# SOSA® Module Integrability

## Achieving MOSA Business Capabilities and Sensor Performance Requirements

*The Open Group FACE® and SOSA® Consortia 2025 Technical Interchange Meeting (TIM), by:*


Jonathan A. Cain
Raytheon, an RTX Business
Vice Chair, SOSA® Consortium Steering Committee
Co-Lead, SOSA® Consortium Enterprise Guidance Committee


October 2025

# Table of Contents

# Executive Summary

The business of systems integration is often perceived to conflict with a Modular Open Systems Approach (MOSA) due to the demand for high-performing, cutting-edge systems. However, recent geopolitical events have shown that low-cost, mass-produced systems and horizontal markets can achieve mission objectives affordably. This raises the question: Where will MOSA be successful or not, and why? In this paper, we address this question from both business and systems engineering perspectives, focusing on the quality attribute of integrability. We conduct a case study using the Sensor Open Systems Architecture™ (SOSA®) Reference Architecture and Technical Standard. First, we introduce integrability from the perspectives of a business capability and separately as a system architecture quality attribute. We then examine the impact of alignment with the SOSA Technical Standard on Systems Verification and Validation (V&V), particularly in decomposing system performance specifications into hardware and software elements, as well as composing such into larger subsystems. We conclude by identifying business capabilities where horizontal markets currently exist, areas where vertical integration is likely to persist, and where the SOSA Technical Standard will have the greatest impact on cost savings due to technology obsolescence or sustainment phase upgradability.

# Introduction

## Vertical Integration as a Business Capability

The business of system integration with the US Department of Defense has a long, storied history, beginning in its modern form at the beginning of the Cold War [1]. Within this model, the primary business capabilities of systems integrators, whether contractors, primes, or Federally-Funded Research and Development Centers (FFRDCs), have been maintaining and developing in-house expertise on how to integrate components or elements,[1] thereby creating a complex system that meets the customer's requirements. Historically, poor component-level technology performance led to component interdependencies where vertical integration provided the only mechanism to attain the system-level performance necessary to meet the customer's requirements. Systems integrator's business organizations reflected this paradigm with optimized business capabilities for supply chain to outsource easily modularized (i.e., adequately performing, reliable components) to suppliers and in-house units responsible for producing corner-stone components of which direct control and tuning enabled the overall system to meet requirements [2] [3]. Much of this early systems engineering research and evolution as a discipline is encapsulated in the Systems V diagram, discussed in detail later in this paper.

Within the commercial industry, there have been many cycles of integration and disintegration. The computer revolution of the 1980s and 1990s saw an incredible transformation from vertical integration to a horizontal market, a central case study for the seminal work *The Innovator's Dilemma*, published by Clayton Christensen in 1997 [4]. Fundamental to Christensen's work is an understanding of a system performance gap versus a performance surplus. A performance gap exists when a technology's or system's performance is less than the customer's or end user's desired performance for that technology or system. That is, the technology or system has a lower performance than desired. On the other hand, a performance surplus is the difference between the actual performance and the desired performance when the actual performance is greater than the desired performance. See Figure 1. Performance Gap and Performance Surplus [5].

Within the defense community, the Modular Open Systems Approach (MOSA), began in earnest in the early 2000s as a direct result of what was witnessed in the computer industry and business strategy analyses like Christensen's, which go well beyond performance gaps versus surpluses [6] [7]. However, the defense industry's ability to adapt its acquisition strategy has not been as easy as perhaps first thought. In particular, the challenges that persist are "communicating and demonstrating government commitment, developing a MOSA-enabled [Intellectual Property] IP and data rights strategy, and establishing standards and interfaces" [8].[2] However, systems engineering modernization practices like Model-Based Systems Engineering (MBSE) [9] or Agile and the Systems Engineering Modernization Lifecycle [10], in their shear breadth of approach, have in many ways failed to pinpoint why traditional systems integrators have successfully maintained business as usual with respect to traditional vertical approaches to systems integration and stunted the disruption promised by MOSA.

[1] Throughout this paper, we use "components" and "elements" interchangeably.
[2] See references [31] and [32] for metrics and strategies that address this challenge.

### *SOSA® Module Integrability*

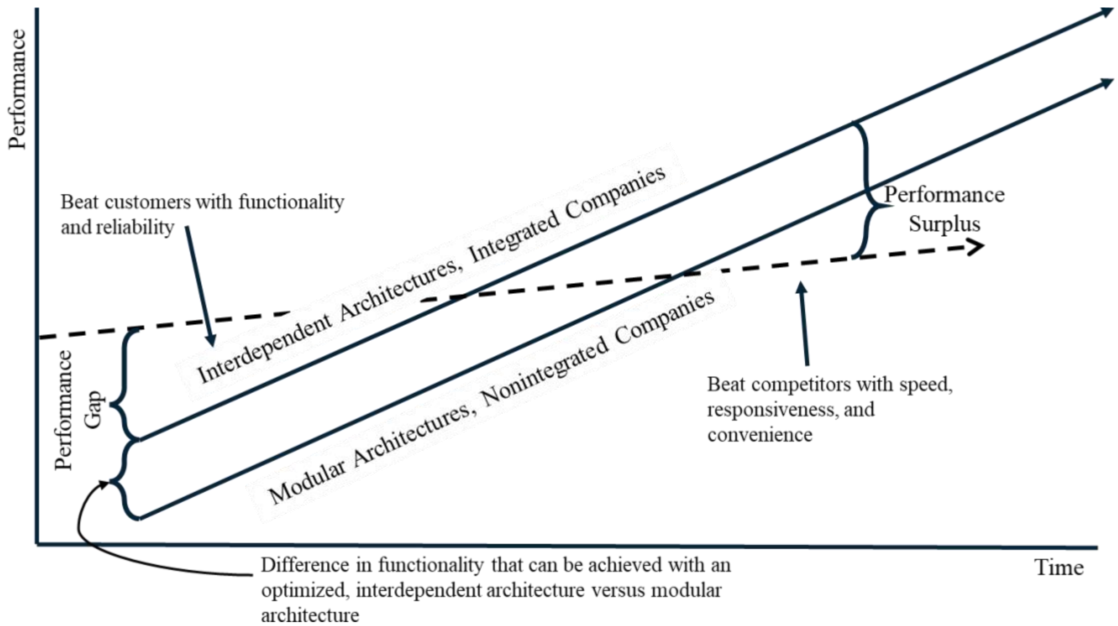



Figure 1: Performance Gap Versus Performance Surplus [5]

We suggest that while there are important and comprehensive approaches to furthering the adoption and maturation of MOSA within the defense community, most approaches fail to address a fundamental aspect of why and how systems integrators still maintain the business capability of vertical integration in most instances, even when MOSA is forefront. The approach we present below adds to this business theory through the examination of integrability of components or subsystems with or without performance gaps or surpluses to achieve overall system performance.

## Integrability as a Quality Attribute

Integrability is not included as one of the ten SOSA Reference Architecture Quality Attributes [11]. However, several recent papers address integrability as a software and/or systems architecture quality attribute. These investigations provide a foundation for our case study into the challenges and opportunities within SOSA Technical Standard to achieve its desired quality attributes by incentivizing industry to develop new or mature existing business capabilities.

The Software Engineering Institute (SEI), an FFRDC, has provided the following definition of integrability. Consider a set of components, $\{Ci\}$, and a system, $S$, for which the components must integrate. Then, "integration difficulty—the costs and the technical risks—can be thought of as a function of the size of and the 'distance' between the interfaces of $\{Ci\}$ and $S$:"

- Size is the number of potential dependencies between $\{Ci\}$ and $S$.
- Distance is the difficulty of resolving differences at each of the dependencies. [12] [13]

This definition is less a prescriptive quality attribute for some ideal architecture, but rather an analysis tool for describing the degree to which the architecture may be integrable. "Integrability analysis is an attempt to predict which changes to $S$ or $\{Ci\}$ will have to be made during a future integration task, based on how far apart $S$ and $\{Ci\}$ are" [13]. Integrability is a categorization of parameters, or systems- or component-level dependencies, each with an assigned qualitative or quantitative measurement ("distance") describing the ease at which a component is integrable into the system. The SEI approach includes a "Playbook for an Architecture Analysis on Integrability," which is seven-step process in three phases. The steps are as follows:

**Step 1:** Collect artifacts (preparation phase).
**Step 2:** Identify the mechanisms used to satisfy the requirement (orientation phase).
**Step 3:** Locate the mechanisms in the architecture (orientation phase).
**Step 4:** Identify derived decisions and special cases (orientation phase).
**Step 5:** Assess requirement satisfaction (evaluation phase).
**Step 6:** Assess impact on other quality attribute requirements (evaluation phase).
**Step 7:** Assess the cost/benefit of the architecture approach (evaluation phase). [13]

Since the introduction of the SEI approach, there have been subsequent developments to integrability useful for our case study. In particular, the consideration of specific intent within some integration scenario. Integrability can then refer to one of the following intents:

**Intent 1:** The ability of a component to be integrated into one or more existing systems. Here, integrability is assessed on the component being integrated.
**Intent 2:** The ability of a system to integrate one or more additional components. Here, integrability is assessed on the system.
**Intent 3:** The ability of two or more components to be integrated to form a system. Here, integrability is assessed on the joining of the two components. [14]

With respect to MOSA, we see that the first intent applies to the goal of solving technology obsolescence and that the second applies to upgradeability during the sustainment phase. However, all three intents are often co-mingled and confused when addressing MOSA holistically.

Prior to MOSA, contractors would rely on vertical integration and begin within the scope of the third intent, historically applying to most Major Development Acquisition Programs, complete control over which "components are designed and developed for integration into a specific weapon system" [15]. The hope would be that good development meeting systems performance requirements would guarantee the same contractor would be chosen as not only sole-source producer for the system, but also sole-source for future sustainment-phase upgrades. This is the second integrability intent above and what has been termed "vendor-lock" in vertically-integrated systems by MOSA-minded customers, and "legacy or franchise systems" by the contractors. Continuing to the first intent, only a few large primes have emerged with enough systems in development or production for product lines of components to be used in one or more existing systems. This was considered a cost differentiator, albeit cornered by the primes themselves, "achieved through the systematic reuse and integration of components" [15]. Thus, historically, contractors have started with integrability in the first intent, matured through the second, and achieved prime or large-scale integrator status by reaching the third.

With MOSA, the government is going the opposite direction: Gaining control of the first and actively finding ways to get to the second and third. They are using Open Systems Architectures (OSAs) and Government Reference Architectures (GRAs) to ensure that the cost-benefits of product line approaches flow back to the customers, not left with the primes. This is especially true of the first intent above, with direct applicability to the SOSA Technical Standard, and even more matured within the industry surrounding the Future Airborne Capabilities Environment (FACE™) Technical Standard [16]. For the second intent, the OSAs and GRAs push towards a "system architecture that allows severable major system components at the appropriate level to be incrementally added, removed, or replaced throughout the life cycle of a major system platform to afford opportunities for enhanced competition and innovation" [17]. This has been a larger challenge due to the system's need to maintain or even improve its performance, and one to which we now present two case studies from the SOSA Technical Standard.

# Integrability Analysis of the SOSA Technical Standard

In the previous section, we introduced the business capability of vertical integration and its application to systems with performance gaps. We also introduced recent scholarly research on the quality attribute of integrability. We now turn to the analysis of integrability for our case study within the SOSA Technical Standard using the approaches of [13] and [14]—one instance with a performance surplus and another with performance gap.

## EO/IR Case Study Description

Consider a sensor system composed of SOSA Software (SW) Elements with the intended use of Electo-Optical/Infrared (EO/IR) Surveillance, Reconnaissance, and Intelligence (ISR), which includes AI-enabled detection, tracking, and target classification. In this case study, we do not consider SOSA Hardware (HW) Elements, but the system can be hypothetically conceived of as a turreted sensor system containing HW Elements like SOSA-aligned 3U VPX. The systems is deployed to several fixed and rotary wing aircraft, and to simplify, we only consider passive EO/IR, and with no Semi-Active Laser capability.

Within the software layer, we adapt the Services View 2 (SvcV-2) from the SOSA Technical Standard [11], with and additional hypothetical example of the SOSA 3.X “Process Signals/Targets” Modules based on [18] and [19]. Figure 2 shows the SvcV-2 for our case study. The deltas with the SOSA SvcV-2 are as follows:

1. Data flow internal to the signal processing modules is specified by the by the dotted purple arrows.
2. Signal processing boundaries input and outputs are specified to specific SOSA Module 3.Xs with solid purple lines
3. SOSA Module 2.4 “Emitter/Collector” is removed.

This SvcV-2 is a view of our hypothetical EO/IR Case Study instance architecture, which is derived from the SOSA Reference Architecture.

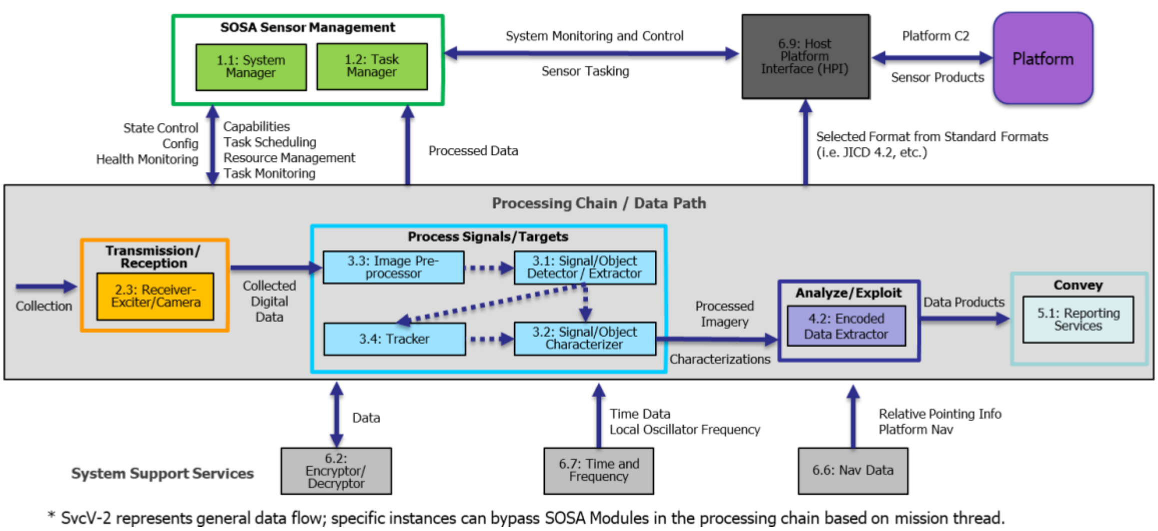



Figure 2: Services View 2 (SvcV-2) for Integrability Case Study

## *SOSA® Module Integrability*

### Scenario 1: Performance Surplus & SOSA Module 1.1 "Systems Manager"

SOSA Module 1.1 "Systems Manager" is well-defined and mature within the SOSA Technical Standard. It provides system management functionality to and for SOSA Modules (and SOSA HW Elements) via a set of system management services. "The ability to manage and interact with these system management services is enabled through well-defined sets of APIs and sensor component messages" across three groups:

- Health includes status, faults, and debugging.
- Operations include state, mode, and application support.
- Configuration includes the sensor inventory, its modules, its hardware and software components, their interconnections, and configuration parameter values. [11]

Turning to the EO/IR Case Study, we claim all the functions and interactions of our hypothetical Systems Management Module are well defined through the SOSA Technical Standard. But how replaceable is the Systems Management Module for a potential future upgrade, i.e., how *integrable* is a new Systems Management Module?

To this scenario, we first employ the SEI "Playbook for an Architecture Analysis on Integrability", which is a three-phased, seven-step process [13], where $C_{Orig,SM}$ is the original SOSA Module 1.1, $C_{New,SM}$ is the new SOSA Module 1.1, and the EO/IR Case Study sensor system transitions from $S_{Orig}$, which contains SW Component $C_{Orig,\mathrm{SM}}$ ($\{C_{Orig,\mathrm{SM}}\} \in S_{Orig}$) to a new system, $S_{New}$, with $\{C_{New,SM}\} \in S_{New}$. Table 1 shows the scenario.

Table 1: Remove and Replace of SOSA Module 1.1 "Systems Manager"

| Scenario Part | Value |
|---|---|
| Source | Technology Obselescence |
| Stimulus | Update/Upgrade Systems Management Module |
| Artifacts | $C_{Orig,SM}$ , $C_{New,SM}$ , $S_{Orig}$ |
| Environment | System has been deployed |
| Response | $C_{New,SM}$ is fully integrated, forming new sensor system: $S_{New}$ |
| Response Measure | - Deployed in 1 month with no more than 1 person-month of effort<br>- Passes Formal Quality Test (FQT): 100% of SW requirements (from SRS) and system performance requirements (from SPS)[3] |

For Step 1, we collect all the architecture artifacts, which in this hypothetical case refers to Table 1, Figure 2, and the SOSA Reference Architecture Quality Attributes. In practice, there ought to be more artifacts. In Step 2, we begin to list the mechanisms (tactics and patterns) that can satisfy the integrability requirement. For our

[3] We assume that the Software Requirements Specification (SRS) and Systems Performance Specification (SPS) are unchanged from $S_{Orig}$ to $S_{New}$ so that the FQT procedures remain exactly the same.

SOSA Module 1.1, this is quite simple, since the Module is both well-defined in the SOSA Technical Standard and inherently includes many of the patterns and tactics. The four patterns of interest from [13] are:

1. Adhere to standards—e.g., the SOSA Technical Standard.
2. Encapsulation—the boundaries of the Module are well-defined in the SOSA Technical Standard.
3. Configure Behavior—the Module is the primary method by which the sensor system controls operations like states and modes.
4. Manage Resources—the Module is the primary method by which the sensor system configures of inventory, among other responsibilities.

The combination of these four patterns positively addresses the challenge of integrability across size, syntactic distance, data semantic distance, behavioral semantic distance, temporal distance, and resource distance. We leave discussion of tactics to the reader.

Step 3 is to locate the mechanisms in Step 2 within the sensor system architecture. While this should be done in further detail, and under the to-be completed guidance of the SOSA Reference Implementation Guide [20], the SvcV-2 in Figure 2 shows that the Systems Management Module is largely independent and autonomous from the other SOSA Modules. The Systems Management Module, as defined by the SOSA Technical Standard, can easily check the boxes for integrability with the proper instance architecture: Allocation of Responsibilities, Coordination Model, Data Model, Mapping Among Architectural Elements, Resource Management, Binding Time,[4] and Choice of Technology. We will find that this is not the case in the next section where we see those choices on how and when to include mechanisms like the four patterns above have profound impacts on the overall integrability of a SOSA Module 3.3 "Image Pre Processor."

In Step 4, we identify the decisions and special cases. The interfaces to the other SOSA Modules are well-defined, though numerous, given that the Systems Management Module interfaces with all other SOSA Modules. The primary areas to understand are the resource allocations and non-interference in binding times between the temporally interdependence SOSA 3.X Modules. For example, if $C_{Orig,SM}$ was originally placed on a HW Element, which was shared with other SOSA Modules, we would need to understand the basis for that decision in terms of memory usage, throughput, and possibly power draw. Similarly, it is assumed that $C_{Orig,SM}$ was originally configured to dynamically interact with the other SOSA Modules so that it does not require health status reporting from a SOSA Module like 3.3 "Image Pre Processor" in the middle of a mission critical task within that module.

Now that the preparation and orientation phases are complete, we evaluate the scenario against the requirements. In Step 5, the response measures—FQT for SRS and SPS compliance—are systematically considered from an architectural perspective. In practice, this represents a significant undertaking and would uncover several issues. For example, it is likely that the Systems Management Module is statically linked, due to its interactions across many SW and HW Elements. This would increase integrability of $C_{New,SM}$ because all the SW Elements of $S_{New}$ would need to be recompiled, which in this instance potentially adds cost and schedule.

[4] "Binding Time" refers to when the element integrates with the rest of the system. For example, SOSA Module 3.3 could have a binding time at system boot for initialization/configuration, then also periodically (at frame rate) during runtime as a measurement update containing each new image.

### *SOSA® Module Integrability*

The final two steps, "assess impact on other quality attribute requirements" and "assess the cost/benefit of the architecture approach" are left to the reader. We conclude this section with Table 2: Conditions for Integrating Identical Functions, summarizing the approach outlined in [14], applied to our hypothetical use case for Scenario 1.

The results from Scenario 1 show that the SOSA Systems Management Module is integrable, albeit tedious, given the many interactions and functions needed throughout the system. Compared with Scenario 2, which we analyze next, Scenario 1 is highly integrable. The only potential downside mentioned above would be significant compute-resource challenges—an integrability dimension influenced by a sensor system performance gap. Putting this aside, the Systems Management Module itself has a performance surplus in that it is quite easy to make a new module and have it function identically, i.e., all verified SRS requirements for $C_{Orig,SM}$ will also be verified for $C_{New,SM}$. Furthermore, the remove and replace scenario would not likely affect the pre-existing performance gap or surplus of the sensor system as verified during FQT of the SPS on $S_{New}$. In conclusion, remove and replace of the SOSA System Management Module in a sensor system has only few dimensions where system-level performance surpluses or gaps affect integration. Thus, it is easy to see why the SOSA Systems Management Module is an ideal candidate for a horizontal market, driving down costs and increasing market supply.

Table 2: Conditions for Integrating Identical Functions – SOSA Systems Management Module

| Element | Value | Description |
|---|---|---|
| Object | Functional Integration | Remove and replace SOSA Systems Management Module in existing system |
| Desired Behavior | Identical functions | $C_{Orig,SM}$ is functionally equivalent to $C_{New,SM}$ ($C_{Orig,SM} \equiv C_{New,SM}$) and $S_{Orig} \equiv S_{New}$ |
| Intent | Component of a system | $\{C_{New,SM}\} \in S_{New}$ |
| Usage Scenario | Operator in SW lab | Integration by SW engineers, FQT by quality engineers |
| Symmetry | Symmetrical | Removing $C_{Orig,SM}$ requires disintegration of $S_{Orig}$ and integration of $C_{New,SM}$ into $S_{New}$ |
| Activity | End State & Process/procedural | The end state is $S_{New}$ and the process of integration is required to be verified in FQT |
| Directionality | Bi-Directional | $C_{New,SM}$ must operate correctly for $S_{New}$ to operate correctly |
| Success Criteria | Probabilistic target & Extent Desired aspect | Integration schedule is measured probabilistically and $S_{New}$ FQT is deterministic |

## Scenario 2: Performance Gap & SOSA Module 3.3 "Image Pre Processor"

SOSA Module 3.3 "Image Pre Processor" is also well-defined and mature within the SOSA Technical Standard. It provides image processing functionality like non-uniform correction, image sharpening like an inverse filter or Roberts Cross, or de-smear and de-roll functionality. Like the previous scenario, we employ the "Playbook for an Architecture Analysis on Integrability" [13], where $C_{Orig,IPP}$ is the original SOSA Module 3.3 and $C_{New,IPP}$ is the new SOSA Module 3.3. The EO/IR Case Study sensor system transitions from

$S_{Orig}$, which contains SW Component $C_{Orig,\mathrm{IPP}}$ ($\{C_{Orig,IPP}\} \in S_{Orig}$), to a new system, $S_{New}$, with $\{C_{New,IPP}\} \in S_{New}$. Note that the source of the scenario is different: In Scenario 1, the Systems Manager was being replaced due to technology obsolescence. The source of Scenario 2, however, is a system performance upgrade, which is required by new SPS for $S_{New}$. Specifically, the new system must increase its Probability of Identification ($P_I$) by 5% with no other effects on existing system performance requirements. The assumption is that we can integrate a new SOSA Module 3.3, without affecting downstream modules like SOSA Module 3.4 and 3.2 (see Figure 2). We discuss the decomposition of system performance requirements in the next section following our analysis of Scenario 2. Table 3 shows Scenario 2, to which we highlight the deltas with Scenario 1; we leave the full analysis to the reader.

Table 3: Remove and Replace of SOSA Module 3.3 "Image Pre Processor"

| **Scenario Part** | **Value** |
|---|---|
| Source | System Performance Upgrade/New Mission Requirements |
| Stimulus | Update/Upgrade Image Preprocessor |
| Artifacts | $C_{Orig,IPP}$ , $C_{New,IPP}$ , $S_{Orig}$ |
| Environment | System has been deployed |
| Response | $C_{New,IPP}$ is fully integrated, forming new sensor system: $S_{New}$ |
| Response Measure | - Deployed in 1 month with no more than 1 person-month of effort<br>- Passes FQT: 100% of SW requirements (from SRS) and system performance requirements (from SPS) |

For the "Playbook" Step 2, the first major delta is that neither encapsulation nor behavior configuration patterns easily apply to SOSA Module 3.3: This is because there is a great behavior semantic distance representing the dependence, for example, of SOSA Module 3.4 "Tracker" tunings on the tuning thresholds within the Image Pre Processor. This distance is quantifiable and is addressed in the following section, Decomposition of Performance Requirements.

In Step 3, the location of the mechanisms for integrability are quite different than SOSA Module 1.1, from which we inherited a great roadmap in the SOSA Reference Architecture. For SOSA Module 3.3, there needs to be much more specified detail in the instance architecture regarding how it (i.e., $C_{Orig,IPP}$) was integrated with respect to the patterns listed in Step 2: Encapsulation and Behavior Configuration. While the SEI "Playbook" rightfully includes the Binding Time integration issue of a similar signal processing component [13], they do not mention the performance dependency, which is a new dimension in the integrability size. The SOSA Tracker Module tuning example in the previous paragraph is one such example of a performance dependency. Furthermore, the method of evaluation (how other areas within the instance architecture are affected) for behavior semantic distance must also be included to effectively integrate $C_{New,IPP}$. For example, image contrast or image entropy are methods of evaluation for SOSA Module 3.3, but with the downstream dependency on our SOSA Tracker Module, so then is the Probability of Track ($P_T$).

Step 4 takes a closer look at this dependency in terms of the decisions or special cases that drive it. SOSA Module 3.3 likely would require many additional decisions and an examination of special cases, more so than SOSA Module 1.1. For example, we claimed that SOSA Module 1.1 ought to consider resource management like throughput and memory on a HW Element (compute) perspective. While this is not a difficult (yet potentially cumbersome) task during a remove and replace scenario with a performance surplus, for SOSA Module 3.3, there is an additional requirement for meeting throughput at frame rate. During the original integration of all $\{C_{Orig,}\} \in S_{Orig}$, throughput was established as an architectural decision: For SOSA Module 1.1, this was likely *dynamic* or *heuristic*, based on understand when to or when not to ask for health or status during post-initialization runtime. However, for SOSA Modules 3.X, this decision was likely a tradeoff, common with performance gaps. In this case, the tradeoff is:

- Allocation of the available throughput within frame rate of the four SOSA Module 3.Xs so that their combined runtime could complete processing a new measurement from the camera within frame time on the desired HW (compute) Element. This is measured *probabilistically* through a spectrum of benign to complex scenes, where frames could potentially be dropped.
- Overall performance (also measured *probabilistically*) of the sensor system and SOSA Modules 3.X, which includes $P_T$, and other downstream modules, including a high probability of correct classification, with a low probability of incorrect classification.

In Step 5, we claimed that the SRS and SPS were likely sufficient to evaluate the response measures for integrating a new SOSA Module 1.1. However, Steps 1-4 of the integration of SOSA Module 3.3 show a much greater complexity and degree of difficulty for integration. It is unlikely that the SRS or SPS would go to this level of detail in their specifications (e.g., allocation of throughput of signal processing modules within frame rate). This information also doesn't exist in the SOSA Technical Standard. Thus, it is up to the instance architecture documentation to carry forward this "tribal knowledge" that may or may not persist through the sustainment phase of a particular system. Given all this, the response measure of 1 month deployment seems to be at risk even from inception: We have demonstrated that Scenario 2 is much more complex and difficult (by design), than Scenario 1. We leave Steps 6 and 7 to the reader.

Table 4 shows the conditions for integrating new capability, which for the desired behaviour in the scenario augments the approach in [14]. While the SOSA Technical Standard creates ideal, MOSA-aligned conditions for integrating identical functions, we must note that the increase in capability required for Scenario 2 is outside the realm of MOSA, explicitly asking for an upgrade in performance. Increasing performance is a business capability—the ability of the new component's functions to perform better than the old one. This performance delta is what companies desire to protect as their Intellectual Property (IP), increasing their competitive advantage, creating innovative new capabilities for the customers and end users.

The scenario also highlights the difference between what is desired and what is current reality (i.e., state of the art) with respect to integration symmetry (see row of Table 4). The desire is for complete symmetry like the Scenario 1, which is remove and replace of SOSA Module 1.1 due to technical obsolescence. Scenario 2 is much more complex, involving many more interdependencies with other modules and a correspondingly larger integration size. In reality, upgrading an existing system's performance (and especially closing a performance gap) is not symmetric, since the tuning of the new system is obviously not transferable back to the original system.

Scenario 2 shows a contrast in the integrability effort required for Scenario 1. There is a new dimension, and additional size to the integration for Scenario 2, directly related to the dependence on the sensor systems performance on the performance a singular SW component (SOSA Module 3.3). However, we made clear that the SOSA Technical Standard will still reduce cost and schedule, even in the more recalcitrant Scenario 2. With these two scenarios described for our use case, we now focus on understanding how to best accomplish Scenario 2. We first turn to an examination of how performance requirements must be decomposed and maintained at much finer degree of detail to address the integrability challenge.

Table 4: Conditions for Integrating for New Capability – SOSA Image Pre Processor Module

| Element | Value | Description |
|---|---|---|
| Object | Functional Integration | Upgrade SOSA Image Pre Processor Module in exisisting system |
| Desired Behavior | Identical functions, improved capability | $C_{Orig,IPP} \equiv C_{New,IPP}$ and $S_{Orig} \equiv S_{New}$, but new capability is required for $C_{New,IPP}$ and $S_{New}$ |
| Intent | Component of a system | $\{C_{New,IPP}\} \in S_{New}$ |
| Usage Scenario | Operator in SW lab | Integration by SW, performance tuning by signal processing engineers, and FQT by quality engineers |
| Symmetry | Desire: Symmetrical<br>Reality: Asymmetrical | In reality, removing $C_{Orig,IPP}$ with disintegration of $S_{Orig}$ would not imply integration of $C_{New,SM}$ into $S_{New}$ could be accomplished without tuning other SOSA Module 3.Xs |
| Activity | End State & Process/procedural | The end state is $S_{New}$ and the process of integration is required to be verified in FQT |
| Directionality | Bi-Directional | $C_{New,CM}$ must operate correctly for $S_{New}$ to operate correctly |
| Success Criteria | Probabilistic target & Extent<br>Desired aspect | Integration schedule and $S_{New}$ FQT is measured probabilistically and is deterministic |

## Requirements Decomposition and Systems Verification and Validation (V&V)

In this section, we examine in further detail the case study Scenario 2, from the perspective of systems engineering. This scenario seeks to upgrade the sensor system so that there is an increase Probability of Identification by 5%. We further expand on Scenario 2 by supposing the upgrade is being performed by the original equipment manufacturer of the sensor system, which is also a large-scale integrator or prime. The left side of Figure 3 shows the original sensor system probability tree from the development program, and the Probability of Identification is defined as the decomposition: $P_I = P_D \times P_T \times P_C$, where $P_D$ is the Probability of Detection, $P_T$ is the Probability of Track, and $P_C$ is the Probability of Classification. A simple breakdown of the requirement is $P_I > 70\% = 88.79\% \times 88.79\% \times 88.79\%$, such that $P_D = P_T = P_C$. In reality:

> "Sometimes one component maker may have a problem that it can solve only at great expense but could be solved much more easily by shifting the requirements of a different [adjacent] component or by altering the interface standard in a way that would cost other component manufactures less" [21].

In fact, this exploitation of margin in adjacent component areas is the same as the degrees of freedom necessary to overcome performance gaps through vertical integration [5].

### *SOSA® Module Integrability*

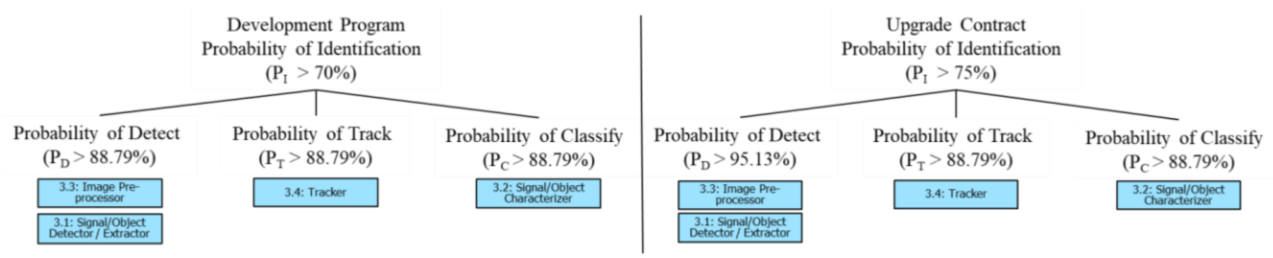



Figure 3: Original (Left) and Updated (Right) Sensor System Probability Trees.

The right side of Figure 3 shows the proposed systems performance requirements decomposition in Scenario 2. If we chose to replace only SOSA Module 3.3 "Image Pre Processor", then we leave the performance of $P_T$ and $P_C$ untouched, such that $P_T = P_C = 88.79\%$. This means that updated SOSA Module 3.3 "Image Pre Processor", in combination with the original SOSA Module 3.1, must achieve $P_D > 95.13\%$. This seems achievable and likely more cost effective than updating all the modules so that each $P_D = P_T = P_C > 90.86\%$. This is the new dimension to the size of the integrability distance arrived at in the previous section for Scenario 2—the top-down decomposition of performance requirements specifications.

Our SvcV-2 from Figure 2 shows that there is an additional, related dimension that is orthogonal to the top-down decomposition—the placement within the signal processing chain. All the subsequent SOSA 3.X Modules consume the processed image from SOSA Module 3.3 "Image Pre Processor". Thus, it is highly unlikely that changing $P_D > 95.13\%$ would imply that $P_T = P_C = 88.79\%$. In fact, a potential better approach for our large-scale integrator would be to update SOSA Module 3.2 "Signal/Object Characterizer", which is the last SOSA Module in our signal processing chain. This way, there are no downstream effects on other modules, which eliminates this additional integrability dimension.

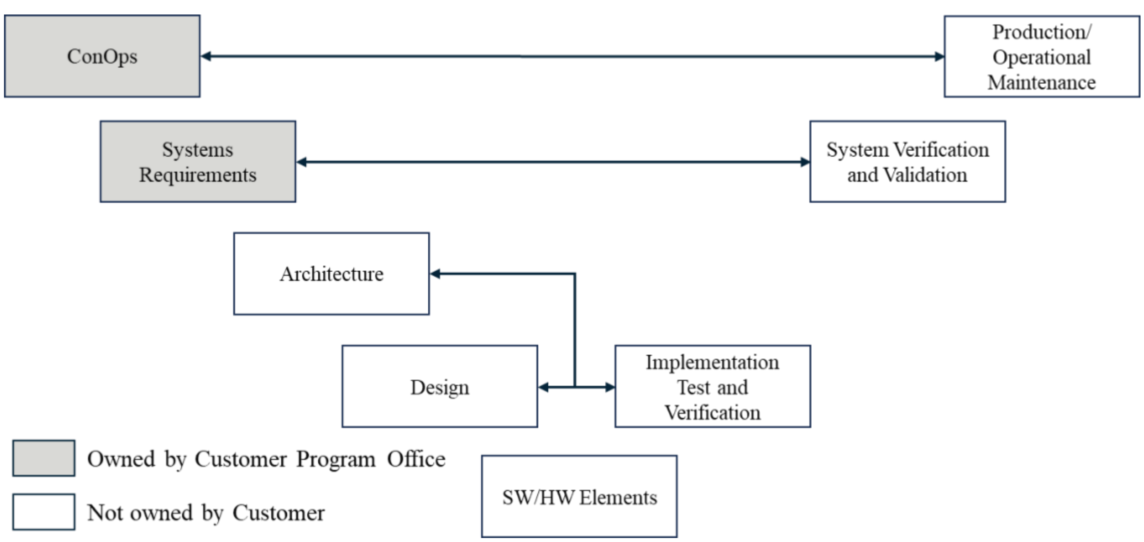



Figure 4: Systems V Diagram

### *SOSA® Module Integrability*

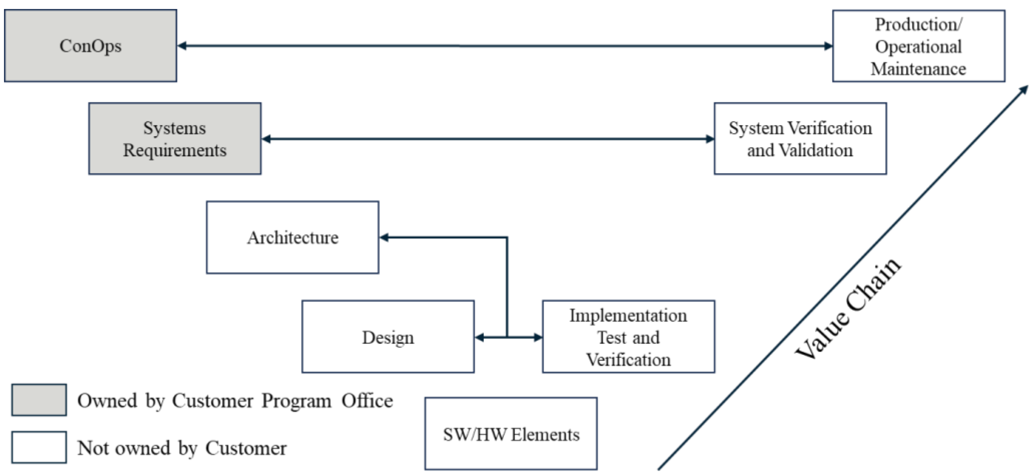
While we have discovered two additional dimensions of integrability due to the decomposition of performance requirements, we also must discuss the verification of these requirements through the Systems V&V process. Figure 4 shows a Systems V diagram. Starting on the top-left, the Customer Program Office (PO) owns the ConOps and Systems Requirements stages. It is here that the top-level requirement for $P_I$ is contained in the SPS. The SRS is also provided to the contractor/integrator by the PO. The rest of the Systems V diagram is owned by the integrator, which in the Design stage, chose to decomposed $P_I$ into $P_D$, $P_T$, and $P_C$. Then, the actual algorithms are developed in the SW/HW Elements stage, for placement into the SOSA 3.X Modules.

The Implementation Test and Verification is the first stage the integrator would piece the signal processing chain together in order to test each of $P_D$, $P_T$, and $P_C$. During the initial development stage, it is unlikely that they meet specification, so the integrator sends them back to the SW/HW Elements stage for further tuning or development, or they can reallocate the probabilities or even change the design in the Design stage. Either way, traditional vertical integration allowed for the integrator to move back and forth between these stages until $P_I$ meets performance specification. The same is true for the SW Modules/Elements integrated onto the HW Elements, as tested against the SRS in the System Verification and Validation stage. At this stage in both the development program and upgrade contract, the integrator has closed the integrability distance, including the two additional dimensions discussed above, and they have verified by the FQT. The system would now be ready for operational validation by the customer PO and end users before moving onto production.

# Business Capability Analysis

## Vertically-Integrated, non-SOSA Approach

Within the vertical integration paradigm, the development program's decomposition of performance requirements is a long-standing, well-documented process undertaken by primes and large-scale integrators. It is an initial task for designing and developing the vertically-integrated system or product so that the end results can then go through the Systems Verification and Validation (V&V) stage prior to the Production/Operational Maintenance stage. Completion of this stage often implies that the integrator then becomes the future original equipment manufacturer. To understand how business capabilities are derived, we apply a value chain to the right side of the Systems V diagram. Figure 5 shows a Systems V diagram with value chain added. As mentioned above, the customer PO owns the ConOps and Systems Requirements




stages.

Figure 5: Systems V Diagram with Value Chain for Vertically Integrated (non-MOSA) Systems with Performance Gap

The Architecture and Design stages happen during the development phase and typically represent only a small portion of the development contract's value. The optimal systems engineering effort in the Architecture and Design stage is approximately 15–20% of the total project effort [22]. However, on some programs these stages, while still producing technical value, are flat, or even negative, with respect to total development contract profitability due to poor systems engineering discipline or design flaws introducing overruns realized further in the development cycle [23]. When development and production contracts are considered together over the lifecycle of the product, the value in the labor hours associated with Architecture and Design stage on the development contract are often dwarfed in value (maybe 5%) by the right side, due to the shear profit generated by selling systems in the Production/Operation Maintenance stage.

Competitive bids by large-scale integrators for a development program include resources from the SW/HW Elements stages that are the primes common products (IP) or Commercial Off the Shelf (COTS) so that the contract does not need to pay for developing them from scratch, unless explicitly desired by the customer PO. The business capability of common products or product families shifts schedules to the left and provides known capabilities to the baseline system—both increase competitiveness of the bid.[5] The contract cost then lies in the labor hours for the engineering team to deploy SW/HW elements (the resources in the SW/HW Elements Stage) in the Implementation Test and Verification stage, which includes nearly all functional engineering disciplines—mechanical, electrical, software, systems, etc. When using common components, the more value is generated by more efficient and optimal processes deployment (integration, tuning, and testing). We define a stage for "Implementation Test and Verification" rather than "Systems Integration and Test" to highlight the need that each discipline has its own approach, best practices, and tools for deploying its components.

The grouping of Design, SW/HW Elements, and Implementation Test and Verification stages can happen at the smallest component level, and then grouping together into larger-and-larger subsystems until finally the System Verification and Validation stage is reached. In the previous section, we described this as a business capability for Scenario 2. To get to this point, each of the smaller subsystem groupings has assigned or emergent subsystem functional and performance requirements. Performance and functional requirements may be formally decomposed, and in systems with interdependent architectures, the architecture and design functions specify detailed requirements and allocations (functional and performance) for each subsystem and component. As alluded to earlier, flexibility in the decomposition of performance requirements throughout the system development lifecycle is a key business capability of primes for vertically-integrated systems.

Returning to Figure 5, the value chain arrow points to the right, because more and more labor and material are needed for the larger and larger subsystems to reach the final System Verification and Validation stage. Here, operational testing often has the highest value, percentagewise, in a development contract and in some (but certainly not all) cases, the labor and materials for a test can equal or exceed the total sum cost of procurement and development of resources within the SW/HW Elements stage. Finally, the most value exists in the Production/Operational Maintenance stage of the system. The value of production contracts is measured in units completed and sold, not in materials and labor. Although labor and materials are part of production, operations, or maintenance contracts, each often has higher margin, and therefore greater profitability than development contracts. This is especially true for exportability and Foreign Military Sales. Thus, as we move stages from left to right, we produce more and more value along the value chain in our Systems V diagram.

### Potential SOSA Approach – the Federation of Business Capabilities in Horizontal Markets

Figure 6 shows a Systems V diagram for a potential MOSA, or SOSA Approach, to Scenario 2. The first difference from the vertically-integrated approach in Figure 5 is the introduction of the Open Systems Architecture/Reference Architecture stage, which is owned neither by the Customer PO nor the integrator or vendors. The SOSA Consortium and SOSA Technical Standard are an example within this stage, where openness has three meanings:

- It is developing, demonstrating, and validating common data standards and Open Systems Architecture (OSA) "through a cooperative partnership with industry and academia". [24] [25]

[5] Recall that large-scale integrators emerged as able to achieve three integrability intents as a competitive advantage through product lines of reusable components in new or existing systems.

- The contents are readily available, often “equally available to the public [Distro A or Distro D] at no cost or with a moderate license fee”, [26] and
- The contents mandate the Form, Fit, Function (FFF) of specific key interfaces, and therefore, “the Government shall have unlimited rights in [FFF] data delivered”. [27]

For our sensor system, the systems integrator is now responsible for developing an instance architecture of the SOSA Reference Architecture (RA) to be used during the design, which is now in the Instance Architecture/Design stage. However, it could also conceivably be owned by customer PO, who as we shall see can start to take on some of the roles previously afforded to large-scale integrators with the further evolution of MOSA.

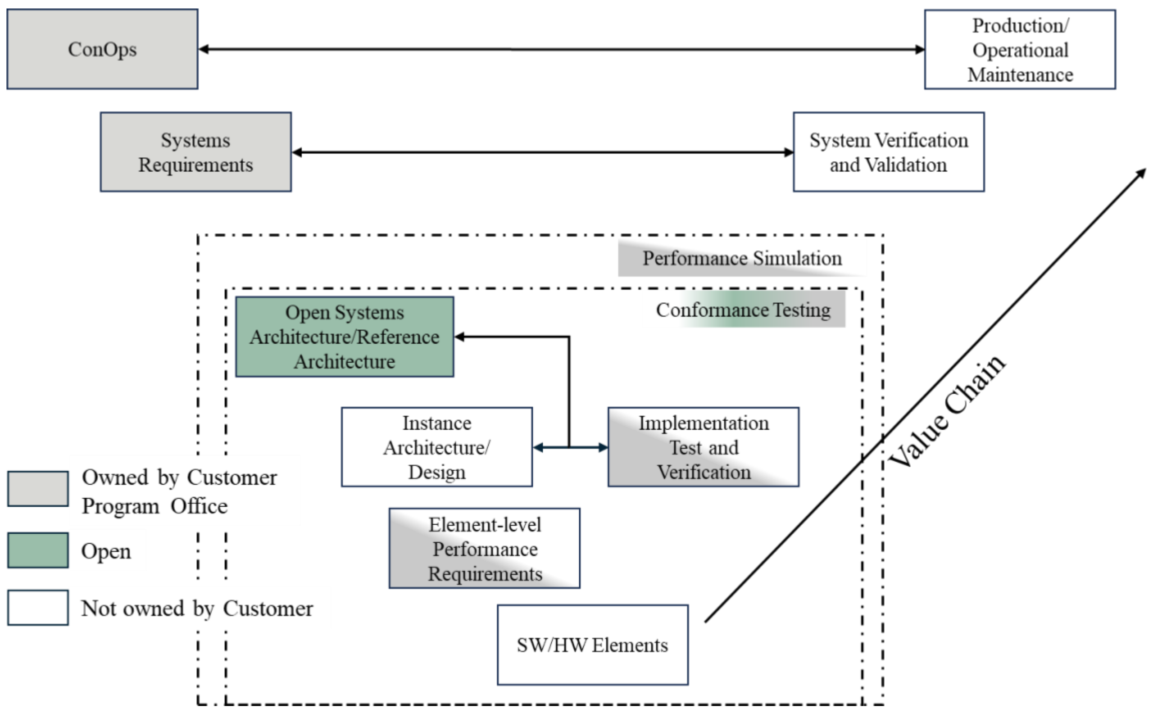



Figure 6: Proposed Systems V Diagram for non-Vertically Integrated (MOSA) Systems with Performance Gap

The next difference is the introduction of the Element-level Performance Requirements stage, which again can be owned by the Customer PO or not. In our vertically integrated example above for Scenario 2, it was owned by the large-scale integrator and applied a specific set of performance requirements to SW/HW elements. This exists independently of the OSA requirements tested during the Conformance Testing to the OSA, because the best performing SW/HW elements will still be produced by vendors in industry, who continue to bring new technology in the form of IP to the warfighter. This is a key business capability that is likely to persist long-term through the continued evolution of MOSA. On the other hand, if for instance architecture is owned by the customer PO, then it follows that the element-level performance requirements and decomposition would also be owned by the customer PO throughout the remaining phases of the system lifecycle.

For the SW/HW Elements stage at the bottom of Figure 6, the OSA specifies the key interfaces of the system and provides functional descriptions of the modules. This is the FFF of the modules and the architecture. The modules contain vendor IP and, when privately funded during development (e.g., using IRAD), the IP is often provided with restricted rights. For vertically-integrated systems, the integrator achieves competitive business capabilities through the cost savings associated with reuse of our resources (differentiators) in the SW/HW Elements stage. However, for MOSA, the intent is for the customer to realize those cost savings by shaping the MOSA acquisition ecosystem so that the vendor will protect their company-funded IP within the module boundaries specified by the OSA. Thus, the key business capability is maintained for the creation of SW/HW elements, whether the same vendor is also the integrator or not, but more cost savings flow back to the customer. However, the competitive bids will include cost structures based on the capability to develop and integrate SW/HW elements, versus when this capability is coupled with the business capability of controlling of the design and decomposition of the system performance requirements discussed above.

For vendors, competitiveness is gained as common products and families are designed to quickly meet conformance across the many required OSAs across a customer's or large-scale prime's portfolio, in addition to servicing the two additional integrability distances from above. Conformance means transitioning the proprietary SW/HW elements into the functions of the OSA-defined SW/HW elements or modules, as well as conforming to the interfaces at the boundaries. Vendors who most quickly and efficiently making this transition increase their competitiveness, because the cost of future contracts is now reduced by the reusable, common products that are now OSA conformant. However, there is an initial sunk cost for this transition, as well as maintenance costs as the OSAs evolve.

Contract value comes from the labor needed in the Implementation Test and Verification stage, and there is even more value at this stage than in the vertical integration paradigm for two reasons. First, as one stage becomes more modular, emergent opportunity exists in an adjacent stage [5]; in this instance, integration, the former bottleneck, is being replaced by modularity, where the modules are the new bottleneck. The systems integrator is held accountable to an OSA, reducing the control the vendor has on the system. Engineers work with less degrees of freedom yet produce results that meet a system-level performance gap. In systems that have typically used the exploitation of margin in or across components and subsystems to realize systems-level performance requirements, this new constraint of module bottlenecks creates more value in the labor hours required to integrate them into a functioning and performing system.

The second reason for increased value in the Implementation Test and Verification stage is the labor hours associated with ensuring the resources in the SW/HW Elements stage meet the requirements of the OSA in the Conformance Testing stage, represented on Figure 6 by the dotted box. While the Customer PO will still provide the system requirements, there are FFF verification requirements at the module level as specified by the OSA. For some OSAs, these tests can be easily included into FQTs. The Conformance Testing stage can be owned, all or in part, by the customer PO, the integrator, the supplier, or the collective "open" group that collaborates on the evolution of the OSA. This area of ownership, for the development of testing tools or deployment of the testing to individual SW/HW Elements or small subsystems is outside the scope of this paper and left to future work. However, it is clear in a development or upgrade contract, there is value in the Implementation Test and Verification stage, and that when not under contract, this increases the sunk costs of the common products in the deployed in the SW/HW Elements stage.

More pertinent to this discussion is the Performance Simulation stage, which is the ability to test the decomposition of system performance requirements on larger and larger subsystems. Separating performance simulation from the systems integrator is an emerging means to break vertical integration by allowing the engineering teams from multiple vendors to work together in a common environment to deploy their IP modules to meet system-level performance requirements. Examples include AFRL's Golden Hoard Colosseum for collaborative autonomy capabilities [28] or App Enabled Rapidly Reprogrammable EW/EMS

Systems (AERRES) [29]. The Performance Simulation stage sits high in the value chain due to the highly complex models and computing resources needed at larger and larger subsystem levels. This role has historically been part of the systems integrator, but the MOSA ecosystem has this as a potential new, independent, and lucrative business capability. The Performance Simulation stage is dependent on the Conformance Testing stage, because resources in the SW/HW Elements stage must easily fit and function together. However, the purpose of the Performance Simulation stage is to go one step further to address the performance of the elements, their combinations in subsystems, and finally at the systems level.

There are several customer benefits to government-funded, proprietary performance simulations throughout the MOSA product lifecycle. Many iterations of simulation and programs will give the customer PO enough knowledge to discover which elements in their system are critical and which modules can be expected to have a performance surplus margin throughout the production lifecycle. Here, we can test how integrability of a particular element, like SOSA Module 3.3 "Image Pre Processor" can effect upgradability of the overall sensor system, or whether perhaps SOSA Module 3.1 was a better choice. Another benefit for the customer is the "App Store." These are OSA-conformant modules that are listed on registry; for example, one of the more mature registries is the Future Airborne Capability Environment (FACE™) Registry, which houses many vendors products and services [30].

For SW elements, like signal processing algorithms and applications, the App Store will likely not disclose any performance data, for competitive reasons. It is likely to only act like a registry of OSA-conformance modules. However, the performance simulation will serve as the future plug-and-play fests for registry products (apps in the App Store) that exercise the performance of the system. The last customer benefit is that the performance simulation will be used during the acquisition, development, and production phases of a program. We noted above the benefits for the Customer PO during the acquisition phase, but the same performance simulation will be used during the development contract (including customer-owned design and requirements decomposition), as well as to facilitate future upgrades to the system. While the customer may own these simulations, an emerging business capability is the creation and maintenance of these highly-complex simulations, where just like common SW/HW elements that exist within large-scale integrators, there is also a pre-existing business capability of simulation.

In conclusion, Figure 7 shows how pre-existing business capabilities are potentially transformed by the continued evolution of MOSA. We showed that for systems with a performance gap, achieving MOSA is still possible, albeit more difficult due to lack of control on many variables throughout the design and subsequent verification of a system. This was demonstrated in our case study where SOSA Module 1.1 "Systems Manager" had a smaller integration distance than the two additional dimensions we discovered for SOSA Module 3.3 "Image Pre Processor".

## *SOSA® Module Integrability*

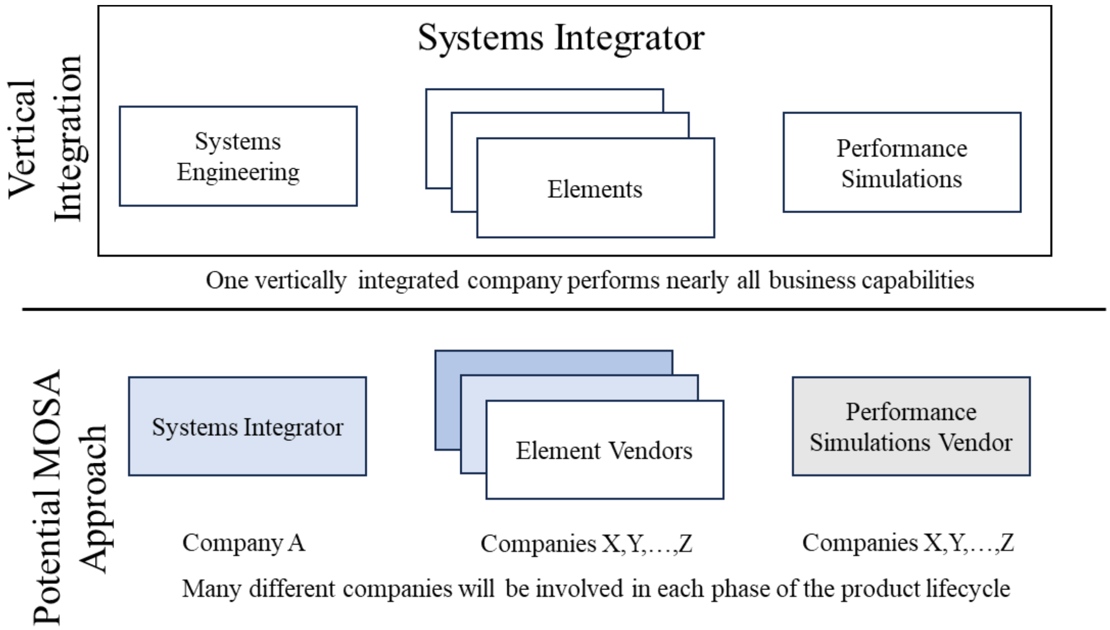



Figure 7: Potential Evolution of Business Capabilities with the MOSA Approach

# Referenced Documents

(Please note that the links below are good at the time of writing but cannot be guaranteed for the future.)

[1] H. M. Sapolsky, "Inventing Systems Integration," in *The Business of Systems Integration*, New York, Oxford Univerity Press, 2003, pp. 15-34.

[2] G. Dosi, M. Hobday, L. Marengo and A. Prencipe, "The Economics of Systems Integration: Towards an Evolutionary Interpretation," in *The Business of Systems Integration*, New York, Oxford University Press, 2003, pp. 95-113.

[3] H. Chesbrough, "Towards a Dynamics of Modularity: A Cyclical Model of Technical Advance," in *The Business of Systems Integration*, New York, Oxford University Press, 2003, pp. 174-200.

[4] C. Christensen, The Innovator's Dilemma, Boston: Harvard Business Review Press, 1997.

[5] C. M. Christensen and M. E. Raynor, The Innovator's Solution, Boston: Harvard Business School Publishing, 2003.

[6] P. Dombrowski and E. Gholz, Buying Military Transformation: Technological Innovation and the Defense Industry, New York: Columbia University Press, 2006.

[7] E. Gholz, "A business model for defense acquisition under the modular open systems approach," *Defense Ac quisition Review Journal,* pp. 216-233, 2007.

[8] G. Sanders, "Open for Business: Business Models for Innovation with Modular Open Systems Approaches," in *Excerpt from the Proceedings of the Nineteenth Annual Acquisition Research Symposium*, Monterey, 2022.

[9] S. Henry, B. Schuerer, J. Bradley and R. Ragyan, "MOSA Implementation Considerations, Information Needs and Metrics," National Defense Industrial Association, Systems Engineering Division, Architecture Committee, 2023.

[10] T. McDermott, "Systems Engineering Modernization: Does SE need to be modernized, and what might that look like?," Systems Engineering Research Center, 21 June 2022. [Online]. Available: https://sercuarc.org/systems-engineering-modernization/.

[11] The Open Group, *Technical Standard for SOSA™ Reference Architecture, Edition 2.0, Version 3,* 2025.

[12] L. Bass, P. Clements and R. Kazman, "Integrability," in *Software Architecture in Practice , Fourth Edition*, Boston, Pearson Addison-Wesley, 2022, pp. 101-115.

[13] R. Kazman, P. Bianco, J. Ivers and J. Klein, "Integrability," Software Engineering Institute, Technical Report CMU/SEI-2020-TR-001, 2020.

[14] A. Salado, "Interpreting integrability," *Systems Engineering,* vol. 27, no. 1, pp. 99-108, 2024.

[15] S. Cohen, A. Glunt, J. Hudak, J. McGregor, G. Moreno and F. Schenker, "A Strategy for Component Product Lines: Report 1: Scoping, Objectives, and Rationale," SPECIAL REPORT, CMU/SEI-2022-SR-002, Pittsburgh, 2022.

[16] S. Simi, E. LeBouthillier and K. Erickson, "IMPACT in Action: Conducting Multiple FACE™ Development efforts aligned MBOSA," in *The Open Group FACE™ and SOSA™ TIM*, 2022.

[17] S. Simi, J. Cain, J. Tokar and A. Stuessy, "A Harmonization of MOSA OSAs One Weapon System and the Use of SOSA™, WOSA, and FACE® Technical Standards Introduction to a Harmonization of MOSA OSAs," in *The Open Group FACE® and SOSA™ Consortia 2024 Technical Interchange Meeting (TIM)*, 2024.

[18] B. Clark and J. Cain, "The SOSA EO/IR Mission Thread: Increasing Velocity through Standards Harmonization," in *The Open Group FACE™ and SOSA™ US Air Force Technical Interchange Meeting (TIM)*, 2022.

[19] J. Cain, A. Stuessy, J. Tokar and S. Simi, "Signal processing applications of the Sensor Open Systems Architecture for missile seekers," in *Proc. SPIE 13048, Radar Sensor Technology XXVIII, 1304808*, National Harbor, 2024.

[20] The Open Group, *SOSA™ Reference Implementaiton Guide,* 2023.

[21] E. Gholz, "Systems Integration in the US Defence Industry: Who Does It and Why Is It Important?," in *The Business of Systems Integration*, New York, Oxford University Press, 2003, pp. 279-305.

[22] E. Honour, "6.2. 3 Understanding the value of systems engineering," in *INCOSE international symposium, Vol 14, No 1*, 2004.

[23] SEBoK Editorial Board, "Economic Value of Systems Engineering," The Guide to the Systems Engineering Body of Knowledge (SEBoK), v. 2.12, N, [Online]. Available: https://sebokwiki.org/wiki/Economic_Value_of_Systems_Engineering. [Accessed 2025].

[24] R. Spencer, M. Esper and H. Wilson, *MEMORANDUM FOR SERVICE ACQU ISITION EXECUTIVES AND PROGRAM EXECUTIVE OFFICERS: Modular Open Systems Approaches for our Weapon Systems is a Warfighting Imparitive,* Washington, DC, 2019.

[25] C. Del Toro, W. Chirstine and F. Kednall III, *MEMORANDUM FOR SERVICE ACQUISITION EXECUTIVES AND PROGRAM EXECUTIVE OFFICERS: odular Open Systems Approach for Department of Defense Weapon Systems,* Washington, DC, 2024.

[26] "Glossary Term: Open Standards," Defense Acquisition University, [Online]. Available: https://www.dau.edu/glossary/open-standards. [Accessed 2025].

[27] "Federal Acquisition Requirements (FAR) 52.227-14 (b) (1) Rights in Data—General," [Online]. Available: https://www.law.cornell.edu/cfr/text/48/52.227-14#:~:text=%281%29%20Except%20as%20provided%20in%20paragraph%20%28c%29%20of. [Accessed 2025].

[28] "Golden Hoard Colosseum," Air Force Research Laboratory, [Online]. Available: https://afresearchlab.com/technology/vanguards/successstories/golden-horde. [Accessed 2025].

[29] "App Enabled Rapidly Reprogrammable EW/EMS Systems (AERRES)," Air Force Research Laboratory, [Online]. Available: https://afresearchlab.com/technology/aerres/. [Accessed 2025].

[30] "FACE Registry," Future Airborne Capability Environment (FACE), [Online]. Available: https://www.facesoftware.org/registry. [Accessed 2025].

[31] D. Barrett, *Development of the Architecture Cost Effective Framework and Application to Open System Architecture,* Dayton: Air Force Institute of Technology, 2017.

[32] Government Accountability Office, "Weapon Systems Acquisition: DOD Needs Better Planning to Attain Benefits of Modular Open Systems," 2025.

# About the Author

Jonathan Cain is an accomplished engineer, innovator, and business leader at RTX. At Raytheon, an RTX business, his various roles have included Integrated Product Team Lead, Cross Product Team Lead, Principal Investigator, Section Lead, and Staff Technologist. He is a Level 5 Raytheon Certified Program Manager, an RTX Certified Architect, and he is Raytheon's 2025 Innovator of the Year. Jonathan is the co-lead of the Raytheon Modular Open Systems Approach (MOSA) Center of Excellence and the RTX Open Systems Architecture Community of Practice, where he oversees budget, strategy, personnel, and advocacy for MOSA at Raytheon and RTX. This includes support of all external OSA/GRA standards development, proposals and capture management, program and engineering execution, portfolio financial analysis, and business and technical education programs. Prior to assuming these leadership roles, he was a technical Subject Matter Expert for SOSA® and the Weapons Open Systems Architecture (WOSA). Jonathan is a Chartered Financial Consultant, certified in The Open Group Architecture Framework (TOGAF®), and a Certified Enterprise Architect Associate from Zachman International. He is an active member, author at SPIE and a session chair at the annual Defense and Commercial Sensing. Jonathan has a BS in math and statistics from the University of Arizona.

## About The Open Group FACE® Consortium

The Open Group Future Airborne Capability Environment® Consortium (the FACE® Consortium), was formed as a government and industry partnership to define an open avionics environment for all military airborne platform types. Today, it is an aviation-focused professional group made up of industry suppliers, customers, academia, and users. The FACE Consortium provides a vendor-neutral forum for industry and government to work together to develop and consolidate the open standards, best practices, guidance documents, and business strategy necessary for acquisition of affordable software systems that promote innovation and rapid integration of portable capabilities across global defense programs.

Further information on the FACE Consortium is available at www.opengroup.org/face.

## About The Open Group SOSA® Consortium

The Open Group SOSA® Consortium enables government and industry to collaboratively develop open standards and best practices to enable, enhance, and accelerate the deployment of affordable, capable, interoperable sensor systems. The SOSA Consortium is creating open system reference architectures applicable to military and commercial sensor systems and a business model that balances stakeholder interests. The architectures employ modular design and use widely supported, consensus-based, nonproprietary standards for key interfaces.

Further information on the SOSA Consortium is available at www.opengroup.org/sosa.

## About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards and open source initiatives by fostering a culture of collaboration, inclusivity, and mutual respect among our diverse group of 900+ memberships. Our membership includes customers, systems and solutions suppliers, tool vendors, integrators, academics, and consultants across multiple industries.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

Further information on The Open Group is available at www.opengroup.org.