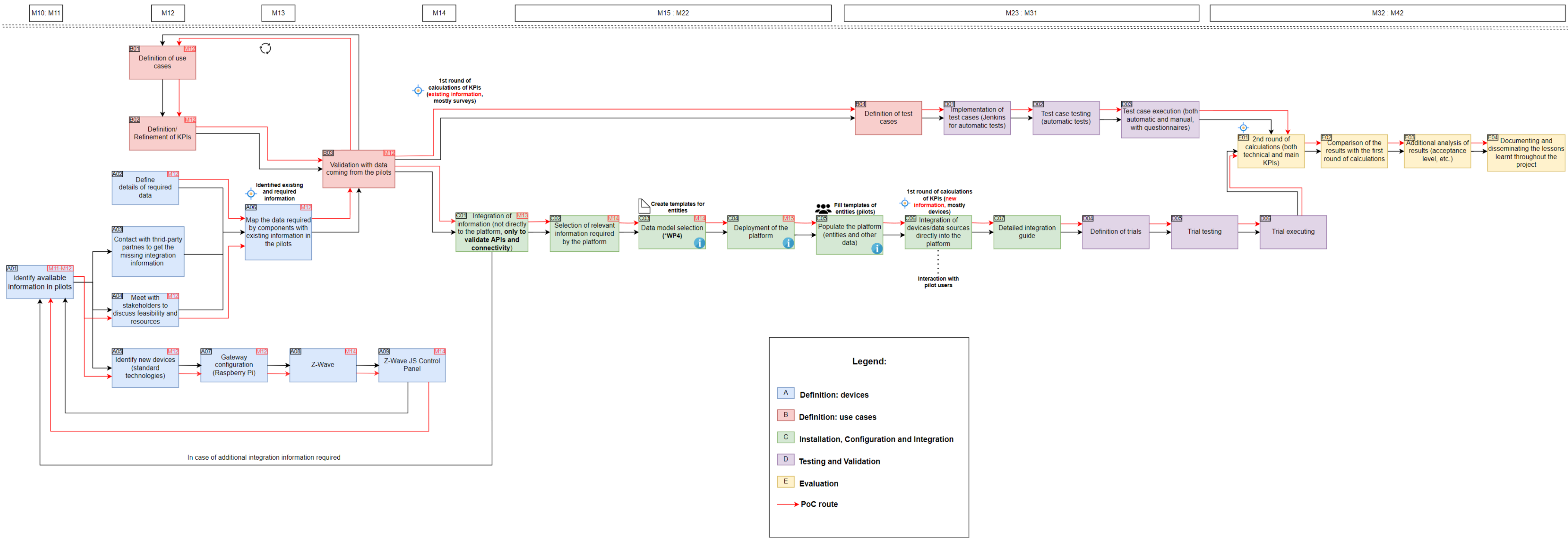

Figure 19: Global Roadmap of WP5

Each task is represented by a box identified by a letter followed by a number in the top left-hand corner, and the timeline follows a chronological order. The letter M precedes the number of months since the start of the project in the top bar.

Some tasks have potential iterative dependencies (e.g. B01, B02 and B03) and each task box has a colour, representing different phases in the broader context of this Work Package:

- The blue-coloured boxes indicate the phase of identifying existing and required devices.
- The red-coloured boxes refer to the definition phase, but this time they refer to information about use cases.
- The green-coloured boxes refer to installation, configuration and integration.
- The purple-coloured boxes represent the tasks carried out in the testing and validation phase.
- The yellow-coloured boxes indicate the tasks present in the evaluation phase.

4.1. Description of tasks

This section describes the high-level tasks involved in the different phases of the project and gives a general description of each one including the **identification code** of the task in the roadmap, a **description**, including its purpose and objectives, the stakeholder **responsible** for carrying it out, potential **risks**, **mitigation strategies** and the **PoC tasks/activities** (only those really relevant have been included not to provide useless information).

4.1.1. Identity available information in pilots

1. **Task identification code**: A01
2. **Description**: The objective of this task is to identify device-related information available in the pilots. It is in this task where the first overview of the existing assets is obtained.
3. **Responsible**: Pilot partners.
4. **Risks**: Delay in communication and availability of data.
5. **Mitigation strategies**: Constant contact promoted by follow-up meetings, e-mail exchanges and the establishment of clear timetables and deadlines.
6. **PoC activities:** A visit was conducted to the PoC with the objective of identifying all available information. During this visit, it was detected that in the *Aulario Norte Building* many monitored rooms stopped reporting on 01/02/2024. In the *Giner de los*

Ríos Lecturing Building there were many monitored rooms with good historical data (meaningful values) but stopped reporting on 01/02/2024 and in the *General Library* there was one sensor partially operational, with bad data (many cuts in the historical data series and not reporting since 01/02/2024).

4.1.2. Define details of required data

1. **Task identification code**: A02
2. **Description**: This task is focused on the definition of the details of the information required from the devices or data sources in general. The detailed description of the inputs and outputs of all the components will be used to check whether it can be provided by the pilot data sources, model the data and analyse where the information will be stored (out of the different platforms of the project, including the *DLT Secure Platform* and other existing ones).
3. **Responsible**: Component Developers.
4. **Risks**: Delay in definition and delay in subsequent tasks.
5. **Mitigation strategies**: Clear definition of the descriptions and objectives of the components agreed in previous deliverables, like D4.1.
6. **PoC activities:** Among the many existing assets in the PoC, most of them are distributed in different platforms and that makes complicated to integrate all the information. Handling this migration is one of the most relevant changes that have been triggered to provide access to data in a unified format, as described in D4.3 and in the ***Integration*** section.

4.1.3. Contact with third-party partners to get the missing integration information

1. **Task identification code**: A03
2. **Description**: This task consists of contacting the third-party partners (municipality, local provider, Distribution System Operator – DSO, etc.) if applicable to the pilot in question to find out how to access information on the equipment to be integrated, if not already provided in task A01. With this, the integration information available in

the pilot depending on this third-party partner will be obtained and used in subsequent tasks.

3. **Responsible**: Pilot partners.
4. **Risks**: Delay in contacting partners, obtaining missing information and delay in subsequent tasks.
5. **Mitigation strategies**: Follow-up meetings with partners and local integrators, and continuous communication to inform them of the status of the task.
6. **PoC activities:** Related to the previous point, a number of meetings were organized with the people responsible of the systems operating in the PoC to gather all the technical information required to do the integration.

4.1.4. Meet with stakeholders to discuss feasibility and resources

1. **Task identification code**: A04
2. **Description**: This task consists of involving all the stakeholders of the pilot to check the feasibility and means of obtaining information if it has not been made available in task A01.
3. **Responsible**: Pilot partners
4. **Risks**: Delay in contacting stakeholders, obtaining missing information and delay in subsequent tasks.
5. **Mitigation strategies**: Follow-up meetings with stakeholders and continuous communication to report on the status of the task.
6. **PoC activities:** UMU's Maintenance Team was contacted to check viability of the integrations due to some restrictions in the access to certain existing systems.

4.1.5. Identify new devices (standard technologies)

1. **Task identification code**: A05
2. **Description**: This task consists of identifying new devices based on the information previously provided in the task *Identify available information in pilots* (A01). If applicable, also considers the input from tasks *Contact with third-party partners to*

get the missing integration information (A03) and *Meet with stakeholders to discuss feasibility and resources* (A04).

3. **Responsible**: Pilot partners and Integrators.
4. **Risks**: Incorrect or partial identification may result in the delay of subsequent tasks. There is always a risk that no compatible devices can be found.
5. **Mitigation strategies**: Follow-up meetings with pilots to constantly check available information.
6. **PoC activities**: Two Z-Wave kits were chosen to cover the missing requirements with each one of them including a USB stick, a *Raspberry Pi model 4*, a power supply, a microSD card and several temperature sensors. On the other hand, in the context of being able to count the amount of people who enter/exit one of the locations in the pilot, a people-counting camera was also included in the list.

4.1.6. Map the data required by components with existing information in the pilots/PoC

1. **Task identification code**: A06
2. **Description**: This task consists of mapping the data required by the components and the data found in the pilots, using the information received by task A02, and A03, A04 when applicable. In this regard, this task will compare both datasets as well as the input information required by the components that is not provided by the pilots based on what was reported in D4.1. With this, the *Definition: devices* phase will be finished.
3. **Responsible**: Component Developers.
4. **Risks**: Incorrect or partial identification may lead to an invalid definition of Use Cases in other tasks.
5. **Mitigation strategies**: Constant checking of the information provided by pilot partners and component developers.
6. **PoC activities:** After collecting the information available in the PoC for all categories of members (individuals such as students and professors, who utilize university facilities, and the University buildings themselves, comprising a significant component of the community), the information required by the components and existing in the PoC was mapped for later validation.

4.1.7. Gateway configuration

1. **Task identification code**: A07
2. **Description:** In case a new Gateway needs to be installed, the proper configuration must be set. This configuration depends on the chosen technology and the selected device, which can be an embedded IoT Gateway, a *Raspberry Pi*, a micro-computer, etc.
3. **Responsible:** Integrators.
4. **Risks**: New devices need to be properly configured and maintained, otherwise data can be lost.
5. **Mitigation strategies**: Ensure that the Gateways are periodically monitored in order to detect any potential problem as soon as possible.
6. **PoC activities:** The Gateways installed in the PoC are *Raspberry Pi model 4* and include the software required to install the integration agents locally. These Gateways can be accessed directly for maintenance purposes.

4.1.8. Z-Wave

1. **Task identification code**: A08
2. **Description:** The integration of Z-Wave on top of any Linux-based system (such as a *Raspberry Pi* [5]) is based on *Node-RED* and the *Z-Wave JS* plugin [8], a library that provides access to a Z-Wave network.
3. **Responsible:** Integrators.
4. **Risks**: Since Z-Wave is a wireless protocol, it is important to test the quality of signal before installing the Gateways to ensure that the devices are reachable, but external noise sources and even changes in climate condition can alter the signal putting some devices out of range.
5. **Mitigation strategies:** Try to install as many Gateways as possible and make sure they have direct vision with the devices (if possible).
6. **PoC activities:** The installation of the Z-Wave devices is right now in progress, although the Gateways are fully configured.

4.1.9. Z-Wave JS Control Panel

1. **Task identification code:** A09
2. **Description:** In terms of configuration, the *Z-Wave JS* library provides a way to easily identify the parameters required to configure each entry that will be mapped to one entity in the system.
3. **Responsible**: Integrators and Pilot partners.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.
6. **PoC activities:** The template of the PoC is partially ready and includes all the entries linked to the Z-Wave devices. With this, the configuration files required by the library have already been created.

4.1.10. Definition of use cases

1. **Task identification code:** B01
2. **Description:** This task is associated with WP2, more specifically T2.3, and is focused on the definition of detailed use cases for each pilot. Since an early and high-level version set of *scenarios* was introduced in earlier stages of the project, an initial knowledge base will be available by the time both component developers and pilot partners start defining them.
3. **Responsible**: Component Developers and Pilot partners
4. **Risks:** Delays in the definition of use cases.
5. **Mitigation strategies:** Monitor these delays and mark clear deadlines.
6. **PoC activities:** Four use cases were defined and can be found in D5.3.

4.1.11. Definition/ Refinement of KPIs

1. **Task identification code:** B02
2. **Description:** This task is associated with WP2, more specifically T2.4 where it will analyse and define the KPIs for each use case.

3. **Responsible:** Component Developers and Pilot partners.
4. **Risks:** Delayed or highly subjective definition, which may require multiple iterations and jeopardise the progress of subsequent tasks.
5. **Mitigation strategies:** Be clear about the objective of the use case defined in the previous step and define KPIs that are consistent with the characteristics and limitations of the use case and the pilot.
6. **PoC activities:** Based on the objective of each use case and the tools provided by the component developers, specific KPIs were defined for each of the four use cases defined previously. More detailed information can be found in D5.3.

4.1.12. Validation with data coming from the pilots

1. **Task identification code:** B03
2. **Description:** This task is the intersection of phases *A (Definition: devices)* and *B (Definition: use cases)* and refers to the validation of information from the pilots to confirm that all the data required to measure the defined KPIs is available. If the validation cannot be completed, a new iteration is required to re-evaluate the set of use cases and KPIs.
3. **Responsible:** Component Developers.
4. **Risks:** Errors or delays in validating information which could affect the progress of subsequent tasks.
5. **Mitigation strategies:** Constant alignment with component developers and partner pilots.
6. **PoC activities:** After defining the four use cases, their respective KPIs and mapping the information required by the components and existing in the pilots, the information from the PoC was validated without the need for an additional iteration to refine new use cases and KPIs. At this stage, only device-related assets have been identified.

4.1.13. Definition of test cases

1. **Task identification code:** B04

2. **Description**: As described in D2.3, test cases are designed to help component developers and pilot partners validate the components before they are used in the trials. A preliminary version of test case template was included in this deliverable.
3. **Responsible:** Component Developers.
4. **Risks:** Definition of tests cases that cannot really be tested.
5. **Mitigation strategies:** Make sure the components provide the defined testing interfaces and are reachable.

4.1.14. Integration of information (not directly to the platform, only to validate APIs and connectivity)

1. **Task identification code:** C01
2. **Description:** This is the first task of phase C (*Installation, Configuration and Integration*), focused on the validation of the APIs received, to confirm that the devices can be integrated, and is a fundamental preliminary stage so that later, in task C06, the integration of data sources can be carried out directly on the Platform.
3. **Responsible:** Integrators.
4. **Risks:** Delay in receiving data for integration, which can derive in late availability of the data.
5. **Mitigation strategies:** Alignment and constant follow-up meetings with component developers and pilot partners.
6. **PoC activities:** The data sources involved had already been integrated in previous projects, including internal BMS, portals and devices compatible with certain standard technologies (Z-Wave). As a result, this stage was not really necessary.

4.1.15. Selection of relevant information required by the platform

1. **Task identification code:** C02
2. **Description:** This task consists of identifying and selecting the information considered relevant by the Platform from all the available assets of the pilots.
3. **Responsible:** Integrators

4. **Risks:** Delays in identifying and selecting data, slowing down the progress of subsequent tasks.
5. **Mitigation strategies:** Alignment and constant follow-up meetings.

4.1.16. Data model selection

1. **Task identification code**: C03
2. **Description:** In this task, the templates for definition of entities (pilots) will be created and validated by the integrators. These data models will include at the beginning information focused on assets available in the pilots (mostly devices) but will be later extended to include social data, results produced by components, etc. This task is directly connected to D4.3.
3. **Responsible:** Component developers and Integrators
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.
6. **PoC activities:** This task was carried out in two stages at PoC. The first one was related to using all the know-how acquired in previous projects to use Smart Data Models [9] for some well-known/existing sources and during the second, additional information was introduced to align them to the specific needs of Energy Communities.

4.1.17. Deployment of the platform

1. **Task identification code:** C04
2. **Description:** Task that involves deploying the platform used in the MASTERPIECE solution.
3. **Responsible:** Integrators.
4. **Risks:** If the deployment is delayed, no data will be available for components.
5. **Mitigation strategies:** Mark strict deadlines to guarantee the platform is ready in time and provide the required information for developers to be able to access the data (APIs, examples, etc.).

4.1.18. Populate the platform (entities and other data)

1. **Task identification code:** C05
2. **Description:** Once the platform has already been deployed in the previous task C04, in this task the pilot partners will fill part of the templates related to their assets, the component developers will add other data of interest to the platform (required by the components) and the integrators will incorporate all this information to the platform.
3. **Responsible**: Pilot partners, Component developers and Integrators.
4. **Risks:** Incorrect data in the templates can generate inconsistencies which could be difficult to fix once the components start working with the information.
5. **Mitigation strategies:** Validate the information included in the templates to guarantee it is consistent.

4.1.19. Integration of devices/data sources directly into the platform

1. **Task identification code:** C06
2. **Description:** The objective of this task is to integrate the devices and other data sources directly into the platform. Once this process is finished, pilot partners will have enough information to run the first round of KPI calculations for those KPIs related to these integrations.
3. **Responsible:** Pilot partners and Integrators.
4. **Risks:** Unstable data streams can complicate the calculations of KPIs, which are critical for evaluation of the pilots.
5. **Mitigation strategies:** Deploy the monitoring tools in parallel with the integrations.

4.1.20. Detailed integration guide

1. **Task identification code:** C07
2. **Description:** The preparation of an integration guide, a product included in this deliverable (D5.1), will facilitate and speed up future integrations by presenting the necessary tasks to be followed and the lessons learnt.
3. **Responsible:** Integrators
4. **Risks:** No risks identified.

5. **Mitigation strategies:** Not applicable.

4.1.21. Implementation of test cases (Jenkins for automatic tests)

1. **Task identification code:** D01
2. **Description:** This task represents the initial effort of Phase D (*Test and Validation*), where test cases are implemented using the Jenkins automation server, which allows for seamless integration with other tools, scalability, comprehensive system analysis, identifying not only immediate problems but also areas for potential improvement, and more.
3. **Responsible:** Component developers.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.

4.1.22. Test case testing (automatic tests)

1. **Task identification code:** D02
2. **Description:** This task entails the automation of test case execution with the objective of ascertaining the software's conformity to the established specifications and its overall functionality.
3. **Responsible:** Component developers.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.

4.1.23. Test case execution (both automatic and manual, with questionnaires)

1. **Task identification code:** D03
2. **Description:** This task consists of both automated and manual test case execution, as well as the use of questionnaires to collect feedback on the user experience, the usability of the software, or to obtain additional information on possible faults or improvements. The aim of this task is to verify compliance with the requirements and

improve the user experience. As a result, in some cases, the pilot partners will participate when it comes to validate the components from an end-user point of view.

3. **Responsible:** Component developers and Pilot partners.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.

4.1.24. Definition of trials

1. **Task identification code:** D04
2. **Description:** Once phase C (*Installation, Configuration and Integration*) has been completed, a set of trials must be defined to detail how the MASTERPIECE solution is going to be evaluated in each pilot.
3. **Responsible:** Component developers and Pilots Partners.
4. **Risks:** A late or incorrect definition of the trials could invalidate the whole evaluation phase.
5. **Mitigation strategies:** Trials must be ready in time and must be properly revised.

4.1.25. Trial testing

1. **Task identification code:** D05
2. **Description:** This stage involves validating in controlled environments the previously defined trials in task D04, either in laboratory or in the pilots.
3. **Responsible:** Component developers.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.

4.1.26. Trial executing

1. **Task identification code:** D06
2. **Description:** At this task, the trial is conducted in accordance with the previously defined procedures and results obtained in the preceding task, D05. The efficacy, safety, and performance of the subject matter are validated through this process.

3. **Responsible:** Component developers and Pilots Partners.
4. **Risks:** No risks identified.
5. **Mitigation strategies:** Not applicable.

4.1.27. 2nd round of calculations (both technical and main KPIs)

1. **Task identification code:** E01
2. **Description:** It is the first task of phase *E (Evaluation),* when the second round of calculation of the main technical KPIs is carried out, based on the data provided by the execution of the Test case and Trial execution, in phase D (*Testing and Validation*).
3. **Responsible:** Component developers and Pilot partners.
4. **Risks:** Delay in obtaining key information for KPI calculations.
5. **Mitigation strategies:** Constant alignment and follow-up meetings.

4.1.28. Comparison of the results with the first round of calculations

1. **Task identification code:** E02
2. **Description:** This task consists of comparing the results obtained by the KPI calculation methodologies in the first round of calculations and in the second round carried out afterwards. In this way it will be possible to evaluate the difference between the results and understand the success rate of the MASTERPIECE solution.
3. **Responsible:** Pilot partners.
4. **Risks:** Not identified.
5. **Mitigation strategies:** Not applicable.

4.1.29. Additional analysis of results (acceptance level, etc.)

1. **Task identification code:** E03
2. **Description:** In this task, further analyses will be conducted with the objective of providing a comprehensive evaluation of the MASTERPIECE solution, encompassing social, environmental, technical and economic impacts. To this end, questionnaires

will be utilised to ascertain the perceptions of stakeholders and energy communities regarding the acceptance and usability of social innovations and digital tools provided by the MASTERPIECE solution prior to the project's completion.

3. **Responsible:** Pilot partners.
4. **Risks:** Delays in carrying out analyses.
5. **Mitigation strategies:** Constant alignment and follow-up meetings.

4.1.30. Documenting and disseminating the lessons learnt throughout the project

1. **Task identification code:** E04
2. **Description:** The objective of this task is to document and disseminate the lessons learnt during the project to the public as a reference document for the formation of new energy communities.
3. **Responsible:** Pilot Partners and Integrators.
4. **Risks:** Not identified.
5. **Mitigation strategies:** Not applicable.

4.2. Roadmap for Component Developers

This section includes the *Roadmap for Component Developers*, which outlines the tasks that are appropriate for Component Developers.

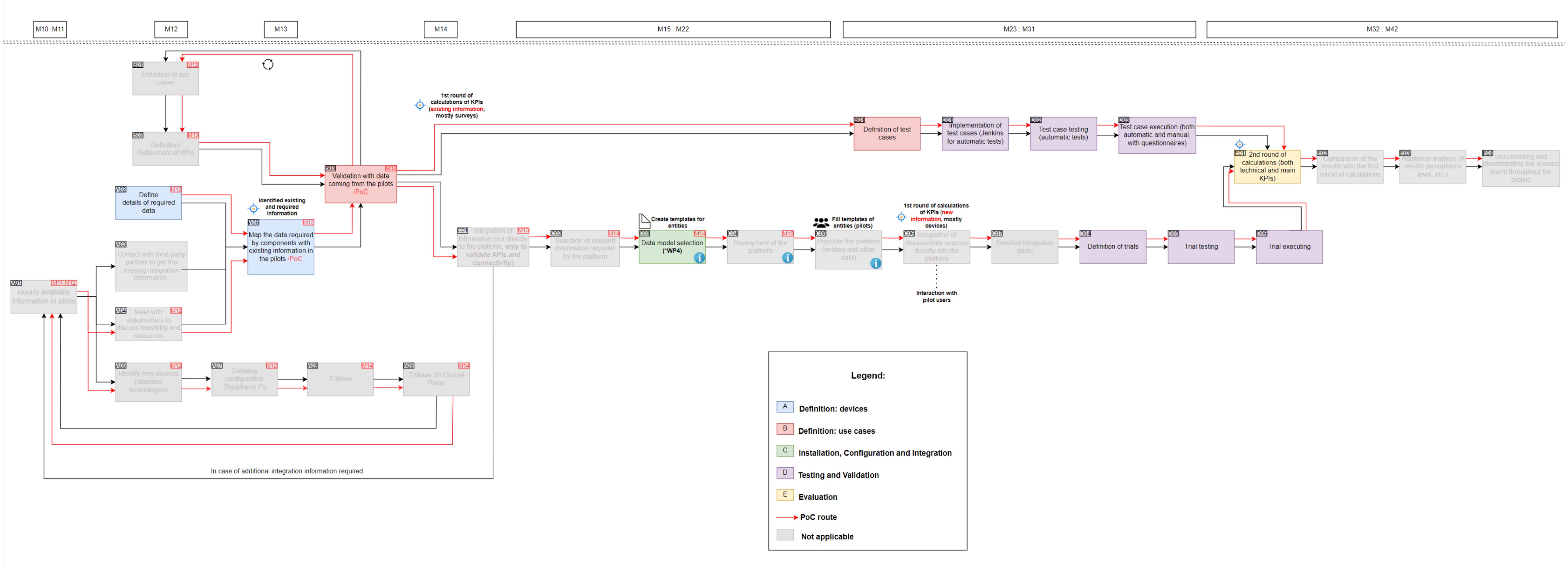

Figure 20: Roadmap for Component Developers

4.3. Roadmap for Pilot partners

This section includes the *Roadmap for Pilot partners,* which outlines the tasks that are suitable for Pilot partners.

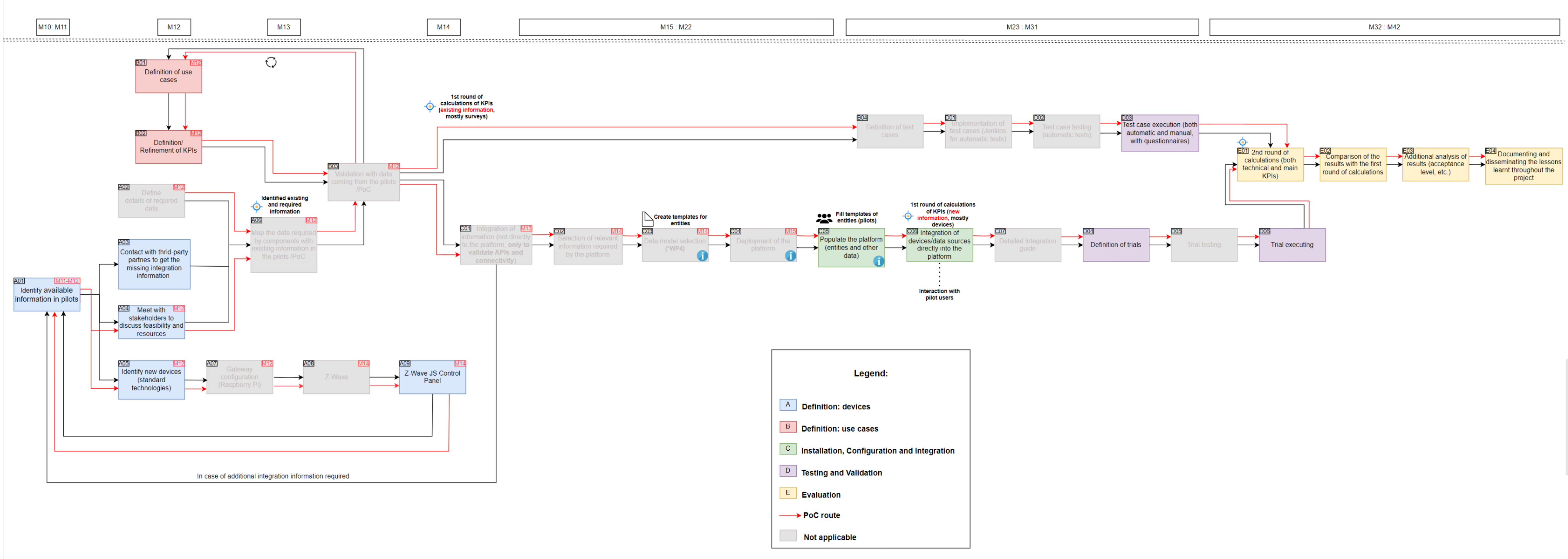

Figure 21: Roadmap for Pilot partners

4.4. Roadmap for Integrators

This section includes the *Roadmap for Integrators*, which outlines the tasks that are suitable for Integrators.

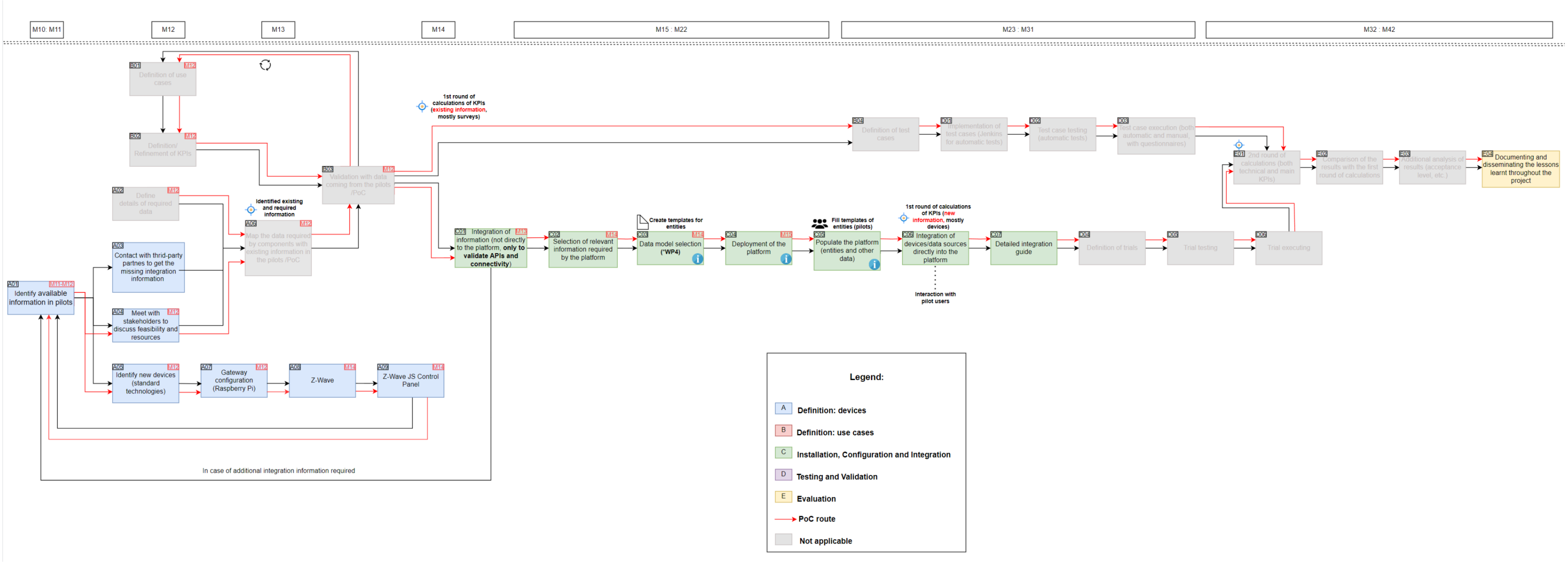

Figure 22: Roadmap for Integrators

5. MONITORING AND VERIFICATION

The monitoring of the performance and availability of the technological elements is a critical aspect to be handled in the project.

In this regard, several items have been identified and will be under the radar of the monitoring process, such as software components, integrations agents, etc. In addition, a number of selected data series will also be monitored to ensure that the data is being received on a regular basis and detect (and notify) potential communication problems that might occur.

5.1. Jenkins automation tool

One of the tools studied for testing services is Jenkins. Jenkins is an open-source server for testing and/or continuous integration. It is a cross-platform tool written in Java, accessible through a web interface.

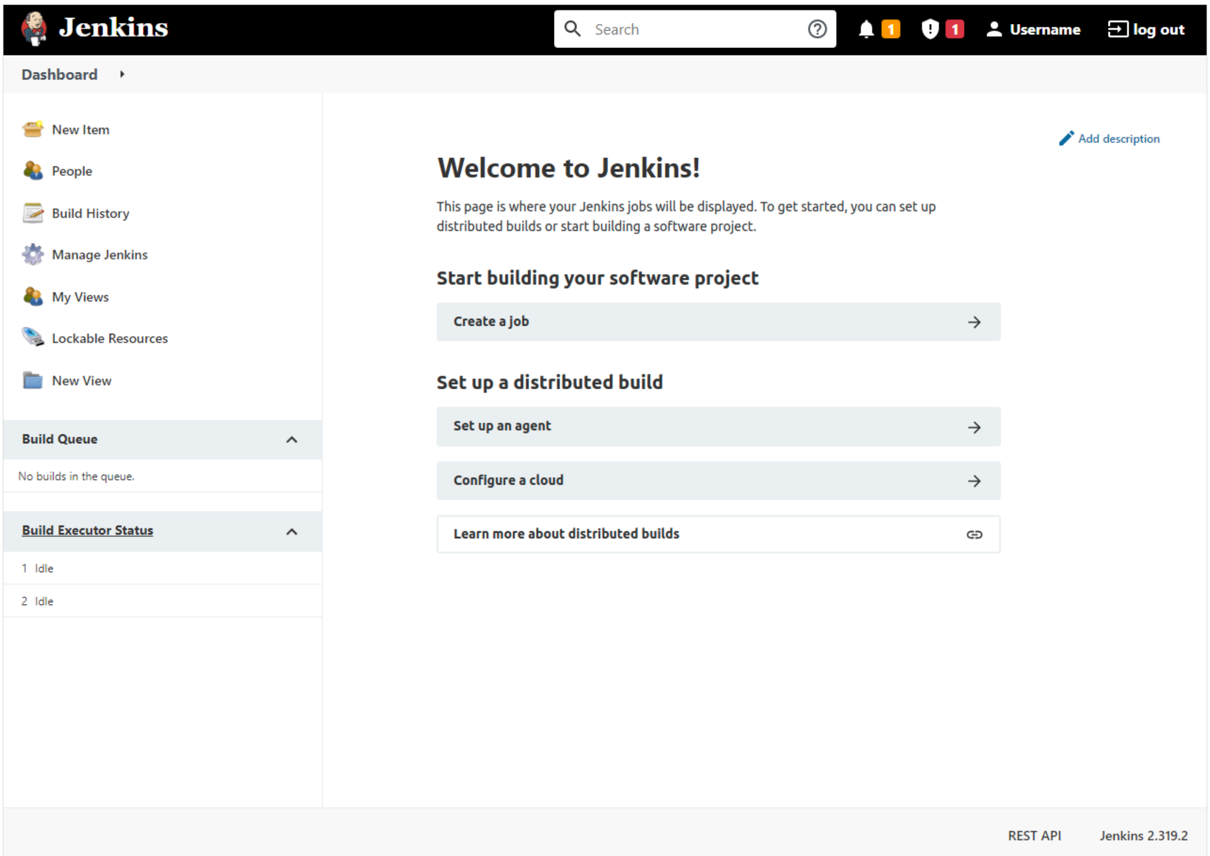

Figure 23: Jenkins frontend

By default, Jenkins offers a set of built-in utilities which offer certain functionality, but can also be extended by installing additional plug-ins, including improvements in the UI, administration tools, integration with Kubernetes, ssh, GitHub, etc.

5.1.1. Main functionality

Out of the many options provided by Jenkins, there are a few of them especially relevant as they are enough to implement fully operational tests:

- Job. A job is a task that can be executed either periodically or triggered by an event, such as the end of another job.
- Pipeline. A pipeline is a type of job that’s formed by a set of ordered stages.
- Stage. A stage is a subset of the tasks of a pipeline. The criteria used for dividing a pipeline in stages depend on the characteristics of the global task. In the end it’s just a logical division.
- Step. Each stage is divided as well in one or more steps. A step is the smallest unit that has its own integrity in a pipeline and its purpose is to perform a very specific operation. Groovy is the language in which the different steps of a pipeline are coded. It could be defined as a sort of simplified and modified version of Java. However additional libraries can be installed for using other languages when coding steps such as python, perl, etc.

Using the frontend, jobs can be created using a wizard.

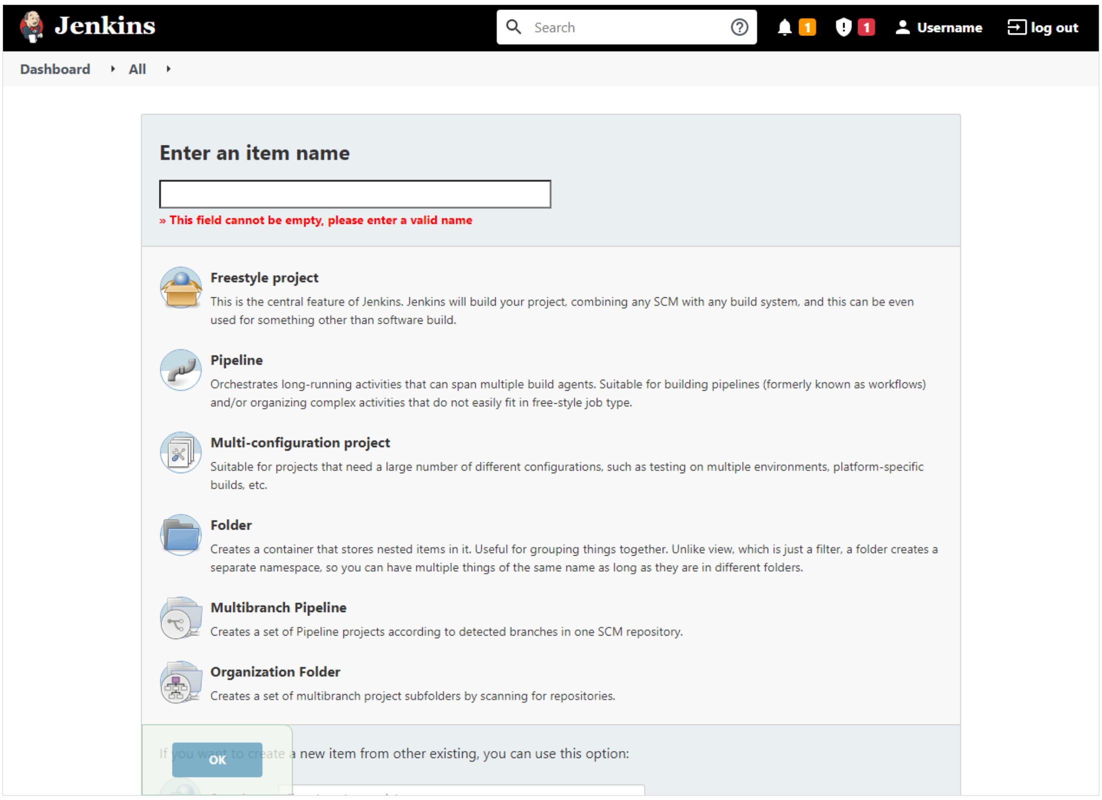

Figure 24: Creating a job

During the job creation, the two most important items that must be configured are the scheduling and the code of the pipeline itself.

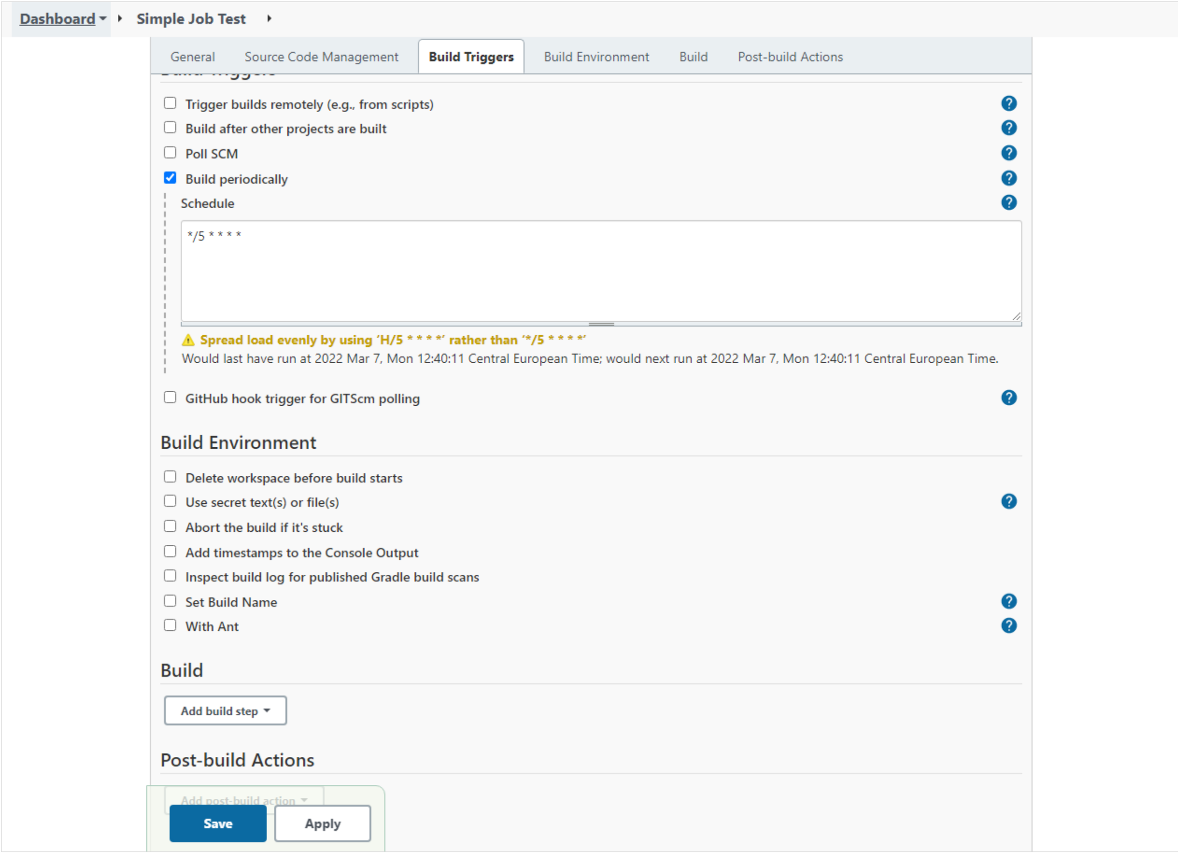

Figure 25: Configuration of a job

For programming the execution of the test, assuming it is going to be launched periodically, Jenkins uses a syntax based on the one used by cron but slightly extended.

The code of the pipeline is configured in the pipeline section and can be defined by either adding steps one after another or by pasting directly the full content of a Jenkinsfile as introduced in the next section.

5.1.2. Jenkinsfile

A Jenkinsfile includes all the code of a pipeline. It provides access to environment variables, definition of stages and steps, etc.

```
pipeline {
    agent any

    stages {
        stage('Service1') {
            environment {
                status1 = """${sh(
```

```
                        returnStdout: true,
                        script: 'curl -i --location --request GET \'http://IP1:PORT1/\' | grep \"HTTP/1.\"
| cut -d \' \' -f2'
                    ).trim()}"""
            }
            steps {
                script {
                    if (status1 == "200") {
                        echo "Service 1 Status: ${status1}"
                        emailext body: """Service 1 Status: ${status1}""", subject: "Simple Pipeline Test
- Service 1 Status", to: "email@mail.com"
                    } else {
                        error "Error status1: ${status1}"
                    }
                }
            }
        }
        stage('Service2') {
            environment {
                status2 = """${sh(
                        returnStdout: true,
                        script: 'curl -i --location --request GET \'http://IP2:PORT2/\' | grep \"HTTP/1.\"
| cut -d \' \' -f2'
                    ).trim()}"""
            }
            steps {
                script {
                    if (status2 == "200") {
                        echo "Service 2 Status: ${status2}"
                        emailext body: """Service 2 Status: ${status2}""", subject: "Simple Pipeline Test
- Service 2 Status", to: "email@mail.com"
                    } else {
                        error "Error status2: ${status2}"
                    }
                }
            }
        }
    }
}
```

As can be seen in the previous sample, direct calls to scripts or command line utilities such as *curl* are supported and, if the scripts are called in the **environment** section, the return value can be used to trigger actions (send emails, launch a script that could update an entity in the *DLT Secure Platform*, etc.) or detect an **error** (that will be logged) and abort the execution of the test.

However, the preferred way to access to web resources is by using the *HTTP Request Plugin*, which offers a high-level vision of HTTP requests and simplifies the integration of any REST API.

5.2. Key Elements for Monitoring

In this context, key elements have been identified that are essential to the correct functioning of the MASTERPIECE solution. These elements not only support the technical infrastructure, but also ensure that all operational aspects are monitored and optimised.

The main elements that need to be monitored are:

- **Portals** (whenever it's possible). This group includes Scadas and BMS used for collecting data from various pilots (mostly from the PoC so far), as well as the *UEDAS AMR System* (REST API) in the case of the Turkish pilot, which has already been integrated.
- **Components**. In the context of evaluation, in the evaluation plan defined in D2.3, *Test cases* were introduced as a way to validate the different components. The idea is to reuse if possible the endpoints offered for testing to monitor them as well, although this strategy will be discussed in the future with component developers.
- **Adapters** (whenever it's possible). The adapters will be monitored and the methodology to be used in each case needs to be chosen based on the nature of each adapter (some of them can also provide monitoring endpoints, other will be controlled through Docker [10] logs, etc.).
- **Data series** (periodically, to detect data losses as a result of connectivity problems, for example). The periodicity will be calculated empirically since it is fully dependent of the involved data source (i.e. some series will be provided by battery-powered devices that do not have real-time communication support).

5.3. Verification

The monitoring process will therefore be based on periodic execution of Jenkins' pipelines in an automated way. By constantly monitoring performance, health and availability metrics, Jenkins provides valuable insight into the health of the system without direct user intervention. This passive/synchronous approach allows anomalies to be detected early, so that preventative action can be taken to avoid disrupting the development process.

The verification process is a combination of this passive automated monitoring handled by Jenkins and active verification actions launched by the user, providing a more robust and resilient development environment. This means other tools will be prepared to check in-depth the status of the system in an attempt to detect other anomalies which had passed the controls executed during the monitoring phase. In the end it is important to find a proper balance between reliability and scalability, and in order to do that, it is important to control the amount of overhead derived of the monitoring.

An example related to this *balance* between reliability and scalability has to do with the monitoring of data series. The monitoring pipeline will be executed periodically several times per day (to be confirmed). With this, and based on the entities that are being monitored, since the *Real-Time Data Repository* that is part of the *DLT Secure Platform* only stores the last value and timestamp of each series, the pipeline will be in charge of detecting whether the last update of each entity is not too much in the past. Still, the fact that this validation is passed does not mean that there was no data loss between two consecutive checks, and that is why this additional verification will access the *Historical Data Repository* to analyse the complete series in detail.

Once the integrations progress and more data are available, a proper *Verification Plan* will be created and the different tools will be deployed. This plan will be reported in D5.2.

Figure 26: Diagram of the relationship between Monitoring and Verification

6. Lessons learnt from the Proof of Concept

A number of lessons have been learnt so far from the PoC and can be forwarded to the other pilots:

- When it comes to try to get access to assets of the pilots and public institutions are involved, it is very important to monitor closely the evolution of the process. In many of these scenarios, certain permissions need to be obtained and it is very common to have to deal with unexpected delays caused by an extra level of bureaucracy that is not so common in other cases (private sector).

- Related to the previous one, if new equipment needs to be installed in public institutions, sometimes multiple quotes from different distributors/installers are required by the public regulations, and this can derive in additional delays.
- Even if it requires some additional efforts during the integration phase, it is highly recommended to use standard/unified data formats. In the case of the PoC, all the integrations have been adapted to a format compatible with the FIWARE MQTT/JSON Agent which is generic enough to allow an easy mapping of the data to different data models depending on the requirements of the project.
- Whenever it is possible, try to rely on standard/open technologies if new devices need to be installed. In the PoC, most of the assets were already installed but still new ones were required, and most of them are compatible with Z-Wave, a standard protocol with a wide community offering software and documentation.
- If possible, try to use remotely-manageable IoT Gateways.

7. CONCLUSIONS

During this document, two clear lines have been analysed in detail. Firstly, the role of the PoC as special pilot in the project, which provides valuable information to other pilots. It was chosen for this purpose based on its unique characteristics, with a significant amount of equipment already available and with a lot of experience from the responsible partners (UMU and ODINS) in integrations and testing thanks to interventions done in the past in the context of other research projects. For these reasons, it has proved to be a suitable choice for a PoC.

On the other hand, this deliverable is directly aligned with T5.1, which covers another key topic to be addressed, the *Execution Plan*, which started as generic description of tasks, but became a good base for a final plan that will be presented in the next deliverable thanks to the lessons learnt from the PoC. In this regard, certain important stages of the project have been highlighted. Additional efforts will be made once the project reaches the point of defining trials, which are one of the main reference results required at the late stages of the evaluation process.

In addition, monitoring and verification have been marked as two critical activities in the path to succeed in the integration. It is important to notice that they must be planned and executed if possible in parallel with the integrations, as well as once the system is fully operational. To address these needs, a *Verification Plan* will be prepared and launched in the following months.

Another subject that will be detailed in future updates is the integration of social data. At this point of the project, with this Work Package having started in month 10, most of the efforts have been focused on the integration of digital data sources (devices) while work on social innovations and digital tools is being carried out in parallel in WP3 and WP4 respectively. Once they are mature enough, their information will be incorporated to the knowledge base obtained from the work done on WP5 to provide a complete view of the MASTERPIECE solution.

8. REFERENCES

[1] Universidad de Murcia, "Campus as Living Lab" HTTPS://CAMPUSLIVINGLAB.INF.UM.ES

[2] ENTSO-E Transparency Platform HTTPS://TRANSPARENCY.ENTSOE.EU/

[3] FIWARE / ORION HTTPS://FIWARE-ORION.READTHEDOCS.IO/EN/MASTER

[4] Z-Wave HTTPS://WWW.Z-WAVE.COM/

[5] Raspberry Pi HTTPS://WWW.RASPBERRYPI.COM/

[6] Node-RED HTTPS://NODERED.ORG/

[7] Node.js HTTPS://NODEJS.ORG/EN

[8] Z-Wave JS HTTPS://GITHUB.COM/ZWAVE-JS

[9] Smart Data Models HTTPS://SMARTDATAMODELS.ORG

[10] Docker. HTTPS://WWW.DOCKER.COM

[11] Jenkins HTTPS://WWW.JENKINS.IO/