This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 680759

ReCaM
Rapid reconfiguration of flexible Production Systems

ReCaM— Rapid Reconfiguration of Flexible Production Systems through Capability-based Adaptation, Autoconfiguration and Integrated tools for Production Planning

Project acronym:	ReCaM
Call and Contract:	H2020-FoF-2015
Grant agreement no.:	680759
Project Duration:	01.11.2015 – 31.10.2018 (36 months)

D7.4: Final Version of Applicable Standards

Organisation name of lead partner for this deliverable: **NXT**

Deliverable nature:	**Report (R)**		
Dissemination level: (Confidentiality)	**PU**	Public	X
	PP	Restricted to other programme participants (including the Commission Services)	
	RE	Restricted to a group specified by the consortium (including the Commission Services)	
	CO	Confidential, only for members of the consortium (including the Commission Services)	
Contractual delivery date:	31.10.2017		
Actual delivery date:	30.10.2017		

File: 20171030_ReCaM_D7.4_Final_Version_of_Applicable_Standards_FINAL.docx

Additional document information

Workpackage	WP7
Tasks:	T7.3
Contributing partners:	All
Authors:	Šejla Trakić, Ronnie Lin, Gernot Kollegger, Sebastian Schröck, Sven Dose, Mildred Puerto Coy, Eeva Järvenpää, Niko Siltala, Giovanni Lugaresi, Anteneh Teferi Yemane, Nicla Frigerio
Version:	1.0
Total number of pages:	26
Keywords:	Standardization, Applicable Standards, ReCaM, Reconfiguration

Versioning and contribution history

Version	Organization	Comment	Date
0.1	NXT	Initial version of deliverable	14.06.2016
0.2	NXT, Bosch	Mid-term draft. Input for Production execution and control.	08.08.2017
0.3	TUT	Input for Capability and Resource management	08.09.2017
0.4	POLIMI	Input for Systems engineering platform and Production and reconfiguration planner	18.09.2017
0.5	NXT, TEC, Bosch	Draft version before review. Input for Product management and Resource management.	06.10.2017
0.6	NXT, Bosch	Ready for review version	12.10.2017
0.7	TEC	Reviewed version	24.10.2017
0.8	NXT	Working on comments from review	25.10.2017
1.0	Bosch	Final Document	30.10.2017

Table of Content

Executive summary

1. Issues (situation, motivation and tasks)

During the first months of the project the consortium has identified many different standards that, at that point seemed to be applicable. The results of that first analysis were presented in D7.2 – "Preliminary list of applicable standards". This document presents the standards that have been chosen by the consortium and have been implemented during the different development activities.

Chapter 3 describes the standardization strategy marking the paths of spreading the project results by using them in other projects or initiatives.

2. Results

In this document the standards are presented based on the Architecture Building Blocks defined in D2.1 – "Overall system architecture". This means that the standards are associated to the building block in which they are used. These standards are described in tables which contain a short description of the standard, its applicability in ReCaM, the benefits and also its weaknesses of using it. These tables give a good overview of the standards and also explain the reasons why they were chosen to be implemented.

3. Conclusions and Consequences

Besides presenting the standards that are being implemented in the project, this document also presents the path towards standardization by linking the projects results to other projects and initiatives like "Industrie 4.0". The way how the standards are implemented can be seen as a white paper on how it is possible to use these standards to support a ReCaM based production system.

1 Introduction

The first step in the standardisation task was to identify relevant standards in different functional areas of the ReCaM system. This was done in deliverable 7.2 in which the consortium has identified many standards, technical specifications and good practices that can be applied in the ReCaM project. The standards where collected in an excel table and classified in the following categories: Communication, Control, Electrical, Mechanical, Methods, Terms and Definitions.

For the preliminary list of applicable standards, we have:

- Shared the excel list of standards we got from TUT with the consortium
- Every partner has reviewed the list and:
  - Marked the standard if it is applicable for that partner
  - Added a new standard, technical specification or good practice if it was applicable but was not in the list
- There was a possibility to rate the standards from 0 (not useable) to 10 (must be used). Based on the knowledge of the project at that time, only few standards were rated.
- Rows marked as applicable for the partners at that point of the project were part of the deliverable 7.2

The standards from the preliminary list of applicable standards were evaluated by the responsible partners during the project. The applicability was further analysed as the system architecture was being defined and the interaction between different parts of the architecture were defined in detail. Also, new standards were added to the list, as at the beginning of the project they were seen as not applicable or were not known to the consortium.

This report will present the standards that have been implemented in ReCaM until this point. The standards are presented in tables that contain the most important information about them: short description of the standard, for what it is used for and why, what are the benefits of using the standard and what are its weaknesses.

As a result of the close collaboration with WP2 – System and module architecture the chosen standards, technical specifications and good practices will be classified by the different functional areas of the ReCaM model. These functional areas correspond to the building blocks of the ReCaM architecture presented in deliverable 2.1 – ‘Overall system architecture’ includes the ‘Building Block specification’ document.

1.1 Document Maintenance

This document will be updated as needed. If the document is written in a format that got outdated (i.e. technically incompatible) during the reporting period, the document will be revised into the latest template format at the next review.

This document contains a revision history log. When changes occur, the document’s revision history log will reflect an updated version number, the date of the new version, the owner making the change, and a detailed description of the changes.

2 Final list of applicable standards

The standards in this document are classified by the different functional areas of the ReCaM system. Therefore, the following paragraphs will be structured along the building blocks within this architecture. In this way, it is possible to identify the areas with missing standardisation as well as standards that need to be adapted/enhanced.

The functional areas for the classification are a result of the Building Block specification for the system architecture described in D2.1 'Overall system architecture'. The following are the level 1 building blocks (Figure 1) of the ReCaM architecture:

- Product Management
- Capability Management
- Resource Management
- Systems Engineering Platform (Greenfield planner)
- Production and Reconfiguration Planner
- Production Execution and Control

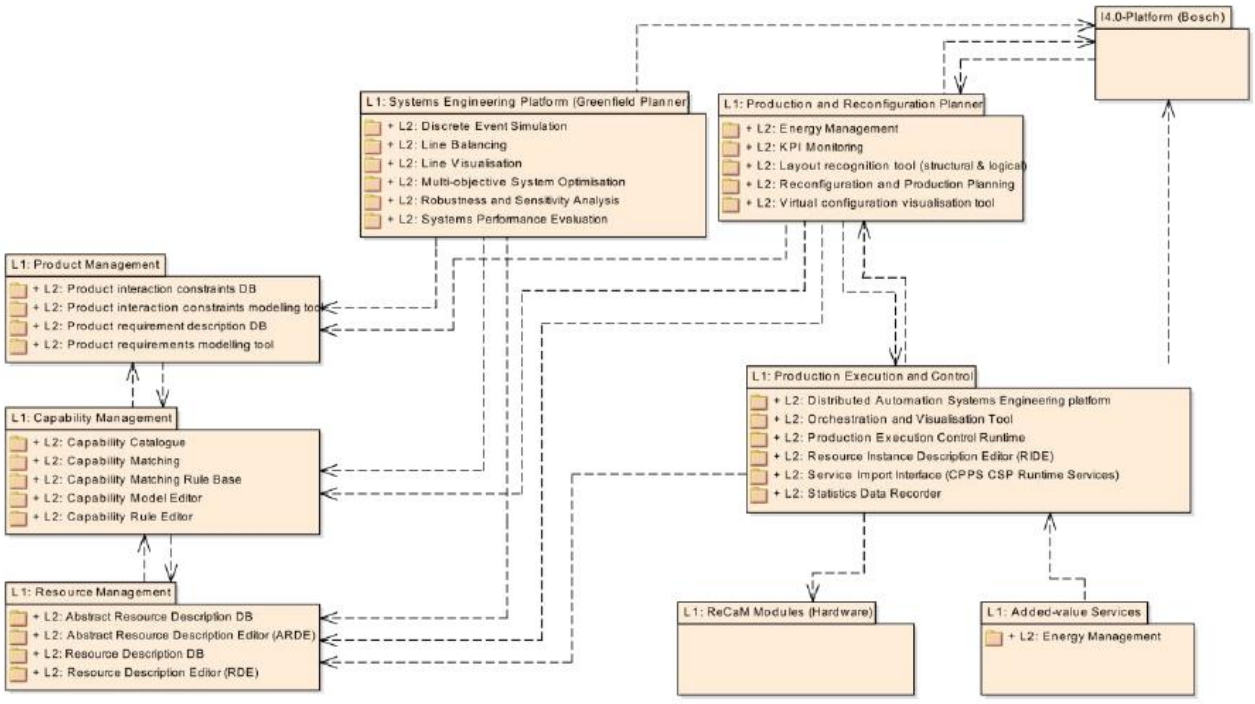

Figure 1: System architecture - level 1 building blocks (refer to D2.1)

2.1 Product Management

The Building Block "Product Management" is focussed on handling all product-related inputs towards the ReCaM-system. This is a prerequisite to enable the (semi-)automatic derivation of the production plans from the results of the product development. During the development of the different interfaces and building blocks, two different aspects towards the product requirement description were identified to be beneficial for the ReCaM system. The process-oriented view as well as the structure-oriented view, which are described in detail in the document D3.2. Unfortunately, no standard is known, which handles both of these aspects. Therefore, D3.2 is mainly based on various aspects of different standards but cannot implement a standard one by one.

Name of standard	VDI3682
Description	The standard VDI3682 "Formalised Process Description" is published by the Association of German Engineers. This standard enables modelling the products, the processes as well as the technical resources conducting these processes. Having this distinction in the models in an explicit way enables focussing on each of the aspects without losing the others out of sight. Behind the objects representing the products, product steps and resources, a number of attributes can be used in specify the objects in detail.
Applicability	This standard is used within the CESA use case in order to describe each of the actuators assembly-instructions in a suitable way to represent the Product Requirement Description. In the implementation of ReCaM for the automation of CESA's operation, it was necessary to describe each of the assembly steps in a formalized manner, aligning the capabilities requirements with the real operation.
Benefits/reasons to use	- Explicit distinction between modelling the products, process steps and resources, while still having these objects in one single model allowing associations representing the relations - Easily human-interpretable - Unify the description of all the assembly instructions used in CESA pilot. - This formalization has the potential to be integrated in the robot process generation, replacing the human intervention. Suitable to be explored as future work.
Weaknesses	- VDI3682 public information is limited. The implementation of this standard was partial (only referred to the visual organization of the information). Any further formalism from this standard was used. - Because of the previous item, the information organized with this formalism still requires it codification in a programming language, being used xml for this purpose.

2.2 Capability Management

In this building block, the Capability Model is defined and maintained using the standard ontology editor Protégé. This model is stored as a Capability Catalogue, which can be accessed by other ReCaM building blocks through an API called Capability Query Library. The ontology editor is used to create new capabilities into the model, to define their relationships as well as the parameters (properties) characterising each capability. Capability matchmaking system and rule base is developed to implement and run the rules, which can automatically calculate the combined capabilities of aggregated resources and to match the product requirements with the resource capabilities. The resource description editor utilizes the Capability Catalogue by assigning each resource the capabilities it has, and filling in the resource specific parameter values. The connection between product requirements and resource capabilities is accomplished by importing the same Process Taxonomy to both the Capability Model and Product Model ontologies. Capabilities in the Capability Model then implement a certain process in the taxonomy, while the product requirement steps require certain processes in the taxonomy.

Name	OWL
Description	The Web Ontology Language OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. Ontologies are a formal way to describe taxonomies and classification networks, essentially defining the structure of knowledge for various domains. OWL is developed as a vocabulary extension of RDF (the Resource Description Framework) and is derived from the DAML+OIL Web Ontology Language. The data described by an ontology in the OWL family is interpreted as a set of "individuals" and a set of "property assertions" which relate these individuals to each other. An ontology consists of a set of axioms, which place constraints on sets of individuals (called "classes") and the types of relationships permitted between them. These axioms provide semantics by allowing systems to infer additional information based on the data explicitly provided.
Applicability	In ReCaM, OWL is used as a representation language for the Capability Model. There are also Resource Model ontology and Product Model ontology, codified in OWL, which are used in the matchmaking between product requirements and resource capabilities. OWL is the most commonly used ontology language, and multiple ontology editors exist, such as Protégé, which can be used to create the ontology.
Benefits/reasons to use	There are several benefits on using OWL-based approach, compared to other approaches, such as XML schemas. The advantages include: - Support for information integration and reuse of shared vocabularies - Ability to cope with semi-structured and undefined data - Separation of syntax from data modelling - Web embedding - Extensibility and resiliency to change - Support for inference and classification, based on formal semantics

	- Representation flexibility, especially ability to model graph structures - Ability to represent instance and class information in the same formalism and hence to combine them. The extensibility and evolution of the model is an important factor in ReCaM, as the Capability Model (catalogue) will be frequently extended with new capabilities. OWL is resilient to change as it is based on the "Open Word Assumption" which means that that the truth value of a statement may be true irrespective of whether or not it is *known* to be true. This assumption has significant benefits for open evolving systems. It means that future extensions to the data models, or integration of unforeseen data sources, are easier because none of the existing processors should be assuming their data sets are complete. This makes the model backward and forward compatible. Thus, no error is produced even if some data would not be defined, or if more data, which doesn't fully comply with the model, becomes available.
Weaknesses	Weaknesses of OWL are: - Weak ability to validate documents - Expressivity limitations, particularly in terms of correlating across different properties of a resource - Performance - Lack of familiarity and long learning curve OWL targets to inference and not on validation of data. Weak ability to validate data is due to the "Open Word Assumption" and "Non-unique naming" assumption. The latter means that if two objects have different name, it doesn't necessarily mean, that they are different objects, if not explicitly stated. The open world assumption implies that if value is not defined for a certain property, we cannot assume that definition to be invalid. One important limitation is that it is not possible to model correlations between different properties. Thus, it is not, e.g. possible to define units for the datatype properties. Currently we model the unit information as a human-readable comment of the property. In order to attach units to numerical data values in a semantic way, one would need to save the parameter values separate instances, which have two datatype properties to represent the actual numeric value and the unit value. This would complicate the model, but allow the data to be represented in different units.

Name	SPARQL
Description	SPARQL is an abbreviation of SPARQL Protocol and RDF Query Language. It is a semantic query language for databases, able to retrieve and manipulate data stored in Resource Description Framework (RDF) format. SPARQL specification defines the syntax and semantics of the query language for RDF. It was made a standard by the RDF Data Access Working Group of the W3C consortium and it is recognized as one of the key technologies of the semantic web. SPARQL allows for a query to consist of triple patterns, conjunctions, disjunctions and optional patterns. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs.
Applicability	SPARQL is used in ReCaM to query information from the OWL ontologies. The Capability Query Library is an API through which the other ReCaM tools can request information from the Capability Model. It has many built-in SPARQL queries to retrieve various capability related information from the model. SPARQL will also be used by the matchmaking software to retrieve information from the Product Model and Resource Model.
Benefits/reasons to use	SPARQL is a standard query language for working with RDF data and OWL-based ontologies. Therefore, it is used in ReCaM. Due to the formal semantics of OWL and the various patterns of SPARQL it is possible to extract information from the ontology, which has not been explicitly stated.
Weaknesses	The flexibility of the semantic data model causes a trade-off with performance compared for example to relational databases.

Name	SPIN
Description	SPIN (SPARQL Inferencing Notation) is a W3C Member Submission that has become the de-facto industry standard to represent SPARQL rules and constraints on Semantic Web models. SPIN also provides meta-modelling capabilities that allow users to define their own SPARQL functions and query templates. Finally, SPIN includes a ready to use library of common functions.
Applicability	One of the basic ideas of SPIN is to link class definitions with SPARQL queries to capture constraints and rules that formalize the expected behaviour of those classes. SPIN can be used to calculate the value of a property based on other properties - for example, area of a geometric figure as a product of its height and width. SPIN can also be used to isolate a set of rules to be executed under certain conditions - for example, to support incremental reasoning, to initialize certain values when a resource is first created, or to drive interactive applications.

	In ReCaM, SPIN is used to implement the rules for calculating the parameters of the combined capabilities from the parameters of the simple capabilities that form them (combined capability rules), and the rules for making match between product requirements and resource capabilities (capability matchmaking rules).
Benefits/reasons to use	With SPIN, rules are expressed in SPARQL, which means that one doesn't need to learn a new proprietary rule language. SPIN allows these SPARQL rules to be attached to the classes of the ontology. A suitable reasoner tool, such as SPIN API, can then infer this extra information created by the rules and use it in SPARQL query execution. SPIN allows also to check validity of the data. For example, one may want to require that a field is entered and/or that the string entered follows your format requirements. SPIN offers a way to do constraint checking with closed world semantics and automatically raise inconsistency flags when currently available information does not fit the specified integrity constraints. Compared to SWRL (Semantic Web Rule Language), another tested rule language, we found the performance of SPIN to be better in our use case.
Weaknesses	SPIN is not a full W3C standard. It seems that currently SPIN is mainly supported by one company that offers tools for working with it.

2.3 Resource Management

The resource description editor inside the 'Resource Management' building block is used by a machine supplier to describe the functionality and characteristics of its products (HW resources). The description is based on capabilities taken from the global capability catalogue and other defined properties and characteristics. The resource description (RD) specifies the parameter values that describe the specific type and model of a mechatronic object. The resource description editor allows defining the resource descriptions that are stored in a global (e.g. as a public provider catalogue) or local (in case of internal machine construction) data base. The description and parameter values of a concrete instance of a mechatronic object can also be specified with the resource description editor and they will be stored on the mechatronic object itself to provide a self-description.

Name	oneM2M
Description	oneM2M calls itself an interworking framework that standardises the interfaces of network participants like devices or applications as well as the principals of the communication. The standard specifies the way the communication is handled, not the exact protocol that is used. The data model of an application entity is therefore described in a tree-like structure out of different elements. At the root of a device or application the element is called an application entity. Below this container can be created that can hold other elements. The elements containing actual data, like parameters, are called content instances. In addition to these elements subscriptions can be used to receive a notification about changes in the content of a container.
Applicability	The data model of an application or a device should contain different logical content. Beneath some general meta data (e.g. vendor, id, version, etc.), a device or application should describe the high-level functionality or services it provides as an administration shell. Related to these services different operations and data should be possible to provide. This logical structure is related to the standard Eclipse vorto, that uses the terms information model, function block and operations, status, events, faults and configuration. oneM2M provides the structural elements that are required to build a data model according to usual needs. It allows subscribing to all relevant changes. Thus, it can be used to control and build up complex functionalities. For semantic description of elements oneM2M offers the base ontology. This is not used in ReCaM yet. It is actually more complex than required for the usual use cases and it is therefore not expected to be accepted by machine builders to engineer all contained data. Nevertheless, it provides options for the future.
Benefits/reasons to use	oneM2M is an open and already widely spread standard that is supported by a huge community of companies and organisations from different domains. A big group of supporters is originated within the telecommunication sector, but SmartCity applications and also industrial use seems more and more within focus.

	- oneM2M does not determine the communication protocol to use. It enables use of HTTP, MQTT, CoAP already. Support for other protocols can be developed (OPC UA and DDS are currently under development). This strengthens the acceptance by allowing different providers for the ReCaM system to use the communication of their choice. - oneM2M supports scalability of the architecture. CSEs can be used to connect different services and devices on production system level. Different CSEs can be connected to enable interoperability between wider distributed or connecting services, like energy management above production system level or automated logistics by autonomous robots connecting different production lines. - OneM2M offers different required functionality out of the box, like security features (e.g. communication with HTTPs) and data access management. - Different implementations of the oneM2M standard already exist and have reached a level of robustness that allows usage for industrial applications.
Weaknesses	- oneM2M does not yet support standards that are used broadly in industrial domain, like OPC UA in Germany. However, the interworking is under development and can hopefully be integrated soon. - OneM2M does support different communication protocols. But this does not mean that legacy devices using these protocols can be used directly. As some mechanisms and data models have to be agreed on, connecting legacy devices and services will probably require a small gateway application translating between ReCaM and the legacy world. However, this would most definitely be the case for any other standard, too. New devices should directly support ReCaM and oneM2M, of course. - OneM2M was not developed by parties from the industrial domain. This might lead to reservation because of a “not invented here” attitude. As this is often not based on technical limitations, ReCaM should focus on convincing others with the benefits of its approach. As oneM2M was developed firstly in other domains. Some questions on how to adapt to automation domain will have to be developed in ReCaM and over time. However, oneM2M offers a lot of answers already.

Name	XML
Description	XML (eXtensive Markup Language) is a format for representing structured, textual data.
Applicability	In ReCaM, XML is used as a base syntax for ARDs, RDs and for RD editor's configurations. Then these documents can be stored and exchanged. It will also be used as a basis format for exchanged information between different SW tools and across building blocks, in which case the XML document conveys more temporary message content.
Benefits/reasons to use	XML is a well-known and widely supported format for structured information. It is readable by both machine and humans. It has a lot of tools and software libraries available for support, such as editors, processing applications, document validators, and SW libraries for implementation. It is foundation format for a larger family of languages, which provide together a supporting framework and added functions for handling such structured documents. It is a good approach for data exchange between heterogeneous systems, independent of used operating system, programming language, and application. It is specified as a W3C Recommendation.
Weaknesses	XML does not define much alone, and needs additional specifications in order to be applicable and for connecting it into a specific context. Specification of the actual content must be done by other ways e.g. with the help of XML Schema. Another issue of XML is that the payload per exchanged data ratio could be less than e.g. with JSON.

Name	XML schema, XSD
Description	XML Schema language (XSD) defines document model and structure for an XML document. XSD appends contextual specifications on top of XML. It defines the expected format and structure i.e. syntax for an XML document. It reveals all available options and details of a language. As a main benefit, it can be used to validate XML documents. Additionally, it is used to share and distribute the model of an XML document between different applications and users.
Applicability	In ReCaM, XSDs are used to define the contextual model of exchanged information (messages or resource data models). It is used to share common specification, so that interacting applications know exactly the format of exchanged messages or information. XSD can be used in SW implementations with help of binding process, which is used to automatically generate code for data model implementation from an XSD.
Benefits/reasons to use	XSD increases the applicability of XML. The application/receiver knows in advance exactly what to expect, and because of the validation feature,

	the incoming information can be validated before further actions are taken. It is specified as a W3C Recommendation.
Weaknesses	Learning the XSD language might be quite hard. Fortunately, there exists a lot of XML editors, which provide support and aids for making XSD documents.

Name	**XSL and XSLT**
Description	eXtensible Stylesheet Language (XSL) is a styling language for XML. With the help of XSLs, XML documents can be transformed from one format to another. These XSL documents can be used to perform XSL Transformations (XSLT) processes. XSL differentiates the transformation instructions from the processing implementation. This means that XSL does not bother about the implementation for processing, and also that the same instructions can be utilised by different processing applications (e.g. on different OS).
Applicability	XSL and XSLT are used to modify and process the XML documents. The processing could be format changes like transforming an XML document into any other textual format like HTML. In ReCaM, XSLTs are used to transform ARD and RD information to other formats. These format changes are such as generation of HTML documents from ARD and RD for human friendly viewing of their content; generation of RDSkeleton (customised resource description template) from an ARD; and generation of a SW-tool specific inputs from (A)RDs. Example of the last point is generation of interface description in NXT engineering environment from a RD.
Benefits/reasons to use	Standard processing applications (and SW libraries) can be utilised for performing various transformations. Standard language (XSL) can be used to define what the transformation should do. An XSL document can be re-applied and reused easily over a set of XML documents which share a common data model. It is enough that one makes a transformation definition, others can use it, even in other kinds of applications. XSL(T) is specified as a W3C Recommendation.
Weaknesses	Learning the XSL language and using it efficiently could be an issue.

Name	WSDL
Description	Web Service Description Language (WSDL) is an XML-based interface definition language that is used for describing the functionality offered by a web service. The description of a web service is stored in a WSDL file, which provides a machine-readable description of how the service can be called, what parameters it expects, and what data structures it returns.
Applicability	Resource Catalogue Platform publishes, with help of WSDL, how to use and access its provided information through the web service interface.
Benefits/reasons to use	WSDL offers exact formalism and specification power to define a web service interface. It is machine-readable compared to a "free form" RESTful interface, which needs human interpretation. However, WSDL can be used to define the formalism for a RESTful interface. There exist comparable specifications targeted only for RESTful interfaces such as Web Application Description Language (WADL). WSDL is a W3C Recommendation.
Weaknesses	Complexity could be the weakness.

Name	Open application platform (Linux, Apache, MariaDB, JSP)
Description	LAMP (Linux, Apache, MySQL/MariaDB, PHP/Perl/Python) based implementation for application transferability across companies and service providers. This is used as common open sources stack for implementing the server-side applications and web services.
Applicability	In case of ReCaM, almost LAMP implementation is followed with the difference of using Java Server Pages (JSP) instead of PHP scripting. This set of technologies are used for implementation of resource catalogue platform server.
Benefits/reasons to use	It is beneficial for transferability of server-side application and code to use common and well known open platform bundle. Sticking in Java programming language in implementations, across the developments, is beneficial for code reutilisation.
Weaknesses	Use of JSP instead of PHP could be a weakness for transferability.

2.4 Systems Engineering Platform

The Systems Engineering Platform supports the planner of a new plant or line (greenfield approach) to develop and design this production line for a new or changed set of product families, also taking into consideration uncertainty.

Name	JSON (JavaScript Object Notation)
Description	JavaScript Object Notation or JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute–value pairs and array data types (or any other serializable value). It is a very common data format used for asynchronous browser/server communication. In some systems, it is used as a replacement for XML. JSON is also the preferred structure for simpler information transfer and common when the exchanged data is temporary in nature.
Applicability	JSON will be used in data exchanges defining call arguments for a service or its responses. These are normally interfacing between applications, and hidden from the user. In ReCaM, this is the format that will be used in the data architectures, e.g., resource catalogues, production plan and reconfiguration plan, system state-space model representation and system layout representations.
Benefits/reasons to use	JSON can be beneficial for ReCaM project in the following terms: - It is easy to read (for example, compared to XML). - It can handle the same amount of data as XML in fewer code lines.
Weaknesses	The limitations of using the JSON standard in ReCaM may be related to: - It is a standard not as widespread as XML. This may result in a higher risk in terms of exploitable results and dissemination purposes. However, JSON has shown a significant increase in its popularity in the last years, thus its application to the project's purposes is promising. - The JSON protocol is not as powerful as XML (no namespace support, thus the poor extensibility).

Name	REST web services
Description	Representational State Transfer (REST) is a way for providing interoperability between computer systems on the Internet. REST is an architectural style, and an approach to communications that is often used in the development of Web services and in particular Cloud-based Services. In the REST architectural style, data and functionality are considered resources and are accessed using Uniform Resource Identifiers (URIs), typically links on the Web. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.
Applicability	Every participating application in ReCaM will provide either RESTful server or client, thus allowing the ReCaM system to communicate to higher level systems. RESTful services allow requesting systems to access and manipulate textual representations of Web resources using a uniform and predefined set of stateless operations. As stated in D2.3, ReCaM architecture will provide a Brokerage service, which the clients can use for searching and interlinking to the service providers.
Benefits/reasons to use	In general, the protocol is *lean* and *lightweight*. RESTful servers are simple to test and analyse with standard web browsers, without prior assumption and implementation of any complicated communication protocol. Therefore, in ReCaM, REST has been preferred over the SOAP (Simple Object Access Protocol) style because REST does not leverage as much bandwidth, which makes it a better fit for use over the Internet. Other reasons are following: - this approach separates the content from its usage and therefore reduces the complexity of the interfaces - performance - scalability - portability of components - it is client/server: the business logic is decoupled from presentation; - It is stateless: all messages exchanged between client and server has all the context needed to know what to do with the message.
Weaknesses	The implementation of RESTful services in ReCaM may present some disadvantages, such as: - every client needs to be configured to connect with every server and service they are utilizing; - the client is sending all messages with redundant information (bandwidth and latency);
Name	SSL (Secure Sockets Layer)

Description	Secure Sockets Layer (SSL) is a cryptographic protocol that provides security over the communications along a computer network. It is the standard security technology for establishing an encrypted link between a web server and a browser. This link ensures that all data passed between the web server and browsers remain private and integral. SSL is an industry standard and is used by millions of websites in the protection of their online transactions with their customers. Among others, its applications include: web browsing, e-mail, internet faxing, instant messaging, and voice-over-IP.
Applicability	In ReCaM project, REST web services (System Engineering Platform) are transmitted over the internet as Base64 encoded text. For this reason, a simple BASIC authentication over HTTP (Username and Password) was not considered as a sufficient measure. Using data encryption techniques like Secure Socket Layer protocol (SSL) is therefore recommended. Currently, all the APIs in the Flexible System Engineering platform are protected by a self-signed certificate. The latter will be substituted by a proper certificate signed by a trusted Authority before the validation phases.
Benefits/reasons to use	SSL encrypts the data exchanged by the ReCaM Platforms end-to-end (browser to server and vice versa) and, together with other security provisions, provides a simple and effective measure to protect sensible data from eavesdropping or similar attacks. Moreover: - the connection is secure because of symmetric cryptography; - integrity of connection is guaranteed since each transmitted message includes a message integrity check; - forward secrecy is allowed.
Weaknesses	The use of SSL for ReCaM may present the following weaknesses: - Certificates may result to be costly. - Slow. The information that is send has to be encrypted by the server. This leads to performance issues as it takes more server resources than if the information was not encrypted.

Name	XML (Extensible Markup Language)
Description	The XML standard provides a flexible way to create and transfer information formats electronically. It is based on a Markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable through use of tags that can be created and defined by users. Data structured in this format can be shared via the public Internet, as well as via corporate networks. Thus, XML is playing an increasingly key role in the exchange of a wide variety of data on the Web and elsewhere.
Applicability	The data communication requirements between various data sources, such as Enterprise resource planning (ERP) systems and Management executive systems (MES) can be efficiently carried out through the usage of XML standard. The information exchange between different ReCaM tools and applications can be based on this standard. The structured format of XML is compatible with the requirements of representing the data structures for representing information about Mechatronic objects, Products, Product processing, system configuration definitions, etc. that are required by the system engineering platform. Moreover, different interacting software service application program interfaces (APIs) providing information to the platform can easily transform their contents to this data format when information needs to be sent. The same can be done for services that are provided by the system engineering platform.
Benefits/reasons to use	The XML standard is beneficial to be used for the ReCaM system engineering platform for the following main reasons: - XML is extendable, thus this allows new requirements which are not included at the first version, to be easily integrated by extending the applicability of the platform. - It is both human and machine readable. In circumstances where a less familiar user needs to access investigate MO, product and reconfiguration related data, the XML format is understandable.
Weaknesses	The limitation of XML standard in ReCaM are basically complementary to the advantages of JSON. In particular: - It may be harder to read. - It requires longer code to represent the same amount of data.

2.5 Production and Reconfiguration Planner

This building block supports the integrated production and reconfiguration decision making, by addressing the production requirements and the goal of minimising energy waste. Concerning the standards, this block will refer to the same standards as in section 2.1.

2.6 Production Execution and Control

This building block manages the system in the state of operation. Therefore, this block receives the production and reconfiguration plan, checks the availability of the current MOs in the system and calls the executable capabilities to execute the production schedule. Additionally, the building block sends the production state / feedback to the ERP and MES as well as to the statistical data recorder.

Name of standard	oneM2M
Description	oneM2M calls itself an interworking framework that standardises the interfaces of network participants like devices and services as well as the principals of the communication. The architecture is based on three layers and 3 types of devices. The first layer is built by application entities (AE). These contain the logic of the system and provide access to the states and methods of MOs via specified interfaces, for example. The middle layer covers the Common Service Entities (CSE) which provide often used services, whereas the Network Service Entities (NSE) provide transportation methods.
Applicability	The communication between the mechatronic objects or services is a very critical part regarding interoperability of products from different suppliers. If interaction between different entities cannot be granted easily, it is highly likely that no wide ecosystem of MOs will be established and building up production lines will very often require effort to customise standard processes to fit ReCaM needs. To avoid this scenario, oneM2M was chosen as the middleware to connect devices and applications, as it is an already widely spread standard that is supported by a huge community of companies and organisations from different domains.
Benefits/reasons to use	*See description for oneM2M in section 0*
Weaknesses	*See description for oneM2M in section 0*

Name	mDNS and SD with multicast
Description	Multicast Domain Name System (mDNS) is a zero-configuration service that enables resolving of host names to IP addresses without the need of a DNS name server. mDNS creates IP packets similar to those of unicast DNS (e.g. A/AAAA record), and send them to a well-known multicast address. The host with the corresponding host name will then replied the sender with its IP address. Service Discovery (SD) with multicast further

	enhances the zero-configuration resolving of host names to IP addresses by utilising the SRV, PTR and TXT DNS records to allow advertisement and discovery of service types, service instances and their domain names, and optional arbitrary text description of the service instance.
Applicability	mDNS and SD is used as discovery service to notify a network about the presence (appearance as well as disappearance) of a device. Specific services can be notified to be available for further use.
Benefits/reasons to use	In comparison to other (network device discovery) protocols, mDNS is a well-known and, in conjunction for service discovery, use-case specific configurable protocol. It allows a distributed server-less name resolution and dependable service discovery and is a lightweight implementation which is important for very small embedded devices.
Weaknesses	Multicast UDP packets are blocked by most firewall by default and requires configuration to allow mDNS to work. Bonjour's implementation of mDNS only allows for link-local multicasting (i.e. only works in a single LAN domain. There are no NS record anywhere in a mDNS domain therefore caching on each client may be needed to improve performance. Without authentication and encryption, it is vulnerable to spoofing attacks by any machine within the multicast IP range. The querying client is responsible to resolve conflicting mDNS responses.

Name	Extensible Messaging and Presence Protocol (XMPP)
Description	XMPP is a communication protocol based on XML. It is a free and open specification started in 1999 for server-client based communication. From an architecture point of view, XMPP communication networks are decentralised like as email servers. It can be used in isolated intranet network to build a robust and secure message network. The core specification addresses security topics like SASL and TLS. One of the most interesting features of specification is its extensibility. Based on XML extensions can be created to be used on top of the basic communication framework.
Applicability	XMPP messages are used to negotiate and autoconfigure an MO's service that requires more data to be communicated than what is allowed in a, for example, mDNS's TXT record.
Benefits/reasons to use	As it is an extensible protocol which completely separates the transport, the connection negotiation and the message payload, it can be extended to the use-cases in ReCaM. The message payload can hold any data structure or schema that will be defined or used (ex. Data structures defined on top of oneM2M or newly introduced for communicating data payloads from the other work packages in the ReCaM project). XMPP with

	its extensions has the potential to cover multiple aspects that are needed for industrial but flexible adjustable I4.0 level communication. XMPP is highly extensible with many formalised extensions allowing free definition of the structure and semantics of data being communicated. Specially for distributed automation environments it is sometimes needed to have a server-less peer-to-peer connection.
Weaknesses	Heavier weighted compared to competing protocols such as MQTT therefore may become a limitation on memory-constrained devices. XMPP is very verbose and the overhead for XML stanzas and stream-setup can be big if the message transferred is small. XMPP lacks an official Quality-of-Service functionality although a yet-to-be-accepted extension has been proposed. Industrial adoption of XMPP outside the chat domain is still limited albeit growing interest.

Name	**IEC 61499**
Description	The international standard IEC 61499 defines engineering of distributed control systems based on function blocks. That means the application logic can be distributed among several control units. The standard favours separation of software and hardware as it defines interoperability, portability and configurability: • Interoperability: Communication during interaction between control units and field devices, even of different manufacturers. • Configurability: Uniform interfaces for configuration of control units and field devices. • Portability: For storage and exchange of engineering data there are open respectively standard formats used. This makes it easy to exchange engineering-tools. In IEC 61499 the cyclic execution model of IEC 61131 is replaced by an event driven execution model. The event driven execution model allows an explicit specification of the execution order of function blocks. The event-based controller is the core that makes easy distribution control program solutions possible. Data and processing have to be synchronised. Therefore, events and data are associated via block interfaces (event-data interfaces).
Applicability	One example of using the IEC 61499 standard is the IEC 61499 UDP based cross communication protocol. The IEC61499 standard itself does not define a device cross communication protocol. However, the standard defines compliance profiles. Cross communication protocol specification is included in such compliance profiles, since it builds the basic for interoperable automation devices. The IEC61499 has a compliance profile reference specification as appendix included. The cross communication of the nxtControl IEC61499 runtime implementation is based on the

	IEC61499 standards reference definition, but has been extended to fulfil industrial level needs for reliable and real time based cross communication. The protocol is used to cross connect nxtControl IEC61499 runtime systems. In conjunction with additional connectivity protocols (ex. DDS, XMPP, RESTful API, MQTT, mDNS-SD) combined in an edge device, this device will be used as gateway component between higher level protocols and shop floor distributed control systems. Such a device can also be used to enable connectivity of legacy devices (conventional SPS) to the cyber physical system enabled I4.0 environment. A second example is the C# IEC61499 Dynamic Adapter communication. Previously, all adapter communications crossing the resource boundary have to be defined at compiled time as the runtime must establish static communication channels between each pair of communicating resources. Now it is possible for the runtime to instantiate open-ended communicate channel with unspecified endpoint that only connects on demand by the request of a foreign device (not necessary another 61499 device) and then allow IEC61499 adapter communication to be built on top of it.
Benefits/reasons to use	IEC61499 supports the integration of different automation tasks very good. The standard helps to reduce the complexity, but at the same time does not limit the flexibility. The IEC61499 offers as well the opportunity to create hardware-independent and vendor-independent control runtime systems. The object orientation of IEC61499 applications is an object model, which expands the encapsulation of data (IEC61499 variables) and methods (algorithms, function block networks) via a defined interface – the Event-Data-Interface to integrate more automation aspects. Basic function blocks describe their connection between events and the executable algorithms via a graphical state diagram (ECC = Execution Control Chart). The readability of such blocks improves greatly. The option to use type-in-type nesting (Composite function block) allow to structure function blocks and allow to build a software representation of real world objects to be represented as a software object. The topology of the system is in IEC 61499 described in the system model. The system model is used to represent an overall automation system and is defined as 'a collection of devices interconnected and communicating with each other by means of a communication network consisting of segments and links'. Each device is capable of performing a set of independent tasks that coordinate by means of a communication network and, thus, constitute a distributed system.
Weaknesses	The spreading of the standard has been slow in the last few years, but there are current projects that want to counteract this.

3 ReCaM standardisation strategy

The main approach of ReCaM towards the topic of standardisation is to base the implementations of the results on common standards in order to enable a wide applicability and access for external partners. By doing so, the project's results can easily be applied by external partners in order to join a potential ReCaM system. Therefore, this document is giving a summary of the standards applied within the implementation of the results. From this point of view, ReCaM is primarily using existing standards instead of creating new standards. But there are other activities going on towards creating or influencing standards as well, which will be described in the next paragraph.

One possible path to spread the results to be available for a wider range of organisations is, communicating and using them in other projects and activities. Within Germany, the results are spread towards the "Plattform Industrie 4.0" as well as the research project BaSys 4.0, which is funded by the "Federal Ministry of Education and Research" and aims towards developing a standard for describing the resources within a future connected industry. This project was initiated by the "Plattform Industrie 4.0", and is therefore having a high visibility. Within Bosch, the results of ReCaM are used for the planning of the production, while BaSys 4.0 aims at developing an open IT platform for the production. The insights of ReCaM towards the production and assembly will also be used as an input for BaSys 4.0 and therefore, for developing the new standard for a future open platform for production. Additionally, the efforts for standardize the Product Management in ReCaM for flexible production are promising, being integrated in the flexible robotics platform used for CESA use case. It is planned to continue the development of this technology in VERSATILE project.

4 Conclusions and Consequences

The first preliminary analysis of existing standards for their applicability in ReCaM, presented in D7.2, resulted in a list of standards that were a starting point for more detailed analysis. During the execution of the project, all standards were continuously analysed. As the project was evolving and the architecture of the system was defined only the standards that fulfilled the requirements of the system were implemented. This document presents the implemented standards in tables stating their applicability, benefits and weaknesses. It gives a good overview of all the standards and in which part of the architecture they were applied.

The standardization strategy of ReCaM is based on influencing existing standards and also providing a new possibility on how existing standards can be combined and implemented to get a ReCaM based production system. The plan for future activities is to spread the results to other active projects and initiatives. This would introduce the results of the project to a wider audience and also increase their acceptance.

List of Figures