Chapter 3
The IMC-AESOP Architecture for Cloud-based Industrial Cyber-Physical Systems

Stamatis Karnouskos, Armando Walter Colombo, Thomas Bangemann, Keijo Manninen, Roberto Camp, Marcel Tilly, Marek Sikora, François Jammes, Jerker Delsing, Jens Eliasson, Philippe Nappey, Ji Hu, Mario Graf

Abstract A coherent architectural framework is needed to be able to cope with the imposed requirements and realize the vision for the industrial automation domain. Future factories will rely on multi-system interactions and collaborative cross-layer management and automation approaches. The Service-Oriented Architecture paradigm empowered by virtualisation of resources acts as a lighthouse. More specifically by integrating Web services, Internet technologies, Cloud systems, and the power of the Internet of Things, we can create a framework that has the possibility of empowering the seamless integration and interaction among the heterogeneous stakeholders in the future industrial automation domain. We propose here a

Stamatis Karnouskos, Ji Hu, Mario Graf
SAP, Germany e-mail: stamatis.karnouskos@sap.com,ji.hu@sap.com,mario.graf@sap.com

Armando W. Colombo
Schneider Electric & University of Applied Sciences Emden/Leer, Germany e-mail: armando.colombo@schneider-electric.com,awcolombo@technik-emden.de

Thomas Bangemann
ifak, Germany e-mail: thomas.bangemann@ifak.eu

Keijo Manninen
Honeywell, Finland e-mail: keijo.manninen@honeywell.com

Roberto Camp
FluidHouse, Finland, e-mail: roberto.camp@fluidhouse.fi

Marcel Tilly
Microsoft, Germany, e-mail: marcel.tilly@microsoft.com

Marek Sikora
Honeywell, Czech Republic e-mail: marek.sikora@honeywell.com

François Jammes, Philippe Nappey
Schneider Electric, France e-mail: francois2.jammes@schneider-electric.com,philippe.nappey@schneider-electric.com

Jerker Delsing, Jens Eliasson
Luleå University of Technology, Sweden, e-mail: jerker.delsing@ltu.se,jens.eliasson@ltu.se

service architecture that attempts to cover the basic needs for monitoring, management, data handling and integration etc. by taking into consideration the disruptive technologies and concepts that could empower future industrial systems.

3.1 Introduction and Vision

The future industrial automation systems are expected to be complex system of systems [7, 4] that will empower a new generation of today hardly realizable applications and services. This will be possible due to several disruptive advances [10], as well as the cross-domain fertilisation of concepts and the amalgamation of IT-driven approaches in the traditional industrial automation systems. The factory of the future will rely on a large ecosystem of systems where collaboration at large scale [16] will take place. This is only realizable due to the distributed, autonomous, intelligent, pro-active, fault-tolerant, reusable (intelligent) systems, which expose their capabilities, functionalities and structural characteristics as services located in a "Service Cloud". Multidisciplinary in nature, the factory appears as a new dynamic Cyber-Physical infrastructure, which links many component systems of a wide variety of scales, from individual groups of sensors and mechatronic components to e.g. whole control, monitoring and supervisory control systems, performing, e.g. SCADA, DCS and MES functions. The resulting combined systems are able to address problems which the individual components alone would be unable to do, and to yield management, control and automation functionality that is only present as a result of the creation of new, "emergent" information sources, and result of cooperation, composition of individual capabilities, aggregation of existing and emergent features [3].

Today, plant automation systems are composed and structured by several domains viewed and interacting in a hierarchical fashion following mainly the specifications of standard enterprise architectures [18]. However, with the empowerment offered by modern Service-Oriented Architectures, the functionalities of each system or even device can be offered as one or more services of varying complexity, which may be hosted in the Cloud and composed by other (potentially cross-layer) services, as depicted in Fig. 3.1. Hence, although the traditional hierarchical view coexists, there is now a flat information-based architecture that depends on a big variety of services exposed by the Cyber-Physical Systems [1], and their composition. Next generation industrial applications can now rapidly be composed by selecting and combining the new information and capabilities offered (as services in the Cloud) to realise their goals. The envisioned transition to the future Cloud-based industrial systems is depicted in Fig. 3.1.

Several efforts so far were directed towards defining structural and architectural aspects of production management systems. The most popular and applied in practice are the definitions set up within the ISA-95 / IEC 62264 standard (www.isa-95.com). Typically, today's production systems (factory and process) are structured in a hierarchical way in a 5-level hierarchical model. IEC 62264 addition-

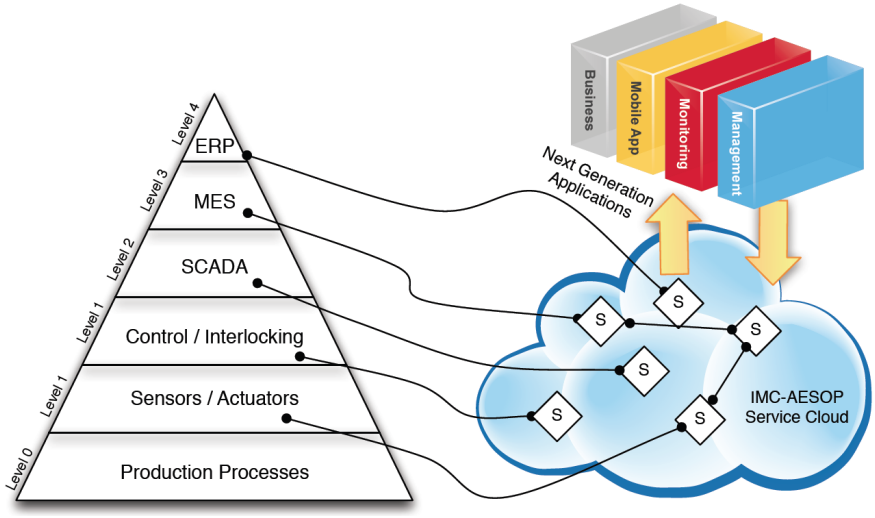

Fig. 3.1 Industrial Automation Evolution: Complementing the traditional ISA-95 automation world view (pyramid on the left side) with a flat information-based infrastructure for dynamically composable services and applications (right side)

ally defines a manufacturing operations management model, implicitly represented by real installations. The standard defines functions mainly associated to level 3 and level 4, objects exchanged, their characteristics and attributes, activities and functions related to the management of a plant, but does neither say anything about the implementations (solutions) hosting a specific operation nor the precise assignment to one of the levels 2, 3 or 4. Realisations depend on individual customer needs and the solution provider's strategies. For instance Maintenance Management operation may typically be assigned to a Computerized Maintenance Management System (CMMS), a Manufacturing Execution System (both being typical Level 3 solutions), to an Enterprise Resource Planning or a Distributed Control System.

Operations can be assigned to specific manufacturing operations management areas, i.e. Production Operations Management, Quality Operations Management, Maintenance Operations, Management Inventory Operations Management. Having a look into these areas, individual activities can be identified and executed within single or distributed source(s). These functions can be implemented using different technologies. Based on these considerations, one can identify distinct directions towards the organisational structure of a production site and the topological or architectural characteristics. From the organisational point of view, the business is structured similar to the levels and operations defined by IEC 62264 – or better to argue in the opposite way: the standard is following what has been developed over the past years.

Topological and architectural characteristics are driven by user or application needs with respect to latest, proofed or acceptable technological capabilities. The

major idea is to establishing a service Cloud fulfilling today's requirements for production management systems. The composition of the Cloud is targeted towards the suitability of supporting IEC 62264 operations and activities. Thus, keeping the organisational aspects established in today's production systems, the migration to future service based architecture exploiting the capabilities inherent to SOA is approached [5].

Within the following sections, considerations taken when designing the new architecture [12, 15] are given to prove the inclusion of the user and application needs. This is the basis for defining services distributed within the Cloud or between the Cloud and those associated to real physical instances within the architectural framework. Finally, directions of further progress within and behind latest developments are discussed. This architecture is considered as a prelude in realising the vision of Cyber-Physical Systems [1], especially in relation to the 4th industrial revolution (referred to as Industrie 4.0 in Germany [8]).

3.2 Design Considerations

In order to design the architecture, a set of use-cases and their requirements, as well as concepts and technology trends have been considered. In this section, we focus on the resulting potential directions that may play a key role for the design of the architecture. More specifically, these are:

- Asset Monitoring
- Backward / Forward Compatibility
- Creation of Combinable Services & Tools
- Cross-network Dynamic Discovery
- Cross-layer Integration & Real-time Interaction
- Infrastructure Evolution Management
- Interoperability and Open Exchange Formats
- System Management
- Mobility Support
- Process Monitoring & Control
- Provision of Infrastructure Services
- Real-time Information Processing
- Real-world Business Processes
- Scalability
- Service Lifecycle Management
- System Simulation
- Unique Asset Identification

3.2.1 Asset Monitoring

The monitoring of assets is of key importance especially in a highly complex heterogeneous infrastructure. In large scale systems [16] it will be practically impossible to do effective information acquisition with the traditional methods i.e. often pull the devices for their status. The more promising approach is to have an event driven infrastructure coupled with Service-Oriented Architectures. As such any device or system will be able to provide the information it generates (data, alarms etc.) as an event to the interested entities.

Considering that there exist basically two major kinds of Monitoring Methods (i.e. feature-based and model-based), the application of the IMC-AESOP approach allows performing both in an individual and also in a combinational manner. On one side, devices and systems are able to expose feature-based monitoring indexes as services, i.e. monitoring indexes generated by the application of "relational" functions between sensor signals and information exposed as Web services and the intrinsic characteristics, both structural and behavioural, of the systems and process behind. On the other side, the model-based orchestration approach that is an inherent component of the SOA-based IMC-System (Intelligent Monitoring and Control System) facilitates the creation of new monitoring indexes, this time, model-based monitoring indexes, that appear as a result of the composition and orchestration of monitoring services exposed by the orchestrated devices and systems. In this case, the rules to orchestrate or compose the monitoring services, follow the process model functions and are offered as a service usually by a constellation of underlying devices and systems. Emergent behaviours obtained by the orchestration and composition of monitoring services are not a rarity and constitute a clear proof of application of the SoS-paradigm.

Due the close relationship between asset monitoring, control and process monitoring, the components required to create a large scale system event-driven architecture are mostly the same; the main difference resides in that assets extend to anything that can create value for the company, and while this includes the machines that are monitored and controlled, it also extends to personnel, material, energy, and to other aspects of the machines that are used in processes.

3.2.2 Backward / Forward Compatibility

The future industrial infrastructure is expected to be constantly evolving. As such it is important to be (i) backwards compatible in order to avoid breaking existing functionality and (ii) forward compatible i.e. feature interfaces and interactions as flexible as possible with possible considerations on future functionality and models to come.

While designing components (devices, gateways, mediator) and their architecture one has to consider regarding backward compatibility:

- State-of-the-art and seriously emerging technological trends
- Use of the most used standard technologies (de facto standards in industry) to address a broad range of applications and technical equipment being on the market today.
- Some commonalities can be monitored like concepts for device descriptions or integration mechanisms into SCADA/DCS
- Going not too many steps ahead to taking the targeted user from where he is today. Changing too many paradigms, or "making the step too large and complex", will probably cause acceptance problems within the addressed user community.
- New engineering approaches have to smoothly integrate the existing ones.
- Taking advantage from past standardisation efforts (e.g. device profiles) will reduce new investments for the establishment of new technology.

While designing components (devices, gateways, or mediator) and their architecture one has to consider regarding forward compatibility:

- Focus on most promising open standard technologies
- Focus on "living" standards that are actually used, not on "sleeping" ones that were once defined and never updated or really used in real environments
- Consideration of hardware capabilities that may have an effect on the architecture and technologies e.g. single- versus multi-core processor systems, single- or multi-stack architecture, multi-purpose controllers versus single-purpose etc.
- Software update and download capabilities (ideally with complete lifecycle management)

3.2.3 Creation of Combinable Services & Tools

The trend in software applications is rapid development of them by combining existing functionality in a mash-up way. It is expected that this trend will also empower next generation industrial applications. Since often the development of such functionality is very much task-oriented, new tools are needed to be developed that ideally can be easily combined in a larger system.

Combinable services and tools should be used. Consider as an example the Unix command line utilities whose functionality can be piped and generate the desirable outcome. Similarly several tools (proprietary or not) should be combined (orchestrated) and their functionality could provide input to mash-up applications and services. Industrial application development may be greatly eased by following this approach. Typical examples of the design goal here are the functionalities offered by the XML Pipeline i.e. connecting XML processes such as XML transformations and XML validations together, Complex Event Processing (CEP) driven interactions, service composition, Yahoo! Pipes etc.

In very large-scale distributed systems, it is desirable to program applications and describe processes at the highest possible level of abstraction. Each service-enabled device abstracts a real piece of equipment functionality or information processing

capability in such a way that it can be used as a building block when describing a higher-level process. The ability to combine atomic services into higher-level composite services, which are themselves abstracted and exposed as services, is one of the fundamental benefits of SOA. A systems-of-systems approach, where services can be composed of other services enables creation mixed systems where high-powered devices, e.g. servers, can provide a complex service composed of a number of underlying services provided by resource-constrained devices. Service orchestration methods, such as BPEL, can be used for managing this.

3.2.4 Cross-network Dynamic Discovery

Large scale process control infrastructures typically span multiple sites or multiple sub-networks. For plant operation, maintenance, and engineering, zero-configuration networking can provide the tools for managing a device, or service, throughout its operational lifetime. For instance a new device or system may automatically announce its presence and allow cross-layer optimisations during its operation. The goal is towards real-time awareness of all cyber-physical parts in the network and their capabilities.

Existing approaches embedded at protocol level e.g. DHCP or direct IPv6 usage, DPWS, WS-Discovery, etc. or indirect approaches such as network scanning can assist towards identifying connected assets and services. A discovery strategy still needs to be investigated. For instance, when a new device is installed: will the device announce itself or will it rely on a master device scanning the network for changes? Filtering and caching will also have to be considered for large scale system discovery (e.g. filtering on service type and/or scope . . .) consisting of heterogeneous networks, e.g. wired and wireless sensor networks etc.

The use of dynamic device and service discovery, especially over sub-network boundaries, can create a substantial amount of network traffic. The use of protocols and mechanisms based on polling can have a large impact on a network's performance, which for wireless networks, e.g. WLAN or sensor networks need extra precautions when deploying. Event-based protocols, when combined with caching mechanisms can help mitigate the performance impact from a large scale deployment of dynamic device and service discovery.

Automatic device and service discovery is one key feature for large scale wireless sensor networks to be maintainable due to the potentially very large number of devices. cross-layer discovery mechanisms helps services and systems outside the sensor network to access devices and services inside the networks, thus enabling interoperability and usability. For battery powered devices is also vital the discovery protocol of choice is lightweight enough so that the node's energy consumption can be minimized.

3.2.5 Cross-layer Integration & Real-time Interaction

cross-layer integration refers to direct communication between different layers in the ISA-95 model (www.isa-95.com) e.g. a production planning system reading sensors in order to estimate when additional supplies are needed. The aim here is the optimisation at architectural and functional levels of the logical and physical network architectures.

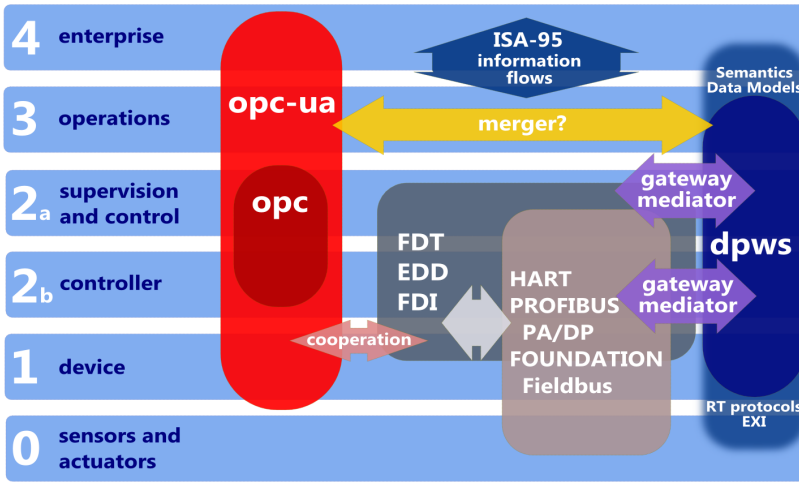

Fig. 3.2 ISA-95 application levels, and relevant current & emerging technologies

To achieve this, several actions must be taken e.g. (i) identify activities and information flows, and their performance requirements (hard real time, soft real time, right time, etc.), (ii) investigate technologies that can be used to meet the identified performance requirements, (iii) determine standard ways for representing non-functional requirements, such as Quality of Service (QoS), and propose solutions where standards do not exist, (iv) determine optimal network infrastructure patterns etc.

3.2.6 Infrastructure Evolution Management

Although industrial infrastructures have up to now been designed for the long run e.g. with 15-20 years lifetime in some cases, in the future they are expected to be more often updated for increased reliability, take advantage of the latest technologies and provision of new functionality. Being technology agnostic of the future advancements, the main challenge is to be able to design today an infrastructure that will be easy to manage and evolve in conjunction with technology.

Better said, the key questions posed are:

- how can one design today the perfect legacy system of tomorrow?
- How can today's functionalities be reused and integrated to tomorrow's infrastructure with minimal effort?
- how can we make sure the transition/migration is smooth and with least impact on key factors such as cost, downtime, maintenance, business continuity etc.?

Typical example scenario is the automatic software update service on all devices in the network, for security and safety reasons. Another example of the infrastructure evolution is the migration as envisioned in the IMC-AESOP project [5]. It is expected that several migration paths will exist, and each of those paths will additionally have its own number and type of migration steps.

3.2.7 Interoperability and Open Exchange Formats

As next generation systems will be highly collaborative and will have to share information, interoperability via open communication and standardized data exchange is needed. System engineering of interoperable systems has profound impact on their evolution, migration and future integration with other systems [6, 4, 3]. There are two dimensions of interoperability to be considered (i) cross-level i.e. communication between the various levels of the enterprise system, from the plant-floor level up to the enterprise level [18], with for example systems like ERP or MES; and (ii) cross-domain: the case of multi-disciplinary systems where devices and systems of different domains must communicate.

3.2.8 System Management

The next generation factory systems will be composed of thousands of devices with different hardware and software configurations. There will be a need to automate as much as possible primarily the monitoring part, decision making and also the soft-control of such systems; hence management is of key importance. Management should hide increasing complexity and should provide seamless interaction with the underlying infrastructure such as making it possible to dynamically identify devices, systems and services offered by the infrastructures. It should be possible to do software upgrades and mass reprogramming or re-configuration of whole systems. Additionally (remote) visualisation of the real infrastructure is a must, as it will give the opportunity of better understanding and maintaining it.

Management of a heterogeneous network of devices and systems is crucial for the feasibility of a Cloud-based large-scale architecture. The use of devices and systems from different manufacturers adds requirements such as flexibility and extensibility to a management system . Using a common communication architecture

will mitigate some of these constraints. Scalability and robustness are also important factors when the number of managed (SOA-enabled) devices increases. A management system must be able to effectively support hundreds of thousands of devices with different software and hardware platforms from different vendors.

3.2.9 Mobility Support

In the factory of the future where modern automation systems are in place, the operators are not bound to specialized control centres but will be able to control and monitor the processes in the shop floor using mobile HMIs. This enables access to real time measurements and statistics at any time and location. Mobility support also enables monitoring of mobile machinery (automatic loaders, robots, vehicles etc.).

Mobility will need to be considered towards different angles

- support for mobile devices e.g. being used as HMIs
- support for mobility of devices i.e. where devices are themselves mobile and the implications of this
- support for mobile users and interaction with static and mobile infrastructure
- support for mobility of services e.g. where services actually migrate among various infrastructures and devices following e.g. user's profile wishes.

3.2.10 Process Monitoring & Control

Although the topology and structure of processing plants are usually fixed, a challenge is still given by the large size of a typical plant, which may have thousands of actuating, sensing and controlling devices. This makes the design, deployment, management, and maintenance of a process monitoring and control system significantly more difficult. A SOA-based approach should address the key challenges in order to enable maximum system flexibility through its entire lifecycle.

Here one has to consider several megatrends in process automation industry. For instance process automation companies are following trends and adopting technologies from the IT sector such as virtualisation and Cloud computing which are being leveraged in deployments of large-scale process monitoring and control systems / DCS systems. Additionally, exploitation of wireless communication further decreases wiring costs and enables to deploy more devices (sensors, actuators, controllers), and thus, extend the scope and enhance quality of process automation functions.

However, as the automation technology increases in complexity and sophistication, operations professionals are faced with increased volumes of data and information they have to process. In addition, end-users: i.e. industrial operating companies experience reductions in skilled resources. Together with data overload and growing

safety, security and environmental concerns, this means that fewer people in operations teams must respond faster, handle more complex processes and make better decisions with bigger consequences.

Process monitoring and control should be eased by the architecture. More specifically it should be possible to easily decouple the device-specific aspects from the more abstract process ones, and enable the various stakeholder to fulfil their roles. In potentially federated infrastructures, processes may need to be coordinated to avoid side-effects that could hamper production lines or avoid intervene with other business goals.

3.2.11 Provision of Infrastructure Services

In the future complex infrastructure envisioned, it cannot be expected that all devices (especially resource constrained ones) and systems will always implement the full stack of software that may assist them in interacting with other systems and their services. As such auxiliary infrastructure services are needed that will enable collaboration of systems and exchange of information.

Therefore generic services need to be designed and put in place. This implies:

- assumption about generic services hosted at devices and more complex systems
- generic services provided by the infrastructure itself and assurance that devices and systems can interact with them,
- dynamic discovery of additional (customized) services and easy interaction with them.

As an example the Infrastructure services should enable (i) peer to peer device/system collaboration (horizontal collaboration) and (ii) device to business collaboration (vertical collaboration) .

What is envisioned and wanted is that the infrastructure enables the horizontal and vertical collaboration and integration [13]. Several requirements that would enable easy integration and collaboration have already been identified, especially when this concerns devices in systems. Basically the infrastructure services should enable collaboration, and therefore we need to consider issues such as: dynamic collaboration, extensibility, resource utilisation, description of objects (interface), semantic description capabilities, inheritance/polymorphism, composition/orchestration, pluggability, service discovery, (web) service direct device access, (web) service indirect device access (gateway), brokered access to events, service life-cycle management, legacy device integration, historian, device management, security and privacy, service monitoring [14].

3.2.12 Real-time Information Processing

Real-time information processing is a broader topic. We have to distinguish between the technical challenges about hard real-time processing which is about predictive and deterministic behaviour on a device and processing of information with low latency from data sources (e.g. sensors) to the consumer, such as dashboard (or operator in front of dashboard), or database etc.

For next generation applications to be able to react timely, apart from real-time information acquisition, we also need real-time information processing. Real-time information processing includes also several other high performance set-ups e.g. in-memory databases, effective algorithms and even potential collaborative approaches for pre-filtering or pre-processing of information for a specific (business) objective and complex analysis of relevant (stream) events in-network and on-device.

Complex Event Processing (CEP) for processing information in conjunction with CEP functionalities on the edge devices are expected to empower us with new capabilities and an architecture should integrate such concepts. Since CEP relies on several steps such as event-pattern detection, event abstraction, modelling event hierarchies, detecting relationships (such as causality, membership or timing) between events, abstracting event-driven processes etc., their requirements and design considerations must also be integrated.

3.2.13 Real-world Business Processes

With the standardisation and easier integration of monitoring and control capabilities in higher layers, the new generation of business processes can rely on timely acquired data exchange with the shop floor. This has as a result the potential to enhance and further integrate real world and its representation in business systems in a real-time manner.

It is expected that the business modellers will be able to design processes that interact with the real world possibly in a service oriented way [13, 17], and based on the information acquired they can take business relevant decisions and execute them. We consider strong integration with enterprise services among other things, as well as the tuning of a large-scale system of systems infrastructure to the business objectives [13].

3.2.14 Scalability

Scalability is a key feature for large-scale systems [16]. There are two kinds of scalability:

- *Vertical Scalability (scale up)*: To scale vertically (or scale up) means to add resources to a single node in a system, e.g. add CPUs or memory to a single computer. Such vertical scaling provides more resources for sharing.
- *Horizontal Scalability (scale out)*: To scale horizontally (or scale out) means to add more nodes to a system, such as adding a new computer to a distributed software application. An example might be scaling out from one web server system to three.

For industrial systems it is expected that scaling up of resources available on single devices will emerge anyway. As such the impact should be considered e.g. at SCADA/DCS/PLC etc. in order to assess what capabilities can be assumed by large scale applications e.g. monitoring. Scaling out is also a significant option to follow, especially relevant to nodes having attached a large number of devices e.g. a SCADA system or even a monitoring application running in the Cloud with thousands of metering points monitored.

The IMC-AESOP architectural approach following the SOA paradigm on all levels, must support very large heterogeneous networks and their capabilities e.g. ranging from gigabit networks to low-bandwidth, energy-constrained networked sensors and actuators connected over unreliable wireless links. This also implies that the overall network must be able support cross-network interaction with devices that are completely different in terms of processing power, bandwidth and energy availability. An one-size-fits-all approach is therefore not applicable; instead, the proposed architecture must incorporate mechanisms that can manage different types of devices, systems and networks. Recourse availability, Quality of Service (QoS), and load balancing are just a few examples of what the system architecture must be able to monitor and manage.

3.2.15 Service Lifecycle Management

The service lifecycle begins at inception (definition) and ends at its retirement (decommissioning or re-purposing). The service lifecycle enables service governance across its three stages: requirements and analysis, design and development, and IT operations. As this is going to be a highly complex system of systems, tackling the lifecycle management especially of composite (potentially cross-domain) services is challenging. To what extend support needs to rely on the core parts of the architecture and what can be realized as optional extensible add-ons that are domain-specific is a challenge. There are several technologies which already include the key concepts of service lifecycle management e.g. the Open Services Gateway initiative framework (OSGi) and these should be integrated in order to enable parallel evolution of the various architecture parts.

3.2.16 System Simulation

Simulations of process systems are pursued in different levels of detail and with different purposes. Three main levels of process simulation to be considered are:

1. *Process design and process control*: At this level the essential operational modes are studied and also the transition between these modes. Main transients and disturbances. Batches and main sequences are analysed. The target is to develop and verify the process design and its control philosophy.
2. *Implementation*: At this level the main focus is on the interface between the field instrumentation and the control system (DCS). There may be less emphasis on the actual process models but more on the signals. The target is to verify the DCS program in terms of logics, interlocks etc.
3. *Operations*: At this level the ability to operate efficiently is analysed. These simulators can be run in real-time and used as training simulators for the plant operator. The process, the automation system as well as the human interface are represented in the simulator. For aspects related to interoperability and the system view, simplifications e.g. in the process models and the automation systems may be assumed.

This break-down is quite rough and may significantly overlap; for instance a simulator for process design and process control design (1) can be further developed into a training simulator (3) where the actual DCS software is executed (2).

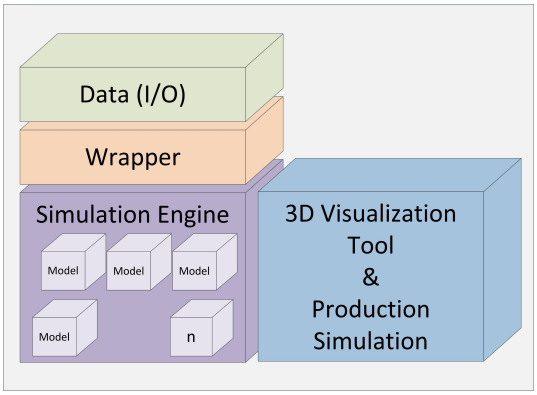

Fig. 3.3 Example of simulation core

One promising architectural approach includes using actual simulation tools and complementing them with an interface/front-end that allows us to simulate actual process and manufacturing systems via an SOA. For example, as shown in Fig. 3.3, having a simulation engine with a message wrapper that can encapsulate simulated events as e.g. SOAP messages may allow us to simulate an event based Large Scale System. Different simulation models can be placed inside the simulation engine,

each having certain pre-programmed behaviour that can help represent actual devices. It is also possible to compliment this architecture with 3D visualisation and production simulation to have a virtually complete system of systems. This kind of architecture approach could allow simulations on levels 2 and 3 mentioned previously. Since the same system could be coupled with SCADA or other supervisions systems, user operation/training simulations can be performed in parallel with implementation tests.

Industrial process plants can be considered as complex systems, where a change in one sub-process may result to unexpected consequences in other parts of the plant. Nevertheless, autonomicity of the sub-processes and the sub-systems is needed in order to achieve overall evolution. Therefore a holistic system analysis is needed in order to identify possible conflicts and side effects at an early stage. Simulations of process systems is pursued at different levels with varying detail. It is expected that system-wide simulations will assist in designing, building and operating future industrial infrastructure and their interactions.

3.2.17 Unique Asset Identification

Some kind of standardized universal asset identification and addressing mechanism is required for the architecture to be able to support service-oriented targeted communication between and inside the systems. This addressing mechanism should be flexible, scalable and it should not introduce additional overhead in configuration, performance and complexity.

In the era of Internet of Things, it must be possible to uniquely identify items and their services. Promising approaches include UUID (e.g. combination of unique data such as IP/MAC/serial number etc.). With IPv6 it might be possible to have these devices directly addressable. Assets are treated as a general case of devices, systems, people and other resources. These assets carry a unique ID with them, as this tag cannot carry too much information, software is used to link additional information to that asset. Then, other relevant wireless technologies (e.g. ZigBee), may be used to obtain the information from the ID.

Unique asset identification is very closely linked to the monitoring of assets, as it enables part of that monitoring. It can be from the location aspect or from the simple awareness of the qualities and properties of the asset that has to be identified. Additionally, the unique device authentication is expected to improve safety (in part by helping to identify counterfeit products and by improving the ability of staff to distinguish between devices that are similar in appearance but serve different functions). It would be useful if this is coupled with dynamic discovery. In a typical scenario a new device is plugged in the network by plant maintenance staff, and it is dynamically discovered and registered. Subsequently an operator is presented in his (mobile) tablet the new device and assigns the necessary configuration to it (this may be done remotely). Hence it is important to be able to distinguish and target specific devices even remotely.

3.3 A Service-based Architecture

The IMC-AESOP project follows a Service-Oriented Architecture, a general overview of which is depicted at high level in FMC notation (www.fmc-modeling.org) in Fig. 3.4. On the left side we see the users who interact with the services (depicted in the middle). The data depicted on the far right side can be accessed with the necessary credentials. Although we consider that the majority of these services will run on the "Cloud" some of these may be distributed and run in more lightweight versions on devices or other systems. As long as the SOA-based interaction is in place, they are considered as part of the general architecture view.

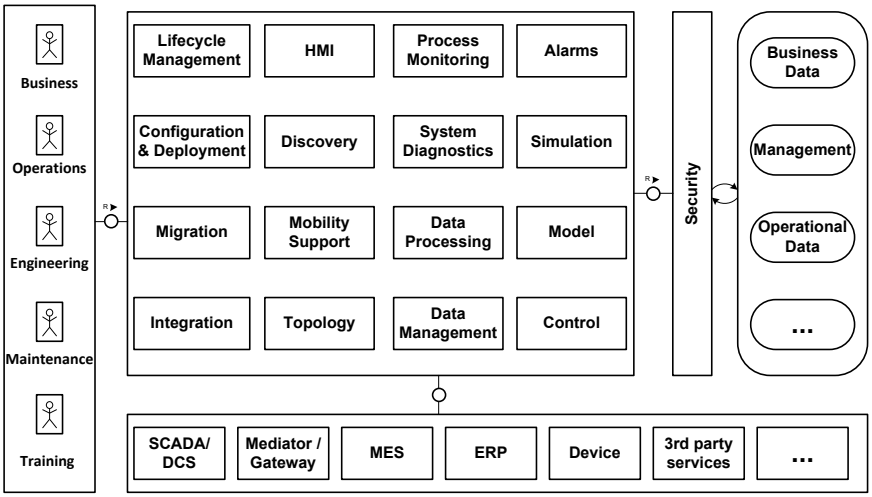

Fig. 3.4 Architecture overview

3.3.1 User Roles

Several *"user roles"* are envisioned to interact with the architecture either directly or indirectly as part of their participation in a process plant. The roles define actions performed by staff and management, and simplifies grouping of tasks into categories.

The *Business* role handles overall plant management and administration and ensures long-term effectiveness and strategic planning. From an IT point of view, this role is operating in the enterprise layer of a process plant, interacting with supporting systems such as Enterprise Resource Planning (ERP), Enterprise Asset Management (EAM), Operational Risk Management (ORM) etc.

The *Operations* role performs the normal daily operation of the plant, hence it handles optimisation of the monitor and control processes. It is also responsible for meeting the production targets while ensuring that the plant is running in the most efficient, safe, and reliable modes. The tasks performed as part of this role are located at the operations layer and use supporting systems such as Operations Control System (OCS) for monitoring and control of the process infrastructure and Process Optimisation Systems.

The *Engineering* role is here divided into two categories: Process engineering and System engineering. The Process Engineer ensures proper design, review, control, implementation, and documentation of the plant processes. It also designs the layout of the process and performs optimisation work with Operations. The System Engineer deals with the deployment of new automation devices, software components and machines, manages configurations, infrastructure and networks.

The *Maintenance* role is responsible for the system operation with optimum performance, and ensures that the the plant's systems and equipment are in a safe, reliable, and fully functional state. The maintenance operations are also part of the operations IT layer of the process plant. The systems that are supporting the tasks performed within the maintenance role are Risk Based Inspections (RBI) systems, Systems Monitoring, Diagnostics and Control etc.

Training ensures that all plant personnel have a basic understanding of their responsibilities as well as safe work practices. Training is performed on a regular basis by all other roles in order to improve work skills. The training planning for each employee must be harmonized with the management strategy planning and can be performed on-site but also using Simulation Training Systems.

3.3.2 Service Group Overview

As depicted in Fig. 3.4, it is possible to distinguish several service groups, namely: Alarms, Configuration and Deployment, Control, Data Management, Data Processing, Discovery, HMI, Integration, Lifecycle Management, Migration, Mobility Support, Model, Process Monitoring, Security, Simulation, System Diagnostic, Topology. These groups indicate high-level constellations of more fine-grained services. IMC-AESOP has defined some initial services which are listed in more detail in Table 3.1.

All of the services are considered essential for next generation Cloud-based collaborative automation systems. Table 3.1 depicts a first prioritisation according to what we consider necessary for future systems. The groups of services have been rated with high priority (+) if they constitute a critical service absolutely mandatory, with medium priority (o) if this is not a critical but nevertheless highly needed service and lastly with low priority (-), which mainly means "nice to have" services that enhance functionalities but are optional.

While most of these correspond to specific real-world scenarios we consider, expanding the potential scenarios may lead to adjustments on the architecture as

Table 3.1 Detailed Architecture Services and Prioritisation

Service Group	Services	Priority
Alarms	Alarm Configuration	+
	Alarm and Event Processing	+
Configuration and Deployment	Configuration Repository	+
	System Configuration Service	+
	Configuration Service	+
Control	Control Execution Engine	+
Data Management	Sensory data acquisition	+
	Actuator output	+
	Data Consistency	o
	Event Broker	+
	Historian	o
Data Processing	Filtering	+
	Calculation Engine	o
	Complex Event Processing Service	+
Discovery	Discovery Service	+
	Service Registry	+
HMI	Graphics presentation	+
Integration	Business Process Management & Execution Service	o
	Composition Service	+
	Gateway	+
	Service Mediator	+
	Model Mapping Service	+
	Service Registry	+
Lifecycle Management	Code Repository	-
	Lifecycle Management	+
Migration	Infrastructure Migration Solver	-
	Migration Execution Service	-
Mobility Support	Mobile Service Management	o
Model	Model Repository Service	o
	Model Management Service	o
Process Monitoring	Monitoring	+
Security	Security Policy Management	+
	Security Management	+
Simulation	Constraint Evaluation	o
	Simulation Execution	o
	Simulation Scenario Manager	o
	Process Simulation Service	o
System Diagnostic	Asset Monitor	+
	Asset Diagnostics Management	+
Topology	Naming Service	+
	Location Service	+

such. Within the IMC-AESOP project, several of these have been implemented as proof of concept. There are also several functional requirements which will need to be further evaluated and may depend on domain-specific scenarios. To what extend they might impact the proposed approach is avenue for further research.

3.3.3 Alarms

The Alarms service group (depicted in Fig. 3.5), contains services for alarm processing and configuration. These services support simple events and complex events that are aggregated from several events. Some of the alarms are generated in lower level services and devices but alarms can be generated also in the alarm processing service using process values and limits. The alarm configuration and processing services also support very flexible hierarchical alarm area definitions.

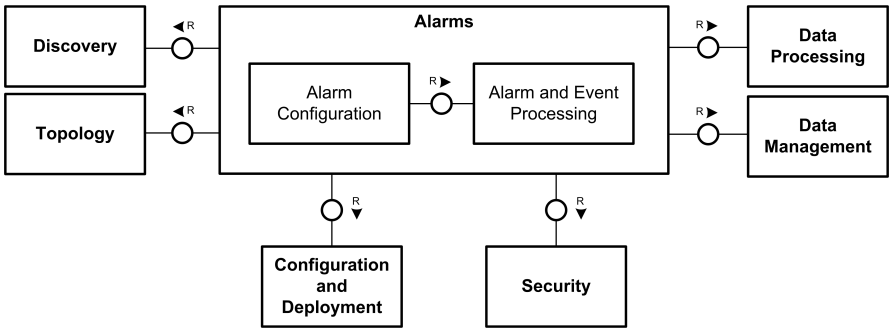

Fig. 3.5 Service Group: Alarms Overview

The alarm configuration service provides help for alarm definitions and maintenance of simple alarms and complex alarms (and events). Each alarm or event can be defined for one or many alarm areas. The alarm areas are hierarchical and there can be several parallel alarm hierarchies. One alarm can belong to one or many alarm hierarchies but it is included only once in one alarm hierarchy.

Complex events are events which are aggregated from several events. Those can also use other complex events but the hierarchy is not limited to levels, i.e. one complex event can use complex events (or events) from any level. Complex events are independent of the area definitions but each complex event can belong to one or many alarm area hierarchies. This service is limited to predefined complex events, so modelling of event hierarchies or detecting relationships is not part of it.

The processing service is able to handle thousands of events and map them to alarms coming from different devices in order to filter and aggregate the alarms. The service is based on Complex Event Processing (CEP) principles but it also supports simple (traditional) alarms. It is receiving the alarm area configuration, simple events configuration and complex events configuration and using it to process the incoming alarms and events. The service is activated every time when a new alarm or event arrives but it can be activated also when the complex event configuration contains time based activations. The configuration can be hierarchical and complex events can trigger higher level complex events. The complex event processing is typically triggered by an event which was created by another service or application

but it can also create its own events e.g. when configured to monitor some values against the limits.

Some of the typical alarm area hierarchies are process areas, instrument areas, safety areas, energy areas and quality areas. The plant personnel scope of responsibilities is linked to these area hierarchies.

3.3.4 Configuration and Deployment

The Configuration and Deployment service group (depicted in Fig. 3.6), is responsible for managing configuration and deployment of various systems from processes to devices. The service group consists of configurable services which enforce and execute the configuration on the device level, system configuration services which deploy configurations of the processes, and a configuration repository where various configuration modes are persisted and retrieved.

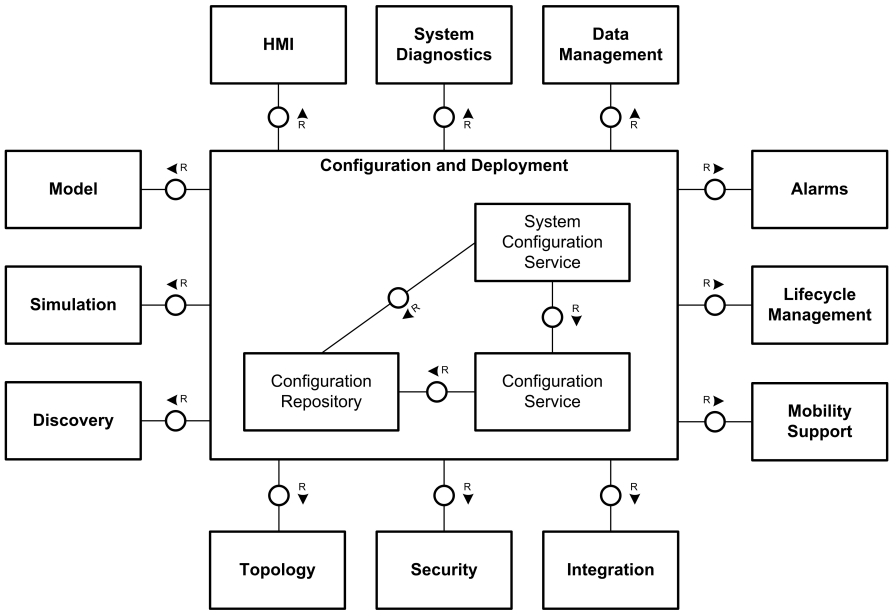

Fig. 3.6 Service Group: Configuration and Deployment Overview

The Configuration service is needed for the plant control strategy configuration. It is using directly the Model service, which supports all the functionalities needed for hierarchical control strategy configuration. In an example scenario where an engineer wants to add a control loop, he would have to add a node e.g. by sending a POST to https://imc-aesop.eu/configuration and pass all necessary parameters e.g. node info, attributes, parameters, control algorithms etc.

The Configuration Repository service is utilizing the Model Repository. It typically has process models for simulation purposes. However, the model repository is not limited to any specific type of hierarchical models e.g. the Configuration Repository service is utilizing the models structure to save the hierarchical configuration structure. Support for several parallel hierarchical models should be there e.g. the possibility to add nodes to each hierarchy separately and it is also possible to merge two hierarchies together. Each node contains some kind of process model or information about the process but the Model Repository service does not understand or care about the internal structure of each node.

The System Configuration Service provides functionalities to manage configurations for different systems such as processes, SCADA/DCS, PLC, and devices. This service is able to check configuration consistency, to send or re-send configuration files to devices, to manage versioned platform specific implementation of services, and to instantiate plant metamodels.

3.3.5 Control

The Control service group (depicted in Fig. 3.7), contains the control execution engine service, which is able to execute the process automation configuration or process models. The execution engine services are distributed to several physical nodes and some of those can be redundant. It also supports the typical on-line (and on-the-fly) changes in configuration while the process is running.

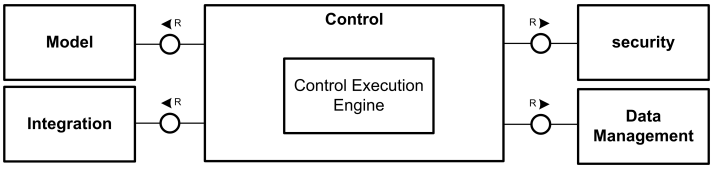

Fig. 3.7 Service Group: Control Overview

The Control Execution Engine service contains the execution engine that is capable of executing the code generated by the Configuration service or the Model Management service. The executable nodes created by the Configuration service typically contain functionalities to control the actual process while the nodes coming from the Model Management service contain process models. The Control Execution Engine service does not distinguish between these two node types. The requirement for each node is that it must contain the Execute method and following predefined attributes, e.g. CycleTimeMS, Phase, Priority and ExecutionOrder.

The Control Execution Engine is a distributed service that can run on tens of nodes simultaneously. Some of the nodes are real-time nodes where the deterministic execution is guaranteed. Two or more Control Execution Engines can be combined as a single redundant execution engine. In this case all the redundant execution

engines contain exactly same (configuration or model) nodes but only one engine (at the time) is responsible for the execution and data is copied to the passive engine(s). This responsibility is transferred to another redundant execution engine in case of a hardware failure.

The executable nodes are transferred from the tools when the engineer selects to load the node to the specified execution engine. The execution engine allocates the required memory for the node and adds it to the execution list with the specified cycle time, phase, priority and execution order. The engineer is then able to start the node execution and the execution engine will call the Execute method in specified cycle or when an execution event is received. It is possible to replace the node with new version on-line by manually stopping the execution, loading the new version and restarting the execution or "on-the-fly" by replacing the old version between two execution cycles without losing any control cycles.

3.3.6 Data Management

The Data Management service group (depicted in Fig. 3.8), encapsulates the functionality of data retrieval, consistency checking, storage and basic eventing. Data management provides services for acquiring data from sensors, consistency checking and plausibility checks, data logging & searching, event generation, and actuator control.

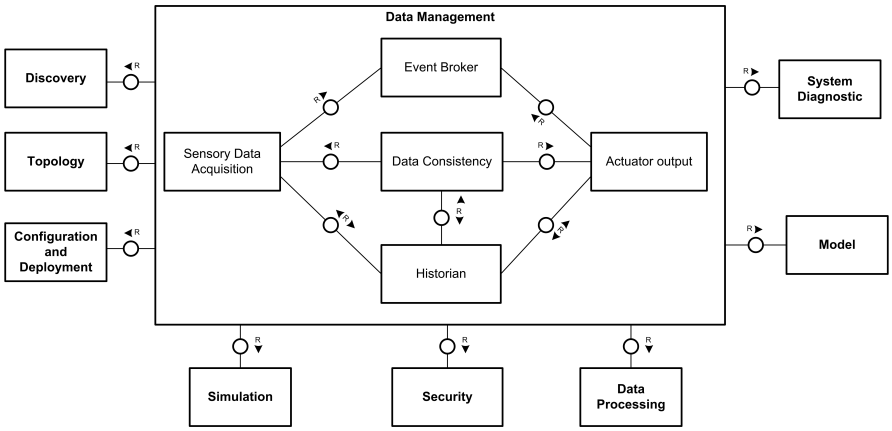

Fig. 3.8 Service Group: Data Management Overview

The sensoryDataAcquisition service provides an interface for retrieving sensor data. It connects physical devices producing data with higher layer services, such as filtering, eventing and processing, in the architecture. The main function provided by this service is reading of sensor data. It provides methods for typing of the data

and mapping to a data model/ontology. Configuration and other features are handled by other services.

The actuatorOutput service is used to control the output of the actuator devices. Typed data, performed by the Consistency check service is used to control the physical output of a device's actuator(s). The main function provided by this service is setting and reading of actuator outputs. It provides methods for typing of the data and mapping to a data model/ontology.

The dataConsistency service validates that data delivered from a device are consistent according to specific rules. The validation can be performed on a device/resource, or within the Cloud. Cloud-based validation enables complex queries involving multiple sources of data to be executed. The Data consistency service also provides methods for filtering and detection of data that have anomalies. This service allows configuration and querying of consistency rules. Moreover, it provides a way to retrieve the inconsistent data for debugging purposes.

Firing and receiving events constitutes a core concept within the ICM-AESOP architecture, and hence the Event Broker plays a pivotal role. In general, each service can act either as a producer of events and/or as a consumer of events. As a producer a service would have to enable consumers to subscribe to topics of events to enable the producer to push events to a given endpoint. A consumer has to provide an endpoint to which a producer can push the events. The Event Broker is a service that can fire and receive events. The service can subscribe to various topics of event provider and offers an interface for consumers to subscribe to topics of events. In addition, this service uses the Historian service to log events for reliability purposes. The Event Broker service can be used in situations when n producers and m consumers need to be connected. Thus, instead of having $n * m$ registrations the Event Service reduces the numbers of registrations to $n + m$. Nevertheless, an Event Service can become also a bottleneck. Therefore, the architecture does not limit the number of Event Services in a system.

The role of the historian service is to keep and manage a record of a time series of data or events. Historical data can include sensor values, device states, calculated or aggregated values, and diagnostic data. Events of interest can include alarms, state changes, operator instructions, system triggers, or any other notification. The data historian exposes an interface for storing, configuring, browsing, updating, deleting, and querying historical data and historical events. A typical scenario would be the logging performance data for system diagnostics. The system diagnostics tools benefit from having a view of the state of certain system parameters over time when diagnosing the source of some fault. Similarly, histories for relevant alarms and events can be kept. Similarly in Process Optimisation, when trying to optimize a process based on some criteria, historical process data can be used to identify where adjustments can be made.

3.3.7 Data Processing

The Data Processing service group (depicted in Fig. 3.9), provides services from simple filtering up to complex analytics. This is meant in a functional grouping and is intended to be used on all levels, from device up to the Cloud.

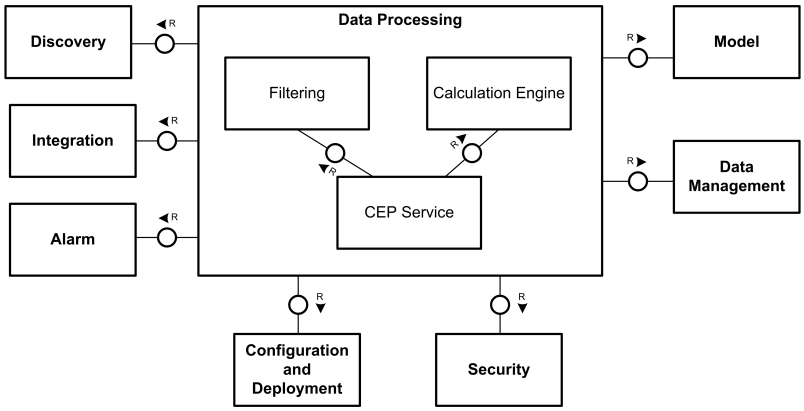

Fig. 3.9 Service Group: Data Processing Overview

Complex Event Processing is a technology for low-latency filtering, correlating, aggregating, and computing on real- world event data. A service offering CEP capabilities enables on one side the consumption of events as inputs and produces (complex) events on the output side. In addition the service enables the deployment and management of rules (or queries) over the incoming events. These rules (or queries) are producing the events on the output. Thus, the service offers also a management API to create, update, or delete these rules. The CEP Service provides functionality of a complex event processing engine as a service.

The purpose of this Calculation Engine service is to provide environment for user-defined calculations including numeric and logic operations. The user-defined calculations are additional, perhaps temporary, calculations which are used e.g. for reporting purposes, process studies, etc. More permanent calculations should be done using normal DCS configurations tools. The user-defined calculations can use, combine and manipulate any process values available in the IMC-AESOP system address space.

3.3.8 Discovery

The Discovery service group (depicted in Fig. 3.10), mainly includes services targeting dynamic discovery that allows to find devices/systems/services by type; and

location and a registry type service, relying on a known registry end-point address, featuring at least register, de-register, search and rating of services operations.

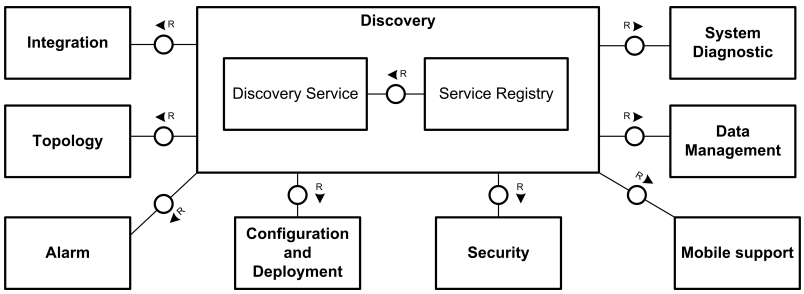

Fig. 3.10 Service Group: Discovery Overview

Any service, either provided by a physical device within the plant premises or hosted in the Cloud, will announce and describe itself when entering the Cloud of services. Any other device or service may request more detailed information (service description) or search for available services in the Cloud of services. Experimentation in IMC-AESOP demonstrators did show that this discovery mechanism could be combined with a static service enumeration, ensuring that services required for the proper operation of the application have been discovered at runtime.

A typical scenario would be the automatic plug and play. As soon as a device is plugged into the Cloud of services, it can automatically search for services that it requires to provide its function and start when these services are available. In the same train of thought, as soon as a device is plugged into the Cloud of services, its provided services are automatically registered in order to provide any management functionality of the Cloud of services.

An automatic discovery mechanism, relying on multicasting or broadcasting as described above, is not compatible with all network architectures and all types of services. A service registry is more generally required for SOA-based architectures where services can be hosted both locally and remotely. It is also required for types of services that do not support discovery mechanisms, REST services for instance.

The registry service is used as a repository for all available services across IMC-AESOP architecture. This repository is accessed either (i) by systems and/or devices that register or de-register their services into the registry, mainly at initialisation time; (ii) by systems and/or devices looking for a specific service.

For example, in case the local network is segmented by routers (physical segmentation) or VLANs (virtual segmentation) then both multicast and broadcast communication will be limited to a local subnet and will not spawn multiple network segments. It is therefore useful to consider a discovery proxy mechanism that any endpoint in the system can access, either for registration or for query, independently from its physical location in the network. This discovery proxy service is by essence a service registry. When for instance a new device is connected to the local network and exposes a well-known maintenance service including various device

configuration and monitoring methods, it automatically registers its service(s) into the IMC-AESOP services registry. Any monitoring application, looking for maintenance services, can query the registry and retrieve the new device maintenance service endpoint.

3.3.9 HMI

The HMI service group (depicted in Fig. 3.11), contains the graphic presentation service which supports the graphical tools in generic web based user interface framework. It provides the generic menu and help functionalities and also the application area where the actual graphical tools are shown.

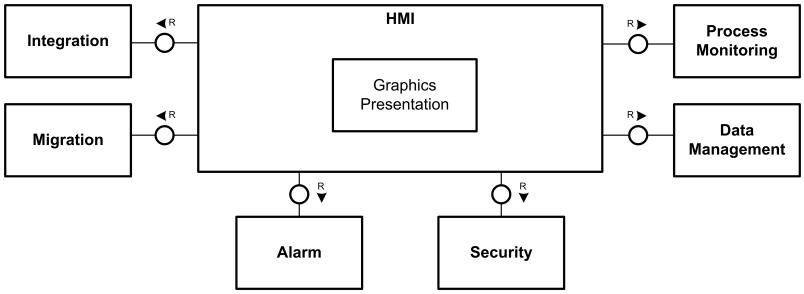

Fig. 3.11 Service Group: HMI Overview

The Graphics Presentation service aims at easing interaction with the multiple heterogeneous visualisation devices and applications we expect to populate future systems. We consider a very important and challenging task to design a new framework that will have basic services that will offer the capability to compose Graphical User Interfaces in a service-driven way. Here the guidelines and concepts from W3C should be followed for the sake of interoperability and openness.

A very simple (and probably only as an intermediate solution) would be to provide each graphical element as a result of a service that could be combined in the screen and be utilized by a specific technology. The amount of active content on the pages is minimized but in some areas it is required because of the performance requirements. However, the active content is transparent for the user and does not require any visible installation procedures or registrations. That is why it is possible to use any computer and browser (which can execute the active content) in the network.

3.3.10 Integration

The Integration service group (depicted in Fig. 3.12), enables the combination of functionality for added value. Heterogeneous components with different communication protocols and data models require services to facilitate their interoperable interaction. Business process management and execution, composition, functionality wrappers (gateway and mediator), and model mapping services are part of this service group.

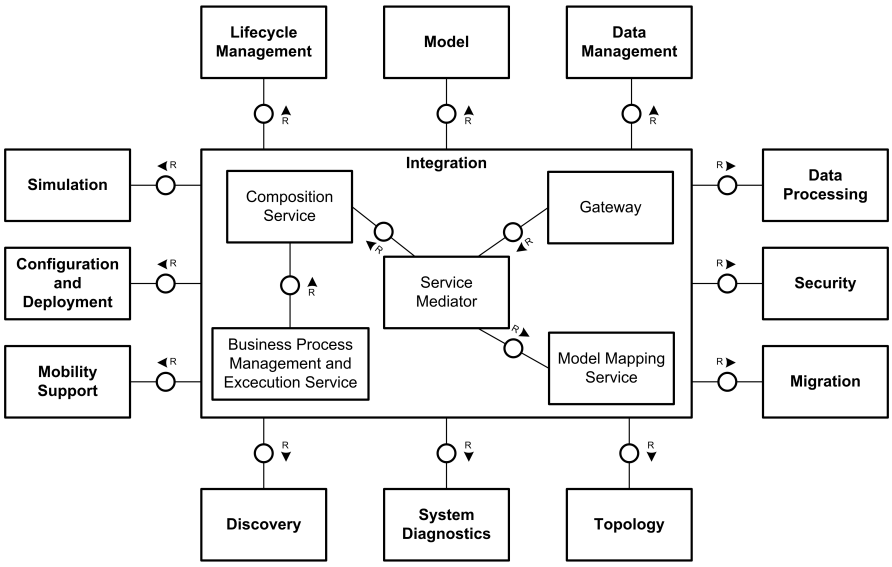

Fig. 3.12 Service Group: Integration Overview

The Business Process Management and Execution service manages and executes business processes. The platform exposes processes as higher-level services, possibly in the form of a WSDL document with semantic descriptions, and can provide additional tools for controlling and analysing the process.

A composition service provides a platform for managing execution of service compositions. This service would receive as input descriptions of service compositions, defined as combinations or sequences of finer-grained services along with descriptions of input and output message exchange patterns, logic describing process flow and error conditions. The platform would then expose the business process as an interface to a higher-level service, handling any input and output parameters specified. An example would be wrapping often-repeated service invocation patterns as a coarser-grained service. An engineer identifies a service invocation pattern that is often repeated in higher-level business processes, for example a multi-step start-up or shutdown sequence, or a complex heating or cooling cycle. He then describes

this pattern using supported notation, and exposes it as a coarse-grained service with more business relevance.

In the context of the integration of legacy systems and devices in the plant, Gateways and Mediators are used to expose the legacy data as high level services using the state-of-the-art meta-model. The Model Mapping Service encapsulates the conversion between the legacy and IMC-AESOP data models, including the semantic level. The Model Mapping Service is used both by Mediators and Gateways. The Legacy Models and Mapping Rules are typically initialized during the start-up of the system but they can be updated all along the plant life-cycle. The Mapping service is called either by the Gateways, generally to react to specific demands, or by mediators on a regular basis.

A Gateway service provides the means to encapsulate legacy protocol and application objects logic. Encapsulation is supported by the Model Mapping Service. It is used to introduce a non-standard service contract with high technology coupling. Based on the model mapping, this service represents semantics of legacy components as they expose within the legacy system.

A Service Mediator service provides the means to encapsulate legacy protocol and application objects logic of a single or several legacy components data and associated functions. Encapsulation is supported by the Model Mapping Service, the Business Process Management and Execution service and the Gateway service. It is used to introduce a non-standard service contract with high technology coupling. Based on the model mapping, this service uses data retrieved from legacy components to provide enhanced semantics, legacy components are not able to expose by themselves.

3.3.11 Lifecycle Management

The Lifecycle Management service group (depicted in Fig. 3.13), is crucial system dealing with the management and evolution of the infrastructure itself. The services provided cover system lifecycle aspects such as maintenance policies, versioning, service management, and also concepts around staging (e. g. test, validation, simulation, production).

Services will need to be maintained, (re)deployed, upgraded etc. over a longer period of time. Hence it makes sense to have code repositories that maintain the various implementations and potentially also the source code of these (if available). The code repository should cover needs such as:

- The need to find code based on criteria e.g. author, execution environment, platform, technology, description, performance, etc.
- The need to describe the developed code based on widely acceptable templates and vocabulary. Using these, the vision of semantic web is promoted and also the automation of tasks such as search, management etc. can be delegated to intelligent technologies e.g. intelligent mobile agents.
- The need to integrate security, trust and availability from day one.

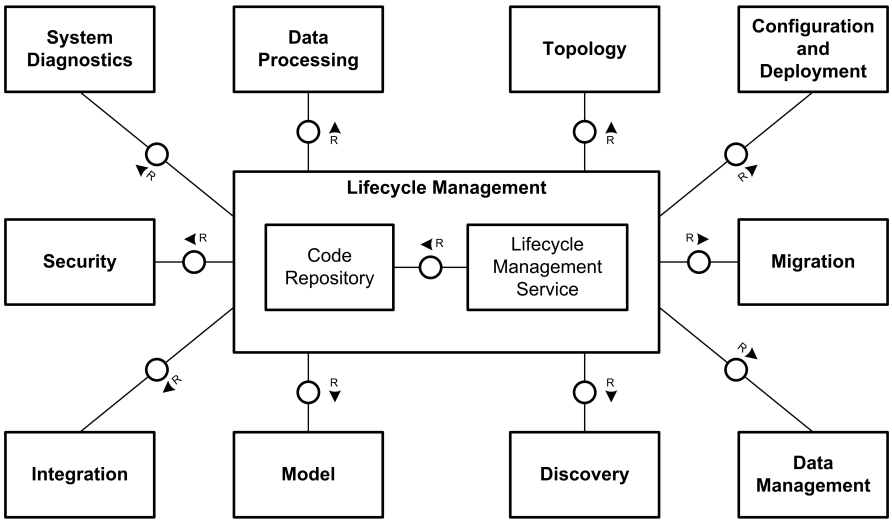

Fig. 3.13 Service Group: Lifecycle Management Overview

Lifecycle Management is to support the various services envisioned in IMC-AESOP from an infrastructure point of view. This indicates enabling support for key aspects including deployment, migration and discovery.

3.3.12 Migration

The Migration service group (depicted in Fig. 3.14), provides support to migrate a legacy system to a new SOA-based system. This group contains two main services, the Infrastructure Migration Solver and the Migration Execution Service. The infrastructure migration solver helps identify dependencies and offers migration strategies and instructions. The migration execution service implements migration process according to dependencies and instructions.

Under the provision of a set of constraints and a model this service evaluates the feasibility of solving a potential migration from the current landscape to the new one. This is a very complex process, where the details are to be captured on the model and constraints themselves. The Migration Execution Service executes the changes needed as identified by the Migration Solver Service. It is assumed that this may be a workflow and step-by-step process where hardware and software parts are migrated.

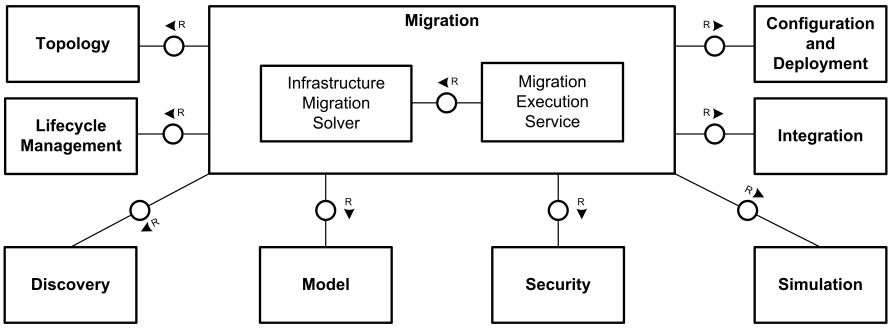

Fig. 3.14 Service Group: Migration Overview

3.3.13 Mobility Support

The Mobility Support service group (depicted in Fig. 3.15), provides services for managing mobile assets, such as mapping/changing IP addresses, asset locations, tracking, etc. It also provides data synchronisation services to enable up to date data access and sharing for mobile services and devices.

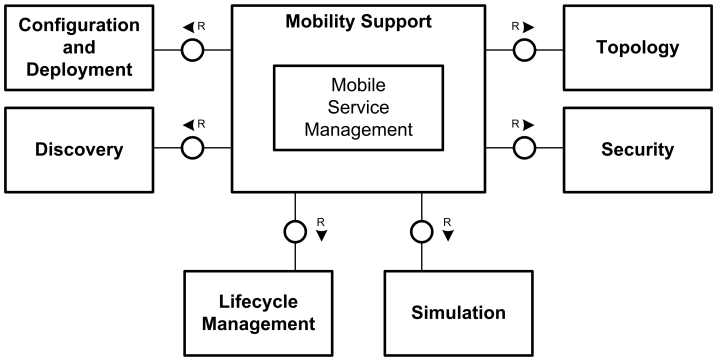

Fig. 3.15 Service Group: Mobility Support Overview

3.3.14 Model

The Model service group (depicted in Fig. 3.16), contains services for model management and repository. These services are very generic and those can be used for process automation configurations or process models, and are not limited to these hierarchical models. The model repository takes care of the structure but not the content, hence it is able to support several model types.

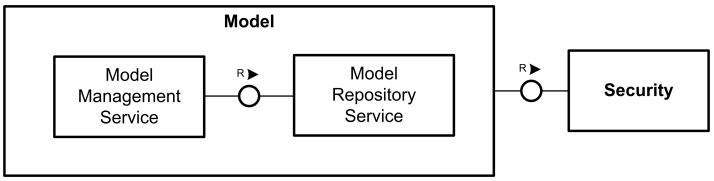

Fig. 3.16 Service Group: Model Overview

Plant Information Model is a model which contains e.g. hierarchical plant control strategy or a hierarchical plant process model. However, the model can be created for any purpose and it is not limited to these examples. Potentially plant maintenance/service, production optimisation or multivariable controls might require a different kind of hierarchical plant information models.

The hierarchy is maintained by Model Repository service which does not know or care about the actual structure of each node. The Model Management service links the nodes, parameters, attributes and methods together. After these definitions it is possible to execute the hierarchy (or part of it) on a distributed or centralized execution engine. The Model Management service contains also some predefined basic data types (e.g. float, double, int, unsigned int, byte, string etc.), some predefined enumeration types, as well as some attributes (e.g. CycleTimeMS, Phase, Priority, ExecutionOrder, etc.). However, with Model Management service it is also possible to add new data types, structures, enumerations and attributes.

The Model Repository service provides an interface to model repository. These models are typically process models for simulation purpose. However, the model repository is not limited to any specific type of hierarchical models. The service supports several parallel hierarchical models. It is possible to add nodes to each hierarchy separately and it is also possible to merge two hierarchies together. Each node contains some kind of process model or information about the process but the Model Repository service does not understand or care about the internal structure of each node.

3.3.15 Process Monitoring

The Process Monitoring service group (depicted in Fig. 3.17), serves as the entry point for the operator through the HMI. It is used to gather information relevant to the physical process e.g. adding semantics to the raw sensor data gathered from Data Processing and Data Management service groups. It also deals with process-related alarms and events.

This service provides an interface to collect and analyse process data using capabilities of other architectural components, compare data against expected or simulated results, attach process semantics to raw data, and calculate process-related KPIs. An example would be the operator monitoring the relevant parameters or mea-

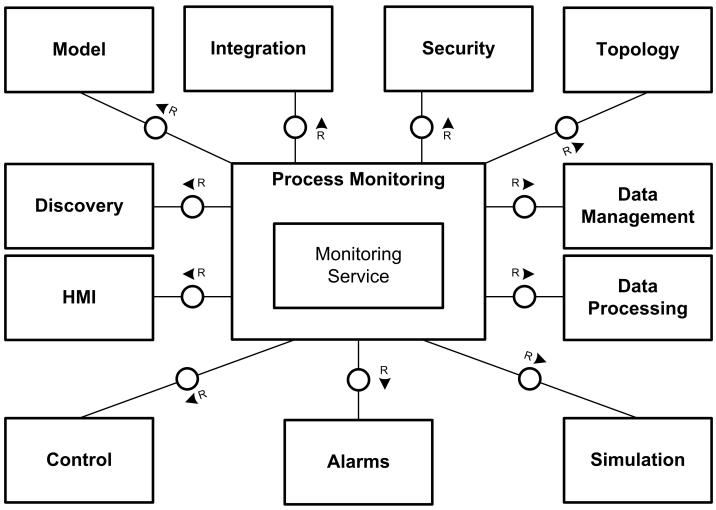

Fig. 3.17 Service Group: Process Monitoring Overview

surements related to the physical process, including levels, flow rates, temperatures, etc. These values, or calculations and aggregates of these values, can be displayed on an HMI

3.3.16 Security

The Security service group (depicted in Fig. 3.18), is of key importance especially when it comes down to enabling interactions among multiple stakeholders with various goals and access levels. The security management focuses on enforcement or execution of security measures and policy management is about definition and management of security rules or policies. The security services are implicitly used by all architecture services.

In IMC-AESOP, services play a central role which connects heterogeneous devices with monitoring and control applications and makes diverse service applications and business processes interoperable. Therefore, the security architecture of IMC-AESOP mainly focuses on the service related security components such as security management and policy management. Security management focuses on enforcement or execution of security measures and policy management is about definition and management of security rules or policies.

The security management service provides fundamental security functionalities such as authentication, authorisation, confidentiality, digital signatures, etc. The service is also able to provide deployment and enforcement support for security policies and rules defined by security administrators. The duties of the security policy management service are two-fold: (i) manage the policies which define access rights

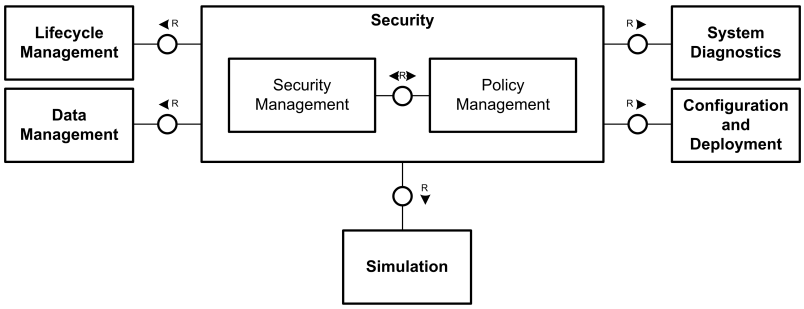

Fig. 3.18 Service Group: Security Overview

to devices or services depending on the user type (identity based or role-based), (ii) manage the policies which define identity federation to establish federation among various service domains.

3.3.17 Simulation

The Simulation service group (depicted in Fig. 3.19), is practically related to every other service group in the architecture as it aims at simulation of multiple systems and their processes. It is in charge of evaluating constraints and simulating execution. It also manages simulation scenarios and uses the exposed simulation endpoints provided by other services to emulate the performance and behaviour of a system (or a multitude of systems). It consists of four main services: Constraint Evaluation, Simulation Scenario Manager, Simulation Execution and Process Simulation Service.

The constraint evaluation service validates a given model with associated constraints and returns possible solutions of the constraint system if those solutions do exist. An example scenario would be the distribution of processes to given topology. The constraint evaluation is able to get as a model the topology with capabilities of the various nodes. In addition the service needs information about the constraints of process which should be distributed, such as worst-case execution time (WCET), network bandwidth etc. Based on this information the constraint evaluation can provide a possible distribution of processes on the nodes fulfilling the given constraints.

Process simulation service is in charge of simulating functional and non-functional behaviours of specific processes and validating the feasibility and performance of the simulated processes. This can be used also for operator training. This service interacts with the simulation scenario manager for managing scenario specific processes to simulate, with the constraint evaluation service to validate processes and with simulation execution service to deploy and execute simulated processes. An example would be where the process simulation tool needs to manage simulated processes. The engineer uses the process simulation tool to create, update or load

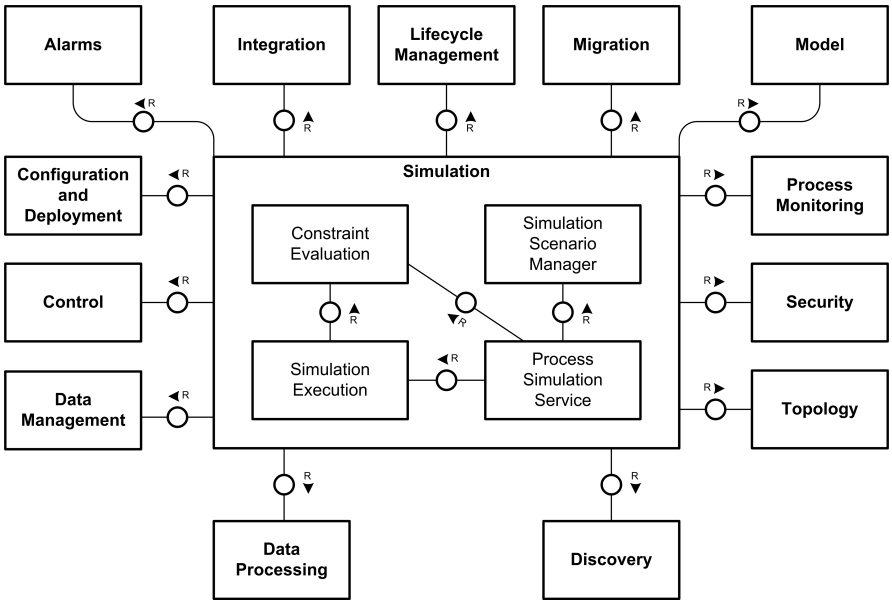

Fig. 3.19 Service Group: Simulation Overview

simulated processes and to validate the processes before simulation execution. The engineer also needs to manage simulated processes under specific simulation scenarios.

The Simulation Execution service is responsible for obtaining the required information to simulate a system(s), or a part of a system(s), and executing said simulation. Within the Simulation Service group it requests the simulation constraints from the Constraint Evaluation Service and the process (if any) that is to be simulated from the Process Simulation Service. At an external level, this service requires interaction with others as depicted also in Fig. 3.19. Each of the previously mentioned services provides the Simulation Execution service with any information, services, processes, workflows, models and logs it might need to execute a successful simulation.

The Simulation Scenario Manager is concerned with the configuration and management of different simulation scenarios. It depends, internally, on the Process Simulation Service and the Constraint Evaluation Service. The Simulation Scenario Manager can be used to configure and create simulation scenarios. These scenarios can be obtained to a certain degree by evaluating the constraints of the different systems. But by setting theoretical circumstances it is possible to simulate systems under different situations.

3.3.18 System Diagnostic

The System Diagnostic service group (depicted in Fig. 3.20), provides features for diagnostics of services and devices. Diagnostics can be used to monitor the health and condition of devices (shop-floor devices, servers, network devices, SCADA systems and PLC, etc.), and status of services. This group is used primarily for maintenance and planning purposes.

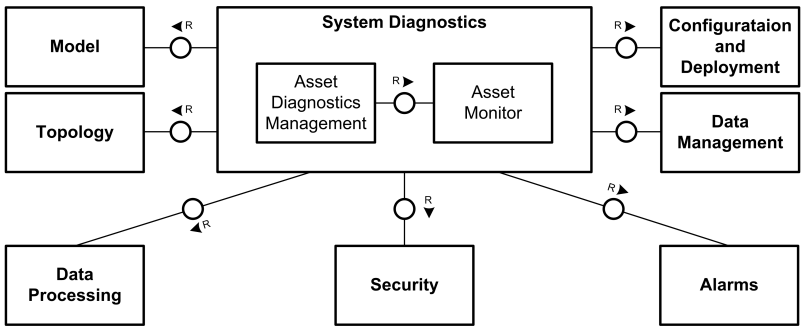

Fig. 3.20 Service Group: System Diagnostics Overview

The Asset Diagnostics Management service is used for controlling debugging, logging and testing capabilities. It can also be used to initiate self-test procedures on a resource. The capabilities of this service include turning on and off debugging and error logging, manual setting and examining different parameters and rebooting a device. The Asset Diagnostics Management service can be used for maintenance purposes in order to detect faults, initiate self-tests and configure logging of warnings and errors to detect malfunction.

The Asset Monitor service maintains the current state for each asset. It is also responsible for keeping log of maintenance interventions and planned maintenance schedule. It should be possible to configure the service with specific parameters and characteristics for each asset. These can include operational lifetime, depreciation rate, energy conservation modes, self-testing intervals and safety checks. Based on this information it is possible to perform complex asset management analysis such as Risk Based Inspections. A possible scenario for the Asset Monitor service is to provide the foundation for Asset Life Cycle Management infrastructure that is capable of optimizing the systems operational efficiency in terms of reduced maintenance costs and energy consumption.

3.3.19 Topology

The Topology service group (depicted in Fig. 3.21), allows describing and managing the physical and logical structure of the system. It includes Domain Name Service (DNS) functionality, location and context management, network management services etc.

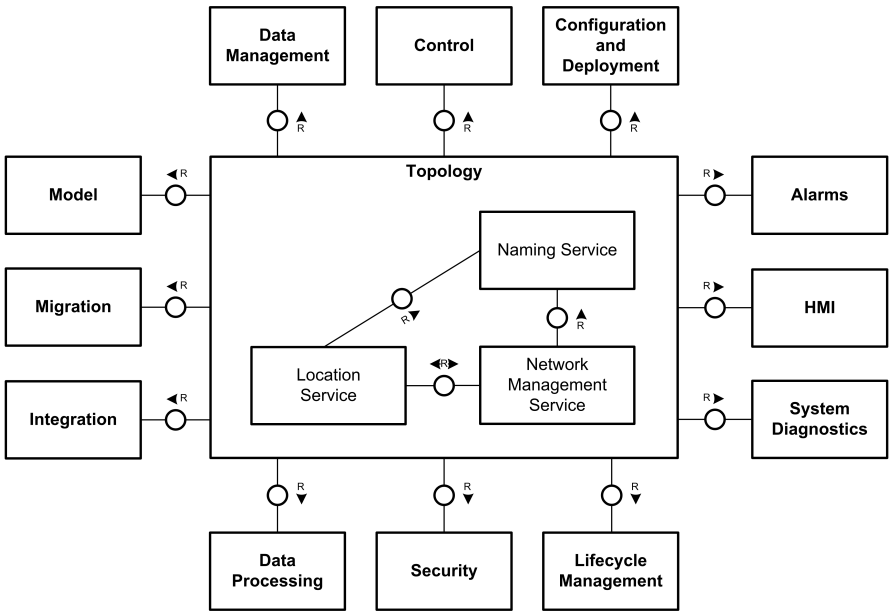

Fig. 3.21 Service Group: Topology Overview

This information is provided to any interested service, which may be application specific services, network management service, naming service (if the application has chosen to use it to build the device/service name), discovery services and more. An example scenario would be where the integrator needs to know the device location in a building to setup the application. In order to setup the building control application, the integrator needs to associate sensors and actuators located in the same room in order to provide automatic control and monitoring of the room. This association can only be based on this location information.

The various assets in the system must be able to interact together without knowing the details of the network addressing. For this purpose, an asset must be able to refer to another one by a name rather than by information related to its network address. This service supports the creation and the update of these names as well as their usage at runtime. The Naming service supports dynamic scenarios where new assets appearing in the system can be automatically discovered and used by other assets. A basic responsibility for the network management service would be

to monitor the health of IMC-AESOP network. This service would regularly (or asynchronously upon user request) scan known endpoints across the network topology to assess their connectivity status. Connectivity here may be defined according to several requirements, including network bandwidth/response time and a link/no-link status.

The Network Management Service (NMS) manages the network elements, also called managed devices. Device management includes faults, accounting, configuration, performance, and security (FCAPS) management. Management tasks include discovering network inventory, monitoring device health and status, providing alerts to conditions that impact system performance, and identification of problems, their source(s) and possible solutions.

The Network Management Service also allows configuring real-time network channels, thus ensuring a proper quality of service for IMC-AESOP real-time services. This service allows configuring the quality of service at router and switch levels through DiffServ and priority queues. IMC-AESOP real-time services, will typically require the existence of such real-time channel configuration and management from the underlying networking infrastructure.

The network management service depends on the location service to be able to walk through the entire network topology and on the naming service for simple endpoint identification, independently from any network-addressing scheme. An example scenario would be that of a device failure. A maintenance technician gets an alarm event on his SCADA application. This event has been sent by the network management service upon detection of a loss of connectivity with a given device. The maintenance technician runs a complete scan of the network to get a detailed health check of the communication layer. This health check should give him enough information to assess the severity of the issue.

3.4 The Next Generation SCADA/DCS

Service-Oriented Architectures are considered a promising way towards realizing the factory of the future, and we have shown that these can be used to empower infrastructures and their components. The IMC-AESOP architecture and its services already described, offer the possibility of realizing the next generation of Cyber-Physical Systems that heavily depend on the cyber part such as the cloud based services. Such an example CPS is the SCADA/DCS systems used today in all industries.

Industrial processes as well as many other critical infrastructures depend on SCADA and DCS systems in order to perform their complex functionalities. The multitude of functionalities that they need to support as well as the exact roadmap is heavily still researched in an environment where disruptive technologies and concepts are developed rapidly [10]. Having in place an architecture as depicted in Fig. 3.4 has profound implications on the design and deployment of future solutions in the industrial automation domain.

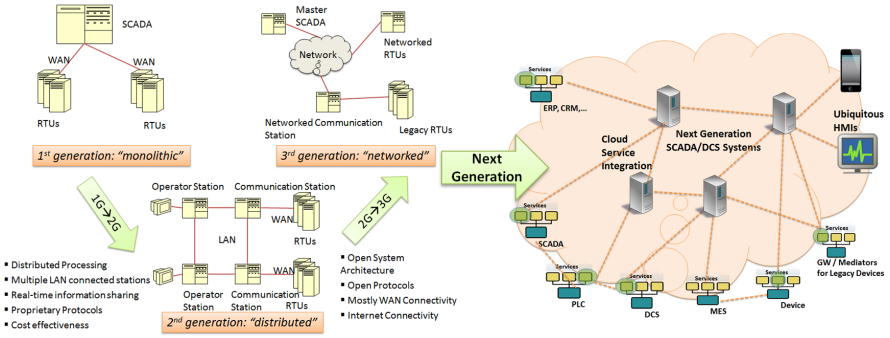

Fig. 3.22 Next Generation SCADA/DCS as a composition in a "Service Cloud" [10]

Cyber-Physical Systems have already undergone significant evolutionary steps the last decades (shown in Fig. 3.22) and are moving towards an infrastructure that increasingly depends on monitoring of the real world, timely evaluation of data acquired and timely applicability of management (control) [10]. The latter is becoming even more difficult to design and manage when massive numbers of networked embedded devices and systems are interacting. As such new approaches are needed that go beyond the classical monitoring and are able to deal with massive data and complex reasoning depending on the affected processes as well as enterprise-wide constraints. Such "capabilities" would by nature require multi-stakeholder involvement and data access that has to go beyond the classical monolithic one-domain and task-specific development approaches.

Currently implemented SCADA/DCS systems architectures [2] were designed for more closed and hierarchically controlled industrial environments, however it is expected that there is potential to enhance their functionality and minimize integration costs by integrating themselves into collaborative approaches with enterprise systems and large-scale real-world services [10]. In this sense, there is a need to consider what the next steps could be towards engineering/designing the next generation of SCADA/DCS systems of systems that could successfully tackle the emerging challenges such as degree of centralisation, optional independence of each of the participating systems, and independent evolution of them. We consider that the cloud-based evolution of SCADA/DCS is the next step to follow.

For some domains e.g. in industrial automation, timely access to monitoring and control functions is of high importance, depending on the requirements the application poses. For instance the "Cloud of Things" [11] may be used to empower the next generation of SCADA/DCS systems in conjunction with several services that may be hosted on the devices, in gateways and systems, in the Cloud as well as cross-layer compositions and interactions among them. For many of these, reliability and high performance interactions are needed, which poses the problem of finding the equilibrium of computation, communication, resource optimisation, openness and user-friendliness in the interactions between the different systems, devices, etc. However for the future we assume that each device or system (generally

each “thing”), can be empowered with Web services either directly (the device is powerful enough to host them locally) or indirectly (the services are provided by a gateway or any other device they are attached to). These services can be accessed directly by applications, systems and other services independent of where they reside empowering an larger collaborative ecosystem of Cyber-Physical Systems such as that envisioned by the IMC-AESOP.

The proposed IMC-AESOP architecture (depicted in Fig. 3.4) could have a significant impact on the way future industrial systems interact and applications are developed. By realizing it, a flat information-based infrastructure (as depicted in Fig. 3.1) that coexists with status quo is created. This means that the next generation SCADA and DCS systems could heavily depend on a set of common services and strike the right balance between functionality co-located on the shop-floor and delegated into the Cloud [10]. The aim is to have an approach that is more fit for the era where the Internet of Things, infrastructure virtualisation and real-time high performance solutions are sought. Hence, the next generation SCADA/DCS systems [10] does not necessarily have to possess a physical nature; this implies that it might reside overwhelmingly on the “cyber” or “virtual” world. As such it may comprise of multiple real world devices, on-device and in-network services and service-based collaboration driven interactions mapped into a “Service Cloud” (as depicted in Fig. 3.22).

A typical example would be that of asset monitoring with future SCADA systems. In large scale systems it will be impossible to still do the information acquisition with the traditional methods of pulling the devices, complemented with an event driven infrastructure. Additionally sophisticated services would perform analytics on the acquired data, and Decision Support Systems would use their results in real-time to take business relevant decisions. Decision taken will then be enforced enterprise-wide. Such systems will blend from the information flow viewpoint the layers among the different systems and realize the envisioned flat information-driven infrastructure that can be used for mash-up applications and services as shown in Fig. 3.1).

3.5 Conclusion

Future industrial applications will need to be developed at a rapid pace in order to capture the agility required by modern businesses. Typical industrial software development approaches will need to be adjusted to the new paradigm of distributed complex system software development with main emphasis on the collaboration and multi-layer interactions among systems of systems which is challenging [9]. To do so, some generic common functionality will need to be provided, potentially by a distributed service platform hosting common functionalities, following the Service-Oriented Architecture approach.

Such a collection of services forming a service-based architecture (shown in Fig. 3.4) is presented, prioritized and their potential impact is analysed. Signifi-

cant effort needs to be invested towards further investigating the interdependencies and needs of all targeted service domains as well as the technologies for realizing them. The proposed service architecture attempts to cover the basic needs for monitoring, management, data handling and integration etc. by taking into consideration the disruptive technologies [10] and concepts that could empower future industrial systems.

Acknowledgment

The authors would like to thank the European Commission for their support, and the partners of the EU FP7 project IMC-AESOP (www.imc-aesop.eu) for the fruitful discussions.

References

[1] acatech (2011) Cyber-Physical Systems: Driving force for innovation in mobility, health, energy and production. Tech. rep., acatech – National Academy of Science and Engineering, URL http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Publikationen/Stellungnahmen/acatech_POSITION_CPS_Englisch_WEB.pdf

[2] Barr D (2004) Supervisory control and data acquisition (SCADA) systems. Technical information bulletin 04-1, National Communications System (NCS), URL http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf

[3] Colombo AW, Karnouskos S (2009) Towards the factory of the future: A service-oriented cross-layer infrastructure. In: ICT Shaping the World: A Scientific View, vol 65-81, European Telecommunications Standards Institute (ETSI), John Wiley and Sons

[4] Colombo AW, Karnouskos S, Bangemann T (2013) A system of systems view on collaborative industrial automation. In: IEEE International Conference on Industrial Technology (ICIT 2013), Cape Town, South Africa

[5] Delsing J, Eliasson J, Kyusakov R, Colombo AW, Jammes F, Nessaether J, Karnouskos S, Diedrich C (2011) A migration approach towards a soa-based next generation process control and monitoring. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia.

[6] Drath R, Barth M (2011) Concept for interoperability between independent engineering tools of heterogeneous disciplines. In: Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on, pp 1 –8, DOI 10.1109/ETFA.2011.6058975

[7] Jamshidi M (ed) (2008) Systems of Systems Engineering: Principles and Applications. CRC Press

[8] Kagermann H, Wahlster W, Helbig J (2013) Recommendations for implementing the strategic initiative INDUSTRIE 4.0. Tech. rep., acatech – National Academy of Science and Engineering, URL http://www.acatech.de/fileadmin/user_upload/Baumstruktur_nach_Website/Acatech/root/de/Material_fuer_Sonderseiten/Industrie_4.0/Final_report__Industrie_4.0_accessible.pdf
[9] Karnouskos S (2011) Cyber-Physical Systems in the SmartGrid. In: IEEE 9th International Conference on Industrial Informatics (INDIN), Lisbon, Portugal
[10] Karnouskos S, Colombo AW (2011) Architecting the next generation of service-based SCADA/DCS system of systems. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia.
[11] Karnouskos S, Somlev V (2013) Performance assessment of integration in the cloud of things via web services. In: IEEE International Conference on Industrial Technology (ICIT 2013), Cape Town, South Africa
[12] Karnouskos S, Colombo AW, Jammes F, Delsing J, Bangemann T (2010) Towards an architecture for service-oriented process monitoring and control. In: 36th Annual Conference of the IEEE Industrial Electronics Society (IECON-2010), Phoenix, AZ.
[13] Karnouskos S, Savio D, Spiess P, Guinard D, Trifa V, Baecker O (2010) Real World Service Interaction with Enterprise Systems in Dynamic Manufacturing Environments. In: Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management, Springer
[14] Karnouskos S, Vilaseñor V, Handte M, Marrón PJ (2011) Ubiquitous Integration of Cooperating Objects. International Journal of Next-Generation Computing 2(3)
[15] Karnouskos S, Colombo AW, Bangemann T, Manninen K, Camp R, Tilly M, Stluka P, Jammes F, Delsing J, Eliasson J (2012) A SOA-based architecture for empowering future collaborative cloud-based industrial automation. In: 38th Annual Conference of the IEEE Industrial Electronics Society (IECON 2012), Montréal, Canada.
[16] Northrop L, Feiler P, Gabriel RP, Goodenough J, Linger R, Longstaff T, Kazman R, Klein M, Schmidt D, Sullivan K, Wallnau K (2006) Ultra-Large-Scale Systems – the software challenge of the future. Tech. rep., Software Engineering Institute, Carnegie Mellon, URL http://www.sei.cmu.edu/library/assets/ULS_Book20062.pdf
[17] Tranquillini S, Spiess P, Daniel F, Karnouskos S, Casati F, Oertel N, Mottola L, Oppermann FJ, Picco GP, Römer K, Voigt T (2012) Process-based design and integration of wireless sensor network applications. In: 10th International Conference on Business Process Management (BPM), Tallinn, Estonia
[18] Xu LD (2011) Enterprise systems: State-of-the-art and future trends. Industrial Informatics, IEEE Transactions on 7(4):630 –640, DOI 10.1109/TII.2011.2167156