1.4

Foundational Technology Layers

SYSTEM OF SYSTEMS

1.4 System of Systems

1.4.1. SCOPE

The System of Systems (SoS) technology layer represents the upper layer of ECS technology stack for digitalisation solutions. This technology layer emerges from the composition of embedded and cyber-physical systems (CPS), connectivity and distributed software platforms.

In the ECS domain, a constituent system of a System of Systems (SoS) is defined as a set of embedded hardware hosting software designed to perform a particular task or solve a specific problem. A constituent system can be distributed, but from a logical/conceptual perspective it is "contained" in one unit and it is autonomous and/or independent from the other constituent systems, (i.e. it shows managerial and operational independence from any other constituent system). The complexity of these constituent systems is rapidly increasing with the development of the underlying HW/SW technologies, as well as the rising demand by the users of these systems for functional and extra-functional requirements.

According to the definition developed by Mayer, 1998[1], SoS must satisfy five characteristics: (i) the operational independence of constituent systems; (ii) the managerial independence of constituent systems; (iii) geographical distribution; (iv) emergent behaviour; and (v) evolutionary development processes. A system that does not satisfy these characteristics is not considered an SoS.

For existing systems this independence results in composing or integrating systems that were not designed together, to perform a combined task besides their 'normal' task. SoS engineering aims at methods and architectures to resolve this, typically addressing resource sharing, and access to data and services. Model-based techniques for the design of an SoS can be used in a similar way as for regular systems; however, the integrating systems rely on different models and paradigms. Further methodology is needed to address that systematically.

Newly developed systems must be designed such that they are prepared for forms of SoS integration. Here, model-based techniques are useful, for example, in the application of AI techniques e.g. for learning dynamically how systems must work together while increasing the semantic level of interoperability. Research should address the development of methodology and standard patterns, interfaces and artifacts for SoS that complement current methodology for system design. Focus should be on the aspects that are specific for SoS such as the mentioned independence, and the integration into an SoS: discovery and use of services, the sharing of data and resources, the support for extra-functional properties and the very late binding, to be negotiated at interfaces. Such negotiation

[1] Architecting Principles for Systems-of-Systems, Mark W. Maier, Systems Engineering journal, John Wiley &Sons 1998

requires predictive models that support taking sharing decisions and build on interoperability and trustworthiness.

In modern hyper-connected digital solutions, systems rarely operate independently. On the contrary, the primary added value of these digital solutions is the cooperation between heterogeneous systems to solve more complex problems by exploiting the set of multi-technology, multi-brand and even multi-domain functionalities generated by the cooperation. While talking or reading, SoS is typically pronounced entirely “System of Systems”. An SoS emerges from the composition/integration of multiple systems to perform a task or reach an objective that none of the constituent systems can perform or reach on their own. In the SoS, each constituent system is considered a “black box”: it remains operational and managerial autonomous and/or independent, relying on its own hardware, software, and networking resources, and remaining focused on its own goals. At the SoS level, the SoS evolves with components, functions and purposes added, removed, and modified, leading to an increasing dynamicity and variability along their life cycle (a life cycle that is intertwined with the life cycles of constituent systems and potentially never finishes!). The SoS structure evolves with the addition or removal of the constituent systems, which always cooperate, coordinate, and adapt to achieve the SoS goals, providing additional features to the SoS as a whole and capabilities and functionalities unavailable in the constituent systems. Having an up-to-date inventory and real time monitoring of the SoS is challenging.

Like a nervous system – i.e., partially centralized, distributed and peripheral – a software integration infrastructure is a key element of an SoS. The nervous system has an architecture, and so does an SoS. The most common architecture approach in SoS is based on SOA (Service Oriented Architecture) and micro-service from edge to cloud. Splitting such architecture into three parts; Infrastructure, Integration platform and Solution implementation, provides a logical base for what can be shared and what will be company specific, among involved stakeholders, as shown in Figure 1.4.1.

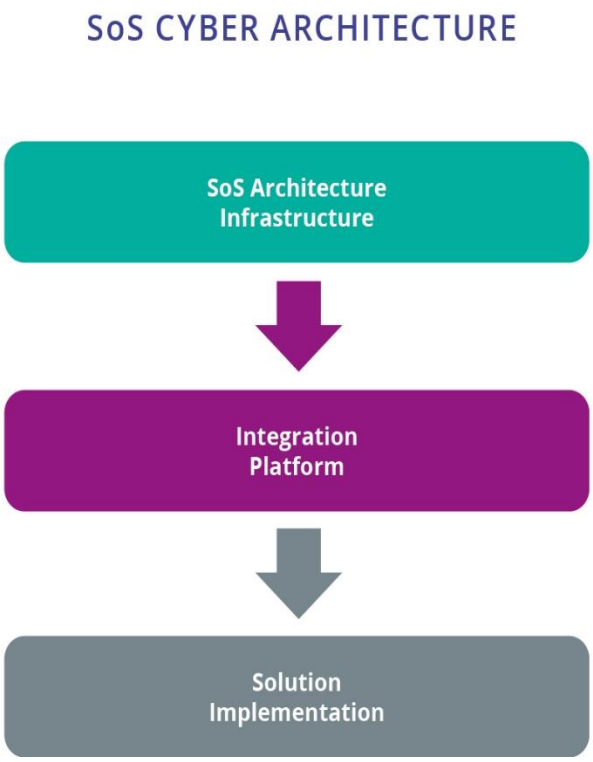

Figure 1.4.1. An SoS cyber architecture provides, based on SOA architecture, an infrastructure supporting fundamental service-oriented properties like, Look-up, Late binding and Loose coupling plus a number of support functionalities to build working solutions.

The SOA architecture infrastructure will provide e.g.:

- Fundamental SOA property support: Look-up, Late binding and Loose coupling
- Security support
- Interoperability support
- System of Systems integration support
- Basic engineering and operations support
- Model-based engineering support

Using the SoS infrastructure, SoS platforms are created by vendors and larger companies. SoS platforms will be used for solution-specific implementations, engineered, deployed and operated.

To create added value, an SoS needs to be trustworthy, and here e.g. end-to-end security issues have to be properly taken into account. A secure SoS should be able to defend against both deliberate attacks and accidental threats, and also its misuse. Moreover, it is not enough to ensure that each of the constituent systems is secure in the pre-deployment phase, but also that the evolved/composed/integrated SoS, whose exact composition may be not known in advance, is secure. Dynamic adaption to e.g. security or safety requirements and risks analysis should be considered over time in relation to emergent functionalities, properties and behaviors arising from the complex interactions among the

constituents of the SoS. New methodology and tools for risk and vulnerability assessment and threat modeling are needed. Artificial intelligence, machine learning and ontology/semantic-based approaches can complement each other for improved knowledge and decision-making processes in an SoS structure. AI/ML can make predictions based on experience or training, while ontologies/semantics provide information based on reasoning which also can optimise and accelerate machine learning processes.

It is unrealistic to imagine that a single SoS infrastructure could drive an entire market because, considering the interdisciplinarity and complexity required to develop them, very seldom will a single vendor be able to provide a complete end-to-end and domain-independent solution. However, platform "competition" will at least have to identify a set of European solutions that covers key vertical domains. For key European vertical domains an SoS has to address a multitude of cross sectorial requirements like e.g. security, safety, evolution, maintenance, trustworthiness. For example, security and safety certification issues both at component, system and SoS level should be properly addressed aiming at really mitigating risks/threats in competitive scenarios, while also considering the EU Cybersecurity Certification framework.

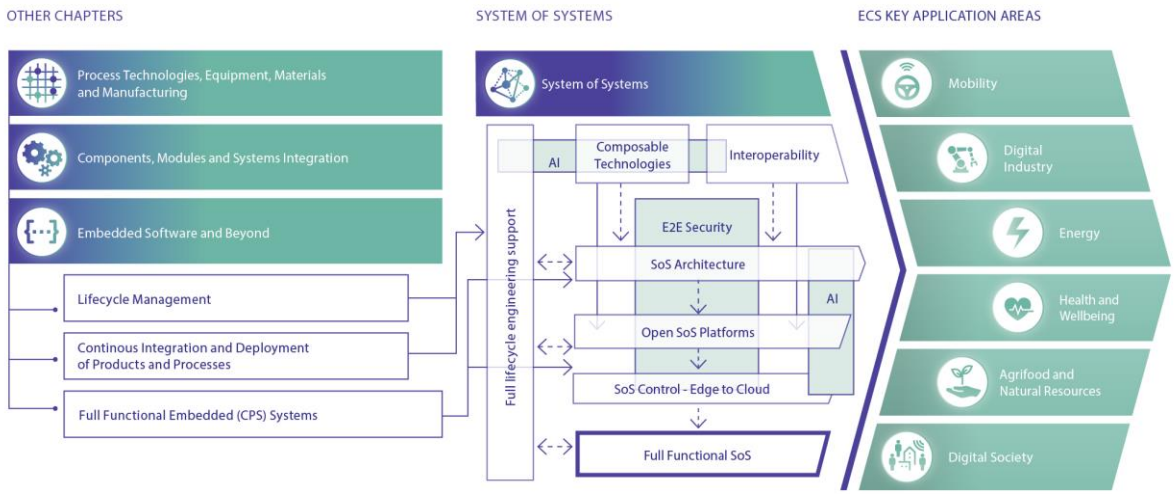

Figure 1.4.2 - Structure: System of Systems

1.4.2. Application breakthroughs

Improvements in SoS technology will have an impact on all ECS application areas. They will enable or support faster translation of ideas into economically viable solutions and might open new markets by further upscaling.

Examples of health and well-being application breakthroughs supported by SoS are:

- Interoperability of health data.
- Strengthening where and how healthcare is delivered, supporting home-based care.
- Supporting the clinical workforce and healthcare consumers to embrace technology-enabled care.

- High level of digital trust.
- Data security technology for interoperability between security hardware and software components.
- Improved integration and analysis of multimodal data.
- Integration platforms for embedded ultrasound, low-power edge computing, and AI and digital health.

For the mobility application area, the provision of EU capabilities within SoS will support breakthroughs regarding:

- Achieving the Green Deal for mobility with the 2 Zero goals of –37.5% CO_2 by 2030.
- Increased road safety through the CCAM[2] programme.
- Improve the competitiveness of the European industrial mobility digitalisation value chain.
- Ensuring inclusive mobility for persons and goods by providing mobility access to everyone, with a focus on special needs.

In the energy application domain, the provision of improved SoS capabilities and engineering efficiency will support breakthroughs regarding:

- Management of multivalent sector coupling (electricity, heating / cooling, mobility) for the future all-electric society.
- Supporting grid stabilisation by intermediate storage share of renewable energies, peak control or viability management for the increase of energy flexibility.
- Energy supply infrastructure for e-mobility, digital live, and industry 4.0.
- "Plug and play integration" of ECS into self-organised grids and multi-modal systems, real-time digital twin capability in component and complete system design (to simulate system behaviour).
- Significant reduction and recovery of losses (application and SotA-related).
- Increased functionality, reliability, and lifetime (incl. sensors & actuators, ECS HW/SW, semiconductor power devices, artificial intelligence, machine learning, monitoring systems, etc.).
- Safety and security issues of self-organised grids and multi-modal systems through smart edge devices and high-level IT security (resilient communications and trustworthy AI).
- Optimisation of applications and exploitation of achieved technology advances in all areas where electrical energy is consumed.
- Energy technologies in the circular economy approach: predictive and condition-based maintenance with repair and recycle capabilities.
- Aligning with standardisation of different energy systems.

In the industry and agrifood application domains, the provision of advanced SoS architectures, platforms and engineering automation will support the EU regarding:

- Intelligent control room systems to enable correlations between machine malfunctions and load parameters to be detected immediately, thereby enabling

[2] https://www.ccam.eu

maintenance work to be carried out early and on schedule, with a reduction in costly downtimes.

- Food industry imposes specific requirements (e.g. in food processing) that may take advantage of smart (bio-)sensing for high-quality monitoring to reduce the amount of water and chemicals used in such processes, and to prevent contamination.
- AI/machine learning (ML) and big data models must be devised and used to offer further intelligent decision-making and, whenever possible, should be employed directly at-the-edge for greater energy efficiency.
- Industrial IoT (IIoT) systems can provide the flexibility to tailor-make new products to help cope with ever-demanding diets.
- Remotely piloted autonomous unmanned aerial vehicles (UAVs), either flying alone or in swarms, to improve efficiency.
- Smart systems based on portable real-time pest disease diagnostics and monitoring platforms to provide rapid local and regional disease incidence alerts (georeferenced) – e.g. weather/climate information for predictive models providing risk assessments and decision support for Integrated Pest Management (IPM).
- IoT devices specialising in pests and disease measurements, such as insect traps and other systems based on image recognition or AI models.
- Large-scale and high-precision measurements of plant growth, architecture and composition.
- Winning the global platform game on various application sectors (that are currently strong) and in building effectively and, at a high level, outperforming applications and systems for industrial and business needs.
- Preparing for the 5G and beyond era in communications technology, especially its manufacturing and engineering dimension.
- Solving IoT and SoS cybersecurity and safety problems, attestation, security-by-design, as only safe, secure and trusted platforms will survive.
- Interoperability-by-design at the component, semantic and application levels.
- IoT configuration and orchestration management that allows for the (semi)autonomous deployment and operation of a large number of devices.
- Decision support for AI, modeling and analytics in the cloud and also in edge/fog settings.

In the digital society application domain, the provision of improved, robust, secure and interoperable connectivity will support the overall strategy regarding:

- Use energy and resources more efficiently within the existing installed base of industrial processes. Reduce or prevent waste.
- AI into the design, manufacturing, production and deployment processes, productivity can be improved.
- Collaborative product-service engineering, life cycle engineering: extending R&D to consider how products and systems will be integrated into the industrial service program of the company. This should possibly be enhanced by obtaining further knowledge to provide services for other similar products (competitors!) as well their own installed base.

- Remote engineering and operations, tele-presence: operating or assisting in operations of industrial systems from remote sites.
- Local and global services: organising services locally close to customers and centrally at vendors' sites.
- Edge/cloud solutions: implementing distributed service applications on effective edge-cloud systems.
- Full lifecycle tutoring: monitoring activities, level of stress and performance-oriented behavior during the product's life, from anticipating its end of life to properly handling its waste and recycling, including improved re-design for the next generation of products.

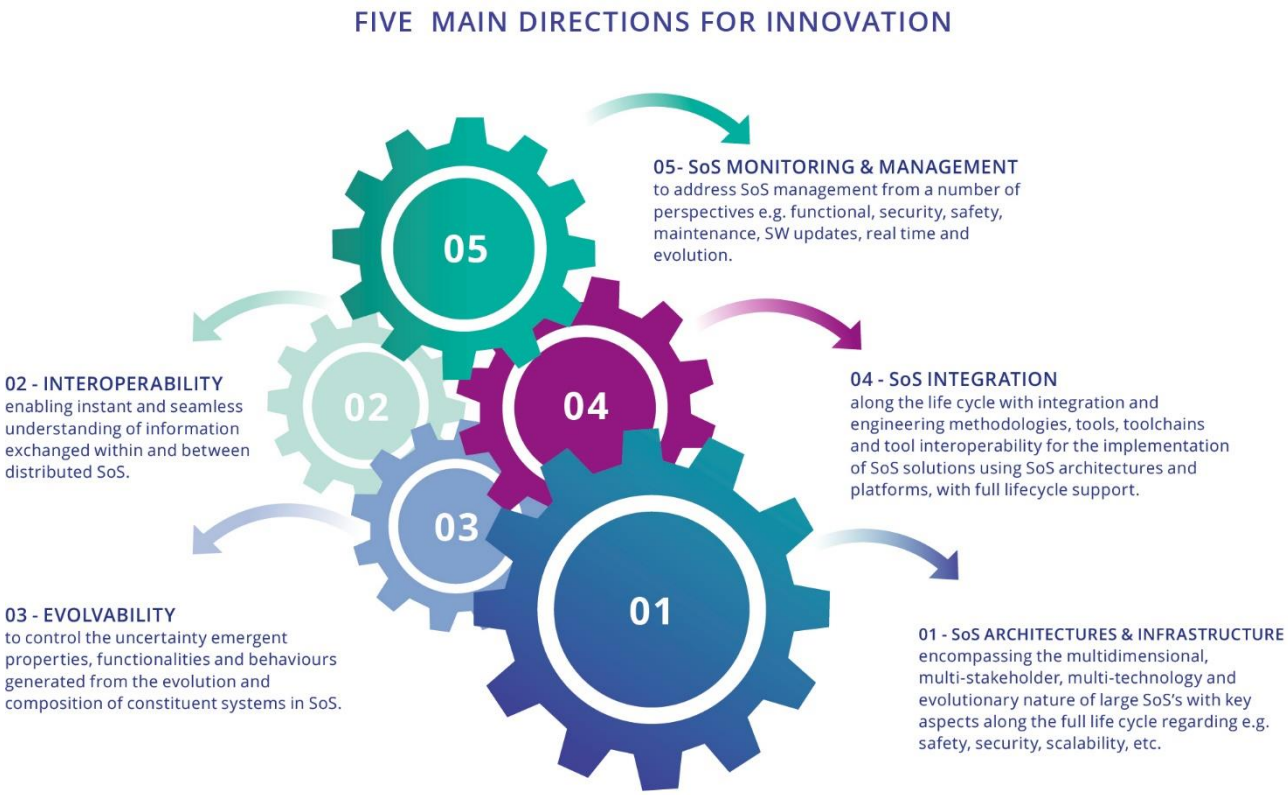

Figure 1.4.3 - Five Main Directions of Innovation (source: Eurotech).

1.4.3. MAJOR CHALLENGES

Five major challenges have been identified for the System of Systems domain:

- **Major Challenge 1**: Open SoS architecture and infrastructure.
- **Major Challenge 2**: SoS interoperability.
- **Major Challenge 3**: Evolvability of SoS composed of embedded and cyber-physical systems.
- **Major Challenge 4**: SoS integration along the life cycle.
- **Major Challenge 5**: SoS monitoring and management.

1.4.3.1. Major Challenge 1: Open SoS architecture and infrastructure

Open SoS architecture and infrastructure encompassing the multidimensional, multi-stakeholder, multi-technology and evolutionary nature of large SoS's with key aspects along the full life cycle regarding e.g. safety, security, scalability, engineering efficiency, real time performance, advanced control, QoS and distributed intelligence.

1.4.3.1.1. State of the art

SoS requires architecture and available infrastructure that encompasses the multidimensional, multi-stakeholder, multi-technology and evolutionary nature. Architecting a SoS is fundamentally different from architecting a single embedded system. The complexity of SoS architecting can be exemplified by the architecture of a complete smart city, with all its subsystem, stakeholders, technologies and evolutionary nature.

The current industrial state of the art consists in a couple of major commercial and proprietary information/communications/control/technology platforms offering industrial solutions for complex solutions from companies like e.g. Schneider Electric[3], Siemens[4], Bosch[5], Emerson[6], ABB[7], Advantech[8], AutoSAR. These proprietary digital platforms, at various levels, support design, implementation and operation of SoS architectures tailored for dedicated solutions in sectors including e.g. manufacturing, water and wastewater, minerals and mining, oil and gas, energy sectors, smart cities and automotive. It is also clear that the fundamental computer science basis for these products is quite old.

The current industrial state-of-the-art SoS's are based on extensions to existing major enterprise resource planning (ERP), manufacturing execution system (MES), supervisory control and data acquisition (SCADA), distributed control systems (DCS), robot controllers (RC), computer numerical controllers (CNC), and programmable logic controllers (PLC) products. Such extensions are mostly based on a central service bus concept. Such service buses are responsible for integrating legacy ERP, MES, SCADA, DCS, RC, CNC and PLC technologies from multiple vendors, at best. For emerging SoS application areas like autonomous driving, smart energy grid, smart agriculture and smart cities, the SoS technology is still in an emerging phase. Still Europe is the leading player for industrial automation and digitalisation, with a very strong position in the upcoming areas of autonomous driving, smart energy, smart agriculture and smart cities.

To take the next step, Europe and other regions have invested in a number of open SoS integration frameworks and platforms. A summary of these is shown in Figure [9].

[3] https://ecostruxure.schneider-electric.com/
[4] https://www.plm.automation.siemens.com/ global/en/webinar/iiot-the-next-big-digital- disruption/31921
[5] https://blog.bosch-si.com/bosch-iot-suite/
[6] https://www.emerson.com/de-de/ automation/operations-business- management/plantweb-digital-ecosystem
[7] https://ability.abb.com/
[8] https://www.advantech.com/resources/ news/advantech-launches-30-iiot-solutions- through-the-co-creation-model-and-the- wise-paas-platform-and-announces-a- large-scale-showcase-in-november
[9] Industrial Frameworks for Internet of Things: A Survey, IEEE System journal 2020

Most platform initiatives are based on Service Oriented Architectures (SoA) and microservices, which points towards a primary technology for such platforms. Although none of these open SoS platforms are currently in wide commercial usage, early examples can be found in small IoT solutions in various application areas. Major industrial usage remains rare, but MES-level adoption can be found in automotive production, for example.

Open architectures and reference implementations such as e.g. the IMC-AESOP approach[10], Eclipse Arrowhead[11], Eclipse Basyx[12], FiWare[13], PERFoRM30[14] are providing a link to standardisation activities in national and international innovation platforms. In the automotive domain, AutoSAR is developing in the microservice direction. Such standardisation activities are e.g. DIN Specification 91345[15] "Reference Architecture Model for Industry 4.0" (RAMI 4.0), the "Industrial Internet Architecture" (IIA), the "High Level Architecture of the Alliance for Internet of Things Innovation", the "NIST Big data Reference Architecture", to name just a few.

Europe has strongly invested in large projects that have delivered open platforms for the implementation of solutions-based on SoS platforms[16]. Considering the platforms referred to in 1.4.4, Eclipse Arrowhead, AUTOSAR, FiWare and BaSyx have all been developed with substantial European leadership and partnership.

[10] https://link.springer.com/ book/10.1007/978-3-319-05624-1

[11] https://www.taylorfrancis.com/ books/e/9781315367897

[12] https://www.eclipse.org/basyx/

[13] https://www.fiware.org/

[14] https://www.taylorfrancis.com/ books/e/9780429263316

[15] https://www.en-standard.eu/din-spec-91345-reference-architecture-model-industrie-4-0-rami4-0/

[16] From Internet of Things to System of Systems – Market analysis, achievements, positioning and future vision of the ECS community on IoT and SoS, P Azzoni, Artemis 2020.

FEATURES	ARROWHEAD	AUTOSAR	BASYX
Key principles	SOA, local automation clouds	Run-time, electronic control unit (ECU)	Variability of production processes
Realtime	Yes	Yes	No
Run-time	Dynamic orchestration and authorisation, monitoring, and dynamic automation	Run-time environment (RTE) layer	Run-time environment
Distribution	Distributed	Centralise	Centralise
Open source	Yes	No	Yes
Resource accessibility	High	Low	Very low
Supporters	Arrowhead	AUTOSAR	Basys 4.0
Message patterns	Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl,
Transport protocols	TCP, UDP, DTLS/TLS	TCP, UDP, TLS	TCP
Communication protocols	HTTP, CoAP, MQTT, OPC-UA	HTTP	HTTP, OPC-UA
Third-party and legacy systems adaptability	Yes	Yes	Yes
Security manager	Authentication, authorisation and accounting Core system	Crypto service manager, secure onboard communication	--
Standardisation	Use of existing standards	AUTOSAR standards	Use of existing standards

FIWARE	IOTIVITY	LWM2M	OCFW
Context awareness	Device-to-device communication	M2M, constrained networks	Resource-oriented REST, Certification
No	Yes (IoTivity constrained)	No	No
Monitoring, dynamic service selection and verification	No	No	No
Centralise	Centralise	Centralise	Centralise
Yes	Yes	Yes	No
High	Medium	Medium	Low
FIWARE Foundation	Open Connectivity Foundation	OMA SpecWorks	Open Connectivity Foundation
Req/Repl, Pub/sub	Req/Repl, Pub/sub	Req/Repl	Req/Repl
TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS	TCP, UDP, DTLS/TLS, SMS	TCP, UDP, DTLS/TLS, BLE
HTTP, RTPS	HTTP, CoAP	CoAP	HTTP, CoAP
Yes	No	No	No
Identity manager enabler	Secure resource manager	OSCORE	Secure resource manager
FIWARE NGSI	OCF standards	Use of existing standards	OCF standards

Figure 1.4.4 - Open SoS integration frameworks and platforms[17]

For the cross-domain requirements on e.g. security, safety, evolution application and business critical details need to be considered. As an example thereof security takes on new dimensions in the case of SoS. In this Chapter, security is taken to be the ability to prevent leaking information and to prevent the taking over of control of the SoS by agents not being part of the SoS, but also the guarantee that no hostile party can prevent the sharing of essential information between the systems comprising the SoS. Several security aspects require attention. First, the level of security of each individual system requires attention: the lower bound to security of an SoS is determined by the system with the lowest security level, and by the link with the lowest security level between systems ("weakest link in the chain"). Thus, requirements like Quality, Reliability, Safety and Cybersecurity at the system and SoS levels are covered in Chapter 2.4 of this ECS-SRIA.

[17] Industrial Frameworks for Internet of Things: A Survey, C. Paniagua and J. Delsing, in IEEE Systems Journal, vol. 15, no. 1, pp. 1149-1159, March 2021, doi: 10.1109/JSYST.2020.2993323.,

However, combining a very large number of systems in an SoS can result in a lower overall security level than the lowest security level of any system in the SoS: an attacker can now combine and relate information from two or more systems, which in combination can reveal new information. Segmentation of an SoS is thus of large importance both for architecture but also for actual implementation and maintenance and updates of the life cycle.

Systems must not only defend against and monitor possible attacks, but also measures must be taken avoiding infection by intrusions from one system to the other systems in the SoS. Only this way resilience and cybersecurity can be attained.

The spectrum of systems making up an SoS includes both systems in the cloud, where security can be closely monitored as in e.g. data warehouses, and systems at the edge. Edge systems pose a higher level of cyber insecurity because of the limited resources often available at the edge (e.g. power, communication bandwidth).

Another aspect is SoS safety. Here architectures and platforms need to address safety from various application domains and their respective standards and regulations. More details related to the ECS application domain requirements on Quality, Reliability, Safety and Cybersecurity at the system and SoS level are covered in Chapter 2.4 of this ECS-SRIA.

1.4.3.1.2. Vision and expected outcome

This Major Challenge is expected to lead to a set of EU-strategic open SoS architectures and infrastructures. From such infrastructures, vendor and large company platforms can be devised, capable of supporting a wide range of solutions in diverse fields of applications covering the ECS supply chain and supporting efficient life cycle management.
This requires new and improved infrastructure technologies comprising:

- Robust design- and run-time infrastructure enabling integration and orchestration of functionalities from edge to cloud.
- Infrastructure support for multi-level security, security management, safety, safety management, scalability, engineering efficiency, real-time performance, closed loop and digital control, QoS, distributed intelligence and other key application area requirements.
- Interoperability to legacy SoS technology (“to-the-past”).
- Interoperability to existing and emerging IoT and SoS technologies and infrastructures (“to-the-future”).
- Support for autonomous operation, resilience, fail-over and mitigation management.
- Enabling SoS flexibility.
- Engineering support through model based engineering and associated domain specific languages (c.f. Chapter 2.3 Architecture and Design: Method and Tools).

The expected outcome is a set of EU-strategic open source platforms. These infrastructure platforms should have long-term governance with industry-friendly licensing schemes such as e.g. Eclipse ECL2. Such platforms should also have strong EU-based value chain support.

To cope with increasing complexity, the SoS engineering community is constantly researching improvements to its engineering processes. To ensure the complexity remains manageable, modeling approaches are used. The challenge in these approaches is to find the right level of abstraction that also allows for reasoning about the system while still containing sufficient information to connect to lower levels of abstraction, often by generating code for some underlying implementation platform.

It is not only that the complexity of the SoS is growing, but there are also extra-functional requirements that are often interlinked playing an increasingly important role. For example, with the demand for greater speed and the concomitant energy consumption, systems are often required to process information quickly but within a tight energy budget. These two requirements are clearly conflicting and choosing the right trade-off can be a balancing task. With the realisation that the planet's resources are limited, as exemplified in the European Green Deal, also comes the demand for resource conservation, resulting in more and intertwined requirements, putting greater demand on the dynamic and evolution capabilities of both the SoS architectures and the architecture tools that support the complexity of SoS.

Some important but necessary aspects of SoS architecture are:

- Security and trust,
- Safety,
- Robustness,
- Composability,
- Evolution,
- Interoperability (data exchange and data models),
- Engineering tools and procedures,
- Energy consumption,
- Unified environmental data model,
- Environmental footprint optimisation,
- Resilience.

1.4.3.1.3. Key focus areas

The key focus is how SoS architectures and their open infrastructure can enable and leverage important and necessary aspects while also enabling efficient adaptation to specific application solutions.
To support EU strategic autonomy, a small number of SoS architectures and integration platforms should be driven by EU-based ecosystems. Important features that such platforms should provide include:

- Robust SoS infrastructure capable of supporting a wide range of solutions in diverse fields of applications,
- SoS infrastructure and associated engineering tools and toolchains that support the complete engineering processes in both design- and run-time, including SoS critical aspects such as e.g. security, safety and risk mitigation,
- Suitable and adaptable engineering processes, with associated training material for solution engineering.

- Methods for the handling of (often wide-spread) legacy elements, e.g. as black box models.

2.4.3.1. Major Challenge 2: SoS interoperability

SoS interoperability enables instant and seamless understanding of information exchanged within and between networked and distributed systems.

2.4.3.1.1. State of the art

Interoperability in the SoS domain is a rising problem for cost-effective engineering and operation of systems of embedded and cyber-physical systems (see Figure 1.4.).

There is currently no industrial solution to this problem. Academia and industry are experimenting with approaches based on, for example, ontologies[18], machine learning[19], model-based engineering and open semantic frameworks[20]. Even if no clear winning approach can be identified based on current research results, growing interest can be noted for e.g. ontology, data and model driven approaches. Automating considerable parts of interoperability engineering (design-time and run-time) will improve SoS operational quality and will be very cost efficient.

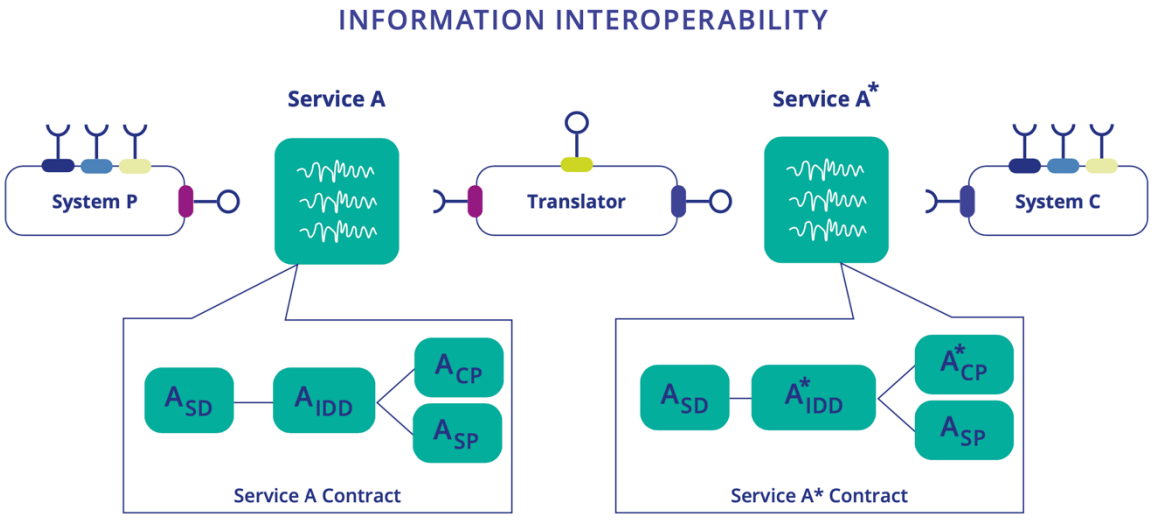

Figure 1.4.5 - Information interoperability between two service providers can be addressed by means of translators. The design of such translators for the payload information is currently necessary to provide for every situation where interoperability is requested.

[18] Extended semantic annotations for generating translators in the arrowhead framework, F Moutinho, L Paiva, J Köpke, P Maló - IEEE Transactions on Industrial Informatics, 2017

[19] Interoperability and machine-to-machine translation model with mappings to machine learning tasks, Jacob Nilsson, Fredrik Sandin and Jerker Delsing, IEEE INDIN 2019

[20] An open semantic framework for the industrial Internet of Things, S Mayer, J Hodges, D Yu, M Kritzler, F Michahelles - IEEE Intelligent Systems, 2017

2.4.3.1.2. Vision and expected outcome

To enhance EU leadership and sovereignty in the field of SoS based on embedded and cyber-physical systems, autonomous information translation for understanding is a necessity. Some integration platforms already focus on protocol and information interoperability (Derhamy, 2018[21]). To enable the cost- and time-efficient engineering of solution integration and extension, their updates and upgrades over the lifecycle is crucial. Therefore, SoS integration platforms have to provide mechanisms for dynamic and instant information translation across the ontologies and semantics used the individual constituent systems of the SoS.

2.4.3.1.3. Key focus areas

To facilitate substantial cost reductions for SoS solutions, autonomous and dynamic mechanisms for information translation are required. Such mechanisms should cover:

- Translation between standardised data models (e.g. ISO 10303[22], ISO 15926[23], BIM[24]).
- Translation between different implementations of standardised data models.
- Automated data model translation.
- Autonomous data model translation.
- Efficient and flexible engineering procedures.
- Engineering tools that support the complete engineering process in both design- and run-time.
- Support for key automation requirements.
- Automated translation engineering e.g. AI-driven, model based code generation.

3.4.3.1. Major Challenge 3: Evolvability of SoS composed of embedded and cyber-physical systems

SoS intrinsic nature is dynamic and SoS evolve with components, functions and purposes added, removed, and modified along their continuously evolving lifecycle (a life cycle that potentially never finishes). An SoS has properties, behaviours and functionalities that mainly do not reside in any constituent system but in the SoS as a whole and allow the SoS to achieve its own goals. These properties, functionalities and behaviours at the SoS level emerge in a direct relationship to the SoS evolution and, being potentially unknown, must be monitored and managed, i.e., detected, identified, understood and controlled. Because the results of the composition/evolution could be uncertain, SoS architectures and platforms, open and proprietary in conjunction with the proper engineering support

[21] H. Derhamy, J. Eliasson and J. Delsing, "IoT Interoperability—On-Demand and Low Latency Transparent Multiprotocol Translator," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1754-1763, Oct. 2017, doi: 10.1109/JIOT.2017.2697718.

[22] https://www.iso.org/standard/66654.html

[23] https://15926.org/home/

[24] https://en.wikipedia.org/wiki/Building_information_modeling

(methods and tools), should provide solutions to manage the evolution and resulting uncertainty emergent properties, functionalities and behaviours.

3.4.3.1.1. State of the art

Evolvability and composability are multi-dimensional aspects of SoS evolution, that affect SoS architectures, properties, functionalities and behaviours from different perspectives (evolvability, trust, interoperability, scalability, availability, resilience to failures, etc.). Primarily, composability must ensure the persistence of the five major attributes that characterise an SoS (see Maier, 1998[25]). Vertical (hierarchical) composability provides the most common way to build an SoS that is typically structured in a hierarchical stack composed of adjacent layers. Vertical composability has to deal with the different abstraction levels of the stack layers, adopting aggregation and de-aggregation solutions as references to compose the constituent systems of the SoS. Architectural composability, on the other hand, is fundamental for SoS design, specifically when critical requirements such as trust or safety must be satisfied (see Neumann 2004[26], for an extensive report on trustworthy composable architectures).

In the hierarchical structure of an SoS, the constituent systems that are at the same level typically compose horizontally (in parallel or serially), potentially generating competing chains of constituent systems. Serial composability represents a critical issue for all properties that are not automatically transitive, such as trust. Indeed, the inclusion of AI in embedded and cyber-physical systems increases the required level of trust, as well as the uncertainty of the results of the composition process (see, for example, Wagner, 2015[27]).

When the constituent systems expose high-level services, service composability allows for the creation and provision of new added-value services at the SoS level, combining the resources, functionalities, information, etc., of the constituent systems. Eventually, the engineering process deals with composability, enabling it by design (already present from the constituent systems level) and/or managing it during the operations of the SoS, to address the dynamic nature of SoS in time (run-time composability associated with evolutionary development and potential emergent properties, behaviours, and functionalities).

3.4.3.1.2. Vision and expected outcome

The dynamic nature of SoS is based on the composition and integration of embedded and cyber-physical systems. The role of composability is to ensure that functional and extra-functional properties (scalability, quality of service (QoS), performance, reliability, flexibility, etc.), and the functionalities and behaviours of the constituent systems are preserved in the

[25] Architecting Principles for Systems-of-Systems, Mark W. Maier, Systems Engineering journal, John Wiley & Sons 1998

[26] Peter G. Neumann, "Principled Assuredly Trustworthy Composable Architectures", DARPA, Computer Science Laboratory SRI International EL-243, 333 Ravenswood Ave, Menlo Park, California 94025-3493, USA.

[27] Wagner, M.; Koopman, P. A Philosophy for Developing Trust in Self-driving cars. In Road Vehicle Automation 2; Meyer, G., Beiker, S., Eds.; Lecture Notes in Mobility; Springer: Cham, Switzerland, 2015; pp. 163–171, doi:10.1007/978-3-319-19078-5_14.

SoS or combined in a predictable and controlled way, even when the constituent systems recombine dynamically at run time. The lack of solutions to dynamically manage composability represents one of the limitations hindering the market uptake and diffusion of SoS.

Composability should be conceived as a quality of SoS that makes them future proof: (i) the relationships between components that allow them to recombine and assemble in different and potentially unlimited architectural combinations, and ensure and exploit the re-use of components; (ii) the extension of components lifetime within the evolution of the SoS during its lifecycle; (iii) the possibility that SoS will easily evolve, adapting to new contexts, new requirements and new objectives; and (iv) the simple substitution of faulty, inadequate and/or new components with a minimal impact for the SoS, guaranteeing the survival and sustainable evolution of the SoS. Composability also has to consider cross-sectorial requirements like e.g. security, safety, trust, evolution.

Ensuring composability at the SoS level represents a very challenging goal, potentially generating serious and critical consequences, and even preventing the integration of the SoS. Indeed, considering a property that characterises a constituent system with a certain attribute, it is not guaranteed that the same property will characterise it when the constituent system becomes part of an SoS. In addition, if the property is still present, it is not guaranteed that it will have the same attribute. The same applies to the constituent system's functionalities, behaviours, etc.

As a consequence, one major effect of the composition, integration, evolution of the constituent systems is the evolution of the SoS, with emergent properties, functionalities and behaviours which generate uncertainty. For instance, when SoS evolution affects security, safety, trust, interoperability, scalability, availability, resilience to failures, etc., the impact of the uncertainty could potentially be extremely serious.

The inclusion of AI in SoS increases the importance of composability, because it may significantly increase the complexity, variability and fuzziness of composability results. AI enables a completely new category of applications for SoS. Therefore, the availability of specific solutions for the validation, verification and certification of SoS composed of AI-based systems is a critical requirement.

Predicting and controlling the effects of composability is also fundamental for the interaction of humans along the SoS lifecycle and the protection of human life should be ensured in SoS evolution. Uncontrolled and unmonitored composition could lead to deviations from expected behaviours or generate unknown emergent behaviours potentially dangerous for humans. The increasing level of automation introduced by SoS accentuates this criticality, and will require that humans still intervene in cases of emergency (for example, in automated driving).

The solutions proposed to manage composability will also have to support the multi-domain nature of SoS, the presence of different stakeholders in its lifecycle, and the different regulations and standards that apply to these domains. From an engineering perspective, emergent behaviours require that the development of SoS, applying composability, is

evolutionary and adaptive over the SoS continuously evolving lifecycle, which potentially may never finish. In fact, SoS architectures and platforms, jointly with the proper engineering support, will have to provide solutions to control the uncertainty of evolvability and ensure adequate countermeasures.

3.4.3.1.3. Key focus areas

Since the technology base, and the organisational and human needs are changing along the SoS lifecycle, SoS architecting will become an evolutionary process based on composability. This means: (i) components, structures, functions and purposes can be added; (ii) components, structures, functions and purposes can be removed; or (iii) components, structures, functions and purposes can be modified as owners of the SoS experience and use the system. In this sense, the dynamically changing environmental and operational conditions of SoS require new architectures that address the SoS goal(s), but thanks to composability will also evolve to new system architectures as the goal(s) change.

Evolution in SoS is still an open research topic requiring significant effort and the key areas of research and innovation include:

- Methods and tools for engineering evolvability of systems of embedded and cyber-physical systems, e.g. AI driven, model based (c.f. Chapter 2.3. Architecture and Design: Methods and Tools).
- Evolutionary architectures in systems of embedded and cyber-physical systems.
- Evolvable solutions for trust, availability, scalability, and interoperability.
- Evolvable solutions capable for managing resulting uncertainty emergent properties, functionalities and behaviours, including resilience to failures.
- Evolvability in systems of cyber-physical systems through virtualisation, e.g. digital twins.
- Methods and tools to manage emergencies in embedded and composable systems of cyber-physical systems.
- Service-based vertical and horizontal evolvability to enable high-level, and potentially cross-domain, interoperability of embedded and cyber-physical systems.

4.4.3.1. Major Challenge 4: SoS integration engineering along the life cycle

Integration and engineering methodologies, tools, tool chains and tool interoperability are fundamental to enable the implementation of SoS solutions using SoS architectures and platform technologies, supporting the whole lifecycle.

4.4.3.1.1. State of the art

Europe is a world leader in the engineering of systems of systems. Major European companies such as Siemens, ABB, Schneider, Valmet, Bosch and Endress+Hauser, together with a number of large system integration companies (e.g. Afry, VPS and Midroc), offer complete engineered solutions, making Europe the leading global automation SoS provider.

Most solutions for embedded and cyber-physical systems engineering are based on highly experienced teams of engineers supported by a heterogeneous set of SoS engineering tools. For example, engineering practice and associated standards provide design-time solutions based on, for example, IEC 61512 (ISA 88)[28], IEC 62264 (ISA95)[29], IEC81346[30], ISO 10303, ISO 15924, IEC 62890[31]. The proposed Industry 4.0 architectures, formally provided by the DIN specification 91345 RAMI 4.0, have not yet made it into industrialised engineering procedures, or associated tools and toolchains. Many of these standards investigate updates of their data models to be based on e.g. ontologies and semantic web.

The current state of the art engineering of SoS remains more an art than a well-structured integration and engineering process. For example, the analysis of emergent behaviour of very large SoS is still at a foundational research level in academia.

4.4.3.1.2. Vision and expected outcome

The European leadership in application fields such as distributed automotive and industrial automation and digitalisation indicates some excellent skill sets in the art of SoS engineering. In the short to medium term, Europe has to transfer these skills into systematic and robust engineering procedures supported by integrated and efficient tools and tool chains. Please also refer to Chapter 2.3 and Chapter 1.3

This is expected to lead to engineering processes, tools and tool chains covering the whole life cycle that to significant extent can be automated while supporting integration between multiple stakeholders, multiple brand and multiple technologies. To support such integration and engineering efficiency, solution quality and sustainability concrete advancements like in Figure 1.4.6[32] will become necessary. The advancement may include integration and engineering process capabilities like:

- Flexible integration and engineering procedures.
- Model-based engineering procedures and tool,
- Supported by interoperable and flexible toolchains.
- Integration of multi-stakeholder engineering processes.
- Automation of substantial parts of the integration and engineering process.

[28] https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa88
[29] https://www.isa.org/standards-and-publications/isa-standards/isa-standards-committees/isa95
[30] https://en.wikipedia.org/wiki/IEC_81346
[31] https://webstore.iec.ch/preview/info_iec62890%7Bed1.0%7Den.pdf
[32] Urgese, G.; Azzoni, P.; van Deventer, J.; Delsing, J.; Macii, A.; Macii, E. A SOA-Based Engineering Process Model for the Life Cycle Management of System-of-Systems in Industry 4.0. *Appl. Sci.* 2022, *12*, 7730. https://doi.org/10.3390/app12157730

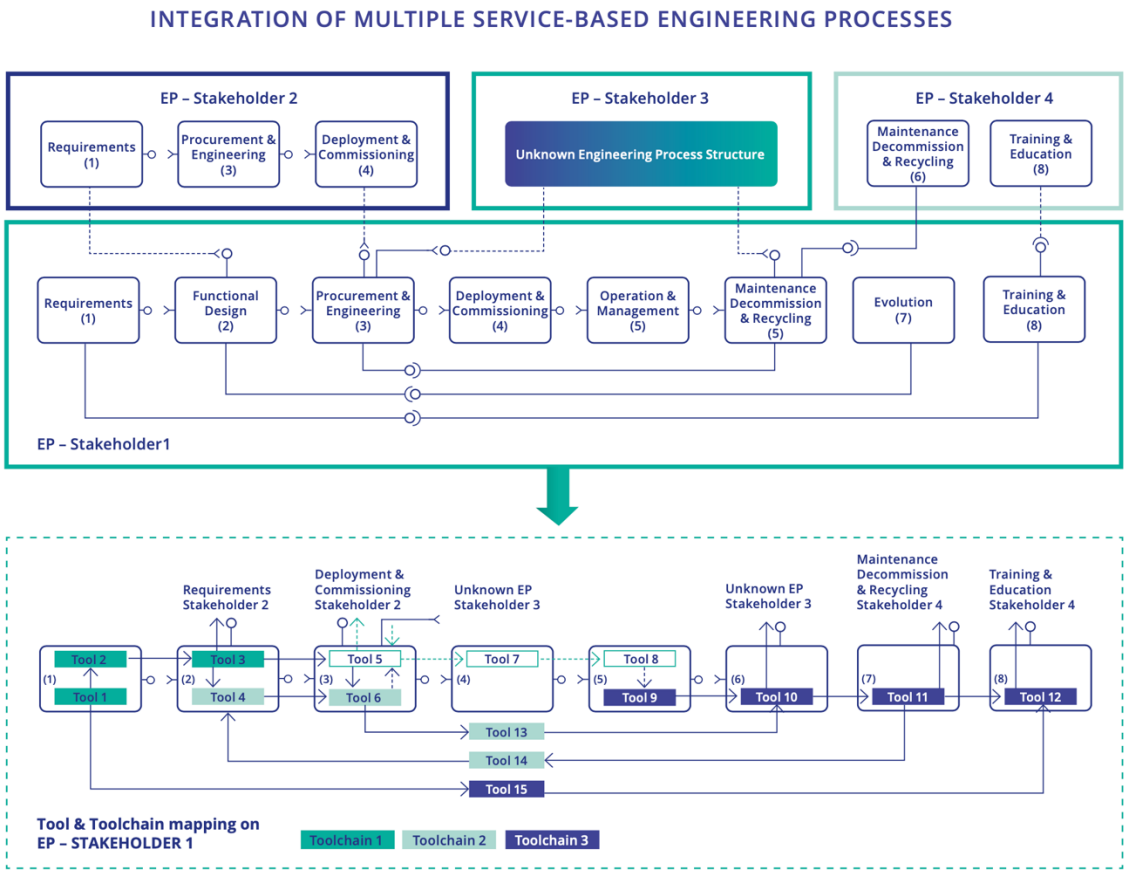

Figure 1.4.6 - Example of conceptual service-oriented view on the integration of multiple service-based engineering processes (EP) from different stakeholders, including the engineering process mapping with integrated toolchains and tools.

4.4.3.1.3. Key focus areas

In support of EU leadership and sovereignty in the field of SoS engineering the ambition is to invest in a small number of architecture infrastructures and their associated tools, tool chains and engineering processes. Strong European-based ecosystems should be created and provided with long-term governance also connected to open source. These engineering processes, methodologies, tools and toolchains shall provide, for example:

- Efficient, flexible and automated engineering processes.
- Model-based engineering.
- AI-supported engineering tools and processes
- Engineering tools supporting the complete engineering process along the system's lifecycle.
- Support for key automation requirements.
- Automated engineering, low-code engineering.
- SoS traceability and analytics interoperable with engineering tools and tool chains.
- SoS evolution impact analysis.
- Automated testing validation and verification (TV&V) along the life cycle.

In particular, SoS TV&V introduces a significant challenge, mainly due to complexity, to the effects of composition (not always known in advance) and to SoS dynamic evolution over time. For SoS, a full TV&V procedure prior to deployment is practically unrealistic. Typically,

the TV&V of each constituent system is asynchronous and independent of SoS, challenging the SoS TV&V with feature and capability evolution. For this motivation, a structured framework methodology and tools is necessary to demonstrate an appropriate level of confidence that the feature under test is present in the SoS, and that no undesirable behaviours are also present. This implies a need for end-to-end system capabilities metrics and, according to the flow of data, control and functionalities across the SoS, additional test points, recurring tests and AI-empowered data collection. This analysis should be considered to address changes in the constituent systems and to receive feedback on anomaly behaviours.

5.4.3.1. Major Challenge 5: SoS monitoring and management

Management of SoA-based SoS will require structured and scalable approaches to status monitoring and strategies and methodologies to address SoS management from a number of perspectives e.g. functional, security, safety, maintenance, SW updates, real time and evolution. It is clear that a high degree of automation and autonomy need to be introduced to keep quality up and cost down. Management and monitoring of SoS play a particularly crucial role when application faults result in personal injuries or property and environmental damages: e. g. critical infrastructures (electric grid, rail network) connected (semi)-autonomous automotive systems, medical monitoring, industrial plant control systems, robotics and automatic pilot avionics.

5.4.3.1.1. State of the art

Current industrial state of the art for monitoring and management of SoS reflects back to monitoring and management of production automation, energy grid automation and similar. Looking closer we find a plethora of commercial application solutions tailored to specific applications. Many of these are very application and site specific and “home brewed”.

There is a wide set of different realms to be monitored and managed, ranging from modern production processes, smart grids, smart cities, automotive traffic networks, only to name some of them. Furthermore, for each of these realms their operation requires different competences and groups within an organization, and it follows different guidelines. Some examples are:

- Status of operation
- Safety
  Real time performance
- Real time monitoring of sensors and actuators, incl. fault detection and isolation
  Validation of signals (using redundancies created by the data network of the SoS)
- Control
- Maintenance
- Assets
- Security

These aspects do have more or less known and understood relationships/dependencies which also will change in run-time. This provides a monitoring and management landscape which is very heterogeneous and dynamic. As a result, management methodologies need to be supported by automated and autonomous control technology. In this context and in view of limiting data traffic in an SoS, synchronisation of systems becomes a major goal as it is directly linked to the stability of the management system.

In addition, management of complex cyber-physical SoSs must address scalability (i.e. to deal with a variable number and interconnection of systems and automated control loops) and network phenomena (such as computation/communication latency, data loss). Looking at the aspect of data management, open SoS control platforms should ensure information security management, SoS scalability, SoS engineering efficiency and also SoS real-time performance.

A wide set of tools is available, each supporting one or a few of these dimensions. In most cases these tools mandate underlying information sources and data models, which sometimes correlates with current major industrial standards like ISA95, BIM, ISO 15926 and ISO 10303.

In summary a very complex and heterogeneous landscape of, to a large extent non-interoperable, tools and methodologies with no or little capacity to be integrated across SoS dimensions.

5.4.3.1.2. Vision and expected outcome

The emerging closer digital integration of industrial and societal functionalities and domains requires SoS integration and associated monitoring and management in very complex and heterogeneous environments. The current state of the art is far from efficiently enabling this. Such enabling will require closer cooperation and integration between several levels of the ECS domain stack and society policies and governance. An example thereof is the integration and functional interoperability between open and proprietary SoS architecture and implementation platforms which reside under different jurisdictions. Here solution requirements on lifecycle and evolution as well need to be considered.

5.4.3.1.3. Key focus areas

To advance towards the vision technology and knowledge steps are required regarding:

- Monitoring and management strategies and architectural concepts in OT-IT environments.
- Methodologies and technologies for monitoring and management of multiple and interrelated SoS dimensions.
- Processes and technology for life cycle monitoring and management over SoS dimensions and society borders.
- Engineering support, tools and methods, for monitoring and management strategy and policy implementation
- Tools for control system analysis of SoS.

- Considering humans, environment and the economy in the loop.
- Engineering tools and methods for SoS control design.
- Reduction of communication effort, variable structure, variable number of systems in control loops.
- Control system testing, validation and verification (TV&V) in design and run-time.

1.4.4. TIMELINE

The following tables illustrate the roadmaps for System of Systems.

MAJOR CHALLENGE	TOPIC	SHORT TERM (2025–2029)	MEDIUM TERM (2030–2034)	LONG TERM (2035 and beyond)
Major Challenge 1: SoS architecture and open integration platforms	**Topic 1.1**: Robust SoS integration platform capable of supporting a wide range of solutions in diverse fields of applications	Architectures and associated implementation platforms with sufficient granularity and engineering support for efficient implementation of real-world Industry 4.0 solutions	Architectures and implementation platform with support for a wide set of autonomous operation e.g. M2M business execution	Architectures with support for self-X e.g. self-healing, self-extension etc.
	Topic 1.2: integration platform and associated engineering tools and toolchains that support the complete engineering process in both design- and run-time, including SoS critical aspects such as security, safety and risk mitigation	Preliminary lifecycle support for extra-functional requirements, such as energy consumption, environmental impact that translates into maintainability, sustainability, etc.	Full lifecycle support for extra-functional requirements, such as energy consumption, environmental impact that translates into maintainability, sustainability, etc.	Autonomous management of functional and non-functional dimensions
	Topic 1.3: suitable and adaptable engineering processes with training material for solution engineering	Hardware and software tools, methodology and training material suited for training of professionals and students at university level	Model based engineering support proving partial engineering automation of solutions	Automated SW engineering for most solution engineering stages.
Major Challenge 2: SoS interoperability	**Topic 2.1**: Translation between standardised data models e.g. ISO 103030, ISO 15926, BIM, …	Translation technologies enabling translation of standardised data models and demonstrated at TRL 5-7	Fully autonomous translation	

	Topic 2.2: Translation between different implementations of standardised data models	Translation technologies enabling translation of different implementations of standardised data models and demonstrated at TRL 5-7	Full cross-domain interoperability	
	Topic 2.3: automated data model translation	Technologies and tools for automating the engineering of data model translations	Fully automated information translation	
	Topic 2.4: autonomous data model translation	Technology and tools for enabling autonomous data model translation in runrime	Fully autonomous translation	
Major Challenge 3: Evolvability of SoS composed of embedded and cyber-physical systems	**Topic 3.1**: methods and tools for engineering evolvability of systems of embedded and cyber-physical systems	Persistence of operational independence, managerial independence, geographic distribution, emergent behavior and evolutionary development	Full predictable and controllable composition of functional and extra-functional properties	Full predictable and controllable composition of functional and extra-functional properties, also covering dynamically recombining SoS
	Topic 3.2: evolutionary architectures in systems of embedded and cyber-physical systems	Modular and evolvable architectures.	Evolvability and composability by design	Automated evolvability and composability analysis in design time and run-time
	Topic 3.3: evolvable solutions for trust, availability, scalability, and interoperability.	Modular frameworks addressing trust, availability, scalability and interoperability-	Modular frameworks and open integration platforms addressing e.g. trust, availability, scalability, interoperability	Open modular frameworks and integration platforms addressing e.g. trust, availability, scalability, interoperability, evolvability, composability
	Topic 3.4: evolvable solutions capable for managing resulting uncertainty emerging properties, functionalities and behaviours, including resilience to failures	Technology frameworks supporting self-adaptability	Failures resilience at SoS level	Automated management of uncertainty and resilience to failures.
	Topic 3.5: evolvability in SoS supported by virtual engineering (e.g. digital twins)	Virtualisation of IoT and edge services based on open SoS architectures and platforms	Automated virtualisation of IoT and edge services based on open SoS architectures and platforms	Dynamic and scalable virtualisation of IoT and edge services based for run-time optimisation on open SoS architectures and platforms
	Topic 3.6: methods and tools to manage emergencies in embedded and composable SoS.	Technology frameworks supporting emergent self-adaptability	Automated technology and tools supporting emergency self-adaptability	Autonomous technology and tools supporting emergency self-adaptability

	Topic 3.7: service-based vertical and horizontal evolvability to enable high-level, and potentially cross-domain, evolvability of SoS	Open services enabling technology and data evolvability cross-domain	Open services and integration platforms enabling technology and data evolvability cross-domain	Open services and integration platforms enabling automated technology and data evolvability cross-domain
Major Challenge 4: SoS integration along the life cycle.	**Topic 4.1**: efficient and flexible engineering processes	SoA-inspired engineering processes, toolchains and tools	Engineering support for SoS emergent behaviours	Engineering support for emergent behaviours of very large SoS
	Topic 4.2: model-based engineering	Partial automated generation of SoS software using model-based engineering and AI tools	Full automated generation of SoS software using model-based engineering	Model based engineering support providing engineering automation for very complex SoS solutions
	Topic 4.3: engineering tools supporting the complete engineering process along the system's lifecycle	Engineering tools enabling run-time engineering	Multi-stakeholders and multi-domains automated engineering process	Highly automated solution engineering in a multi-stakeholders and multi-domains SoS environment
	Topic 4.4: support for key automation requirements	SoS engineering process and tools partial support for fundamental automation requirements like e.g. real time, security, safety	SoS engineering process and tools full support for fundamental automation requirements like e.g. real time, security, safety	
	Topic 4.5: automated engineering	Automation of SoS software engineering from requirements to deployment	Technologies and tool for highly automated design time control analysis in SoS environments	Technologies and tool for autonomous run-time control analysis in SoS environments
	Topic 4.6: automated testing validation and verification (TV&V)	Automated and runtime SoS TV&V for parts of the engineering process	Automated runtime SoS TV&V for the entire engineering process	Autonomous runtime SoS TV&V
Major Challenge 5: SoS monitoring and management	**Topic 5.1**: Monitoring and management strategies and architectural concepts in OT-IT environments	Real time monitoring and management of evolving OT.IT environments	Scalable monitoring architecture applicable to large scale SoS	SoS integration platforms including scalable, and manageable monitoring capabilities
	Topic 5.2: Methodologies and technologies for monitoring and management of multiple and interrelated SoS dimensions	Functional, security and safety interrelations monitoring and management	Manageable monitoring architecture of multiple SoS dimensions	SoS management based on multi-dimensional monitoring

	Topic 5.3: Processes and technology for life cycle monitoring and management over SoS dimensions	Approaches to life cycle monitoring and management for multiple SoS dimensions. Like e.g. functionality, security and safety	SoS monitoring architecture along its life cycle	SoS integration platforms supporting SoS monitoring and management evolution along its life cycle