
# Transform safety-critical Cyber Physical Systems into distributed solutions for end-users and partners

## D23 (D1.3)

## TRANSACT transition guide to facilitate safety-critical distributed CPS solutions v1

This project has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No 101007260. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Netherlands, Finland, Germany, Poland, Austria, Spain, Belgium, Denmark, Norway.


# Document Information

| | |
|---|---|
| **Project** | TRANSACT |
| **Grant Agreement No.** | 101007260 |
| **Work Package No.** | WP1 |
| **Task No.** | T1.3 |
| **Deliverable No.** | D23 |
| **Deliverable No. in WP** | D1.3 |
| **Deliverable Title** | TRANSACT transition guide to facilitate safety-critical distributed CPS solutions v1 |
| **Nature** | Report |
| **Dissemination Level** | Public |
| **Document Version** | v1.0 |
| **Date** | 31/05/2023 |
| **Contact** | Nika Chkhaidze |
| **Organization** | PMS |
| **Phone** | +31621978524 |
| **E-Mail** | nika.chkhaidze@philips.com |

# Authors Table

| NAME | COMPANY | E-MAIL |
|---|---|---|
| Abel Gómez | UOC | agomezlla@uoc.edu |
| Anders Holme | NVT | Anders.holme@navtor.com |
| Arnold Akkermann | DLR-SE | Arnold.Akkermann@dlr.de |
| Bjørn Åge Hjøllo | NVT | Bjorn.hjollo@navtor.com |
| Dan Davies | FLEET | dan.davies@fleetonomy.ai |
| Desi Esclapez | DAM | desi.esclapez@dam-aguas.es |
| Iván Alfonso | UOC | ialfonsod@uoc.edu |
| Jeanneth Nodland | NVT | Jeanneth.nodland@navtor.com |
| Mika Jaakonaho | FLEET | mika.jaakonaho@fleetonomy.ai |
| Nika Chkhaidze | PMS | nika.chkhaidze@philips.com |
| Robert Hofsink | PMS | robert.hofsink@philips.com |
| Krzysztof Oborzynski | PMS | krzysztof.oborzynski@philips.com |
| Wolfram Ratzke | AVL | Wolfram.Ratzke@avl.com |

# Reviewers Table

| VERSION | DATE | REVIEWER |
|---|---|---|
| 0.5 | 08/02/2023 | Teun Hendriks (TNO), Lukasz Szczygielski (GUT), Wouter Tabingh Suermondt (TNO) |
| 0.8 | 15/05/2023 | Teun Hendriks (TNO), Wouter Tabingh Suermondt (TNO) |
| 0.9 | 26/05/2023 | Sasa Marinkovic (PMS) |

# Change History

| VERSION | DATE | REASON FOR CHANGE | AFFECTED PAGES |
|---|---|---|---|
| v0.1 | 27/07/2022 | Initial version. Defined initial table of contents and the chapter structure of the document. | All |
| v0.2 | 03/10/2022 | Added initial content to Sections 4.2, 4.3, 4.5. | All |
| v0.3 | 07/11/2022 | Sections 4.2, 4.5 ready for internal review.<br>Updated 4.3.<br>Added initial content to sections 4.1, 4.4. | All |
| v0.4 | 09/01/2023 | Sections 4.2, 4.3 4.5 ready for formal review.<br>Updated sections 4.1, 4.4. | All |

| VERSION | DATE | REASON FOR CHANGE | AFFECTED PAGES |
|---|---|---|---|
| v0.5 | 08/02/2023 | Chapters 3 and 5 completed.<br>All sections of Chapter 4 completed.<br>Version for the first round of external review. | All |
| v0.6 | 05/04/2023 | Incorporated comments in Chapter 3 and 5.<br>Merged sections describing the methodology into Chapter 3. | Page 13-28, 59-60 |
| v0.7 | 14/04/2023 | Incorporated comments in Chapter 4. | Pages 30-57 |
| v0.8 | 12/05/2023 | Addressing comments from the 1st review and submitting for the 2nd review round. | All |
| v0.9 | 19/05/2023 | Addressing comments from the 2nd review. | All |
| v1.0 | 30/05/2023 | Finalizing the document for the submission. | All |

# Table of Contents

**1 GLOSSARY ..... 9**

**2 INTRODUCTION ..... 11**

2.1 PURPOSE OF THE DELIVERABLE ..... 11
2.2 RELATIONSHIP TO OTHER TRANSACT DOCUMENTS ..... 11
2.3 STRUCTURE OF THE DELIVERABLE ..... 11

**3 INTRODUCTION TO THE TRANSACT TRANSITION METHODOLOGY ..... 12**

3.1 TRANSFORMATION AREA: BUSINESS ..... 14
3.1.1 Business drivers ..... 15
3.1.2 Solution requirements ..... 16
3.2 TRANSFORMATION AREA: ARCHITECTURE ..... 16
3.2.1 Product Architecture ..... 17
3.2.2 Product Development ..... 22
3.2.3 Product Operation ..... 23
3.2.4 DevOps ..... 25
3.2.5 Infrastructure ..... 27
3.3 TRANSFORMATION AREA: ORGANIZATION ..... 27
3.3.1 People/skills ..... 28
3.3.2 Processes and technology ..... 28
3.3.3 Organization structure ..... 29
3.4 TRANSFORMATION CROSS-CUTTING ASPECTS ..... 29
3.4.1 Cross-cutting aspects: Safety ..... 30
3.4.2 Cross-cutting aspects: Performance ..... 33
3.4.3 Cross-cutting aspects: Security and privacy ..... 36
3.4.4 Cross-cutting aspects: Regulatory and certification ..... 40
3.5 PLANNING AND EXECUTION OF THE TRANSITION ..... 42
3.6 SUMMARY ..... 42

**4 TRANSITION OF USE CASES TO TRANSACT TRANSITION METHODOLOGY ..... 44**

4.1 TRANSITION OF USE CASE 1: REMOTE OPERATIONS OF AUTONOMOUS VEHICLES FOR NAVIGATING IN URBAN CONTEXT ..... 44
4.1.1 Transition to TRANSACT Reference Architecture ..... 46
4.1.2 Organizational Changes to support the transition ..... 47
4.1.3 Planning and execution of the transition ..... 48
4.1.4 Validation and verification ..... 50
4.1.5 Lessons learned ..... 50
4.2 TRANSITION OF USE CASE 2: CRITICAL MARITIME DECISION SUPPORT ENHANCED BY DISTRIBUTED, AI ENHANCED EDGE AND CLOUD SOLUTIONS ..... 50
4.2.1 Transition to TRANSACT Reference Architecture ..... 52
4.2.2 Organizational changes to support the transition ..... 53
4.2.3 Planning and execution of the transition ..... 54
4.2.4 Lesson learned ..... 56
4.3 TRANSITION OF USE CASE 3: CLOUD-FEATURED BATTERY MANAGEMENT SYSTEM ..... 56
4.3.1 Transition to TRANSACT Reference Architecture ..... 57
4.3.2 Organizational changes to support the transition ..... 59
4.3.3 Planning and execution of the transition ..... 60
4.3.4 Lesson learned ..... 60
4.4 TRANSITION OF USE CASE 4: EDGE-CLOUD-BASED CLINICAL APPLICATIONS PLATFORM FOR IMAGE GUIDED THERAPY AND DIAGNOSTIC IMAGING SYSTEMS ..... 61

4.4.1 Transition to TRANSACT Reference Architecture ..... 62
4.4.2 Organizational changes to support the transition ..... 63
4.4.3 Planning and execution of the transition ..... 64
4.4.4 Lessons learned ..... 65
4.5 TRANSITION OF USE CASE 5: CRITICAL WASTEWATER TREATMENT DECISION SUPPORT ENHANCED BY DISTRIBUTED, AI ENHANCED EDGE AND CLOUD SOLUTIONS ..... 65
4.5.1 Transition to TRANSACT Reference Architecture ..... 66
4.5.2 Organizational changes to support the transition ..... 70
4.5.3 Planning and execution of the transition ..... 72
4.5.4 Lessons learned ..... 72
**5 SUMMARY ..... 73**
**6 REFERENCES ..... 74**

# List of Figures

Figure 1: Initial TRANSACT transition methodology elements ..... 12
Figure 2: TRANSACT transition: Business Transformation Area elements ..... 15
Figure 3: TRANSACT transition: Architecture Transformation Area elements ..... 17
Figure 4: TRANSACT Reference Architecture ..... 18
Figure 5: DevOps main phases ..... 26
Figure 6: TRANSACT transition: Organization Transformation Area elements ..... 28
Figure 7: TRANSACT transition: Cross-cutting aspects ..... 30
Figure 8: DevOps pipeline with the safety engineering activities ..... 33
Figure 9: Y-chart based performance modelling ..... 35
Figure 10: MAPE-K autonomic loop (from (Kephart & Chess, 2003)) ..... 35
Figure 11: DevOps pipeline with the performance engineering activities ..... 36
Figure 12: DevOps pipeline with the security and privacy activities ..... 40
Figure 13: DevOps pipeline with the regulatory activities ..... 41
Figure 14: The remote operations use case cloud-edge-device continuum ..... 45
Figure 15 The transition to TRANSACT reference architecture specific to UC1 ..... 46
Figure 16: UC2 targeted solution in which navigators and operators are supported by edge and cloud monitoring and decision support services ..... 51
Figure 17: Overview over the advisory service framework of UC2 to be developed and implemented in the TRANSACT project ..... 51
Figure 18: Representation of UC2 TRANSACT architecture mapping. ..... 52
Figure 19: High-level picture for Use Case 3. The Battery Management System is connected to a cloud in order to transmit telemetric data and to receive software updates. Beside of this, in-deep analytics can be performed by applying neural networks, either direct ..... 57
Figure 20: A local and isolated Battery Management System. ..... 58
Figure 21: System design of the baseline setup. The BMS is only accessible via the On-Board Diagnosis (OBD) interface which requires a wire-bound connection. The communication to other electrical control units (ECU) is established via the CAN bus. ..... 58
Figure 22: System design after the transformation. The Gateway and Cloud can host several additional components and services to provide additional features. ..... 58
Figure 23: Connection between TRANSACT components and BMS functions and services. ..... 59
Figure 24: Example workflow for image guided diagnosis and therapy. ..... 61
Figure 25: Typical setting during image-guided therapy with physicians utilizing medical imaging equipment for the minimally invasive treatment of patients ..... 61
Figure 26: Mapping of Use Case 4 to Transact Reference Architecture ..... 62
Figure 27: Use Case 5, High Level Architecture ..... 68
Figure 28: Radiatus PaaS ..... 71
Figure 29: Kumori Platform is a PaaS for deploying and managing services based on containers ..... 71
Figure 30: UC5 IT infrastructure ..... 72

# List of Tables

Table 1: Terms Definitions ..... 9
Table 2: Abbreviations ..... 9
Table 3: Migration strategies to be considered per the TRANSACT functions types ..... 20

# 1 Glossary

Table 1: Terms Definitions

| TERM | DEFINITION |
|---|---|
| Distributed CPS solution | The CPS product deployed on the device-edge-cloud infrastructure. |
| CPS Product | Applications/services/functions providing the CPS functionality. |
| Infrastructure | The device-edge-cloud environment using proprietary and public cloud-based solutions (IaaS/PaaS/SaaS). |
| Functional Requirement | Describes a software system or its component functionality. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. |
| Non-functional Requirement | Define system attributes such as security, reliability, performance, maintainability, scalability, and usability (also known as system qualities). Non-functional requirements ensure the usability and effectiveness of the entire system. Failing to meet any one of them can result in systems that fail to satisfy internal business, user, or market needs, or that do not fulfil mandatory requirements imposed by regulatory or standards agencies which may , in some cases, non-compliance can cause significant legal issues (privacy, security, safety, to name a few). |

Table 2: Abbreviations

| ACRONYM | DEFINITION |
|---|---|
| AI | Artificial Intelligence |
| AIS | Automatic Identification System |
| API | Application Programming Interface |
| AV | Autonomous Vehicle |
| AWS | Amazon Web Services |
| BMS | Battery Management System |
| CPS | Cyber-Physical System |
| DDS | Data Distribution Service |
| ECDIS | Electronic Chart Display and Information System |
| FMEA | Failure Mode and Effects Analysis |
| FTA | Fault Tree Analysis |
| GDPR | General Data Protection Regulation (EU) |
| HIPAA | Health Insurance Portability and Accountability Act |
| IaaS | Infrastructure as a Service |
| IoT | Internet of Things |
| ISMS | Information Security Management System |
| ISO | International Organization for Standardization |
| IT | Information Technology |

| ACRONYM | DEFINITION |
|---|---|
| KPI | Key Performance Indicator |
| LAN | Local Areas Network |
| LTE | Long Term Evolution |
| MD-SysPE | Model Driven System Performance Engineering |
| ML | Machine Learning |
| MQTT | Message Queuing Telemetry Transport |
| MR | Magnetic Resonance |
| NFR | Non-functional Requirement |
| PAAS | Platform as a Service |
| PKI | Public Key Infrastructure |
| QoS | Quality of Service |
| SLA | Service Level Agreement |
| SLO | Service Level Objective |
| SW | Software |
| SaaS | Software as a Service |
| TRL | Technology Readiness Level |
| UC | Use Case |
| V&V | Verification & Validation |
| VM | Virtual Machine |
| WP | Work Package |
| WWTP | Waste Water Treatment Plant |

# 2 Introduction

The purpose of the document is to propose a methodology for transforming the local safety critical CPS into the distributed safety-critical CPS solution deployed across device-edge-cloud continuum. The TRANSACT project follows use-case driven approach therefore, the transition methodology is build based on learnings from the use-cases and their transition to the TRANSACT Reference Architecture.

This document presents an *initial* version of the TRANSACT transition methodology. It focuses on the key areas that play important role for the transition, namely, business, architecture, and organization. Next to those areas, it explicitly evaluates the impact on those key areas by safety, performance, security, privacy, and regulatory aspects.

More details about each aspect of the transition will be added in the second version of this document when the final deliverables from WP1, WP2, WP3, and WP4 will be available, including the lesson learned from the use-cases transition to the TRANSACT Reference Architecture (that will form the generic recommendations for the planning and tactics when executing the transformation).

## 2.1 Purpose of the deliverable

This document has the following major purposes:

- Consolidating information about the transition from standalone on-device CPS system to the distributed CPS solution.
- Collecting information about the impact of the changes on the architecture and organization to support the transition based on use-cases migration experience.
- Providing guidelines for planning and execution of the transformation based on the TRANSACT transition methodology.

This version of the deliverable collects current project results about migrating the use-cases towards the distributed CPS solution to form the *initial* TRANSACT transition methodology. The next version (V2) of this document will consolidate final results and derive domain independent methodology to transform a local, stand-alone CPS into a safe and secure distributed safety-critical CPS solution.

## 2.2 Relationship to other TRANSACT documents

This document relates to the following TRANSACT deliverables:

- TRANSACT reference architecture (D2.1),
- Use case descriptions and related end-user and technical requirements (D1.1, D1.2),
- Safety, Performance, Security and Privacy concepts (D3.1, D3.2, D3.3, and D3.4),
- Strategies for continuous updating and independent releasing (D4.2).

## 2.3 Structure of the deliverable

The structure of this deliverable is as follows. First, Section 3 gives an introduction to the *initial* TRANSACT transition methodology with focus on business, architecture, and organizational areas that are impacted by such a transition. In addition, the critical cross-cutting aspects like safety, performance, security, privacy and regulatory are analysed to clarify their relation to the transition. As TRANSACT follows the use-case driven methodology, Section 4 presents so far the lesson learned from migration of the use-case towards the TRANSACT reference architecture. The final use-case evaluation and general recommendations for transforming the local safety critical CPS into the distributed safety-critical CPS solution will be provided in the next version (V2) of this deliverable.

# 3 Introduction to the TRANSACT Transition Methodology

The TRANSACT Transition Methodology focuses on the transformation of the monolithic Cyber Physical Systems to distributed solutions—such transition helps moving from a *product-centric* to a *solution-centric* propositions. However, in order to successfully execute such a transformation, it may require changes not only to the product but also in the business and organization areas. Therefore, the proposed TRANSACT Transition Methodology spans over business, architectural, and organizational areas. In addition, there are the cross-cutting aspects impacting multiple parts in each area that are critical in the CPS transformation to the device-edge-cloud continuum, namely: safety, performance, security, privacy, regulatory, and certification—Figure 1 presents initial TRANSACT transition methodology.

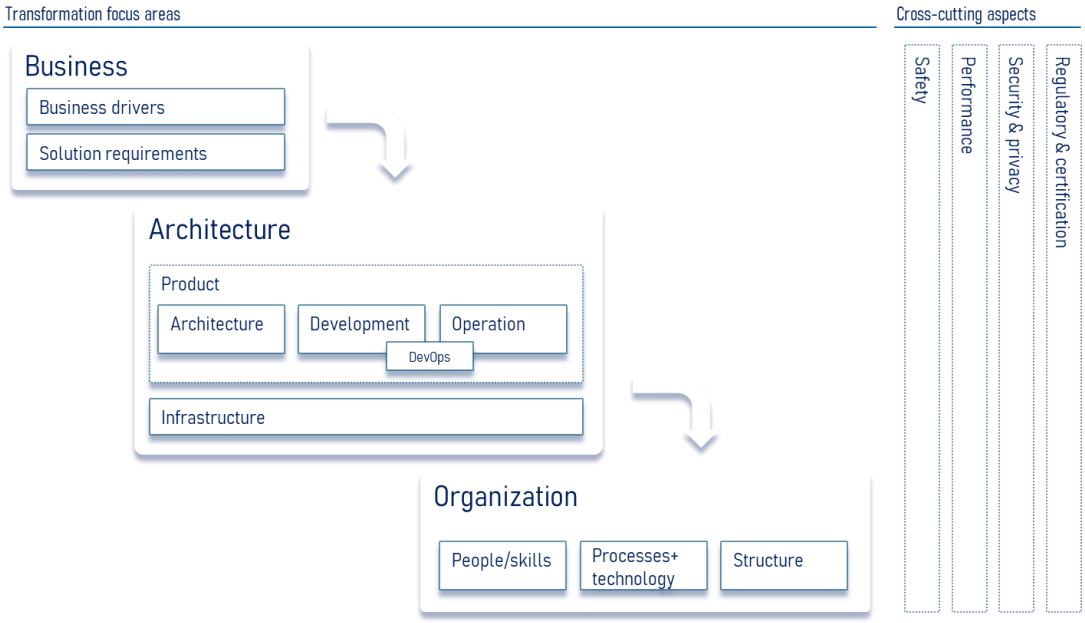


Figure 1: Initial TRANSACT transition methodology elements

The initial TRANSACT transition methodology is primarily based on the transformation focus areas that are influenced by the CPS critical cross-cutting aspects (see Figure 1). Specifically, the main elements are:

- **Transformation focus areas**: the core parts that are impacted by the transformation, namely:
  - **Business**: it focuses on answering the question of *WHY to engage into the transformation* to the edge/cloud-based solution and what is the business value for the products and company. Therefore, it concentrates on identifying and defining *business drivers* that are the reasons to switch to the new edge/cloud solutions. The identified business drivers require re-evaluation and redefinition of the existing *solution requirements* (as the main input for the new architecture) therefore they are also part of this area.
  - **Architecture**: it focuses on answering the question of *HOW to transform existing solution towards safe and secure edge/cloud-based solution* that is compliant with the regulations and satisfies the business and customer needs. The architecture covers the realization of the overall distributed CPS solution[1], i.e., the *product* components, services, and applications together with the *infrastructure* the product runs on. However, the architecture is concerned not only with the product design, but also with the technological choices to effectively *develop* and *operate* the

[1] Solution = Product component/services/applications + Infrastructure

new edge/cloud-based solution. Moreover, the effective development and operation can be boosted by making choices helping to leverage the *DevOps* practices.

- **Organization**: it focuses on answering the questions: 1) *WHO can build and operate* distributed CPS solutions in terms of *peoples/skills* and the related teams' *structure*, and 2) *HOW effectively arrange the way-of-working* in terms of the *processes* and chosen *technologies*. It means that the organization need to ensure that the people with right edge/cloud technological knowledge and experience are available, the teams are arranged effectively (e.g., development vs operation teams) and they are supported by the efficient processes. In case, embracing DevOps approach the teams need to be self-organized in order to be end-to-end responsible for building, deploying, and operating business applications. Such a shift is also part of this area as it may require cultural transformation in order to build cost-efficient and innovative company.

- **Cross-cutting aspects**: are concerns that affects (in various level of degree) all the transformation focus areas, namely:
  - **Safety**: it covers safety aspects related to the CPS product and their impact on the business decisions and organization. Specifically, the *business* area needs to understand the impact on the solution safety when moving to the edge/cloud technologies; the *architecture* area is affected by focusing on safe product development and safe product operation in the new the edge/cloud context; and the *organization* area is affected in terms of safety-related processes that may need adaptation for the new edge/cloud setup.
  - **Performance**: it covers performance aspects related to the CPS product and their impact on the business decisions and organization. Specifically, the *business* area needs to understand the impact on the solution performance when moving to the edge/cloud technologies; the *architecture* area is affected by guaranteeing the required performance characteristics of the edge/cloud solution to support its safety and optimal customer experience; and the *organization* area is affected in terms of new skills and knowledge required to address new performance challenges as a result of migrating to the new edge/cloud setup.
  - **Security & privacy**: it covers security and privacy aspects related to the CPS product and their impact on the business decisions and organization. Specifically, the *business* area needs to understand the impact on the security and privacy regulations compliance on the business and the customer data when moving to the edge/cloud technologies; the *architecture* area is affected by focusing on secure product development and its secure operation in the edge/cloud environment; and the *organization* area is affected in terms of new skills and knowledge required (in the technical and legal domains) to address new security and privacy challenges as a result of migrating to the new edge/cloud setup.
  - **Regulatory & re-certification**: it covers regulatory and certification aspects related to the CPS product and their impact on the business decisions and organization. Specifically, the *business* area needs to understand the impact on the existing and new regulations compliance when moving to the edge/cloud technologies; the *architecture* area is affected by incorporating all the regulatory design requirements in the new solution and its operation; and the *organization* area is affected in terms of the regulatory-related processes that may need adaptation for the new edge/cloud setup.

The TRANSACT transition methodology recommends starting from the *business* perspective to underpin the needs and clarify what are the drivers and benefit that bring value to the business by providing solution spanning device-edge-cloud continuum. From that analysis should be clear which critical business goals are affected and which requirements need to be changed to support those new business goals.

Having updated requirements for the safety-critical CPS solution based on the device-edge-cloud continuum, the next transformation focus area is the new distributed CPS solution *architecture*. Based on the new

requirements the technical aspects of the current CPS system need to be re-evaluated, focusing on the product architecture, and its development and deployment. In additional, the product operational aspects need adaptation which will require new technologies and infrastructure to be in place to connect it to the edge/cloud services and functions.

Finally, the organizational area needs attention to ensure effective creation and maintenance of the new solution. Primarily, it requires peoples with new skills and knowledge related not only to the cloud/edge technologies but possibly with different way of thinking about product development, deployment and operation (DevOps). In addition, the product realization processes need adaptation to optimize solution creation. The new edge/cloud-based solution may require new organization structure to cover new areas that have been less relevant for the device-based solution, for example, a new dedicated security and operational teams may be needed, due to cloud privacy regulations there could be legal team extension needed, the billing of the new solution's functionality may impose changes to the financial department, etc.

Next to the three main transformation focus areas (business, solution architecture, and organization) there are key cross-cutting aspects that impacts considerably the transition steps and require special attention to ensure success of such a transition. Those are: safety, performance, security and privacy, and regulatory and certification.

The TRANSACT transformation approach does not require to address all the areas at once or in a sequence BUT having clear business vision and realizing it architecture are prerequisite to build coherent step-wise and iterative approach to achieve successful transformation of the device-based CPS solution to the edge/cloud-based CPS solution. Only by having clarity of the risks and trade-offs (e.g., accepting sub-optimal organization structure for developing the cloud-based solution) that lead to deliberate decisions and actions can bring the transformation to the successful conclusion.

The following sections elaborate about implications of transforming to the distributed CPS system solution for each focus area and the key cross-cutting aspects, specifically: Section 3.1 details Business transformation area, Section 3.2 details Architecture transformation area, Section 3.3 details Organization transformation area, and Section 3.4 details each cross-cutting concern as defined in the TRANSACT transition methodology.

## 3.1 Transformation area: Business

The Business area is concerned with the fundamental questions what the transformation means for the business goals and objectives and how the business and its customers benefit from transforming the existing standalone safety critical- CPS systems to the edge/cloud-based solution. Based on that impact, the requirements for the new solution should be formed so the solution architecture can be created. Therefore, this section focuses on the business drivers and new solution requirements as first steps in the transformation to distributed CPS solution—see Figure 2.

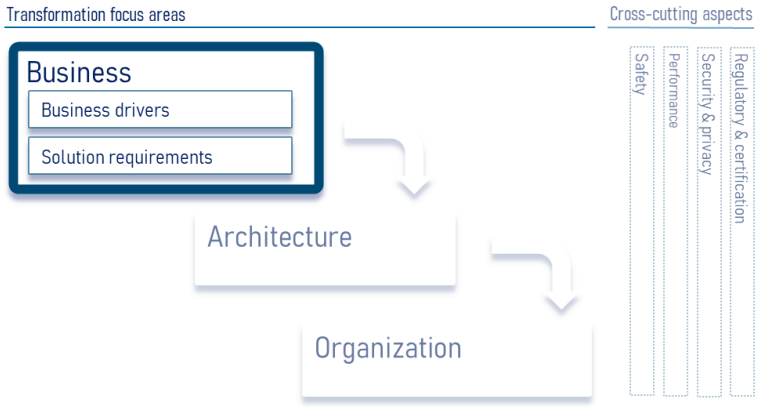


Figure 2: TRANSACT transition: Business Transformation Area elements

### 3.1.1 Business drivers

EU based initiatives have identified an extensive list of the drivers as well as potential barriers for embracing the Cloud, Edge and IoT technology in various industries and related use cases (EUCloudEdgeIoT.eu, 2023). The modern-day technological advances enable new opportunities to improve performance and add value to different areas or industries including smart manufacturing, autonomous vehicles, medical procedures, and other activities that involve CPSs in their supply chains. However, before starting any transition, it is vital to work out what the value of edge/cloud computing means for the company and its business. Therefore, the value of transforming the business to deliver edge/cloud based solutions needs to be clarified and well understood to reach the most optimal return on investment. Especially, in the context of the safety -critical CPS systems it has to be validated that such a transition does not jeopardize the system safety nor legal obligations therefore threating the business continuity.

Typical business drivers focus on improving the innovation capabilities, improving the speed of delivering solutions, and at the same time saving costs. Specifically for the safety-critical CPS the business drivers are:

- Improved user experience enabled by technological novelties,
- Faster delivery of innovative solutions to the market,
- Reduced R&D costs,
- Reduced Infrastructure costs by leveraging scalability of cloud platforms,
- Increase profit by developing new business and charging models for provided services,
- Enablement of remote operation of CPS,
- Enablement of remote maintenance and update of CPS,
- Reduction of response time to address system failures, especially, in the safety, performance, security, and privacy areas.

However, fulfilling these business drivers involves overcoming a set of barriers and challenges in terms of organization (e.g., lack of technical skills and knowledge), new edge/cloud-based solution architectures challenges (e.g., ensuring safety, performance, and security of the new systems, design based on more loosely coupled services), limitation of the current IT infrastructure (e.g., virtualization technologies, possible integration challenges with edge/cloud platforms), costs and complexity in operational system management, impact on regulatory and (re-)certification effort.

Such an analysis may lead to definition of new business models for the distributed solutions to monetize the resulting products. The new business models require special attention to the role and provisioning of the cloud services, the role of 3rd party services, the open-source software solutions, regulatory and certification.

Having clear business drivers and understanding the business-needs helps to clarify the requirements for the business and the products architecture and the future organization structure.

### 3.1.2 Solution requirements

Having business drivers clear the next step is to identify the relevant requirements that need re-evaluation and adaptation in the new edge/cloud solution. TRANSACT proposes to derive and classify the requirements based on the domain use-cases. Requirements in TRANSACT are classified (from the most generic ones to the most particular ones), into *End User Requirements* (EUR), *Functional and Non-Functional Requirements* (FR/NFR), and *Technical Requirements* (TR).

The *End User Requirements* represent the characteristics that the system must cover from the end user's point of view. These can be classified into categories, such as:

- General Requirements,
- Interoperability Requirements (including requirements for the communicating with the legacy systems),
- Command, Control and Coordination Requirements,
- Communications and Networking Requirements,
- Safety Requirements,
- Performance Requirements,
- Security Requirements.

Although the EURs involve information relevant to the system, it is necessary to derive the *Functional and Non-Functional Requirements* that directly influence the definition of the architecture and features of the system to be implemented. FRs describe what the system must do (i.e., the system functionalities), while NFRs describe the system's properties. Compliance with these requirements guarantees the quality of the system.

The *Technical Requirements* realize the specific behaviour of the FR/NFR assigned to the logical components. A TR can be a requirement for a hardware component, software component, or a combination of both. The requirements should allow the description of all necessary inputs, outputs and relationships between inputs and outputs, including constraints, and the interactions of the system with operators, maintainers and other systems.

Next, the requirements are grouped into clusters of functions and services (e.g., Identity and Access services, Auditing services, Safety, Performance and Security Monitoring Services, Data Services, etc.). These clusters of requirements are the inputs to the subsequent architecture activities.

## 3.2 Transformation area: Architecture

Having clarity why moving to the edge/cloud-based solution is desired from the business perspective, the next step in safety-critical CPS transformation is to map the business and product(s) requirements to the desired solution architecture. However, the complete solution architecture covers not only the product and related technologies but also the (edge/cloud) infrastructure the product will be developed-with, deployed-on and running-on. Therefore, the solution architecture should be a blueprint defining the overall approach to the product architecture, development, deployment, and its operation environment, including the underlying infrastructure. Such a blueprint should focus on assessing how to realize the transition so the *distributed* safety-critical CPS solution ensures safety, performance, security, privacy, and regulatory requirement compliance.

In order to make a successful transition from the on-device solution to the distributed solution using edge and cloud services requires careful consideration and planning. First, it is needed to thoroughly review and understand the current system architecture and the technology stack it is built on. Secondly, the cloud providers capabilities and services need to be explored, in order to optimally match application workloads to cloud provider environment and services in order to find the best technology and cost combination of services. Thirdly, the high level blueprint of the new solution should be created covering the product design and needed infrastructure supporting development, deployment, and operation of the solution. The following sections focus on each of those aspects as the next step in the transformation to distributed CPS solution—see Figure 3.

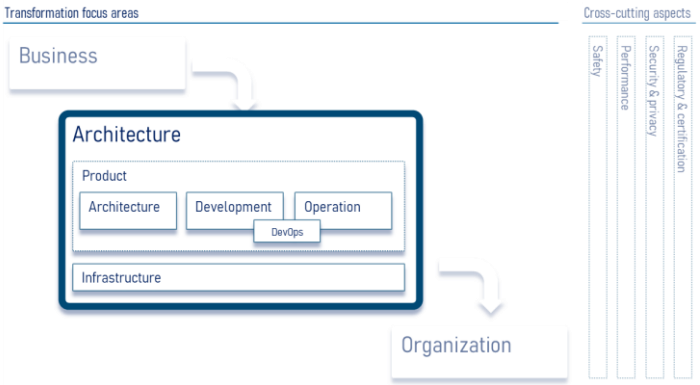


Figure 3: TRANSACT transition: Architecture Transformation Area elements

### 3.2.1 Product Architecture

Figure 4 presents the TRANSACT reference architecture that can be a starting point to define a specific product architecture and the needs from the underlying infrastructure (both for the product development and the product operation). This architecture defines 3-tiers (device, edge, and cloud) where the distributed safety-critical CPS solutions is deployed. Each tier provides a specific quality of service level especially with respect to performance aspect (such as response times and data transfer guarantees) which are critical for the safety critical and mission critical functions. For example, safety-critical applications often have hard real time constraints which lead to severe failures when missed, whereas the mission critical functions may have soft real time constraints which may degrade the system's quality of service when missed, but do not necessarily lead to failures. Next to safety also security and updates are important aspect of the distributed system.

Therefore, this architecture introduces several core services deployed across all the tiers to ensure safety, performance, and security of the new solution, i.e., the *Safety, Performance and Security Monitoring Services* are responsible for monitoring, detecting, and preventing safety, security and performance failures; the *Identity & Access service* contributes to system security by granting/denying access to system resources based on defined policies. To support the management and operation of the system, the *Remote Update Client* (running on the device) and the *Remote Update Coordinator* (running at the edge/cloud) are proposed to cooperate across tiers and perform service updates in a secure and safe way. All the services and their role in the architecture are presented in deliverable D2.1 (Arjona & et.al., 2022)—the roles and responsibilities of the services indicate the capabilities that need to be added or changed while migrating to the distributed CPS solution.

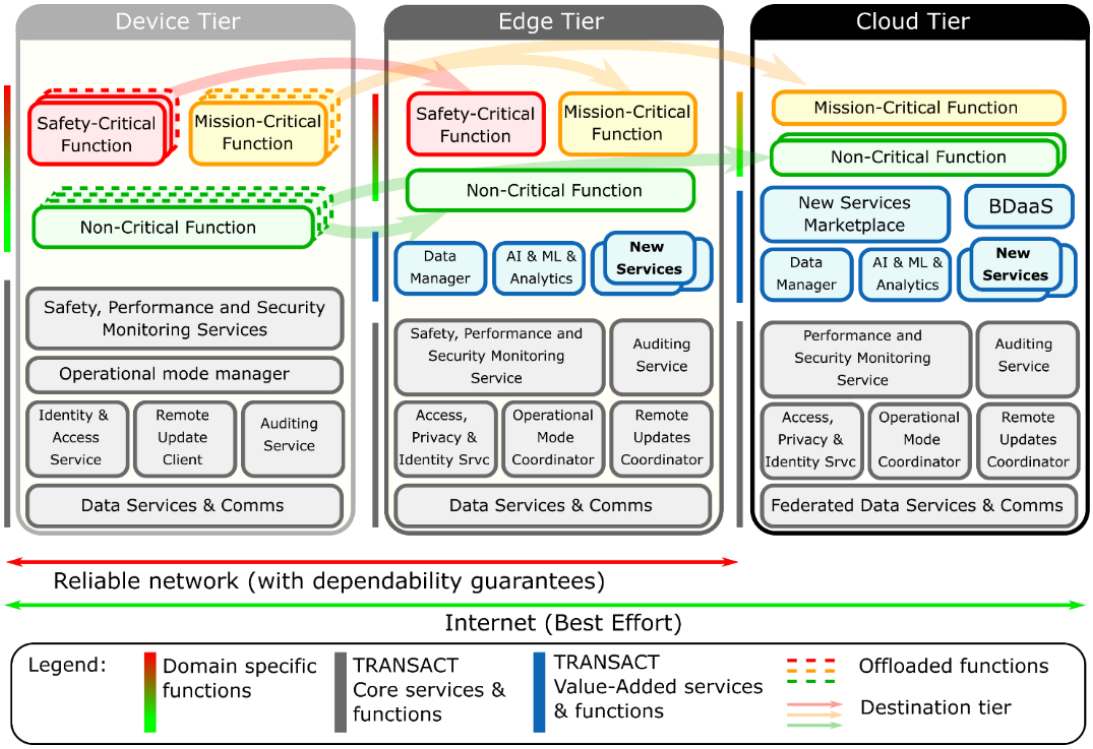


Figure 4: TRANSACT Reference Architecture

The following aspects impacting the architecture should be considered while transforming on-device CPS solution to the distributed CPS:

- Architectures for distributed safety-critical solutions,
- Migrating safety-, mission-, and non-critical functions to new architecture,
- System security,
- System updates,
- Observability as a key concept for system monitoring.

#### 3.2.1.1 Architectures for distributed safety-critical solutions

There are many (software) architecture styles described in the literature that may be applicable to safety-critical systems. Examples of such architecture styles are: layered style, microkernel style, pipe & filter style, event bus style, microservices architecture style (see (Buschmann, Meunier, Rohnert, Sommerlad, & Stal, 1996) (Schmidt, Stal, Rohnert, & Buschmann, 2000) (Kircher & Jain, 2004) (Buschmann, Henney, & Schmidt, Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing, 2007)).

When moving to the distributed environment it is highly recommended to consider building services and applications based on the loosely coupled services/components following the microservices architecture. This type of architecture is well supported by the cloud platforms as splitting the solution into smaller loosely couple services improve the overall robustness of the solution (the failure propagation is contained and not propagated), allows easier scaling of the functionality that needs more resources, allows incremental deployment of the functionality. However, next to the mentioned advantages the microservices architecture increases the complexity of the solution (as more elements needs to be managed), complicates problems troubleshooting and testing, and each microservice can be built with a different technology, so care needs to be taken to ensure easy development and management of used technologies. Therefore, it has to be well assessed to which system parts apply the microservices architecture. Deliverable D2.1 (Arjona & et.al., 2022) provides extensive discussion about the architectures applicable for distributed cyber-physical systems

including their strengths and weaknesses with special focus how those architectures impact the safety aspects.

#### 3.2.1.2 Safety-, Mission-, and Non-critical functions migration

The identification and classification of CPS domain functions is one of the tasks required for the TRANSACT reference architecture (see Figure 4), i.e., the CPS domain functions are categorized into three groups: safety-critical, mission-critical, and non-critical. These functions can be offloaded from the device to any of the other tiers (edge and cloud) and their classification depends on constraints and criticality (see D2.1 (Arjona & et.al., 2022)). Specifically, the safety-critical functions are characterized by hard real-time constraints and are recommended to be deployed at the device or edge tiers to guarantee fast response. The unavailability of this type of function can generate a series of severe failures impacting the quality of service of the system. In contrast, mission-critical functions tend to have soft real time constraints and could be deployed on any of the tiers of the architecture. The unavailability of a mission-critical function can degrade the system's quality of service but does not necessarily produce severe failures. Finally, non-critical functions are those that are not essential to the operation of the system.

One of the main purposes of performing the identification of CPS functions and components following the classification proposed by the TRANSACT architecture, is to plan an optimal offloading of service functions across the architecture tiers. While the device and edge layers offer advantages in terms of latency, security, and bandwidth consumption, they also have resource limitations. Therefore, services and functions could be deployed according to their classification. That is, the device tier should host only basic and safety-critical functions while the rest of the functions could be offloaded to the edge and cloud tiers. This way, reliability and performance at the device tier would improve because resources are dedicated to safety-critical functions. Additionally, edge and cloud facilitate the deployment and update of new functions using several well-established cloud's methods and technologies.

For each critical functionality it must be decided how it will be positioned and transformed to fit in the new edge/cloud architecture. Amazon AWS proposes for the existing services/applications six different migration strategies to-the-cloud ("*6 Rs*" (Orban, 2016))[2]:

- **Retire**: discard service/application as not bringing value to the business.
- **Retain**: do nothing and keep the existing service/application as-is on the device.
- **Rehost** (aka *lift-and-shift*): move the existing services/applications and their environment to the cloud as-is (no modifications required).
- **Repurchase** (aka *replace* or *drop-and-shop*): replace the existing service/application with a cloud-native alternative.
- **Replatform** (aka *lift-thinker-and-shift* or *revise* or *lift-and-reshape*): keep the core service/application architecture the same but refactor it to use relevant cloud features and functionality so it is better suited for the edge/cloud platforms.
- **Refactor** (aka *re-architect* or *rebuild*): redesign and rewrite the entire service/application to make it cloud-native (primary using the microservices architecture).

The above strategies can be considered for migrating on-device's services/applications to the new edge/cloud architecture—see Table 3 for possible strategies per defined TRANSACT's CPS functions types.

[2] The AWS migration strategies are extending the Gartner's "5 Rs" approach (Watson, 2010).

Table 3: Migration strategies to be considered per the TRANSACT functions types

| TRANSACT FUNCTION TYPE | MIGRATION STRATEGY | REMARKS |
|---|---|---|
| Safety-critical | Can be moved to the edge tier?<br>THEN *Rehost* or *Replatform*<br>ELSE: *Retain* | Depending on domain/workflow and feasibility |
| Mission-critical | Start with *Rehost*,<br>THEN consider *Replatform*<br>(THEN, if it brings values: *Refactor*) | Refactor step depends on cost vs value of that step. |
| Non-critical | Start with *Rehost* or *Replatform*<br>THEN *Refactor* or *Repurchase* | Refactor/Repurches step depends on cost vs value of that step. |
| Value added | Start with *Replatform*<br>THEN *Refactor* or *Repurchase* | Refactor/Repurches step depends on cost vs value of that step. |

The proposed strategies in Table 3 are initial recommendations that should be evaluated for their effectiveness and the best business value per function type taking into consideration the currently used technologies. In addition, the *Safety, Performance and Security Monitoring Services* responsible for monitoring, detecting, and preventing safety, security and performance failures likely will be impacted by the chosen migration strategy.

#### 3.2.1.3 System security

Moving to the edge/cloud deployment significantly impacts the overall security of the new architecture that requires thorough re-evaluation or redesign of offloaded services to cover new challenges. The access to the system, auditing how it is being used and by whom, and ensuring secure data transmission across the tiers are the key topics to address for the transition. As those topics are critical, the TRANSACT architecture proposes to include in the architecture: the *identity and access* services (to manage secure access to the system functionality), the *auditing* services (to collect information about accessing and using the system in by authenticated users and in an unauthorized way), and the *(federated) data services and comms* services (helping in efficient and secured data handling, both, in transit and at rest).

The techniques helping in securing the system are extensively presented in deliverable D3.2 (Pop & et.al., 2022), and deliverable D3.4 (Kirichenko & et.al., 2022). Additionally, Section 3.4.3 provides details about security as the cross-cutting aspect in the transition process.

#### 3.2.1.4 System updates

Another area that requires attention for transition is updating the system distributed over device-edge-cloud. To achieve safe and predictable system updates the following core services have been identified: the *remote update client* (running on the device) and the *update coordinator* (running at the edge/cloud). Those services cooperate across tiers to perform remote automatic updates of the different device services in a secure and safe way. Each update activity is coordinated with the *operational mode coordinator* service to keep the system in the safe state at any time The updates have to ensure uniform software versions on the tiers and keeps the system services up-to-date with the latest functionality. In addition, the design of the automatic updates should allow rolling-out a new functionality or introduce new value-added services minimizing system downtime.

Deliverable D4.2 (Mortier & et.al., 2022) provides extensive coverage of the strategies for continuous updating and independent releasing of services in distributed CPS architecture that can be considered during the migration.

#### 3.2.1.5 Observability-by-design

Observability is the key capability that is needed in the new distributed CPS solution. The TRANSACT reference architecture includes dedicated *safety, performance and security monitoring services*, which are responsible not only for monitoring of the current system status but also for detecting and preventing safety, security and performance failures. However, to be effective those services require telemetry information from all the relevant systems' services and applications. Telemetry information refers to data emitted (typically in the form of *traces*, *metrics*, and *logs*) about the service/application behaviour or system state. Specifically:

- **Metrics** provide time-based numerical measurements, time-series, on aspects of the application or system. An example of the application-level metric is the number of users using an application or the number of successful payments; an example of the system-level metric is a resource usage, such as CPU, memory, disk, network bandwidth.
- **Logs** are timestamped (structured or unstructured) messages emitted by a component of the system (a service or an application). An example of a log is an information about the system state or its behaviour useful for troubleshooting.
- **Traces** provide an understanding of how requests/transactions flow between system's functions/ components/ services/ applications. An example of a trace is the information helping to build a graph showing the involved components/services in the payment transaction for a specific user.

The above telemetry data types are complementary and when used together can provide insight into the system health and performance. For example, a trace can tell which part of a service is slow, but metrics and logs are needed to explain why. However, the key to effective system observability is the *correlation* between all the telemetry data so they can be combined and analysed together helping pinpoint the contributing factors when analysing the failures (especially issues involving services and applications distributed over various tiers).

Having system with properly *designed-in observability* capacities facilitates not only effective fault analysis and debugging but also:

- It helps in better understanding of the system run-time behaviour (through dynamic inference of service dependencies and their lifetime).
- It enables tracking actual performance against specified performance budgets, helps to identify the performance bottlenecks or the root causes of a performance anomaly across the three tiers that may impact the safety, performance, or security of the system;
- It is an input for alerting about abnormal system states;
- It is an input for the operational mode managers/coordinators to trigger mode changes based on monitored data;
- It is a prerequisite for model learning and calibration, services auto-scaling, system health assessment.

However, for the distributed system deployed across device-edge-cloud continuum build-in-observability of the product itself is not sufficient due to the shared responsibility model of edge/cloud tiers. Specifically, by using the edge/cloud platforms there is no control or access to the physical data centers, networking infrastructure, underlying virtual machines, etc. So, it is important to ensure that the metrics, logs, and traces provided by the edge/cloud vendors can be incorporated into the system observability design. In addition, the architecture needs to address a challenge how to connect and coordinate the monitoring services across

the tiers. Using open standards, that guarantees interoperability of shared data, is an important factor to consider while addressing the observability. For example, OpenTelemetry (OpenTelemetry, 2023) is an open standard that can be considered as a solution. It is available in many programming languages and has wide industry support and growing adoption. It is a collection of tools, APIs, and SDKs that are used to instrument, generate, collect, and export telemetry data (metrics, logs, and traces) to help in software's performance and behaviour analysis (OpenTelemetry, 2023).

There are several aspects to consider for the effective monitoring system as its power depends on the *observability signals* available from the application itself and the underlying platform infrastructure, the *monitoring tooling* able to collect the observability signals from many sources, the efficient *storage* of the collected observability signals, and effective *visualisation* tools to quickly understand the system runtime status. Also, when coordinating the exchange of telemetry data the architecture has to ensure that such communication is done in secure and privacy compliant way as some data may be sensitive.

Deliverable D3.3 (Nasri & et.al, 2022) explores the trade-off and provides guidelines about the observability solution selection.

### 3.2.2 Product Development

Migrating safety-critical CPS to the solution that is distributed over device-edge-cloud continuum will also impact the way how the product is developed. The aspects that need attention are: CI/CD pipeline, testing approach, source code version management.

#### 3.2.2.1 CI/CD pipeline impact

The CI/CD pipelines help to minimize bottlenecks and optimize the software development and delivery process by automatic and frequent integration of changes and their testing so the product can be ready for release. A typical CI/CD pipeline consists of the following stages: the build stage, a number of the tests stages, and the deploy stage ( (Humble & Farley, 2010)). When moving to the edge/cloud based architecture each CI/CD state should be assessed for the impact on the new system development, i.e.:

- the *build stage* can be impacted by
  - the code repositories setup, i.e., the new architecture may require building additional code repositories for the edge/cloud applications which may result that, e.g., *new* applications code is stored in the Git-based repository but the existing applications code is stored in the SVN-based repository;
  - new technologies used for creation of the system artifacts, e.g., using containers instead of pure binaries;
  - using new compilers for additional programming languages used to create the edge/cloud services/applications.
- the *test stage* can be impacted by:
  - the new tools needed to be integrated for efficient testing of services deployed directly or as containers or as serverless functions.
  - the scalability testing or load testing may require new tools and additional test infrastructure.
- the *deploy stage* can be impacted by:
  - adding new deployments environments (in which the new solution should be tested) that require new tools, scripts, and integration with the cloud platforms, especially, in terms of cloud resource provisioning;
  - how the artefacts should be prepared for release or deployed in production;
  - embrace infrastructure-as-code principles to automate deployment environments (re-)creation.

In addition, it could be beneficial to switch to the CI/CD pipeline tooling that allows smooth integration with the cloud platform of choice.

#### 3.2.2.2 Source code version management

The technologies used for the new edge/cloud services may impact the way how the source code version control needs to be arranged. Therefore, it has to be assessed and decided how to arrange in the new architecture the code repositories. Specifically which repository patterns to work with, such as:

- Single repository (aka *monorepo*) approach: *all* the source code is located in *a single* repository. It has advantage of simple build, development (e.g., shared library changes are atomically applied to all dependent services), and dependency verification for shared libraries. However, the disadvantage is that the system builds may take long if there is no proper dependency management in place. Also it may be more difficult to release independent services/applications. In this approach, there is a single repository in which next to the existing device-based sources, the new services and applications' code is stored.
- Multiple repositories (aka *multirepo*) approach: the source code is distributed over *multiple* repositories as per defined scope, e.g., a single service or a single application. The advantage of such approach is very quick system builds of the individual assets (i.e., only those that has been changed) to be ready for testing and release. Downside is more complex CI/CD pipeline and difficulty of applying changes that span multiple repositories. In this approach, the existing device-based sources can be stored in one repo and the new services and applications have their own individual repos.

Decision which approach to choose depends on the size, current technology and tooling used in development of the current CPS solution.

#### 3.2.2.3 Testing approach

Testing is another area impacted by the transformation. Typically, the tests are organized in a pyramid with unit/integration/end-to-end/manual tests. Depending on the chosen architecture the transition to the distributed device-edge-cloud solution will impact at least the integration and end-to-end tests. Those tests need to be revised and setup differently depending on which tier the services/applications are deployed or how the existing or new services would be integrated and interacting with each other's.

In addition, new types of testing may be required for delivering edge/cloud services, such as: functional *service* tests, *consumer-level* tests, *API* tests, etc. (Clemson, 2014) or non-functional tests, such as the *scalability* tests or the *load* tests.

### 3.2.3 Product Operation

Product operation is another part that is impacted by the transition to the distributed solution, as the product operation is concerned with deploying and monitoring the delivered solution in the device-edge-cloud continuum. Therefore, the current (device-focused) operation team needs to expand their capabilities to also cover the edge and cloud concepts and technologies to ensure most efficient deployment and support of the new edge/cloud solution. In general, in the new system architecture the product operation team needs to focus on the following aspects:

- product deployment: ensure that the infrastructure can support the new solution;
- product maintenance/update: ensure that the product works with the new version with minimum downtown;

- product runtime monitoring: ensure that the product works as expected and fulfils agreed SLAs and SLOs;
- product runtime optimization: using monitoring to optimize the infrastructure resources usage to minimize the operational costs.

#### 3.2.3.1 Product deployment

Before the new device-edge-cloud solution can be deployed into production the operation team needs to perform upfront planning for all needed edge and cloud resources, and select the most efficient configuration for the compute, storage, and networking infrastructure and related services. One of the critical resources to configure is the identity and access management (IAM) service with defined security groups, roles and policies that control the authentication/authorization to the infrastructure's resources ensuring secure access.

After the infrastructure is prepared then the solution is deployed. Since the new migrated solution spans over the device-edge-cloud tiers the deployment strategy may be different for each of these tiers with some commonalities. However, creating a common deployment strategy can be challenging due to the difference in the tiers infrastructure, their functional capabilities, non-functional constraints, and different tools available—deliverable D4.2 (Mortier & et.al., 2022) provides details about the possible approaches.

#### 3.2.3.2 Product maintenance/update

Software updates can be used to fix security vulnerabilities, improve system performance by offloading tasks between different architecture tiers, or adapt the system to the execution environment. However, an automatic deployments on safety-critical real-time architectures involves additional concerns that are not commonly addressed by traditional software update solutions. For example, how to guarantee the execution of critical real-time tasks that cannot be interrupted during the update process, or how to avoid memory overhead problems during the software updates on devices with severe resource constraints.

Therefore, similar to the initial deployment, the update (and upgrade) scenarios require good preparation and well thought rollout strategy. Such an update strategy takes into account the impact on the system and how it affects the users. Ideally, a zero-downtime deployment is desirable, but it depends if the system design allows it without impacting safety/critical functionality during the update roll-out. Typical patterns involving the system updates are: the *big bang update* (all changes are rolled out to all users and systems in one go) and the *phased update* (rolling out changes in an incremental manner to systems or users).

The cyber-critical CPS systems are typically complex and provide safety functionality therefore any updates should be accompanied with good disaster recovery plan in case an update fails. In this context the big bang update approach is not recommended as it imposes challenges in case of rolling-back the changes of the distributed solution over the installed tiers. Typically, most updates follow the phased approaches, such as, the blue-green deployment updates, rolling updates, or canary updates (see also (Mortier & et.al., 2022)). In case of the distributed CPS system the updates can be orchestrated differently per tier or delivered functionality depending on the resources and safety constraints of the updated solution.

Deliverable D4.2 (Mortier & et.al., 2022) provides detailed patterns and strategies of updating safety-critical CPS systems that can be considered when preparing the transition to the new distributed architecture.

#### 3.2.3.3 Product runtime monitoring

Product runtime monitoring concerns ongoing observation of the system workloads and comprehensive, real-time reporting of all the current state of the services and applications, and all the incurred costs. The operation team uses monitoring to ensure the SLAs and SLOs are fulfilled and help to address any failures of the system. However, in the distributed solution spanning over edge/cloud makes the monitoring challenging because, in the shared responsibility model, the operation team does not have control or access to the

edge/cloud platforms physical infrastructure (e.g., networks) or underlying virtual machines—it can observe only their status via available edge/cloud platforms monitoring signals (e.g., logs, traces, and metrices). As a result, the operation team derives the system status based on the combination of the infrastructure monitoring and application monitoring capabilities. Therefore, to equip the operation team in the most valuable information for their activities the application monitoring is a critical element of the new distributed solution and it should be part of the overall product design (see Section 3.2.1.5, "Observability-by-design" for more details).

Having application and infrastructure monitoring combined helps the operation teams to facilitates insights and advance system behaviour analysis that enhances searching and investigation capabilities. In addition, using the AI and ML-based services can aid operation team in data analysis and alerting of undesirable situation or uncover new failure-situations not seen earlier. However, to build coherent runtime monitoring capabilities over the device-edge-cloud based solution is a challenging task where not only the services and applications functionality is observed BUT also their performance, safety, and security status. The SIRENA platform (described in (Hendriks & et.al., 2022) and (Pop & et.al., 2022)) can helps monitoring the distributed solution—it uses a real-time machine-learning technologies for detecting safety, security, and privacy anomalies in the industrial applications by monitoring operation and network behaviour. Such a tool automates detection of variations and issue the alerts that operation team can assess for further actions.

#### 3.2.3.4 Product runtime optimization

Product runtime optimization focuses on the learning from the system performance (through the runtime monitoring) and incurred costs to improve resource utilization by applying manual or automatic corrections and enhancements. Optimization could include resizing the instances, switching to different resource type (e.g., from CPU optimized to GPU optimized), shutting down unused instances, or scaling up/down services based on the new workflows characteristics. Typically the cloud providers offer monitoring services that analyse the resources configuration, their utilization and related costs, and provide feedback and recommendations how to reduce the cost and improve the performance of the workloads (for example, see AWS Compute Optimizer (Amazon AWS-Compute Optimizer, 2023), the AWS Cost Management (Amazon AWS-Cost Management, 2023)).

The AI and ML services can also help to further optimize the infrastructure utilization runtime environment and lowering the costs of the solution (see also (Wikipedia-AIOps, 2023)).

However, not all optimizations are possible due to the solution architecture or design of the individual services or applications. Therefore, next to optimizing the infrastructure usage the learnings can be looped back to the development teams so the next versions of the services and applications enables improved infrastructure usage and lower the cost of the end solution.

### 3.2.4 DevOps

Traditional solution creation approaches have strict separation of roles like development, operation, quality engineering, and security, that may lead to inefficiencies and consequently in the delays of the new product releases. DevOps is an approach to software product development aiming to improve the speed and efficiency of software development and product release process from start to finish. Specifically, *DevOps* is a set of methodologies, practices, and technologies that allow development and operations teams (see Section 3.2.2 and Section 3.2.3) to work together to streamline product development in order to reduce the time it takes to deliver software updates and features to users. It combines the elements of CI/CD software development practices (*Dev*) and the product operation practices (*Ops*). The DevOps approach emphasizes the collaboration between the development teams and the operations teams helping to remove silos between teams, which often leads to delays and bottlenecks. Therefore, the DevOps approach emphasizes rapid and effective response to the *production issues* as identified by the operation team and can effectively

use CI/CD pipeline to address them, and finally, DevOps is about improving the speed, quality, and efficiency of software delivery to production—Figure 5 shows all major activities covered by DevOps.

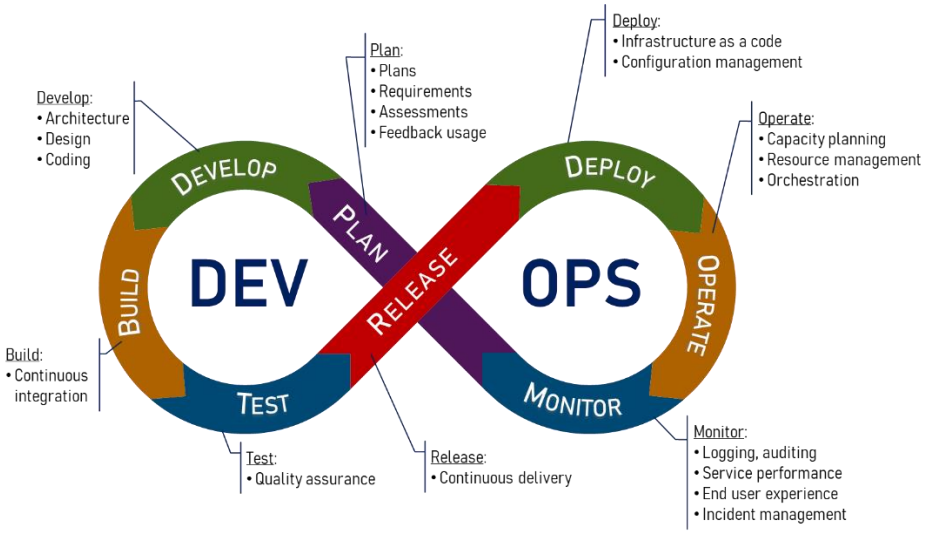


Figure 5: DevOps main phases

The main DevOps phases are:

- Plan (dev pipeline): it covers planning the project, related technologies, environments, structure, and architecture based on the requirements and feedback from the stakeholders and customers.
- Development (dev pipeline): it covers developing the product including its architecture, design, and coding.
- Build (dev pipeline): it covers creating the releasable assets ready for testing and bug fixes in the development test environments.
- Test (dev pipeline): it covers all the required testing to ensure the product assets are functioning as expected.
- Release (dev pipeline): it covers all required actions to ensure that the assets are ready to be deployed in production.
- Deploy (operational pipeline): it covers deploying the released product assets into the production environment.
- Operate (operational pipeline): it covers activities ensuring that the deployed product is running as expected in production.
- Monitor (operational pipeline): it covers collecting data and providing analytics about the product performance and customer behaviour, errors and more. This stage is directly related to feedback loop into the development organization.

In the new distributed CPS solution the operation team will play significant role in the product lifecycle, especially, due to complexity of the new deployment over the three tiers, more complex product design (based on mixture of new and old technologies), and new concerns in the area of safety, performance, and security. Therefore, DevOps practices are recommended as they can help to address the above concerns.

There are enhancements proposed to the DevOps approach in various non-functional areas, such as safety, security or performance making it relevant for the new CPS development as well. Specifically, the DevOps approach aims on more effective and efficient realization of the cross-functional requirements by "*shifting left*", i.e., making the quality concerns (safety, security, performance, …) part of the regular development activities (left part of Figure 5). Specific enhancements to the DevOps development approach for the safety

related activities is presented in Section 3.4.1.4, for the performance related activities is presented in Section 3.4.2.4, for the security and privacy related activities is presented in Section 3.4.3.4, and for the regulatory activities is presented in Section 3.4.4.3.

### 3.2.5 Infrastructure

To support new device-edge-cloud continuum solution architecture the *product development infrastructure* and *product deployment infrastructure* need to be adapted as well. In general, the new solution architecture will depend on the infrastructure and SW services that are *not* own by the organization, but depend on the infrastructure of the cloud provider (in case of IaaS deployment) or the cloud specific services and solutions of specific cloud platform (in case of PaaS deployment).

The *product development infrastructure* covers all the development environments and tooling used to build and test the product assets. By moving to the cloud, the new solution incorporates more loosely-coupled services and components which may require different development tools and CI infrastructure that supports virtualization and containerisation for building and testing new assets. In addition, the CI infrastructure may incorporate cloud environments to deploy the integrated services and applications in the environment that is as-close-as-possible to the production environment to get higher confidence of correct quality of the released assets. This infrastructure is used by the development teams (see also Section 3.2.2).

The *product deployment infrastructure* covers the CD environment and tooling used to deploy and monitor the running solution over the device-edge-cloud continuum (including safety-critical and mission-critical functions). The deployment infrastructure on which these functions are running, needs to provide capabilities to enable configurations that ensures safety, performance, and security of the whole solution. The product deployment infrastructure is typically based on the cloud platforms provided infrastructure that helps to build high-performing and scalable distributed applications and provides tools to dynamically adjust the applications' resources to accommodate high or low demanding workflows. However, only proper usage of provided platform's resources, services, and tools can help to design and build the end system that fulfils the needed scalability and performance requirements. Therefore, the product deployment infrastructure has to ensure predictable and scalable configuration options to support active online management and scaling of heterogeneous resources without jeopardizing performance and specific SLAs and SLOs attached to the running applications and services. This infrastructure is used by the operation teams (see also Section 3.2.3).

When organization considers adopting the DevOps model (as presented in Section 3.2.4) then the development and deployment infrastructures should be tightly integrated for efficient and optimized product deployment, updates, and operation. However, it is not a prerequisite for the edge/cloud transition to start.

## 3.3 Transformation area: Organization

The organization that onboards new cloud-based development and related technologies needs to account for:

- people development: to enable better understanding and efficient usage of new edge/cloud-technologies,
- adaptation of the way-of-working processes: to optimally perform activities required to create the new edge/cloud solution,
- investment in the infrastructure enabling efficient development, deployment, and operation of the edge/cloud-based CPS solution,
- changes in the organization setup to optimally serve the new edge/cloud-based solutions.

These topics are important for the transition as they accelerate realization of the new solution architecture by building successful teams working optimally in the organization structure dedicated for the edge/cloud

solution creation. The following sections focus on each of those topics as the next step in the transition to the distributed CPS solution—see Figure 6.

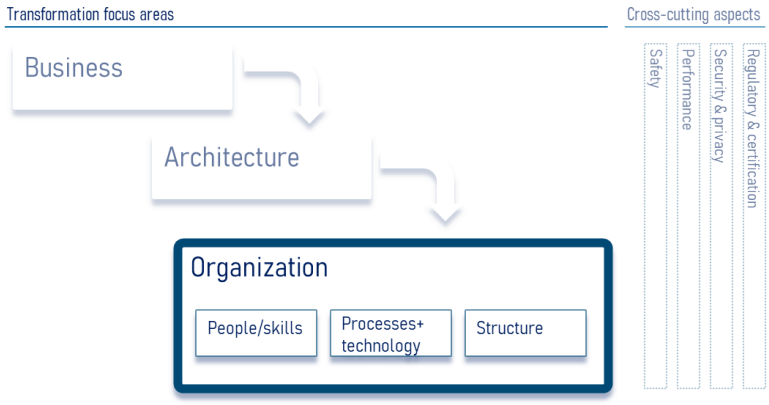


Figure 6: TRANSACT transition: Organization Transformation Area elements

### 3.3.1 People/skills

The organizational personnel impacted by the new edge/cloud-based architecture involve many teams with different competences, for example:

- the development teams: need to use new (edge/cloud) technologies and concepts to build the distributed solution,
- the operational teams: need new tools and approaches to deploy and monitor the new solution across three tiers: device, edge, and cloud,
- the marketing teams: need to adapt to the new offering by understanding the landscape of the new edge/cloud-based solution,
- the legal teams: need to ensure that new solution adheres to the applicable laws and regulations,
- the financial team: need to adapt new way of billing for the system functionality usage.

The edge/cloud-based system architecture requires investment, not only in the organization's personnel but also in the customer to clarify the benefit and added value of the new approach. Therefore, development of training programs for the system end users are needed to clarify the new system services, conditions of use, its capabilities and limitations. For example, in Use Case 5 (see Section 4.5.2), it is necessary to train end users such as plant managers, middle managers, and other IT department profiles to adopt the new technologies used throughout the wastewater treatment processes.

### 3.3.2 Processes and technology

The transformation of a system architecture usually involves upgrading and introducing new technologies either to support new services or to improve and optimize the organizational structure of the company. The technological changes (such as upgrades to the IT infrastructure or new development/deployment tools), are one of the organizational issues that can have the greatest impact on successful transformation to the edge/cloud-based architecture.

Therefore, the selection of new technologies demands detailed research to ensure that the requirements of the organization are met. The IT department has an important role to contribute with the study and evaluation of critical aspects before implementing these new technologies. For example, the impact on the organization, information security vulnerabilities, investment costs, vendor lock-in, and learning curve are

some of the factors that must be analyzed to design efficient change management plans and strategies. In this way, reduce the uncertainty generated by technological change that impacts the teams and even end users.

Next to the technology updates also the way of working processes need adaptation to better serve the new edge/cloud-based solutions. The adaptation of way of working may result with new operational models, e.g., when adapting the DevOps approach (see Section 3.2.4) then the development and operational teams are collaborating very closely. Another example is the risk assessment process that needs to accommodate security and privacy concerns from transmitting and storing the customer data in the cloud. The billing processes may need adaptation due to different ways of calculating the cost of the services provided.

### 3.3.3 Organization structure

While changing the way of working processes to better fit the edge/cloud solution creation it may require changing the organization structure as well to support those processes. Organizational structure changes may involve the creation of new departmental units, changes in the chain of command, redesign of the work structure, modification of responsibilities, and other types of structural aspects. Transforming a stand-alone CPS into a safe and secure distributed safety-critical CPS solution may require structural changes in the following areas:

- Teams restructuring: people-driven organizational change aims at reorganizing departments, work teams, roles, and responsibilities to improve efficiency in managing the new processes and technologies implemented. Based on services and applications defined by the architecture, the teams can be reorganized to better map the ownership of delivered artefacts.
- Departments restructuring: adding new responsibilities to existing departments to cover new areas of expertise. For example,
  - the billing department needs new tools and way of handling edge/cloud services usage by the customer,
  - the legal privacy department needs additional expertise on handling personal data in the cloud to comply with required privacy laws and regulations,
  - the customer service organization, initially focused on the device only installation, needs to expand their capabilities to cover also the applications and services deployed in the edge and cloud platforms as part of the overall solution (the organization structure will be also influenced by the chosen operational model (Microsoft, 2023) , i.e., decentralized operations, centralized operations, or hybrid operations model may lead to changes of the needed engineering skills: field service- engineers vs remote -support engineers).

## 3.4 Transformation cross-cutting aspects

Next to the main transformation areas (Business, Architecture, and Organization) there are cross-cutting aspects that impact many activities in those areas. The main aspects for transforming safety-critical CSP to device-edge-cloud continuum are: safety, performance, security and privacy, and regulatory and certification. Each of these aspects is presented in the following sections by primarily showing their impact on the product design, development, and operation and, when relevant, impact on the business and organization areas—see Figure 7. Since the DevOps approach improves the product releases (see Section 3.2.4) the possible impact on that approach by each cross-cutting aspects is explicitly explored.

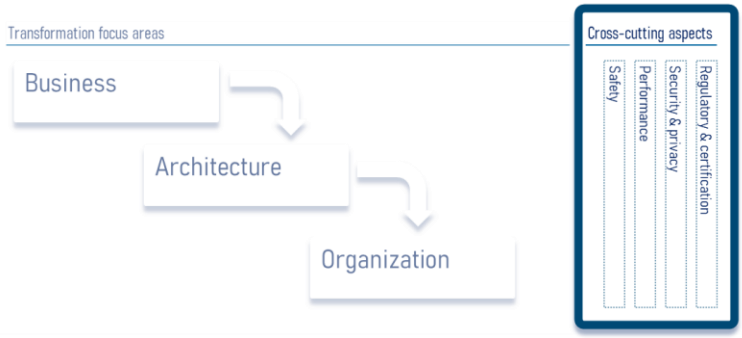


Figure 7: TRANSACT transition: Cross-cutting aspects

### 3.4.1 Cross-cutting aspects: Safety

Safety is the most critical aspect of the safety-critical CPS products, because failure or malfunction of such a system may result in death or serious injury to people, loss or severe damage to equipment/property, or environmental harm. The product's safety is a combination of the safe product design and the underlying infrastructure capabilities to allow safe usage of the product—the safety-critical functions may be potentially affected by edge and cloud functionality. Therefore, when migrating to the distributed CPS architecture the safety should be addressed during the product design and during product operation. The main safety related topics to pay attention to while migrating the CPS products are:

- Safety risk analysis,
- Safety design patterns in distributed systems,
- Edge/cloud infrastructure runtime guarantees,
- Real time safety monitoring,
- Development process improvements following DevOps approach with the safety aspects.

Deliverables D3.1 (Hendriks & et.al., 2022) and D3.3 (Nasri & et.al, 2022) provide details about safety and its impact on the distributed CPS when following the TRANSACT Reference Architecture.

#### 3.4.1.1 Safety risk analysis

Safety engineering (Safety Engineering, 2022) is concerned with assuring that engineered systems provide acceptable levels of safety. Traditional methods include failure mode and effects analysis (FMEA), fault tree analysis (FTA), and Bow Tie analysis (Ferdous, Khan, Sadiq, Amyotte, & Veitch, 2013). However, those methods primarily look at the failure modes for each piece, part, or component of the system but when there is undesired interaction between (correctly behaving) components, or in specific scenarios, which can cause unsafe situations other methods are more suited. Therefore, when migrating the safety-critical CPS to distributed architecture the following methods can help identifying the safety concerns at the design time:

- System Theoretic Process Analysis (STPA) (Leveson, 2016)—it is a hazard analysis technique which not only includes component failures but also considers accidents that can be caused by unsafe interactions of system components, none of which may have failed.
- Identification and Quantification of Hazardous Scenarios—it is an integrated method for safety assessment of automated driving functions which covers the aspects of functional safety and safety of the intended functionality, including identification and quantification of hazardous scenarios. The

method is tailored to support existing safety processes mandated by the standards ISO 26262[ (ISO, 2011)] and ISO/DIS 21448 [ (ISO, Under development)] and complements them where necessary.

- The MAGERIT[3] method for risk assessment—it is a standard that establishes principles for the effective, efficient, and acceptable use of IT—it helps organisations balance the risks and encouraging opportunities arising from the use of IT. It has been prepared by the CSAE (Spanish Higher Council of E-Government (HIGHER COUNCIL FOR ELECTRONIC GOVERNMENT)) and it is recommended in Europe by ENISA (the European Union Agency for Cybersecurity) (MAGERIT, 2005).

Deliverables D3.1 (Hendriks & et.al., 2022) and D3.3 (Nasri & et.al, 2022) provide detailed description of the safety risk analysis methods and their fit with the distrusted safety-critical CPS based on the TRANSACT Reference Architecture.

#### 3.4.1.2 Safety–by–design

There is a broad diversity of safety tactics, principles, patterns, and techniques discussed in the literature. This demonstrates that there is no single overall solution that improves safety while keeping complexity and costs in check. Typically, a carefully deliberated selection of strategies, principles, patterns, and techniques is combined to meet the system and safety requirements as well as the overall system constraints. The specific safety tactics (such as: failure avoidance, failure detection, failure containment) and patterns that can be applied in the context of distributed safety-critical CPS system are elaborated in deliverable D2.1 (Arjona & et.al., 2022).

#### 3.4.1.3 Edge/cloud infrastructure runtime guarantees

Knowing the safety risk of the device-edge-cloud solution then provisions shall be taken to safeguard the safety of the system operation covering not only the device but also underlying edge/cloud infrastructure. The special attention should be given during the transition to the real-time computing and dependable communication guarantees of the edge/cloud infrastructure.

**Real-time computing**

The real-time systems are systems whose functions have timing requirements that must be satisfied to guarantee correct and safe operation. In many cases, this involves ensuring that a certain computation is guaranteed to be completed within a given deadline (typically in the order of microseconds of milliseconds) with (very) high probability. To enable real-time computing, hardware must be appropriately selected in each tier. While hardware in the device tier is generally appropriately selected for the needs of the specific solution, this is less likely to be the case for the edge tier (which may support multiple (types of) systems) and even to the lesser extend for the cloud tier (which may be operated by a commercial provider and support systems from different organizations and domains). This limitation affects the decision which functions can be offloaded from the device: functions with stringent timing requirements that must always be satisfied are unlikely to be offloaded, while the functions with less strict requirements that may occasionally be violated can be offloaded.

One solution could be hypervisors allowing usage of the real-time operating systems. Such an approach can be used to create robust partitions that spatially and temporally isolate functions (of the same or different criticality) from each other, as necessary to fulfil the safety requirements. In addition, it may be relevant to use isolation to separate mission-critical functions in the cloud from third-party applications and services downloaded from the marketplace.

[3] Methodology of Analysis and Management of Risks of Information Systems

When safety-critical and mission-critical functions are offloaded from constrained to more powerful devices, not only real-time computing concepts are required, but also the performance of data transmission between the tiers contributes to a successful (i.e., safe) execution of the task. Therefore, for safety-critical CPS, special emphasis should be placed on ensuring dependable (wired or wireless) communication, i.e., data exchange.

**Dependable communication**

Dependable communication describes the reliability, timeliness, and availability of data transmission, i.e., it aims to minimize packet loss, latencies, and energy consumption. Dependable communication is required between each of the three tiers in the TRANSACT Reference Architecture. Each connection (i.e., device-edge, edge-cloud, and device-edge-cloud) will have more rigorous end-to-end requirements depending on the use case: for example, when offloading safety-critical functions from the device to the edge, this connection has more stringent requirements than sending data for non-critical functions from the edge to the cloud. Because of the application-dependent requirements, several technologies may be used for data exchange, and therefore different concepts should be combined to ensure dependable data transmission.

The following concepts can be considered to address the end-to-end dependability requirements:

- *Monitoring and adaptation*: Link monitoring and measurements can be used to get an estimation of the link quality (e.g., observing packet reception rate and latencies). Based on the link quality, the communication can be adapted, for example, by changing specific protocol parameters or using redundancy techniques to guarantee timely and reliable data transmission.
- *Time-triggered transmission*: it allows devices to exchange data in reserved time slots and therefore guarantee reliable communication. Such approaches, however, typically require precise time synchronization.
- *Synchronous transmission*: it is a special concept for wireless communication, which exploits the capture effect and constructive interference to effectively flood information in a network. Flooding-based solutions were shown to be reliable, while minimizing end-to-end latency and being energy-efficient (Zimmerling, Mottola, & Santini, Synchronous transmissions in low-power wireless: A survey of communication protocols and network services, 2020).

Deliverables D3.1 (Hendriks & et.al., 2022) and D3.3 (Nasri & et.al, 2022) provide detailed description of the safety concepts for edge/cloud platforms and their fit with the distrusted safety-critical CPS based on the TRANSACT Reference Architecture.

#### 3.4.1.4 DevOps pipeline with the safety related activities

Safety is a system quality that emerges through the life cycle of a system. If organization is building software following the DevOps approach (see Section 3.2.4), then it would be beneficial to consider during the transition to include specific safety-related activities at every stage of the DevOps process. In addition, to increase the speed of releasing, it is crucial to automate the safety assessment pipeline through adaptation of model-based techniques (Nasri & et.al, 2022).

Figure 8 shows possible safety activities that can be added to the DevOps pipeline stages.

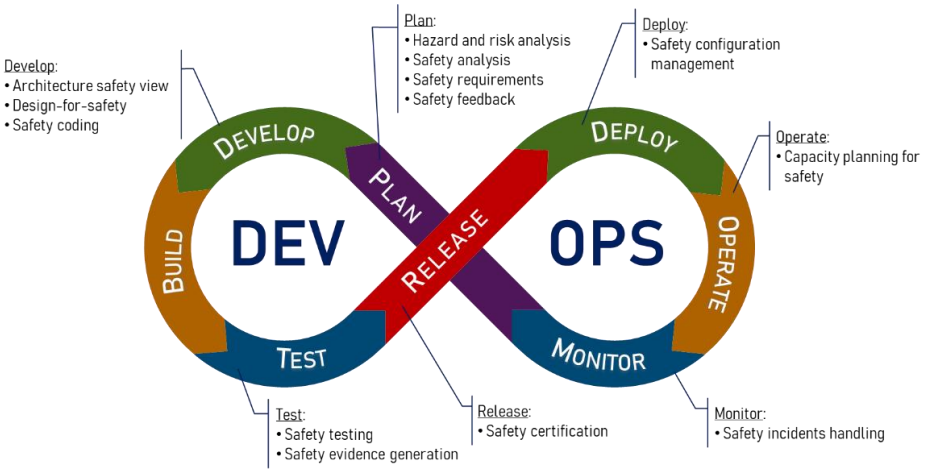


Figure 8: DevOps pipeline with the safety engineering activities

### 3.4.2 Cross-cutting aspects: Performance

Performance is another critical aspect of the safety-critical CPS systems, because it supports system safety-critical functions and enables other functionality successful and timely execution, i.e., in the real-time-based workflows, timing performance is an integral aspect of the functional behaviour of the system, whereas, in safety-critical workflows, the performance may be essential to the safety of a system. Therefore, when migrating a safety-critical CPS to the distributed edge/cloud architecture with ensuring required or optimal performance is extremely challenging as it is influenced by the interaction of many components that are physically distributed, often heterogeneous, and not necessarily subject to a single point of control. In addition, the overall performance of such a distributed system does not only depend on the product services and components but in part on the underlying edge/cloud platform infrastructure services. Therefore, when migrating to the distributed CPS architecture the performance should be addressed during the product design and during product operation. The main performance related topic to pay attention to while migrating the CPS products are:

- Performance impact analysis,
- Performance–by–design,
- Runtime performance monitoring,
- Development process improvements following DevOps approach with the performance aspects.

#### 3.4.2.1 Performance impact analysis

Performance is critical for many aspects of the CPS systems. The performance requirements and the results of the safety risk analysis (see Section 3.4.1.1) are the bases for assessment of the performance impact on the new distributed edge/cloud based CPS system. Specifically, the impact on the system design and runtime system operation has to be analysed to ensure meeting the business objectives (especially due to deployment in the edge and cloud infrastructure). In addition, the edge/cloud technological solutions, tools, and concepts needs to be explored and selected to guarantee safety and predictable performance of the new solution.

The explicit activity focusing on the overall system performance architecture helps to determine the performance aspects that need to be taken into account at the start of the transition process and during the system life cycle.

#### 3.4.2.2 Performance–by–design

When migrating a safety-critical CPS to the distributed edge/cloud architecture then ensuring required or optimal performance is extremely challenging. The reason is that distributed system performance is influenced not only by the system's services and application but also by the underlying edge/cloud platform's infrastructure services. The edge/cloud resources are shared and managed in unpredictable ways, therefore, the system design need to have provision to avoid impact performance required by the system workflows.

To address the performance challenge of the migrated distributed CPS system the TRANSACT proposes to follow the Model Driven System Performance Engineering (MD-SysPE) methodology (see (Hendriks & et.al., 2022)). This methodology embraces modelling formalisms, methods, techniques, and industrial practices to design for performance (Sanden, et al., 2021). MD-SysPE defines the following focus areas:

- *Performance architecting* to determine the performance aspects that need to be considered at the start of the development process and during the system life cycle.
- *Design-space exploration* to explore the trade-offs and find optimal designs within a given system architecture.
- *Performance modelling and analysis* to express and analyse the performance of specific system configurations.
- *Scheduling and supervisory control* to achieve the required performance during system operation.
- *Data-driven analysis and design* to enable model learning, model validation and model calibration.

MD-SysPE covers the complete system lifecycle, from the system design (addressing performance requirements and objective) till the system operation (addressing optimal runtime performance). However, the key aspect of this methodology is emphasis on using the feedback from system operation, i.e., the operational data may be used to improve the installed system performance at runtime and through system updates. In addition, it serves as valuable input for the development of next generations of the systems architectures.

For distributed CPS system deployed on the device-edge-cloud continuum the correct performance is difficult to predict due to unpredictability of the edge/cloud environments in which the applications are running. The performance modelling (as part of the *Performance modelling and analysis* MD-SysPE phase) can help to predict performance qualities of a system based on the system settings such as resource allocation, quality settings of an application or its operational modes. The results from the system modelling will guide the system design to ensure that all the relevant parameters and configurations are available in the product.

To model the CPS system performance TRANSACT proposes to use the *Y-chart paradigm* (see Figure 9). The Y-chart paradigm (Hendriks, Basten, Verriet, Brassé, & Somers, 2016) (Kienhuis, Deprettere, Vissers, & Wolf, 1997) (Lapalme, et al., 2009) proposes to model application functionality and the implementation platform as separate elements, with an explicit mapping as variation point between them. This allows easy variation of application functionality, platform resources, and mapping choices and facilitates analysing the performance impact of these choices, forming a convenient basis for (automated) design-space exploration to systematically explore design alternatives around these variation points. The Y-chart modelling can be combined with numerous performance modelling approaches such as data flow, timed automata, stochastic processes, queuing networks, discrete-event simulation, and the machine learning approaches. As a result, the most appropriate approach can be used to model different CPS characteristics, support different properties to be analysed, and with different degrees of accuracy and efficiency.

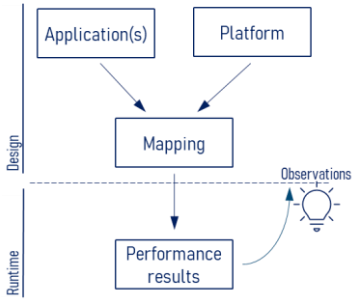


Figure 9: Y-chart based performance modelling

The performance modelling is also an input for design of the observability signals (see Section 3.2.1.5 "Observability-by-design") that will be used in the feedback loop to learn about the actual system behaviour and the model accuracy (the "Performance results" block in Figure 9). The observability signals are important ingredients in system operation because they are the base for the performance monitoring and run-time performance management which help ensuring optimal performance of the system workflows and provide important feedback to the design-performance models updates.

### 3.4.2.3 Runtime performance management

After system is deployed across device-edge-cloud continuum it is critical to ensure keeping the system's workflows performing as expected. The operating environment of distributed CPS system relies in significant part on the underlying edge/cloud infrastructure which enforces increases the need for dynamic adaptation to its context and environment to guarantee the workflows operational performance. The self-adaptation techniques applicable in the design of CPSs often use the well-established MAKE-K model (Kephart & Chess, 2003). This model defines the four phases that an adaptive system performs: *Monitor-Analyse-Plan-Execute* and the one cross-cutting concept: *Knowledge*, that the system has about itself and its context (see Figure 10, (Hendriks & et.al., 2022), and (Nasri & et.al, 2022)).

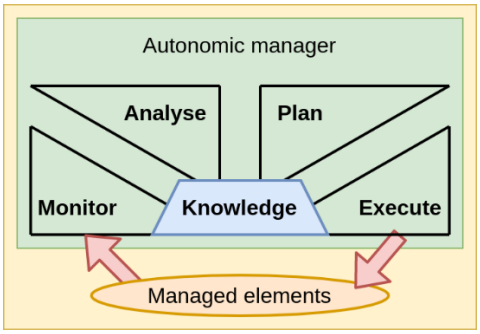


Figure 10: MAPE-K autonomic loop (from (Kephart & Chess, 2003))

The performance monitoring is critical part of the MAKE-K cycle as it provides observations about the temporal behaviour of a system, taking advantage of the statistics provided by performance monitors (Valente, et al., 2021). These monitors are watchpoints that collect designed-in system's metrics or events (see also Section 3.2.1.5 "Observability-by-design"), typically characterized by the time they occurred, metric values the type of event, and any additional attributes required to describe it. The captured events are filtered or pre-processed for storage or further transmitted to be used by other components.

A number of components that consume monitored signals are part of the performance management. The goal of performance management is to influence the relevant aspects of performance of an application to adjust to run-time situations and to stay within limits of performance requirements. Management strategies

may include adjusting resource allocation or quality settings of an application, shaping, or balancing of workload. Performance management may involve continuous adjustment of system settings, but also reconfiguration of operational modes of the system or application, for instance placing functions at different tiers, such as edge or cloud, in which case the reconfiguration process itself may be subject to performance constraints.

Detailed strategies for performance monitoring and management are presented in (Hendriks & et.al., 2022) and (Nasri & et.al, 2022) together with selected tooling that can help to realize them.

#### 3.4.2.4 DevOps pipeline with the performance activities

If organization is building software following the DevOps approach (see Section 3.2.4), then it would be beneficial to consider during the transition to include specific performance-related activities at every stage of the DevOps process. Figure 11 shows possible mapping of performance engineering activities to the DevOps pipeline stages.

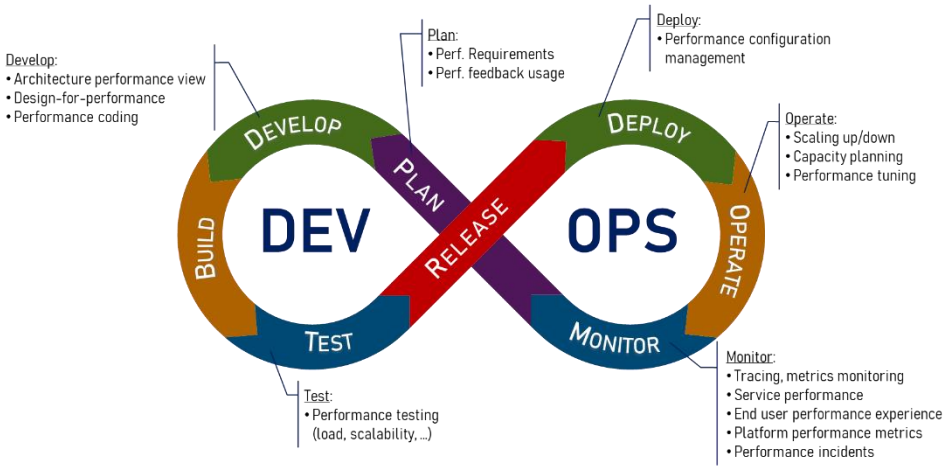


Figure 11: DevOps pipeline with the performance engineering activities

Detailed description of the performance practices in development workflow is presented in deliverables D3.1 (Hendriks & et.al., 2022) and D3.3 (Nasri & et.al, 2022).

### 3.4.3 Cross-cutting aspects: Security and privacy

By moving safety-critical CPS architecture away from the centralized, on-device solution toward the distributed, cloud-based architecture significantly increases the attack surface of the new solution by making it more vulnerable for security attacks. Also, the data privacy concerns are growing significantly in such an architecture as the user data, especially in automotive and healthcare domains, is highly sensitive and require special care not to be exposed due to being transfer over a public network or due to security attacks and software vulnerabilities. The security and privacy related topics to pay attention to while migrating the CPS products are:

- Security and privacy risk analysis,
- Security and privacy–by–design,
- Runtime security monitoring,
- Development process improvements following DevOps approach with the security and privacy aspects.

#### 3.4.3.1 Security and privacy risk analysis

Since the CPS products and the data they process may be exposed to the edge and cloud infrastructure it is essential to perform security and privacy risk analysis to understand the impact on the product design and its operation. The product security risk assessment helps to determine the security weaknesses of the products at the early stages of the product development process. It helps to identify, communicate and understand product/solution threats and vulnerabilities and identify additional countermeasures and operational controls to be implemented during the design and development phases. Typically, the security risk assessment focuses on Identifying the security risks and their likelihood and impact in order to identify the risk mitigations and, if not designed-out, the way how to manage them in the field. The privacy risk assessment in principle may follow the same steps as the security risk assessment. In essence, the privacy threats can be identified by leveraging the threat model created during the security risk assessment, including the way of scoring the risks' likelihood, impact, and the identified mitigation.

The security and privacy risk assessments will be more extensive due to exposure of the current system to the edge/cloud environments and also because it interact with wide range of cloud services and applications. As part of the system transition it is advantageous to start with the current system security and privacy risks and ensure that the new architecture maintains the same or lower risk level as the old one. To develop a security risk analysis, the following approaches can be used:

- the MAGERIT methodology: it can be used for security risk assessment in a similar way as for safety analysis presented in Section 3.4.1.1;
- GConsulting tool. it is based on the international standard on how to manage information security ISO/IEC 27001. GConsulting takes into account the threats identified by MAGERIT and allows to include certificate-based security solutions. It can monitor and centralise all the information related to the risk security assessment.

The details of MAGERIT, GConsulting and other approaches helpful in performing the security risk analysis are presented in (Pop & et.al., 2022) and (Kirichenko & et.al., 2022).

#### 3.4.3.2 Security– and privacy–by–design

The security and privacy aspects span all the product deployment tiers (device-edge-cloud) and all the product layers (from the core services layer to the application layer, see Figure 4: TRANSACT Reference Architecture). Therefore, after the security and privacy risks for the new edge/cloud architecture are clear then the next step in the transition is to ensure that the edge/cloud CPS system design ensures that the risks of security breaches and privacy violations are minimized. The aspects that should be taken into account for the product design are:

- Data confidentiality/integrity/availability (CIA[4]): all the current controls have to be re-evaluated in the context of the edge/cloud exposure of the data (in the TRANSACT Reference Architecture CIA is addressed by the *Data Services and Communication* services);
- Identity and access control: the mechanisms for the data protection, such as user authentication, authorization, and role-based access control, have to be enhanced to securely cover the edge/cloud

[4] *Confidentiality controls* ensure that only the right/authorized users/services can use the system and its data, while prevents sensitive information from reaching wrong/unauthorized users/services. *Integrity controls* ensure consistency, accuracy, and trustworthiness of data. *Availability controls* ensure that system data and resources are available to authorized users when need.

as well (in the TRANSACT Reference Architecture the identity and access control is addressed by the *Identity and access* services);

- Accountability: it serves two main purposes: to provide information about user activities in relation to the data (e.g., who accessed the data, when, and what action performed), and information helping to identify potential security incidents (that may impact the data). The accountability is critical to meet the regulatory requirements for some domains (e.g., healthcare domain) (in the TRANSACT Reference Architecture accountability is addressed by the *Auditing* services).

There are well known best security practices and approaches (known as the security patterns) that should be considered while migrating to the edge/cloud deployment. In general, the *security patterns* are a broad set of solutions that address specific security problems by controlling (stopping or mitigating) a set of specific threats through a dedicated security mechanism defined in a given context (Fernandez, 2013). However, unlike, the design patterns, that are primarily used during the system design and development, the security patterns cover also deployment aspects and security-processes enhancing system development, deployment, and operation. In general, applying security patterns should help to proactively adopt the security measures already during system design and development and this way (by design) ensuring more secure system deployment and its operation. Therefore, due to CPS inherent complexity and safety nature using the security and privacy pattern during the system development can strengthen the core security principles around user/patient data confidentiality, integrity, and availability. In addition, they can also improve handling of user identity, services and data access control, and accountability for the performed actions on the system. The broad set of the security and privacy patterns is presented in (Arjona & et.al., 2022) and many security and privacy design solutions targeting distributed safety CPS are presented in (Pop & et.al., 2022) and (Kirichenko & et.al., 2022).

#### 3.4.3.2.1 Identity and access control

To maintain confidentiality, integrity, and availability of any information system, it is necessary to implement security rules or policies that restrict the behaviour of all system users. Access control policies can be implemented as an effective cybersecurity strategy ensuring proper guarding of the critical system functions (safety or performance) and essential system or user's data.

One of the security strategies to address this concern is Role Based Access Control (RBAC). RBAC allows users to be restricted in the use of the system according to their role. Each role can have a set of permissions or authorizations to access, modify, or manage resources and services. RBAC is one of the key mechanisms widely implemented in cloud environments (Li, 2015) but using it in the context of distributed CPS system is little explored.

When migrating to distributed CPS deployment the user permissions need to be explicitly designed and managed across all three tier (device, edge, and cloud) which may be a major concern depending on number of services and components deployed on each tier. In addition, if the number of users is high and dynamic, the authorization granting and revocation operations can grow, making it difficult to manage. To overcome those challenges, the Domain Specific Language (DSL) can be defined covering the main concepts enabling the modelling of RBAC for the distributed CPS architecture. Such DSL would be a specification of business/design level policies to grant/deny access to system resources. Details of such solution is presented in (Kirichenko & et.al., 2022).

### 3.4.3.3 Runtime security monitoring

Security monitoring is the key aspect of the safety CPS in order to detect and then properly respond to the security and privacy incidents. The security posture of the distributed system comes from the properly addressed security and privacy risks: 1) in the product (by security-in-depth design and privacy-in-depth

design), 2) in the underlying infrastructure, and 3) in the security incident-handling processes. In other words, securing cloud-based services is a shared responsibility of the product builder and the cloud provider.

The main role of the security monitoring is helping to detect the security breaches in order to take proper corrective actions to handle the security incident. Embracing the edge/cloud as part of the new architecture significantly widens the security area therefore combining cloud platform security tools with the observability signals of the product is critical to successful defence of the product security posture. From security perspective detecting infrastructure security issues and detecting anomalous user behaviour are import subject to consider in the transformation security analysis.

The cloud infrastructure security monitoring tools can be divided into two categories–compliance-based tools that inspect the current state of the infrastructure against rulesets (such as Cloud Security Posture Management), and real-time tools that monitor the log flow or control plane activity of the infrastructure. All cloud providers offer native security tools that often have capabilities from both categories, for example, AWS GuardDuty (Amazon AWS, 2023) can detect both compliance violations and suspicious API activity.

Another tool helping monitoring the security and privacy posture of the distributed solution is the SIRENA tool (Kirichenko & et.al., 2022). It uses a real-time machine-learning technologies for detecting safety, security, and privacy anomalies in the industrial applications by monitoring operation and network behaviour (using Nozomi Networks solutions as the market leader for industrial cybersecurity monitoring technology). It can also distinguish between legitimate use and a malicious attack. In addition, the SIRENA platform can link the monitoring output with the security risk analysis done with the GConsulting tool (see Section 3.4.3.1) enriching the original probe information with the business information, therefore, better manage the security risks. The SIRENA tool is being explored by the Spanish railway company (RENFE) in Madrit, by the Valencian's harbour under Port 4.0 project, and by four hospitals in Generalitat Valenciana.

Detecting anomalous user behaviour is another challenge to be considered during the transition due to the wide variety of user behavioural patterns that might exist. There are solutions based on rules or heuristics but they lack the flexibility and accuracy necessary to capture the large number of possible user behaviours. Additionally, these solutions are not scalable, since manually creating and maintaining a set of rules for each user requires huge effort. Alternative method to detect anomalous user behaviour is the User and Entity Behavioural Analytics (UEBA) approach based on the behavioural machine learning models. Those models represent the behaviour of a large user base using, so called, user profiles. Given a sample of user-generated activity, these models can learn how to discriminate between anomalous and normal behaviour of a user or any relevant entity in the system under analysis, such as processes, endpoints, and IoT devices. Deliverable D3.4 (Kirichenko & et.al., 2022) gives more detailed explanation of the tool and its applicability in the security and privacy context.

#### 3.4.3.4 DevOps pipeline with the security and privacy activities

The software development approach that takes security into consideration is the DevSecOps approach. *DevSecOps* is an extension of the DevOps approach (see Section 3.2.4) by including specific security and privacy activities at every stage of the software development process, from design, coding, through testing to deployment—this way the organizations can reduce very early the likelihood of introducing security vulnerabilities into their products. Since the security is one of the critical factors when considering move from on-device to the edge/cloud distributed CPS architecture, therefore it is a good approach to consider when migrating to the new distributed CPS solutions.

The possible security and privacy activities that can be added to the DevOps stages are shown in Figure 12.

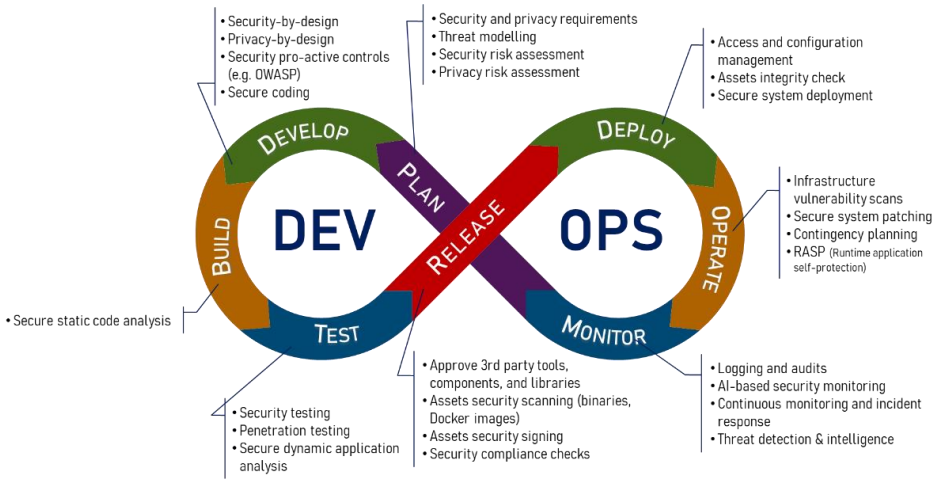


Figure 12: DevOps pipeline with the security and privacy activities

Detailed description of the secure software development practices in development workflow is presented in deliverable D3.4 (Kirichenko & et.al., 2022).

#### 3.4.3.5 Security and privacy impact on organization

Next to the product impact, as presented in Section 3.2.3, the security and privacy aspects may impact the organization as well by requiring adaptation of the business and organizational processes to cover security and privacy practices (see DevSecOps in Section 3.4.3.4). In addition, it may be needed to create a dedicated security and privacy team with required capabilities to address the new edge/cloud security and privacy challenges.

### 3.4.4 Cross-cutting aspects: Regulatory and certification

Regulatory and (re-)certification is another cross-cutting aspect that impacts, not only the product, but also the business and organization itself. Business leaders need to decide which certification to obtain and which new regulations the business needs to comply with. This decision will impact the way how the product is developed (new design rules may be imposed on the development process or the verification and validation process), how the business is organized (there could be additional teams needed to ensure security and privacy compliance) or what new skills need to be acquired in the organization. The safety-critical CPS systems based on the edge/cloud architecture are primary impacted by the safety and security/privacy regulations.

#### 3.4.4.1 Regulatory impact analysis

Regulatory impact analysis should be done from the business perspective to ensure that new distributed solution still comply with all the needed regulations. Since the distributed solution will span over the edge/cloud platform it is critical to ensure that the used edge/cloud platforms have required certifications needed in the domain. For example, lacking compliance with healthcare privacy regulations, such as GDPR (GDPR, 2016) or HIPAA (HIPPA, 1996) , may be a stopper for usage of such platforms. Conversely, selecting the cloud platform that provides key domain capabilities and certification (e.g., healthcare) can lower the risk and costs, and enabling development of new value--added solutions that require minimal compliance work.

Another set of constraints from regulatory perspective can be impact on the system customer that needs to be analysed and managed as well, for example, it may be needed to obtain the customer's consent to move their data to the edge/cloud infrastructure, if the customer has additional regulatory requirement that needs to be taken into account for the product, etc.

Finally, it has to be analysed how the re-certification should be executed for the new distributed solution and which changes in the organization are required to be effective.

#### 3.4.4.2 Regulatory design impact

The inputs from the regulatory analysis will impose additional constraints on the product design and architecture to ensure compliance. Moreover, chosen regulatory compliance framework may impact the development processes (see next section).

#### 3.4.4.3 DevOps pipeline with the regulatory activities

The safety-critical systems need to create safety assurance evidence that proves the risks are managed and therefore acceptable for the new product. When developing safety related electronic and programmable control systems, there are a number of sector specific standards and regulations that need to be considered, e.g., EU's Medical Device Regulation (UNION, 5 April 2017), IEC 60601 (IEC, 2005), IEC 82304 (IEC, Health software - Part 1: General requirements for product safety , 2016), IEC 61508 (IEC, IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010), ISO 26262 (ISO, 2009), and RTCA DO-178B (RTCA, 1992).

The conceptual basis for certification is that the evidence anticipates the possible circumstances that can arise from the interaction between the system and the environment, to show that these interactions do not pose an unacceptable risk. Several ARTEMIS and ECSEL projects have addressed safety assurance challenges, related, e.g., to modular, incremental and re-certification (Martin, November 2018), for example, CHESS (Cicchetti, et al., 2012), CONCERTO, SAFECER (Mälardalen University, 2023), AMASS (AMASS, Project Deliverables, 2017) and SafeCOP (SafeCOP, 2023). The assurance approach in those projects relies on model driven methodology for the design, verification and implementation of Cyber-Physical Systems, where the components are annotated using assumption-guarantee contracts (Albert Benveniste, 2012) to facilitate the independent development of cooperative safety functions. Those methodologies could be considered while migrating to device-edge-cloud continuum architecture and ultimately make it part of the DevOps activities with the regulatory activities focus.

The DevOps with the regulatory activities can be considered as generalization of the DevOps pipeline with the safety activities (see Section 3.4.1.4), but having a broader scope covering not only safety but also security, privacy and other relevant regulations for a specific domain.

The possible regulatory activities that can be added to the DevOps stages are shown in Figure 13.

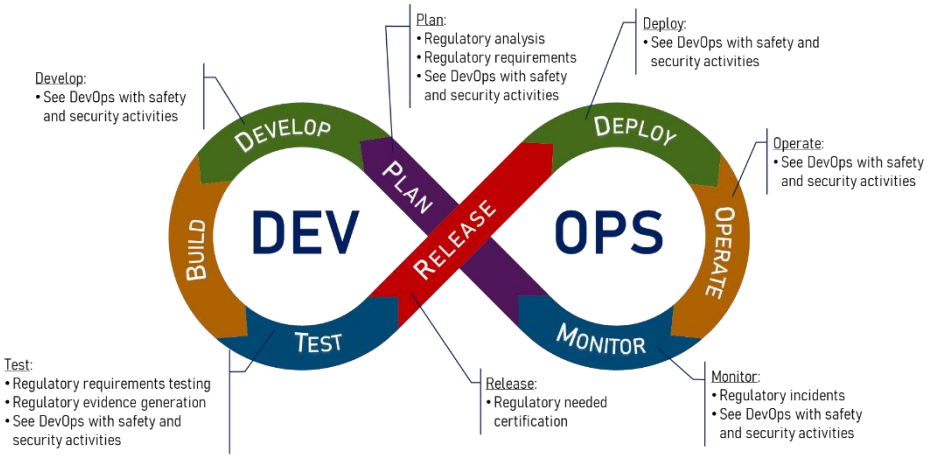


Figure 13: DevOps pipeline with the regulatory activities

More information about the regulatory development practices in development processes is presented in deliverable D3.3 (Nasri & et.al, 2022) and D3.4 (Kirichenko & et.al., 2022).

#### 3.4.4.4 Organization impact

The way of working may also be impacted if new certifications imposes additional requirements on organization or the product development practices. The ISO/IEC 27000 series is a well-known family of standards that introduces a concept of an Information Security Management System (ISMS). The ISMS is a centrally managed framework consisting of policies and procedures to manage information systematically in a secure way using a risk-based approach. It covers many organization aspects, such as processes, people, and IT systems. The ISO27002 framework provides best-practice guidance on applying the controls defined in ISO27001. When determining which security controls should be selected and implemented, a risk-based approach should be followed.

## 3.5 Planning and execution of the transition

Transformation of the device-based safety-critical CPS system to the distributed safety-critical CPS solution can be split into four phases:

- Create a business case: demonstrate why there are benefits from moving to the edge/cloud solution including the consequences of such move for the business, the solution architecture, and the organization;
- Create an architecture: create a new distributed safety-critical CPS architecture with classifying each function as safety-critical, mission-critical, or non-critical together with the migration type (see Section 3.2.1.2) and assigning each function to the relevant tier (device, edge, or cloud)—this should help to define the functions migration order and its effort;
- Plan transition: define the steps to execute the transformation covering all the relevant topics, such as, organizational changes, regulatory (re-)evaluation, the order of workflows/functions migration;
- Execute transition: realize the transformation as per defined plan;
- Optimize: make efficiency improvements based on the workflows monitoring in the production environment.

Except the first step, the rest of the steps can be executed iteratively bringing the incremental value to the organization and making the transition gradual and more controlled, so the risks of wrong steps are minimized. This process will be reassessed based on the use-case transition experience in the next version (V2) of this document.

## 3.6 Summary

The transition to the distributed CPS solution raises several concerns and challenges that must be addressed by the business, by the new product architecture, and by the organization composition to effectively deliver new edge/cloud-based solutions. The advantages of edge and cloud computing are numerous. However, before engaging into the transformation of current on-device CPS system to the distributed CPS solution it is essential to build a valid business case that clarifies the benefit of such a move. Only having clarity of the benefits and feasible realization prediction, the next steps should be taken, i.e., assess impact on the architecture, organization, relevant processes, and finally on the customers themselves.

The TRANSACT Reference Architecture is taken as blueprint for the aspects to be concerned-with while performing the transition. Therefore, as outlined by the TRANSACT transition methodology, next to the business, architecture, and organization areas, there are critical cross cutting aspects (safety, performance,

security, privacy, regulatory) that need to be considered for successful transition to the edge/cloud deployment. Those aspects, require deep analysis to fully grasp the impact of the changes.

The next section presents the transition experience of the TRANSACT project's use-cases towards the distributed CPS solution based on the TRANSACT Reference Architecture.

# 4 Transition of use cases to TRANSACT transition methodology

This section describes the steps taken and highlights the strategies for transforming a local CPS to the device-edge-cloud continuum following the TRANSACT Reference Architecture. It covers for each use-case the following aspects as initially evaluated:

- Transition to Transact Reference Architecture,
- Organizational changes to support transition to the reference architecture,
- Planning and execution of the transition, and
- Lessons learned from the transition so far.

The above aspects are presented per each use-case in the following sections:

- Section 4.1: Transition of Use Case 1: Remote operations of autonomous vehicles for navigating in urban context
- Section 4.2: Transition of Use Case 2: Critical maritime decision support enhanced by distributed, AI enhanced edge and cloud solutions
- Section 4.3: Transition of Use Case 3: Cloud-featured battery management system
- Section 4.4: Transition of Use Case 4: Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic imaging systems
- Section 4.5: Transition of Use Case 5: Critical wastewater treatment decision support enhanced by distributed, AI enhanced edge and cloud solutions.

## 4.1 Transition of Use Case 1: Remote operations of autonomous vehicles for navigating in urban context

In this use case, Fleetonomy and partners DENSO, Nodeon, Nunsys, Singlar Innovacion, ViNotion and VTT will develop a solution for remote control of (semi-) automated vehicles for navigating in urban environments (see Figure 14). The solution will allow vehicles to be moved from one location to another even without a driver, but with a remote operator. The operator will receive continuous feedback on vehicle state and environment, allowing him/her to assist the vehicle to navigate through urban traffic. The vehicle will have autonomy provided by current state-of-the-art automated driving solutions taking care of normal driving, and capable of detecting and reacting to arising hazardous situations.

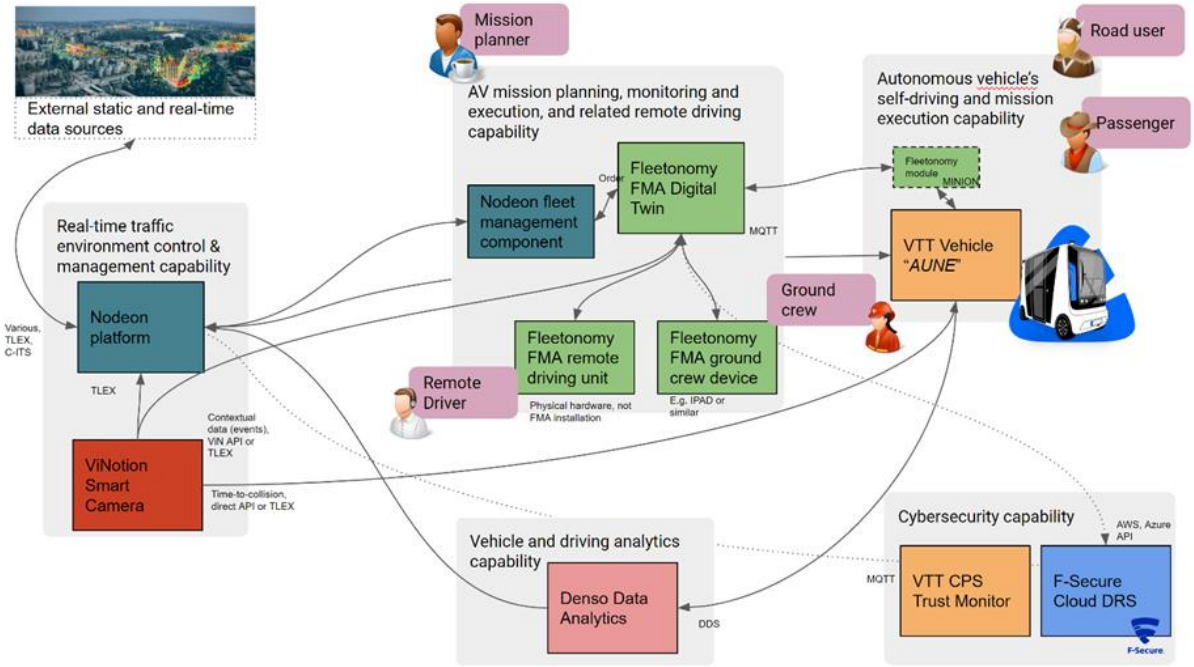


Figure 14: The remote operations use case cloud-edge-device continuum

Use Case 1 showcases a unique solution for remote fleet management of remote vehicles. Fleets operate in high-quality Digital Twin environments, which increases the situational awareness of the remote driver leading to better understanding and safety. Seamless data integration with various external systems occurs so it can be used by one operational Digital Twin. The Fleet Management Application is vehicle-provider and vehicle-type neutral, i.e., vehicles from various manufacturers, and even various types (like UGVs, AGVs, UAVs/drones etc.) can be controlled with one system.

Key challenges have been noted as follows:

- Traffic is a complex environment, ranging from the technical properties of the vehicles to traffic regulation to other external inputs (such as weather, risk of collisions, other road users etc.). For a distributed cyber-physical system, the transition challenge is to manage the data flows in the device-edge-tier continuum, balance the workload of different systems and ensuring temporal correctness of the data.
- Autonomous operation capabilities are increasing, but human intervention is needed regularly. For validation and verification process Software-in-the-loop and Hardware-in-the-loop is no longer enough, as Human-in-the-loop must be introduced too.
- In order to be efficient, human intervention must be done remotely
- Use Case 1 demonstrates very challenging real-time remote operation over high-capacity networks. Challenges related to network bandwidth with function transitions of a CPS are mostly with data flows, lag times, glass-to-glass latency and reaction times of the personnel.
- UC1 works on integration from all sides of the challenge, from vehicle side to roadside units to remote fleet management
- Remote fleet management needs to be (vehicle) technology neutral to work with real-life fleet deployments
- There is an over-arching transition challenge in Use Case 1 with shifting any Safety-Critical Functions from Device Tier to Cloud Tier. Mostly those functions, which concern vehicle, operational personnel and road-side users are difficult to remove from the Device Tier e.g., the autonomous vehicle or roadside sensors, without impacting road safety.

### 4.1.1 Transition to TRANSACT Reference Architecture

The transition to TRANSACT reference architecture specific to UC1 is shown in Figure 15.

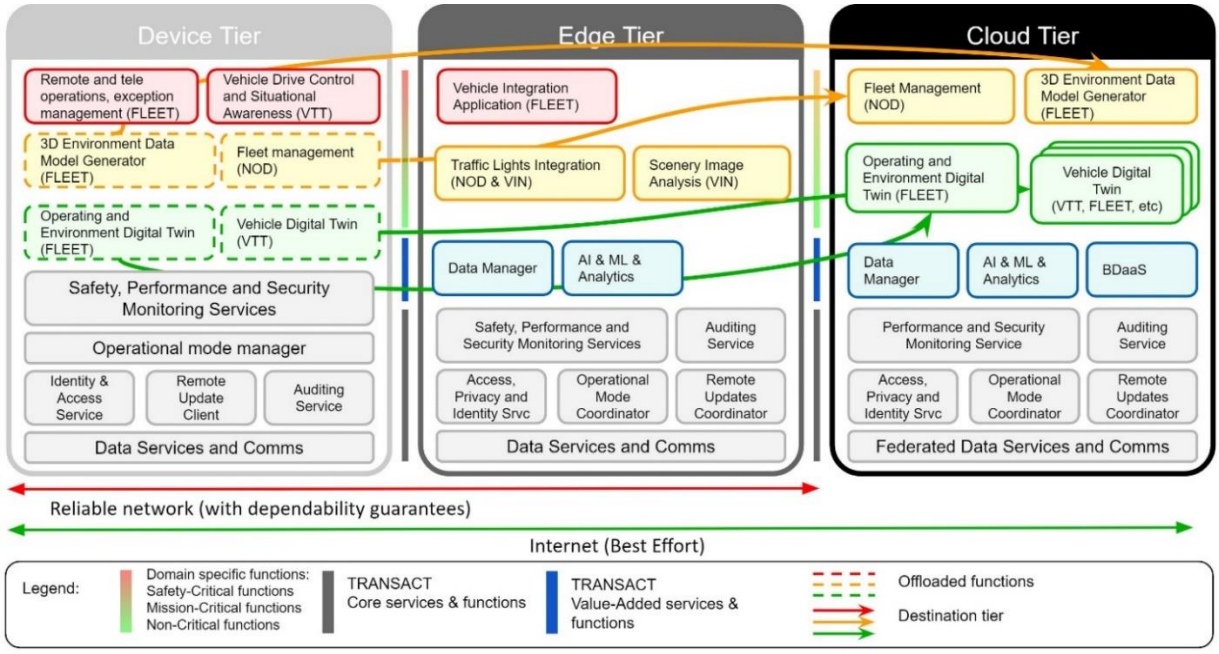


Figure 15 The transition to TRANSACT reference architecture specific to UC1

Access control and we are also raising this to become safety critical and exploring the possibility to shift the service to the cloud tier. Use Case 1 is evaluating a service that would allow us to implement a “Zero Trust” identity management service that we can both manage in the cloud and we can cache in any operational location. The identity service needs to be accessible regardless of the data connectivity, increase in safety, topic level access control and ease of use to be used effectively. Also, using for example a Zero Trust process and system to check the Remote Driver’s credentials before they can take over the driving of the vehicle can be deployed over a cloud service rather than giving directly access to the vehicle (device). Use Case 1 is developing other scenarios similar to this practice, but it cannot be conclusively determined at the time of reporting whether they are feasible to implement within project scope or schedule.

Out of the Mission-Critical Functions the 3D environment data model generator and the Fleet Management system for planning missions and routes to the vehicle can be moved from device to cloud tier. Both functions and services can be deployed from a cloud hosting service instead of residing on vehicle (e.g., device tier) companion computer to balance the workload and free up resources for sensor data processing and other on-board vehicle functions.

At the reporting time two major Non-Critical Functions can be shifted again with a load-balancing and bandwidth considerations. Data collected by vehicle sensors needs to be passed on to cloud-based Operating and Environmental Digital Twin to ensure accuracy especially if more than one simulated or physical vehicles are part of the operation. Likewise, the Vehicle digital twin can be moved from device to cloud tier on a similar reasoning.

### 4.1.2 Organizational Changes to support the transition

Remote operation of automated vehicles requires significant organizational additions compared to a more traditional vehicle fleet operation. Autonomous vehicles inherently require an IT infrastructure to support various roles, technical solutions and network connectivity. Bus or taxi operator, which relies on traditional fleet of cars with in-car drivers is able to sustain an operation on the strength of its drivers and non-automated vehicles.

#### 4.1.2.1 DevOps unit

Setting up a DevOps organizational unit is needed to begin operating services. This requires hiring specialized DevOps-capable developers for the enhancement of the CI/CD processes.

Failure Prevention and fast reparation service to include:

- Reduction of manual work phases in installation
- Configuration management process enhancement
- Automated installation testing
- Operations planning, analysis, testing and implementation enhancement
- Continuous training of stakeholders and customers

#### 4.1.2.2 Remote Operating Centre

To set up the autonomous vehicle fleet operation one of the key organisational elements is the Remote Operating Centre. For convenience the Remote Operating Centre can be co-located near the actual operation area, but technically the Remote Operating Centre can be set up anywhere within good network connectivity. Personnel stationed in the Remote Operating Centre are **Remote Driver**, a person trained for remote driving of the vehicle who is on guard during the operation. The basic requirements are a secure and distraction-free office-type location where the Remote Driver's control setup can be assembled. The operational parameters dictate several variables, such as the number of Remote Drivers, including **Backup Remote Drivers,** working in shifts to cover the daily, weekly or monthly timeframe of the operation. Remote Operation Centre may need **Remote Operation Supervisors** to add a redundancy element while Remote Driver is engaged with the remote operation platform and moving vehicle(s).

#### 4.1.2.3 In-Vehicle Safety Supervisor

In-Vehicle Safety Supervisor is a new role that needs to be fulfilled. As the Safety Supervisor's main task is to ensure vehicle and road-user safety in the immediate vicinity of the vehicle while stationary and moving, they need to understand the basic requirements of the remote operation, top prioritised tasks and they need to be able to react to exceptional situations, where the vehicle needs to be stopped safely, a transfer of control needs to be initiated or the operation aborted entirely. This is a specialised role that does not exist beyond the remote and autonomous vehicle operation.

#### 4.1.2.4 Ground Crew

The Ground Crew consists of persons responsible for transferring the vehicle to and from the area of operation at the beginning and end of the vehicle's operation schedule. This involves performing security and safety checks before approving the vehicle for operation.

### 4.1.3 Planning and execution of the transition

#### 4.1.3.1 Feasibility Study

Multiple feasibility studies have been conducted similar to this Use Case in previous projects including FABULOS, SESAR "GOF" and FortumGO, all involving autonomous vehicles under fleet management control. These are used as baseline pre-TRANSACT studies so we can review how operational management and systems architecture can be amended in accordance with the TRANSACT Methodology.

#### 4.1.3.2 Safety & Security Risk Assessment

In regard to the security risk assessment for the core solution in UC1, we have worked on the model and include below some of the content from this that is pertinent to other reviewers.

This security evaluation was based on VTT researchers analysing the initial system architecture and a survey on publications concerning security of similar embedded systems and selected protocols. The attack classifications according to (Abu Daia, Ramadan, & Fayek, 2018) were considered. The Message Queueing Telemetry Transport (MQTT) is an integral part of the system and this protocol is evaluated in more detail.

As can be seen in the system architecture the Message Queueing Telemetry Transport is an essential component of the system. The architecture and environment of a system (or a part of it) can be considered as a sensor network. This point of view is taken into account by considering attacks described in Abu Daia et Al. The paper lists 22 different types of attacks possible in sensor network environments.

Our system relies on TLS in its VPN implementation (this is in line with MQTT specification). TLS itself is not reviewed (out of scope) as it can be considered secure when decent implementations are utilized and they are set up correctly.

As with any system used by humans, there might be data that can be considered someone's private personal information. The use of such information is regulated in the EU by the General Data Protection Act.

Most of the time it is easy to recognize personal information that needs to be kept private. According to GDPR personal data is any information relating to an identified or identifiable natural person. An identifiable person is anyone who can be identified – directly or indirectly – by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person. In other words, practically any information that can be connected to a person might be subject to GDPR. Location information should be considered as personal data.

Furthermore, there is a wealth of existing information specific to cyber security risk assessments within the connected autonomous vehicle (CAV) ecosystem and as such we will not look to reinvent the wheel but rather, we would use extant materials to support our case. In this instance we will refer to the EU-funded "LEVITATE" project and provide these excerpts as they are pertinent to Use Case 1:

> *A security risk analysis was conducted to identify possible cyberattacks against a future transport system consisting of autonomous and connected vehicles. Six scenarios were developed:* ***joyriding, kidnapping, domestic abuse, autopilot manipulation, a large transport accident, and paralysis of the transport system.*** *Even if it were possible to increase the difficulty of conducting such cyberattacks, it might be impossible to eliminate such attacks entirely. Measures that limit the consequences will therefore be necessary. Such measures include safety measures in vehicles to protect their occupants in traffic accidents and measures that make vehicles easier to remove in case they do not function.*

This report is available in full from the LEVITATE project and outlines the main critical elements for any security assessment. There is also a new report authored and published by the UK-based automotive cyber

security team "Copperhorse" with references for software coding standard and best practice for secure systems in connected autonomous systems (Tyrrell & Rogers, 2023).

We have also reviewed multiple reports provided via BSI setting out standards for autonomous vehicle systems and safety including Remote Operations and Assuring Operational Safety (PAS1881:2022)

Specific to safety and risk assessments, we are using the same framework methodology as used by regulators for the use of unmanned aerial vehicles (UAV) and based on our extensive experience in operations in this area. The "Specific Operations Risk Assessment" (SORA (EASA, 2023)) documentation is clearly structured and can operate as a template for ground vehicle operations. The basis of the SORA procedure is to guarantee the same level of safety for the operation as in manned aviation. The SORA procedure consists of 9 different steps, which are described below. We have extensive and practical operations experience of setting up SORA and negotiating safety principles with the Finnish governmental traffic regulatory body Traficom, so this structure will apply across the entire EU. An example structure to SORA is noted here as reference and is available in full on request:

**Specific Operations Risk Assessment – Table of Contents**

**0. PREFACE – Permits of Exception Needed**
**1. CONOPS – Concept of Operations: Technical, operational and system information needed to assess the operational risk**
  1.1. Remote Operating Platform
  1.2. Specific Model Information (Vehicle Platform)
  1.3. Control System Components
  1.4. Parameters of vehicle operation
**2. Determination of the intrinsic UAS Ground Risk Class (GRC)**
**3. Final Ground Risk Class (GRC) determination**
*4. Determination of the initial Air Risk Class (ARC)*[1]
**5. Tactical Mitigation Performance Requirement (TMPR) and Robustness Levels**
  5.1. See and avoid: VLOS / EVLOS (Visual Line Of Sight / Extended Visual Line Of Sight)
  5.2. Manual Flight / Drive and Automated Flight / Drive
  5.3. Remote Monitoring
**6. Final assignment of Specific Assurance and Integrity Level (SAIL) and Operational Safety Objectives (OSO)**
**7. Identification of the Operational Safety Objectives (OSO)**
**8. Adjacent area and airspace considerations**
**9. Comprehensive safety portfolio**
  9.1. Emergencies
  9.2. Exceptions
  9.3. Aerial and ground risk mitigations

1 Not applicable for autonomous ground vehicle operations

Whilst the above example is written for UAVs, the sections are flexible enough to be also tailored to ground vehicles and as such we see this as a viable, and industry-understood, methodology to adopt for UC1. As we progress with this work, we will update SORA to reflect the ground-based operations under UC1 in this same format and level of detail.

One of the lessons learned of using generic and EU-standardised approach is operational planning and risk management consistently becomes part of the process and is always approached in a systematic way. For autonomous vehicle operations there are always certain same variables that need to be addressed,

regardless of the place of operation, the types or number of vehicles, operational parameters or command and control systems used. It also addresses human-in-the-loop elements of the operation and requires autonomous vehicle operators to identify operational safety objectives directly affecting the whole operation.

Specific Operation Risk Assessment aims to address operational safety issues and mitigate risk and while originally intended for aerial vehicles and not always a mandatory process, its principles can be applied in almost any kind of autonomous vehicle operation.

### 4.1.4 Validation and verification

Our validation and verification (V&V) process is mature and has been tested through many projects and contexts. Fleetonomy always runs a sandbox environment to enable simulation-based checks on vehicle system so we can understand in fine detail how the vehicles are expected to behave. We most often find that manufacturer-provided information related to performance and operations in simulation can be different to real life hence the requirement to always test in the field. As part of this we therefore run full end-to-end simulation prior to field-based live tests and this remains our suggestion method of operations for teams looking to replicate UC1 in their projects.

Our V&V process has not needed to change to conform to the TRANSACT transition. Our aims are to attempt to push the majority of work to a developer role prior to needing to spend time on field tests as these are, by far, the most expensive in terms of cost and time for any project. The more we can do bug testing and integration from the developer (office based) the more focused, and economically efficient, our time is then spent in the field with live and physical tests. This is a recommendation that should fit well across all use cases.

### 4.1.5 Lessons learned

There are several lessons learned already from the reporting period, summarised below.

- When implementing a system-of-systems the lead times of development and testing cycles can be drastically reduced by having simulators available for all sub-systems. As research and development and subsequent field testing is expensive and time-consuming, the ability to simulate against functional and technical requirements is essential.
- Employing Docker-based containerisation improves simulators' maintainability and portability and enables the modular deployment of multiple vehicles into the simulation.
- Improving the situational awareness of the vehicle can be done with edge (e.g., roadside sensors) or cloud-assisted services (traffic data, weather data etc.), but for safety purposes the SAWA functionality should still reside in the vehicle. This can be viewed as much an operational as technical implementation choice regarding the overall vehicle and traffic safety.

## 4.2 Transition of Use Case 2: Critical maritime decision support enhanced by distributed, AI enhanced edge and cloud solutions

The maritime use case (UC2) will demonstrate advancements in safe and efficient maritime navigation, made possible by extending and enhancing the existing basic edge/cloud technologies in the NAVTOR e-Navigation Suite (see Figure 16). This will be achieved by integrating advisory services, AI-based services, and data-analytics services into the device-edge-cloud continuum. The demonstrators will show a more integrated and connected architecture allowing for enhanced decision support and common situational awareness for three end users: the navigator onboard ships, the operator on the shore-based bridge, and the operator of the unmanned service vessel in harbour areas.

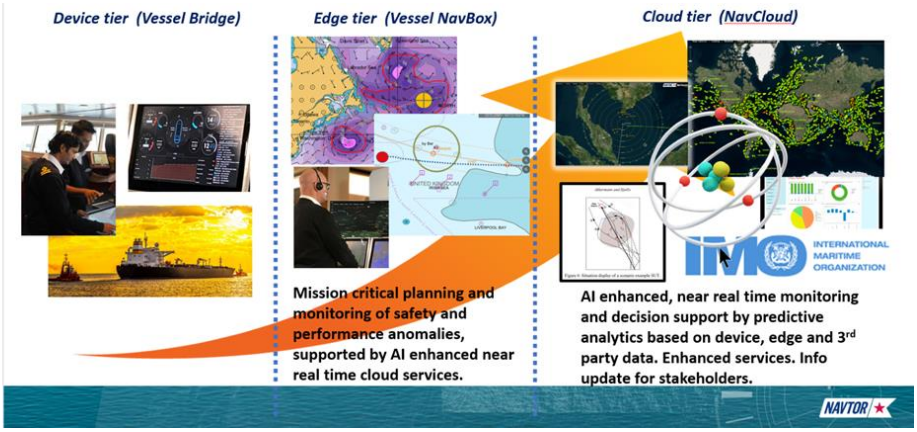


Figure 16: UC2 targeted solution in which navigators and operators are supported by edge and cloud monitoring and decision support services

UC2, the NAVTOR e-Navigation Suite will be upgraded and empowered to allow for additional information from onboard and off-device to be collected, quality checked, stored, and exchanged ship-shore-ship, either aggregated or in near real-time depending on modus of operation. Current and new information will be utilized to advance performance and shipping analytics into Sustainable Shipping both in e-navigation and performance solutions. This will allow for new business opportunities such as multiple advisory services running on the same platform as well as data exchange with third parties. Through UC2, both the planning device onboard and the web-portal onshore will increase decision data visualization, and effort will be made to create a common situational awareness on both components, allowing additional off vessel support from operators onshore to navigators onboard. Less workload for the navigator will in turn increase safety, reduce risk of grounding, and support a more sustainable, global shipping.

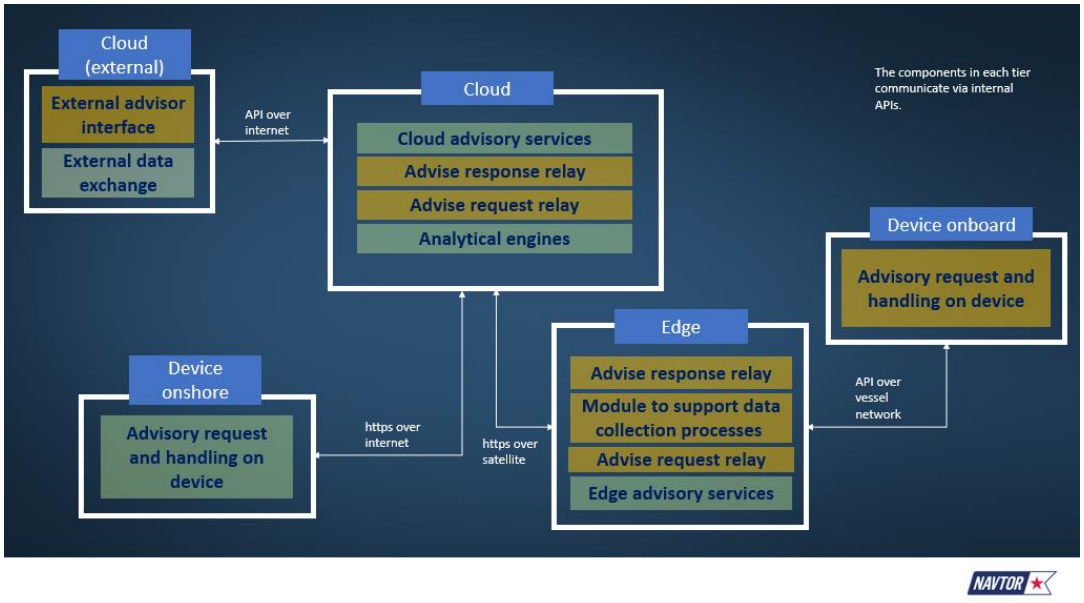


Figure 17: Overview over the advisory service framework of UC2 to be developed and implemented in the TRANSACT project.

In UC2 AI-enhanced advisory services (see Figure 17) on the edge and cloud tier will be researched by SIMULA RESEARCH LABORATORY AS. This research will focus on AI-enhanced route optimization, AI-based traffic and

congestion predictor, AI-enhanced fuel optimization and AI-based abnormal vessel behaviour detector. AI assisted decision support services will allow navigators with critical maritime decision support and operators with advice review services. Such services (that are currently not available) paving the way for faster, cheaper, and more flexible solutions tailored to customers' needs. UC2 also aims to show the AI advanced services benefitting the operators onshore, developing new services for the shore-based monitoring system sending notifications or warnings of low efficiency of their vessels.

The Cloud and edge-based computing solutions for near real-time monitoring and decision support that will be made available through UC2 will mean a breakthrough in reducing groundings and other incidents, however the need for this transformation becomes even more apparent considering environmental challenges. The shipping industry is faced with clear expectations to lower fuel consumption and current and foreseen EU regulations and IMO environmental policies shows measures that will need to be implemented to create necessary cuts in GHG emissions supporting sustainable shipping. UC2 will try to address these challenges by providing monitoring of expected and actual fuel consumption and possible measures to take to reduce fuel consumption, such as advice to alter course or speed providing better just in time arrival estimations. Today the lack of trust in just in time arrival estimations is a prominent causer of unnecessary fuel consumption.

### 4.2.1 Transition to TRANSACT Reference Architecture

The transition to TRANSACT reference architecture specific to UC2 is shown in Figure 18.

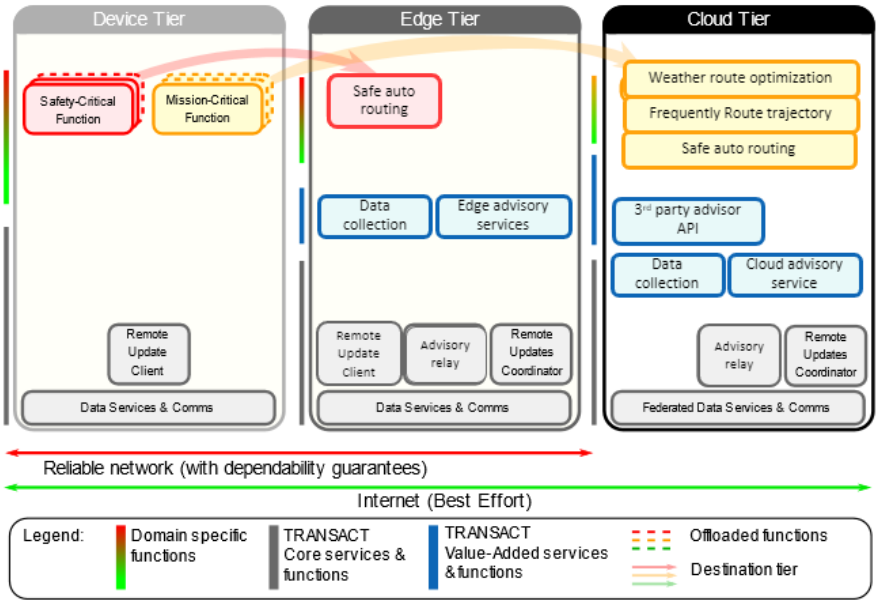


Figure 18: Representation of UC2 TRANSACT architecture mapping.

The main safety-critical equipment for navigation onboard ships is the ECDIS—an Electronic Chart Display and Information System that displays a ship's position on electronic nautical charts in real-time and provides navigators with monitoring and simple warnings in real-time. However, the ECDIS is governed by several standards and regulations, and any request for it to handle decision-support services or to process additional navigational information is both timely and costly. Navigators looking for additional decision support during navigation, will therefore need to extend the navigational equipment with an ecosystem that is able to request and handle advisory support and effectively take advantage of the increased digital information continuously becoming available in our digital age.

While evaluating the requirements identified in the project proposal an architecture for supporting advisory services utilizing each tier of the TRANSACT continuum was developed. Attention was given to support both traditional and AI-enhanced advisory services. In order to support fast innovation and release to marked the advisory service solution was designed to support generic advisory services.

In the development of advisory services safety has been given a high priority, ensuring in that the advice provided through the advisory services does not result in unsafe operations. The importance of being able to provide safe navigation advice has taken priority over faster release to marked and performance of these services. In the development of advisory services security has been defined as high, but not critical. Justification for this is that the advisory services go into planning devices and is not in direct contact with any device with operational control. This also applies to safety considerations. The ECDIS will check the suggested route regarding its safety when it is set. Therefore, there is not direct interactions with the advisory service without the user being engaged. Due to selected security requirement level the developed solution won't be integrated directly into the operation control unit onboard (ECDIS). There is no personal data being handled by the system so privacy requirements and potential trade-off concerning GDPR are non-applicable.

By enabling advisory services on remote tiers, the navigator is able to access safer navigation routes faster, than manually doing them onboard with limited outside support. In addition, advisory services can monitor abnormal vessel behaviour and provide notification for onshore and onboard personnel.

By enabling advisory service on remote tiers, the navigator is able to access mission critical monitoring services that allow better just in time arrival predictions. And non-critical services such as weather optimized advisory services.

Some of the TRANSACT Value-added services and functions that have been introduced in the use case are:

- Data manager has been introduced in UC2 and will be implemented in Y2 and Y3 of the project (NAVTOR onboard DB).
- AI/ML/Analytics is a core value-added service and function introduced to the e-navigation suite through UC2.
- The customer service marketplace has been extended to include external advisory services providers, as well as novel internal advisory services.
- Monitoring services for Safety and Performance are value added services and functions that will be developed in the advisory service framework of UC2.

### 4.2.2 Organizational changes to support the transition

There are several organizational changes necessary to support the transition. As the developer services will make information and functions available to the shore staff, several process activities can be transferred to the shore-based organization. First the transition of functionality will require an extension of the existing DevOps organizational model to support new tools and systems.

NAVTOR AS has a procedure for the introduction of new services to customers. Therefore, the introduction of new advisory services running on the edge and cloud tier will need to follow the same procedure as other services that has previously been introduced. This includes providing training and documentation to the existing customers- and technical support units within NAVTOR AS.

Monitoring tools are being implemented as part of the development of the new service framework. About Human Resources, operators are required for maintaining initial versions of the advisory services. There will be effort made in automating these processes through use of ML.

Regarding the IT infrastructure, no changes to existing processes and organization will be required. However, independent service agreement amendments need to be included for all developed advisory services that rely on 3rd party collaboration.

### 4.2.3 Planning and execution of the transition

At the beginning of the project partners in the maritime use case (UC2) began the use case specification process, based on a user driven methodology. First, they identified high-level end user requirements (EURs) (Szczygielski & et.al., 2022) to enhanced efficient and safe maritime navigation based on current demands from end users that could be made possible by applying the TRANSACT methodology. These EURs should represent enhancements that could be made possible by shifting functionality to the remote edge and cloud tiers, that allows not only for increased utilization of available digital information coming from the connected vessel, but particularly the application of AI to the data such as ML and data analytics.

Specific user scenarios that explained the flow of events to achieve these EUR, including analysis of the current state of the art and identification of improvements beyond state of the art was then completed. Lastly UC2 partners identified goals and reasons for the transition to a safe and secure distributed solution, defining in the process the baseline and targeted KPIs for the UC.

A high-level architecture prior to and post application of the TRANSACT methodology, including identification of components that needed to be developed to realize the transition. For these components the starting and targeted technology readiness level (TRL) were identified (Akkermann & et.al., 2022).

Technical partners and end users then derived UC2 specific functional and non-functional requirements, that served as the basis of technical requirements (including safety, security, privacy, and performance requirements). The derivation of these requirements was based on a spiral model.

As part of the product management life cycle process a feasibility evaluation was conducted prior to start of development. There were several risk assessments tools applied to UC2. UC2 partners contributed both in the performance of a Hazard identification and assessment for the monitoring and the advisory services, as well as a risk assessment regarding security. Both manual and automatic safety checks was conducted on the response advice to clarify the navigational safety of the advice. Additional risk assessments are also in progress as part of the UC2 activities, including a risk assessment pertaining to the use of sensor data AIS in the development of AI-enhanced services. Security monitoring solution for detection and mitigation of security intrusion has been developed in the cloud tier. This will be extended to the edge communication in Y2.

The security assessment for UC2 was made under the following considerations:

- A continuous activity of risk management, and a risk analysis itself, provide a model of the system, in terms of assets, threats, and safeguards. This is the foundation for controlling all activities on a well-founded base.
- As a continuous process, management system of the information security is formed by four main processes: Plan, Do (implementation and operation), Check (monitoring and assessment), and Act (maintenance and improvement).
- This process of risk analysis not only helps to identify potential threats and apply safeguards, but also allows the organization to make decisions.

The risk assessment considers these three elements:

- Assets: considered as every element in the information system that are direct or indirectly valuable to the organization.
- Threats: potential incidents that may impact the assets, causing damage to the organization.
- Safeguards: defence elements deployed so that threats do not cause (so much) damage.

These elements allow the estimation of:

- Impact: what may happen.
- Risk: what is likely to happen.

Furthermore, an extensive risk analysis was conducted on the use of AIS data for route prediction of vessels. This analysis based on the Threats, Vulnerabilities and Risks (TVR) of a system. Vulnerabilities are the gaps or weaknesses in a system that make threats possible and tempt threat actors to exploit them. Within this analysis, the risk lies in the sole use of AIS data for the prediction of ship routes and a certain vulnerability/unreliability of the AIS data. Due to the lack of security devices within the AIS data, such as Quality of Service (QoS), manipulation of the AIS data from the outside is possible. By using AIS data alone, there is no possibility of merging with data from other sensors.

Threats are security incidents or circumstances that can have a negative outcome for a network or other data management systems. In case of using AIS data, the security incidents can be caused especially by jamming and spoofing and hinder/impede the overall functioning of a system and, on the other hand, generate false information within an AIS data set leading to drawing wrong conclusions from the analysis of the data sets. From the technical point of view, AIS is a system that produces streams of data. For example, for position reports, each vessel transmits information about its current location at time intervals that are predefined by the technical specifications. Collecting such data allows a ship's trajectories to be analysed. Static data are entered into the system during the AIS installation and need only to be changed if the name of a vessel changes or if the vessel undergoes a conversion to another ship type. Examples of static data are name, MMSI-number (Maritime Mobile Service Identity), vessel type, length and position of the AIS transmitter on the vessel. Voyage related information concerns issues like the vessel's draught, destination and ETA (Estimated Time of Arrival). This information needs to be kept up to date manually by the ship's crew, which makes it sensitive to errors and uncertainties.

Dynamic data originate from the vessels navigational instruments; examples are the vessel's position, heading and rate of turn. The position of the vessel is reported by three parameters: longitude, latitude and a position accuracy report. The longitude and latitude are given in 1/10,000 minute, which is -in latitudinal direction- equal to approximately 20 cm. In longitudinal direction this number depends on the distance to the equator (Netherlands ≈ 11 cm). The position accuracy report indicates how accurate the two mentioned position reports are.

The Risk Analysis contains the following risks in using AIS data:

- Message integrity: Wrong values in AIS Destination Field,
- Invalid MMSI and Multiple transceivers sending info for the same MMSI,
- AIS a system with weaknesses
- Trajectory Outliers,
- Ship Route Planning Using Historical Trajectories Derived from AIS Data,
- AIS Data Anomaly,
- AIS Jamming.

Regarding the existing risks in AIS data exploitation, the analysis provided a detailed description of the safeguards recommended to address these risks. The recommended measures elaborated in detail are:

- Destination Cleaning-Matching Approach
- MMSI Cleaning-Matching Approach
- Message integrity: Cleaning Wrong values
- Approach Cleaning Vessel Trajectories Outliers
- Outliers removal

- AIS Data Normalization and a

Spoofing Detection Model.

-

Most of these approaches are very important to predict vessel's movements and destinations. NVT is planning to use some of those methods (1,2,5) as a preparation phase for the Demo 3. Partially, auxiliary variables for describing journeys' segments were calculated and visualized. The visualization is expected to clarify specific issues in AIS data and detect threshold limits for specific periods or specific areas. Such specific irregularities or violations makes the process of data curation not very generic.

### 4.2.4 Lesson learned

Through the development of the UC2 solutions these are some of the important lessons learnt:

- In solutions with a large number of modules and independent development teams, a good total solution description, high-level data flow diagrams and clear interface specifications have been important contributions to the success of the development.
- Designing the solution as a multi entry point service, allowing for similar service request both on edge and cloud side have proven valuable in further product and service development.
- Implementation of the Advisory Service Framework agnostic of specific advisory services have enabled rapid deployment of new and extended services in the NAVTOR ecosystem.
- After initial implementation it has become clear that a stronger enforced modular design on the edge device will be beneficial. This will be addressed through the development of a Docker based edge infrastructure.

## 4.3 Transition of Use Case 3: Cloud-featured battery management system

In a cloud-featured battery management system, electric vehicle battery data is collected and transmitted using an advanced and secure data logger and transferred encrypted to a data broker cluster; the data is stored in an optimized database. All of this is happening while the Electric Vehicle Fleet (EVF) is driving.

Figure 19 depicts the high-level data exchange pattern between the services according to the TRANSACT architecture. Safety, time and performance critical communication, control and decision-making is mainly restricted to and between the device and edge tiers. The battery management system (BMS) measures battery specific quantities and controls the system accordingly. Namely, the BMS does not require a persistent connection to the cloud in order to run and perform properly. On the other hand, the BMS is connected to the cloud via an LTE Gateway in order to send telemetric data to the cloud and receive optimization and software updates.

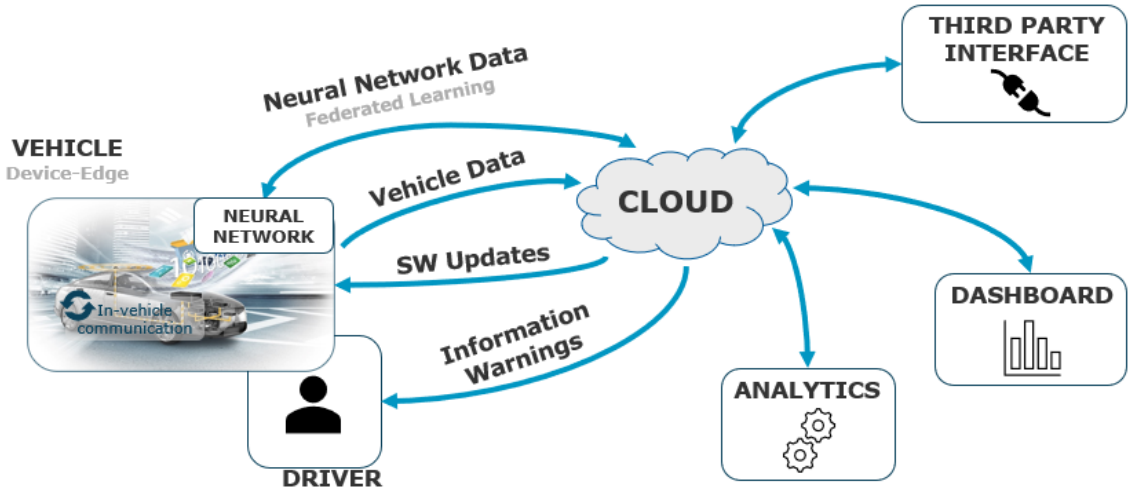


Figure 19: High-level picture for Use Case 3. The Battery Management System is connected to a cloud in order to transmit telemetric data and to receive software updates. Beside of this, in-deep analytics can be performed by applying neural networks, either direct

Up to now, a Battery Management System is an isolated and disconnected device. However, the next-generation will be a connected and distributed solution. UC3 aims at introducing additional features due to the transformation.

### 4.3.1 Transition to TRANSACT Reference Architecture

A Battery Management System (BMS, see Figure 20) is an embedded system to control the interaction and current flow between the battery, the powertrain and the charging unit. By sensing physical quantities as voltage, current and temperature, the optimal and safe operation conditions are set in order to protect the battery from over-charging, over-discharging, over-current and short circuits (safety-critical functions). Additional values are derived by models running on the BMS - so called state estimators - which describe the inner states of the battery which are not directly measurable, e.g. the electrode potential, State-of-Charge (SoC) and State-of-Health (SoH). Those values are also used to control the BMS, e.g. for cell-balancing. Additionally, based on these values, further information is generated and provided to the driver, e.g. range estimation (mission-critical function). Apparently, there is a trade-off between the safety of the system and the utilization of the battery and accordingly the comfort of the driver. Operational and charging limitations imposed by the BMS can lead to a reduced range and to a slower charging. These control parameters are defined during the development and testing process at the early product cycle with a strong bias towards safety. However, after years of operations, there may be a better state estimator which can lower the limitations and increase the comfort. So far, the update of the model via flashing the BMS software requires a wire-bound connection to the BMS which is only possible when the vehicle is brought to the workshop and the update is then performed manually (see Figure 21). In the scope of TRANSACT, the BMS will be enhanced to enable non-critical functions which have been not applicable before. The state-of-the-art BMS is an isolated and disconnected system. The transformation according to the TRANSACT architecture will promote the system to a connected system which has the capability to communicate with a cloud backend but also indirectly with other connected Battery Management Systems to enable cross-device but also cross-silo communication. In this way, all participants benefit from swarm-intelligence.

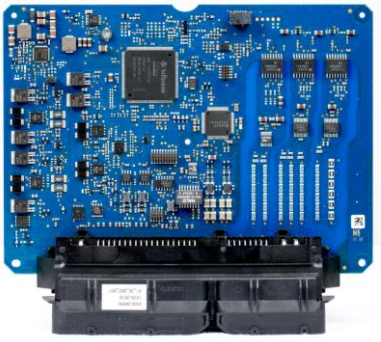

Figure 20: A local and isolated Battery Management System.

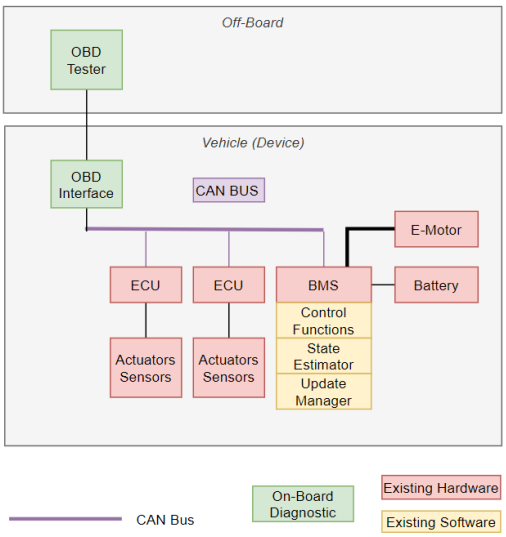


Figure 21: System design of the baseline setup. The BMS is only accessible via the On-Board Diagnosis (OBD) interface which requires a wire-bound connection. The communication to other electrical control units (ECU) is established via the CAN bus.

#### 4.3.1.1 Transformation towards a distributed solution

Figure 22 shows how the system could look like after the final transformation. The initial stand-alone BMS is distributed over the device-edge-cloud continuum and extended by additional functionalities.

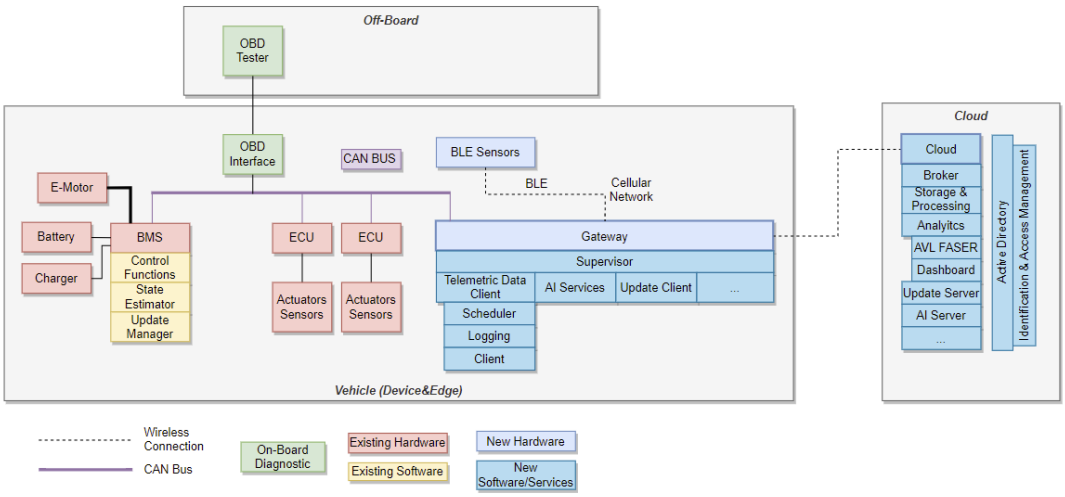


Figure 22: System design after the transformation. The Gateway and Cloud can host several additional components and services to provide additional features.

Figure 23 depicts the mapping between the components and the TRANSACT reference architecture. The BMS (device) is connected via CAN to a LTE gateway (edge), which not only transmits the data to the cloud but

also comes with enough computational power to run AI models and the training. It further hosts the update manager for flashing the BMS. An additional Bluetooth Low Energy (BLE) interface enables to connect additional sensors but also in general to connect the BMS wireless to the gateway. In order to secure the transmission, a public key infrastructure (PKI) is employed. The cloud backend provides a broker end-point for all clients. The data are centrally aggregated, analysed and stored in the Fleet Analytics Services application (AVL FASER). The same app also provides the possibility to deploy custom analytics scripts – handled by the Script Processor. The bi-directional wireless communication can be used to provide further results from the deep analysis of data in the cloud, send warnings but also updates of AI models (Federated Learning) and software for both the BMS but the gateway itself.

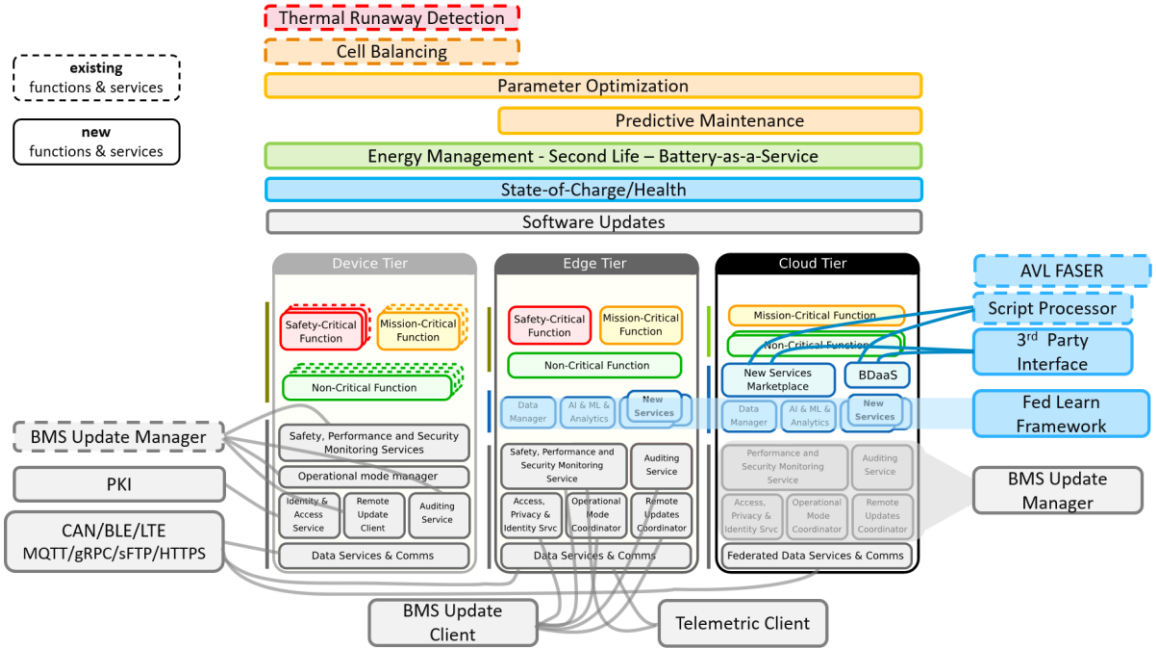


Figure 23: Connection between TRANSACT components and BMS functions and services.

Due to the time criticality, it is not feasible to offload safety-critical functions from the embedded hardware. However, safety-critical functions can be updated and lead to strict requirements for the services. The main focus lies in improving and adding mission-critical and non-critical functions.

Conclusion

The extension of the BMS device over the edge and cloud enables to integrate new features related to safety-critical, mission-critical and non-critical functions. In this way, the end-user benefits from increased safety but also from a plus of comfort.

### 4.3.2 Organizational changes to support the transition

The extension of the BMS requires the inclusion of additional teams. Before the transition, mainly the BMS Hardware, Software and Safety teams have been involved in the product development cycle. After the transition, the teams of Digitalization, Data Analytics, Security and IT need to be included. This describes quite well the increase in complexity of the distributed solution.

Depending on the business model, either the existing AVL setup for the solution can be duplicated or scaled on the existing infrastructure or the whole infrastructure must be deployed and integrated at customer site. There can also be mixed setups: E.g. the Gateway software is integrated at a given hardware but the cloud is

provided by AVL. The customers' personal must be trained to maintain the system but also to understand and integrate the analytical parts of the backend and communication paths, including security features.

### 4.3.3 Planning and execution of the transition

Recalling that the project takes three years, one should assume that interacting and connected peripheral components significantly evolve during this period. In this specific Use Case, the complete electrical-electronical architecture (EEA) of the vehicles undergoes a major change towards a server-zone architecture. That means, that the logic from light-weight control units (ECU) is offloaded to central domain controller with more computing resources. This leads to massive changes regarding the total system architecture of the Cloud-Featured BMS. It was initially planned to extend the BMS itself with a telemetric unit. This plan was changed after the project start and re-focused on the concept of having a more powerful domain controller. The target vehicle system is not the current state-of-the-art but an upcoming one. Research of the current evolution was necessary to identify the upcoming generation of vehicle architecture and technology in order to create a future-proof solution.

The core of UC3 is data: The data acquisition (Demonstrator #1) and the data usage (Demonstrator #2). So far, data from vehicles and especially from the BMS are rare accessible. The availability of data is relevant for anyone who is involved in the battery lifecycle.

In order to kick-off the technical transformation, end users and stake holders have been identified. Depending on their demands, relevant services have been defined. The next step was to derive the requirements to host these services and figure out, how they have to be distributed over the device-edge-cloud continuum. The flexibility to add additional services at any time and the capability to integrate the software on a different target system but also integrate third party solutions gave additional constraints. The system design was then analysed, decomposed and mapped to the tiers to derive a software architecture in alignment with the reference architecture.

The scope of Demonstrator #1 is the data acquisition. A Gateway has been developed which enables bi-direction communication between the vehicle and the cloud system. It was necessary to start from scratch with the development as no available solution on the market was able to meet all requirements or came along with impractical restrictions in terms of technical features or software. Unfortunately, this has introduced additional unplanned efforts. However, the developed solution provides now exactly what we need.

### 4.3.4 Lesson learned

From the gathered experience so far, the following generic points are derived.

- As the development takes years, perform an analysis of current and future trends and technologies.
- As the system increases in complexity, identify the additional stake holders and onboard them from the very beginning of the transformation process.
- As many different domains are combined, start with a high-level context and system scheme that everyone understands and such that everyone can contribute with his/her specific domain knowledge.
- Identify dependencies between components and teams. Ensure that either you get rid of the dependencies or that the teams are well aligned.
- Find a balance between keep-it-as-it-is, re-design and re-distribution. I.e. alter only the existing system (device) if really needed. "Never touch a running system."

## 4.4 Transition of Use Case 4: Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic imaging systems

Use Case 4 "Edge-cloud-based clinical applications platform for Image Guided Therapy and diagnostic imaging systems" is a healthcare use case, aiming to improve the workflow and interoperability in hospitals (see Figure 24). In particular, the use case addresses image based minimally invasive clinical procedures which are typically performed in Cathlabs or Operating Rooms. Clinical procedures in such environments are typically very complex and involve a team of healthcare professionals (with a variety of expertise in multiple disciplines) and many specialized devices (see Figure 25).

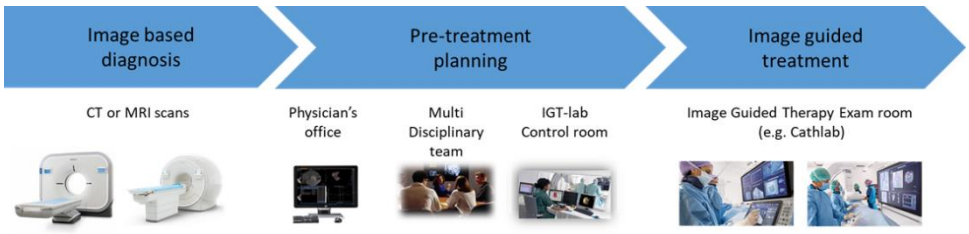


Figure 24: Example workflow for image guided diagnosis and therapy.

In order to deliver optimal treatment, all collected (multi-modal/multi-source) data should be easily available for sharing and interpretation during examination for the healthcare professionals. However, currently the exchange of information is often hampered by a lack of tools to support efficient collaboration between disciplines and often requires collecting and reviewing information in the room outside the Cathlab which slows down the performed procedure and possible effectiveness of the overall treatment.

Figure 25: Typical setting during image-guided therapy with physicians utilizing medical imaging equipment for the minimally invasive treatment of patients

In UC4, a medical imaging device performing safety-critical applications, such as live X-ray imaging, will remain as an embedded functionality in the device. However, mission-critical functionality, such as non-real-time image processing, offline planning and intra-operative analysis tools will be deployed outside of the device, i.e., either on the edge or in the cloud.

In addition, simulations using different system parameter settings (bandwidth, latency, system availability etc.) and real clinical workflow parameters (e.g. image reading turnaround time) will allow assessment of the

impact of cloud-based components on the actual end-user workflow. These simulations will be performed even beyond the strict IGT scope and include also other imaging modalities (e.g. MRI).

### 4.4.1 Transition to TRANSACT Reference Architecture

Use Case 4 is positioned in the healthcare domain and addresses the distribution of system functionality over the device, edge and cloud layers. In this process, a careful analysis is made which critical functionalities should remain on the device and which less critical functionalities can be transitioned to edge or cloud tier. The instantiation of the reference architecture for UC4 is shown in Figure 26 together with allocation of the main functionalities over the device, edge and cloud tiers.

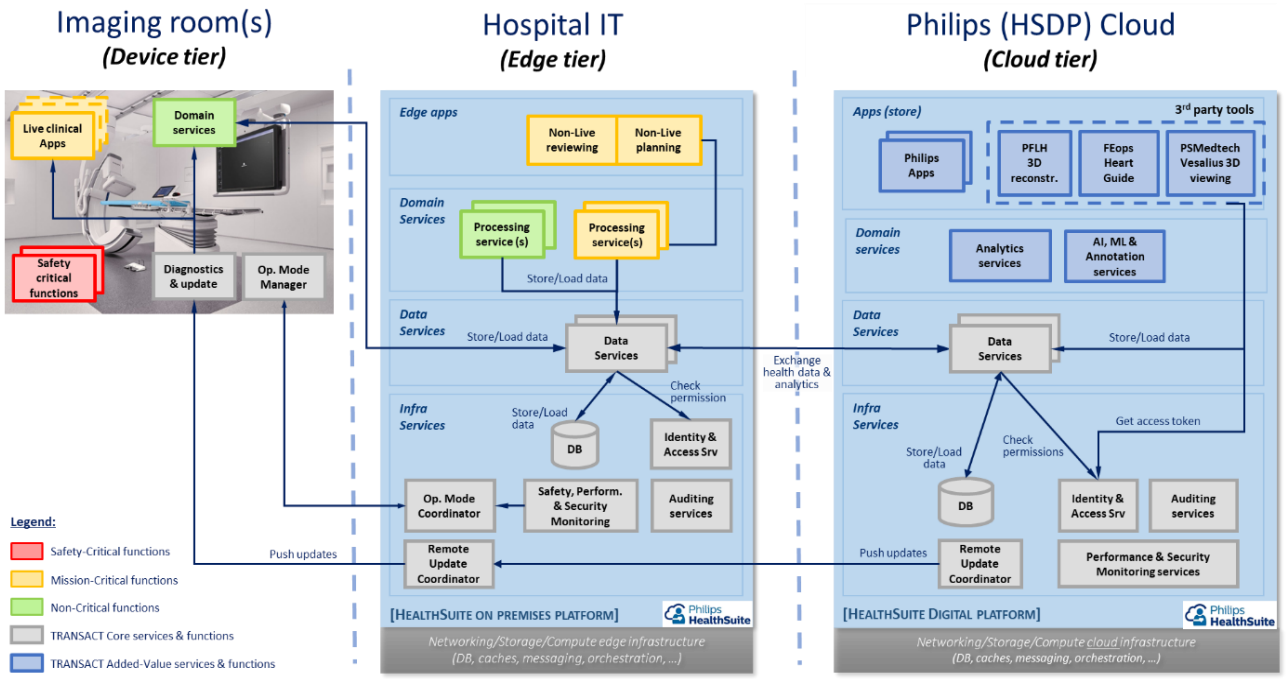


Figure 26: Mapping of Use Case 4 to Transact Reference Architecture

The real-time control of the X-ray source and associated X-ray exposure control are essential for the safety of patient and staff during imaging. Due the hard real-time constraints, the implementation of this safety critical function remains in the device tier.

Soft real-time applications for planning and image viewing are considered mission critical and can be allocated to the edge tier or cloud tier. In the UC4 context this can be the hospital-IT compute platform or e.g. a cloud platform. An important consideration for the transition is the ability to scale easily towards imaging applications that are computationally demanding. Another consideration is the Cost of Ownership of the medical compute platform: having a medical image processing PC for each imaging lab in the hospital, while its utilization is low during a day is not cost-efficient. Sharing a single image processing PC for multiple rooms leads to a much more efficient utilization and reduces the cost for the hospital.

Big data analytics and AI-enabled applications are valuable for gathering data insights about the system usage and the optimization of the user experience. Such functions are non-critical and are allocated to the cloud tier.

In addition to that, applications that allow planning of the intervention, e.g. based on prior acquired CT, MR data can also benefit from cloud deployment. Since the pre-interventional planning is not core to IGT

business, these applications are also developed by other Philips businesses than IGT. These *Philips Apps* benefit from a common technology stack for medical image viewing and analysis. These capabilities are supported by the HealthSuite platform.

Performance monitoring and management is an important element in the UC4 architecture and is closely linked to operational mode management. A dedicated collaboration between Philips and TU/e has been established to develop a performance monitoring framework which will provide the data on which the operational mode management system can act.

SaaS cloud infra services for clinical data storage, device connectivity, logging, monitoring, identity and access management need to be provided. These are not core to the business of IGT and are services that are required for other Philips solutions and products as well. These capabilities are supported by the HealthSuite platform.

### 4.4.2 Organizational changes to support the transition

With the transition from a monolithic architecture towards a hybrid Cloud-supported architecture comes the need to establish a development organisation that facilitates this transition. Organizational changes in the development team have started in three areas: (1) Cloud technology, (2) Data & AI and (3) 3rd party integration.

Originally a dedicated IGT software platform development team worked on the design and implementation of such platform. For the realization of the TRANSACT architecture, it was necessary for the organization to on-board expertise in the area of cloud technologies. Multiple workshops and design sprints have been organized with medical compute experts from AWS and Nvidia to explore the routes to such a hybrid architecture.

As the Philips IGT organization recognizes the importance and value of Data & AI driven solutions we have set up a dedicated team addressing the opportunities and challenges that come with digital transformation of image guided therapy solutions. It entails setting up infrastructures for the data collection from systems installed worldwide and the generation of actionable insights from this data. Another important area of interest for the data transformation team is the development of AI based applications and the labelling and annotation of data that is required to develop such applications.

SaaS cloud infrastructure provide by HealthSuite platform requires that IGT has a different operational model with this part of the Philips organization: IGT-HealthSuite will have to agree on SLAs and setup interaction channels for DevOps for monitoring, alerting and upgrading the lower level cloud infrastructure. HealthSuite needs to further professionalize the onboarding, scalability and functional fit of their services to cater for IGT needs.

IGT is relying on the pre-interventional planning outcome to give proper guidance during the intervention. Typically, IGT created these planning applications themselves, but will reorganize to rely more on other Philips Apps to provide planning applications, e.g. to enable planning outside the interventional room. These other departments work closely together with HealthSuite to provide common image viewing and analysis software components.

Finally, with the ambition to integrate with 3rd party applications, more effort is dedicated to the integration efforts with such parties, the exploration of joint development agreements and investigation of new business models for joint propositions.

A cloud-based solution requires a different support organization. From a help desk perspective to answer customer calls there is not so much a difference. But as part of the solution, continuous monitoring of the solution is required which also gives extra opportunities. One can and should continuously monitor whether all services run properly and act as soon as an anomaly happens. If needed, the customer can be warned

upfront. Typically, this setup also provides the opportunity to add a more digital channel for customer interaction. As part of the solution, one may add the option to give direct feedback, which is immediately sent to the cloud services and can be handled immediately. For the user, it becomes much easier to contact the supplier for issues and ideas.

A cloud solution also gives the possibility to perform faster updates. Almost instantaneously the whole installed base can be upgraded, or a fix can be rolled out. On the development side this means much faster update cycles, which requires an efficient CICD pipeline to develop, test and deploy efficiently. From a release process this also requires a change of mindset. It must be possible to fix, test and release fast with minimal overhead. More and more test automation is an absolute necessity for this. Also the development and documentation processes need adaptation to enable this.

### 4.4.3 Planning and execution of the transition

The transition towards a hybrid architecture started with a creation of the Value Proposition including thorough analysis of the customer needs and business needs. Validated learnings have been collected about the desirability, acceptance and viability by the customer of a cloud supported platform. Business needs have been analysed as well, addressing amongst others the positioning in the digital transformation journey, extending the existing portfolio of IGT solution propositions and the ability to access big data which is enabled through a cloud platform.

As part of the transition the Total Cost of Ownership (TCO) of a cloud-supported solution has been analysed. The costs associated with cloud infrastructure are typically based on subscription fees, whereas the installation of an on premise system usually is determined by the one-time costs of purchasing the compute infrastructure that will be installed on the customer's site. As part of the planning phase an analysis was made of the overall proposition. This analysis for the transition towards cloud supported functionality included: strategic fit, financial plan, market factors, customer value, risks, uncertainties and capabilities to execute.

Once the proposition was approved, the execution phase started for which a multidisciplinary team has been assembled comprising of experts from the innovation team, system architects, clinical science and marketing. The main activity of this team was to perform a Concept & Feasibility study and realize a prototype of the cloud supported solution. As part of the prototype a single clinical software application has been taken as a reference, where part of the functionality has been realized in AWS. Subsequently the prototype was used to gather insights on e.g. workflow and responsiveness and collect early feedback from physicians. Following the Concept & Feasibility study a Product Realization Project has been initiated which is now working on the first release of the IGT Cloud solution.

For providing the SaaS infrastructure, HealthSuite has established standardized ways for onboarding, communication, SLA, upgrades, customer request etc. In Transact this is adopted as-is.

For the creation of the cloud-enabled Philips Apps, important technology choices need to be made in both HealthSuite (providing basic building blocks towards Philips Businesses / productizing few of the applications) and the other Philips Businesses (productizing the majority of the applications). To support the Transact use-case HealthSuite has setup different teams that developed a cloud native solution, starting with basic 2D viewing capabilities (as demonstrated in first year) and initial versions of cloud-native services for 3D rendering and segmentation. The development effectiveness was evaluated together with other businesses and a transition to a new approach is executed which will be used in Transact (in 3rd year)

### 4.4.4 Lessons learned

The work done on the transition towards cloud supported applications led to the following insights:

- Transitioning towards a hybrid, cloud-supported architecture forces one to think carefully about critical functionalities, their performance requirements and their allocation in either device, edge or cloud tier.
- A centralized cloud-based compute facility for multiple clients with high demanding requirements in terms of data size and response time, requires the implementation of a performance analysis and management system to ensure proper end-to-end system performance.
- Transitioning from a monolithic to a hybrid architecture requires a new way of calculating the total cost of ownership, taking into account the differences in initial and recurring costs.
- When transitioning to cloud you should consider the economy of scale and innovation speed of the major cloud infrastructure provider(s). Some of the lessons learned within Philips are listed below:
    - For the SaaS infrastructure the HealthSuite platform initially had a portable cloud strategy. This led to additional complexity and need for portability didn't become a reality. Hence, the decision was made to adopt an AWS only strategy.
    - HealthSuite has created functionality on top of AWS and other technologies specifically for clinical purpose (e.g. security certification, medical image storage). This led to a lot of responsibility in the HealthSuite group slowing down innovation in the Businesses. Hence a decision was made to move to shared responsibility model, providing business a set of best-in-class recipes, guardrails and default implementations based on a more native AWS experience to reduce the business' time to market. To leverage economy of scale, leverage the fact that AWS is moving up the stack (additional functionality), and AWS has alternatives to the 'other technologies', Philips is more strongly partnering up with AWS. For example, enable the direct use of AWS native services (with proper guardrails), security certification of (specific) AWS services, co-development with AWS.

When transitioning to cloud and you have a large product portfolio on a legacy technology, consider a slow migration that can still cater for existing products instead of a migration on a technology stack that isn't functionally complete enough to replace existing products. For the Philips Apps, the transition to a new technology in a greenfield approach, which ultimately should support the evolution of many existing products into cloud-enabled visualization and analysis applications, seems to be infeasible. In essence it's not feasible to 'make it cloud AND don't stop delivering (to enhance existing product)'. The cost of and duration for re-developing functionality that is already existing in current (non-cloud enabled) products is too high. An approach is evaluated to leverage existing components in a cloud deployment and slowly migrate to more cloud-native approaches, falling back on existing components for their functional richness while expanding cloud-native technologies to gradually replace the functionality of existing components.

## 4.5 Transition of Use Case 5: Critical wastewater treatment decision support enhanced by distributed, AI enhanced edge and cloud solutions

Wastewater Treatment Plants (WWTP) are urban infrastructures that reproduce the biodegradation processes that occur naturally in rivers in an intensive way to remove pollutants added to the water after public consumption, industrial uses and storm water drainage to an acceptable level. This level of acceptance is regulated by local, national and European authorities and depends on the use of the reclaimed water (e.g., reuse in agriculture or industry, discharge into different categories of natural environments). The most extended kind of WWTPs involves physic-chemical treatment and biological treatment in different stages for removing solids and pollutants, breaking down organic matter and restoring the oxygen content of treated water.

Today's WWTPs are automated with Supervisory Control and Data Acquisition (SCADA) systems that centralize the control and monitoring of the WWTP executing multiple local control loops in a monolithic and constrained system. SCADA software allows monitoring the state of the processes, acting on the components of the process control system. It allows the operation, maintenance and supervision of the process, modify recipes or batch sequences, edit actual values or contact the process through automated systems.

The benefits of moving from the current monolithic/centralized SCADA to a distributed system will allow the achievement of the following specific objectives:

- Anticipate and avoid failures due to undesired discharges to the sanitation network through an enhanced control loop that allows to speed up the WWTP response, lowering risks.
- Adopting predictive maintenance strategies to avoid downtime and improve operation by cross-WWTP analysis.
- Creating a data lake at the cloud for cross-WWTP analysis and creation of AI models.
- Allowing current systems and equipment to scale backed by the edge tier.
- Improving the management and operation of similar WWTPs through joint data analysis.
- Minimising operating costs in terms of energy and reagents, this will not only aim at cost reduction but also at reducing environmental impact, reducing the production of greenhouse gases thanks to an improved energy management.
- Assist WWTP personnel in the undertaking of safety-critical decisions.

UC5 will implement the DEMOs initially in three WWTPs sites in the Region of Valencia (Spain), in facilities owned by the public sanitation administration of Valencia (EPSAR).

### 4.5.1 Transition to TRANSACT Reference Architecture

Currently, SCADA systems are the standard system to control processes in medium/large WWTPs through a series of control loops and actuators such as pumps, compressors or valves. Applied control techniques may span classic control (PI / PID, Cascade, etc.), heuristic (fuzzy, rule-based, neural networks, etc.), and based on models (optimal, predictive, adaptive, etc.). Its selection and precision depend on how powerful the computing equipment is. Most or all processes are run locally, occasionally storing some data in the cloud (e.g., Google Drive) with sharing purposes.

WWTP SCADA is typically PC-based and located in a control room at a treatment plant, allowing operators to view the entire process and perform control actions. Within the plant, process controllers or programmable logic controllers (PLC's) supervise unit processes, such as chemical treatment and filters. A local area network (LAN), such as Ethernet, links the controllers to the workstations as well as to one another. Remote terminal units (RTU's) are used at remote sites and usually exist in vulnerable areas, such as pump stations.

Use case 5 will take place in at least four different locations, three WWTPs and the DAM headquarters. Figure 27 provides an overview of the different elements and deployments planned for its execution. In the starting situation of the environmental use case (baseline), only the Device layers existed, except, obviously, the TRANSACT core services and functions, so that the new architecture implemented thanks to the transact process includes new advanced services and functions in the edge and cloud layers. It should be noted that in this use case the pre-existing functionalities in the Device layer have not been moved but TRANSACT Value-added services and functions complementary to the existing ones have been created.

The demonstrators to be carried out in the different scenarios are three: an enhanced spill control for Muro and, potentially in the future, for Algemesí; a mechanism for predictive maintenance; and a series of dashboards and managerial tools to improve the visualization and interpretation of data across WWTPs to improve their operation and exploitation.

To perform these tasks three edge nodes is expected to be deployed, one per WWTP. The goal of these nodes is to host the enhanced spill detection tool in Muro and Algemesí (if it was finally to be deployed) and the local predictive maintenance module in the three WWTPs. For the local predictive maintenance module, it is expected to need, at least, Radiatus, a BDaaS, that will offer some extended visualization tools and host a data repository to store the data coming from the WWTPs; and from TRANSACT core, the monitors and data services and communication modules. In addition to these, for the enhanced spill detection, the edge nodes will also include additional AI, ML and data modules as well as operational mode coordinator. The rest of the modules are not foreseen to be used at this stage or their use will be conditional on the functionalities offered by the components. Also, at the device level, the modules most likely to be used are the monitors, the operational mode manager and the data and communication services. Finally, the cloud tier will include a series of AI and ML tools to aggregate and analyse the data coming from the different edge nodes. Radiatus will be used to deploy a data lake and AI, ML and visualization tools to create the required dashboards and management tools. From TRANSACT core, this tier will probably also include the auditing and access, privacy and security services.

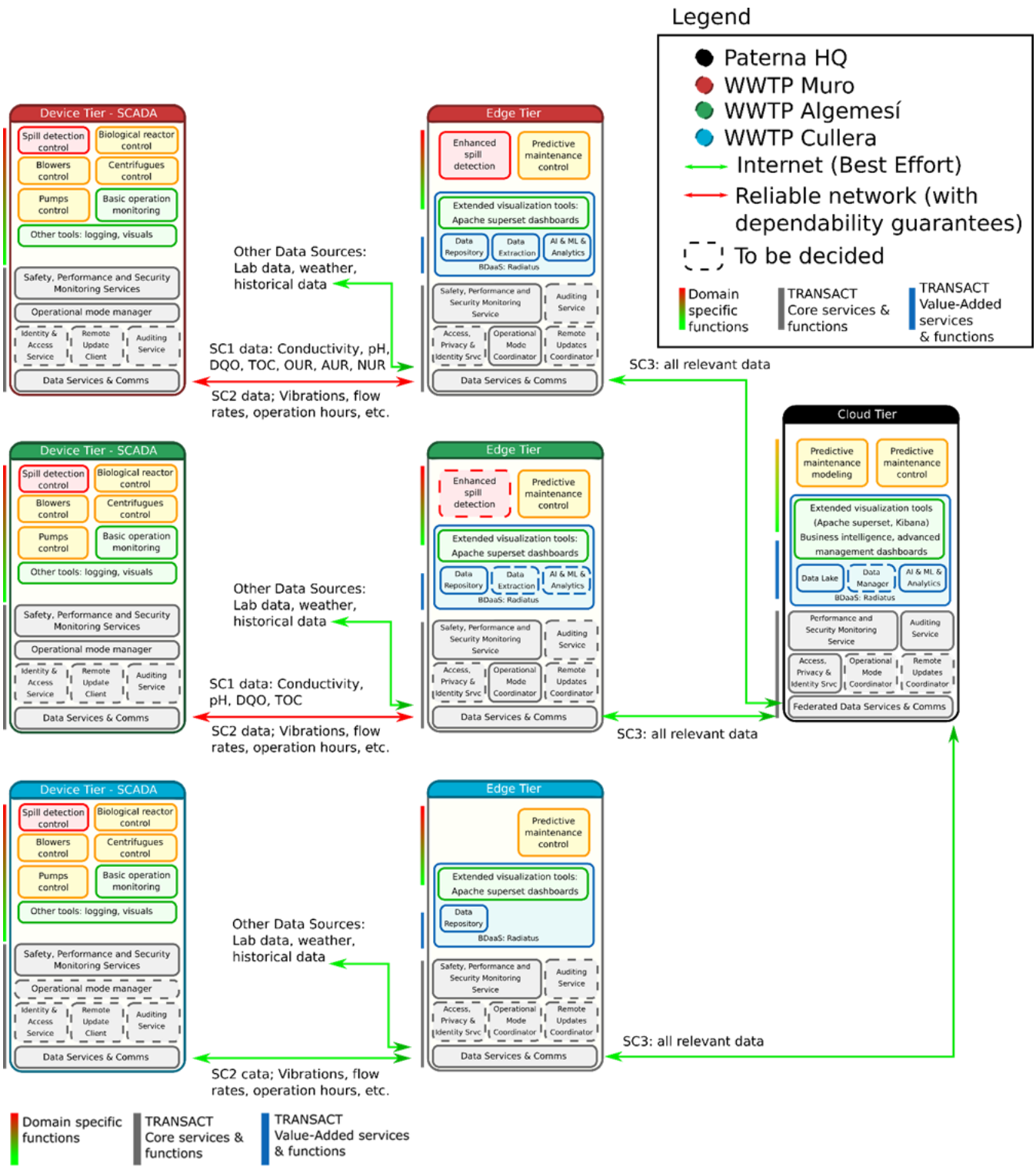


Figure 27: Use Case 5, High Level Architecture

The newly developed **Safety-Critical Function** regarding enhanced Industrial spills early detection and prediction tool based on AI will be distributed in the edge tier. The complete function will not be moved to the edge layer but will be complemented by value-added services based on AI at the edge tier. This will allow, among other things, alerts and warnings to be sent to technicians when a spill is about to reach the critical

points of the plant, which would not be possible using only the device layer, as it would not have sufficient resources for its execution.

The new **Mission-Critical Functions** about Predictive Maintenance Control and Predictive Maintenance Modelling will be both deployed on the edge and cloud tiers. As in the previous case, we will complement the device controls (data sources) with cloud modelling of predictive maintenance and on the edge will report alerts (shutdowns or recommended maintenance). Again, it is an enrichment of the device tier functionalities for value-added services.

The **Non-Critical Functions** that have been distributed to the remote tiers are related to the Extended visualisation tools related to Apache Superset dashboard in the edge tier and Business Intelligence Advanced Management Dashboards in the cloud.

The **TRANSACT Core services and functions**, methods and tools provided by TRANSACT reference architecture that are foreseen to be applied when implementing the use case are related to:

- Monitoring services (Safety, Performance, Security). UC5 will deploy real time monitoring solutions in the plant by means of an industrial probe that will continuously scan the inventory to add automatic labelling. The system is capable of not only translating an alarm by identifying the assets involved, but can also indicate the risks, controls and actions associated with that alarm, assigning a responsible party and a deadline for intervention.
- Operational mode coordinator: the added value services and functions introduced in DEMO1 and DEMO2 will launch alerts, switching between operational modes (e.g., emergency or action modes or return to normal state) depending on sensor signals.
- Identity, Privacy and Access services: it will be possible to define different levels of access in communications to specific users, accordingly with the conclusions reached at WP3.
- Data services and comms: KAFKA-based or MQTT-based subscription publishing model will be used, as well as Apache Superset-based dashboards for data access and visualisation

The new **TRANSACT Value-added services and functions** introduced in the use case are related to a series of novel decision support tools:

- An enhanced spill detection function, with a predictive flavour able to assist operators and potentially automate some actions based on AI. This function complements a more basic one at the device/SCADA tier (Edge Tier).
- 2A Maintenance Management function to deploy models resulting from the predictive maintenance modelling performed at the cloud tier based on AI. These models may apply to critical equipment like, for instance, centrifuge equipment (Edge Tier).
- 3A Predictive Maintenance modelling function, gathering data from multiple WWTPs to create an accurate model that can be leveraged in the different plants (Cloud Tier) based on AI.
- A Cross-WWTP operation analysis service for comparing different performance ratios and metrics coming from different WWTPs to assist current decision mechanisms (Cloud Tier)
- A series of improved visualization tools at WWTP and cross-WWTP levels (Edge and Cloud tiers)

Therefore, the different demos proposed in Use Case 5 are based on extending the available functionalities of the device, the WWTP itself, by leveraging resources at the edge and in the cloud. This implies that the operation of various processes already existing in the plants will be complemented, aided or assisted by processes that will be deployed at the edge or in the cloud.

### 4.5.2 Organizational changes to support the transition

In UC5, the changes foreseen in the organization or processes to support the transition to the distributed TRANSACT reference architecture, from the end user point of view, are mainly related to the following areas (having in mind that DAM has the role of UC owner and end user of the technology):

- One of the most relevant aspects is to guarantee the security of communications, avoiding or minimising the risks related to cyber threats or possibilities of remote access that could sabotage the operation of the WWTP. This is currently the main market barrier for the implementation of project results in critical water cycle infrastructures. It is crucial to be able to implement systems capable of ensuring security against attacks in order to generate confidence in the environmental authorities that own the public infrastructures that make up the complete water cycle. In this use case, efforts are being made in this sense by applying the NOZOMI, SIRENA and G-Consulting tools developed by SINGLAR.
- In agreement with the previous bullet point, it is essential to generate confidence in the environmental authorities, especially in aspects related to data security and privacy, for which demonstration projects such as TRANSACT play a very important role. It is very important to make a strong effort in the dissemination of the results obtained in fairs and congresses of the sector, beyond the merely scientific or academic field.
- It may be required an effort for further sensing of some sewage treatment plants and to enable the data transfer. One of the main limitations for a more widespread utilization of WWTP AI models is generally related to the scarce data sets measured at the inlet of the WWTP. The high cost (both in terms of workload and financial resources) related to experimental collection of an extended dynamic influent dataset is one of the main reasons. The use of on-line sensors still remains complicated, since the sticky materials of raw wastewater and the heavy deposit of pollutants make their maintenance cost considerable.
- It is necessary to enable the transfer of data from installations by installing platforms or gateways to take signal readings from equipment and make them accessible and in some cases, it is necessary homogenise the way in which data from WWTPs is collected and reported from the different facilities and regions.
- Human resources: for the implementation of TRANSACT methodology in the water cycle facilities managed in a wider extent it would be necessary to make an effort on training the plant managers and middle managers as end users of the new services implemented. It would be also advisable to recruit new profiles with a data science background for the IT Department of the company.
- Investment in new infrastructure for the deployment of the novel AI services developed within TRANSACT project. For instance, within the project implementation, project partners have provided the following infrastructure:
    - Server for AI with the following characteristics: 4 vCPUS with 32 GB RAM (In this server you have to install the ecosystem to be able to perform the data analytics and train the algorithms) if it can be better with GPUs.
    - For the deployment of the different solutions and extract, transform and load: In this case, a Kubernetes cluster with 8 nodes with 4 vCPUs, 16 GB RAM, 250 GB each. In this cluster we will set up a pre-production, production and testing platforms and possibly a development environment.
    - Space: at least 1TB to host the received data
- On the other hand, changes in IT infrastructure on the end user side are expected too for adapting to TRANSACT platform that encompasses the compute continuum across three tiers: cloud, edge and device. Therefore, the new added value services developed will be integrated in Radiatus. It is a platform that manages the distributed deployment of multiple technologies, some of them dedicated

to Big Data Analytics and visualization. It facilitates the usage and integration of these technologies by applications.


Figure 28: Radiatus PaaS

- Furthermore, Kumori Platform is a PaaS for deploying and managing services based on containers. It provides platform operators with a reliable cluster distribution that runs Kubernetes under the hood, well-tested, strictly versioned and based on open-source components. For service developers, Kumori provides an expressive Service Model for defining their microservice-based services. This allows developers and integrators to provide a declarative high-level description of the service architecture, including the communications topology between the microservices. It also provides powerful automation for those services, driven by their Service Model definition.

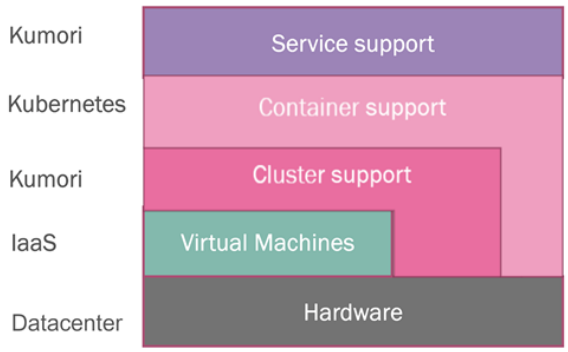


Figure 29: Kumori Platform is a PaaS for deploying and managing services based on containers

In this UC, Kumori will be deployed on the cloud and edge tiers, acting as a container orchestrator for cloud-oriented scalable applications. Radiatus will also be deployed on top of Kumori, providing the integration of various technologies to the applications. As an example, Apache Superset, on the cloud tier, provides functionality related to dashboards for data visualization. ThingsBoard, deployed on the edge tier, focuses on managing groups of devices on the device tier.

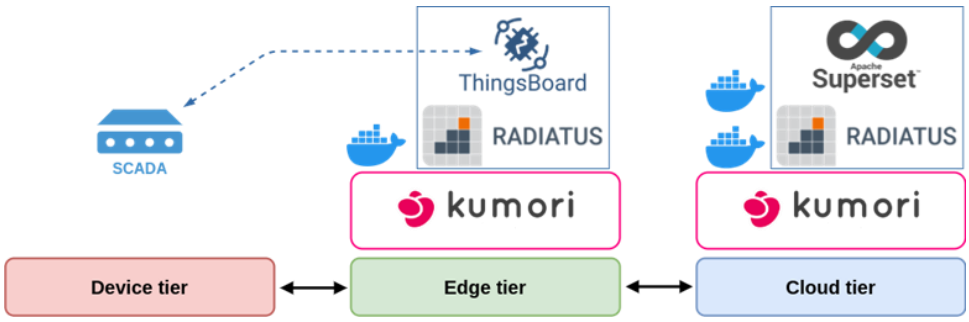


Figure 30: UC5 IT infrastructure

### 4.5.3 Planning and execution of the transition

Transition to TRANSACT methodology has required certain preparation duly documented in the deliverables of TRANSACT. The objective of this use case is to move from the current centralized SCADA to a distributed system to improve plant operation: detect and anticipate failures due to undesired discharges to the sanitation network, adopt predictive maintenance strategies to avoid downtime, create a cloud-based dataset for cross-cutting analysis of the plants, and assist staff in making safety-critical decisions. The process started with a **deep feasibility** study of the state of the art about the management of wastewater facilities, and the digitisation of the water cycle in general. Some major challenges and opportunities were identified, together with the risks and constraints involved.

A conscious **risk analysis** of the plant assets was carried out by a systematic mechanism and approach for distributed CPS that ensure homogeneous and appropriate security and privacy (GDPR) levels are being adopted, based on MAGERIT methodology.

DAM, as use case owner, defined the **End User Requirements**, after holding meetings with plant managers and analysing the results of surveys addressed to the main technical managers of the company. Then the project partners involved in UC5 (UOC, NUN, KUM, ITI, SNG) worked together with DAM for properly stablishing the **technical requirements** including safety, security, privacy and performance requirements (D1.2) and planning the implementation, having in mind validation and verification methodologies.

### 4.5.4 Lessons learned

We learned several valuable lessons during the implementations of the UC5 DEMOS.

- Platforms and gateways should be installed to enable the transfer of data from installations and improve the quality and quantity of data available for AI models. On the other hand, training plant managers and middle managers as end-users, and recruiting new profiles with a data science background for the IT department is necessary to manage the system effectively.
- Changes in IT infrastructure on the end user side are expected too for adapting to the TRANSACT platform, for encompassing the compute continuum across three tiers, thus optimizing the performance and efficiency of AI models. In this sense, utilizing Kumori and Radiatus Platforms will help wastewater treatment plants deploy and manage AI models effectively and efficiently.
- As a consequence, investment in new infrastructure such as servers for AI, Kubernetes cluster, and space to host the received data is required to process large volumes of data in a timely and efficient manner. For instance, the new Safety-Critical Function for the prediction of industrial spills will be accompanied by additional AI-based value-added services at the edge tier, enabling alerts and warnings to be sent to technicians when a spill is close to reaching critical points in the plant, which would not be feasible with only the device layer, as it lacks the necessary resources for proper execution.

# 5 Summary

This deliverable has presented the initial version of the TRANSACT transition methodology to transform a local, stand-alone CPS into a safe and secure distributed safety-critical CPS solution. The methodology is based on three core focus areas: business, architecture, and organization, and four cross-cutting aspects: safety, performance, security and privacy, and regulatory and certification. The impact of the transition to the edge/cloud environment on each focus area has been presented together with possible solutions as worked out by other TRANSACT's work packages.

The transition has been evaluated in the context of all the TRANSACT's use-cases from the automotive, healthcare, maritime, and wastewater treatment domains. Each use-case is described the transition to the TRANSACT Reference Architecture, followed by the organization changes needed to support such a transition. Then planning and execution of the transition was described together with the current lessons learned from performing the use-case transition.

The second version of this deliverable will continue to consolidate the results of the use-case transition and update the TRANSACT transition methodology accordingly. In addition, the domain independent transition guidance will be created to facilitate and accelerate future transformations of safety-CPSs to support the European high-tech industry.

# 6 References

Abu Daia, A., Ramadan, R., & Fayek, M. (2018, 10). Sensor Networks Attacks Classifications and Mitigation. *Annals of Emerging Technologies in Computing*, pp. 28-43.

Ahlbrecht, A., & Durak, U. (2021). Integrating Safety into MBSE Processes with Formal Methods. *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)* (pp. 1-9). IEEE.

Akkermann, A., & et.al. (2022). *D1.2: Technical requirements and TRANSACT transition methodology commonalities.* TRANSACT.

Albert Benveniste, B. C.-B.-V. (2012). *Contracts for System Design.* HAL Inria.

Albrecht, A., & Bertram, O. (2021). Evaluating System Architecture Safety in Early Phases of Development with MBSE and STPA. *2021 IEEE International Symposium on Systems Engineering (ISSE)* (pp. 1-8). IEEE.

Alfonso, I. G. (2021). Self-adaptive architectures in IoT systems: a systematic literature review. *Journal of Internet Services and Applications*, 1-28.

Al-Fuqaha, A. a. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 2347--2376.

AMASS. (2017). *AMASS Project Deliverables.* AMASS .

AMASS. (2017). *Project Deliverables.*

Amazon AWS. (2023, 5 1). *Amazon GuardDuty: Protect your AWS accounts with intelligent threat detection*. Retrieved from Amazon GuardDuty: Protect your AWS accounts with intelligent threat detection: https://aws.amazon.com/guardduty/

Amazon AWS-Compute Optimizer. (2023, 05 03). *AWS Compute Optimizer*. Retrieved from AWS Compute Optimizer: https://aws.amazon.com/compute-optimizer/

Amazon AWS-Cost Management. (2023, 5 3). *What is AWS Cost Management?* Retrieved from What is AWS Cost Management?: https://docs.aws.amazon.com/cost-management/latest/userguide/what-is-costmanagement.html

Arjona, J., & et.al. (2022). *D2.1: Reference architectures for distributed safety-critical distributed cyber-physical systems v1.* TRANSACT.

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2016). Microservices architecture enables devops: Migration to a cloud-native architecture. *IEEE Software 33(3)*, 42-52.

Bass, L. a. (2015). *DevOps: A software architect's perspective.* Addison-Wesley Professional.

Burckhardt, S. a. (2014). Principles of eventual consistency. *Foundations and Trends in Programming Languages*, 1--150.

Burnes, B. (2004). Kurt Lewin and the planned approach to change: a re-appraisal. *Journal of Management studies*, 977--1002.

Buschmann, F., Henney, K., & Schmidt, D. (2007). *Pattern-Oriented Software Architecture, Volume 4: A Pattern Language for Distributed Computing.* Chichester, UK: Wiley.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture, Volume 1, A System of Patterns.* Chichester, UK: Wiley.

Chowdhury, N. M. (2009). Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, 20--26.

Cicchetti, A., Ciccozzi, F., Mazzini, S., Puri, S., Panunzio, M., Zovi, A., . . . Last, F. (2012). CHESS: A model-driven engineering tool environment for aiding the development of complex industrial systems. *27th IEEE/ACM International Conference on Automated Software Engineering*, (pp. 362-365).

Clemson, T. (2014, 11 18). *Testing Strategies in a Microservice Architecture*. Retrieved from Testing Strategies in a Microservice Architecture: https://martinfowler.com/articles/microservice-testing/

Douglass, B. P. (2005). *Real-time design patterns: robust scalable architecture for real-time systems.* Addison-Wesley Professional.

EASA. (2023, 4 25). *Specific Operations Risk Assessment (SORA)*. Retrieved from Specific Operations Risk Assessment (SORA): https://www.easa.europa.eu/en/domains/civil-drones-rpas/specific-category-civil-drones/specific-operations-risk-assessment-sora

Ernst, D. a. (2019). Rapid canary assessment through proxying and two-stage load balancing. *2019 IEEE International Conference on Software Architecture Companion (ICSA-C)* (pp. 116--122). IEEE.

EUCloudEdgeIoT.eu. (2023, 05 07). *Use cases, adoption drivers and barriers*. Retrieved from Use cases, adoption drivers and barriers: https://eucloudedgeiot.eu/use-cases/

Ferdous, R., Khan, F., Sadiq, R., Amyotte, P., & Veitch, B. (2013). nalyzing system safety and risks under uncertainty using a bow-tie diagram: An innovative approach. *Process Safety and Environmental Protection, 91*(1-2), 1-18. doi:10.1016/j.psep.2011.08.010

Fernandez, E. B. (2013). *Security Patterns in Practice: Designing Secure Architectures Using Software Patterns.* John Wiley & Sons.

GDPR. (2016). *General Data Protection Regulation.* The European Parliament and of The Council. Retrieved from https://gdpr-info.eu/

Goniwada, S. R. (2022). Observability. In S. R. Goniwada (Ed.), *Cloud Native Architecture and Design: A Handbook for Modern Day Architecture and Design with Enterprise-Grade Examples* (pp. 661–676). Berkeley, CA: Apress. doi:10.1007/978-1-4842-7226-8_19

Heinrich, R., van Hoorn, A., Knoche, H., Li, F., Lwakatare, L. E., Pahl, C., . . . Wettinger, J. (2017, April). Performance Engineering for Microservices: Research Challenges and Directions. *Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering Companion* (pp. 223–226). New York, NY, USA: Association for Computing Machinery. doi:10.1145/3053600.3053653

Hendriks, M., Basten, T., Verriet, J., Brassé, M., & Somers, L. (2016). A Blueprint for System-Level Performance Modeling of Software-Intensive Embedded Systems. *Software Tools for Technology Transfer 18(1)*, 21-40.

Hendriks, T., & et.al. (2022). *D3.1: Selection of concepts for end-to-end safety and performance.* TRANSACT project.

HIGHER COUNCIL FOR ELECTRONIC GOVERNMENT. (n.d.). *PORTAL ADMINISTRACIÓN ELECTRÓNICA.* Retrieved from https://administracionelectronica.gob.es/pae_Home/dam/jcr:80b16a91-75b1-432d-ab23-844a12aab5fc/MAGERIT_v_3_book_1_method_PDF_NIPO_630-14-162-0.pdf

HIPPA. (1996). *Health Insurance Portability and Accountability Act.* US Department of Health and Civil Services.

Humble, J., & Farley, D. (2010). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation.* Addison Wesley.

IEC. (2005). *Medical electrical equipment - Part 1: General requirements for basic safety and essential performance.* International Electrotechnical Commission (IEC).

IEC. (2010). *IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems.* IEC.

IEC. (2016). *Health software - Part 1: General requirements for product safety .* IEC.

IEC. (2016). *IEC 61511 Functional safety - Safety instrumented systems for the process industry sector.*

ISO. (2005). *ISO 17894:2005, Ships and marine technology.*

ISO. (2009). *ISO 26262 - Road vehicles—Functional safety. International Organization for Standardization / Technical Committee 22.* ISO TC 22.

ISO. (2011). *ISO 26262 - Road vehicles -- Functional safety.* ISO, Geneva, Switzerland.

ISO. (Under development). *ISO/FDIS 21448 - Road vehicles — Safety of the intended functionality.* ISO.

Kienhuis, B., Deprettere, E., Vissers, K., & Wolf, v. d. (1997). An Approach for Quantitative Analysis of Application-Specific Dataflow Architectures. *Proc. IEEE Int. Conf. on Application-Specific Systems, Architectures and Processors, ASAP'97* (pp. 338–349). IEEE.

Kircher, M., & Jain, P. (2004). *Pattern-Oriented Software Architecture, Volume 3: Patterns for Resource Management.* Chichester, UK: Wiley.

Kirichenko, A., & et.al. (2022). *D3.4: Solutions for end-to-end security and privacy for distributed CPS v1.* TRANSACT.

Kotter, J. P. (2012). *Leading change.* Harvard business press.

Lapalme, J., Theelen, B., Stoimenov, N., Voeten, J., Thiele, L., & Aboulhamid, E. (2009). Y-chart Based System Design: a Discussion on Approaches. In *Nouvelles approches pour la conception d'outils CAO pour le domaine des systems embarqués.* Université de Montreal.

Leveson, N. G. (2016). *Engineering a safer world: Systems thinking applied to safety.* Boston: The MIT Press.

*MAGERIT*. (2005). Retrieved from European Union Agency for Cybersecurity: https://www.enisa.europa.eu/topics/threat-risk-management/risk-management/current-risk/risk-management-inventory/rm-ra-methods/m_magerit.html

Mälardalen University. (2023, 5 10). *SafeCer - Safety Certification of Software-Intensive Systems with Reusable Components*. Retrieved from SafeCer - Safety Certification of Software-Intensive Systems with Reusable Components: http://www.es.mdu.se/projects/294-SafeCer___Safety_Certification_of_Software_Intensive_Systems_with_Reusable_Components

Martin, H. (November 2018). *AMASS Deliverable D6.8 "Methodological guide for cross/intra-domain reuse (b).* ECSEL Research and Innovation actions (RIA).

Microsoft. (2023, 04 20). *Review and compare common cloud operating models*. Retrieved from Review and compare common cloud operating models: https://learn.microsoft.com/en-us/azure/cloud-adoption-framework/operating-model/compare

Mortier, P., & et.al. (2022). *D4.2: Strategies for continuous updating and independent releasing v1.* TRANSACT.

Nasri, M., & et.al. (2022). *D3.3: Solutions for end-to-end safety and performance for distributed CPS.* TRANSACT.

OpenTelemetry. (2023, 05 03). *OpenTelemetry*. Retrieved from OpenTelemetry: https://opentelemetry.io/

Orban, S. (2016, 11 1). *6 Strategies for Migrating Applications to the Cloud*. Retrieved from 6 Strategies for Migrating Applications to the Cloud: https://aws.amazon.com/blogs/enterprise-strategy/6-strategies-for-migrating-applications-to-the-cloud/

Pop, P., & et.al. (2022). *D3.2: Selection of concepts for end-to-end security and privacy for distributed CPS solutions.* TRANSACT.

Renisch, S., Netsch, T., & et.al. (2022). *D4.1: AI services integrated per Use Case v1.* TRANSACT.

RTCA. (1992). *RTCA DO-178B. Software Considerations in Airborne Systems and Equipment Certification.* Radio Technical Commission for Aeronautics (RTCA).

Rushby, J. (2008). Runtime Certification. *International Workshop on Runtime Verification.* Budapest, Hungary: Springer.

SafeCOP. (2023, 5 10). *Safe Cooperating Cyber-Physical Systems using wireless communication*. Retrieved from Safe Cooperating Cyber-Physical Systems using wireless communication: http://www.safecop.eu/

*Safety Engineering*. (2022, February 14). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Safety_engineering

Sanden, v. d., Hi, Y., Aker, v. d., Akesson, B., Bijlsma, T., Hendriks, M., . . . Basten, T. (2021). Model-driven system-performance engineering for cyber-physical systems. *International Conference on Embedded Software (EMSOFT)* (pp. 11-22). ACM.

Schmidt, D., Stal, M., Rohnert, H., & Buschmann, F. (2000). *Pattern-Oriented Software Architecture, Volume 2: Patterns for Concurrent and Networked Objects.* Chichester, UK: Wiley.

Svorobej, S. a. (2020). Orchestration from the Cloud to the Edge. *The Cloud-to-Thing Continuum*, 61-77.

Szczygielski, L., & et.al. (2022). *D1.1: Use case descriptions, end user requirements, SotA and KPI's.* TRANSACT.

Technology, C. M. (2017). *Edge computing trend report.* Retrieved from https://lp.stratus.com/wp-content/uploads/stratus-edge-computing-trend-report-americas.pdf

Tyrrell, J., & Rogers, D. (2023). *The Applicability of Automotive Cybersecurity Standards.* SecureCAV. Retrieved from https://copperhorse.co.uk/wp-content/uploads/2023/01/The-Applicability-of-Automotive-Cybersecurity-Standards-2023.pdf

UNION, T. E. (5 April 2017). *REGULATION (EU) 2017/745 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL.* THE EUROPEAN PARLIAMENT AND THE COUNCIL OF THE EUROPEAN UNION.

Watson, R. (2010, 12 3). *Migrating Applications to the Cloud: Rehost, Refactor, Revise, Rebuild, or Replace?* Retrieved from Migrating Applications to the Cloud: Rehost, Refactor, Revise, Rebuild, or Replace?: https://www.gartner.com/en/documents/1485116

Wikipedia-AIOps. (2023, 5 3). *Artificial Intelligence Operations*. Retrieved from Artificial Intelligence Operations: https://en.wikipedia.org/wiki/Artificial_Intelligence_for_IT_Operations

Yu, M., Wu, C., & Tsung, F. (2019). Monitoring the data quality of data streams using a two-step control scheme. *IISE Transactions, 51*(9), 985-999.

Zeller, M. (2021). Towards Continuous Safety Assessment in Context of DevOps. *arXiv preprint arXiv:2106.07200*. Retrieved from arXiv preprint: https://arxiv.org/pdf/2106.07200.pdf

Zhang, G. a.-S.-R. (2019). GRIT: consistent distributed transactions across polyglot microservices with multiple databases. *2019 IEEE 35th International Conference on Data Engineering (ICDE)* (pp. 2024--2027). IEEE.

Zhao, Y. a. (2019). Edge computing and networking: A survey on infrastructures and applications. *IEEE Access*, 101213--101230.