Document title: Use cases analysis

Version	Status	Date
1.0	final	2019-12-10
Author	**Contact**	
F. Montori, M. Tatara, S. Bocchio	federico.montori2@unibo.it, marek.tatara@dac.digital, sara.bocchio@st.com	

Use cases analysis

DAC: Marek Tatara
marek.tatara@dac.digital

IUNET: Federico Montori
federico.montori2@unibo.it

ST-I: Sara Bocchio
sara.bocchio@st.com

Abstract

In this document, we analyze each use case starting from their architecture. We then address their gaps by proposing dedicated needed tools and toolchains for each of them. For each tool, following the approach and the best practices stated in O2, we outline their main characteristics that are necessary to frame them within the big picture of its use case as well as in the phases of the engineering process proposed.

ECSEL EU project 826452 - Arrowhead Tools
Project Coordinator: Professor Jerker Delsing | Luleå University of Technology

Table of contents

1. UC-1 [CAMEA, BUT, CVUT, EXPLEO and AIT]: Automated Formal Verification

Contact: Lukáš Maršík [l.marsik@camea.cz]

1.1 Use Case Overview

The Automated Formal Verification Use Case concentrates on one of the phases critical in the development of embedded devices and that is *verification*. More concretely, the use case aims at systems developed for *intelligent traffic surveillance* in an SME (CAMEA in particular). In this use case, we consider automated techniques and toolchains for verification and testing that aim at an improvement of engineering processes of development of intelligent traffic systems in CAMEA (and systems of similar kinds). These systems include complex and interconnected HW and SW components written in different high- and low-level languages, and have to ensure various safety, security, and performance requirements.

1.2 Identified Gaps

Currently, there is no sophisticated verification process employed in CAMEA. All new systems are roughly validated in the company using basic tests and simulated data/inputs and situations. Then, the system is mostly field-tested on installation sites, and most critical bugs are caught during a short testing period in a real environment. Further, adapting the system to new environments (a different road topology, usage of new sensors, new algorithms, etc.) is a time demanding and costly process, which can introduce new errors. In particular:

1. The use case is not currently using latest advanced verification techniques (be it static analysis with formal roots or advanced dynamic analysis) at all. The aim is to improve the situation by allowing the company to use advanced verification techniques.
2. The systems developed by CAMEA are currently not connected to the Arrowhead Framework. Advanced verification techniques should also assist in checking that the newly developed interconnection is correct.
3. Moreover, most of the tools that are of interest for verification and testing of the considered systems are currently not interconnected in any way.

As an alternative to verification, automated synthesis of parts of the considered systems, schedules, etc. may be considered, reducing the amount of needed verification efforts. Such approaches are currently not in use in CAMEA at all.

1.3 Identified Needed Tools

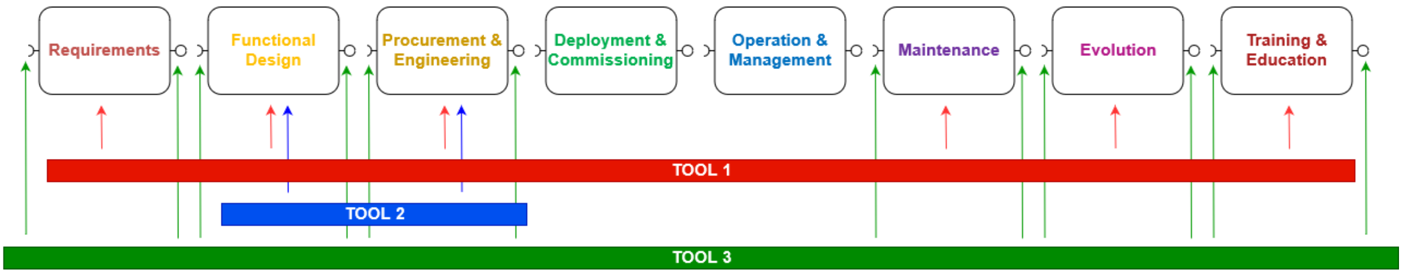
Tool 1. Advanced verification and testing tools enabling both static and dynamic analysis across the design process of SMEs.
   a. Design-time tool
   b. Connected to the design-time tools of the AHF
   c. Across all phases apart from Deployment & Commissioning and Operation & Management EPP1, EPP2, EPP3, EPP6, EPP7, EPP8
   d. Inputs: [environment conditions, device setup, specifications to be checked]
   e. Outputs: [verification results]

Tool 2. Connectors between some components of the intelligent control systems developed in CAMEA and the AHT framework.

a. Run-time tool
b. Connected to the AHF
c. Functional Design; Procurement & Engineering EPP2, EPP3
d. Inputs: [requests from others sources]
e. Outputs: [outputs for other entities within the AHF]

Tool 3. Open Services for Lifecycle Collaboration (OSLC) adapters for selected verification and testing tools

a. Design-time tool
b. Connected to the design-time tools of the AHF
c. Across all phases apart from Deployment & Commissioning and Operation & Management (phases in between) EP-I1, EP-O1, EP-I2, EP-O2, EP-I3, EP-O3, EP-I6, EP-O6, EP-I7, EP-O7, EP-I8, EP-O8
d. Inputs: [environment conditions, device setup, specifications to be checked]
e. Outputs: [verification results]



2. UC-02 [PHC, TUE] Engineering processes and tool chains for digitalized and networked diagnostic imaging

Contact: Geran Peereen [geran.peeren@philips.com]

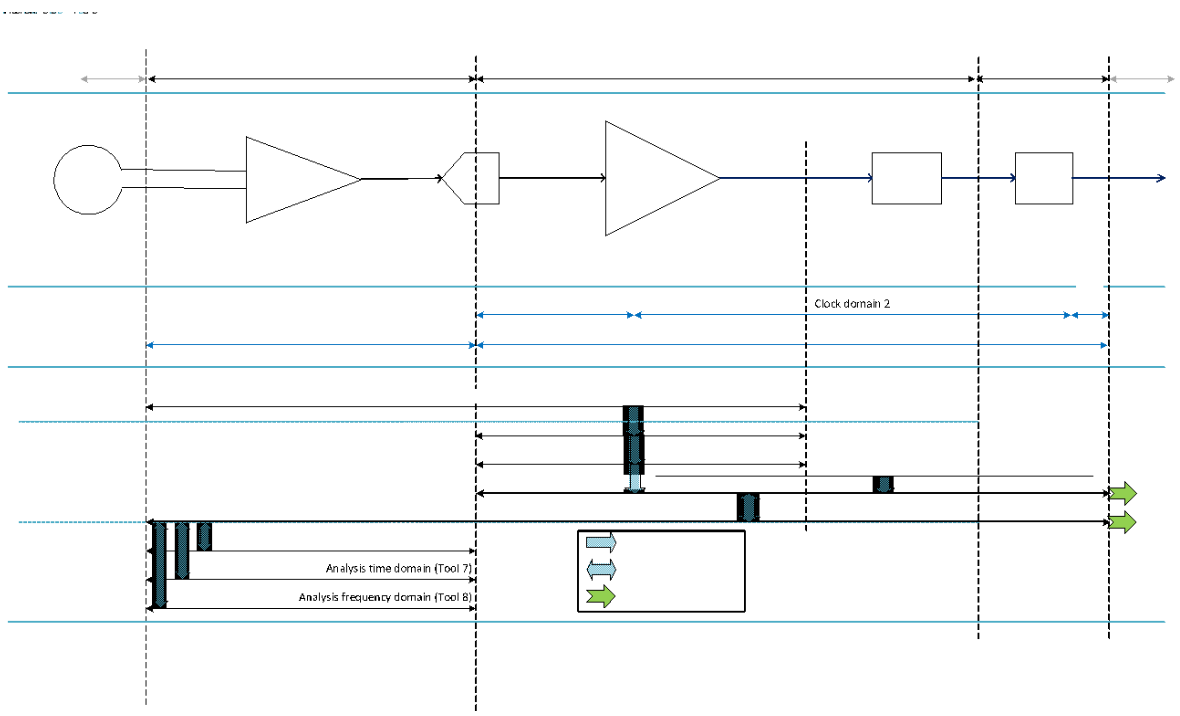
2.1 Use Case Overview

In an MR-receiver, signal processing is in general performed by hardware for the analog domain and firmware for the digital domain. In its design, Matlab models are used for a number of functional signal processing blocks. From the models the required parameters for the implementations are derived, as well as the calculation methods for the digital domain.

2.2 Identified Gaps

Currently, these models are translated by hand, and the output of the model is compared with the output of the implementation to test the correctness. In addition, necessary adaptations to the calculations in the firmware, e.g. using fixed-point arithmetic vs floating point in the Matlab model, lead to differences between output of the model and the implementation.

Further, during system lifetime maintenance is required and components from the original design may no longer be available. This requires a form, fit and function compatible replacements together with verification that the operation of the repaired system remains identical to that which was previously validated. For this purpose the output of the model is not as important as the output of the original system. This verification can be done by recording the outputs of the original system in response to stimuli and confirming that the outputs of the updated system remain consistent. In itself this is relatively straightforward, but in reality there may be variations due to the updated components and so a more intelligent comparison is required which recognizes 'acceptable' changes and flags 'unacceptable' ones.

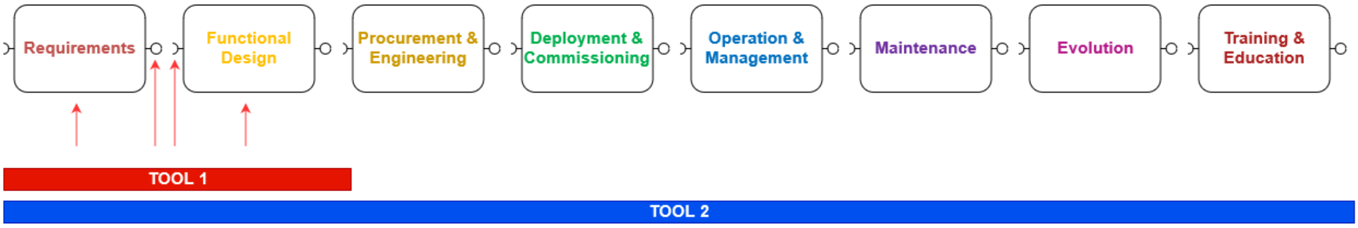


2.3 Identified Needed Tools

Tool 1. **Connection tool**: A tool is needed to ensure that the models of functional blocks in the digital signal processing domain are sufficiently reflecting their corresponding implementation in VHDL and to ensure that future changes to system components (via in-lifetime moves and changes) can deliver verifiably consistent outputs.

a. Design-time tool
b. Should be connected to the AHF
c. EPP1, EPP2, EP-O1, EP-I2
d. Inputs: VHDL models, DSP domain functional blocks (Matlab)
e. Outputs: statement of compliance, test reports, consistency reports.

Tool 2. **Diff tool**: A tool is needed that can record responses of the system in a given (normally, original) configuration and then compare that with the responses of an updated system for the purposes of verifying that updates to the configuration or implementation of the system do not result in unexpected changes in outputs, while filtering out 'acceptable' variation.
Currently, a more specific description for such tool is in progress.



3. UC-3 [ULMA, UC3M, REUSE]: Integration of electronic design automation tools with product lifecycle tools

Contact: Jose María Alvarez Rodríguez [josemaria.alvarez@uc3m.es]

3.1 Use Case Overview

This use case is applied to whole development lifecycle of critical systems for IoT and SoS. More specifically, it focuses on the possibility of making a better reuse of physical models covering the abstraction, selection, representation and customization of system artefacts from the whole development lifecycle. The reuse of any system artefact goes beyond the mere discovery of potential reuse and it must focus on evaluating what can be reused (requirements, analytical models, descriptive models, test cases, etc.) when a match is discovered. To do so, quality usually has some impact since it is assumed that high-quality system artefact may help to improve the reusability factor of a system artefact. Furthermore, in this use case, there is another major objective focusing on the improvement of traceability to be able to keep trace from the very early stage of development to the final release of a complex product. Traceability is a critical activity to ensure that every system artefact exists for a reason.

The main motivation of this use relies then on being able to access and exchange different system artefacts (and tools) enabling a reusability layer based on interoperable standards.

3.2 Identified Gaps

In general, all tools offer native ways of connecting and accessing system artefacts. However, there is some room for improvement:

- In many cases, functionalities available in the tools (via their user interface) are not exposed so it is complicated to provide a strategy to reuse not just data, but operations as well.
- The use of native connectors allow developers to easily integrate tools from the same vendor. However, if a third-party wants to consume that information the cost and time to understand both data and communication models prevents a proper reuse of data and operations. As a consequence, some manual work must be done by engineers instead of improving the connection capabilities between tools.
- Although the aforementioned drawback could be mitigated by the use of standards, the reality is that the interpretation of standards may vary from tool to another and the serialization of system artefacts using standards usually do not include all information available in the tools. Thus, a proper and complete reuse is not possible.
- All these points lead us to think that sometimes is not a technical problem, but a lack of a conceptual framework to enable interoperability within the toolchain.

3.3 Identified Needed Tools

Currently, specific definitions of inputs and outputs of each tool is in progress.

Tool 1. IBM Doors – for system requirement definition
   a. Design-time tool
   b. Not connected to the AHF
   c. EPP1
   d. Inputs: TBD
   e. Outputs: TBD

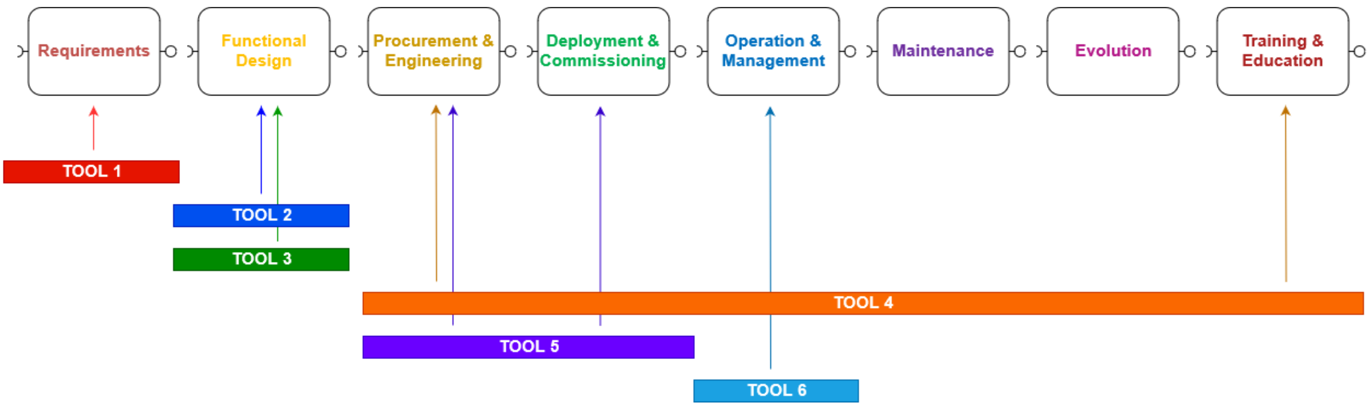
Tool 2. Architecture Definition
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2
  d. Inputs: TBD
  e. Outputs: TBD

Tool 3. Design definition
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2
  d. Inputs: TBD
  e. Outputs: TBD

Tool 4. Implementation
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP3, EPP8
  d. Inputs: TBD
  e. Outputs: TBD

Tool 5. Verification & Validation
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP3, EPP4
  d. Inputs: TBD
  e. Outputs: TBD

Tool 6. Information Management
  a. Run-time tool
  b. Connected to the AHF
  c. EPP5
  d. Inputs: TBD
  e. Outputs: TBD



4. UC-4 [ASML, ICTG, TUE]: Interoperability between (modelling) tools for cost effective lithography process integration

Contact: Ramon Schiffelers [ramon.schiffelers@asml.com], Sven Weber [sven.weber@asml.com], Koen van Wijk [koen.van.wijk@ict.nl], Oscar Reynhout [oscar.reynhout@ict.nl], Sander Thuijsman [s.b.thuijsman@tue.nl], Ferry Timmers [f.timmers@tue.nl], Alireza Mohammadkhani [a.mohammadkhani@tue.nl], Jeroen Voeten [j.p.m.voeten@tue.nl], Marc Geilen [m.c.w.geilen@tue.nl], Jan Friso Groote [j.f.groote@tue.nl], Michel Reniers [m.a.reniers@tue.nl], Loek Cleophas [l.g.w.a.cleophas@tue.nl]

4.1 Use Case Overview

ASML is the world's leading provider of complex lithography systems for the semiconductor industry. The design process of these Cyber-Physical Systems of Systems involves multi-disciplinary engineering teams focusing on functional specification and verification of scenarios and mono-disciplinary engineering teams focusing on the realization of these scenarios in a platform composed of mechanical, optical, electrical and software components.
These engineering disciplines each use a specific set of engineering methods, tools and technologies that are loosely coupled both on a syntactic and semantic level. This has a major impact on engineering efficiency. It hampers verification early in the development process, especially concerning system-wide (performance) aspects (e.g. throughput and accuracy). In addition, it hinders system evolvability and deployment, i.e. introducing new scenarios or adapting existing ones.

4.2 Identified Gaps

To significantly improve engineering efficiency, the goal of this use case is to establish seamless syntactic and semantic interoperability between the multi-disciplinary modelling tools enabling rapid development and deployment of (new) machine calibration, performance and diagnostics test scenarios and effective prediction and trading-off of key system aspects concerning performance and correctness. The objective of this use case is to develop a tool chain, mainly focusing on the functional design phase, that establishes seamless interoperability between modelling tools to facilitate multi-disciplinary engineering teams providing functional engineers with an efficient feedback loop to develop and qualify calibration, performance and diagnostic test scenarios. The transition between phases in the baseline is rather manual as descripted in the following: Matlab is used during the requirements / investigation phase (1), if the requirements phase is successful the functional design (2) is captured with Word and Visio. These unprecise specifications are manually translated to a software specification during the procurement and engineering phase (3) and manually implemented in Python software in the same phase using version systems ClearCase. The deployment and commission (4) use Linux patching to deploy to the field. The operations and management (5) of the scenarios are captured in field procedures in Word. When maintenance (6) is needed on the software the scenario is retrieved from ClearCase and updated and newly deployed to the field. The evolutions in general lead to a new scenario which includes the whole above described phases. Training (8) of the scenarios is included in the field procedures. Several tools are already implemented in the loop, although the level of integration in the baseline is low. In the following table the complete set of tools (legacy and proposed) is shown:

AHT-EP phase	UC-EP tool/methodology groups
Requirements	Matlab, Word documents

Functional design	Word documents & Visio diagrams Matlab Stateflow / LSAT (scenario and platform specification) CIF/SDF3 (scenario synthesis) mCRL2 (formal analysis (model checking))
Procurement & engineering	Python code, ClearCase versioning
Deployment & commissioning	Linux patching
Operations & management	ClearCase / AIR / FCO
Maintenance	ClearCase / AIR
Evolution	ClearCase / AIR
Training	Field procedures documented in Word

4.3 Identified Needed Tools

Tool 1. **Stateflow** (developed by MathWorks) is a control logic tool used to model reactive systems via state machines and flow charts within a Simulink model. Stateflow uses a variant of the finite-state machine notation established by David Harel, enabling the representation of hierarchy, parallelism and history within a state chart.

a. Design-time Tool
b. Not connected to the AHF
c. EP-I2, EPP2, EP-O2
d. Inputs: [Graphical specification of state transition diagrams, flow charts, state transition tables, and truth tables. Models are persisted using XML format]
e. Outputs: [Simulation results, code]

Tool 2. **LSAT** (developed by ASML, TNO-ESI and TUE) provides a formal modeling approach for compositional specification of both functionality and timing of manufacturing systems. The performance of the controller can be analyzed and optimized by taking into account the timing characteristics. Since formal semantics are given in terms of a (max, +) state space, various existing performance analysis techniques can be applied.

a. Design-time Tool
b. Not connected to the AHF
c. EP-I2, EPP2, EP-O2
d. Inputs: Specification of available resources and their capabilities and timings in terms of actions and activities, plus specification of requirements, safety and functional automata in CIF. The specification is in the form of a DSL-model. The DSL itself is defined by means of an EMF-Ecore metamodel. The models are persisted in an XMI format. The tool also provides a concrete combined textual/graphical interface to input, edit and view the model.
e. Outputs: A throughput optimal scenario/schedule, which also conforms a 'schedule DSL' which is defined in terms of an EMF-Ecore metamodel.

Tool 3. **CIF (C**ompositional **I**nterchange **F**ormat for hybrid systems, developed by TUE) is an automata-based modeling language for the specification of discrete event, timed, and hybrid systems. The CIF tooling supports the entire development process of controllers, including, amongst others specification, supervisory controller synthesis, simulation-based validation and visualization, verification, real-time testing, and code generation.

a. Design-time Tool
b. Not connected to the AHF
c. EP-I2, EPP2, EP-O2

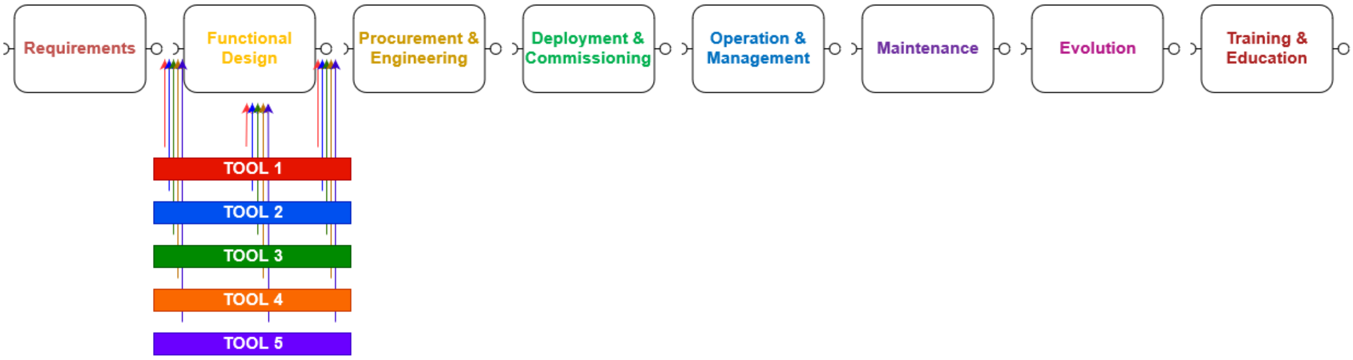
d. Inputs: A set of requirement models and plant models. Plants are specified in terms of hybrid or discrete (extended) finite state automata. Requirements are specified in terms of automata or predicates over automata states and event labels. The models conform to the CIF DSL which is defined by means of an EMF-Ecore metamodel. The models are persisted in an XMI format. In addition, a concrete textual syntax is available
e. Outputs: A maximal permissive, nonblocking, controllable supervisor automaton can be synthesized that ensures that the supervised plant satisfies the requirements. Code can be generated from automata, for example conversion to PLC, UPPAAL or mCRL2 specification. Also many CIF to CIF transformations exist to create specifications that behave in the same manner, but are expressed using different language constructs. For simulation, visualizations are available by means of function graphs or SVG drawings

Tool 4. **SDF3** (developed by TUE) is a tool for analysis and synthesis of Synchronous DataFlow Graphs (SDFGs). It includes an extensive library of SDFG analysis and transformation algorithms as well as functionality to visualize them. It also includes analysis algorithms for switching (max,+) models. The tool can also create SDFG benchmarks that mimic DSP or multimedia applications.

a. Design-time Tool
b. Not connected to the AHF
c. EP-I2, EPP2, EP-O2
d. Inputs: Dataflow models (in XML format), optionally platform models
e. Outputs: Performance properties such as latency and throughput bounds, Mappings, Component abstractions for the QRML DSL that detail the budget-performance trade-offs

Tool 5. **mCRL2** (developed by TUE in collaboration with University of Twente) is a formal specification language with an associated toolset. The toolset can be used for modelling, validation and verification of concurrent systems and protocols. The toolset supports a collection of tools for linearisation, simulation, state-space exploration and generation, and tools to optimise and analyse specifications. Moreover, state spaces can be manipulated, visualised and analysed.

a. Design-time Tool
b. Not connected to the AHF
c. EP-I2, EPP2, EP-O2
d. Inputs: A process algebraic description of the behavior of a system and a set of properties to be formally verified
e. Outputs: State-space of the concrete behavior as a labeled transition system. Boolean verdict per property and counter examples or witnesses



5. UC-5 [KAI, IFAT, IFAG, UC3M, REUSE]: Support quick and reliable decision making in the semiconductor industry

Contact: Anja Zernig [anja.zernig@k-ai.at]

5.1 Use Case Overview

To support quick and reliable decision making in the semiconductor industry, three tools will be improved and implemented in the course of the project. They are TePEx (Test Pattern Extraction), WHF (Wafer Health Factor) and DR (Digital Reference). At the beginning of the project, the TePEx and WHF are available, but act as stand-alone tools, i.e. an extra software application (implemented algorithm, equipped with a GUI), which can be used by the user parallel to his daily work. The feasibility of the tools for a limited amount of product has already been proven, but the scalability to other products and other company sites is still open. Hence, as a first step within the project, the tools have to be evaluated on a broad product portfolio, to prove their generalization and to detect further possible improvements, based on core user feedback. Then, as the second step, the tools improvements are implemented, but still in the "stand-alone version". Therefore, as a last step within this project, the tools are provided in a way that the final integration into the toolchain, or more specific, the tools output (results of the algorithm) can be implemented in the productive environment. The DR is a tool developed in Productive4.0 that models semiconductor supply chains and supply chains containing semiconductors in a Semantic Web representation, which will be extended in the course of this project.

5.2 Identified Gaps

TePEx and WHF are not integrated in the toolchain yet. This will happen after successful *Tool evaluation*. The phases of tool evaluation, tool improvement and integration can be enumerated. The interactions between such phases is always provided in a manual form (no automation). The UC does not envision the development of new tools, rather, the improvements of the current tools within an automated toolchain. The DR it is not tailored for decision-making based on wafermap analysis but will be investigated during the timespan of the project. The possible improvements in such sense are listed briefly below:

- Possible improvements for TePEx: automated adjustment of single parameters, used in the algorithm; suitable visualization of the TePEx output; granularity of the TePEx output (dependent on core user feedback).
- Possible improvements for WHF: Automated adjustment of single parameters of single components; performance improvement of the algorithm; suitable visualization of the WHF output; use of standards and ontologies to be able to share algorithm output (and receive input) via a common ontology.
- Possible improvements for DR: A flexible, fast and high quality consensus algorithm for updating the digital reference might be needed.

5.3 Identified Needed Tools

Tool 1. **TePEx**, an algorithm able to detect test patterns, which are related to malfunctioning testing equipment.

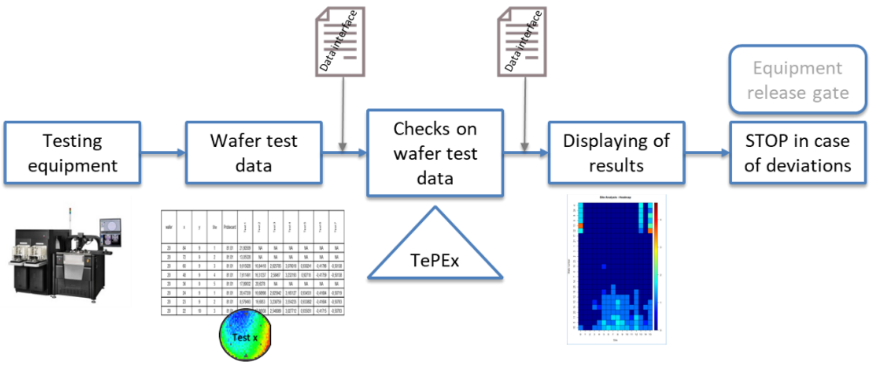


Figure 1: Toolchain integration of TePEx. Two data interfaces are needed.

For the TePEx algorithm, wafer test data are used, which are electrical tests, taken per device. The relation between wafer test data and the testing equipment comes over the probe card, which is the part, connecting the testing equipment with the wafer to take the electrical test as depicted in Figure 2.

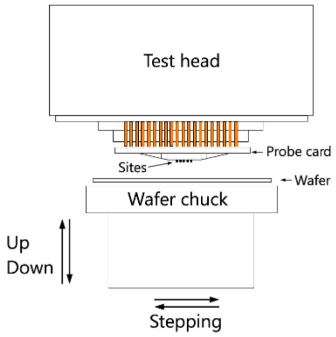


Figure 2: Testing Equipment

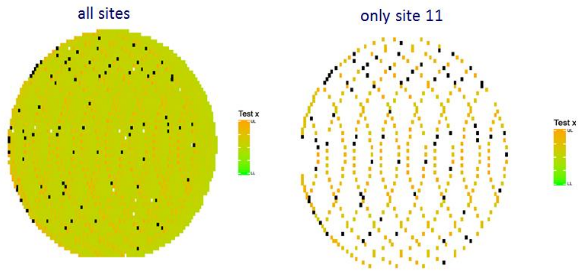


Figure 3: A systematic pattern is clearly visible on the wafermap of the full wafer (left) and can be related to a single site, site 11 (right).

More specific, a probe card consists of multiple “sites” in order to contact and test multiple devices in parallel. In case of e.g. degradation of one site, so-called test patterns are visible on the wafermap, which is a representation of one electrical test on the corresponding x-y position at the wafer, see Figure 3.

Since a probe card consists of multiple sites (see Figure 2), each site needs to be investigated on its own. Hence, the first data interface must provide the following information in order to apply the TePEx algorithm: Wafer ID, electrical tests, site number. With the TePEx algorithm, for each wafer and each site one value per electrical test, i.e. per wafermap, is calculated, indicating whether a test pattern is visible (value > 0) or not (value = 0). Hence, the second interface must provide the output format of the TePEx algorithm, which is one column, containing the calculated TePEx values ≥ 0, additionally to the information from the first data interface (Wafer ID, electrical tests, site number), visualized in the heatmap of Figure 1.

a. Design-time tool
b. Not connected to the AHF
c. EPP1, EPP3, EPP4, EP-O4
d. Inputs: [environment conditions] [Wafer electrical tests]
e. Outputs: [verification] EP-O4

Tool 2. **WHF** (Wafer health factor) - an algorithm able to detect process patterns, which are related to deviations during production.

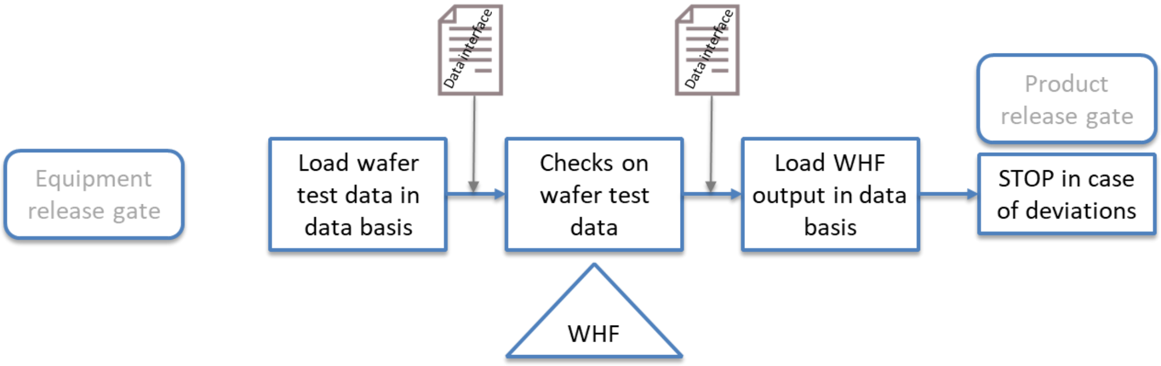


Figure 4: Toolchain integration of WHF. Two interfaces are needed for WHF input and output.

For the WHF, also wafer test data provide the input for the WHF calculation. The WHF calculation consists of a Machine Learning pipeline (preparation, transformation, feature extraction, classification) and acts as an early warning system to detect upcoming critical process patterns at an early stage. Here, compared to TePEx, the same information is needed for the first data interface, except for the site.
The output of the WHF is one value per wafermap, or can also be aggregated to one value per wafer. The value is between 0% and 100% reflecting the health status, dependent on the presence of detected process patterns. Hence, 0% means that the wafer is “unhealthy”, because strong process patterns are visible, whereas 100% means that no critical pattern is present at all and hence, the wafer (or wafermap) is healthy.

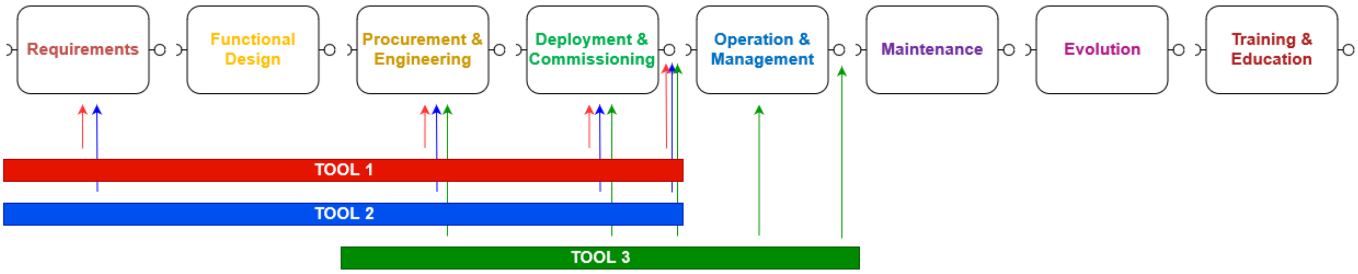
a. Design-time tool
b. Not connected to the AHF
c. EPP1, EPP3, EPP4, EP-O4
d. Inputs: [Wafer electrical tests]
e. Outputs: [health status of the wafermap under investigation] EP-O4

Tool 3. **DR** (Digital Reference): During the engineering phase, integration of added goods and services is a key step for the engineering phase. Thus, interconnectivity and interoperability should be guaranteed. The proposed Semantic Web representation of the Supply Chain, namely Digital Reference is a lingua franca understandable by machines as well as humans. Semantic Web implementation can guarantee interoperability as it creates an abstraction layer that defines concepts and relationships between heterogeneous data sources. Digital Reference allows the interconnectivity between different physical or virtual systems, machines and users.

The Digital Reference is a generic approach to enable sharing and integration of information, databases and tools. The Digital Reference, which is based on W3C standards, goals and procedures leads to a digital twin and hence to digitalization of the industry. However as of today, despite the theory seems to be perfect, the practical implementation lacks details in the ontologies of the digital reference – e.g. it is not only sufficient to have all the details of the production flow up to the latest detail in the Digital Reference in today’s world. Moreover, time dependent custom information needs to be maintained to ensure optimized decisions at the current situation – considering capacity, supply, demand and governmental regulations. Arrowhead promises that it finds, authenticates and orchestrates the sensors for many applications like the supply chain.

The lack of information will be tackled by updating the current version of the Digital Reference that incorporates the Arrowhead (Tools) sub-ontology.

a. Design-time tool
b. Connected to the AHF
c. EPP3, EPP4, EP-O4, EPP5, EP-O5
d. Inputs: [Sensor and Environmental Data]
e. Outputs: [Semantic data representation] (EP-O4, EP-O5)



6. UC-6 [LBB, LQT, POD, ADV, LTU]: Production preparation tool chain integration

Contact: Lars Oskarsson [lars.oskarsson@lindbacks.se]

6.1 Use Case Overview

In this use case we have three parties. Lindbäcks Bygg AB (Lindbäcks), Lundqvist Trävaru AB (Lundqvist) and PodComp AB (PodComp).

Lindbäcks has a process that takes the drawing from the customer and sends it to the automated wood house factory.

Today Lindbäcks sends pdf drawings to PodComp, at PodComp the pdf drawings are manually transfer into a drawing system connected to their production system, here machine files are created and verified in a simulator before they are put into their production line.

Lundqvist Trävaru AB has a system where their customers configure desired products in a 3D configurator. The 3D configurator has the engineering logic and transfers the information to drawings and bill of materials that is the basis for purchasing and production. However, the system is not scalable, and it is getting harder to update and enhance it with new systems. This in turn is an obstacle for growth and further digitalization in the production process.

6.2 Identified Gaps

Lindbäcks has an engineering process from architect drawing to the automated wood house factory. The technology used in the engineering process originates from the 1980s, it consumes a lot of engineering time, since it creates deviations a lot of manual verification is needed and sometimes errors slip in to production that causes downtime and delays in the production. To be able to keep the pace in the factory Lindbäcks buys completed bathroom pods from PodComp, these are lifted in place into the wooden modules that Lindbäcks produce in their factory.

The system is not scalable and it is getting harder to update and apply new systems to it. PODCOMP has a good production and manufacturing line, however a lot of their information flow from order to machine is handled manually. PodComp has a good production and manufacturing line. However, a lot of their information flow from order to machine is handled manually. This is limiting PodComp to be to extend their footprint on the market towards smaller companies or even end users (i.e. more and smaller orders).

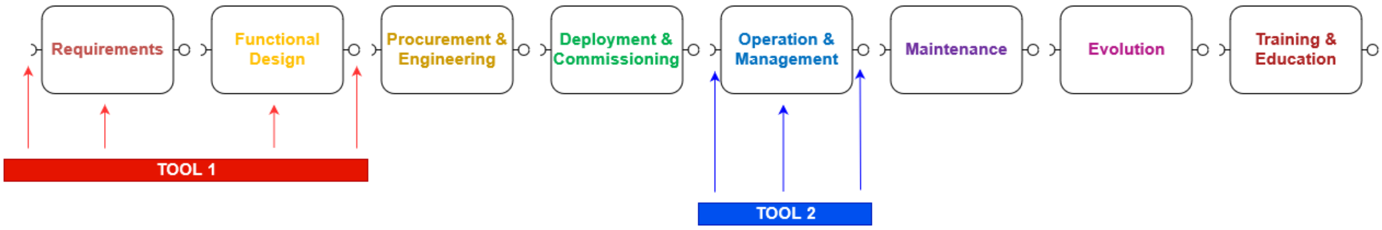
6.3 Identified Needed Tools

Tool 1. **Vertex** - drawing tool with BIM-abilities is being implemented at Lindbäcks to replace DDS

a. Design-time tool
b. Not connected to the AHF
c. EP-I1, EPP1, EPP2, EP-O2
d. Inputs: dwg drawing (EP-I1)
e. Outputs: ifc-files (EP-O2)

Tool 2. **3D configurator** - tool for making machine files understandable by ABB robots from ifc files

a. Run-time tool
b. Connected to the AHF
c. EP-I5, EPP5, EP-O5
d. Inputs: ifc-file (EP-I5)
e. Outputs: machine file (EP-O5)



7. UC-7 [FAUT, IKERLAN]: CNC machine automation

Contact: Carlos Yurre [yurre@aotek.es]

7.1 Use Case Overview

Customers of Fagor Automation usually spend one or two days parameterizing and tuning the machine axes. Customizing HMI is done with a proprietary tool based on legacy technologies on a computer.

Tuning tools are both located at the CNC or with a computer connected with DCOM protocols. Version tracking and adaptation between CNC and tools is usually a source of problems. The new tools should reduce by 50% the time needed by our customers. The new tools will follow the evergreen approach to cope with the versioning problem.

The current tools are mostly integrated in the CNC and as such are difficult to update (need to update the full code). The new tools will follow a modular approach and rely on open source standards for version managing if necessary. Moreover, for better interoperability, the tools will use semantics from standards, as OPC-UA and, more importantly, related companion standards like the already published by VDW for machine tools. This will improve integration with third party tools.

There is currently a CNC simulator for Fagor Automation's CNC. The system is very good for CNC programming training, but can not be used easily with customizing and tuning tools. This simulator will be modified up to some extent with control algorithms as a platform where both intended tools of the project will be demonstrated. In the related Work Package at least a lathe and a milling machine will be shown with a small tutorial for tuning.

7.2 Identified Gaps

The main drawback of the actual tools relate with data exchange formats, interoperability with external or third party tools, and lack of independence between them.

Many of the tools are outdated both in technology and in look or portability. Applications and HMIs are tied to Windows in many cases, and instead of being based on client-server or similar architectures, rely on shared memory or proprietary interconnection mechanisms. All of these characteristics make difficult the connection or adoption of different platforms or third party software.

7.3 Identified Needed Tools

Configuration and mapping Toolchain

Tool 1. *Topology validator*

a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: stored or detected topology: XML
e. Outputs: Javascript Object

Tool 2. *Topology Editor*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Javascript Object, User Input]
e. Outputs: modified topology: XML

Tool 3. *Machine Parameter validator*
a. Design-time and Run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Stored Parameter file (XML), Parameter Descriptors (XML)]
e. Outputs: modified Parameter File: XML

Tool 4. *Legacy Parameter Converter*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Legacy Parameter file (TEXT), Parameter Descriptors (XML)]
e. Outputs: Parameter File: XML

Tool 5. *Connection Editor*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [System Configuration (Javascript Object), Device descriptors (XML)]
e. Outputs: [Connection Map(XML) Parameter File(XML)]

Tool 6. *Parameter Editor*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Parameter file (XML), Parameter Descriptors (XML)]
e. Outputs: Parameter File: XML

Control Loop Tuning Toolchain identified Tools:

Tool 7. *Legacy scope data conversion (Octave or Matlab command tool)*
a. Design-time and run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Acquired Data and Configuration (Matlab.m)]
e. Outputs: [Data Acquisition Data file (JSON?), Data Acquisition Configuration File (JSON, XML?)]

Tool 8. *Fast data acquisition Tool*
a. Run-time tool

b. Not connected to the AHF
c. EPP4
d. Inputs: [Data Acquisition Configuration File (JSON, XML?)]
e. Outputs: [Acquired Data (bin) , Persisted Acquired Data(JSON)]

Tool 9. *Data acquisition configuration tool*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Data Mapper File (XML) , user interaction]
e. Outputs: [Data Acquisition Configuration File (JSON, XML?)]

Tool 10. *Data plotting tool (including logarithmic, Bode, etc…)*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Acquired Data (bin) , Persisted Acquired Data(JSON), Plot configuration (JSON)]
e. Outputs: [Plotted Data (HTML), Plot configuration (JSON)]

Tool 11. *Transfer Function Identification*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Persisted Acquired Data(JSON), Configuration data (JSON, XML)]
e. Outputs: [Linear Model (JSON, XML)]

Tool 12. *Control Loop Optimizer Tool*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Persisted Acquired Data(JSON Linear Model (JSON, XML), Strategy(XML)]
e. Outputs: [Parameter File(XML)]

Smart Graphical 2D5 Editor and operations management Tools;

Tool 13. *dxf importer to geometry*
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [dxf file (TEXT, 2013)]
e. Outputs: [Geometry Description File (JSON)]

Tool 14. *Legacy profile and ISO (G-code) importer to geometry*
a. Design-time tool
b. Not connected to the AHF

c. EPP4
d. Inputs: [*ISO (G-code*)]
e. Outputs: [Geometry Description File (JSON)]

Tool 15. *Technology Tables reader and validator*
a. Design-time and run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Technology Table (XML)]]
e. Outputs: [Javascript Object]

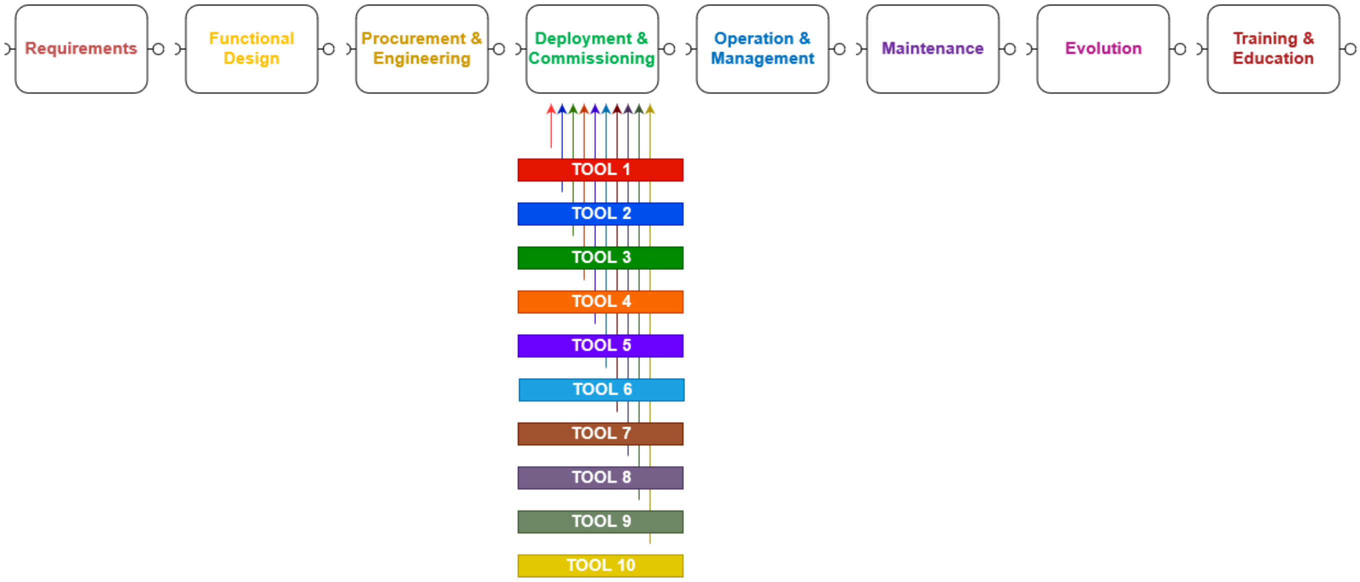
Tool 16. *ISO(G-code) generator from geometry*
a. Design-time and run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Geometry Description File (JSON)]
e. Outputs: [ISO(G-code) File(Text)]

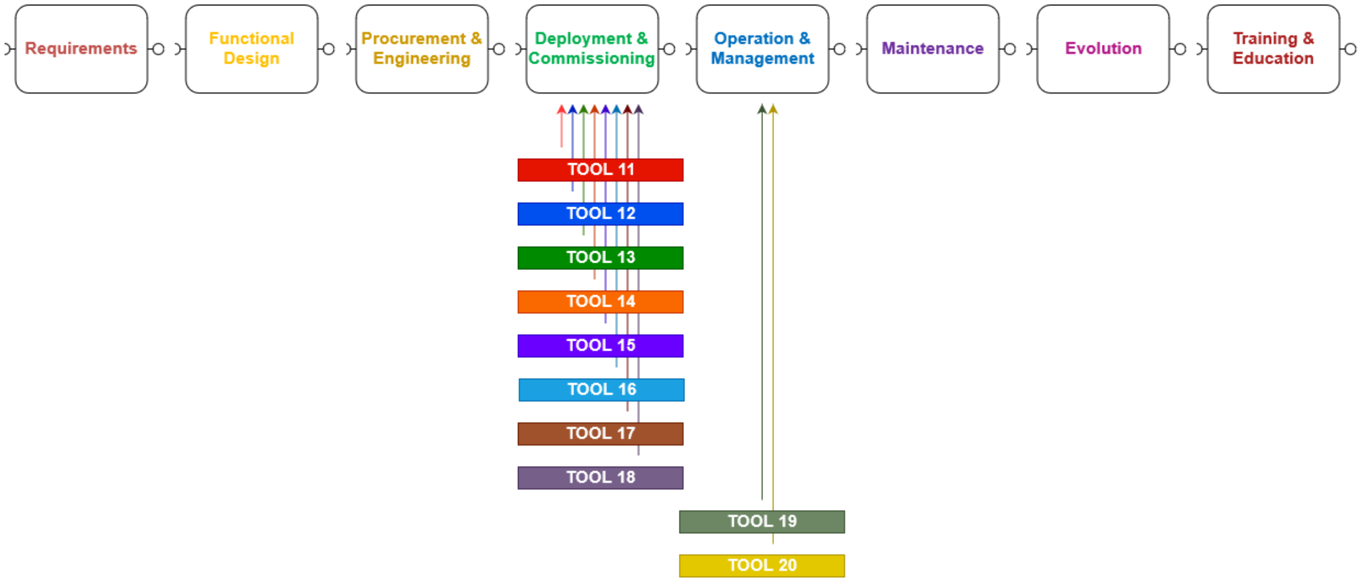
Tool 17. *Part description reader and validator (against tools…)*
a. Design-time and run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Part piece description (STEP?, XML...)]
e. Outputs: [Javascript Object]

Tool 18. *Machining strategies reader and validator (against machine, material, tools)*
a. Design-time tool and run-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: [Strategies File (XML, JSON?), Geometry Description File (JSON), Part Piece description (Javascript Object)]
e. Outputs: [ISO(G-code) file(text), Advanced Operations File(JSON)]

Tool 19. *Geometry Editor*
a. Design-time tool
b. Not connected to the AHF
c. EPP5
d. Inputs: [Geometry Description File (JSON), user input]
e. Outputs: [Geometry Description File (JSON)]

Tool 20. *Advanced CNC operations generator*
a. Design-time tool
b. Not connected to the AHF
c. EPP5
d. Inputs: [System Configuration (Javascript Object), Device descriptors (XML)]
e. Outputs: [Advanced Operations File (JSON)]


Requirements
Functional Design
Procurement & Engineering
Deployment & Commissioning
Operation & Management
Maintenance
Evolution
Training & Education
TOOL 1
TOOL 2
TOOL 3
TOOL 4
TOOL 5
TOOL 6
TOOL 7
TOOL 8
TOOL 9
TOOL 10


Requirements
Functional Design
Procurement & Engineering
Deployment & Commissioning
Operation & Management
Maintenance
Evolution
Training & Education
TOOL 11
TOOL 12
TOOL 13
TOOL 14
TOOL 15
TOOL 16
TOOL 17
TOOL 18
TOOL 19
TOOL 20

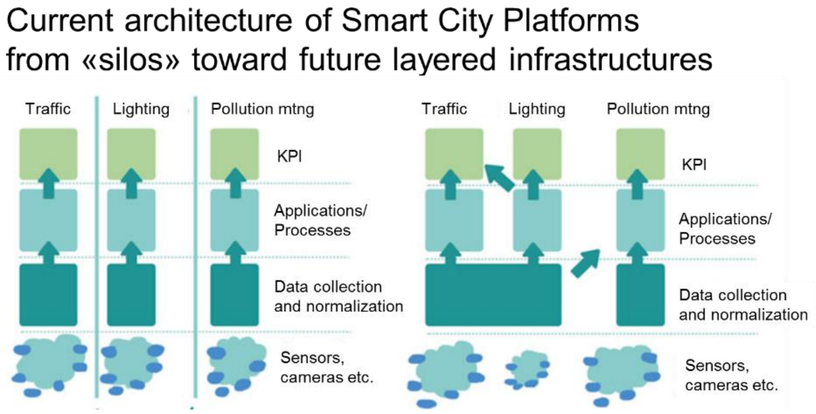
8. UC-8.1 [ST-I, Eurotech, IUNET, POLITO, REPLY, BEIA, ROP] SoS engineering of IoT edge devices (Environmental Monitoring)

Contact: Maurizio Griva [m.griva@reply.it]

8.1 Use Case Overview

Worldwide, cities need to “get smart” about pollution.The World Health Organization estimates that 92 percent of the world’s population doesn’t have access to clean air. This is more than a health problem. The World Bank estimates that in 2013 air pollution was responsible for $225 billion in lost productivity. Action must be taken by policymakers, the private sector, and citizens in order to safeguard the breathed air we breathe. The conditions are worst in cities, where most of population lives. Children are hardest hit by poor air quality, meaning that the pollution in major population centers is already affecting everyones’ future.

Digital technologies are the building tools to tackle this problem head-on, but they have to face huge challenges on size, pervasivity, cost and effectiveness of proposed solutions.
An end-to-end solution that collects air quality information across large areas must be deployed at a fraction of the cost (and size) of previous ICT systems and it must be deployed throughout a city or industrial zone, radically increasing the precision of air quality data. While cities might have once paid $150,000-$250,000 for a single unit, the deployment cost of AHT Tools-enabled, smart city environmental quality digital platform must leverage on low cost sensors and IoT infrastructure at all levels, affordable enough to be distributed throughout many neighborhoods or industrial areas. This is critical as disparities in traffic, population density, and industrial activity can mean drastically different levels of pollution across a city. High deployment costs are mostly due to “silos” from legacy systems and/or proprietary deployments with reduced or no interoperability.



Each device pushes measurements to the cloud, where they are broken down with analytics that are easy to understand. This puts large volumes of high-quality data in the hands of city or industry decision makers, giving them the tools to measure how well clean air policies are working. In a major step forward, this data can also be shared directly with the public, empowering city inhabitants with real-time information about their surroundings. Knowing that pollution levels are high may lead some people to stay home, for example, or change their running route to avoid lung damage.

8.2 Identified Gaps

If we integrate an air quality monitoring system with a smart traffic network, we could detect traffic jams with high levels of air pollution as motors idle. It would be possible to redirect the flow of traffic, or instruct drivers to turn off engines as they wait. Given access to large new pools of city data, entrepreneurs can transform issues like congestion into opportunities with traffic management solutions, connected lighting, and smart parking. City planners and developers will integrate connected features into their design process. There is much to be excited about; the rise of such smart cities will bring many positive changes to residents in tomorrow's urban centers.

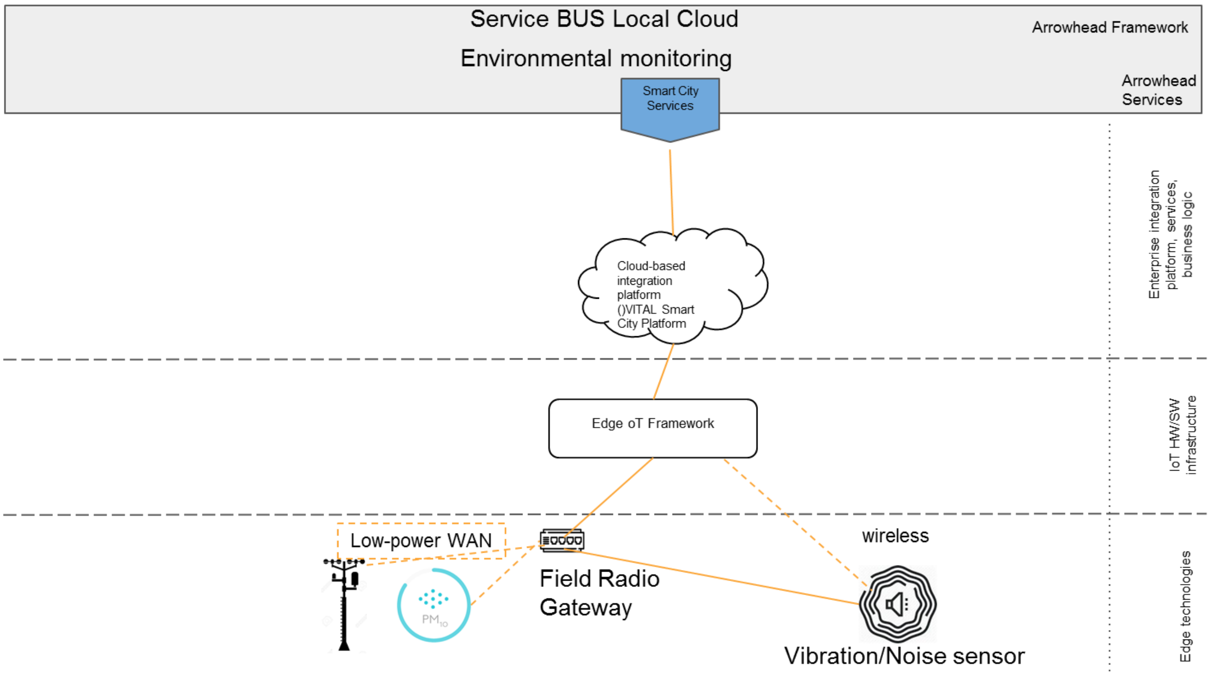
The vision goes beyond the monolithic monitoring platform because in recent years it has been proven that top-down smart city deployment cases lag behind mainly to the difficulty of deriving actions from the huge amount of data streamed out from the sensors. Furthermore, the environmental data can be used in business value chains to increase citizen awareness of public activity (and regulations) that target the urban safety and quality of air, water etc.

The cross-interoperability of sensor data is a major factor that drives the vision, together with the capability of deploying large number of sensing points at a fraction of the current implementation cost., thanks to next generation silicon and components. That vision perfectly matches the Arrowhead Tools vision: " Enable collaborative automation by networked embedded devices". In the Smart City Use Case, automation is mainly pervasive automated sensing and localized actions to react to complex, multi-parameter functions that describe environmental situations and their changes.

A highly pervasive sensing infrastructure must provide chemical, PM 2.5-10, noise, temperature, weather data at sampling times of less than 1 minute and potentially with a spatial granularity of less than 100 m. Such constellation of devices require an extreme attention on power consumption, that can be accomplished only by using next generation silicon sensors, energy-aware software applications and low-power, medium-to-long range wireless communications. Gas sensor node concerning the development of systems and algorithms for being sensitive and selective to CO, Ammonia, H_2, CH_4, benzene, ethylene, ethanol, toluene; Nitrogen dioxide and NO_2 to be proven. Those apply respectively for urban environment (benzene, toluene, NO_2), domestic (CO, CH_4), industrial (H_2, CH_4, ethanol, toluene, ammonia...). It will be necessary the study of potential poisoning and interfering gases Gas sensor compensation algorithms based on temperature and humidity.

Noise levels must be measured against spectrum and signal strengths in order to enable AI algorithms to identify potential recurring patterns and derive meaningful results on root cause and actions. Often, effective activities targeting pollution level reduction require a massive behavioural change of citizen population. Such changes are hard to achieve, and awareness from positive trend data can be used as a valid motivational and rewarding point to keep focus on the pollution reduction targets.

An IoT architecture must take into account the environmental constraints where the sensors will be deployed.The constellation of sensors and edge devices capable of collecting, processing and transmit data from the field will exploit wireless connections that preserve bandwidth, battery duration and extend the sensor's life spanning multiple years without maintenance.



8.3 Identified Needed Tools

The ultimate definition of tools for this Use Case is still in progress, nonetheless we are able to list a high-level proposal for them:

Tool 1. Autonomous sensing enabling kit: Design and integration of smart sensors and smart meters

a. Design-time tool
b. Not connected to the AHF
c. EPP2, EPP7
d. Inputs: TBD
e. Outputs: TBD

Tool 2. Edge side processing enabling kit: Design, configuration and operation of Edge Devices

a. Design-time tool
b. Not connected to the AHF
c. EPP2, EPP7
d. Inputs: TBD
e. Outputs: TBD

Tool 3. Eclipse IoT Kura: IoT framework for data collection and edge devices integration

a. Run-time tool
b. Connected to the AHF
c. EPP5
d. Inputs: TBD
e. Outputs: TBD

Tool 4. Tool to integrate the Web-of-Things Framework and the Arrowhead Framework: a major software component for integration of web-side applications and Arrowhead Tools Framework

a. Run-time tool
b. Connected to the AHF
c. EPP4, EPP5
d. Inputs: TBD
e. Outputs: TBD

Tool 5. IoT oriented cloud platform: Eclipse IoT Kapua, Java: Cloud backed to interface with Arrowhead Platform for large deployments

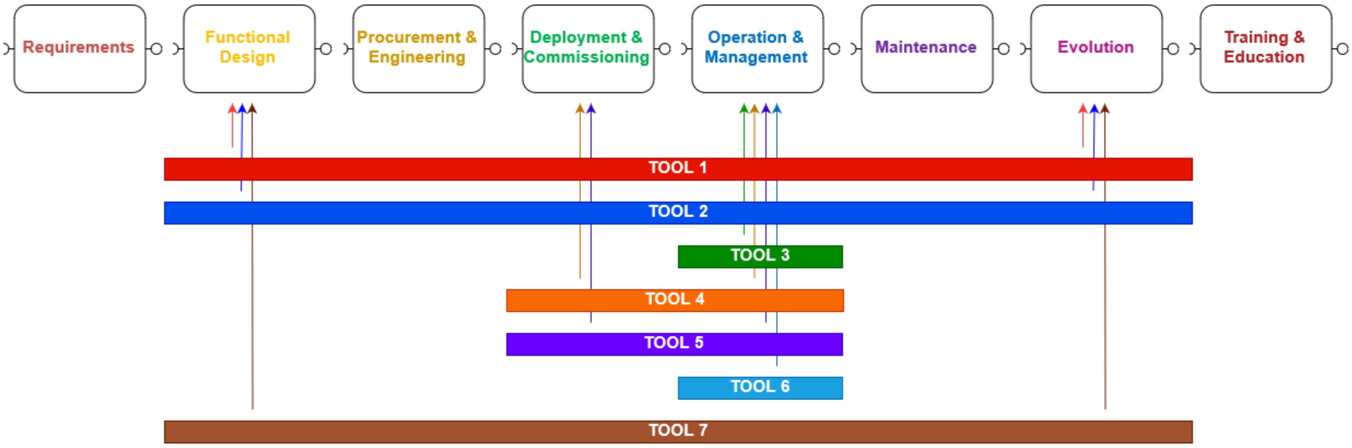
a. Run-time tool
b. Connected to the AHF
c. EPP4, EPP5
d. Inputs: TBD
e. Outputs: TBD

Tool 6. Semantics discovery: SPARQL Event Processing Architecture: Semantic module for the VITAL IoT Sensor interoperability modules – Smart City platform

a. Run-time tool
b. Connected to the AHF
c. EPP5
d. Inputs: TBD
e. Outputs: TBD

Tool 7. Arrowhead Framework Application development kit: Development kit for Smart City and Smart Energy Solutions based on Arrowhead Framework

a. Design-time tool
b. Connected to the AHF
c. EPP2, EPP7
d. Inputs: TBD
e. Outputs: TBD



9. UC-8.2 [ST-I, Eurotech, IUNET, POLITO, REPLY, BEIA, ROP] SoS engineering of IoT edge devices (AI-Driven Environmental Monitoring)

Contact: Davide Brunelli [davide.brunelli@unibo.it]

9.1 Use Case Overview

Intelligent surveillance is one of the most challenging applications of Smart Cities; stimulated by a confluence of simultaneous advances in key disciplines: computer vision, image sensors, embedded computing, energy harvesting, and efficient wireless network of sensors. However, computer vision typically requires notable amounts of computing performance, a considerable memory footprint and high power consumption. Thus, wireless smart cameras pose a challenge to current hardware capabilities in terms of low-power consumption and imaging performance. For this reason, wireless camera systems still require considerable amount of optimization in different areas such as computing architectures, image processing algorithms, power management, energy harvesting and distributed engine. In this use case, we introduce a smart wireless smart camera equipped with neural engine and a Vector processor to boost Artificial Intelligence capability directly on the camera, instead to a server side such normally happens. Moreover, thanks to the low-power consumption, an energy harvester the system will be energy autonomous and will permit to save cost for installation and maintenance.

Despite the current availability of different systems based on neural networks for the recognition of the presence and location of objects in images and videos, their application in real contexts is still an active research field. In particular, porting on embedded devices, reconfigurability, re-training on new classes of objects are still open issues. The solutions available in the literature are different, including those designed specifically for mobile or embedded architectures (e.g. YOLOv3, MobileNet). Regardless of the solution chosen, the same deep architecture can be used for different tasks by updating the set of parameters.

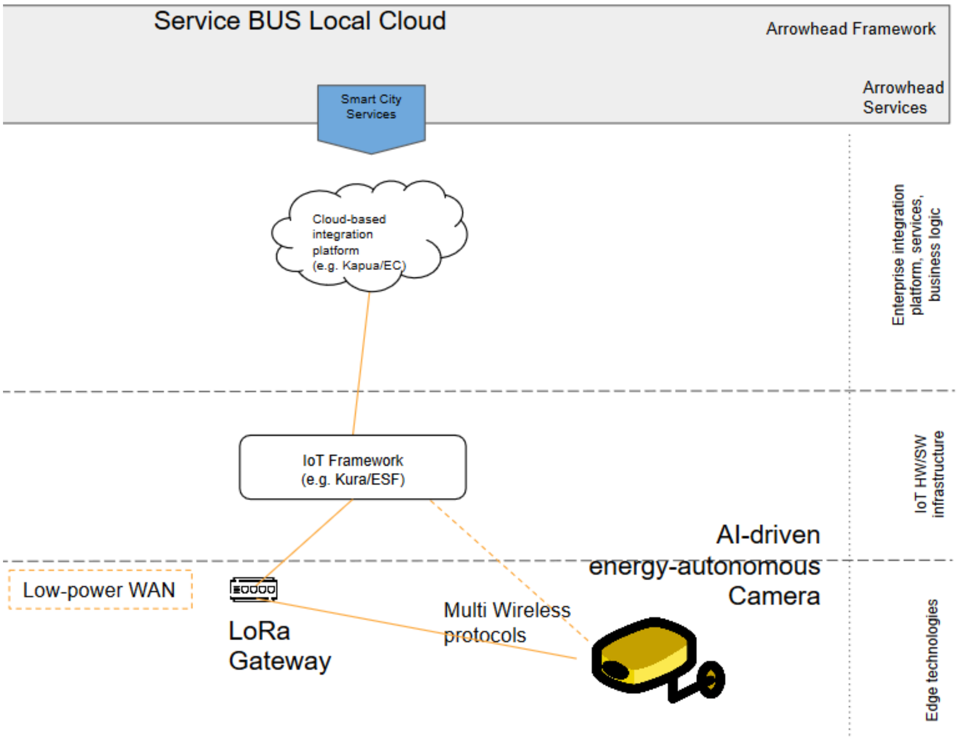
Within the project, the possibility of reconfiguring the parameters of a detection network online will be studied, to dynamically adapt the behavior and therefore the final objective of a system based on a smart camera. As an example, the device will be able to identify pedestrians and people on certain days of the week or times of the day, while during other moments the network can be devoted to detect vehicles and to estimate their respective flows.

The reconfiguration of the device (i.e., by sending the deep network parameters to the device) requires occasional broadband transfers, while the sending of detection results from the device to the users will be done constantly, albeit with less bandwidth requirements.

The reconfiguration capability of the device allows the reduction of installation, maintenance and hardware costs.

In the following the functional blocks of the AI-drive Camera for Environmental monitoring, as described in the figure:

- AI-ZeroEnergy-camera: consisting in a multiradio protocol, a QVGA camera and a RISC-V vector processor;
- Cloud-Based Services: Data are collected and sent in burst mode via Wi-Fi or LoRaWAN on the cloud;



9.2 Identified Gaps

The AI-driven core algorithms are developed for an embedded ultra-low power Vector processors and uses algorithms tested and validated on a server-side tool. The training phase is still on server side (SQL/NoSQL dB), while classification is expected real-time on the meter.

Objective of application developers is to make full use of their processing power. As these systems have multiple cores and different shared resources, the problem of how to balance workload among multiple cores comes when they try to assign tasks to cores. Moreover after the partitioning a huge effort should be carried on during the validation of the partitioned solution both for functionality assessment and for performances.

From the point of view of energy-neutral condition, a tool that helps to design the most suitable energy harvester for the camera, is missing. According to the final deployment environment, a tool for harvesting design is necessary.

9.3 Identified Needed Tools

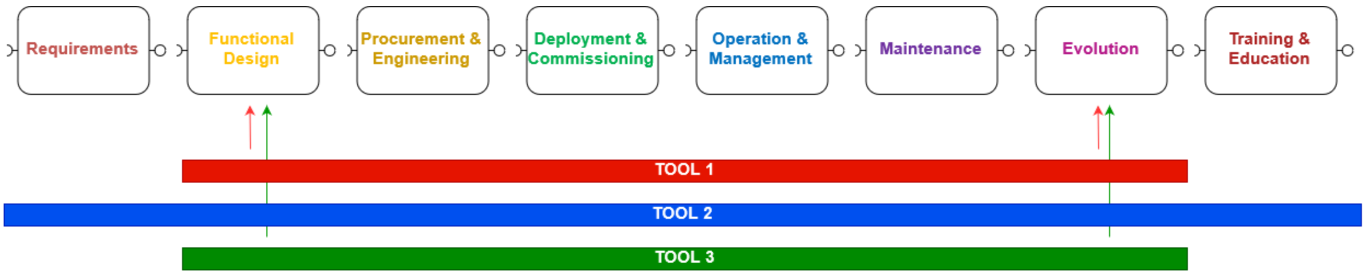
Tool 1. Edge side CV processing enabling kit (task partitioning and allocation)
   a. Design-time tool
   b. Not connected to the AHF
   c. EPP2, EPP7
   d. Inputs: C sequential code code
   e. Outputs: C task partitioned code

Tool 2. Edge side CV processing enabling kit (energy-optimized last-meter IoT sensor nodes at minimum marginal cost)

a. Design-time tool
b. Not connected to the AHF
c. EPP2, EPP7
d. Inputs: Matlab, C/C++
e. Outputs: C/C++/C#

Tool 3. Energy Harvesting design Tool

a. Design-time tool
b. Not connected to the AHF
c. EPP2, EPP7
d. Inputs: electrical parameter
e. Outputs: Energy harvester Configuration detail



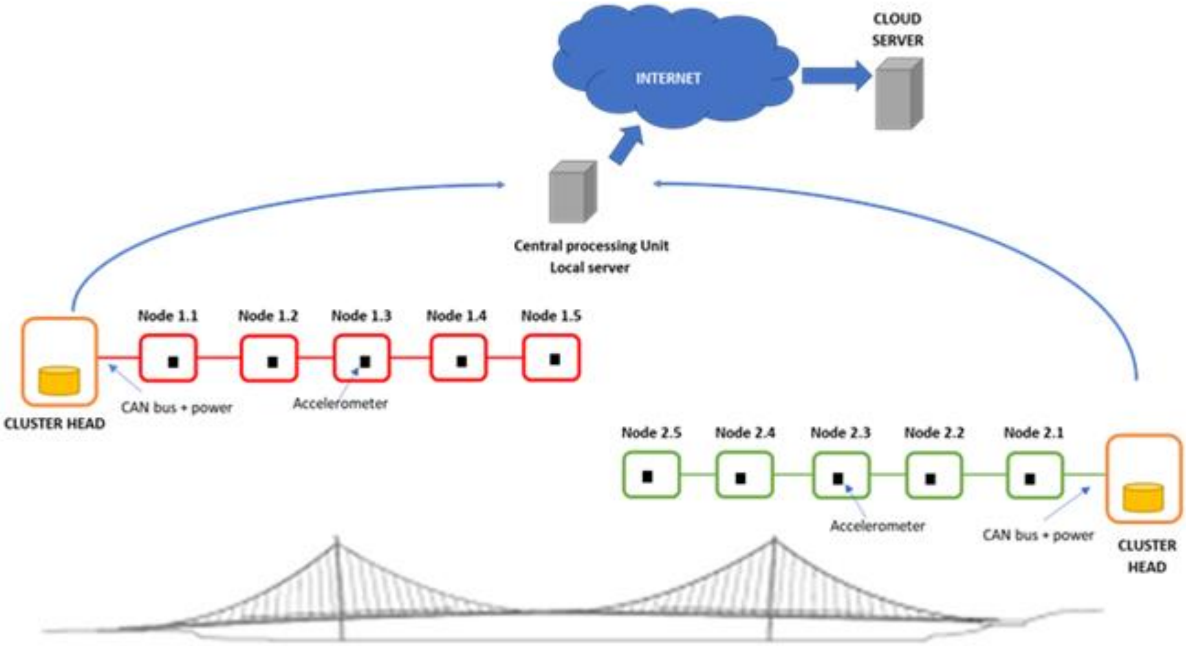
10.UC-8.3 [ST-I, Eurotech, IUNET, POLITO, REPLY, BEIA, ROP] SoS engineering of IoT edge devices (Condition Monitoring)

Contact: Tullio Salmon Cinotti [tullio.salmoncinotti@unibo.it], Riccardo Venanzi [riccardo.venanzi@unibo.it]

10.1Use Case Overview

The introduction of new technologies and services associated with the Internet of Things is revolutionizing many applications in the field of smart cities. In particular, the new devices have to be more efficient, intelligent, aware of context and more connected. In case of condition monitoring applications, they must also be more robust and ensure greater safety for humans and minimize the downtime of the provided service.For example, concerning condition monitoring of public infrastructure, like bridges, the early detection of structural failure is the key element to avoid catastrophic disasters.Condition Monitoring is the process of measuring the status of an entity over time. Entity related condition data are collected to establish trends, recognize anomalies, degradation, failures, failure risk increase and estimate remaining life. Through condition monitoring the maintenance process can be optimized, by improving the trade-off between accepted risk level and maintenance cost through efficient data analysis.One of the most promising application field of Condition Monitoring is the so known Structural Health Monitoring (SHM).One of the limiting factor in the extensive use of Structural Health Monitoring Systems (SHMS) are the initial costs related to the design of the sensing architecture, its installation and the maintenance costs.The nowadays solutions, indeed, are extremely complex and require a big effort in terms of time and costs at design time. Moreover, there is no standardization in the data exchange between different SHMS limiting the possibility to obtain data set usable for the implementation of new smarter, adaptive and fast ML and AI algorithms for an effective monitoring of the structure of interest. Another critical aspect is related to the maintenance costs of a SHMS. A condition-based maintenance approach drives maintenance actions based on the observed condition of the structure of interest. Maintenance is performed before failures and only when necessary. This is the base to an approach looking forward to Predictive Maintenance, where maintenance requirements are predicted in advance. Appropriate and prompt maintenance of the SHMS must be ensured to the critical operations required. The investigation of smart Structural Health Monitoring Systems (SHMS) solutions has important effects from both social and economic point of views and for these reasons it has been chosen as use case to demonstrate the effectiveness of tools developed within the AF to reduce the engineering costs facilitating the extensive introduction on the market of SHMS solutions. From a social point of view, for example, under normal conditions it allows to have updated information about the current health of all the monitored structures within a city or a country enabling the actuation of smart plans to avoid traffic congestions with the consequent increment of the CO_2 and NO_x emissions that are detrimental for the environment. Under an extreme event, such as an earthquake, instead, SHMS are used for rapid condition screening. This screening is intended to provide, in near real-time, reliable information about system performance during such extreme events and the subsequent integrity of the system reducing the downtime of the normal services (e.g. public transports).

Our baseline consists of dedicated sensors and sensor networks, organized in an IoT-inspired architecture, enabling, in principle, the creation of large data sets. The demonstrator is going to be a structural health monitoring system (SHMS) for a target structure (e.g. a section of a bridge or a construction, that could be civil or industrial). The SHMS will consist of multiple IoT networks, including networks of sensors installed on the target structure, a network of environmental monitoring sensors provided by Eurotech, that collects and analyzes air quality parameters, electromagnetic fields, and ionizing radiation (ReliaSens), and additional third parties datasources, such as for example meteo data services.



10.2 Identified Gaps

One of the limiting factor in the extensive use of SHMS are the initial costs related to the design of the sensing architecture, its installation and the maintenance costs. The nowadays solutions, indeed, are extremely complex and require a big effort in terms of time and costs at design time. Moreover, there is no standardization in the data exchange between different SHMS limiting the possibility to obtain data set usable for the implementation of new smarter, adaptive and fast ML and AI algorithms for an effective monitoring of the structure of interest. Another critical aspect is related to the maintenance costs of a SHMS. A condition-based maintenance approach drives maintenance actions based on the observed condition of the structure of interest. Maintenance is performed before failures and only when necessary. This is the base to an approach looking forward to Predictive Maintenance, where maintenance requirements are predicted in advance. Appropriate and prompt maintenance of the SHMS must be ensured to the critical operations required.

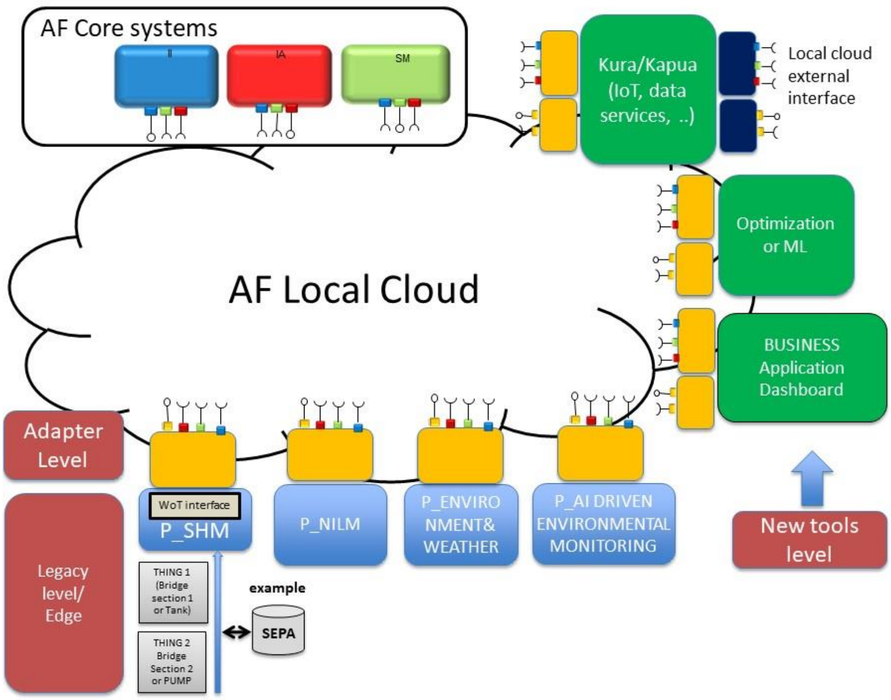
Our baseline might be drastically moved forward with appropriate tools to overcome the costs and technical difficulties during the specification, design, setup, operation and maintenance phases. An appropriate tool chain could also increase the flexibility and the “learning”/”evolution” capability of condition monitoring platforms as well as decrease the training costs of the personnel involved along the life-cycle. Based on current opinions, to achieve this goal a “prerequirement” is the ability to overcome the limit of using only data originated by isolated and dedicated sensors and sensor networks. Heterogeneous context data need to accompany the specific information collected from the monitored asset. A SoS-based solution, capable to orchestrate and manage with a specific IoT platform a heterogeneous set of data sources, represents an effective solution to overcome those limits. In conclusion, we observed the need for optimizing the maintenance process, minimizing operational risk levels and maintenance costs (through condition monitoring).To meet such a requirement we need to:

a) **Collect data** from multi-vendor heterogeneous sensors and sensor networks (i.e. edges and extreme edges) and make the collected data consistent with a shared data model.
b) Create tools to accompany the engineering process of IoT based monitoring platforms that optimize the data collection process and associated costs.
c) Create tools to support prognostics based on condition monitoring.

Our hypothesis is that the AHF should be the enabling technology to address these requirements, and to demonstrate this, we propose a general architecture suitable for condition monitoring both in industrial and smart city scenarios.

10.3 Identified Needed Tools

In this scenario, the data that will be collected by the demonstrator that will be realized in this use case and the tools developed in this use case, compliant with the Arrowhead Framework (AF), aim to provide the means to overcome these issues, simplifying the design process, reducing engineering costs and also the maintenance costs for the end user. The demonstrator will be used to validate a tool-chain prototype to support the engineering process of condition monitoring systems that might consist of a chain of four tools:



Tool 1. A tool to support **functional specification**, having as output the floorplan of the sensing infrastructure, mapped onto the target structure
   a. Design-time tool
   b. Connected to the AHF
   c. EPP2
   d. Inputs: [Application constraints]
   e. Outputs: [Optimal placement of sensors] EP-O2

Tool 2. A tool to support sensor network **configuration design** and power management design, having as output the appropriate mix of sensors and energy harvesting devices, with the appropriate configurations. A simple tool to speed up the design of a complete condition monitoring system for Structural Health Monitoring (SHM) applications will be developed. In particular, the tool will provide the guidelines needed to design an effective Energy Harvesting system starting from a set of inputs depending on the specific application (e.g. identification of the best energy source accordingly with the placement of the sensor node in the real environment where it will operate, application duty cycle, etc…). The tool will also help in the choice of the most suitable components acting as selector guide (e.g. MEMS sensors, PMUTs, including all relevant sensors and modules provided by ST, microcontrollers provided by ST), accordingly with application driven constraints (e.g. output data rate, application duty cycle, computational/processing load needed at sensor node level, communication protocol). Finally, accordingly with the application-related constraints, the tool will estimate the power consumption of each component of the SHM system (e.g. cluster-head) by combining the estimated power consumption of the components chosen among the ones available at edge technology level in the ST portfolio.

a. Design-time tool
b. Connected to the AHF
c. EPP3
d. Inputs: [Application constraints][Functional Design] EP-I3
e. Outputs: [Choice of Sensor models] EP-O3

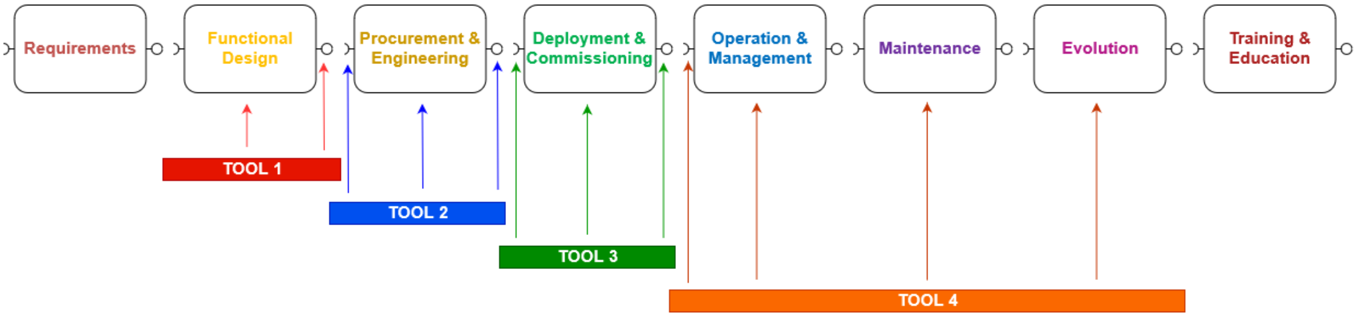
Tool 3. A tool to support the **deployment of the configuration** of the sensing infrastructure according to the design plan. The purpose of this tool is to configure all parameters of the sensors network, as defined at design time by the previous tool of the tool chain.

a. Design-time tool
b. Connected to the AHF
c. EPP4
d. Inputs: [Sensor specification] EP-I4
e. Outputs: [Sensor configuration] EP-O4

Tool 4. A tool to support the **operation and evolution** of the sensing infrastructure by providing data to context sensitive optimization tools possibly provided by third parties. It is expected to deal with other third party sensor platforms that address additional environmental conditions (such as weather, pollution, or other context related aspects, such as traffic in transport applications) are also integrated in the AF. Furthermore it will act as an optimization tool, with context dependent business logic which interacts with the sensor platform to optimize, *operation, maintenance and evolution* of the addressed plant

a. Run-time tool
b. Connected to the AHF
c. EPP5, EPP6, EPP7
d. Inputs: [External Sensor Data][Sensor configuration] EP-I5
e. Outputs: [Data Analysis][Sensor Scheduling]

The tools need to be made interoperable thanks to a shared data model understood by all tools. Such a data model has to describe all entities involved in the engineering process and the associated properties. Most likely it will rely on standard ontologies, such as SAREF (to model devices) and SOSA (to model sensors and actuators).


Requirements
Functional Design
Procurement & Engineering
Deployment & Commissioning
Operation & Management
Maintenance
Evolution
Training & Education
TOOL 1
TOOL 2
TOOL 3
TOOL 4

11. UC-8.4 [ST-I, Eurotech, IUNET, POLITO, REPLY, BEIA, ROP] SoS engineering of IoT edge devices (Smart Home)

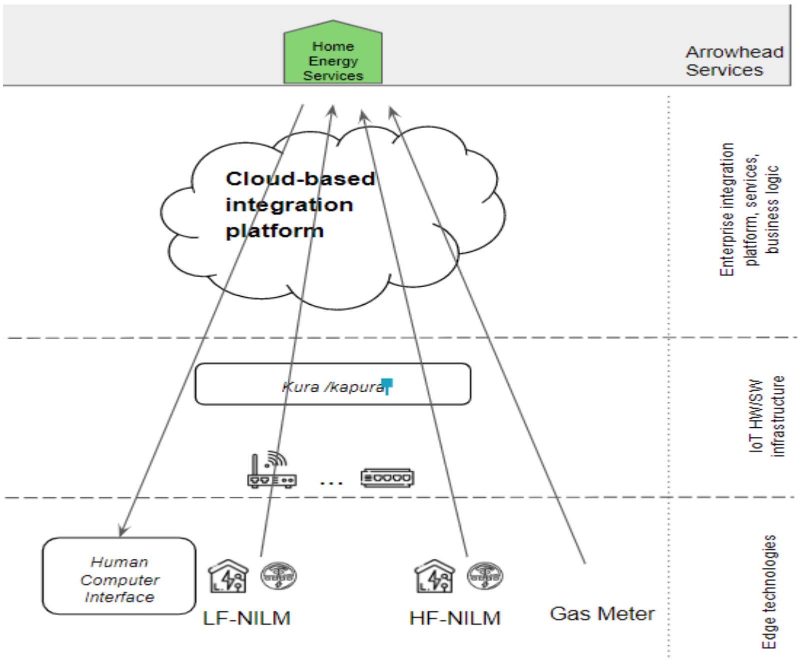
Contact: Davide Brunelli [davide.brunelli@unibo.it], Edoardo Patti [edoardo.patti@polito.it], Gianvito Urgese [gianvito.urgese@polito.it], Sara Bocchio [sara.bocchio@st.com]

11.1 Use Case Overview

This use case is applied to whole development lifecycle of an IoT and SoS for a smart energy smart home application.
The UC-08_04 will provide the ***Edge side processing enabling kit*** composed by two components. In the following a short description of each component.

- **Last-meter IoT sensor nodes at minimum marginal cost** group three sensing services:
    - **Low-Frequency NILM (Polito)** The home NILM software is a cloud application of a technique for load pattern recognition starting from the measurement of the aggregated home energy, taken by a custom IoT hardware. Data measured by the hardware are sent via MQTT to a cloud infrastructure and the software is scheduled every day to analyze 24 hours of data. The results of the analysis are stored in a database and called by a mobile application.
    - **High-Frequency NILM (UNIBO)** The home high-frequency NILM software executes the task of disaggregation power consumption of single appliances from an agglomerated mains power measurement. From the machine learning point of view, this is considered a single-channel blind source separation problem, where multiple sources need to be extracted from one combined measurement.
    - **Gas Meter**
      NB-IoT is a leading long-range wireless communication technology based on cellular networks. NB-IoT features security, wide coverage, low power consumption, massive connectivity, and low cost. In the smart gas field, NB-IoT enables stable, real-time traffic data collection from gas meters, device monitoring, command delivery, and additional remote operations. After the gas meter data and status information is collected in a secure and low-cost manner, it can be analyzed and handled promptly to implement targeted, scientific, and dynamic management. This improves the management efficiency and service satisfaction of natural gas utilities.

- **Task partitioning and allocation**, will be developed for partitioning and mapping a data stream application on multi-core edge devices. It will analyse the application and automatize the identification of tasks and subtask that will be mapped on the available resources by following objective functions imposed by the developers (optimise for power, or for performances).



The second component, called *task partitioning and allocation*, will be applied to the electric smart meter attached to LF-NILM.

11.2 Identified Gaps

All the previous phases have tools helping the development or user. Most of the NILM approaches are off-line and execute on server side. Task partitioning and mapping on multicore architecture is performed by hand, making the procedure slow and error-prone. More in detail, here are the current baseline technologies used so far:

Edge side processing enabling kit (last-meter IoT sensor nodes at minimum marginal cost)

- **Low-Frequency NILM (Polito)**
  The NILM core algorithms are developed in Matlab and then embedded in a C# project, linked to a SQL DB in an Azure cloud infrastructure. The mobile application (Android/iOS) are developed using native technologies

- **High-Frequency NILM (UNIBO)**
  The NILM core algorithms are developed for an embedded ultra-low power Vector processors and uses algorithms tested and validated on a Matlab project. The training phase is still on server side (SQL/NoSQL dB), while classification is expected real-time on the meter.

- **Gas meter**

  The adoption of smart gas meters in Europe is picking up speed. This year alone, the installed base of remotely read gas meters for consumers nearly doubled to 17.3 million units in EU28+2. The growth is mainly driven by nationwide rollouts in France, Italy, the Netherlands and the UK. A variety of communication technologies are in use for the rollouts, including fixed RF networks, mobile networks and local interfaces to master electricity meters. As next generation IoT-optimized 4G/5G mobile technologies become more widely available, it is expected that there will be a shift from legacy RF platforms to emerging standards like NB-IoT.

Edge side processing enabling kit (task partitioning and allocation)

Objective of application developers is to make full use of their processing power. As these systems have multiple cores and different shared resources, the problem of how to balance workload among multiple cores comes when they try to assign tasks to cores. Since the workload of system is determined not only by applications that run on processing cores, but also of data network traffic. It is not obvious how to reasonably assign tasks to each core so that there will not be any bottleneck that might compromise the performance of the system. Moreover after the partitioning a huge effort should be carried on during the validation of the partitioned solution both for functionality assessment and for performances

11.3 Identified Needed Tools

Tool 1. Edge side processing enabling kit (task partitioning and allocation)
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2, EPP7
  d. Inputs: c sequential code code
  e. Outputs: c task partitioned code

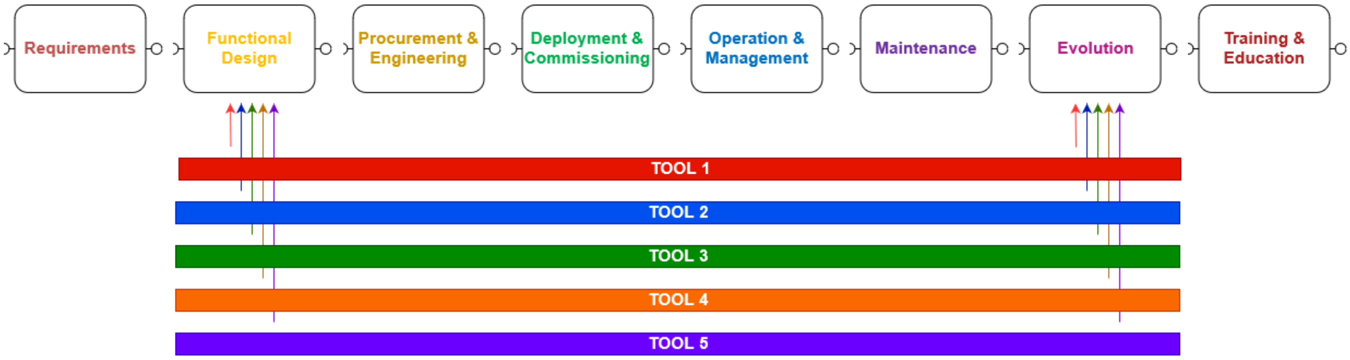
Tool 2. Edge side processing enabling kit (energy-optimized last-meter IoT sensor nodes at minimum marginal cost): HF-NILM
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2, EPP7
  d. Inputs: Matlab
  e. Outputs: C/C++/C#

Tool 3. Edge side processing enabling kit (energy-optimized last-meter IoT sensor nodes at minimum marginal cost): LF-NILM
  a. Design-time tool
  b. Connected to the AHF
  c. EPP2, EPP7
  d. Inputs: Matlab
  e. Outputs: C/C++/C#

Tool 4. Edge side processing enabling kit (energy-optimized last-meter IoT sensor nodes at minimum marginal cost): Gas Meter
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2, EPP7
  d. Inputs: Matlab
  e. Outputs: C/C++/C#

Tool 5. Human-Computer Interaction module (Mobile application).
  a. Run-time tool
  b. Connected to the AHF
  c. EPP2, EPP7
  d. Inputs: [NILM Measurements]

e. Outputs: [Usage Stats][Commands to the NILM]



12. UC-8.5 [ST-I, Eurotech, IUNET, POLITO, REPLY, BEIA, ROP] SoS engineering of IoT edge devices (Industrial Energy Monitoring)

Contact: Antonio Lionetto [antonio.lionetto@st.com]

Use Case Overview

The world of manufacturing is changing. Call it Industry 4.0, the 4th industrial revolution, or the Industrial Internet of Things (IIoT), it's all about doing things just in time, concurrently, more efficiently, with greater flexibility, and in a safer and more environmentally friendly manner. The introduction of new technologies and services associated with the Internet of Things is revolutionizing many industrial applications. Initiatives such as those in factories regarding automation and industrial predictive maintenance, and initiatives to build smarter working environments are creating opportunities for new entrants and traditional players to offer innovative solutions that change business models. To ensure the high level of automation required in today's industrial applications, equipment must be more efficient, intelligent, aware of context and more connected; it must also be more robust and ensure greater safety for humans interacting with them. Anyone running a factory wants to keep it running at optimal speed with minimal downtime. They are also aware that any machine with moving parts suffers wear and tear and inevitably requires servicing and repair.
One approach is to simply schedule maintenance tasks at fixed intervals, regardless of the actual condition of the equipment. This is simple to plan, but the maintenance may not occur in time to prevent equipment damage and dangerous situations, or it may be carried out when it is not necessary. A smarter approach is Condition-based Maintenance. This approach drives maintenance actions based on the observed condition of the machine. Maintenance is therefore performed before failures and only when necessary. The drawback is that not implementing maintenance until a machine shows signs of failure is often risky and may interrupt production runs at highly inconvenient times. An even better approach is Predictive Maintenance, where maintenance requirements are predicted well in advance. Predictive Maintenance combines condition monitoring with a dynamic predictive model for failure modes. This approach promises maximum protection of machinery and minimum productivity impact, without necessarily increasing overall system complexity.

In this project Condition Monitoring applications in an end to end framework will be exploited by implementing a use cases strictly correlated with smart sensor vibration and ultrasound nodes. A new cloud-based tool will be developed and adopted for Industry 4.0 combining the information coming from the physical entities object of monitor and surveillance on their operative status.

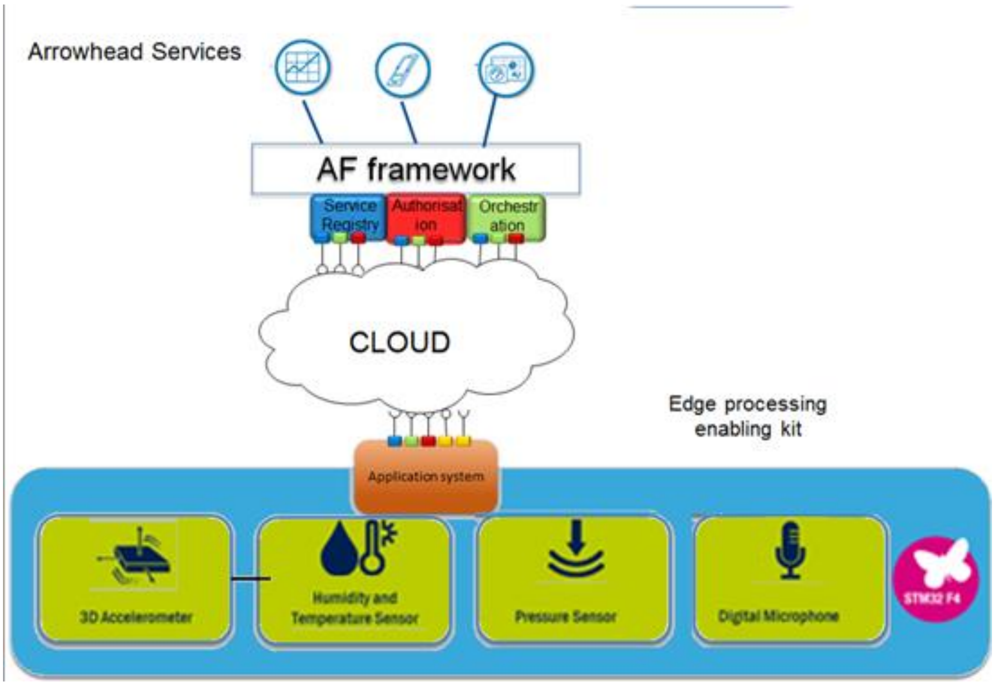
Identified Gaps

Condition monitoring and anomaly detection is the process to monitor relevant physical characterizes such as vibration, noise, temperature, power adsorption of an equipment and part of this to recognize the trend versus an anomalous behavior. Advanced monitoring is based on the appropriate set of sensors in which sensor fusion capability reinforce the status detection and providing the required data collection enable their analysis through the different vertical layer providing advanced services on base of this information and elaboration chain.
This industrial scenario consists of:

- Smart sensor nodes deployed at the critical machines

- Collection of sensors data and preprocessing with / without machine learning at the edge (either the node or a gateway / industrial PC)
- High level analytics on company premises or on cloud for provisioning the nodes with updated algorithms

Today the applications enabled by this approach based on smart sensor vibration and ultrasound nodes are growing providing an advanced monitoring on production line and to ensure a condition maintenance for critical machineries avoiding downtime; the exploitation of new machine learning techniques are under development to facilitate and accelerate introduction in the field of advanced features as the predictive maintenance that allows the optimization of maintenance plan for the production line.



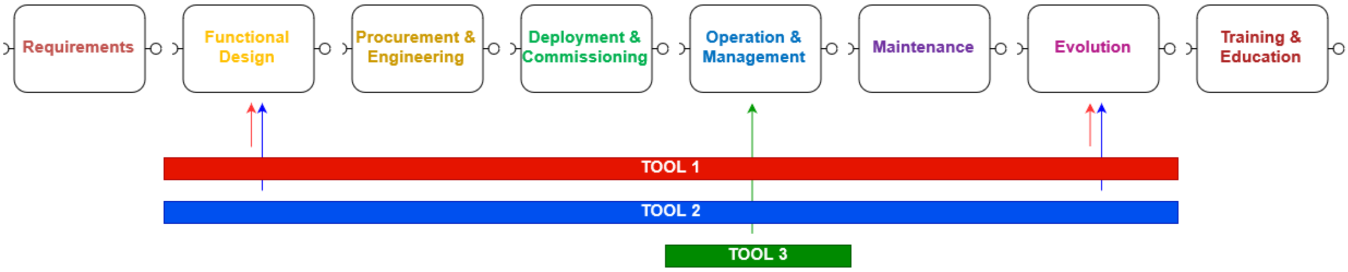
Identified Needed Tools

Tool 1. Autonomous sensing enabling kit: Design and integration of smart sensors and smart meters
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2, EPP7
  d. Inputs: TBD
  e. Outputs: TBD

Tool 2. Edge side processing enabling kit: Design, configuration and operation of Edge Devices
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP2, EPP7
  d. Inputs: TBD
  e. Outputs: TBD

Tool 3. programming and debugging tool equipped with firmware for algorithms on advanced time and frequency domain signal processing and analysis of the 3D digital accelerometer

a. Run-time tool
b. Connected to the AHF
c. EPP5
d. Inputs: TBD
e. Outputs: TBD



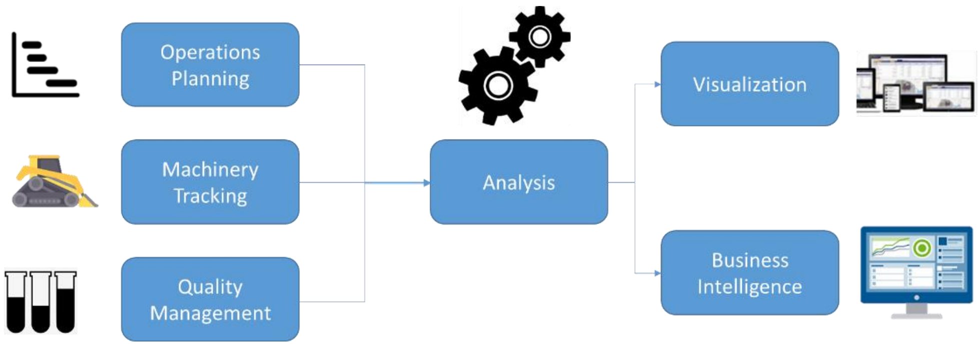
13. UC-9 [Acciona, dotGIS, AITIA, BME]: Machine operation optimisation

Contact: José Luis Burón [joseluis.buron.martinez@acciona.com]

13.1 Use Case Overview

The use case is focused on the development of a digital platform for remote monitoring and optimization of earthworks in construction projects. Therefore, the production environment targeted for digitalization is a construction site where heavy machinery of different types (e.g. trucks, excavators, etc.) perform excavations in certain areas of the construction site, transport the excavated materials, and either dump the materials in a landfill, or use them as filler in other areas of the construction site.

The main objective is to track the operations of the machinery in real time in order to monitor the progress of the earthworks, detect deviations from the original plan, analyze productivity and key performance indicators, and detect opportunities for optimization of the operations.



13.2 Identified Gaps

It would be interesting to explore ways of automating the interaction between the different phases of the UC-EP, as the current interactions are rather manual.

13.3 Identified Needed Tools

Tool 1. Tools for automating the link between product backlog and tasks defined in the Functional Design phase

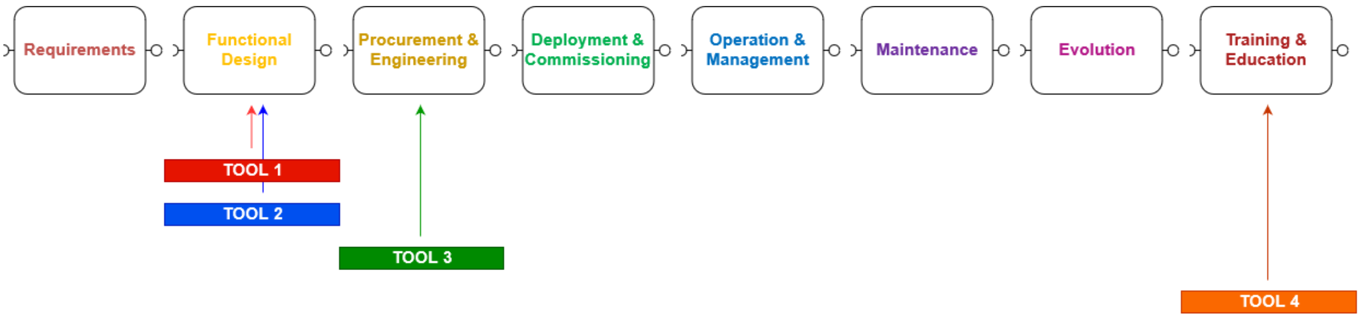
a. Design-time Tool
b. Not connected to the AHF
c. EPP2
d. Inputs: TBD
e. Outputs: TBD

Tool 2. Tools for modelling the platform architecture and for the management of the specifications of the different platform modules. Currently, separate specification documents (Word, PowerPoint and/or Excel files) are created for each module and stored in a folder of the repository of Microsoft teams. Therefore, what is lacking is a tool for a structured management of the platform architecture and of the specifications of the different modules

a. Design-time tool
b. Not connected to the AHF
c. EPP2
d. Inputs: TBD
e. Outputs: TBD

Tool 3. Tools for automating the Quality Control stage in the Engineering Phase, facilitating the definition and management of tests for verifying the software modules produced in the Development stage of the Engineering Phase
a. Design-time Tool
b. Not connected to the AHF
c. EPP3
d. Inputs: TBD
e. Outputs: TBD

Tool 4. Tools for automating the Documentation stage of the Engineering Phase, especially for software modules
a. Design-time Tool
b. Not connected to the AHF
c. EPP8
d. Inputs: TBD
e. Outputs: TBD

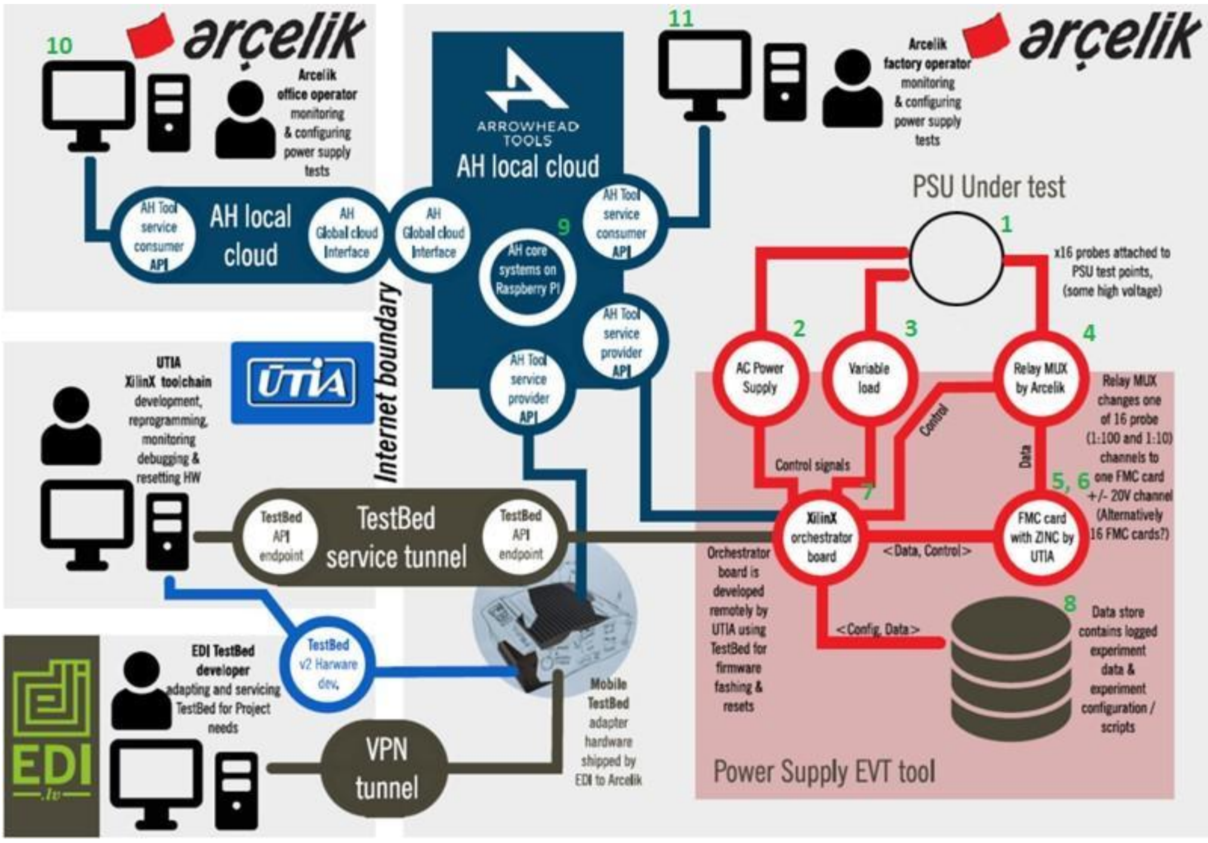


14. UC-10 [ARCELIK, EDI, UTIA]: Rapid HW development, prototyping, testing and evaluation

Contact: Mustafa Küçükkuru [mustafa.kucukkuru@arcelik.com], Alper Özel [alper.ozel@arcelik.com], Çağlar Henden [caglar.henden@arcelik.com]

14.1 Use Case Overview

The use case is focused on the development of a data acquisition platform for remote monitoring of power supply boards (PSU Under Test) in terms of reliability and functional performance. The environment targeted for digitalization is a design&approval laboratories and electronic board manufacturing sites. The main objective is to track voltage stresses on critical electronic components/voltage responses on functional nodes of power supply boards in several test conditions that can be defined by an administrator. The test conditions will be based on voltage input and load condition parameters.Together with EDI TestBed, Power Supply EVT tool will have the feature to update&debug firmware of via VPN tunnel.



14.2 Identified Gaps

The Xilinx Vivado HLS toolchain is Eclipse based and integrates HW design flow and SW design flow for fixed selection of Zynq Ultrascale+ devices present on university boards: ZCU102, ZCU104 and ZCU106. The existing design flow does not provide unified configuration and project-bringup-scripts to address industrial modules fitted with variable sizes of Zynq Ultrascale devices in design phase. The existing design flow does not manage different platform configuration like

(FMC data acquisition board A or board B) with different requirements for the HW IP interfaces. UTIA will provide and unified set of configuration and project-bringup-scripts to address these variations for the Zynq Ultrascale devices on industrial modules.

At the existing level of implementation EDI TestBed does not support any FPGA based systems, we are considering adding the support later.

14.3 Identified Needed Tools

Tool 1. Xilinx Vivado HLS toolchain with project-bringup-scripts: simplifies the process of selecting an optimal industrial device for the use case. It also features a Testbed by EDI that allows for fast prototyping and reduced costs.

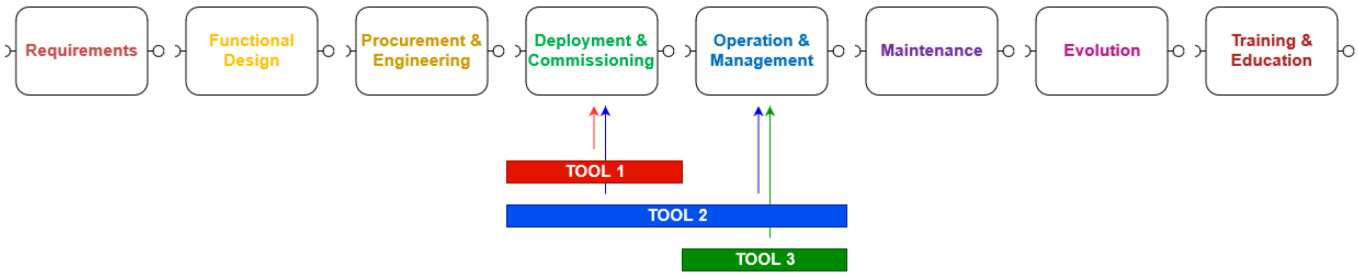
a. Design-time tool
b. Not connected to the AHF
c. EPP4
d. Inputs: environment conditions, device setup
e. Outputs: optimal configuration

Tool 2. Front-end tool: Visualization of data from DAQ (Data Acquisition) system through an operator UI, which can also give access to the test configuration phase.

a. Run-time tool
b. Connected to the AHF
c. EPP4, EPP5
d. Inputs: environment conditions, device data
e. Outputs: visual information

Tool 3. Local cloud services : As repository for collected data from DAQ system. In such case an adapter for the AHF is needed in order to use the storage that is connected to the core services.

a. Run-time tool
b. Connected to the AHF
c. EPP5
d. Inputs: sensed data
e. Outputs: data in the storage



15. UC-11 [DAC, GUT]: Configuration tool for autonomous provisioning of local clouds

Contact: Marek Tatara [marek.tatara@dac.digital]

15.1 Use Case Overview

The aim of the use case is to show the applicability of part of Arrowhead Tools to deploy and manage local cloud in logistics. The use case will be focused mainly on the Authorization and Authentication system in connection with the on-boarding process of a new device. For the purpose of the use case, we assume that the complete, base model of a system is known and properly engineered.

15.2 Identified Gaps

Currently, the framework cannot be deployed automatically to become a local cloud, therefore the dockerized version of it should be developed.

There is a lack of low-level implementation of end-device consumer or provider services, and thus, sensors must be connected to a device with high processing capabilities to become a part of an Arrowhead local cloud.

There is no tool aiding the onboarding process of a new AH-compliant device. Therefore, UI that guides the user through the configuration process is required.

15.3 Identified Needed Tools

Tool 1. Onboarding tool – UI that guides users through the configuration of a new AH device that will be connected to a local cloud
   a. Design-time tool
   b. Not connected to the AHF
   c. EP-I3, EPP3, EP-O3, EP-I7, EPP7, EP-O7
   d. Inputs: configuration parameters, status of a local cloud
   e. Outputs: authorization credentials for the new device

Tool 2. Performance assessment tool – used for evaluating current parameters and efficiency of a local cloud
   a. Run-time tool
   b. Connected to the AHF
   c. EPP5, EPP6
   d. Inputs: performance data
   e. Outputs: processed performance data

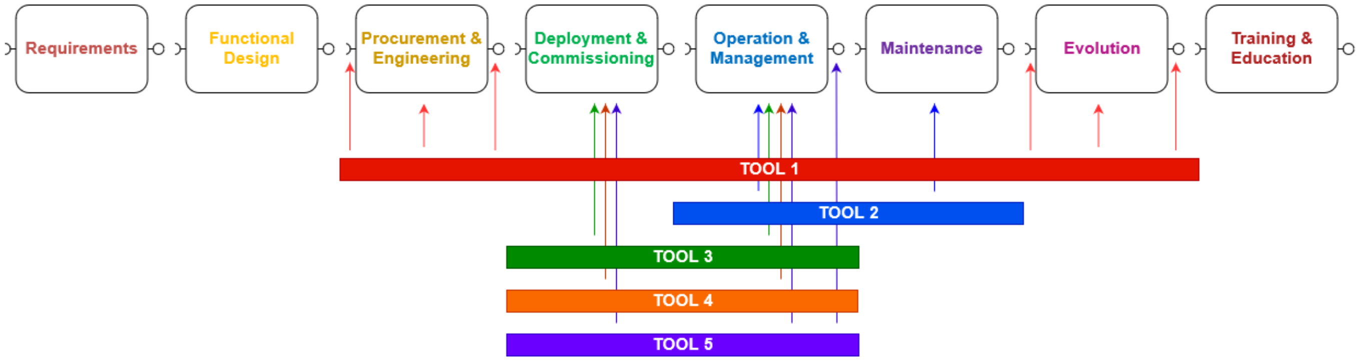
Tool 3. Dockerized version of the core services – used for automatic management of boot and control over the stability of the AH core services
   a. Design-time tool
   b. Not connected to the AHF
   c. EPP4, EPP5
   d. Inputs: configuration of a local cloud
   e. Outputs: commissioning status

Tool 4. Arrowhead compliant small-footprint producer node - suited to work with the developed onboarding process

a. Run-time tool
b. Connected to the AHF
c. EPP4, EPP5
d. Inputs: none
e. Outputs: connection status, payload

Tool 5. Arrowhead Local Cloud wireless localization tool - used to provide the spatial information about AH nodes, which might be used to enhance the wireless security

a. Run-time tool
b. Connected to the AHF
c. EPP4, EPP5, EP-O5
d. Inputs: none
e. Outputs: List of AH nodes with their spatial information

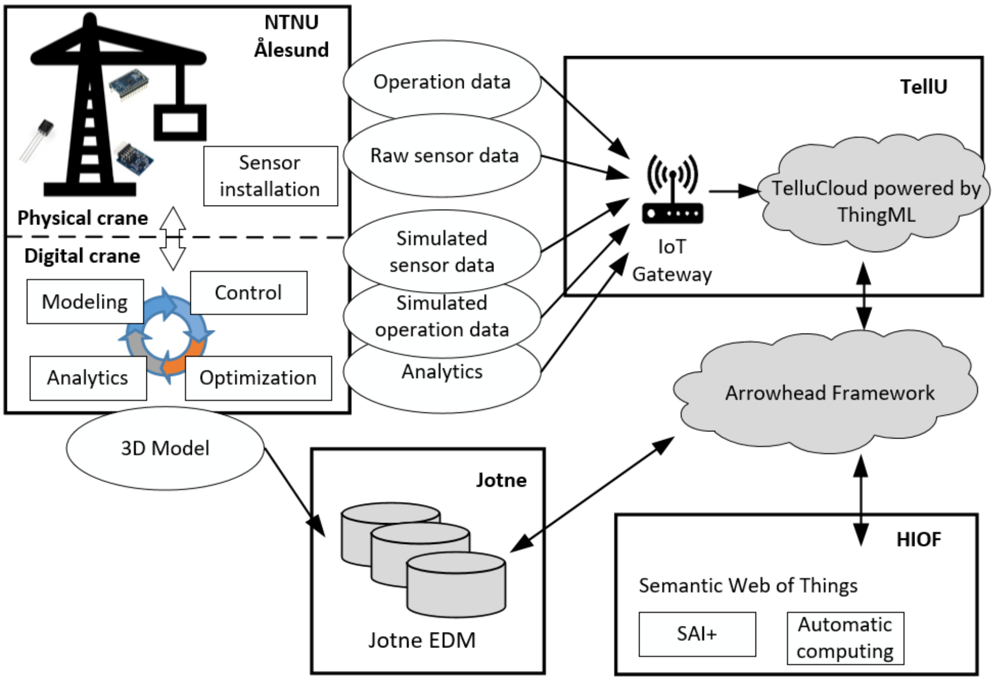


16. UC-12 [SAP, Jotne, Tellu, NTNU, HIOF]: Digital Twins and structural monitoring

Contact: Kjell Bengtsson [Kjell.Bengtsson@jotne.com]

16.1 Use Case Overview

The crane use case will demonstrate how both safety and reliability can be improved by the use of Digital Twins for monitoring of structural integrity. This is especially important in populated areas like construction sites, offshore platforms and marine supply boats. There is a trend to use heavier cranes and smaller boats in offshore applications that finally will compromise safety. A few physical sensors can be augmented by an unlimited number of virtual sensors that can continuously monitor the structural integrity in all situations. The Selected Digital Twin use case is the SAP Crane. This crane is under construction and will be used by 2 NTNU PhD students to develop and benchmark cable & pulley models and methods for obstacle avoidance. SAP has therefore all structural and component information needed to develop and benchmark digital twin solutions. The Norwegian use case will therefore use a Digital Twin (DT) to monitor the structural integrity and performance of the SAP crane. This crane is designed as an offshore knuckle-boom crane with all features to develop and benchmark new DT technologies.



16.2 Identified Gaps

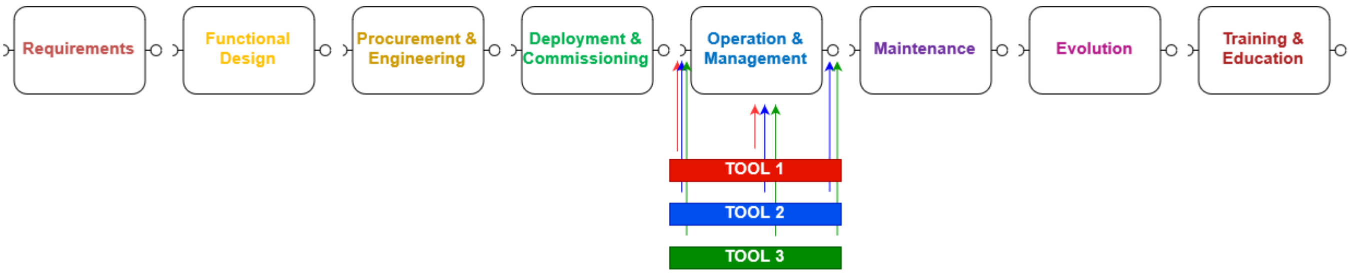
The Use Case Key Performance Indicators (KPIs) are: 1. KPI-1; The static and dynamic loads must be predicted with 20% accuracy. 2. KPI-2; Optimal sensor distribution for load observation must be identified to save 50% in hardware costs. 3. KPI-3; Sensor Drifting must be 90% reduced. 4. KPI-4; Predictive Maintenance must capture 80% of all structural failure modes. 5. KPI-5; A generic cloud based framework for structural monitoring shall send alerts before severe accidents occur.

16.3 Identified Needed Tools

Tool 1. 3D visualization and playback system, plans to visualize the operation using operation data and sensor data. This tool will be developed together with Jotne.
  a. Run-time tool
  b. Possible for being Connected to the AHF
  c. EP-I5, EPP5
  d. Inputs: [virtual sensor and operation data]
  e. Outputs: [visual representation of the crane]

Tool 2. Remote access and control tool, it will be a tool run together with TellU cloud. It will transmit data and control commands from remote end.
  a. Run-time tool
  b. Connected to the AHF
  c. EP-I5, EPP5, EP-O5
  d. Inputs: [request of control command of the crane] [query of sensor information]
  e. Outputs: [control command confirmation] [sensor data with timestamp]

Tool 3. Data analysis tool
  a. Run-time tool
  b. Connected to the AHF
  c. EP-I5, EPP5, EP-O5
  d. Inputs: [sensor data] [operation data] [crane configuration]
  e. Outputs: [statistical results including forces, velocity and displacement] [risk evaluation result in the operation]



17. UC-13 [Boliden, LTU, BnearIT]: Deployment engine for production related sensor data

Contact: Markus Frank [markus.frank@boliden.com]

17.1 Use Case Overview

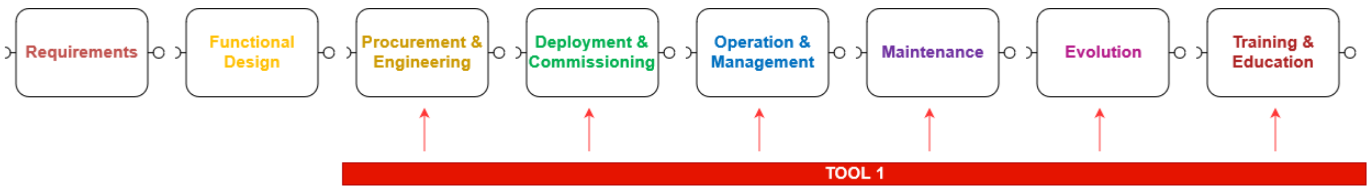
The legacy way of working for production based integrations has been handled with same integration technology as traditional integration such as from and to Financial systems. There are also exceptions with individual solutions based on closed vendor solutions. With the use case of building up an SOA based integration platform the basis should be laid to significantly reduce complexity and speed to implementation while cater for requirements for production critical integrations. Based on the urgency to start building a platform, Boliden has decided to move forward with a proprietary approach and it is planned to look together with LTU and BnearIT at benchmarking, mirror/enhance functionality with the Arrowhead Framework.

17.2 Identified Gaps

Currently the integration approach is not expected to keep up with integration needs in terms of volume, real-time requirements and speed from plan to deployment. The idea is to set up a separate integration platform for production related data.

17.3 Identified Needed Tools

Tool 1. Simple Integration Box (SIB)
a. Run-time tool
b. Not connected to the AHF, discussion ongoing to extend / benchmark with AH together with LTU
c. EPP4, EPP5, EPP6, EPP7, EPP8
d. Inputs: [production data from various sources]
e. Outputs: [production data to various destination systems]



18. UC-14 [IFAG, IFAT]: Smart Diagnostic Environment for Contactless Module Testers

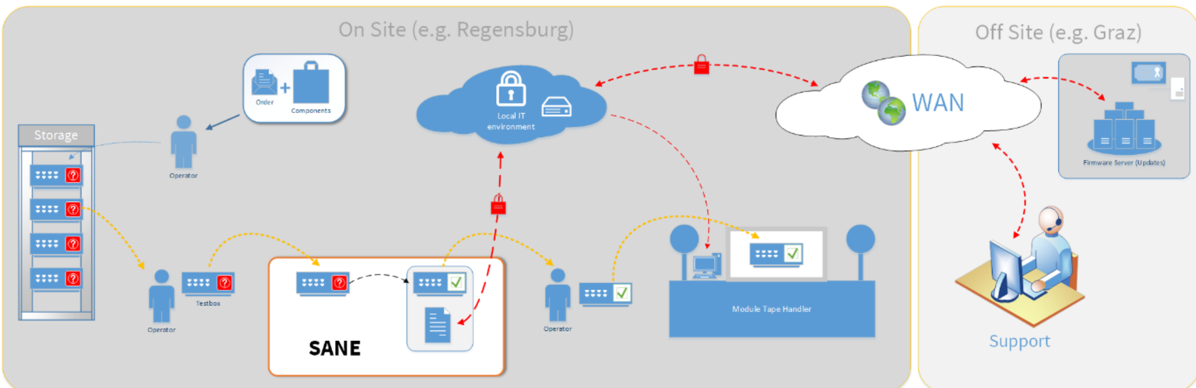
Contact: Christian Hanser [christian.hanser@infineon.com], Georg Skacel [georg.skacel@infineon.com]

18.1 Use Case Overview

The Smart Diagnostic Environment for Contactless Module Testers use-case is based on an existing stand-alone diagnosis tool - the so-called Fully Automated Diagnostic Environment for ISO Module Test (FADE) tool. It allows to calibrate as well as to diagnose the state of test boxes and of readers. These checks are necessary before installing new test boxes and in case of test box failures.
This improved approach for ISO-module test box diagnosis and maintenance will make it possible to further optimize contactless chipcard controller backend manufacturing process. It will help reducing downtimes and help in further improving the adherence to scheduled delivery dates. In close cooperation between IFAT engineers in Graz and IFAG test engineers in Regensburg, the successful development will be verified by installing and running the tool at the IFAG backend manufacturing site in Regensburg.

18.2 Identified Gaps

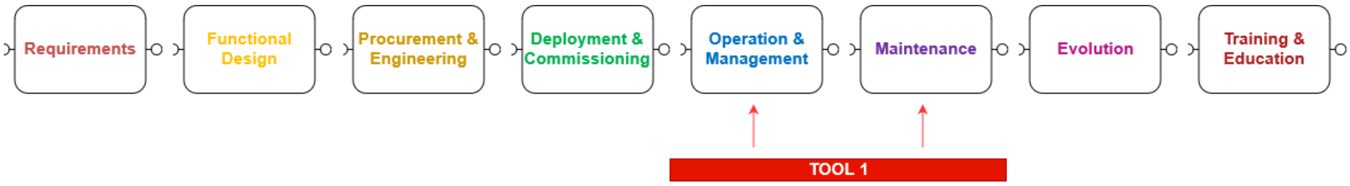
Currently, however, the FADE tool lacks support for remote diagnosis and remote maintenance. Thus, right now on-site expert support is needed every time an error occurs. Moreover, downtimes to search failures as well as additional measurement cycles (as the same tape must be re-measured after a repair) result in high costs. So does regular maintenance - mainly due to maintenance contracts and high shipping costs, as servicing is currently only done at the IFAT premises in Graz. The new SANE tool will improve upon the FADE tool by building a smart remote environment for the accurate, step-wise smart diagnosis and maintenance of ISO-module test boxes.



18.3 Identified Needed Tools

Tool 1. **SANE** tool, which is apparently devoted to a number of things: firmware update, remote support, secure communication etc..
   a. Run-time tool
   b. Not connected to the AHF

c. Operation & Management, Maintenance EPP5, EPP6
d. Inputs: sensor data from the boxes
e. Outputs: reports of failures and accurate diagnosis



19. UC-15 [VTC, LTU, BnearIT] Virtual Commissioning of a Cyber-Physical System for increased flexibility

Contact: Richard Hedman [richard.hedman@volvo.com]

19.1 Use Case Overview

Kitting is to ensure material is delivered and presented to the operator in an optimal way. It gives the opportunity of optimizing material flow and to balance the workload efficiently and effectively. With the introduction of electromobility and automation, our assembly lines and material flow will be even more complex than today. The work of this use case consists in developing concepts for smart and automated kitting operations.

19.2 Identified Gaps

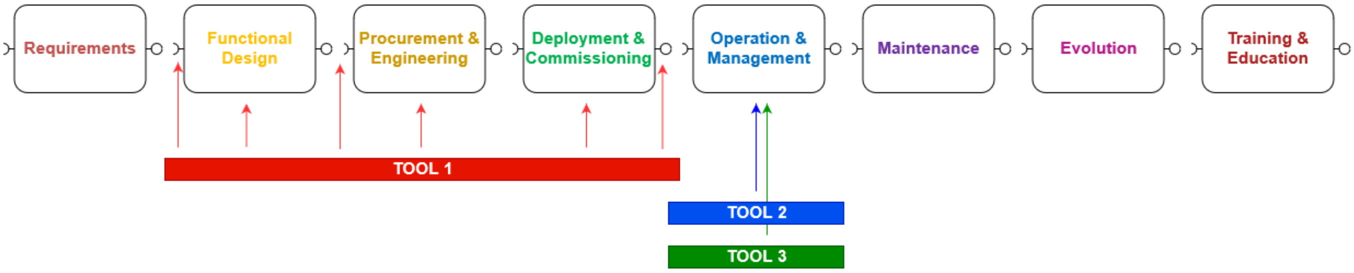
Today, the automation in the kitting processes mainly concern operator support, i.e. pick to light. There is little automation in terms of process planning and monitoring and control of kitting operations. Also, there is a low level of interoperability between concerned systems, such as MES and the logistics planning systems. Initially, the use case will focus on the engineering process related process preparation for kitting. It involves actions of the roles of Logistics Engineer and Production Engineer, aiming for virtual preparation of components, tools and work instructions.

19.3 Identified Needed Tools

Tool 1. MPMLink (Windchill) – used for adaptive control over the kitting wagons and
  a. Design-time tool
  b. Connected to the AHF
  c. EPP2, EPP3, EPP4
  d. Inputs: DCN [EP-I2, EP-I3]
  e. Outputs: [definitions what and how to assemble] [EP-O4]

Tool 2. Central Planning Office - used to define products that have to be assembled
  a. Run-time tool
  b. Connected to the AHF
  c. EPP5
  d. Inputs: -
  e. Outputs: [order specification]

Tool 3. Manufacturing Execution System - used for supervision of the manufacturing process progress
  a. Run-time tool
  b. Connected to the AHF
  c. EPP5
  d. Inputs: -
  e. Outputs: [order progress]


Requirements
Functional Design
Procurement & Engineering
Deployment & Commissioning
Operation & Management
Maintenance
Evolution
Training & Education
TOOL 1
TOOL 2
TOOL 3

20. UC-16 [IFAT, IFD, IFAG, TUD, ifak, KTW, UZL, EXPLEO, SEMANTIS, Bosch, Bosch SI, HTW, EDMS, UzL, ViF]: Production Support, Energy Efficiency, Task Management, Data Analytics and Smart Maintenance

Contact: Germar Schneider [germar.schneider@infineon.com]

20.1 Use Case Overview

The use case addresses important sensor issues in the semiconductor industry with the aim to monitor and control processes within the semiconductor production with the help of different sensors. With the aid of sensors and a simple sensor integration in the future, the semiconductor production will be continuously improved. In addition to production processes, the focus is also on energy efficiency. Sensors won't be only used in the semiconductor front-end production, but also in facility areas such as special vibration sensors that support improving maintenance processes. Another major focus is data processing using predictive maintenance approaches and big data analytics to establish new production and maintenance strategies.

20.2 Identified Gaps

Topics like sustainability, flexibility, efficiency and competitiveness are high level topics in today's society, which are in turn driven by important societal issues, such as environmental sustainability, the availability of energy and other raw materials, and rapidly changing market trends. To contribute to these topics, the rising field of Internet of Things (IoT) can be tamed in combination with the approach of a Service-oOriented Architecture. According to Evans the amount of IoT-devices will increase tremendously from 12.5 billion in the year 2010 to 50 billion in the year 2020. Setting up a solution needs to fill some gaps: interoperability of a wide range of IoT and legacy devices, automation requirement on latency guarantee/prediction for communication and control computations, scalability of automation systems enabling very large integrated automation systems, multi stakeholder integration and operations agility, security and related safety of automation systems as well as the ease of application engineering.

20.3 Identified Needed Tools

The definition of the tools is still in progress.

21. UC-17 [IFAT, AEE, EQUA, EQCH, AIT, FB]: Linking a Building Simulator to a Physical Building in Real-Time

Contact: Dagmar Jähnig [d.jaehnig@aee.at]

21.1 Use Case Overview

The goal of the use case is to run a building and system simulation of a demonstration building in real time during the operation phase using real-time data recorded on site. The main tool to be developed is a building tracker that forces a simulation program to match the data from the building in real time.
Secondly, the connection between the building sensors and the software will be investigated including safety and security issues. The real-time information about the building status can then be used to optimize the control parameters of the building.

21.2 Identified Gaps

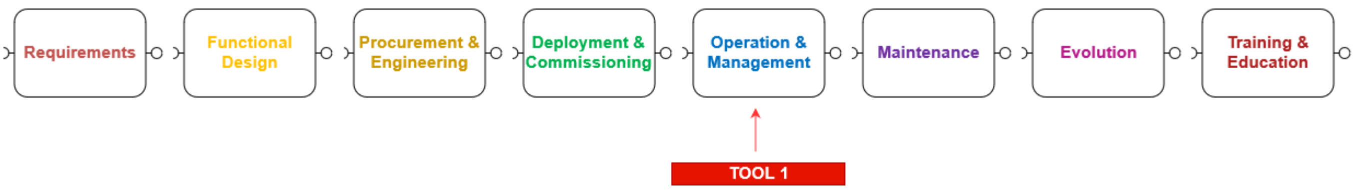
Up to now, optimization of a building's HVAC system is done

- during the planning phase: by means of building and system simulations using standard boundary conditions
- during the operation phase: by means of a monitoring program and manual adjustments to the control settings

21.3 Identified Needed Tools

Tool 1. **Building tracker software** - for control of the Building Management System on the basis of measured data

a. Run-time tool
b. Planned to connect with AHF
c. EPP5
d. Inputs: [Measured data of the boundary conditions: weather data, status of doors, windows, shading devices, CO2 levels in rooms; Measured data of building status for verification of tracker operation]
e. Outputs: [Signals to building management system (e.g. optimized set values, signals to actors such as valves or shading devices, …), virtual sensors that can be used for control purposes or fault detection]



22. UC-18 [Boliden, LTU, BnearIT]: Secure sharing of IoT generated data with partner ecosystem

Contact: Markus Frank [markus.frank@boliden.com]

22.1 Use Case Overview

Data in the mining industry is a key component and currency for further and also disruptive increase in operational efficiency and safety. SCADA's , PLC's, machines, vehicles, any equipment used in the production process generates valuable data and from may sides vendors / companies are pushing into this space to secure position to build further /new business or optimize current operations. Boliden has the position to own all data generated by equipment in operation for Boliden. Main purpose of this activity is to enable Boliden to have a secure way to make relevant data available via an open platform approach (does not have to be Boliden internal) to securely share data within the partner ecosystem including tracking use rights, traceability of usage and possibility to invalidate data.

Based on business urgency Boliden has started a prioritized activity to implement such a platform in combination with Use case - Deployment engine for production related sensor data and the ambition is to finalize, validate and evaluation real world verification of sharing use cases. Activities with LTU and BnearIT are in planning to benchmark, mirror / extend the sue case using Arrowhead framework.

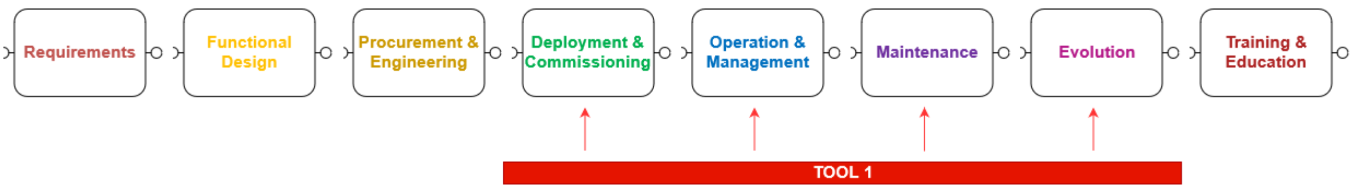
22.2 Identified Gaps

Today, data access is handled by existing tools and following issues exist: 1. Potential legal issues arise, as legacy systems and current approaches do not provide possibility to security segment data. 2. Concept for data tagging is not available as standard 3. Possibility to trace where Boliden data is going and possibility of invalidation is not existing today. There are many tools to potentially fulfill this requirement but this issue has not been solved with respect to the above challenge.

22.3 Identified Needed Tools

Tool 1. Boliden data platform (**BDP**)

a. Run-time tool
b. Not connected to the AHF, discussion ongoing to extend / benchmark with AH together with LTU
c. EPP4, EPP5, EPP6, EPP7
d. Inputs: production data from various sources
e. Outputs: production data (transformed) to various destination systems



23. UC-19 [3E, ALLTalk, Sirris]: Deployment and configuration

Contact: Jérôme Genot [Jerome.genot@3e.eu]

23.1 Use Case Overview

3E provides a hardware independent software-as-a-service platform, called SynaptiQ, to perform performance monitoring of renewable energy assets. SynaptiQ Solar is the trusted B2B performance monitoring solution of key reference players, steadily growing in connected capacity with a CAGR +30%. It is now connected to more than 6.000 commercial solar power plants that represent together about 3 GW of electricity production capacity in more than 20 countries. At each of the sites, SynaptiQ collects time series data, status and event data from power converters, power or energy measurement devices, meteorological sensors. In total more than 300.000 devices push at least every 10 minutes data to the SynaptiQ servers. Applying a domain model of the monitored power plants, SynaptiQ performs data enrichment and data aggregation on top of the incoming data streams in order to provide real-time analytical services.

23.2 Identified Gaps

Efficient configuration of the SynaptiQ services for new assets is hindered by the lack in industrial standardisation of the communication protocols of the local devices and the communication gateways. 3E has developed a mediation module that allows for efficient parsing of the incoming data flow in function of the given domain model. This method, strongly depending on the correct definition of the domain model, allows to prevent manual configuration of the parser and realizes considerable engineering time reduction. 3E aims to further reduce the configuration engineering time by further innovation in the process optimization and automation of exception handling. Two typical examples that would benefit from this are:

- The definition of the domain model is still a manual task performed by a domain expert and is therefore subject to errors. These errors are often detected only late after the deployment of the new plant in SynaptiQ and thus result in high costs for remediation of the historical data resulting from the wrong configuration. Developing automated processes that would detect configuration error and eventually correct errors in an early stage, would realize serious cost reduction. An additional step that will be analysed in this use case, is the import of domain model information from other sources (e.g. design software) that could in this way prevent configuration errors. We estimate that 10% of the plants in SynaptiQ suffer at this moment from minor to major configuration issues and that this results per wrongly configured site in at least 2 hours of reconfiguration effort per year. For the existing SynaptiQ portfolio this would represent a value of over 120 kEUR per year.
- During the lifetime of a solar asset, the domain model changes continuously because of equipment failure and equipment replacements. Typically 2% of the solar inverters are replaced every year in the system. At the moment this requires a manual reconfiguration of the domain model by changing the serial number and address of the replaced inverters. By advanced exception detection on the incoming data flow, this type of configuration changes could be detected automatically and a reconfiguration proposal could be made. In the best case even a fully automated reconfiguration could be established. Considering that the 3 GW capacity connected to SynaptiQ represents 60.000 inverters, this means that every year 1.200 inverters fail. Estimating the inverter reconfiguration engineering cost at 100 EUR (1h) per inverter, this brings the annual gain of a fully automated process to 120 kEUR per year for the existing portfolio.

It would be interesting to have a communication backbone where each module could listen and post. A common database with all common information about the configuration would be used by each module. An engineering tool would be useful to communicate with the common database and the communication backbone to detect every changes.

23.3 Identified Needed Tools

Currently the definition of the tools for this use case is still in progress concerning inputs and outputs.

Tool 1. SynaptiQ Configurator Module: defines the data model and the driver model

   a. Design-time tool
   b. Not connected to the AHF
   c. EPP2
   d. Inputs: TBD
   e. Outputs: [XML]

Tool 2. SynaptiQ Mediation Module: handles the heterogeneous date inflow from on-site devices

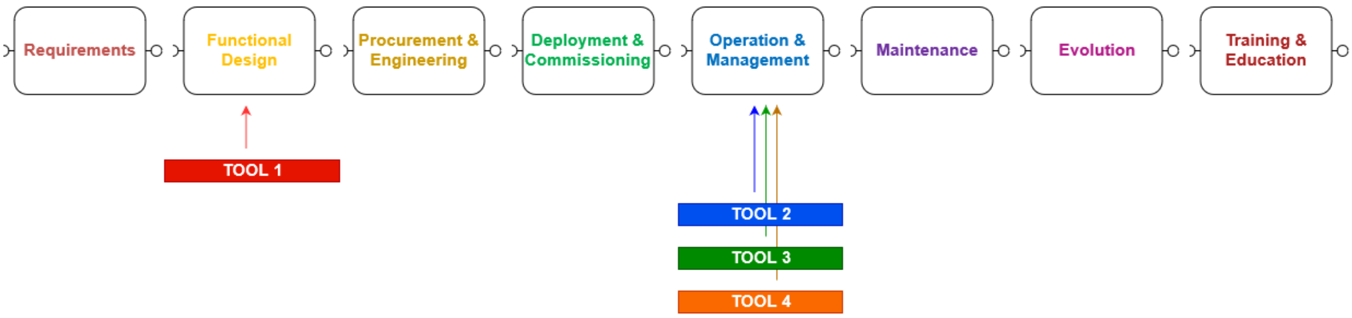
   a. Run-time tool
   b. Connected to the AHF
   c. EPP5
   d. Inputs: TBD
   e. Outputs: TBD

Tool 3. SynaptiQ Data Engine: processes the inflow of information

   a. Run-time tool
   b. Connected to the AHF
   c. EPP5
   d. Inputs: [XML]
   e. Outputs: TBD

Tool 4. SynaptiQ Operational: handles the visualization of the aggregated data and alarms

   a. Run-time tool
   b. Connected to the AHF
   c. Operation & Management EPP5
   d. Inputs: [alarms info, XML data]
   e. Output: TBD



24. UC-20 [FARR, IKERLAN]: Elastic Data Acquisition System

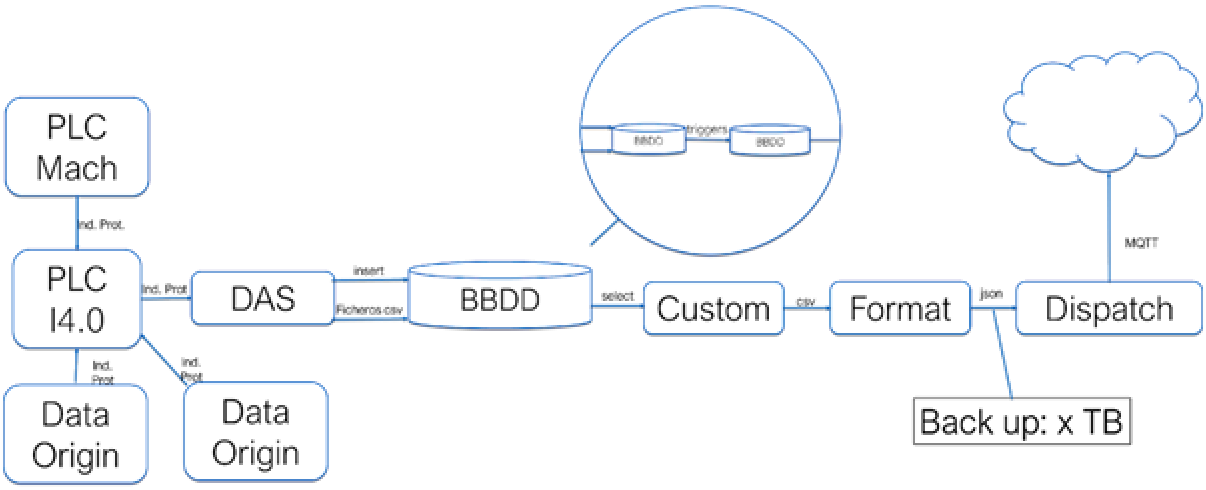
Contact: Jon Rodriguez, [j.rodriguez@fagorarrasate.com]

24.1 Use Case Overview

The aim is to develop an Elastic Data Acquisition System that implements different services that will help in Deployment & Commissioning, Operation and Management and Maintenance engineering processes.

In the next figure, the overview of the baseline of the system is shown. The system is composed of different elements:

- PLC: Siemens, Beckhoff, ...
- DAS: .NET based custom software
- BBDD: SQL Server / Redis
- Custom: Custom applications developed by engineers or Data Scientists.
- Format: .NET based software prepared to format the data.
- Dispatch: .NET based software prepared to send the data.
- Cloud: Fagor Arrasate's IoT Platform



With the whole system it is possible to define the PLC variables that are going to be uploaded to the cloud. To do this, several parameters must be specified, e.g. the sampling frequency for each variable and the industrial protocol to use or the final cloud endpoint to upload the variable.

24.2 Identified Gaps

Deployment & Commissioning: two main problems exist in this phase: 1) the configuration of the variables that need to be monitored are manually configured. 2) the variables to monitor can evolve over time, this makes mandatory to reconfigure the system in order to start monitoring the new variables. These two problems make the process costly in terms of time and therefore as it is not an automated process, this one is error-prone witch makes the process maintenance being more expensive. Thus two missing tools are identified in this phase:

- **NT1**- Deployment configuration of the DAS: Services to configure the DAS variables remotely, get the current configuration and son on.

- **NT2**- Validation of the new configuration before deploying: The configuration done should be automatically validated before deploying.

Operation & Management: the main problem of this phase is that the configuration of the dispatcher to optimize the number of threads according to the number of resources is manual. Thus two missing tools are identified in order to automate this process:

- **NT3**- Monitoring and diagnosis of the active services/deployed versions: The system will be performing a self-diagnosis in order to know which is the status of every layer of the system.
- **NT4**- Automatic reconfiguration of the number of threads according to the number of resources: The system will be automatically performing a self-diagnosis and based on that it will automatically configure the number of threads to increase performance.

Maintenance: the main problem of this phase is that the updates of every system are manual and face to face. Thus, in order to reduce costs and streamline the process the next tool is identified:

- **NT5**- remote updates of the entire system: Within this tool, we will be able to know which is the version running and we will be able to deploy a newer one remotely.

24.3 Identified Needed Tools

Currently the definition of the tools for this use case is still in progress concerning inputs and outputs.

Tool 1. **NT1 –** deployment and configuration of Data Acquisition System (DAS)
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP4
  d. Inputs: TBD
  e. Outputs: TBD

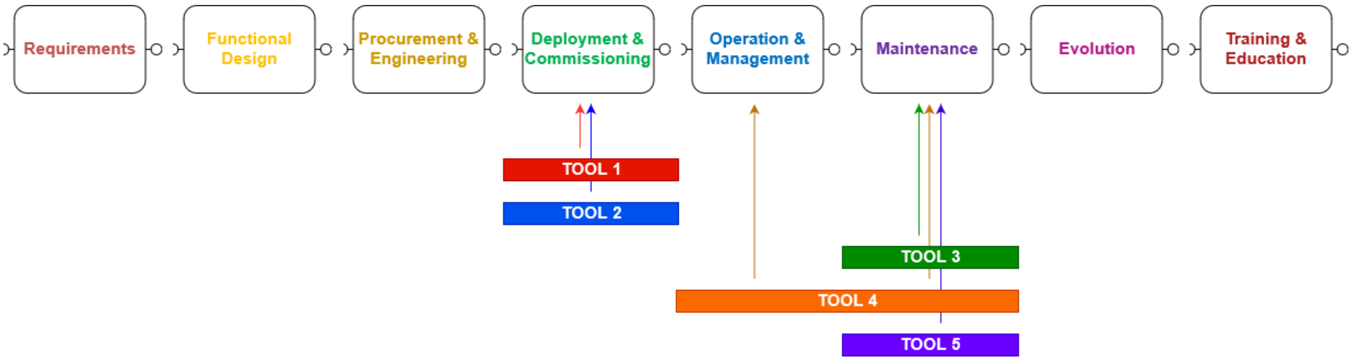
Tool 2. **NT2 –** validation of the configuration before the deployment
  a. Design-time tool
  b. Connected to the AHF
  c. EPP4
  d. Inputs: TBD
  e. Outputs: TBD

Tool 3. **NT3 –** monitoring and diagnosis of the deployed services
  a. Run-time tool
  b. Connected to the AHF
  c. EPP6
  d. Inputs: TBD
  e. Outputs: TBD

Tool 4. **NT4 –** the reconfiguration of the number of threads with respect to the available resources
  a. Run-time tool
  b. Connected to the AHF
  c. EPP5, EPP6
  d. Inputs: TBD
  e. Outputs: TBD

Tool 5. **NT5** – remote updates of the system

a. Run-time tool
b. Connected to the AHF
c. EPP6
d. Inputs: TBD
e. Outputs: TBD



25. UC-21 [ABB, CSC, Wapice, VTT]: Data-based digital twin for electrical machine condition monitoring

Contact: Jan Westerlund (ABB) [jan.westerlund@fi.abb.com], Peter Råback (CSC) [peter.raback@csc.fi], Laurentiu Barna (Wapice) [laurentiu.barna@wapice.com], Janne Keränän (VTT) [janne.sami.keranen@vtt.fi], Jari Halme (VTT) [jari.halme@vtt.fi]

25.1 Use Case Overview

Digital twin is a major growing technology trend. The same applies for machine learning, which all the time is finding new uses. Being able to link these technologies via the Arrowhead Tools allows their use in a variety of different domains. The use case will develop data-based digital twin for electrical machine condition monitoring, and will utilize different analysis methods for electrical machine performance evaluation and condition monitoring consisting functional design, engineering, operation management, maintenance and evolution phases.

25.2 Identified Gaps

A major **missing element** is a direct link from the design phase to performance evaluation, along with improved performance and condition evaluation models.
The current status is that **manual work is needed** in between the engineering phases, although separate sub-phase activities are mainly automated. In the future, the objective is to integrate automatically different phases in order to get economical benefit.

25.3 Identified Needed Tools

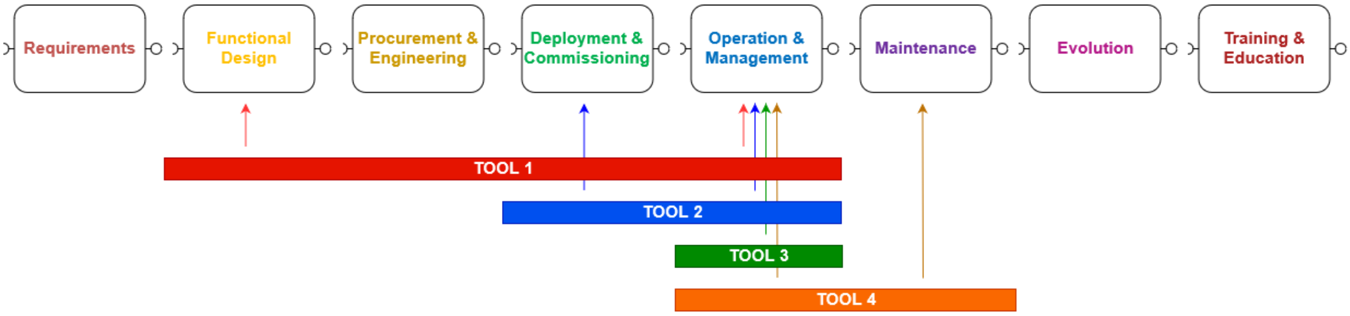
Tool 1. Integrated data transfer tool from electrical machine functional design to operation management and performance evolution.
   a. design-time tool
   b. not connected to the AHF
   c. EPP2, EPP5
   d. Inputs: Functional design plans
   e. Outputs: Performance model data or operational system descriptions

Tool 2. Local cloud integration of IoT-Ticket and ABB Ability
   a. run-time tool
   b. connected to the AHF
   c. EPP4, EPP5
   d. Inputs: IoT platform data from IoT-Ticket and ABB Ability
   e. Outputs: digital twin data

Tool 3. Performance data storage based on MariaDB
   a. Run-time tool
   b. Connected to the AHF
   c. EPP5
   d. Inputs: performance data
   e. Outputs: data in the storage

Tool 4. Kubernetes-based coupling of FEM and AHF
   a. Run-time tool

b. Connected to the AHF
c. EPP5, EPP6
d. Inputs: Parameters for the running electrical machine in a common exchange format, e.g., JSON
e. Outputs: Parameters describing the operation of the electrical machine in a common exchange format, e.g., JSON



26. UC-22 [STM, TECHNE, Magillem, CEA, LTU, BME]: Arrowhead Framework training tool

Contact: Asma Smaoui (CEA) [asma.smaoui@cea.fr], Emmanuel Vaumorin (Magillem) [vaumorin@magillem.com], Marcello Coppola (STM) [marcello.coppola@st.com]

26.1 Use Case Overview

The use case Arrowhead training tools will provide engineering content for the Training engineering phase. Taking into account the requirements structure issued by task 3.1.3, the task provides a definition of the targeted process to be supported by the Arrowhead Framework for educational use case. The outcome of this task is a pilot training/educational program that can be easily customised to other specific needs of other training programs and their objectives.

26.2 Identified Gaps

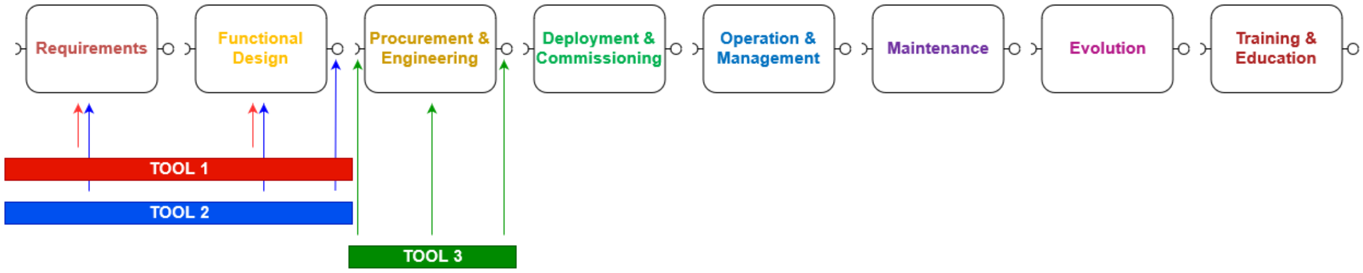
The gaps are to be identified as a part of the Use Case.

26.3 Identified Needed Tools

Tool 1. Papyrus - a tool for defining the requirements and for software design
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP1, EPP2
  d. Inputs: [Hardware-related information]
  e. Outputs: [Blockly configuration file]

Tool 2. Magillem tool - for the requirements definition and hardware design
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP1, EPP2
  d. Inputs: -
  e. Outputs: [Hardware platform-relative information] [EP-O2]

Tool 3. Hardware programming tool - a tool that uses Blockly and Scratch to program STM32MP1 card
  a. Design-time tool
  b. Not connected to the AHF
  c. EPP3
  d. Inputs: [Configuration file] [EP-I3]
  e. Outputs: [STM32 hardware blocks description] [EP-O3]



27. Revision history

27.1 Contributing and reviewing partners

Contributions	Reviews	Participants	Representing partner
X	X	Marek Tatara	DAC
X	X	Federico Montori	IUNET
X		Sara Bocchio	ST-I
	X	Zoltán Béres	IQL
	X	Kadosa Koltai	IQL
	X	Davide Brunelli	IUNET
X	X	Dave Marples	TNL
	X	Gianvito Urgese	POLITO
	X	Edoardo Patti	POLITO
	X	Richard Hedman	VTC
	X	Barbara Jung	Expleo
	X	Carlos Yurre	AOTEK
	X	Mikel Carrasco	Fagor Automation
	X	Mustafa Küçükkuru	Arcelik
X	X	Anja Zernig	KAI

27.2 Amendments

No.	Date	Version	Subject of Amendments	Author
1	2019-10-02	0.1	First draft	Marek Tatara
2	2019-10-25	0.2	Extended description of the use cases	Federico Montori, Marek Tatara
3	2019-11-11	0.3	Insertion of new use cases	Marek Tatara
4	2019-11-22	0.4	Insertion of new use cases	Federico Montori
5	2019-12-05	0.5	Integration of the peer revision	Federico Montori, Marek Tatara
6	2019-12-06	0.6	Format adjustment	Federico Montori
7	2019-12-10	1.0	Final version	Federico Montori, Marek Tatara

27.3 Quality assurance

No	Date	Version	Approved by
1	2019-12-11	1.0	Jerker Delsing, TBC