



**Total Resource and Energy Efficiency
Management System for Process Industries**

Deliverable **5.2**

Final MAESTRI Platform Architecture Design & Specification

Date: 28/02/2017

WP5 IoT Platform Development

T5.1 Architecture Design

Dissemination Level: Public

Project Website: www.maestri-spire.eu

SPIRE Sustainable Process Industry through
Resource and Energy Efficiency



Total Resource and Energy Efficiency Management System for Process Industries



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 680570

The sole responsibility for the content of this document lies with the authors. It does not necessarily reflect the opinion of the European Union. Neither the EASME nor the European Commission are responsible for any use that may be made of the information contained therein.

Legal Notice: The information in this document is subject to change without notice. The Members of the project consortium make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the project consortium shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material. Possible inaccuracies of information are under the responsibility of the project. This report reflects solely the views of its authors. The European Commission is not liable for any use that may be made of the information contained therein.

Authors

Name: Gunnar Große Hovest
Organisation: ATB

Name: Kevin Nagorny
Organisation: ATB

Name: Abhijit Vyas
Organisation: ATB

Name: Alexander Schneider
Organisation: FIT

Name: Otilia Werner-Kytölä
Organisation: FIT

Name: Enrico Ferrera
Organisation: ISMB

Name: Rosaria Rossini
Organisation: ISMB

Name: Eduardo João Silva
Organisation: ISQ

Name: Cristina Garrido
Organisation: MICRO

Document history

VERSION	DATE	DESCRIPTION
0.1	20.01.2017	First draft version by ATB, extending and adjusting the previous D5.1 document
0.2	30.01.2017	Version commented by ISMB
0.3	10.02.2017	Updated all architecture figures and adjusted/extended text, taking into account comments and inputs from ISMB, FIT, ISQ, MICRO, UCAM and LEI. Removed end user scenario, as an updated version of the scenario is now part of D1.5.
0.4	20.02.2017	Updates based on comments and inputs from INEGI, FIT, ISMB, MICRO and GLN. Several changes in the figures and minor text improvements. Moved some details from section 1 to section 2.
1.0	28.02.2017	Several minor improvements of text and figures.

Internal review history

REVIEWED BY	DATE	DESCRIPTION
Karl Krone (OAS)	27.02.2017	Review by OAS, accepted with minor comments

Document details

FILE NAME	VERSION
D5.2 - Final MAESTRI Platform Architecture Design & Specification v1.0.docx	1.0

DOCUMENT OWNER	ORGANISATION
Gunnar Große Hovest	ATB

Executive Summary

The MAESTRI project aims to advance the sustainability of European manufacturing and process industries, providing a management system in the form of a flexible and scalable platform, guiding and simplifying the implementation of an innovative approach, the MAESTRI Total Efficiency Framework. The overall aim of this framework is to support the improvement of companies' environmental and economic performance.

Aiming to support the analysis and the continuous monitoring of the companies' efficiency and eco-efficiency, the MAESTRI Work Package 5 (IoT Platform development) will provide a middleware platform, based on Internet-of-Things (IoT) technology. This platform will facilitate the data transfer from machines, systems, and sensors at the industrial sites to end user software tools and applications, in particular to MAESTRI front-end applications like MSM[®] and ecoPROSYS[®].

This document presents the MAESTRI IoT Platform Architecture Design and Specification. At first, it gives an overview of the basic concepts of the IoT Platform and the envisaged relationships with existing enterprise applications, machines in the shop floor as well as software tools from other MAESTRI work packages. Secondly, the overall IoT Platform architecture is presented, explaining its functional blocks and their relationships, covering also the envisaged connections to other systems. This architecture is based on the requirements collected in work package 1 (Requirements Engineering) and on the features of the pre-existing LinkSmart middleware. LinkSmart will be re-used and extended in MAESTRI as foundation for the IoT Platform. A subset of the IoT Platform's functional modules is already covered to a certain extent by functionality of the LinkSmart middleware, while other functional modules will need to be developed as new services within MAESTRI.

It was agreed to first implement a simple prototype of the IoT Platform, which is currently under development. This prototype facilitates the communication and further alignment between the different MAESTRI work packages as well as the continuous refinement/extension of the requirements towards the platform. An initial simple application scenario was agreed among partners from different work packages, which defines the functional scope of the envisaged first prototype. Starting from the scenario's end user point of view, the requirements towards the IoT Platform were derived. The architecture of the first prototype is presented in this document, as well as technical details of the prototype currently under development, serving as a concrete detailed example implementation of the generic concepts of the IoT Platform.

4

The work results presented in this document will serve as basis for the further IoT Platform development. They will be updated and extended iteratively during the development process. Thus, although the basic overall MAESTRI architecture is considered mostly final, slight adaptations/extensions will still be done in the future. Details of the architecture will be continuously improved and refined, following an iterative process, based on the lessons learned from the IoT Platform's application in the four MAESTRI industrial case studies.

Table of contents

Executive Summary	4
List of figures	6
Abbreviations	6
1 Introduction	7
1.1 Purpose, scope of this deliverable and approach applied	7
1.2 Basic IoT Platform concept	9
1.3 Document overview	10
2 IoT Platform architecture and relationships to other systems	11
2.1 Overview	11
2.2 Existing systems	12
2.2.1 Shop Floor	13
2.2.2 Enterprise Applications	13
2.2.3 Enterprise Application Integration Services	13
2.3 MAESTRI IoT Platform	13
2.3.1 LinkSmart® LocalConnect	13
2.3.2 Platform Services	15
2.4 MAESTRI Front-End Applications	17
2.5 MAESTRI Application Back-End Services	18
2.6 LinkSmart® SDK Tools	18
2.6.1 Business Process Modeller	18
2.6.2 Domain Mapping Tool	18
2.6.3 Deployment Mapping Tool	18
3 First prototype application of the IoT Platform	20
3.1 Motivation	20
3.2 Data flow to be supported by the IoT Platform	21
3.3 Prototype architecture	22
3.4 Technical implementation example of a basic data flow	24
3.5 Further implementation plan	28
4 Conclusion	30

List of figures

Figure 1 – Relationship between WP5 and other work packages..... 7
 Figure 2 – Role of the IoT Platform as central middleware 9
 Figure 3 – Architecture of the IoT Platform and relationships to other systems..... 12
 Figure 4 – Data flow in the prototype scenario 22
 Figure 5 – Initial software prototype architecture 23
 Figure 6 – Technologies and communication protocols used for the prototype implementation 25
 Figure 7 – Interaction between different software components..... 27

Abbreviations

API	Application Programming Interface
BC	Business Case
BPMN	Business Process Model and Notation
CSV	Comma Separated Values
D	Deliverable
DB	Database
ecoPROSYS®	Eco-Efficiency Integrated Methodology for Production Systems
ERP	Enterprise Resource Planning
FTP	File Transfer Protocol
GUI	Graphical User Interface
ICT	Information and Communication Technology
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MBD	Model Based Development

MBSE	Model Based Software Engineering
MES	Manufacturing Execution System
MQTT	Message Queue Telemetry Transport
MSM®	Multi-Layer Stream Mapping
OPC-UA	Open Platform Communications – Unified Architecture
PC	Personal Computer
PLC	Programmable Logic Controller
REST	Representational State Transfer
SDK	Software Development Kit
SoA	Service-oriented Architecture
STDOUT	Standard Output
T	Task
TCP	Transmission Control Protocol
WP	Work Package
XMI	Extensible Markup Language (XML) Metadata Interchange

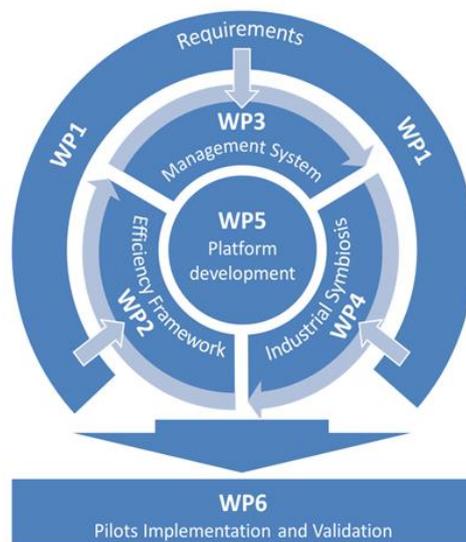
1 Introduction

1.1 Purpose, scope of this deliverable and approach applied

MAESTRI is a project dedicated to advance the sustainability of European manufacturing and process industries. It aims to provide an innovative approach in terms of a flexible and scalable platform, guiding and simplifying the implementation of an innovative approach, the MAESTRI Total Efficiency Framework. The purpose of this framework is to encourage a culture of improvement within process industries by assisting the decision-making process, supporting the development of improvement strategies and helping to define the priorities to improve the companies' environmental and economic efficiency. The Total Efficiency Framework is based on four main pillars:

- I. An effective management system targeted at process and continuous improvement;
- II. Efficiency assessment tools to define improvement and optimisation strategies and to support decision-making processes within the scope of eco-efficiency analysis and overall operational efficiency analysis;
- III. Integration with a toolkit for Industrial Symbiosis focusing on material and energy exchange;
- IV. A software platform, based on Internet-of-Things (IoT) technology, to simplify the concept implementation and provide an integrated data provision tool to support the improvement process.

This document addresses the fourth of these pillars, which is in the focus of the work package "IoT Platform development" (WP5). The objective of WP5 is to provide a concrete ICT instrument, supplying common sustainability tools with data from the shop floor and from existing enterprise applications. Figure 1 gives an overview of the basic relationships between WP5 and the other MAESTRI work packages.



7

Figure 1 – Relationship between WP5 and other work packages

As the IoT Platform should support the transfer of data from systems at industrial companies to end user tools developed in other MAESTRI WPs, the requirements of those WPs have to be taken into account. Therefore the starting point for the work in WP5 are the results of WP1 (Requirements Engineering). In order to gather the requirements that the IoT Platform – and the MAESTRI Total Efficiency Framework as a whole – should fulfil, at first the user centred requirements from industrial partners were collected through interviews and industrial visits (user requirements workshops). Besides the end user requirements, also the needs of WP2's Efficiency Framework, representing the integration of the pre-existing tools ecoPROSYS® and MSM®, had to be taken into account. Those existing tools, as well as the tools and methods planned by WP3 (Management System) and WP4 (Industrial Symbiosis) were analysed regarding their needs for data from the shop floor and from existing systems, which shall be provided via the IoT Platform. After their development and integration, the IoT Platform and the other tools of the MAESTRI Total Efficiency Framework will be implemented and validated in the pilot implementations at the industrial partners (WP6 Pilot Implementation and Validation).

Having in mind these relationships, it is clear that requirements from all of the aforementioned WPs have to be taken into account for the IoT platform development. The requirements gathered until now were documented in the deliverables D1.4 (Initial Requirements Report) and D1.5 (Lessons Learned and Updated Requirements Report). The purpose of this document D5.2 (Final MAESTRI Platform Architecture Design & Specification) is to present the results of task T5.1 (Architecture Design), which at first focused on the translation of the platform specific requirements into functional blocks of the IoT Platform. During this definition of the functional blocks, also the features of the pre-existing LinkSmart¹ middleware were taken into account, which will be the basis for the MAESTRI IoT Platform. LinkSmart was developed in the previous EU project Hydra (FP6-IST-034891) and further extended in the course of a variety of research projects. The existing LinkSmart functionality was analysed in order to define in how far the MAESTRI requirements can be fulfilled using the existing software and which extensions of LinkSmart will be needed. To cover those requirements that are not yet supported by LinkSmart, new software modules will be developed, which will be loosely coupled with existing modules in a service-oriented architecture (SoA).

Both the existing and the newly defined functional blocks were grouped and their basic relationships - among each other and with other systems - were defined, leading to the definition of the overall MAESTRI architecture, including the IoT Platform and its relationships to other systems. This architecture helps to communicate the functional specifications of the MAESTRI key components to the entire consortium and serves as basis for the subsequent platform development tasks.

8

In order to get a realistic first prototype of the IoT Platform, supporting the further discussion about additional and refined requirements, an initial pilot application scenario was

¹ <https://linksmart.eu/>

agreed among partners from different work packages. The requirements that cover the needs of this scenario are implemented first.

As the requirements engineering is an ongoing task in MAESTRI, it is expected that some additional requirements will be gathered in the future, which will also have to be taken into account. Therefore, although this document presents the final results of T5.1 Architecture Design, the design process will be continued in an iterative way to incorporate further requirements, which could appear especially during pilot implementation of the IoT Platform in the MAESTRI industrial case studies.

1.2 Basic IoT Platform concept

The main objective of the IoT Platform is to provide middleware functionality, which facilitates the data transfer from machines, systems and sensors in the industrial shop floor to end user software applications. This basic concept is depicted in Figure 2.

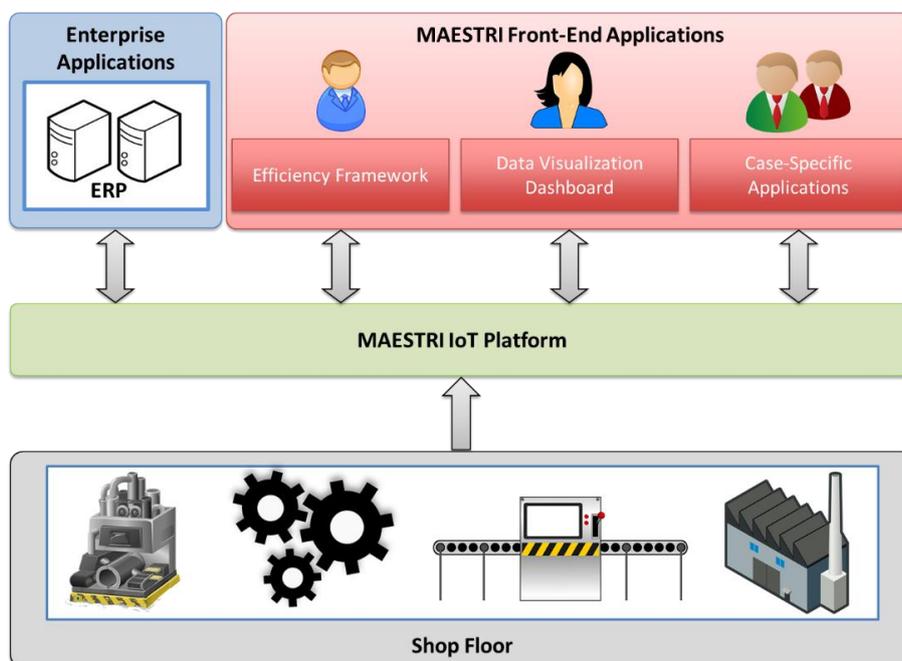


Figure 2 – Role of the IoT Platform as central middleware

From the point of view of the IoT Platform, all systems to be connected can be seen either as data sources or as data consumers:

- Data sources will be hardware and software components in the shop floor as well as (pre-existing) enterprise applications of the industrial companies. They produce data to be analysed, processed and interpreted in order to calculate KPIs relevant for the evaluation and optimization of the industrial processes.
- Data consumers will be the MAESTRI Front-End Applications, but also other software tools in the companies could require access to data from the shop floor via the IoT Platform. Therefore existing enterprise applications could possibly play both roles, i.e. they could be data sources and data consumers at the same time.

The MAESTRI Front-End Applications currently defined to be connected to the IoT Platform are MSM® and ecoPROSYS®, which will be integrated and together form the Efficiency Framework (which is the overall results of WP2). Besides these tools, there is also a requirement for a dashboard GUI that supports basic visualisation of (current and historical) shop floor data. In addition, it is expected that also other (custom developed) applications could be needed in each concrete business case pilot implementation at the MAESTRI industrial partners' sites. Therefore one of the general requirements towards the IoT Platform is to be open for flexible integration of additional end-user applications. In any case, throughout the rest of this document the term MAESTRI Front-End Applications will be used as collective term for all end user software tools/applications that will get data via the IoT Platform.

As the IoT Platform will mainly serve middleware purposes, currently just administrative user interfaces are planned. The end user in everyday operation of MAESTRI is expected to work just with the MAESTRI Front-End Applications but not to communicate directly with the IoT Platform.

To facilitate the setup of the connections between existing systems and the IoT Platform, it is planned to provide tools for Model Based Software Engineering (MBSE), enabling developers to create technology agnostic models through high abstraction procedures, hence simplifying the software life cycle management. These models will be deployed on the IoT Platform in order to support its runtime configuration and operation.

1.3 Document overview

This document is organized as follows:

- Section 2 describes the overall architecture of the IoT Platform and its relationships to other systems. This overall architecture was defined taking into account the requirements from other WPs – collected in WP1 Requirements Engineering – plus the features of the existing LinkSmart middleware;
- Section 3 describes the first prototype of the IoT Platform, which is currently under development. The architecture of that prototype consists of a subset of the components of the overall architecture from Section 2. Some technical implementation details are presented for the prototype, as well as an initial implementation plan, which will be further refined along the development in an iterative way.
- Section 4 summarizes the content of this document and gives an outlook on the future tasks that will be based on the work results presented here.

2 IoT Platform architecture and relationships to other systems

2.1 Overview

This section explains the functional software modules of the IoT Platform and their relationships to MAESTRI Front-End Applications, enterprise applications, and the shop floor. Although the requirements gathering work done in WP1 was the starting point for the architecture design, additional work was necessary to derive the initial architecture of the IoT Platform for the following reasons:

- Some of the original requirements were not detailed enough and had to be further elaborated into more concrete requirements;
- Some requirements were addressing mainly the end-user functionality to be provided by MAESTRI Front-End Applications, and so they were only indirectly leading to IoT Platform requirements (because the Front-End Applications need data to be provided by the IoT Platform).

Besides the requirements gathered in WP1, also the architecture of the existing LinkSmart middleware was taken into account, as LinkSmart is the foundation of the IoT Platform and therefore strongly influences its basic structure.

The necessary functionality of the IoT Platform was defined and grouped into functional blocks. Figure 3 shows an overview of the envisaged IoT Platform architecture, including related software components, which are seen as data sources or data consumers from the IoT Platform's point of view.

It should be noted that this section focusses on a generic architecture of the MAESTRI IoT Platform, thus the indicated connections to existing systems and devices are just examples. Concrete implementations of this IoT Platform in real industrial environments could possibly use subsets of the platform's functional blocks and could also require additional connectors to enterprise applications or shop floor installations.

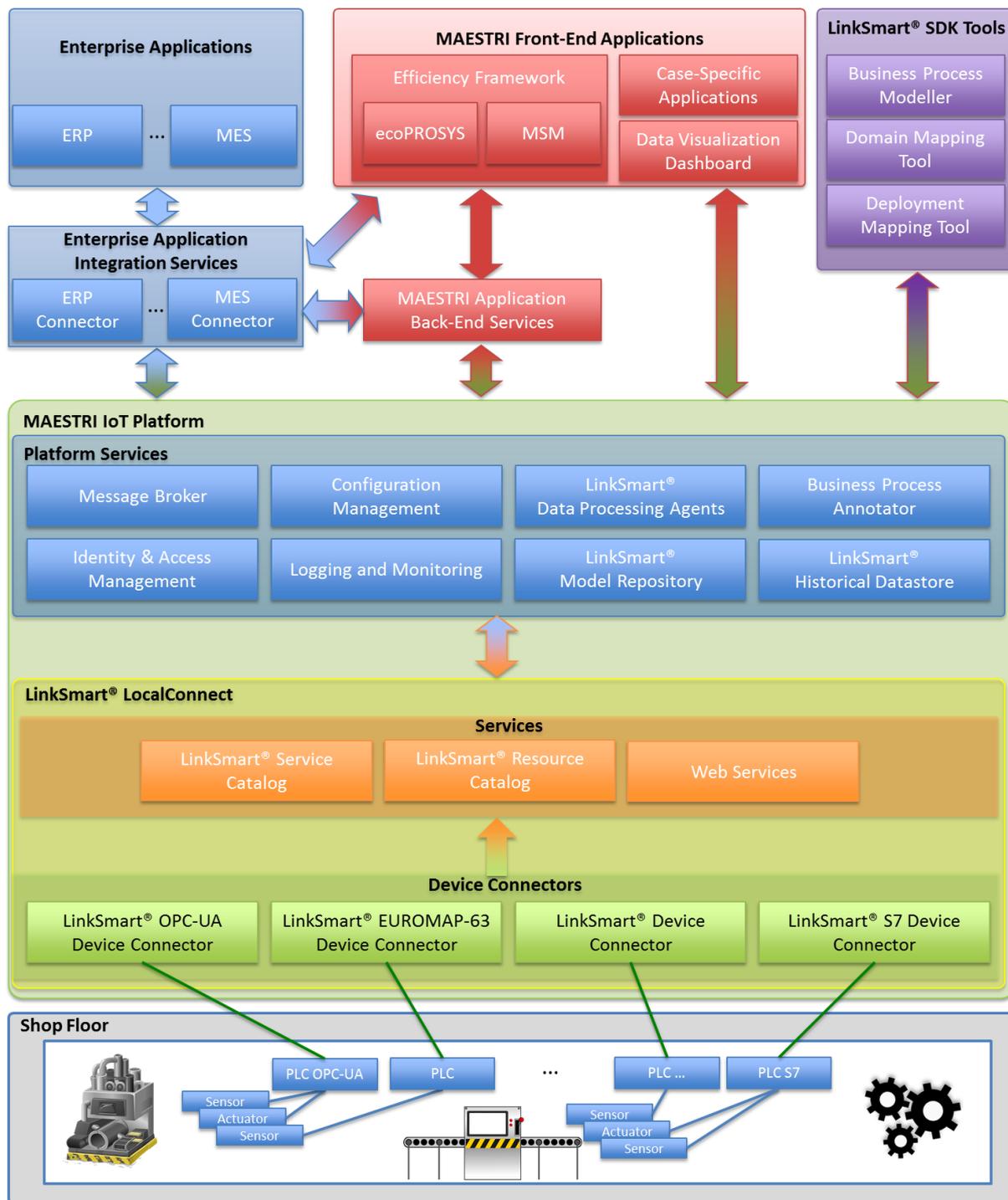


Figure 3 – Architecture of the IoT Platform and relationships to other systems

12

Each of the functional submodules of the architecture is explained in the following subsections. The description starts from the existing systems to be connected to the platform and then follows along the envisaged data flow to the MAESTRI Front-End Applications.

2.2 Existing systems

Existing systems will be connected to the platform, serving mainly as data sources but possibly also as data consumers.

2.2.1 Shop Floor

Production lines, devices and machines, sensors and actuators: usually the shop floor will be the place where the major part of the data relevant for MAESTRI is being produced. Looking at the overall architecture, there are basically two ways how such data would enter the IoT Platform: either via dedicated device connectors or via already existing connections from shop floor installations to enterprise applications. An example for the latter could be an existing MES that is already connected to a number of sensors/actuators in the shop floor. In such a situation, depending on the specific development efforts needed, it could be easier to get access to this whole “sensor/actuator network” via the MES, rather than via several individual device connectors.

2.2.2 Enterprise Applications

Enterprise applications are the second type of data source for MAESTRI. It is expected that especially ERP and MES systems as well as energy management systems will be connected to the IoT Platform in order to complement the data from the shop floor. As already mentioned above, some enterprise applications like MES systems could themselves be connected to the shop floor, i.e. they could represent and be a proxy for whole sensor/actuator networks towards the IoT Platform.

Besides being data sources for MAESTRI, the enterprise applications could also receive data from the shop floor via the IoT Platform, therefore – depending on the specific scenario – also a bi-directional connection would be possible between the IoT Platform and enterprise applications.

2.2.3 Enterprise Application Integration Services

To integrate the IoT Platform into existing business environments, these services will provide functionality to connect specific enterprise applications to the platform, acting as communication and translation services. These integration services are not considered as part of the (generic) IoT Platform, as the connectors to enterprise applications are expected to be case-specific and not – or only partly – reusable for connecting to other systems.

Enterprise application integration services could support bi-directional data transfer, as enterprise applications could be both data sources and data consumers.

2.3 MAESTRI IoT Platform

The IoT Platform comprises different modules, which can be grouped into LinkSmart® LocalConnect and Platform Services.

2.3.1 LinkSmart® LocalConnect

LinkSmart® LocalConnect provides a set of components allowing to set up local smart environments, consisting of a variety of devices, applications and services, which can be discovered and communicated with using publish/subscribe or request/response messaging. It is the main layer for connecting the physical world to the IoT Platform. It

consists of device connectors (specific for each individual type/protocol of connected device) as well as the Resource Catalog and the Service Catalog.

2.3.1.1 Device Connectors

A device connector provides the means for a connected IoT device to communicate with the rest of the world regardless of the communication protocol it uses. A device connector can run on a computer or another type of device that is connected to the TCP/IP network and on which LinkSmart can be deployed. A device gateway hosts the proxies of the IoT devices and turns them into LinkSmart-enabled devices. Device connectors need to be developed specifically for each new device (type) or protocol. It is envisioned that several new device connectors will need to be developed within MAESTRI, e.g. a EUROMAP-63 device connector (as explained in Section 3). Device connectors should be as generic and reusable as possible (e.g. the planned EUROMAP-63 device connector should be reusable for different machines providing a EUROMAP-63 interface).

It should be noted that, besides reading data from connected devices, the actuation of devices would also be possible via device connectors. However, this is currently not foreseen in MAESTRI, as the main objective of the IoT Platform is to collect data that is needed for (eco-) efficiency analysis and calculation of KPIs.

2.3.1.2 Web Services

LocalConnect contains functionality that can be configured to provide REST endpoints for parameters that are provided by device connectors. This functionality is part of the device gateway, a software component that is re-usable as basis for the development of new device connectors. Each parameter can be given a name and can be configured in terms of which REST methods to support, and which content-types to use (e.g. "text/plain"). These web services always provide the latest value that the device connector has received from the actual device.

2.3.1.3 LinkSmart® Resource Catalog

The Resource Catalog maintains the list of all LinkSmart enabled resources that are connected within the local environment. The catalog provides a registry of available IoT devices and resources. It exposes a simple JSON-based RESTful API, which is intended to be used by:

- Device connectors to register the available devices and their LinkSmart enabled resources;
- Applications to discover the LinkSmart enabled devices and learn how to access them and communicate with them.

2.3.1.4 LinkSmart® Service Catalog

The Service Catalog provides functionality for services analogous to what the Resource Catalog provides for resources. It maintains the records of services available in the network and provides a JSON-based RESTful API for service registry and service discovery.

2.3.2 Platform Services

The Platform Services are components providing additional functionality that goes beyond gathering data from devices and distributing them. These services are not mandatory to be used in specific deployment scenarios, but provide an extensible toolbox to be used as needed. Some of these components have already been implemented within the LinkSmart middleware platform, but it is envisioned that the Platform Services need to be extended substantially based on the requirements from other work packages and the lessons learned during the project runtime.

2.3.2.1 Message Broker

The Message Broker is responsible for providing a communication infrastructure, processing and distributing all events generated by IoT devices and systems. It provides interfaces that allow other components to subscribe to events coming from data sources. Usually device connectors and enterprise application connectors will act as data sources that are publishing events, but the Message Broker is of course accessible by any component through the IoT Platform. The broker functionality can be implemented based on different technologies in order to support special requirements like real-time support etc. Currently LinkSmart supports an implementation based on MQTT, which is also used in a wide variety of IoT applications.

2.3.2.2 Business Process Annotator

The Business Process Annotator service will be used when the support for monitoring process steps is needed. Based on a BPMN model, the data from IoT devices and systems will be annotated with information about the process step in which the events have been generated. To do this, the Business Process Annotator will get the events from the Message Broker, annotate them with additional metadata and then publish the annotated events through the Message Broker as well.

2.3.2.3 Identity & Access Management

Services for the identification of users as well as managing access rights and providing services to restrict access to specific groups of users are located here. LinkSmart already provides basic support for such functionality and is designed in a modular way. Therefore the integration of other solutions, if need be, is expected to be straightforward.

2.3.2.4 Configuration Management

This service provides the means to store, update and retrieve configuration information, such as deployment information for components of the platform.

2.3.2.5 Logging & Monitoring

Different services for logging can be used, e.g. for logging into files or integration with other logging systems. For monitoring, service components can be used that issue alarms depending on certain conditions or that are integrated with other systems providing contextualized condition monitoring. Other systems like Nagios² can be integrated as well.

2.3.2.6 LinkSmart® Historical Datastore

Through the functionality made available by this module, the IoT Platform can store data from data sources and provide (aggregated) data at a later point in time to the data consumers (e.g. to continuously store data from machines, and then later provide an aggregation of collected data to front-end applications). In contrast to the data processing agents/pipelines explained in Section 2.3.2.7, the LinkSmart Historical Datastore does not work continuously, but it processes the data in configurable time intervals. This also allows to provide functionality to reduce the amount of storage space needed by aggregating the data for past periods and just storing the aggregated results.

2.3.2.7 LinkSmart® Data Processing Agents

LinkSmart Data Processing Agents are highly individual components that process data in ways specific to the intended deployment scenario. These services need to be implemented individually for each specific scenario, however LinkSmart provides a framework and APIs so that they can be easily integrated into the platform, thus reducing the development efforts.

A specific type of data processing agents are data processing pipelines, which provide the functionality to process incoming data from the Message Broker on the fly before other systems get access to the data. This can be used e.g. for data fusion, where the amount of data is reduced in order not to flood data consumers with too much data. It is also possible to provide services for complex event processing such as Apache Storm³ or Esper⁴.

2.3.2.8 LinkSmart® Model Repository

LinkSmart and also the IoT Platform highly utilize the concept of Model Based Development (MBD) and Model Based Software Engineering (MBSE). The LinkSmart Model Repository provides a centralized way to store, retrieve and update models by managing them with versioning. In that way, platform components can also be configured to use models for a certain revision only instead of the most current ones, which greatly enhances the functionality and the resilience to errors in an IoT scenario.

² <https://www.nagios.org/>

³ <http://storm.apache.org/>

⁴ <http://www.espertech.com/esper/>

2.4 MAESTRI Front-End Applications

This box indicates all MAESTRI end user software tools and applications, which are the main data consumers from the point of view of the IoT Platform.

The main MAESTRI Front-End Applications to be connected to the IoT Platform are MSM[®] and ecoPROSYS[®]. These two applications will be integrated and together form the Efficiency Framework (WP2). The Efficiency Framework's connection to the IoT Platform is (indirectly) via MSM and EcoPROSYS, as it works on the integrated results of these two tools.

The Management System (WP3) tools and the Industrial Symbiosis (WP4) toolkit will be primarily accessing data via the WP2 tools (ecoPROSYS[®], MSM[®] and MAESTRI Efficiency Framework), thus WP3 and WP4 plan to make use of the IoT Platform in an indirect way. They do not need any direct software interfaces to get online data via the IoT Platform.

Besides the WP2 tools, also other mobile or web applications could be needed as front-end applications. It is currently planned to implement a basic shop floor data visualisation dashboard, i.e. a GUI module that allows a visualization of online or historical data without the need to use complex end user tools. In addition, it is expected that also other (custom developed) applications could be needed in each concrete business case pilot implementation.

Looking at the current overall architecture, the MAESTRI Front-End Applications could get their input data in different ways:

- In case that front-end applications are able to subscribe to MQTT topics, they could directly listen to the topics published by the Message Broker.
- In case that the existing raw format of the data fits for the application and it is able to access RESTful web services, it can get data directly from the web services provided by LinkSmart LocalConnect.
- In case that a front-end application needs to get data in specific pre-defined formats other than those provided by the Message Broker or the REST web services, there would be a need for custom application back-end services to "translate" the data format from "IoT-Platform-internal format" to "front-end application format".
- In specific cases it could be possible that front-end applications are getting their data directly from the Enterprise Application Integration Services (i.e. in case they need inputs from enterprise applications).
- Obviously there should also always be an option for manual data input into the front-end applications. On the one hand, manual input could be a fallback solution in case that data is available neither from sensors via the IoT Platform nor from enterprise applications, on the other hand, it would be possible that the automatization of data transfer for specific data sources would not make sense after doing a trade-off between implementation costs and benefits (e.g. because they are not needed often, there are no appropriate sensors available, or sensors are too expensive). In addition, manual input should also always be considered as an approach that facilitates quick initial testing of MAESTRI Front-End Applications,

while the automatization of data transfer, which will usually take some time and efforts for development, could be done in a later step.

2.5 MAESTRI Application Back-End Services

The MAESTRI Application Back-End Services are highly specialized services that are implemented based on the needs and requirements of specific deployment scenarios. They need to be developed individually and are envisioned to act as a middleware between the IoT Platform and the MAESTRI Front-End Applications – in order to develop reusable components. The back-end services should provide data from the IoT Platform – either current or historical data – in a format as needed by the front-end applications.

The Application Back-End Services could get their data either from the IoT Platform or from the Enterprise Application Integration Services directly.

2.6 LinkSmart® SDK Tools

This section is about components for model-based development. Models can be edited using the SDK tools before being deployed to the IoT Platform (see also section 2.3.2.8 about the Model Repository). While the other architecture components presented above work at runtime, the SDK tools are used by administrators beforehand for setup and configuration of the IoT Platform.

2.6.1 Business Process Modeller

The Business Process Modeller is a graphical tool for creating and updating models. It is easy to use, and can therefore be utilized also by non-ICT experts or in collaboration with them. The business process models can be used to automatically annotate events within the IoT Platform in order to allow for deeper analysis of the processes and identify optimization potentials. The Business Process Modeller also creates BPMN files in a standard compliant XML format that are used in the Business Process Annotator component of the IoT Platform.

2.6.2 Domain Mapping Tool

The Domain Mapping Tool allows to create a domain model, which acts as a description of the actual physical environment that is to be integrated through the IoT Platform. A domain model describes the entities of the business environment that are relevant to the end-user applications. It is a way to enhance the communication between different stakeholders by reducing the complexity of the description of the physical environment. This allows for more targeted discussions, for example about which machines, sensors and actuators are necessary to fulfil the aim of the end-user applications, and provides the basis for the model-based software engineering approach of MAESTRI.

2.6.3 Deployment Mapping Tool

Besides the domain model, which describes the most important entities of the physical world, the platform components need additional information to be able to connect to the physical devices. Deployment information, such as IP addresses, ports or protocols to

be used to communicate with devices, is stored in the deployment model and is tightly connected to a certain revision of a domain model. In order to create and update the deployment model, a tool is available that also provides additional functionality like checking if necessary data is missing.

3 First prototype application of the IoT Platform

3.1 Motivation

To assure that the IoT Platform development will follow a realistic, practical and business-driven approach, an initial end-user driven scenario for the application of the platform was defined. This scenario serves as basis for the realization of the first MAESTRI software prototype, which will be iteratively extended afterwards.

The aim is to get a first running prototype in a timely manner, which should facilitate the continuous refinement and extension of the requirements towards the IoT Platform and support the further alignment between the different MAESTRI work packages. The MAESTRI business cases were analysed in order to find a rather simple, but realistic example, including the connection to one shop floor data source and one existing enterprise application. It was agreed between partners from the different work packages that this scenario should focus on a specific production process of an injection moulding company. The overall aim is to provide information related to process status and efficiency, in order to facilitate the analysis of the efficiency and eco-efficiency of the production process.

As a first step, the initial state at the injection moulding company had to be analysed. For this purpose, workshops have been held with partners from WP2 (Efficiency Framework), WP3 (Management System) and WP4 (Industrial Symbiosis), and the relevant KPI and variables to be monitored were defined using Eco Orbit View and field session interviews. In addition to the characterisation of the initial state, as a result from these workshops, some initial improvement measures have also been identified.

In order to build the first prototype, WP5 partners had also to assess the software systems and tools currently implemented at the company, in order to evaluate the availability and accessibility of basic data required for the defined KPI and variables to be monitored. Based on this analysis, it was decided to first develop an interface enabling the collection of data from an injection moulding machine, and afterwards to build an interface to the existing ERP system, aiming to get additional data that is complementing the data from the machine.

Having in mind that the first prototype should not be overly complex, it was agreed that the initial analysis should be based just on a subset of the data that the injection moulding machine can provide.

The starting point for the further definition of implementation details was the definition of an application scenario, explaining how an end user would use the MAESTRI Front-End Applications, which are provided with data via the IoT Platform. As this application scenario describes requirements towards MAESTRI as a whole, not just towards the IoT Platform, its details are not presented here but in deliverable D1.5 (Lessons Learned and Updated Requirements Report 1).

3.2 Data flow to be supported by the IoT Platform

The following MAESTRI Front-End Applications are part of the first prototype scenario:

- MSM[®] is used to analyse the efficiency (manufacturing and process perspective) of the process: value-added vs. non-value added energy consumption, material consumption (thus resource efficiency), time spent on producing pieces, and other KPIs of the injection moulding business case, namely some KPIs related to operation variables monitoring;
- ecoPROSYS[®] is used to assess the eco-efficiency of the process and to analyse the impact of alternative scenarios from the environmental, costs and value perspectives;
- A basic dashboard is used to visualize on-line data from the shop floor, so it shows “how the production is currently running”.

These applications require the following data/information to be provided by the IoT Platform:

- A definition/modelling of the process steps;
- A mapping of available variables/parameters to process steps;
- Energy and material consumption data and production time per process step;
- Specifically for MSM[®]: value-added and non-value added energy and material, other process operation variables, such as temperature of the mould cooling fluid, etc.;
- Specifically for ecoPROSYS[®]: economic value (and costs) of the process operation and outcomes from the production process.

The relevant data sources at the company for the defined application scenario are the ERP system and the injection moulding machine. Data that will be needed are (1) energy/material consumption, production time per piece, provided by the injection moulding machine, and (2) economic value and process related costs, provided by the ERP system.

The IoT Platform should be used to connect these data sources to the MAESTRI Front-End Applications in order to automatize the data transfer. The high-level data flow between these software modules is indicated in Figure 4.

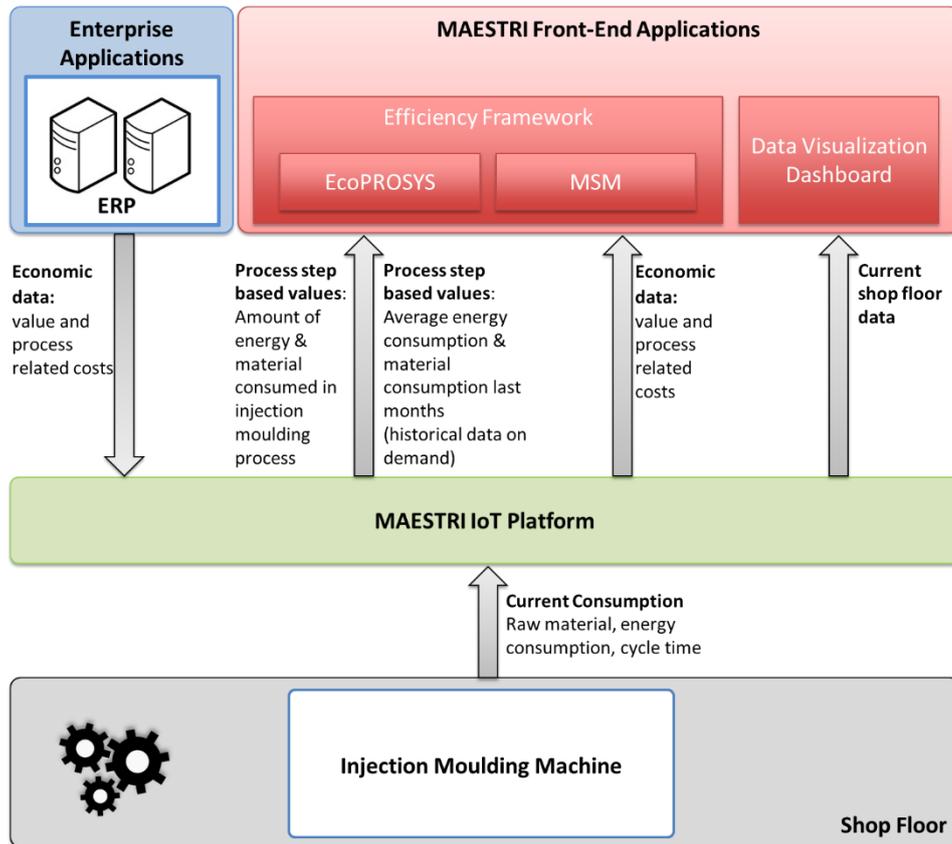


Figure 4 – Data flow in the prototype scenario

To support this data flow, one of the major tasks is to create the necessary interfaces to connect data sources and MAESTRI Front-End Applications to the IoT Platform. It will be necessary to:

- Connect the IoT Platform to the EUROMAP-63⁵ interface of the injection moulding machine: a file based device connector will be needed which reads the data files and provide their contents to the IoT Platform;
- Connect the IoT Platform to the company's ERP: a specific connector will have to be developed for this purpose, which translates the data from the ERP – provided via web services – into a format that is usable by the IoT Platform;
- Create an interface so that MSM[®] can get data from the IoT Platform (this is currently planned as a REST web service);
- Create an interface so that ecoPROSYS[®] can get data from the IoT Platform;
- Implement and configure the Data Visualization Dashboard.

3.3 Prototype architecture

Having in mind the prototype application scenario and its specific requirements towards the IoT Platform, an initial architecture for the first MAESTRI prototype was defined (see

⁵ <http://www.euromap.org/euromap-63>

Figure 5), implementing a subset of the modules from the overall architecture (presented in Section 2).

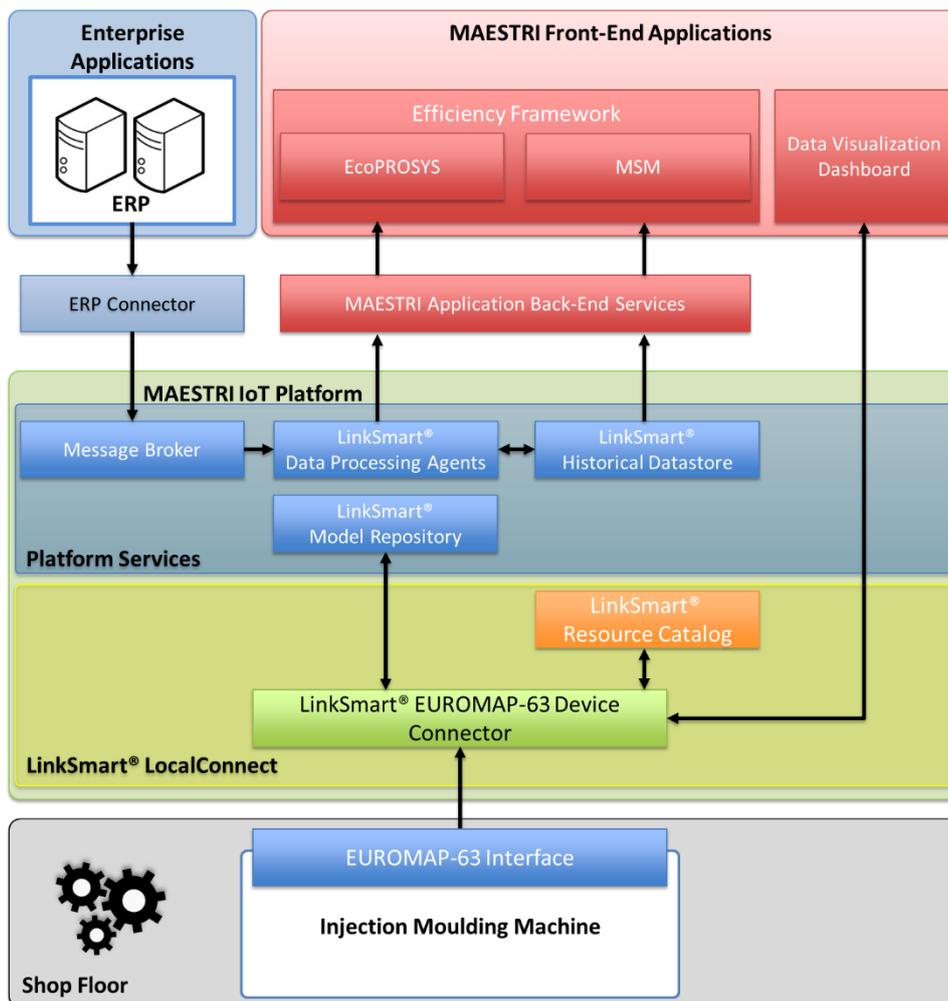


Figure 5 – Initial software prototype architecture

As already explained above, the initial data sources in this scenario will be the injection moulding machine and the ERP system. Dedicated interfaces will be needed to connect both data sources to the IoT Platform. These will be the EUROMAP-63 Device Connector and the (web-service-based) ERP Connector for the injection moulding machine and ERP system, respectively. The device connector will register the available resources in the LinkSmart Resource Catalog and will publish data from the machine to the Message Broker at system runtime. The ERP Connector will be running outside of the IoT Platform, representing a kind of additional simple middleware between the platform and the ERP. The initial data flow from the ERP system will be unidirectional for the first prototype.

The Message Broker will handle incoming data from both data sources based on a publish/subscribe mechanism. The Data Processing Agents, who will be the main data subscribers, will have to be programmed specifically for this scenario and will provide data pre-processing functionality, e.g. aggregation of data in order to reduce bandwidth needs and provide the data in a form directly usable by the MAESTRI Front-End Applications. Online data can be provided by the Data Processing Agents directly to the

MAESTRI Front-End Applications via the Application Back-End Services, which are planned to be implemented as RESTful web services. The Data Processing Agents will also write data to the Historical Datastore to allow for later analysis of aggregated data. When historical values are required, it will be accessed on demand by the Front-End Applications via the Application Back-End Services.

The Model Repository will be used to manage and provide domain and deployment models. The domain model tells the middleware which sensors and machines are accessible, while the deployment model contains e.g. the physical addresses for the connections. The device connector will access these models on start-up to configure itself. For example, the deployment model will tell the device connector about the address through which the EUROMAP-63 interface of the injection moulding machine can be accessed.

3.4 Technical implementation example of a basic data flow

To give a concrete example how the technical implementation of a basic data flow from the shop floor to an end-user application – using the MAESTRI IoT Platform – can look like, this sub-section provides technical details about the first part of the prototype currently under development. Figure 6 indicates the technical implementation of a subset of the components and their communication relationships, supporting the data transfer from the injection moulding machine to the Data Visualisation Dashboard.

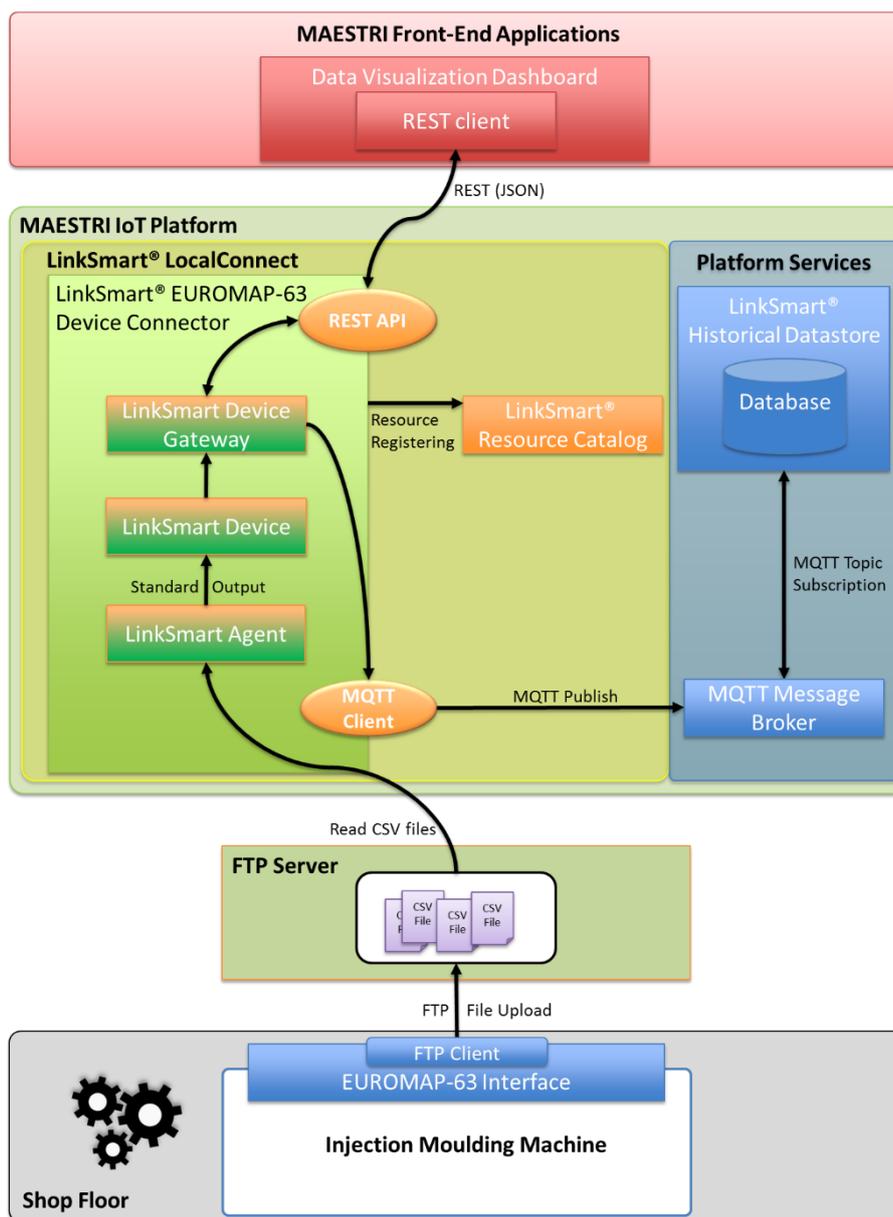


Figure 6 – Technologies and communication protocols used for the prototype implementation

The EUROMAP-63 Interface is programmed to generate file(s) with the data of the selected variables (e.g. machine status, parts produced, cycle time) with a specified frequency (e.g. once per second). The file(s) in CSV format are written to the FTP server.

The LinkSmart Agent of the EUROMAP-63 connector will be running on the FTP server. This application reads each line of the CSV file and publishes the data of each machine variable to the MQTT Broker, using the MQTT Client that is provided by the Device Gateway. The Historical Datastore will be subscribed to this data and will store the published data in a database.

Whenever a REST request from the Data Visualisation Dashboard is received for a specific machine parameter, the Device Gateway will return the latest value for that parameter,

which was published beforehand by the LinkSmart Agent. The typical runtime interaction between the components is detailed in Figure 7.

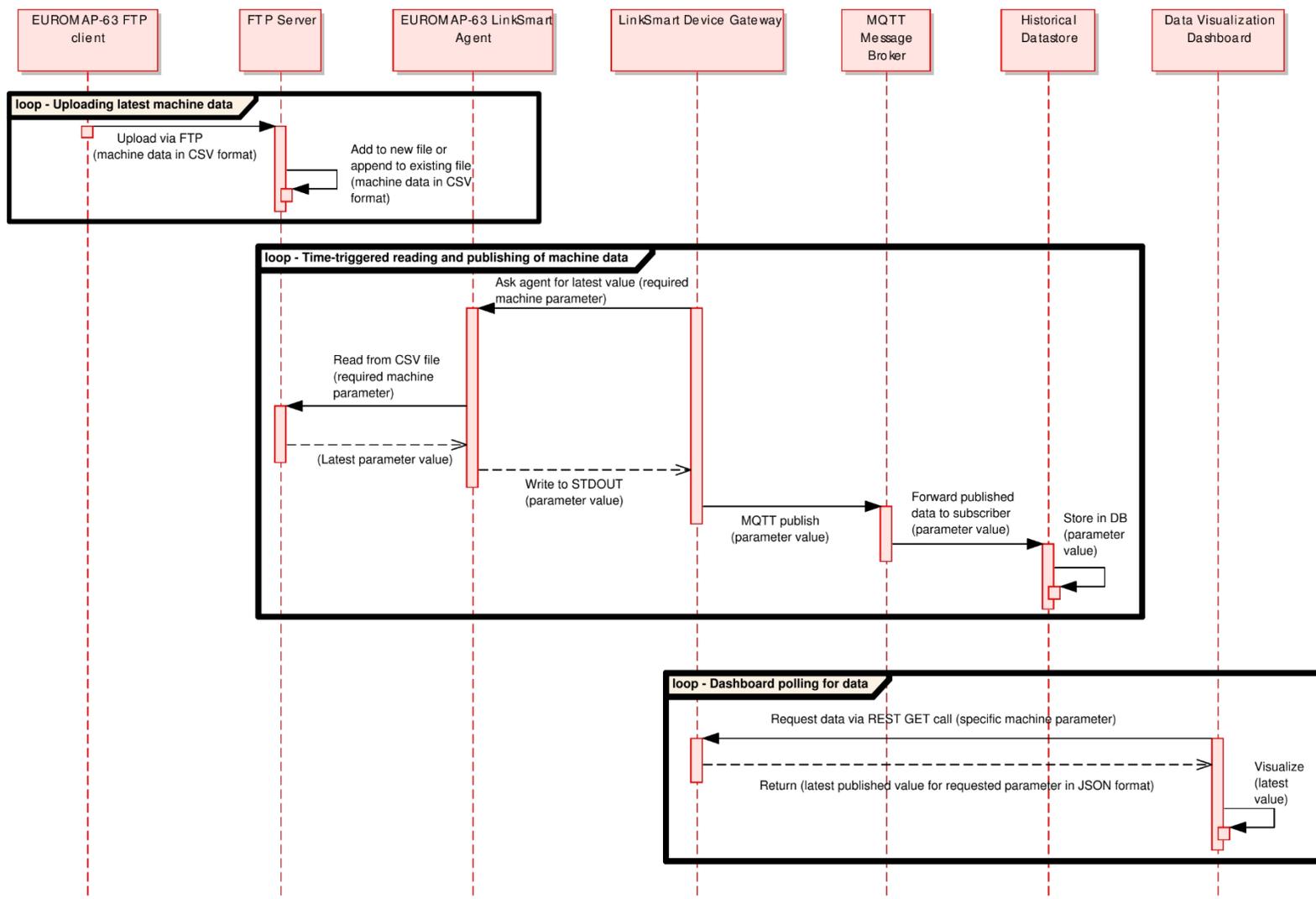


Figure 7 – Interaction between different software components

It should be mentioned that there are basically three independent interaction loops, which are all time-triggered. The machine interface writes data in a specified frequency, the device agent reads data in a specified frequency (triggered by the gateway) and the dashboard asks for latest data in a specified frequency.

Instead of polling for data in a specified frequency, there are two other possible options for a data consumer to get data from the IoT Platform:

- Data consumers can subscribe to specific MQTT topics and thus will be notified whenever new data is published by the EUROMAP-63 Agent.
- In case that instead of the latest (online) data some historical data is required by the data consumer, then the Historical Datastore can be accessed via its RESTful API.

This rather simple subset of the first prototype contains several typical implementation patterns that can be replicated in other, also more complex, application scenarios. A major part of the basic technical implementation approach can easily be re-used, just the device connector will obviously have to be adjusted/replaced in case there are other types of data sources.

3.5 Further implementation plan

The development of the IoT Platform will be done in a stepwise and iterative way. The first prototype is currently under development, as presented above, and it will serve as basis for further discussion and refinement of the IoT Platform requirements. Lessons learned from its implementation will provide valuable input to for necessary improvements/extensions of the IoT Platform and will also facilitate the industrial pilot implementation in the other business cases.

Later in the project, two main software deliverables are planned:

- D5.5: Initial Software Development Kit 1 [M27]: Initial Software Development Kit produced before the pilot implementation.
- D5.6: Final Software Development Kit 2 [M45]: Final Software Development Kit that includes lessons learned and refinements following the pilot implementation.

In addition to these main software deliverables, several intermediate development steps are planned, which will be defined and refined along the process in an iterative way.

Several technologies and tools will be needed to support the development of the described modules of the IoT Platform and the integration of all parts of the system, e.g.:

- Git (<http://git-scm.com/>) is used as version control system, a common source code repository was already established;
- JIRA (<https://www.atlassian.com/software/jira>) is used for requirements and bug tracking.

It should be noted that there is no need to agree on a common programming language, as the service-oriented architecture foreseen for the IoT Platform is basically language-

agnostic. Different software modules could be developed in different languages and communicate via commonly agreed (RESTful) web services and/or the MQTT-based Message Broker of the IoT Platform.

4 Conclusion

The objective of Task 5.1 Architecture Design was to translate the requirements regarding the IoT Platform into functional modules to be developed and implemented in the scope of the MAESTRI project. The results of this work were presented in this document.

As a starting point for this task, all MAESTRI requirements that had been gathered until the time of this deliverable were taken into account as far as they are directly or indirectly addressing the IoT Platform. These requirements were further refined and discussed among partners of work packages 2 (Efficiency Framework), 3 (Management System), 4 (Industrial Symbiosis), and 5 (IoT Platform Development). The result is the definition of the functional modules and the architecture of the IoT Platform as well as its envisaged interactions with other systems, especially the MAESTRI Front-End Applications. A subset of the IoT Platform's functional modules is already covered to a certain extent by functionality of the existing LinkSmart middleware, which shall be reused in the scope of MAESTRI. Other functionalities need to be developed as new services in the scope of Task 5.2 Platform Engineering.

The initial implementation plan for the IoT Platform aims to first develop a simple running prototype, which should facilitate the further alignment between work packages and the continuous refinement/extension of the requirements towards the IoT Platform. Therefore an initial IoT Platform application scenario was agreed upon among the partners, which defines the functional scope of the envisaged first prototype. Starting from the scenario's end user point of view, the requirements towards the IoT Platform were derived and an initial architecture of the first prototype was presented, using a subset of the overall IoT Platform architecture plus case-specific modules that will have to be developed for the concrete industrial application. First technical implementation details of the prototype, which is currently under development, were also presented in this document.

Based on the feedback to the prototype implementation as well as based on the additional requirements that will be gathered in the scope of T1.5 (Evolutionary Requirements Elicitation), the work results presented in this document will be updated iteratively during the IoT Platform development. Therefore, although this document presents the final results of T5.1 and the basic overall architecture is considered as mostly final, further refinements will probably be needed in the future and thus the architecture design process will be continued in an iterative way.