Project information

Project title	Coordinating Optimisation of Complex Industrial Processes
Project acronym	COCOP
Project call	H2020-SPIRE-2016
Grant number	723661
Project duration	1.10.2016-31.3.2020 (42 months)

Document information

Deliverable number	D3.1
Deliverable title	Software architecture description for the runtime system
Version	1.0
Dissemination level	Public
Work package	WP3
Authors	TUT
Contributing partners	TUT, BFI, IDE, VTT
Delivery date	27.3.2018 (M18)
Planned delivery month	M18
Keywords	COCOP, system architecture, software architecture, software system, integration

This project has received European Union's Horizon 2020 research and innovation funding under grant agreement No 723661.

VERSION HISTORY

Version	Description	Organisation	Date
0.1	First outline	TUT	12.1.2018
0.11	Module classes added	TUT	22.1.2018
0.12	Text added in various places	TUT	23.1.2018
0.13	Architecture requirements elaborated	TUT	25.1.2018
0.14	Entity type explanations added; text added to "information security"; text added to "integration requirements"; restructurisation	TUT	26.1.2018
0.15	Text added in various places	TUT	15.2.2018
0.16	Text added to "Requirements for Systems Integration"	TUT	20.2.2018
0.17	More text added to "Pilot Case Requirements > Copper" More text added to "Enabling Distributed, Loosely Coupled Systems" Added a figure to "Message Bus Architecture for Scalable, Loosely Coupled Systems"	TUT	22.2.2018
0.18	Requirements chapter combined with architectural philosophy	TUT	27.2.2018
0.19	Executive summary. More text for architectural requirements and core aspects	TUT	27.2.2018
0.20	Text added to multiple places; some existing text improved	TUT	28.2.2018
0.30	Improvements based on internal review	TUT	27.3.2018
1.0	Finalization for deliverable submission	TUT	27.3.2018

EXECUTIVE SUMMARY

COCOP aims for plant-wide monitoring and control of complex distributed processes. Control of complex industrial processes is typically dependent on computational models and on having up-to-date data available to be used when optimising production and reducing the environmental impact. Many of these processes are distributed which is also the case for the control systems typically used at plants and at sub process levels. The distribution makes it challenging to monitor and control the processes when information can not be easily communicated and direct integration of information systems is challenging.

In this document, the general COCOP system architecture is presented. This deliverable is the first of two deliverables describing the software architecture, and focuses on laying out the fundamental architecture directions both for the internal composition as well as external integrations. The architecture presented does not try to create a new control system but rather a concept for exchanging information and data in distributed systems, so that coordinating control applications can be created in a scalable and more flexible manner.

The target of COCOP is to enable the decomposition of the plant-wide coordination task into unit process level scheduling tasks. The decomposition and associated coordination promote the use of a model-based, predictive, coordinating optimisation concept in integration with plant's automation systems. This means that existing local control systems are used to control sub processes, and the plant-wide control applications built using the COCOP architecture act as the coordinating layer transmitting events, restrictions, set points and targets for the plant-wide monitoring and control.

The general COCOP architecture is based on loose coupling of systems using a message bus architecture. The main arguments are scalability, reducing the amount of point-to-point integrations, decoupling data and event producers from their consumers, and increasing flexibility of applications that can be built on top of the conceptual architecture. A fundamental design principle is retaining the freedom in what kind of messages are transmitted in order to leave sufficient room for case specific applications to specify and build applications actually supporting the production.

ABBREVIATIONS

Abbreviation	Full name
DCS	Distributed Control System
ERP	Enterprise Resource Planning
HMI	Human Machine Interface
MES	Manufacturing Execution System
PLC	Programmable Logic Controller

TABLE OF CONTENTS

1 Architectural Requirements and Core Aspects

Plant-wide industrial process control and monitoring applications face challenges from system complexity as well as from multi-disciplinary networked system integrations. The distributed control systems used in production are typically vendor- and application-specific, which makes it challenging to integrate into plant-wide control functions. Production processes may also span beyond one single plant, which introduces new requirements on system integrations in order to optimise production beyond the local processes. Further challenges may arise from the synchronisation of possibly conflicting data and events, e.g., combining estimated values with actual measurements. A traditional periodic control approach (that scans or queries for values and then decides its control actions) may prove complex, rigid and laborious to implement, especially if several point-to-point connections to other systems need to be implemented and maintained.

1.1 Decoupled Data-driven and Event-driven Architecture

The COCOP system architecture strives for scalability and enabling extensive utilisation of data either as refined information or as massive amounts of raw data. One of the main drivers is also decoupling the information producers from their consumers through a centralised bus or - depending on the implementation - a pool of queues for data and events.

The event-driven approach comes from the requirement for reactive actions possibly also in real-time. In some cases, the end of a process step may trigger the start of another. For instance, when a batch of process finishes, it may enable the execution of another process step for that particular piece of material.

From a performance point of view, scalability is achieved by removing redundant queries to low-level systems and by having a central managed bus architecture in between producers and consumers of information and data. A centralised message bus allows for efficient scaling, caching and managing access independent of the consumers and producers. From a systems integration point of view, a centralised bus provides uniform access to the data thus reducing engineering effort for the plant-wide control applications. A centralised bus, however, can introduce a single point of failure, and additional preventive measures might be needed to ensure reliability, e.g., through redundancy or in restrictions how plant-wide monitoring and control applications are required to operate in case of manual intervention.

This kind of architecture benefits especially the development of new plant-wide monitoring and control applications in a platform-like development framework. From a development perspective, the burden is on integrating existing systems with a multitude of different communication interfaces to the bus and the event-driven approach. The development of adapters is needed, as it is not intended to replace existing control systems but integrate them into the new (COCOP) control environment (full replacement would be expensive and create a significant hurdle towards implementation in plants as a new systems with new, unknown risks with potential production losses).

1.2 Enabling Distributed, Loosely Coupled Systems

Distribution is inevitable in industrial environments. The larger a production plant is, the more distribution there usually is. A production plant may even consist of several factories that have a dedicated task each.

Loose coupling is a fundamental design principle in COCOP. The goal is to enable an environment where various modules interact in a way that hides the internal details of each module. That is, the interfaces of modules should only expose their functionality or contents, not the underlying platform or implementation technologies. In addition, the integration technologies should be such that enable the modules to operate with minimal dependencies to one another, and their mutual integration should be easy.

Point-to-point integration should be avoided. Point-to-point means direct connections between systems, which often leads to a high number of direct dependencies between systems. Then, scalability issues would appear, as one of more systems are changed, updated or replaced in the future.

The downside with a bus approach is that producers of data and events do not know who and how information is utilised. As a result, additional information security measures may be needed in the message bus or even in the plant-wide control applications built on top.

COCOP focuses on the integration between systems rather than the systems themselves. Thus, COCOP does not provide any engineering tools, run-time of controllers or similar concrete applications. Instead, it is an intermediary for connecting and accessing events and data.

1.3 Monitoring and Controlling Distributed Production

In production plants, a relaxing characteristic is that most systems are static. In contrast, there are also environments where new network nodes may appear or leave the network at any time (i.e., the network is dynamic). Such environments include the fleets of mobile machinery or vehicles, for instance. Fortunately, the network nodes of a static production system do not typically appear or disappear arbitrarily. Naturally, any node may temporarily lose its connectivity due to a failure, and any related network nodes should have a design that is robust enough to remain operable in such occasions. Still, the set of available nodes is determined by the physical production equipment. That equipment is always installed, operated and uninstalled by plant personnel. Thus, the network of nodes does not change its structure very often, and the plant personnel commits those changes.

The information available and of value in control may change more frequently. This includes, for example, the deployment of new measurement systems or laboratory procedures, the use of improved models and algorithms more accurately estimating process values as well as other results available from processing vast amounts of data. One of the intentions with COCOP is to have a more flexible platform that can make use of external or processed information that can be used e.g. as a constraint in optimisation tasks.

1.4 Integration to Existing Systems

1.4.1 Integration Requirements

Multiple types of existing systems may be integrated with COCOP. The following table gives a few examples. Each example is explained after the table.

System	Examples of potential needs
DCS, PLCs, laboratory, quality control, other production-related resources	Measurement values from production processes
HTTPS	Hypertext Transfer Protocol (HTTP) over Transport Layer Security (TLS)
MES	Plant-level production coordination; plant-level production-related information, scheduling restrictions, availability of resources
ERP	Enterprise-level production coordination, resources and financial costs
Logistics	Constraints related to material flows, transport capabilities

DCS (Distributed Control System), PLC (Programmable Logic Controller) are systems that provide means for human workers to control complex, distributed production processes.

In some production plants, laboratory systems are utilised to provide information to help process control, e.g. adjust the process conditions. They analyse the substances that are involved in a production process. They may, for instance, estimate the concentration of specific substances, which may provide advice to reach closer-to-optimal operation. Laboratory systems are often also part of Quality control systems help reaching the desired quality of the end product. With appropriate quality control the yield can be improved and also the amount of waste may be reduced, which increases productivity.

A Manufacturing Execution System (or MES) may provide production-related information at the plant level. Rather than controlling the individual low-level processes, MES systems coordinate plant operation as a whole. Concerning COCOP, MES systems may provide schedules or other coordinative information not available in the unit process level.

ERP (Enterprise Resource Planning) systems have business matters as the scope. From the production point of view, an ERP system may, for instance, receive production-related data to indicate production performance, but it may also coordinate production in the enterprise-wide scope.

Logistics is an important aspect in production optimisation. To optimise production, multiple logistics-related factors may be relevant, including the timely delivery of raw materials, intermediate products or an appropriate amount of materials in the storage.

1.5 Pilot Case Requirements

The COCOP architecture concept is developed for process industry in general. However, in the early stages of development, the design is focused on developing a solution to support the pilot cases in copper and steel production.

1.5.1 Copper

In a plant that refines copper from sulphide ores, there is a great degree of distribution and concurrency in production. The unit processes of such plant are operated locally, but the material flows between the unit processes create dependencies. The most important unit processes of a plant may be, e.g. (Schlesinger et al., 2011, pp. 1-12):

1. Flash smelt furnace (FSF)
2. Peirce-Smith converters (PSC)
3. Anode furnace and casting (AF)
4. Electrorefining
5. Melting, casting

Starting from unit process 1, each unit process provides the raw material for the next. Consequently, any processing-related shortcomings in a unit process may have adverse effects on the following phases.

Besides, efficient production requires further coordination. The waste (or slag) from each phase requires further processing, because it still contains some copper. Thus, a slag cleaning furnace (SCF) or a slag concentrator may exist, or the slag may also be circulated back to FSF. Furthermore, some of the end products are harmful to the environment. Some heavy metals are generated from the ore, and the gases generated from each unit process contain, for instance, sulphur dioxide. To process sulphur dioxide, the plant has off-gas handling and ventilation systems connected to an acid plant to transform the sulphur dioxide to sulphuric acid. As the capacity of the acid plant is typically limited and, on the other hand, the emissions of sulphur dioxide should be minimised, the operation of the acid plant may restrict the execution of other aforementioned unit processes. Within COCOP, the document D2.3 System Requirements Specifications (2017) explains the requirements that are considered in particular.

In the coordination of copper production, correct timing is essential. The execution of each unit process should be scheduled considering the entire plant. Event-driven operation is important, as changes in process states (such as start or end) may trigger or even hinder an operation. Still, due to slow process dynamics, the required resolution of response times is approximately one minute in most tasks. In comparison, some other production plants may require a timing accuracy of fractions of a second.

In summary, the operation of a copper plant requires coordination with a plant-wide perspective. Thus, for optimal operation, each unit process should be operated within plant-wide constraints.

1.5.2 Steel

The increase of the Chinese steel production has permanently changed the worldwide market. The fierce competition has led several producers to close their production while the ones still operating have seen their margins drop. Even in this situation, European producers have showed higher resilience due to the specialisation and the high quality of their products. Nonetheless, the European companies deal with the overcoming challenge of maintaining their competitiveness while reducing their emissions considerably and competing with countries with lower costs for energy and workforce.

In this framework, European producers are making great advances towards the optimisation of their production, targeting not only environmental improvements but also increased quality and resource use.

Two main process routes for steel production are Electric Arc Furnace (EAF) and Blast Oxygen Furnace (BOF). BOF and EAF processes both produce steel as the end product while having a different way of producing it. BOF plants use Iron ore, Coal and limestone as primary raw materials while the EAF use scrap steel as the main input. In the BOF, Sinter plant, Coke furnaces and blast furnaces are used for preparing the material which is then fed into the Blast oxygen Furnace. In the EAF route, the scrap metal is directly loaded into the electric arc furnace, thus having a much simpler route for the production. However, these two main routes represent only the first stage of the steel production, which is called primary metallurgy. From this point, the steel goes through a long set of processes before the end product is obtained. First, the steel goes to the secondary metallurgy where the composition of the steel is adjusted to the specific needs of the client. After this, the steel is casted into subproducts (slabs, billets, ...) that are easier to work with. These subproducts are then used as input for the production of steel products.

As mentioned earlier, the main optimisation targets of steel plants are the reduction of raw material use (either scrap or primary materials) and the efficiency of the processes in energy terms. Observing the European steel makers, a clear picture of the complexity of the production chain can be obtained. A regular steel plant can produce around 100 different grades of steel (different chemical compositions) and for each one of them, be able of producing several products (e.g. bars, sheets) in a wide variety of sizes. Accordingly, the steel plants can account for hundred of thousands different end products being manufactured in their plants, including in this manufacturing process, several unit processes. With this scenario, it is clear that optimising at local level the efficiency of the different processes is not enough. Great efforts need to be made plant-wide for arranging optimal production planning and for ensuring final product quality.

These two aspects have a great impact in the overall efficiency of the plant, the first, by reducing the energy and time usage for sub-process adaptation for new orders and the second, reducing scrap generation and ensuring client satisfaction (high quality products). In the COCOP project, main focus will be in the optimisation of the final product quality (reducing rejection rate) through higher knowledge (and control) of the involved production steps and their parameters.

2 Platform Architecture Design

2.1 Message Bus Architecture for Scalable, Loosely Coupled Systems

The architecture design is illustrated in the following figure. The communication platform enables message exchange between various entities that do not have any direct mutual dependencies.

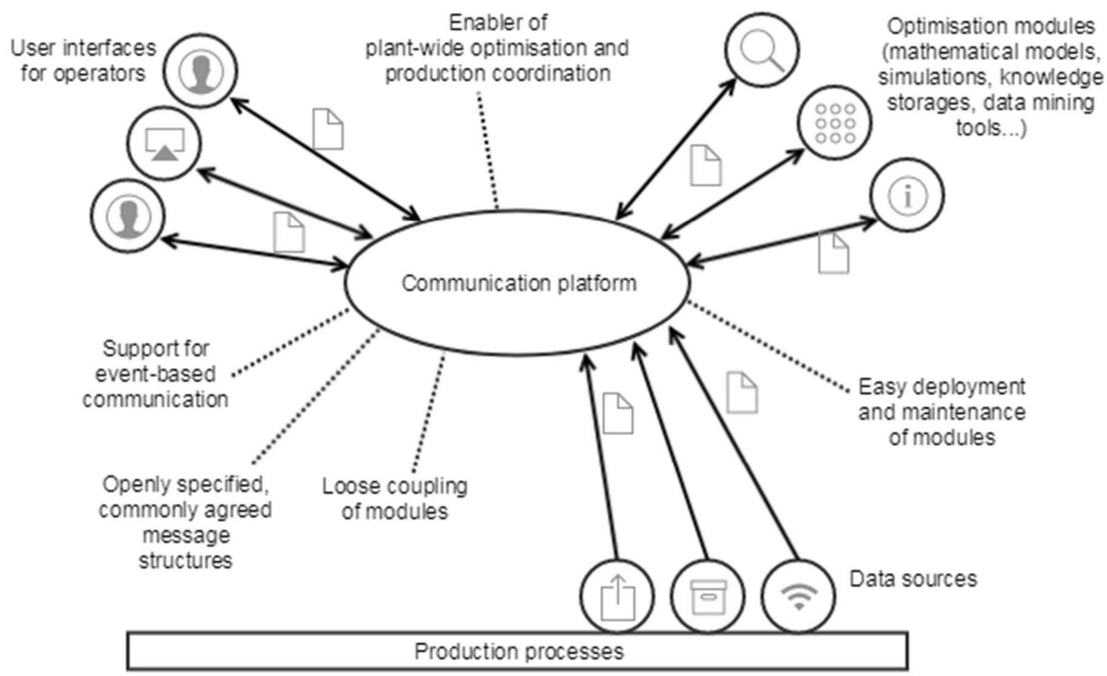

Figure 1 Architecture illustrated

COCOP aims for a message bus based approach mainly brokering data and event messages in contrast to Enterprise Service Bus (ESB). ESBs are typically used to deploy, connect and manage integrations between different information systems in the form of services, and they typically include support for several protocols including advanced transformation capabilities as well (Bhadoria et al. 2017). Including the necessary adapters from existing systems to the COCOP concept architecture the final technology stack can resemble that of common ESBs.

To implement the COCOP concept, application-specific customisation and specification is possible and even necessary. The COCOP architecture does not explicitly specify or limit what kind of messages are transferred between system components, although suitable message structures are presented in D3.5 Interface and protocol definitions. Similarly, COCOP does not enforce certain messaging patterns, although several patterns are supported.

2.2 Consumer Application State Considerations

A message bus communication can have different kind of implementations and features. The most primitive ones transmit messages received from producers either to all consumers or only those that have registered for the particular messages. More advanced message bus implementations include advanced routing, caching and access management. Some implementations can even store messages for a certain duration and may, for example, retain the most recent messages for new consumers to receive. This is especially beneficial for new data or events that is seldom updated. From a plant-wide monitoring and control application development point of view, the state management responsibility is transferred to the particular applications and case implementations.

Implementing the event-driven approach will require either that 1) all operational data is periodically updated (even if no changes occur), 2) most recent messages are retained (as some events can happen very seldom), or 3) there are means to also query recent data and events. To enable integration with various systems, adapters are typically utilised to wrap existing systems to new interfaces. The first option of always updating values is straightforward to implement in adapters but increases the amount of data transferred unnecessarily. This also puts more pressure on consumer applications to know when and if to process new measurements. The second option is easier to implement in client applications but reduces the number of standard message buses and protocols available as many of such features are implementation specific. The third and the favourable option provides most flexibility and also enables request-response like behaviour but incur additional functional requirements for adapters realising the query capabilities.

2.3 Supporting Client-Server Communication

It has been identified that many traditional systems currently in use operate on a polling or request-response basis. For example, Human Machine Interfaces (HMI) scan periodically for new values through queries and update the user interface displays accordingly. A message bus solution - depending on the implementation - may not store the last value, and therefore it is required for the COCOP concept to have request-response behaviour available for client-server communication as well.

In industrial environments, there are also de facto protocols in use, such as OPC UA (OPC, 2015), that provide standard connectivity between systems. The standardisation may enable close to plug-and-play behaviour when integrating components. Encouraging the use of standard solutions facilitates the interoperability of industrial control systems, and it is one of the COCOP architecture objectives. Interfaces and protocols suitable for realising the COCOP system architecture are detailed in D3.5 Interface and protocol definitions.

To realise client-server communication - while using an asynchronous message bus as the core - requires new functionality for the adapters or systems directly integrated to the COCOP architecture. In addition to pushing new information onto the bus, the adapters need to listen for specific requests from others and then be capable of returning the requested information onto the bus. These messaging patterns will be developed during the project and detailed in the future deliverable D3.7 Software architecture description for the runtime system (update).

2.4 Integration Approach

To enable integration with various systems, adapters are utilised. Adapter is an approach where an existing system is wrapped behind another interface (see the following figure). In principle, any existing interface may be wrapped with any other interface. Thus, to unify interfaces, adapters are a powerful approach; no matter how heterogeneous interfaces the existing systems have, it is possible to unify them. A similar approach has been documented as a generic design pattern in software design (Lasater, 2006, p. 206).

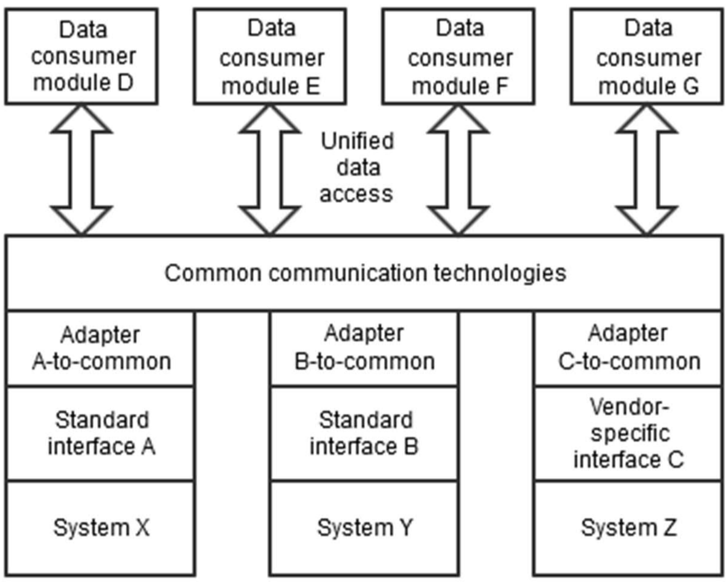

Figure 2 Adapter-based integration illustrated

2.5 Entity Types and Message Structures

The COCOP system consists of entities (also called components or modules). The entities have various types as given in the following table. Each entity type is explained in the coming subsections. The entities and their associated APIs and message structures will be detailed during the project and described in the future deliverable D3.7 Software architecture description for the runtime system (update).

Entity type	Purpose
Data source	To retrieve data from existing production-related information systems
Data mining tools	Data mining tools to discover information in production data

Entity type	Purpose
Models and optimisation	Modules that optimise production
Data output	Modules that provide information to external systems (e.g., operator interfaces)

2.5.1 Data Source Entities

Data sources provide production-related data from related information systems. The messages may cover, for instance, measurement values from the production process, both actual and historic data. Potentially, the data may also cover equipment state or process status information. For instance, the information about finishing a process step could trigger a scheduling operation for the next step. Also, condition monitoring and wear information is of importance as it failing equipment can cause bad quality or complete disruption of the production process.

2.5.2 Data Mining Tool Entities

Data mining and data analytics describe a group of known techniques used to extract information from data. Examples of these techniques are multivariate non-linear statistics, neural networks or decision trees. Beneath (offline) analytics, data-based or data-driven models can be developed that can be used, for example, as a soft sensor (regression model) or as decision support for the operators (classification models). Together with existing physical or first principle models, data analysis is often used for parameter estimation on the basis of historical data or in various combinations as a hybrid model.

Within the framework of COCOP, the Data Mining Tool Entity is used in the steel application case for the development of data-driven models. The necessary data is requested from the Data Source Entity via the message bus. No data or metadata is transmitted directly to the COCOP solution via the message bus. All information collected is used indirectly during modelling in the Model Entity or in the design of the data preprocessing during the data request of the models.

2.5.3 Model and Optimisation Entities

Modelling and optimisation tools help the management of production processes. They may provide, for instance, production schedules or other assistance.

2.5.4 Data Output Entities

The data output entities refer to any items that expose information to higher-level systems, such as the graphical user interfaces of production operators. The actual data output is likely provided by computational models and other optimisation-related entities, but there may also be other output entities that gather or reformat the information supplied by other entities.

2.5.5 Message Structures for Entities

To enable communication between entities, appropriate message structures must be specified. It may be possible to reuse a single message structure to provide data from all entity types. Although the entity types have varying requirements, a generic message structure may be sufficient to hold data items. Still, there are likely message-related details that are specific to entity types. The message structures will be detailed in the planned update of this deliverable.

COCOP does not discriminate (there are no limitations) as to whether the message structures are utilised for internal or external communication. Due to the generic nature of the structures, there is no need for such limitations.

2.6 Information Security

User (or system) authentication should be required to access information transmitted using the COCOP platform. Using authentication, it is possible to prevent unauthorised access. Even in cases where the system is utilised within a single facility, layered security is desirable. Then, even if a malicious user were able to break into a system, authentication would form another obstacle. However, some production plants may actually run multiple facilities separated from each other. In such cases, some communication may occur via public network routes, which makes user authentication and encryption obligatory. In modern, distributed plants it is also increasingly common that several systems and system vendors communicate using the same networks.

Data encryption should also be applied in communication, and especially in multi-vendor or globally spanning networks. Then, there is no straightforward means to interpret any messages that could be captured. Furthermore, a suitable encryption technique also hides the traffic of user authentication. The importance of encryption increases in complex, geographically distributed production plants.

The following figure illustrates the security approach. Both user authentication and data encryption are required for secure communication.

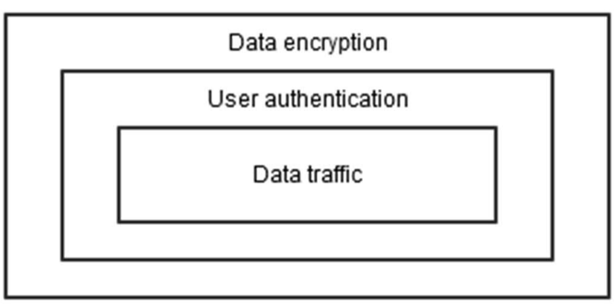

Figure 3 The security approach covers encryption and user authentication

Information security may be implemented either using the built-in features of the message bus (if available) or separately in an application specific manner. Using built-in features is obviously limited to the chosen message bus implementation; most implementations offer (application level) authentication and encryption, e.g. username/password or certificate and HTTPS. Application- or case-dependant authentication requires further implementation but increases the flexibility of the approach, as it can be integrated to any existing access rights management means being used. Embedding application-specific encryption of messages further improves the security, as not even the broker can interpret the message content.

2.7 Robust Module Design

The COCOP architecture does not specify the internal design of the modules that are connected to the system but only their external behaviour. The modules may provide data to others or consume data. A module may also be both a consumer and a provider.

However, each module should be designed robust and independent (in accordance with industry practices). For instance, this means that the internal algorithms of a module must not get confused if some message is submitted multiple times for one reason or another. Although consistent messaging is a design goal as well, each module should process data in way not too fragile in terms of inconsistencies.

3 Conclusion

This document outlined the general principles of the COCOP system architecture. It is the first out of two deliverables describing the general COCOP system architecture that is being developed iteratively during the project.

COCOP aims for plant-wide monitoring and control of industrial processes. These processes are typically dependant on computational models as well as having up-to-date data available when optimising production and the use of resources. Industrial processes are distributed - also globally - which makes it challenging to monitor and control the processes when information can not be easily communicated and the direct integration of multiple information systems is required.

COCOP does not implement a new control system but a model-based, predictive, coordinating optimisation concept that operates in integration with existing automation systems of a plant. The COCOP system architecture is developed as a platform for exchanging information and data in distributed processes, thus enabling coordinating control applications to be created in a scalable and more flexible manner.

The general COCOP architecture is based on loose coupling of systems using a message bus architecture in a data-driven and event-driven style. The arguments for this design are scalability, decoupling message producers and consumers, reducing direct system integrations, and facilitating building of new monitoring and control applications based on the conceptual architecture. An important design principle is retaining flexibility in what kind of messages are transmitted in order to leave sufficient room for case specific applications to specify and build applications supporting the production.

This document defined general architecture concepts, means for communication, integration requirements, and identified internal entities of the COCOP system. The future deliverable D3.7 Software architecture description for the runtime system (update) will further detail the roles of the identified entities, their APIs and message structures, messaging patterns as well as use of adapters for integrating existing information and control systems into COCOP.

References

"D2.3 System Requirements Specifications," https://cocop-spire.eu/content/deliverables (accessed 22.2.2018), 2017.

Robin Singh Bhadoria, Narendra S. Chaudhari, Geetam Singh Tomar, The Performance Metric for Enterprise Service Bus (ESB) in SOA system: Theoretical underpinnings and empirical illustrations for information processing, Information Systems, Volume 65, 2017, Pages 158-171, ISSN 0306-4379, https://doi.org/10.1016/j.is.2016.12.005.

Lasater, C. G., Design Patterns. Wordware Publishing Inc., 2006.

"OPC unified architecture specification part 1: Overview and concepts. Release 1.03," https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/ (accessed 16.1.2018), OPC Foundation, 2015.

F. Jammes et al., "Technologies for SOA-based distributed large scale process monitoring and control systems," IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, QC, 2012, pp. 5799-5804. doi: 10.1109/IECON.2012.6389589

Schlesinger, M.E., King, M.J., Sole, K.C., and Davenport, W.G., "Extractive Metallurgy of Copper," Elsevier, 2011.