Getting Started Guide

Grant Agreement No.	873087
Project Name	Smart Human-Oriented Platform for Connected Factories (SHOP4CF)

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 873087. Neither the European Commission nor any person acting on behalf of the Commission is responsible for how the following information is used. The views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect the views of the European Commission.

Table of Contents

List of Figures

Abbreviations

RAMP	Robotics and Automation Marketplace
SHOP4CF	Smart Human Oriented Platform for Connected Factories
SME	Small and Medium enterprises
QA	Quality assurance
SCM	Software Configuration Management
IoT	Internet of Things
URN	Uniform Resource Name
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
NGSI	Next Generation Service Interfaces

1 Introduction to the Getting Started Guide

Welcome to the Getting Started Guide for RAMP which is a part of the EU-funded project SHOP4CF. To accommodate you as a user of RAMP, this guide is meant to help and guide you through the steps to acquire the necessary knowledge that you need to be able to use the platform. The guide will introduce you to the RAMP platform, FIWARE, and the installation- and upload of a component.

2 Introduction to RAMP

RAMP is a platform specifically designed for Small and Medium size Manufacturers (Manufacturing SMEs). It enables Manufacturing SMEs to collect and analyze factory data, ensuring that you fully understand your factory processes and how you can optimize them. It allows you to make informed decisions about your next investment in automation technologies and see their full impact. RAMP is based on FIWARE open technology, ensuring you are always free to choose the next best automation solution for your factory. In the platform, you can find data visualization solutions, a marketplace of technology providers, and a community of manufacturing SMEs.

- The data visualization solutions are designed to help you gain awareness of your processes through an improved overview of your factory's data. This in return should help enable you to address possible issues within the production line.

- The marketplace is a meeting point for professionals that are looking to connect and discover specialized partners and state-of-the-art solutions that will level up their businesses. Navigate through numerous technology providers that offer a variety of useful tools and services.

- The community of manufacturing SMEs makes it possible to connect with fellow SMEs that can share their experience and explore all the next steps for automating your business.

2.1 Quality assurance for RAMP components

For software components and solutions to be made available on the RAMP marketplace, it is essential that certain Quality Assurance (QA) procedures are standardized and followed. This will allow developers to freely contribute to the RAMP marketplace and will also ensure that added components can be utilized easily by the system integrators to create reliable solutions for factory automation. The basic business model is that component suppliers get paid by the system integrators while system integrators charge the price to the end-user of the solution. The current business model assumes that transactions only take place once with a fixed price and does not assume pay-per-use, annual fees, or other revenue streams.

Below are the first set of QA requirements, which shall not be considered the final set. The purpose of these requirements is to align the existing tools and components with the framework of the RAMP marketplace so the components can be utilized in the Experiments carried out in various projects.

The requirements will be further detailed, elaborated, and added on the following two bases:

(i) Align the components with the framework of the RAMP marketplace
(ii) Feedback received from the Manufacturing SMEs, Technology Suppliers, and System Integrators during the Experiments

As a starting point, these requirements are derived from the existing set of FIWARE requirements given at:
https://fiware-requirements.readthedocs.io/en/latest/GE_Requirements/index.html.

The URL can be accessed for further details. However, certain requirements differ due to differences in FIWARE and RAMP's operational models.

2.1.1 General:

- All components shall have a seamless integration with the FIWARE NGSI, which is currently FIWARE NGSIv2 and will be compliant with ETSI NGSI-LD in the future.

2.1.2 Licensing:

- All closed-source components must specify the license to use the component.
- All open-source components shall be released under Apache License 2.0 (Apache-2.0).

2.1.3 Software Configuration Management (SCM):

- All closed-source components must use the RAMP GitHub repository.
- All open-source components must use the RAMP GitHub repository.

2.1.4 Documentation:

- All components must provide the following documentation:
  - readme.MD: It should tell a summary of what the component does, why someone should use it, and how it can be used.
  - api.md
  - architecture.md
  - getting-started.md
  - installationguide.md
  - user manual. MD

2.1.5 Testing:

- All components shall include a suite of functional tests to verify the proper integration with FIWARE Orion Context Broker.

2.1.6 Software release:

- All components shall be tagged with a semantic release number. Every release shall have a unique version number.
- Every release shall have an associated Release Notes entry in the GitHub Releases section.

2.1.7 Integration:

- All components shall have a continuous integration system.

2.1.8 Tracking:

- A Tracking system shall be used for all components to manage the development work. Such a tracking system shall include at least all your component's bugs/known issues.

2.1.9 Code quality:

- All open-source components must meet the coding standards defined in the GitHub repository.

2.1.10 Binary release:

- All components shall be made available on the RAMP marketplace as a docker image.
- The availability of the latest docker image on the RAMP shall be automated.

3 Introduction to FIWARE

Digital solutions in smart industry gather data from many different sources, and process and analyze it to provide smart features. To accelerate strategic and operational activities, the data must be shared among components of the industrial solution. FIWARE provides an enhanced OpenStack-based cloud environment and a set of open standard APIs that make it easier to connect to the Internet of Things, process and interpret Big data and real-time media. The platform provides data spaces for effective cross-domain data sharing and exchange. This enables a consumer to gather data from a variety of devices or sources, without worrying about the low-level IoT connection or data protocol.

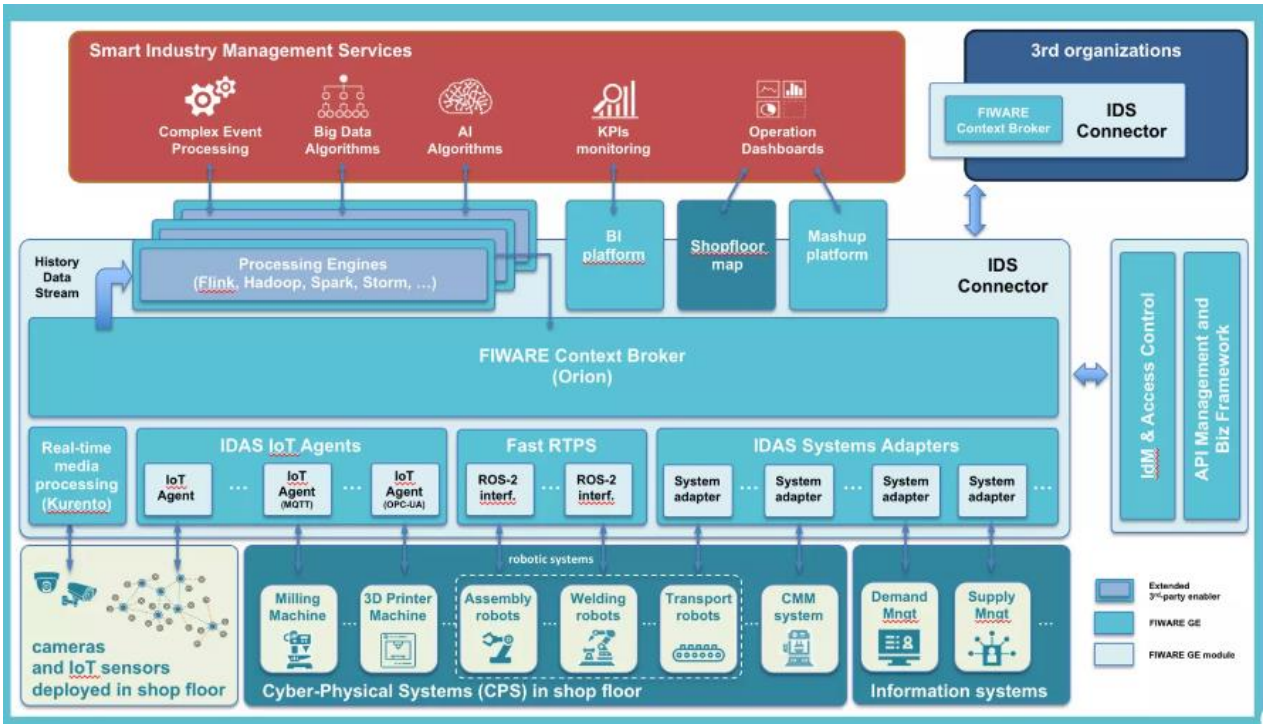

Figure 1: FIWARE Smart Industry Architecture (source: www.fiware.org)

Figure 1 shows the architecture of a FIWARE framework showing the four main components:

i. Context broker
ii. Interfacing with the Internet of Things (IoT), Robots and third-party systems
iii. Context Data/API management
iv. Processing, analysis, and visualization of context information.

Among the four components FIWARE solutions must implement the core Context Broker component, that enables us to manage context information in a highly decentralized and large-scale manner. The rest of the components are optional. FIWARE NGSI (Next Generation Service Interfaces) API enables the easy implementation of the context broker, by providing means to produce, gather, publish, and consume context information at large-scale. The Orion Context Broker currently provides the FIWARE NGSI v2 API based on standard REST APIs.

3.1 FIWARE Data Models

FIWARE follows a set of standardized data models for interoperability among components. The data models contain a set of attributes, their definitions and their data types that can be combined into a single entity. Each Data Model is programmatically defined using a JSON

Schema with a key-value representation of FIWARE NGSI format. There are two leading versions of the format: NGSI v2 and NGSI -LD.

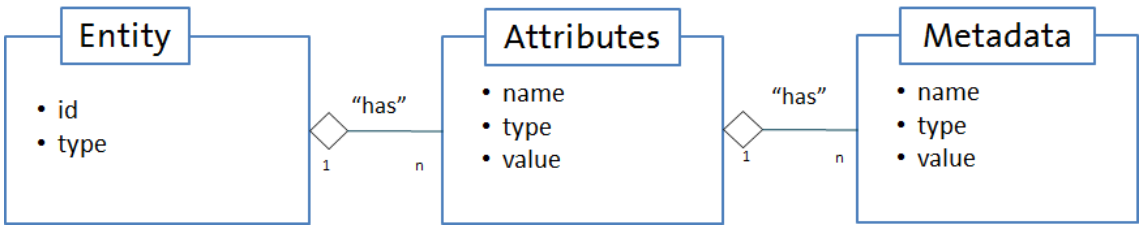

Figure 2: FIWARE NGSI v2 Data Model (source: here)

3.2 NGSI v2 Data Model

The main elements in the NGSI v2 data model are:

i. Context entities
ii. Attributes
iii. Metadata

A context entity represents a physical or a virtual object (e.g., a sensor, a person, a room, a manufacturing task to be executed, etc.). Each entity has an entity id and an entity type, following the type system of FIWARE NGSI. The entity id and the type jointly identifies an entity uniquely. Example: a context entity with id sensor-365 and type temperatureSensor. Context attributes are properties of context entities. Each attribute consists of an attribute name, an attribute type, an attribute value, and metadata. For example, the current speed of a car could be modelled as attribute current_speed of entity car-104. The optional metadata field describes properties of an attribute value e.g. accuracy, provider, or timestamp. The metadata can also be defined through a dedicated field Context Metadata, which consists of metadata name, metadata type and a metadata value.

A sample NGSI v2 data model is described below.

```
{
  "id": "entityID",
  "type": "entityType",
  "attr_1": <val_1>,
  "attr_2": <val_2>,
  ...
  "attr_N": <val_N>
}
```

3.3 NGSI-LD Data Model

NGSI-LD is an evolved version of FIWARE NGSI v2 to better support linked data (entity relationships), property graphs, and semantics. Unlike NGSI v2, the main components of NGSI-LD are:

i. Entity
ii. Property
iii. Relationship

The property attributes represent the context data that changes over time. The relationship attributes represent the connections between existing entities using linked data.

NGSI-LD API is defined using the JSON-LD specification to resolve conflict among data attributes coming from different sources. JSON-LD introduces the concept of the @context element which can be used to assign unique long URIs for every defined attribute. The @context element provides additional information allowing the computer to interpret the conflicting data with more clarity and depth. Some of the main differences between NGSI v2 and NGSI-LD are that Entity IDs are URIs (URLs or URNs), the metadata is represented using nested Properties of Properties.

An example of Migration from NGSI v2 to NGSI-LD is shown in Fig.3.

Figure 3: Example of Migration from NGSI v2 to NGSI-LD (source: here)

3.4 SHOP4CF Data Models

SHOP4CF architecture is based on FIWARE Smart Industry architecture shown in Figure 1. The Data Models used in SHOP4CF are based on FIWARE NGSI-LD Data Models, but are customized.
Specifically, an entity identifier in the SHOP4CF data model should be an Uniform Resource Name (URN), built as:

Build
urn:ngsi-ld:<entity-type>:<factory-id>:<entity-id>

Example
urn:ngsi-ld:Device:company-xyz:sensor-abc-12345.

The SHOP4CF Data Models are divided into two main groups:

i. Design data models
ii. Execution data models

Design Data Models define entities that do not change their status during the manufacturing process, e.g., the shop-floor locations, definitions of manufacturing processes, task definitions, etc.
Execution data models are the entities that may change their status during manufacturing. Some examples are tangible resources on a shop floor, tasks instance under execution, alerts, etc.

For more details, please refer to section 8 of the SHOP4CF Architecture, and a few sample data models belonging to these two groups can be found on the SHOP4CF GitHub page.

4 Installation of component

The following installation of a component is a general guideline on how to install a component, therefore please beware of specific installation requirements for the specific component you are downloading and installing.

4.1 Install Docker container for FIWARE context broker

To install a RAMP component, the server must have FIWARE running correctly, meaning it must be properly installed and functioning without issues.

You can find a script to set up FIWARE in the following repository:
https://gitlab.lst.tfo.upm.es/shop4cf/fiware-deployment

To use the script, follow the steps:

- Clone the repository
- Open an Ubuntu terminal inside the directory of the cloned folder
- Execute the installation file
- Check that the Context Broker is working properly by going to http://localhost:1026/version

4.2 Previous steps before installing component Docker image

Once FIWARE and the Context Broker are running on your server, it is time to start with the component's installation. Every RAMP component includes specific instructions for proper operation, including specific requirements and software installations that you may need to complete before starting to use the component. Be sure to read the available documentation before beginning the installation process. All the components installation process follows these simple steps:

Install a RAMP component:

1. Access the RAMP Catalogue: Go to the component catalog in the RAMP marketplace (https://ramp.eu/#/component/catalogue).

2. Choose the component: Select the component you wish to use from the component catalog.

3. Check for customization: Review the component's documentation/repository to determine if any customization is necessary.

4. Download Docker Image: If no customization is required, download the component's Docker image from the RAMP catalog and proceed to step 7.

5. Customize component (if necessary): If customization is necessary, follow the component's documentation or reach out to the developers for assistance.

6. Create Docker Image: Create the component's Docker image with your customizations.

7. Install Docker: If not already installed, download and install Docker (https://www.docker.com/).

8. Upload Image to Docker: After installing Docker, upload the component's Docker image using the command "docker load -I [image name].tar".

9. Run Image: Run the component's Docker image using the command "docker run [image name]".

10. Verify Installation: Verify the successful installation of the Docker image by using the "docker ps" command to check that the image is running, and all containers are active.

4.3 Install component Docker image

1. Download the component Docker image: If no customization is needed, download the Docker image of the component from the catalog in RAMP and go to step 4.

If any adaptation is necessary:

2. Adaptation of the component: Follow the documentation of the component or contact the developers to adapt the component if possible.

3. Docker image: Create the Docker image of the component with the adaptations you have made.

4. Install Docker: If you do not already have Docker installed on your system, you must download and install Docker before continuing (https://www.docker.com/).

5. Upload the image to Docker: Once Docker is installed, you must upload the image to Docker. You can do this using the command "docker load -I [image name].tar".

6. Run the image: After loading the image into Docker, you must run the image. You can do this using the command "docker run [image name]".

7. Verify the installation: Finally, you must verify if the installation of the docker image was successful. You can do this by using the "docker ps" command to verify that the image is running and that all containers are active.

5 Upload of component

Promote your organization by uploading components to RAMP.

Upload a component to RAMP

1. Register an account on RAMP and create your organization by clicking on your profile name and clicking on the 'My organization' tab.

2. In the 'My organization' tab click on 'create a new organization'.

3. When you have access to your organization click on 'components' and click on 'add component'.

4. The create a new component guides you through what must be filled out to upload your component by clicking 'next step' and finally clicking 'save'.

    a. In the 'General information you must fill out the following:
        i. Upload an image (an image that promotes your component).
        ii. Component title.
        iii. Summary.
        iv. Applicable industries.
        v. Description.
        vi. (Optional) A favorite keyword that categorizes your component when searching for your component.
            1. To create a new favorite keyword, simply type the keyword and hit enter.

    b. Now click 'next step'. In the next step, you can fill out the optional fields, video, and key Hardware Software Dependencies, when finished click 'next step'.

    c. Next step has four fields where only one must be filled out which is 'Select licenses' and the last three are optional: Executable, Source-Code, and Docker image.

    d. The last step is 'Technical details' where you can provide optional information for Documentation, Interfaces Data input and output, and Key Hardware Software Dependencies. The only mandatory field is the type of component that you are uploading which needs to be specified by the drop-down menu. When finished click 'Next.

5. In the 'Characterization' tab, add predefined keywords to improve component description and discoverability. Choose keywords from 'Support workers' and 'Support tasks' categories (refer to Appendix 7.1). Click 'Save' when done.

6. Your component can now be found in the '3rd-party' tab under 'Software' and your 'My organization' 'Components'.

5.1 Edit of a component on RAMP

Managing and editing your components on RAMP is easy.

Managing your components

1. Make sure you are logged in on your profile and click on your profile name.

2. Click on the desired organization 'components' tab to see and manage your organization's components.

3. In the organization's components you can edit and delete your components.

Edit a component

1. Make sure that you are logged in on your profile and click on your profile name.

2. Click on the desired organization 'components' tab to see and manage your organization's components.

3. Click on the desired component your wish to edit by clicking 'edit'.

4. You can now edit your component by following the same steps as when you upload a component and to save your component click on 'next step' and 'save'.

5.2 Upload of Docker Image to RAMP

Uploading your Docker Image to RAMP, please use the following link and remember to have a login to access the site:

Docker Registry (ramp.eu)

5.3 Create a Docker Image

First time using Docker Image to share your component, please use the following link to help guide you through the process:

https://docker-curriculum.com/#docker-images

It is a complete guide to how you get started with Docker Image provided by Docker.

5.4 Getting started with FIWARE

To get started with FIWARE, please use the following link to help guide you through the process:

https://github.com/FIWARE/tutorials.Getting-Started

The guide covers an introductory tutorial to the FIWARE platform.

6 References

- FIWARE. “The Open Source Platform for Our Smart Digital Future.” *FIWARE*, www.fiware.org/.
- Cantera Fonseca, José Manuel, et al. “Fiware-Ngsiv2-2.0-2018_09_15.” *Fiware.github.io*, 2018, fiware.github.io/specifications/ngsiv2/stable/. Accessed 14 Feb. 2023.
- FIWARE. “NGSI-LD How to - Fiware-DataModels.” *Fiware-Datamodels.readthedocs.io*, fiware-datamodels.readthedocs.io/en/stable/ngsi-ld_howto/index.html.
- Francisco Carvajal, Diego. “SHOP4CF / FIWARE Deployment.” *GitLab*, 2022, gitlab.lst.tfo.upm.es/shop4cf/fiware-deployment. Accessed 14 Feb. 2023.
- Srivastav, Prakhar. “A Docker Tutorial for Beginners.” *A Docker Tutorial for Beginners*, docker-curriculum.com/#docker-images. Accessed 14 Feb. 2023.
- Fox, Jason, et al. “Architecture.” *GitHub*, 5 July 2022, github.com/FIWARE/tutorials.Getting-Started.
- Aromaa, S., Heikkilä, P. and Kuula, T. (2022) “D2.6_Guidelines for human-centered manufacturing environments 1.”

7 Appendix

7.1 Appendix 1. Description of characteristics

In the 'Characterization' tab, you can select the predefined keywords from two categories:

- Support workers; including keywords e.g., mastering complexity, solving problems, working safely (etc.)
- Support tasks: including keywords e.g. designing, planning, assembling (etc.)

The keywords related to supporting workers are based on framework of Future Industrial Worker characteristics (FIW). The FIW framework includes four main characteristics describing future work and worker: smart, resilient, interactive, and healthy, and each of these include four sub-categories. It is beneficial that component developers familiarize themselves with the FIW framework to understand the background of the keywords they are selecting. See table 1.

Table 1: FIW framework - Description of characteristics

SMART
• **Complexity master:** Ensuring smooth production by monitoring work processes and systems and managing dependencies and priorities between the operations. • **Problem solver:** Preparing to solve problems quickly and patiently by utilising evolved practices or guidance proactively created for problem situations. • **Proactive decision maker:** Considering information from different sources when making decisions for example about the operations or maintenance of machines. • **Sustainability oriented:** Considering sustainability when making daily decisions, for example related to materials, waste, and maintenance of machines.
RESILIENT
• **Creative**: Seeking for new solutions to challenges, which may lead to improved work engagement or new innovations. • **Self-leader**: Guiding oneself to achieve work objectives, for example by taking initiative, organising work tasks, and following progress. • **Flexible**: Seeing changes as opportunities and seeking ways to adapt work tasks to new requirements or unexpected changes. • **Continuous learner**: Acquiring new knowledge and competences to expand skills, which benefits current work tasks as well as fosters up- and re-skilling if needed.
HEALTHY
• **Motivated**: Willingness to make an effort to achieve good results, which is shown as energy, commitment and persistence when completing work tasks. • **Balanced**: Feeling balanced of one's work load and responsibilities as well as balancing requirements of professional and personal life, which fosters well-being and recovery between work shifts. • **Capable**: Having sufficient resources, skills, and capabilities to complete the work tasks without compromising work ergonomics. • **Focused**: Being able to concentrate on the ongoing work task, staying aware of the environment but not getting distracted too easily.
INTERACTIVE

- **Communicative**: Openly sharing task related information, plans, ideas, and potential challenges with others.
- **Collaborative**: Fluently collaborating with team members as well as with interactive technology, such as collaborative robots.
- **Inclusive and intercultural**: Valuing diversity and including all workers with different backgrounds (e.g., genders, ages, cultures and nationalities) in joined activities.
- **Safety oriented**: Following and developing safety procedures and prioritizing safety when completing work tasks.

(Aromaa et al., 2022)

Instructions:

1. Think about your component from the end-user/worker point of view and describe what kind of benefits it provides for the workers.

2. Read through the FIW framework categories and select those characteristics your component supports in the work

3. Select suitable keywords based on the FIW framework (tab 'Support workers'). Additionally, select keywords on work tasks ('Support tasks').

4. Describe shortly (1-2 sentences) how your component supports each of the selected characteristics by clicking on the comment tab