System-of-Systems Orchestrator Pattern

An Industry IoT Consortium Architecture Pattern

2023-03-09

Authors

Hiroshi Yamamoto (Toshiba), Daniel Young (Toshiba).

Contents

Figures

1 General Information

1.1 Name

System-of-Systems Orchestrator Pattern.

1.2 Pattern Type

System Architecture Pattern.

1.3 Description

The System-of-Systems (SoS) Orchestrator Pattern is a collection of systems, each capable of independent operation, that interoperate together to achieve additional desired capabilities.

2 Context

As industrial IoT (IIoT) systems have matured, they have created opportunities to pool their resources and capabilities together to create new, more complex systems that offer more performance, functionality and overall benefits than the systems can provide by themselves. Linking these systems together do not require them to be modified, or even become aware of each other. In fact, these systems can be combined such that their independence from each other is maintained.

2.1 Typical/Known Scenarios

Scenarios include use cases where separately managed systems can be combined together to offer new services. The separately managed systems may be part of a single organization or part of multiple organizations.

2.2 Typical/known problems addressed

System-of-system patterns address the growing need for IIoT systems to combine their distinct resources together and offer new services that the individual systems cannot offer by themselves (for example, a feed-in premium service for the renewable energy market).

3 Model

At a high level, the IIoT system may be configured as one or more distributed, independent constituent systems, connected to a single SoS Orchestrator, which is connected to one or more SoS Services, creating a system-of-systems (see Figure 3-1).

System-of-Systems Orchestrator Pattern

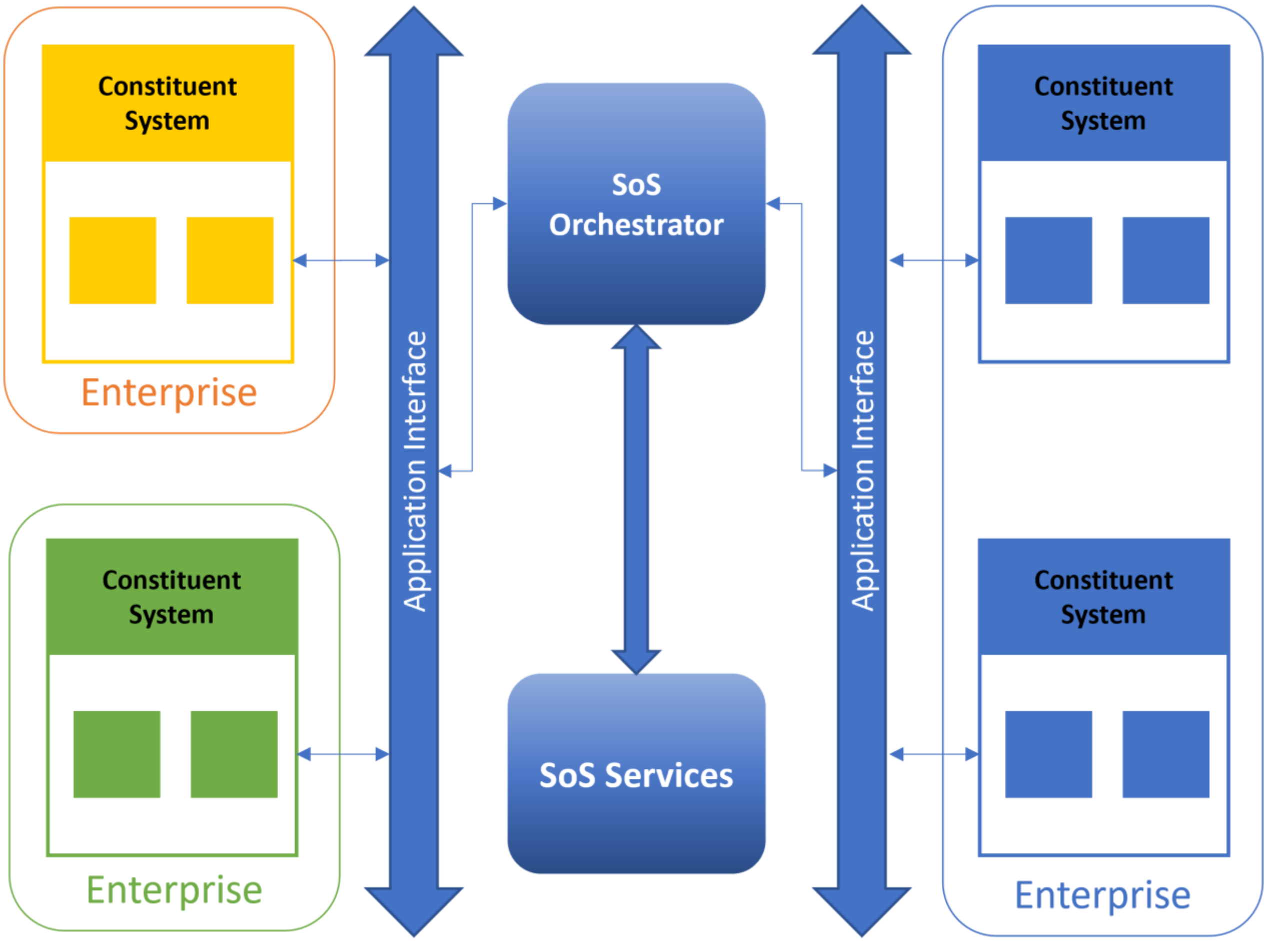

Figure 3-1. System-of-Systems.

The central characteristic of a system-of-systems is that they are formed from a variety of constituent systems: existing or legacy systems, newly engineered from the “ground-up” custom systems, and potentially tailored existing commercial-off-the-shelf (COTS] systems. In addition, five additional characteristics[1] can be found in most system-of-systems designs:

Operational independence of the individual systems: The constituent systems within the system-of-systems achieve well-substantiated purposes by themselves and continue to operate in this way and accomplish their individual purposes even if detached from the overall system.

Managerial independence of the systems: The constituent systems are managed in large part for their own purposes rather than the purposes of the whole.

---

[1] Mark W. Maier, "Architecting Principles for Systems of Systems", Wiley, 1999. *https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1520-6858(1998)1:4%3C267::AID-SYS3%3E3.0.CO;2-D*.

Geographic distribution: The constituent systems are most likely located or dispersed in various geographies–within a factory, a city, state, country or even worldwide.

Emergent behavior: While meeting the above constituent system needs, the system-of-systems enables new capabilities that are generally unachievable by the individual systems acting independently.

Evolutionary development: A system-of-systems may not appear fully formed and functional at the start. Its development is evolutionary in the sense that functions and purposes are added, removed, iterated, with experience in use of the system-of-systems.

Constituent systems can belong to the same or different enterprises (i.e. a business association or organization); they are managed independently, even when they belong to the same enterprise. Since constituent systems operate with no inherent dependencies on each other and may not be aware of each other's existence, their coordination is performed by the SoS orchestrator, which then connects to one or more SoS services.

The SoS orchestrator communicates and coordinates with the constituent systems over an application interface. The SoS orchestrator also gathers data from the constituent systems and provides insight acquired by analyzing this data to one or more SoS services. Because of the managerial and operational independence of each constituent system, it is the SoS orchestrator's role to communicate between them. The relationships between an SoS orchestrator to any constituent system is loosely coupled, since it is most likely that constituent systems are unaware of the existence of a SoS orchestrator. Therefore, communication is generally a publisher-subscriber (aka pub/sub) model.

SoS services take the collective resources of the constituent systems and offer new emergent behaviors in the form of additional capabilities. They provide end user interfaces, allowing these capabilities to be consumed by end users and other external systems (i.e. outside the SoS). The relationship between the SoS orchestrator and SoS service is tightly coupled, therefore client-server communication models may be more appropriate.

At the logical level, the pattern allows only one SoS orchestrator in each system-of-systems. All coordination between constituent systems and SoS services is handled by the SoS orchestrator. Data moves between a constituent system and the SoS orchestrator, and between an SoS service and the SoS orchestrator. The SoS orchestrator is a gatekeeper, allowing constituent systems to maintain their independence from each other while restricting them from having direct access to SoS services.

3.1 System-of-System Subtypes

There are two main subtypes, *vertical* system-of-systems and *horizontal* system-of-systems. The major difference between the two subtypes is whether the constituent systems belong to a single enterprise or to multiple enterprises.

3.2 Vertical System-of-Systems

A vertical system-of-systems (vertical SoS) contains independent constituent systems, each of which belong to the same enterprise. Although each constituent system has managerial independence from each other, they will share common enterprise-level concerns and tools, for example an enterprise-wide database. One area of common concern is security; another is enterprise objectives. The SoS orchestrator supervises each constituent system and manages a common set of shared enterprise objectives. A typical use case for a vertical SoS would be offering workload balancing or resource optimization services for disparate systems.

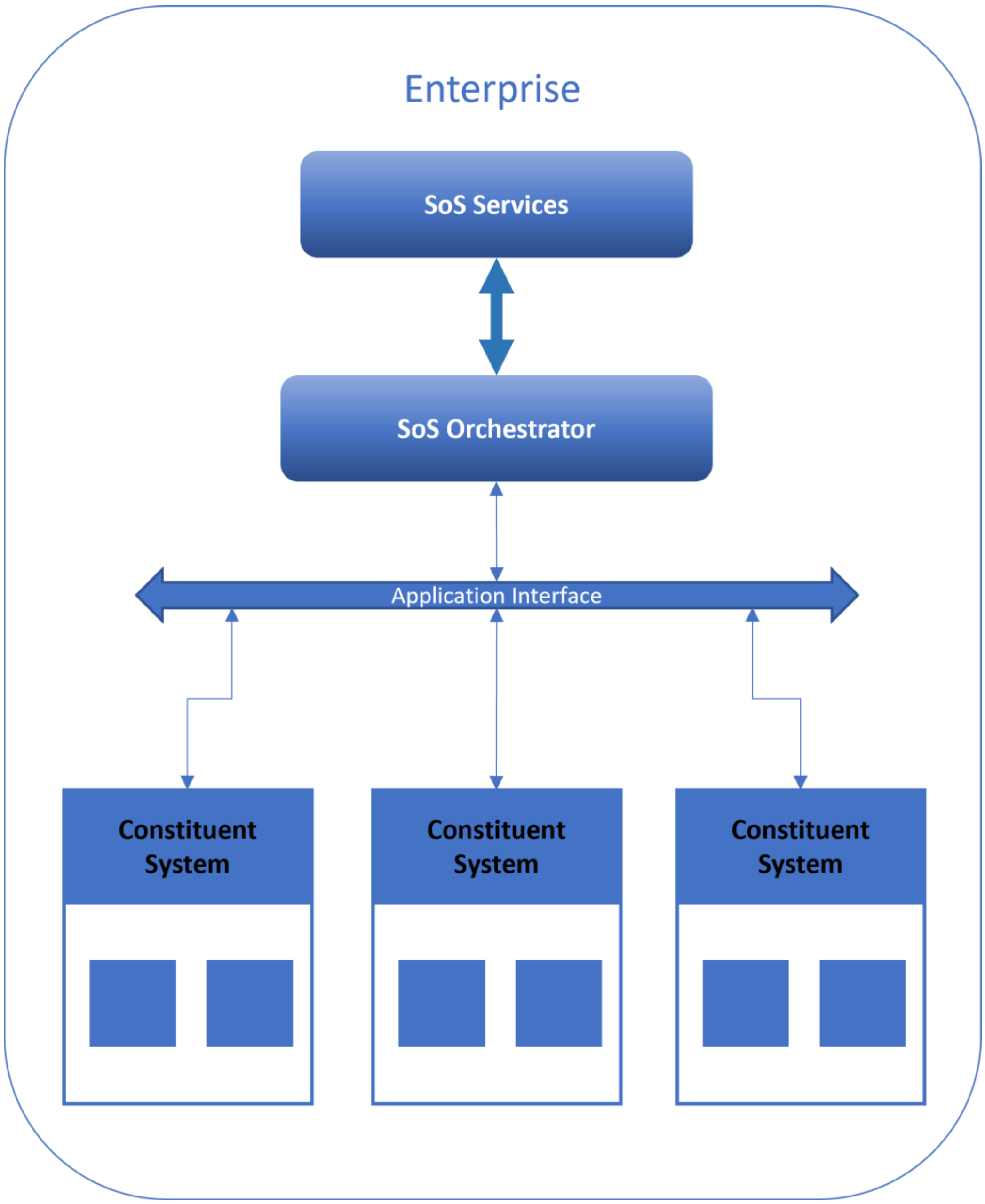

Figure 3-2. Vertical System-of-Systems.

3.3 Horizontal System-of-Systems

A horizontal system-of-systems (horizontal SoS) contains independent constituent systems that belong to different enterprises or industries. The SoS orchestrator coordinates the communication of each constituent system. The main activity of the SoS orchestrator is mediation for each system whose enterprise objectives may be different. For example, the SoS orchestrator and SoS services may be in an internet environment and not part of any constituent system enterprise. In this case, each system has its own independent security realm. A typical use case for a horizontal SoS would be correlating information gathered from each constituent system to provide new business services to other enterprises.

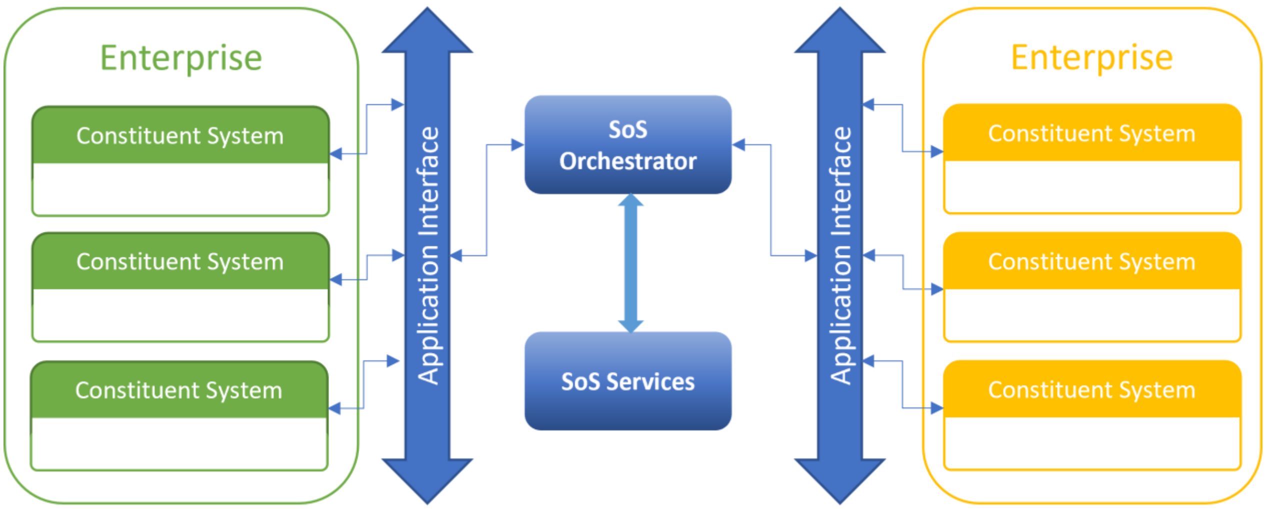

Figure 3-3. Horizontal System-of-Systems.

4 Guidance

Combining constituent systems to form a system-of-systems affects each viewpoint in the IIRA,[2] and new sets of concerns may be raised by the system-of-systems that were not original to the constituent systems themselves. The new concerns can have a cascading and comprehensive impact on both the system-of-systems and the individual constituent systems, especially in terms of system characteristics and trustworthiness. Systems-of-systems, while not necessarily large, are by nature complex, providing higher levels of sophistication and efficiency. The resulting system should be evaluated by its own set of metrics, and not just the net combination of the constituent system's metrics.[3]

[2] *www.iiconsortium.org/IIRA.htm*

[3] ISO/IEC/IEEE: "ISO/IEC/IEEE 21839 Systems and Software Engineering — System-of-systems (SoS) Considerations in Life Cycle Stages of a System", 2019. *https://www.iso.org/standard/71955.html*

4.1 ADVANTAGES

A system-of-systems offers new, emergent capabilities (i.e. SoS services) that constituent systems cannot offer on their own. To achieve these capabilities, the collective operation of constituent systems must be coordinated (by the SoS orchestrator); however, the orchestration does not need to be performed inside the enterprise(s) in which the constituent systems operate in.

4.2 DISADVANTAGES

The combination of independent systems into a system-of-systems results in higher complexity and requires additional considerations, especially in areas of cost, governance, operation, and trustworthiness.

5 EXAMPLES

5.1 VIRTUAL POWER PLANT (VPP)

The Virtual Power Plant manages and orchestrates the power generation of various electrical power and distribution systems (for example: coal-fired, photovoltaic and wind). By combining the power generation of each system together, the electric grid can become more adaptive to demand-response needs, and more stable as renewable sources power output fluctuates.

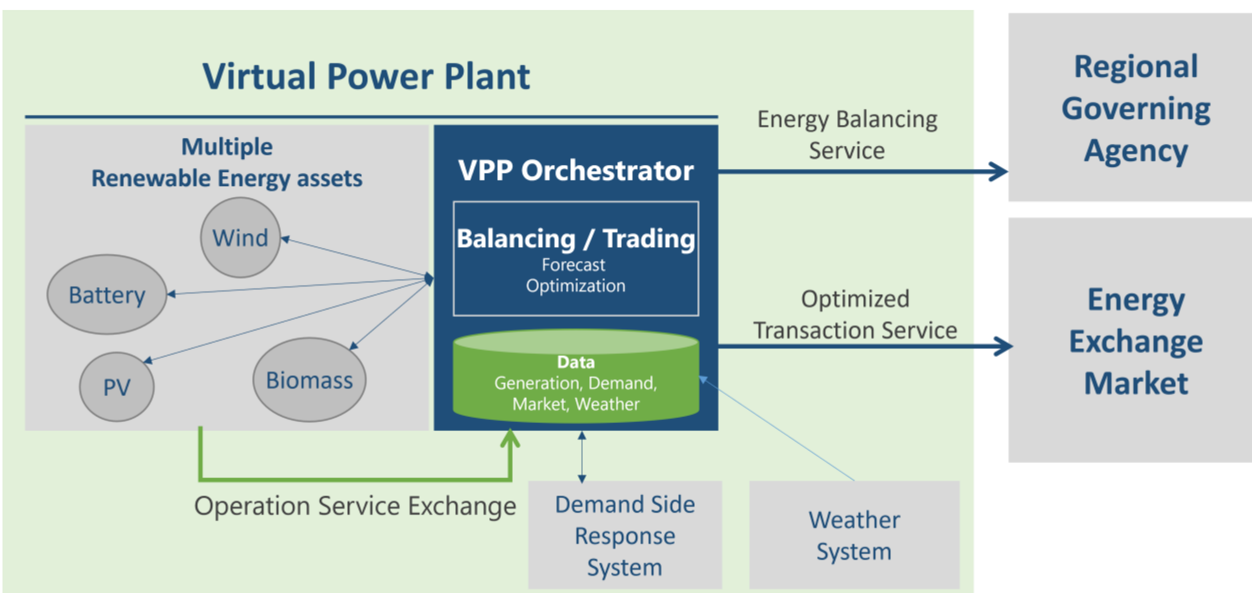

Figure 5-1. Virtual Power Plant SoS.

6 AUTHORS & LEGAL NOTICE

This document is a work product of the Industry IoT Consortium Patterns Task Group, chaired by Mr. Daniel Burkhardt (Steinbeis Transfer Center Innovationsforum Industrie) and Mr. Eric Simmon (NIST).

Authors: The following persons contributed substantial written content to this document: Hiroshi Yamamoto (Toshiba) and Daniel Young (Toshiba).

Technical Editor: Stephen Mellor (IIC CTO) oversaw the process of organizing the contributions of the above Authors and Contributors into an integrated document.