SCA / SOSA® Interoperability Concept of Operations (CONOPS)

Document WINNF-TR-5011

Version 1.0.0

19 September 2025

TERMS, CONDITIONS & NOTICES

This document has been prepared by the Software Defined Systems (SDS) Tactical Communications Standards Project Team to assist The Software Defined Radio Forum Inc. (or its successors or assigns, hereafter "the Forum"). It may be amended or withdrawn at a later time and it is not binding on any member of the Forum or of the Tactical Communications Standards Project Team.

Contributors to this document that have submitted copyrighted materials (the Submission) to the Forum for use in this document retain copyright ownership of their original work, while at the same time granting the Forum a non-exclusive, irrevocable, worldwide, perpetual, royalty-free license under the Submitter's copyrights in the Submission to reproduce, distribute, publish, display, perform, and create derivative works of the Submission based on that original work for the purpose of developing this document under the Forum's own copyright.

Permission is granted to the Forum's participants to copy any portion of this document for legitimate purposes of the Forum. Copying for monetary gain or for other non-Forum related purposes is prohibited.

THIS DOCUMENT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY USE OF THIS SPECIFICATION SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE FORUM, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER, DIRECTLY OR INDIRECTLY, ARISING FROM THE USE OF THIS DOCUMENT.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the specification set forth in this document, and to provide supporting documentation.

This document was developed following the Forum's policy on restricted or controlled information (Policy 009) to ensure that that the document can be shared openly with other member organizations around the world. Additional Information on this policy can be found here: http://www.wirelessinnovation.org/page/Policies_and_Procedures

Although this document contains no restricted or controlled information, the specific implementation of concepts contain herein may be controlled under the laws of the country of origin for that implementation. Readers are encouraged, therefore, to consult with a cognizant authority prior to any further development.

Table of Contents

List of Figures

List of Tables

Contributors

The following individuals and their organization of affiliation are credited as Contributors to development of the specification, for having been involved in the work group that developed the draft then approved by WInnForum member organizations:

- Steve Bernier, Viavi Solutions
- Michael Denton, L3Harris
- David Hagood, Cynosure (Guest),
- David Murotake, HKE,
- Randy Navarro. Leidos (Guest)
- Kevin Richardson, MITRE,
- Robert Sklut, JTNC.

Document history

Version	Date	Contents
1.0	19 September 2025	Initial release

SCA / SOSA(TM) Interoperability Concept of Operations (CONOPS)

1 Introduction

This document WINNF-TR-5011-V1.1.0 provides the technical description of the Concept of Operations for the functional operation of an Software Communications Architecture (SCA) compliant waveform application within an ecosystem defined and designed in accordance with the Sensor Open Systems Architecture (SOSA®) Technical Standard [3]. Note that for the scope of this specific document, every use of the term "SCA" is targeted at CORBA OE applications, however, an SCA 4.x implementation not using a CORBA OE can be supported using this same architecture pattern using interfaces and adapters based upon whatever OE is used.

1.1 SCA Overview

1.1.1 Background

The objective of the JTRS program was to consolidate service programs into an interoperable, joint program for the development and acquisition of affordable, high-capacity tactical radios to meet the bandwidth needs of various echelons. A high-level view of a deployed SCA system is provided in Figure 1. The premise behind the objective was that the single function hardware design of legacy

communications systems could not take advantage of rapid changes in commercial technology and thus provide the functionality and flexibility necessary to achieve the following:

- maintain information superiority
- support the rapid mobility required by today's armed forces

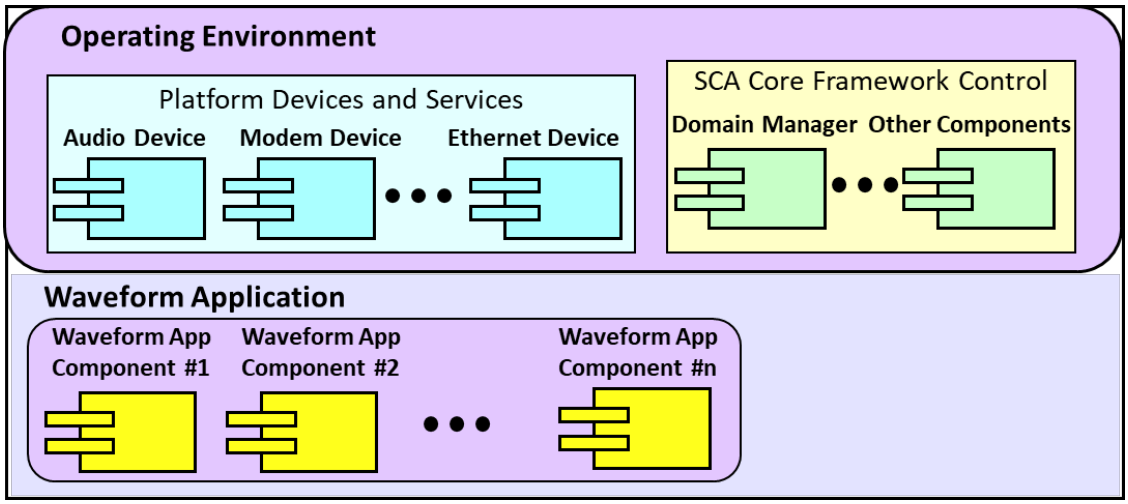

Figure 1 – SCA System Composition

1.1.2 Architectural Objective

The proposed solution to be realized by the JTRS program was to introduce a software-programmable and hardware-configurable digital radio system that would provide increased interoperability, flexibility, and adaptability in support of the varied mission requirements of DoD warfighters and other stakeholder communities.

1.2 SOSA® Background

1.2.1 Purpose

The goal of The Open Group SOSA Consortium is to develop open architecture at the right level for Communications (Comms), Electro-Optical/Infra-Red (EO/IR), Electronic Warfare (EW), Radar, Signals Intelligence (SIGINT), and Directed Energy Weapon Systems (DEWS) sensor systems. The architecture should be flexible enough to support multiple sensor modalities across multiple domains, i.e., airborne, subsurface, surface, ground, and space. The SOSA Consortium strives to develop an ecosystem that allows interoperability, reuse, and faster delivery of products to market through vertical integration from cables, mechanical interfaces, hardware, software, and system designs.

The SOSA initiative addresses the challenges of rapid, affordable capability evolution for today's military community. A component of the SOSA approach is the development of an Open Systems Architecture (OSA), that provides the necessary abstractions, technical specifications, and interfaces to provide an implementation agnostic model that addresses stakeholder requirements for sensor systems to be acquired by the disparate communities. The SOSA Technical Standard is designed to

promote software, hardware, and interface portability and encourage the development of modular products that can be leveraged and reused across the Communications, EO/IR, EW, Radar, SIGINT, and DARE (Data at Rest) communities.

SOSA has been adopted as a key component of the DoD OSA strategy, as evidenced by its inclusion in the DoD tri-service memo for modular open systems [1] . The memo states that Modular Open Systems (MOSA) standards should be included in all requirements, programming and development activities for future weapon system modifications and new development programs to the maximum extent possible. It also cites the SOSA Consortium as a developer of representative standards.

1.3 Project Overview

1.3.1 Objective

The WInnF Tactical Communications Standards (TCS) project group proposed an activity to document the interoperability CONOPS for using waveform applications defined in accordance with the Joint Tactical Networking Center (JTNC) SCA standards within a host environment consisting of modules aligned with the SOSA technical standard. The goal of this effort was to develop a high-level SCA ←→ SOSA interoperability strategy accompanied by a set of artifacts that could be used to translate the vision to the level of detail required to produce a viable design and implementation. This CONOPS only provides a suggestion of the interactions that would need to occur to realize an SCA-based Waveform Application within a SOSA-based platform. System developers should be able to customize the provided interactions to develop a communications capability for their own use.

1.3.2 Approach

The intent of this work is to identify and describe the expectations, interactions, and constraints associated with high-level description provided by the SCA ←→ SOSA Interoperability Approach Brief [14]. The document will describe a strategy that allows existing, or new, SCA-based waveform applications to be utilized within a sensor platform composed of SOSA modules and infrastructure elements with limited modifications required within the waveform application implementation.

The TCS team will identify the SOSA modules, protocols, and requirements that are essential for interoperability between the SCA waveform application and SOSA-based platforms and then highlight the “cross platform” interactions that are necessary for the proposed target architecture to provide communications functionality within a system populated by SOSA modules.

1.3.3 Scope

This paper will highlight and describe the interactions that will need to occur for the waveform application to function appropriately within the SOSA ecosystem. It is not the intent of this paper to be a low-level primer for SCA, SOSA, or the design or implementation of specific waveform applications. To gain that type of knowledge, it is recommended that the reader reference the standards themselves [3][4]. However, the descriptions provided should be at a level such that they can be mapped to a high-level solution architecture that describes how the individual components should be integrated to provide a communications sensor. To achieve this goal, this CONOPS will

identify key scenarios that should establish a pattern that illustrates how the individual pieces should work together to achieve typical Software Defined Radio (SDR) functionality, which can be extended by the reader to provide a full, production quality implementation. Within the scenarios, we will attempt to call out any requirements, constraints, or simplifying assumptions that may impact the interoperability approach or need to be resolved by the system developer.

2 SCA ←→ SOSA® Interoperability Approach

The objective of the SCA ←→ SOSA Interoperability approach is to provide a pathway for SCA-based waveform applications to be used, in an open architecture conformant method, with a SOSA aligned environment. Although the focus for this effort is on SCA waveform applications, the same approach could be applied to applications based on other frameworks or architecture models in a straightforward manner.

This approach uses Open Container Initiative (OCI) container technology to encapsulate SCA waveform applications within SOSA Ecosystem Elements, each containing a CORBA ORB and enough of the SCA operating environment to allow the contents to operate within its own SCA context. The structure of the container is illustrated in Figure 2. The presence of the Core Framework, middleware, and Operating Environment adapters enables the container to function as a native SCA ecosystem that allows existing SCA-based applications, services, and devices to operate with limited source code changes. The approach results in a flexible ecosystem that supports both SOSA and SCA devices and services.

Conceptually speaking, each SCA container could be identified as a unique SOSA Module in the context of the SOSA System Management Infrastructure. This container would provide its own functionality and, with the assistance of an adaptation layer, implement the responsibilities common to all SOSA Modules.

This interoperability approach provides a strategy to accelerate implementation and delivery of an OSA communications capability within SOSA, and it also expands the breadth of providers that could readily develop and deliver waveform applications for SOSA aligned systems.

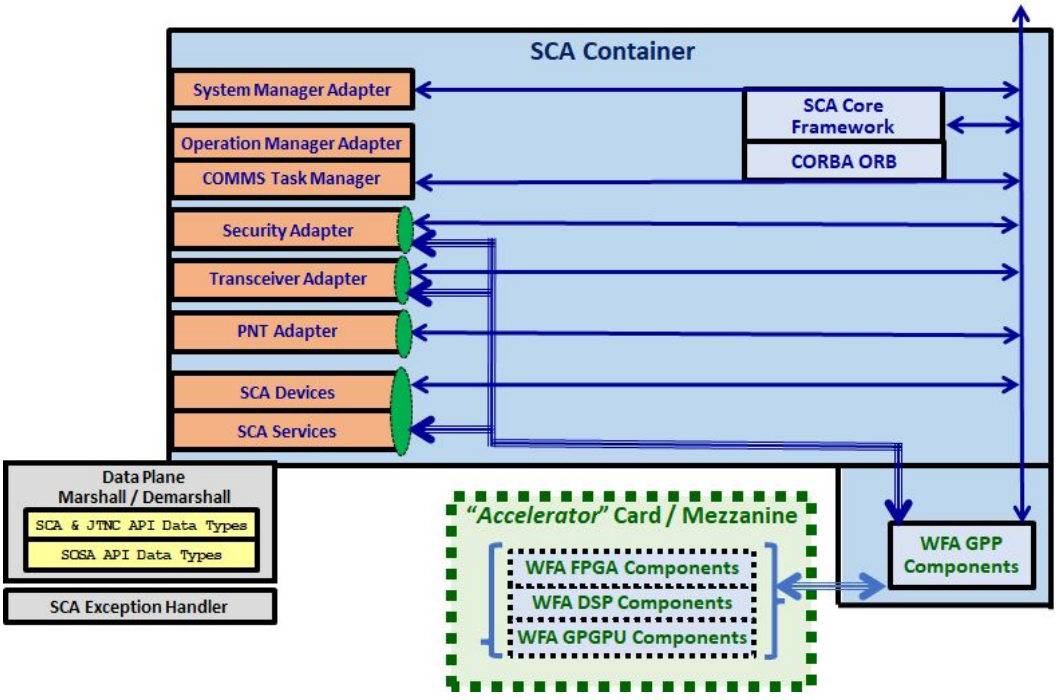

Figure 2 – Overview of SCA Container

2.1 Assumptions and Principles

2.1.1 Definitions

The following definitions constrain the SCA/SOSA interoperability approach that will be described within this paper.

Waveform – An electromagnetic or acoustic signal-in-space, typically defined by Open Systems Interconnection (OSI) model layers 1 through 3, along with the controls and processes for how the waveform is generated or received in support of a desired function or application. These processes do not include the message content.

Waveform Application - The interactions between a client (human or technology) and the RF / Optical Medium that represent functionality that is completely independent of the radio platform or hosting system that supports the waveform applications. The waveform application defines the detailed over the air encoding / signaling formats, and the nature and substance of the interactions with the communications client. The waveform application includes all the controls and processes needed to satisfy its functionality and those needed to support the waveform that it generates or receives.

Container – A unit of software delivery. A Container defines a set of standard operations for functions to create, stop, or start the container. The goal of a Container is to encapsulate a software component and all its dependencies in a format that is self-describing and portable, so that any compliant runtime can run it without extra dependencies, regardless of the underlying machine and the contents of the container [5]. Consequently, an SCA Container provides a native SCA Ecosystem, that allows existing SCA-based waveform applications, services, or devices to operate with no or few source code changes.

SCA Waveform – A waveform application that is implemented to provide its native, defined functionality integrated with the SCA interfaces and requirements.

SOSA Module Agent (SMA)– The software component that provides the connectivity between the SCA components within the SCA container and the SOSA ecosystem outside of the container. The Module Agent provides the requisite mappings to facilitate communication between both SCA and SOSA elements. Specifically, the SMA presents a SOSA aligned façade to collaborating actors outside of the SCA container. Once the Module Agent receives the external message or requests it, translates it into a format that can be used within the container. For method invocations that originate within the container and are directed to SOSA modules, the SMA maps those calls into SOSA Data Messages that are routed to the designated endpoint.

Control plane data bus connections – The location that connects the control path of the SOSA module containing the SCA container with the remainder of modules in the SOSA ecosystem. The control path typically consists of relatively short and infrequent interactions. Control messages may be exchanged across the SOSA Message Interconnect (SMI) or the SOSA Wideband Low Latency Interconnect (SWLLI); however, there is an important distinction based on the message origin or destination. Control messages from the native SOSA ecosystem are transported on the bus aligned to their interface specification.

Data plane data bus connections – The location that connects the data path of the SOSA module containing the SCA container with the remainder of modules in the SOSA ecosystem. The SMA

contains interfaces that allow it to receive SDMs exchanged across the SMI or the SWLLI. The data path typically consists of relatively long and frequent interactions involving the transfer of opaque data between modules.

SOSA module – An architectural entity that has open, specified functional behaviors and interfaces, could be instantiated using hardware elements and/or software components and conforms to the complete definition (functionality, behavior, and interfaces) as defined in the SOSA Technical Standard [3]. The module that hosts the SCA solution is an emulated SOSA module that utilizes the SOSA Container Run-Time Environment profile.

2.1.2 Assumptions

The encapsulated SCA container provides a SOSA conformant Module profile to the remainder of the SOSA system of systems platform.

When the SCA-based Waveform Application uses native SCA Operating Environment (OE) components (i.e., Devices and Services), those interactions within the container use the SCA required transport mechanism for the selected implementation Platform Specific Model (e.g., CORBA calls).

When the SCA-based Waveform Application uses SOSA OE components, the communication protocol of at least one of the endpoints must be adapted so that they are able to interoperate. In accordance with the proposed interoperability approach, the role of the SMA is to provide those bridges. To enable the transformation, the SMA will incorporate proxy OE components/interfaces as plug-ins and those interfaces will be accessible by the other elements within the container in accordance with the SCA CORBA PSM.

When the SCA container is completely self-contained, the SCA Control Plane never crosses SOSA module boundary.

The proxy SOSA module provides a mapping to allow SOSA platform to SCA Waveform Application bulk data transfers. The SCA data plane never crosses the SOSA module boundary.

2.1.3 Reference Platform

The CONOPS document will use the VHF-UHF Standard (VULOS) waveform, STANAG 4204 / 4205, as its conceptual model (note: that we will not address any of the implementation level particulars of its implementation.

VULOS supports 22 narrow and wideband Line of Sight communication modes across the VHF (30-88 MHz) and UHF (225-512 MHz) bands. The waveform application is capable of data rates ranging from 2,400 to 56,000 bits per second (bps). It is a narrowband single-hop waveform application designed for short-distance voice or data communication. This WFA enables the selection of Amplitude Modulation (AM) or Frequency Modulation (FM), which may be configured on-the-fly, alongside the modulation index. The WFA's channel bandwidth is adjustable up to 25 kHz, with channel spacing adjustable up to 25 kHz. Lastly, VULOS can utilize digital and analog voice Coder-Decoders (CODECs) installed on the radio [5][6].

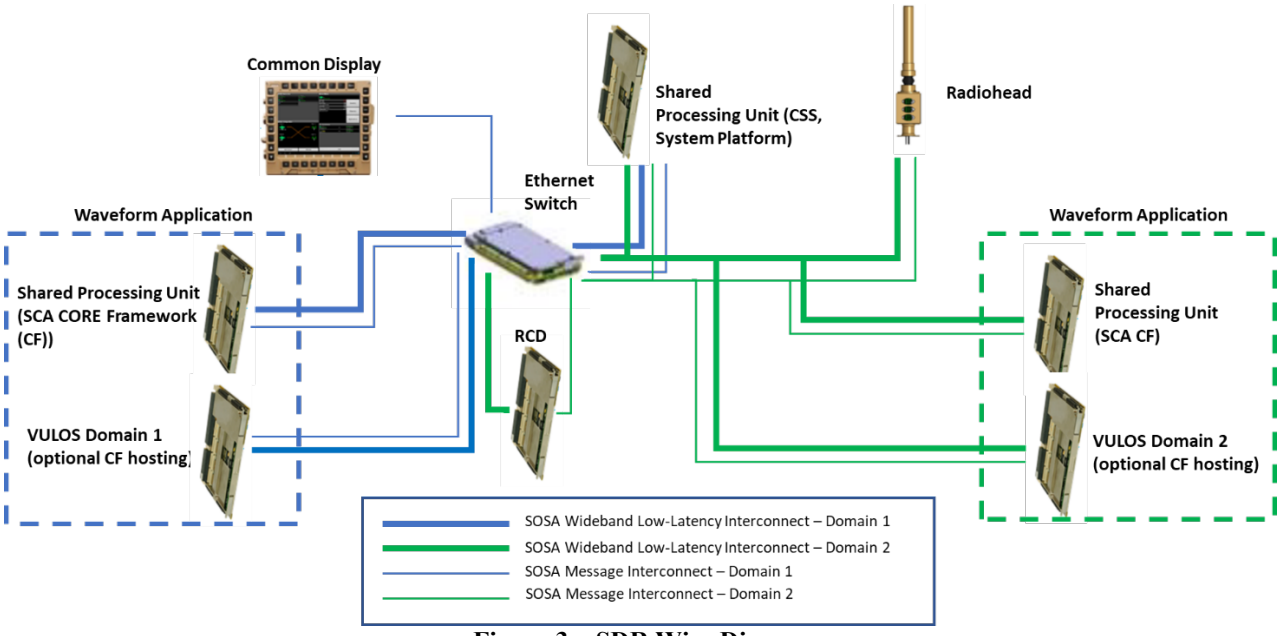

Figure 3 – SDR Wire Diagram

The high-level model of the SDR implementation is contained in Figure 3. The dashed lines represent the boundaries of the Waveform Application within the diagram. The following elements comprise the platform for the SDR system that we will discuss within this paper:

- Cryptographic Subsystem (Shared Processing Unit) – the set of hardware, software, and/or firmware that implements approved[1] security functions (including cryptographic algorithms and key generation) contained within the cryptographic boundary.
- Radiohead – A device that contains an antenna (i.e., element that captures or radiates electromagnetic energy in support of receive and/or transmit operations) and may contain additional elements that realize the RF chain. An RF chain is a cascade of electronic components and sub-units which may include amplifiers, filters, mixers, attenuators and detectors [7][8].
- Ethernet Switch – Manages the flow of data through a network, sending data it receives in one port to another port based on information in a data packet's header. Functionally, a switch will receive messages, queue the data, and transmit the messages to the appropriate subset of its ports. Switches typically have options relative to how, when, and which data is queued, dropped, and transmitted to which ports are controlled by the addressing, routing, and QoS protocols being implemented [9].
- MORA RF Conditioning and Distribution (RCD) – Device that contains resources which support system receive and or transmit capabilities. The presence of the RCD highlights an architectural alternative that exists within these types of systems as it provides an additional candidate location for the RF chain functionality.

[1] FIPS PUB 140-2 SECURITY REQUIREMENTS FOR CRYPTOGRAPHIC MODULES

- Common Display – Device used to provide a human machine interface (HMI) between the SDR platform and its human user. The interface may be leveraged to facilitate interactions between users and WFAs, for purposes such as installation, maintenance, control, or operational use.
- Waveform Application (WFA) - the implementation of the waveform application consists of multiple components, which are deployed across two information domains and are separated by a Cryptographic Subsystem.

This diagram provides a notional architecture of the elements / components that will need to exist as part of our target SDR; however, they are required to construct an operational system. This document exists as a primer for the integration of OSA Communications and Platform capabilities. Standards such as the SOSA Technical Standard, the Software Communications Architecture and STANAG 4204 / 4205 provide much more extensive and authoritative information regarding what is necessary to design and implement a modular, production communications system.

2.1.4 Detailed Platform Architecture

2.1.4.1 Overview

The intent behind the SCA ←→ SOSA Interoperability approach was to establish a model and pattern that allows the community to continue to evolve and explore new form factors and technical strategies while ensuring that as we move forward, we do not lose the benefits of the tactical communications investments made and technical discoveries achieved over the last couple of decades. The working group identified the work of the SOSA Consortium as a target because of the notoriety that it has gained within DoD [1] the breadth of sensor types that it is targeted towards, and the focus that the consortium has had in alignment with MOSA principles. In addition, the group believed that it was important to pursue this work in a forum where the findings and recommendations would be publicly accessible and open to a wider audience so that it can be matured and refined quicker and more broadly.

The basis of the architecture of the interoperability approach took the following principles and objectives into account:

- Maximize WFA portability
- Minimize changes to the existing SCA-based WFAs
- Provide a mechanism to allow tactical communications to be provided using an OSA conformant approach based on the SOSA Technical Standard
- Leverage the lessons learned in previous tactical communications initiatives
- Develop a flexible, extensible approach

The group coalesced on the design provided in Figure 4 which architecturally follows the principles of the adapter pattern [10].

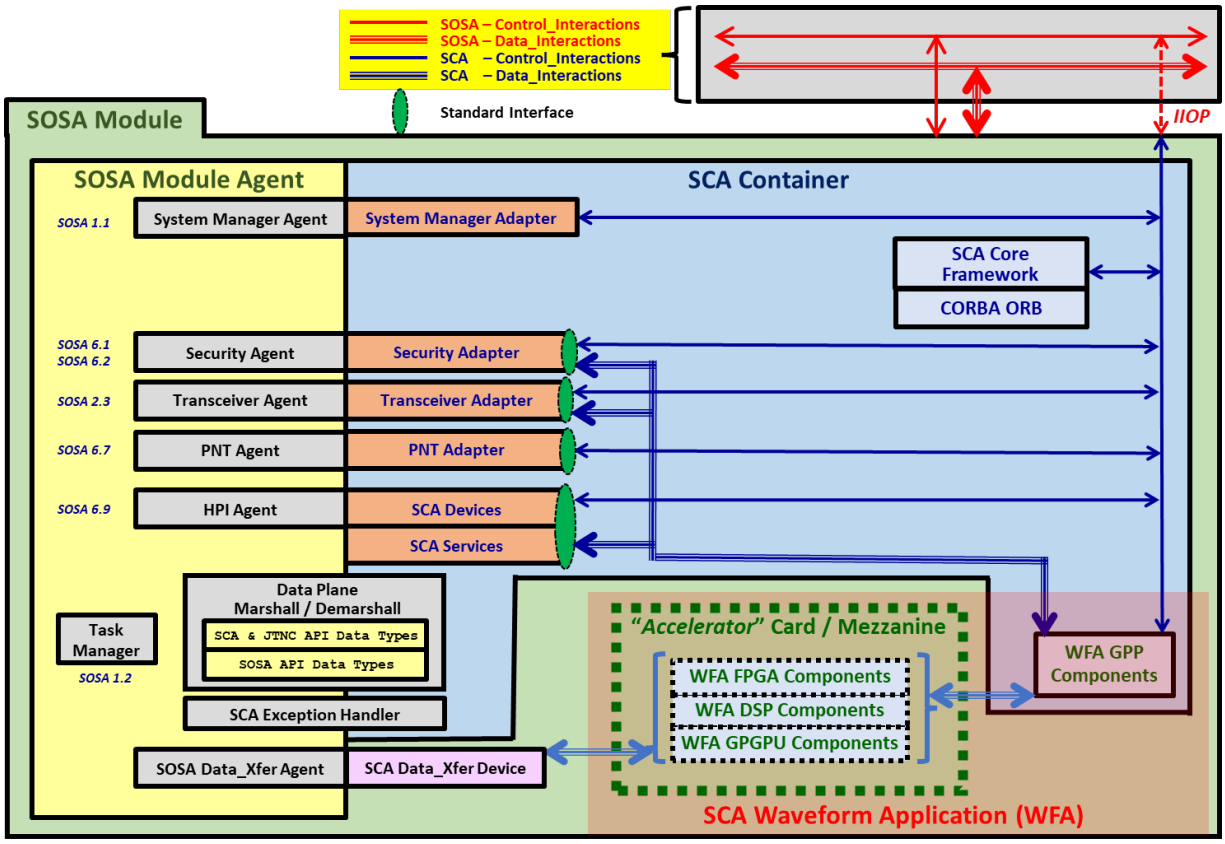

Figure 4 – Container to SMA Interactions

The SOSA Technical Standard decomposes sensor functionality into several lower-level procurable entities identified as modules. The modules provide general services, such as system management, or domain specific functions such as image processing. When taken as a whole, the collection of modules supplies the services that are necessary for the SDR to operate like a radio. The IP required to process the waveform is encapsulated within the WFA. The core idea of the approach is to keep that implementation intact to the greatest extent possible and have the WFA interact with the corresponding infrastructure elements provided by SOSA rather than those provided by the SCA, hence the adaptation layer. This is possible because the SCA components or SOSA modules both provide abstractions that act as intermediaries between the WFA and the resources, physical or logical, that provide the actual service.

2.1.4.1.1 Interoperability Approach Support for Portability

This architecture maximizes WFA portability in two methods. First, it preserves the SCA Application abstraction as the representation and composition of the implementation of the communications capability. An SCA Application interacts with its external collaborators via ports, which are accessed via APIs. The APIs are the constructs that allow waveform application functionality to be kept separate from the infrastructure. Secondly, the architecture uses the SOSA Container RTE profile to host the WFA. Utilizing containers makes the WFAs more "change resistant" because in addition to

the WFA implementation the container also satisfies the necessary dependencies that enable the WFA to operate, thus minimizing any dependencies on the external environment.

2.1.4.1.2 Interoperability Approach Support for SOSA

A target of the architecture is for it to be consistent with the requirements and principles of the SOSA Technical Standard. A key aspect of the project was to reuse the existing SCA-based WFAs, but the target for the application was to "gravitate towards" SOSA. Consequently, the resulting approach needed to include an environment that allowed the WFA to function and also identify a method for the WFA to be integrated with the other SOSA modules in the target system implementation. Containers were a natural fit for the platform operation because of the reasons highlighted in the earlier section. This architecture introduces a concept identified as the SOSA Module Agent (SMA) to facilitate the WFA interoperability. The SMA is the component that encapsulates the WFA and allows entities (human or non-person) to access the application functionality. The SMA appears as a SOSA module to the other system entities. It connects the system buses and transmits and receives SDMs. The agent is responsible for receiving incoming requests, and either directly or using a separate adapter, translate them to their native SCA data exchange format, and directing the requests to the appropriate component endpoint. Similarly, the adapter receives outgoing messages, translates them to an appropriate SOSA format and transmits them to the target module.

2.1.4.1.3 Interoperability Approach Support for Minimizing WFA Change

The presence of the adapters minimizes the need for the WFA implementation to change when it is migrated to this environment. The modular aspect of SCA drives the establishment of partitions between WFA and Platform functionality. The WFA cooperates with Devices and Services aligned with the JTNC APIs [11] and, management and deployment capabilities provided by the SCA Core Framework (CF) [4]. The architecture limits WFA changes by preserving its abilities to invoke the same interfaces using the same technology binding as it did in its original implementation. The cost of this is not free though, the architecture requires that intermediaries are included within the container that provide/use the same interface as its original implementation. The intermediaries are paired with a corresponding adapter (i.e., subagent) which has a direct line of communication with the corresponding SOSA module or provided capability. To support implementation flexibility, the intermediaries can forward the call to the SOSA infrastructure or provide a local implementation. Note, that if the local implementation approach is selected, the functionality must be provided within the container, and the developed capability won't necessarily be accessible by entities outside of the container.

2.1.4.1.4 Interoperability Approach Support for Lessons Learned

This architecture accommodates several features that have been learned from prior communications system implementations. Two significant inclusions are related performance and multi-domain operations. The design allows for multi-domain operations by not constraining the cardinality of the containers or the connections to the buses. WFAs are often segmented into multiple domains, perhaps separated by a system cryptographic capability. The approach allows for 1…N containers to exist within a single Plug in Card (PIC). This partitioning along with the container principles allows for a software enforced separation of the different portions of the WFA. If need be, the containers can be connected to different system buses if the WFA separation needs to be extended to different parts of the system. From a performance perspective, the design allows for portions of the WFA to be

deployed to differing processing environments. The command-and-control portion of a WFA usually performs operations that require little input or output data, do not involve significant computing complexity, and do not contain stringent real-time constraints so that functionality can be implemented on a general-purpose processor. However, the signal processing portion of sophisticated WFAs often needs to perform complex calculations within highly constrained time frames, so they require specialized processors such as FPGAs or GPUs to perform those calculations. The design enables access to mezzanine or accelerator cards within the bounds of the "module" to provide the necessary high-performance environment.

2.1.4.1.5 Interoperability Approach Support for Flexibility and Extensibility

There are two characteristics of the interoperability approach that reinforce the extensible nature of the design. First, the approach is based on a well-known, proven pattern and not a "one off" or specialized solution. When the SOSA Module Agents are evaluated, one will discover that there is nothing that is SCA specific within the bounds of the agents. Any of the SCA specific dependencies or constructs exist behind the adapter within the scope of the WFA implementation. As a result, this approach can be thought of as a pattern that can be applied to other communications applications or general sensor implementations. Secondly, the initial architecture has been specified at a coarse-grained level that can be easily extended or specialized. For example, if a system designer wanted to access an additional system wide capability, it would be a straightforward process to add an additional sub-agent within the SMA that is directed towards that function and develop it to be an additional connection point between the system and WFA.

2.1.4.2 Sample Data Flow

The following diagram in Figure 5 attempts to explain the data flows into and out of the Comms module running within the SCA container. For a TX voice operation, we start at the top with the Audio Data coming in over the SMI bus to an Audio Agent in the unencrypted domain (Domain 1). Depending on the voice format, this may very well be a networking agent for SIP packets rather than one specifically for voice but the point is the Comms module will receive voice data on the SMI bus. The Audio Agent then passes this data to the Audio Device inside the SCA container. The mechanism for doing so is unspecified and will be implementation dependent. The Audio Device then uses an SCA defined interface to pass data into the Waveform Application.

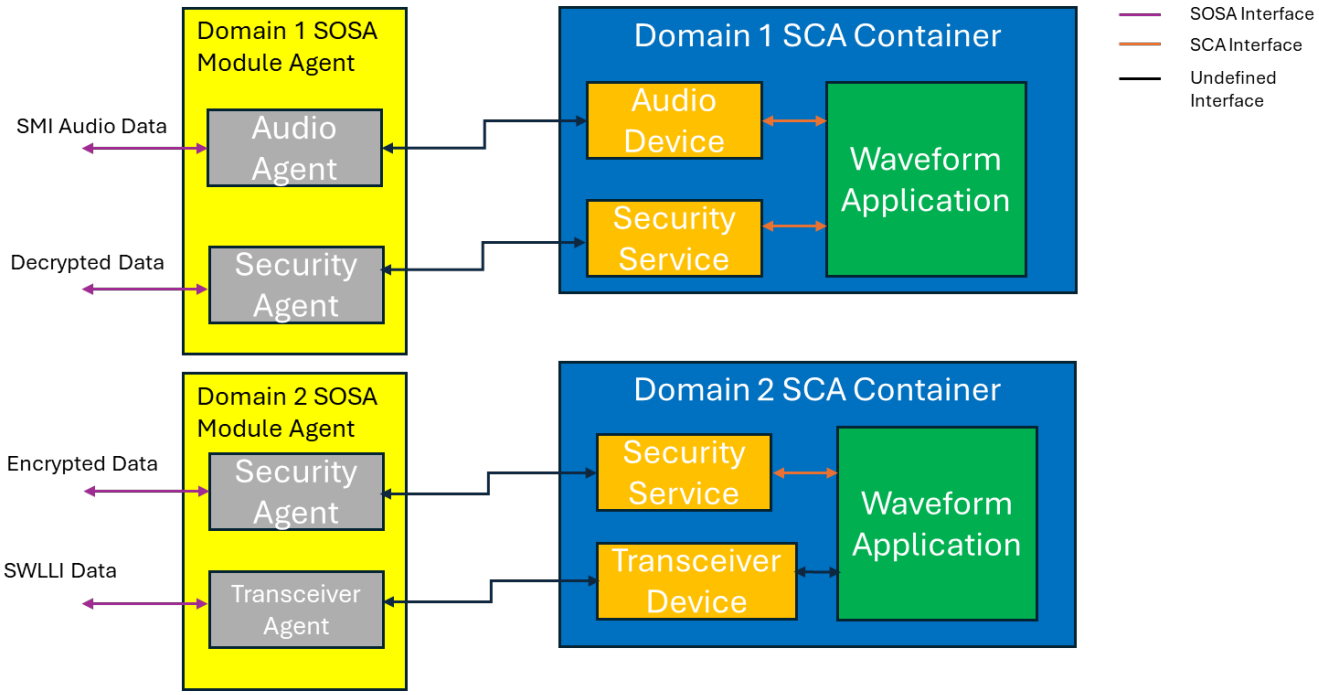

Figure 5 –Audio Data Flow

The Waveform Application at this point needs to perform whatever voice processing is necessary for the audio data and then encrypt it. Encryption is performed by using an SCA interface to the SCA Security Service within the container. The Security Service then passes the data over an unspecified interface to the SMA Security agent, which will use a defined SOSA interface to send the data to the SOSA encryption module. Once encrypted, the voice data will come back to the SMA Security Agent on the encrypted domain (Domain 2). The SMA Security Agent will send this encrypted data to the Security Service in the Domain 2 SCA container, which will use an SCA interface to pass the data to the Waveform Application. The Waveform Application then does whatever processing is necessary and sends the data to the Transceiver Device. The Transceiver Device in the SCA container then sends the digital IQ samples to the SMA Transceiver Agent which will use the SOSA SWLLI, i.e., low latency interconnect, bus to transfer the samples to the SOSA Emitter.

[Note that the interface between the Waveform Application and the Transceiver Device is not labelled as a SCA / JTNC interface. JTNC never specified a Transceiver interface so this connection is implementation dependent. We recommend the WINNF Transceiver Facility Standard for use in this case.]

2.1.5 Interfaces

2.1.5.1 Communication Protocols

The objective of the interoperability approach is to integrate an SCA-based WFA within a platform infrastructure populated by SOSA modules to provide a communications capability. Since an

objective of the approach is to minimize impact on the SOSA Modules within the system, most of the WFA interactions with other system elements are governed by the rules established in Sections 15.1, Interactions on the SOSA Message Interconnect, and 15.4.2, Interaction Binding Technology Selections of the SOSA Technical Standard, which describe the interaction patterns and technology stacks used for inter-module communication.

The only deviation of the inter-module communications approach exists for control plane communications between SCA components across a container boundary. These invocations use the native communication mechanisms of the selected SCA Platform Specific Model with the calls being transported along the SOSA Message Interconnect.

When the SOSA Container RTE is used, SOSA module software communicates across the module boundary using the SOSA interaction bindings. However, the specification does not constrain software component to component interactions that occur within a container, or between containers collocated within a module. SCA component to component communications within the "module" boundary will use the protocols and interaction mechanisms defined by their native PSM.

2.1.5.1.1 Message Format

The SMA provides the adaptation layer between the SCA and SOSA ecosystems. Assuming the SCA implementation uses the CORBA PSM, then the role of the SMA is to perform a bidirectional transformation between a SOSA Data Message (SDM) transported on the SOSA Wideband Low-Latency Interconnect and a General Inter-ORB Protocol (GIOP) message conveyed on the selected transport, e.g., TCP.

The specifics of the SDM format are described in Section 15.3.2.1 of the SOSA Technical Standard [3].

2.1.5.2 Overview

Table 1 identifies the principal interfaces utilized by a WFA to communicate with the SOSA Modules and Infrastructure elements that constitute the sensor system, i.e. SDR, platform. It is worth reiterating that the role of the SMA is as an intermediary that is responsible for converting the platform specific, e.g., CORBA, function call to the appropriate function name and over the wire protocol required for syntactic interoperability from the SCA Waveform Application to the SOSA entities that exist as part to the system. The participant on either side of the interaction must take the necessary actions to ensure that there is semantic interoperability across the SCA ←→ SOSA boundary.

SMA Agent	SOSA [3] Reference	Direction of Travel	WFA Implementation Notes
System Manager (SOSA 1.1)	Sec 6.2 - defined by In-Band System Management Interface	In/out	Allowable implementation approaches are described in Section 15.3.2.1 of the SOSA Technical Standard [3]. Some System Manager functions may be provided by the SCA CF

Task Manager (SOSA 1.2)	Sec 7.1 - SOSA Task Manager Module Interface	In/out	Allowable implementation approaches are described in Section 15.3.2.1 of the SOSA Technical Standard [3] Some Task Manager functions may be provided by the SCA CF
Cryptographic (SOSA 6.2)	Sec 12.2 - SOSA Encryptor/Decryptor Interfaces	In/out	The International Radio Security Services (IRSS) API is specified in Object Management Group IDL[13]
Host Platform Interface (SOSA 6.9)	Section 12.9 – defined by Host Platform Interface	In/out	Allowable implementation approaches are described in Section 15.3.2.1 of the SOSA Technical Standard [3]
Positioning, Navigation, Timing Data (SOSA 6.7)	Section 12.6 – defined by PNT Service	In/out	Allowable implementation approaches are described in Section 15.3.2.1 of the SOSA Technical Standard [3]. The JTNC Time Service API is typically utilized for timing capability and the JTNC GPS interface is used for position information – both APIs are specified in OMG IDL. In addition, two HW Signals exist for distribution of system time across module boundaries
RF Signal Layer (i.e., Transceiver) (SOSA 2.3)	Section 8.1	In/out	The WInnForum Transceiver PIM is specified in OMG IDL with Platform Specific mappings for FPGA, SCA, and native C++ [12]

Table 1 Waveform Application Interfaces

3 CONOPS

The CONOPS identify key scenarios, which in total should provide enough context for readers to understand the interaction patterns that should be followed for SCA-based WFAs to be integrated within SOSA modules and system infrastructure to provide the functionality of a typical SDR. The sequence diagrams contained within the document do not provide a low enough level of detail to implement a WFA or develop a system, but they should highlight most instances which are critical for effective integration of the SCA-based WFAs. The authors will be the first to admit that while all the instances where "cross platform" interactions will take place during WFA operations are not included within these diagrams, they should be sufficient to establish patterns that can be used to inform design decisions to fill the gaps.

3.1 Scenarios

The document contains eleven scenarios that are representative of common events that occur in the operational lifespan of a WFA. The specific scenarios are as follows:

- Card ecosystem instantiated
- SCA OE and waveform application instantiated
- Enable waveform application
- Update waveform configuration
- Report status
- Send voice message
- Receive voice message
- Send data message
- Receive data message
- Disable waveform application
- Deconstruct waveform application

The specific objectives, assumptions, and conditions associated with each of these scenarios are described in the subsequent subsections.

3.1.1 Card Ecosystem instantiated

Use Case: Card Ecosystem Instantiated	
Objectives	• Create Infrastructure within a PIC suitable for hosting SCA Waveform applications
Actors	• Client – human or non-person entity (NPE) that interacts with the communications system • SOSA Infrastructure – the collection of SOSA Modules and Infrastructure Elements (e.g., container engine) that constitute the communications system

<table>
<tr><td>Assumptions</td><td colspan="5">

The SOSA System Manager and Task Manager are instantiated and operating within the SOSA Ecosystem (i.e., outside of the card where the SCA Waveform Application will execute)
Any SOSA modules that will interact with WFA are operational within the communications system
MORA Resource Management will be taken care of within the SOSA Infrastructure. There should be limited/no raw MORA interactions that take place within the WFA card
SOSA modules will not need to have awareness of any of the components within the WFA implementation
In an operating platform the Domain 1 and Domain 2 interactions will occur in parallel, these interactions are shown sequentially in the diagrams for clarity and convenience

</td></tr>
<tr><td>Pre-conditions</td><td colspan="5">N/A</td></tr>
<tr><td>Initiating Event</td><td colspan="5">Power applied to card</td></tr>
<tr><td rowspan="5">Sequence of Events</td><td></td><td>Event name</td><td>Event</td><td>Input(s)</td><td>Outputs(s)/Outcomes(s)</td></tr>
<tr><td>1.</td><td>POST</td><td>Card self-test</td><td>N/A</td><td>Hardware diagnostics have been performed, and the card elements have been validated as capable of supporting WFA operations</td></tr>
<tr><td>2.</td><td>Start SMA</td><td>SOSA Module Adapter and its sub-agents are instantiated</td><td>Identity of sub-agents to be instantiated</td><td>SMA Instantiated and linked to corresponding SOSA Modules</td></tr>
<tr><td>3.</td><td>BIT</td><td>Built-in test</td><td>N/A</td><td>Key hardware and software elements have been validated, and basic functionality has been confirmed</td></tr>
<tr><td colspan="5">WFA Card in a “Ready to Operate” state
SMA up and running</td></tr>
<tr><td>Post Conditions</td><td colspan="5">Communications card successfully powered up
A singleton SMA has been instantiated on the card (should be brought up automatically)
The addressable components of the communications card, i.e., SMA and subagents, have provided their location information, “registered” with the SOSA infrastructure</td></tr>
<tr><td>Expected Outcomes</td><td colspan="5"></td></tr>
</table>

	• The operating environment and ecosystem will have been instantiated for all the WFA domains • External (i.e. other than those hosting or collocated with the SCA WFA) cards and other entities are able to interact with the communications card
Notes and Issues	• The communications card will likely be a Single Board Computer (SBC) that provides a general computing platform

Figure 6 contains the sequence diagram for the Card Ecosystem Instantiated use case and the subsections that follow provide additional information related to the interactions within the diagram.

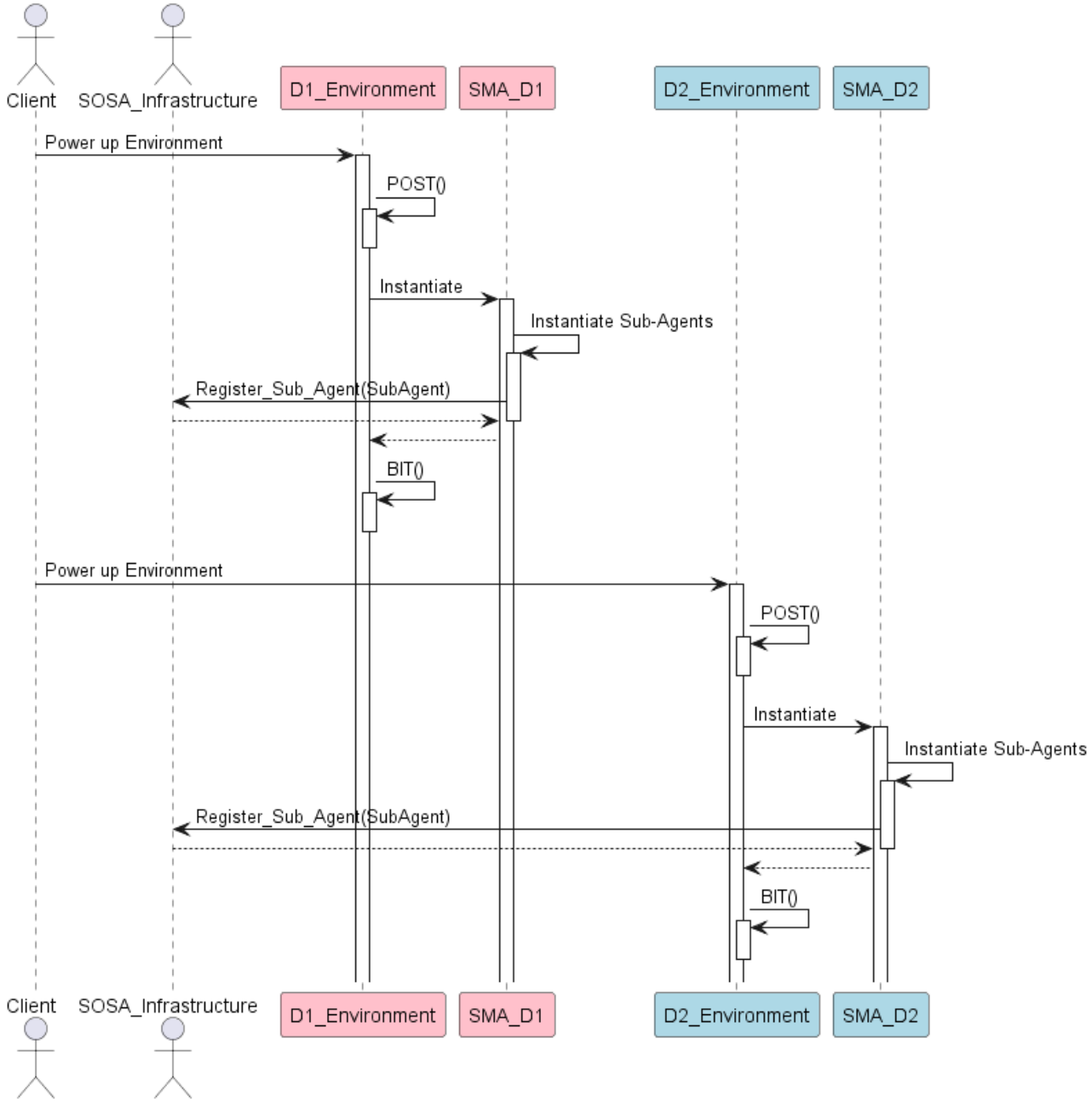

Figure 6 – Card Ecosystem Instantiated Sequence Diagram

3.1.1.1 POST

POST (Power On Self-Test) performs basic sanity checking, e.g. ensuring that the necessary processing resources exist, and validates that the identified components can execute rudimentary tasks. The presence of POST is inherent within the card, i.e., no external stimulation is required to initiate the behavior.

3.1.1.2 Start SMA

The invocation and execution of the SMA occurs in an implementation dependent manner. Upon completion of the SMA initiation, the primary agent and some subset of subagents will have been initiated, but there is no fixed guidance on how that realization needs to occur. A single call could initiate a chain of calls, individual calls could be made to each of the individual agents, all the agents that exist within the system could be started, or only the required subset of subagents could be started. Each of the agents that come into existence will register with their corresponding entity in the SOSA infrastructure after they come into existence. As a result of the registration a bidirectional path will be established between the SOSA and SCA endpoints to facilitate communication and minimize latency between the environments.

The following agents represent the minimal set to be instantiated at start up:

- System Manager Agent
- Task Manager Agent
- PNT Agent

3.1.1.3 BIT

BIT (Built-in test) is a testing approach that is integrated within system entities (e.g., software components or physical equipment) to verify their health and functional operations. BIT is contained within the entity under test so multiple system items could be tested in parallel in a non-interfering manner. Because of the nature of BIT and how it is performed (i.e., there is not a specific sequential point in time where it could be located), this step is more of a placeholder to indicate an action that should occur as part of the instantiation process and not a prescription of when it needs to be performed.

3.1.2 SCA OE and Waveform Application instantiated

Use Case: SCA OE Waveform Application Instantiated	

Objectives	• Instantiate the SCA container(s)
	  • Create the SCA OE
	  • Connect the SCA OE components to their corresponding entities "agents" within the SMA
	  • Launch the Waveform Application within the container
	  • Inform the SOSA environment of the WFA location

Actors	• Client – human or NPE that interacts with the communications system

<table>
<tr><td>Assumptions</td><td>

The SOSA System Manager and Task Manager are instantiated and operating within the SOSA Ecosystem
Multiple (self-contained) containers may operate on the communications card; however, this version of the scenarios only address situations where there is a single container. The more complex case could be addressed in subsequent revisions.
The (outside of the WFA card) task manager controls access to all WFA execution threads.
Execution threads are managed within the WFA communications card
Zero to many WFA components may interact with other parts of the application, external CSS, or support applications
Core Framework/ORB (Upper) may call remote SCA interfaces via the regular speed data bus
A single message to the communications capability will instantiate the initial WFA and operating environment

</td></tr>
<tr><td>Pre-conditions</td><td>

Card ecosystem initiated
Communications links are established between the SOSA infrastructure and the SMA agents
SCA Image(s) exist (contain the following elements)

CF (as required)
OE Device(s)
OE Service(s)
WF Application(s)

</td></tr>
<tr><td>Initiating Event</td><td>

Start-up (initializing) message sent from SOSA management entity

</td></tr>
<tr><td>Sequence of Events</td><td>
<table>
<tr><th></th><th>Event</th><th>Event</th><th>Input(s)</th><th>Outputs(s)/Outcomes(s)</th></tr>
<tr><td>1.</td><td>Create SCA Ecosystem Container(s)</td><td>Instantiate container</td><td></td><td>Appropriate container manifest located, and SCA container created

SCA OE instantiated within container</td></tr>
<tr><td>2.</td><td>Connect SMA to SCA Container</td><td>Connect SMA to container</td><td></td><td>Communications pathway established from SOSA environment to SCA management infrastructure

SCA OE components bridged to SMA subagents</td></tr>
</table>
</td></tr>
</table>

	3.	Instantiate and Configure Waveform Application	Instantiate WFA		WFA instantiated, configured, and linked Secure data exchange path established within WFA
Post Conditions	• Initial SCA Services and Devices instantiated and initialized within container • Waveform Application instantiated within container • WFA data path allocated and established				
Expected Outcomes	• Selected / requested Waveform Applications are accessible by SOSA infrastructure • SCA WFA configured to utilize SOSA platform operating environment entities • SCA WFA resource needs factored into SOSA capacity model • Waveform Applications ready to start providing service				
Notes and Issues	• The SCA waveform application and environment should be thought of as a self-contained system. The objective of this approach is to be able to port the WFA to the SOSA environment with as few changes as possible. Consequently, the SCA container will provide the SCA infrastructure (i.e., components) required for the WFA to operate. • Most of the SCA OE components will be proxies to the SOSA infrastructure, but the WFA will not need to know that. However, the additional "hops" introduced by the adapters and proxies may introduce latencies that may affect WFA porting. • WFA(s) will be "managed" primarily from SOSA 1.1 and/or 1.2 Modules. The calls to the SOSA management modules will be proxied by the SCA management components and specialized APIs provided by the SCA components • The extent to which the native SCA management capabilities will be leveraged still needs to be determined. As part of minimizing WFA porting, the existing calls to the SCA management components will be preserved, but there hasn't been closure on whether the functioning CF will be utilized directly by any client other than SOSA modules.				

Figure 7, Figure 8, and Figure 9 contain the sequence diagrams for the SCA OE and Waveform Application instantiated use case, and the subsections that follow provide additional information related to the interactions within the diagrams.

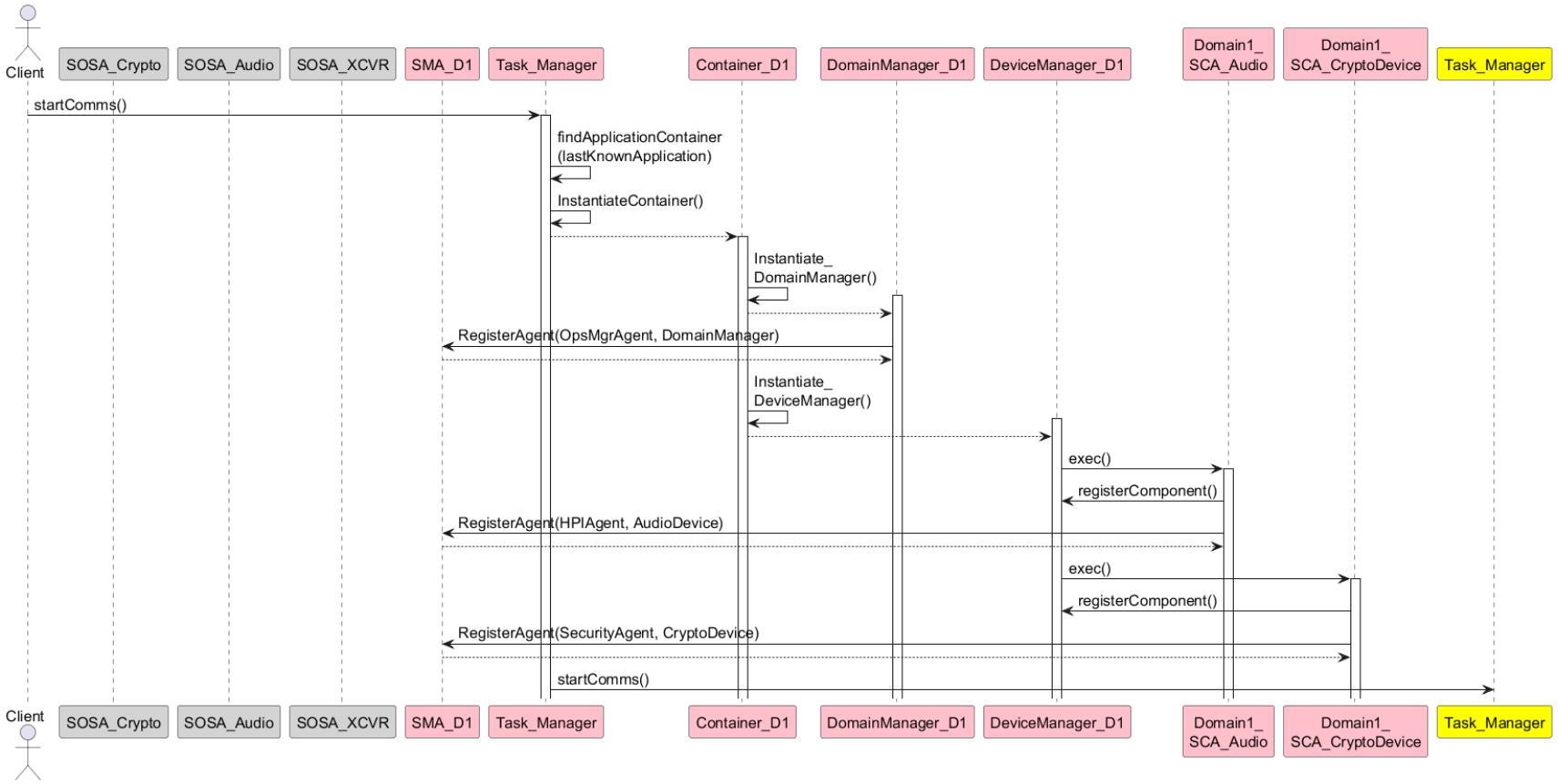

Figure 7 – SCA OE and Waveform Application instantiated Sequence Diagram – Part 1

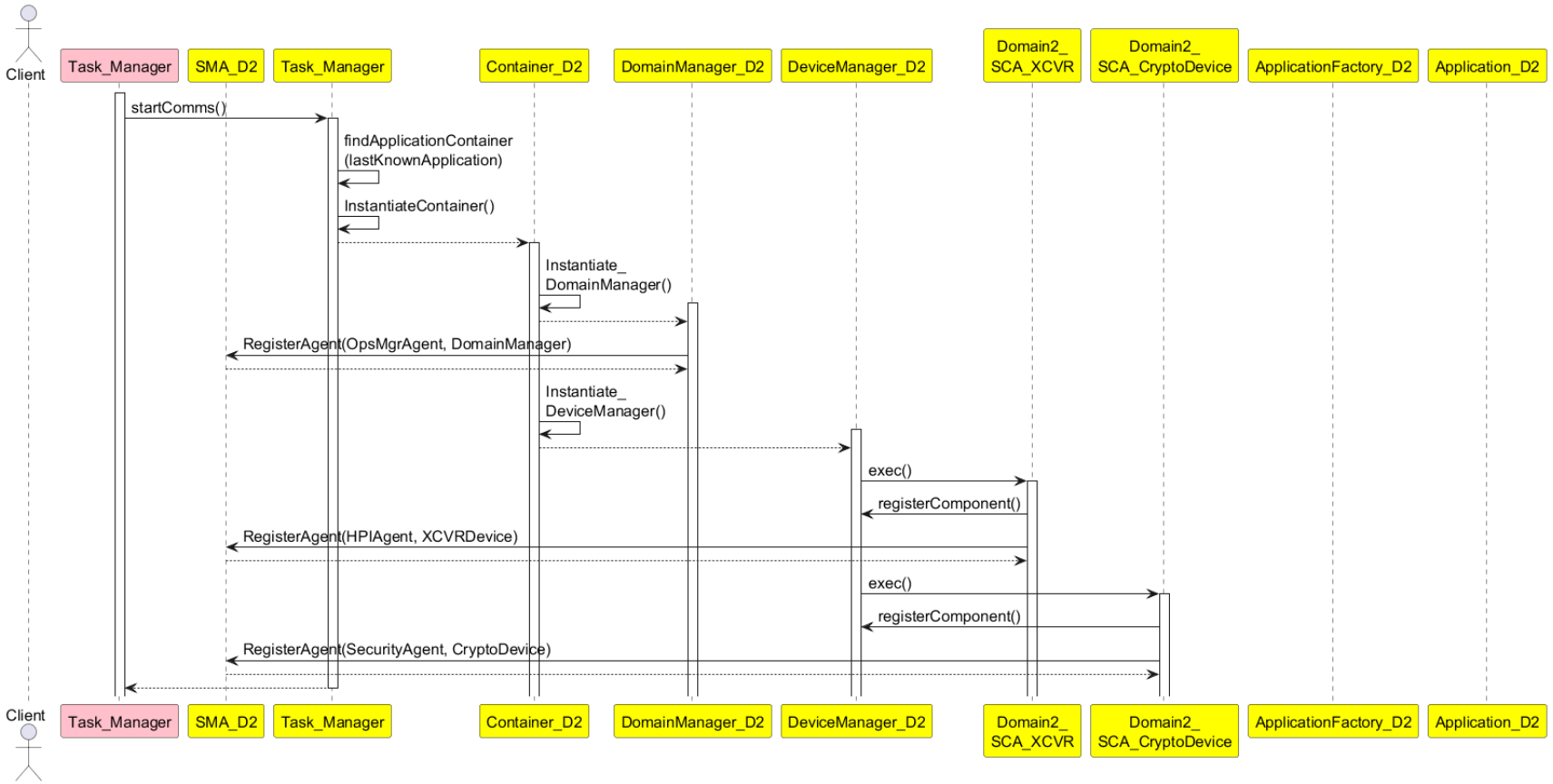

Figure 8 – SCA OE and Waveform Application instantiated Sequence Diagram – Part 2

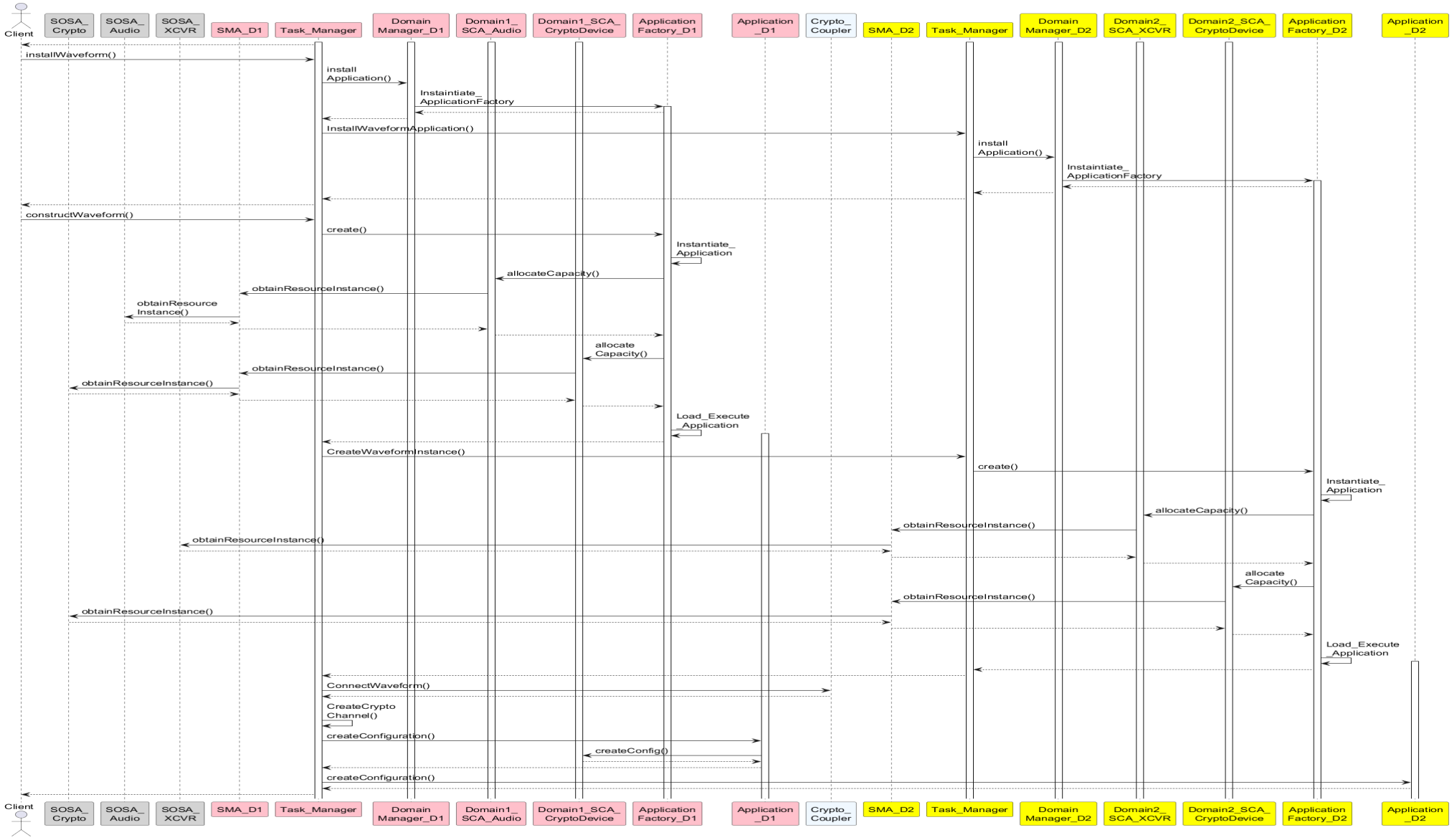

Figure 9 – SCA OE and Waveform Application instantiated Sequence Diagram – Part 3

3.1.2.1 Create SCA Ecosystem Container(s)

The system will likely have its WFA images stored in a repository and accessible via a registry. Having the "library" of available images will expand the ability of the system to have an operating profile that is flexible in nature and able to support a breadth or WFAs, assuming the platform HW is able to support the over the air and resource requirements of the candidate applications.

After the container is instantiated, a series of steps are followed to have the environment function identically to how it would have been in a native SCA implementation. The two primary SCA OE components, DomainManager and DeviceManager are instantiated. A byproduct of the DeviceManager instantiation is the creation of any operating environment Devices or Services that are identified by the manager's descriptor file. As described elsewhere, the Device or Service implementations may be actual implementations or proxies for the SOSA environment.

3.1.2.2 Connect SMA to SCA Container

After the OE components come into existence, we want to ensure that they can be managed by the SOSA management modules. To provide this capability, the OE components must make their identity known outside of the SCA components. The sequence diagram shows each OE component registering with the DeviceManager that instantiated it per normal SCA functionality. This behavior was not modified from its original implementation, but an additional call was introduced to have the OE component also register with its corresponding SMA subagent. Different implementation approaches could be taken at this point, e.g., the DeviceManager could aggregate all of its registrants and perform a bulk registration, but either approach achieves the same end. Additional analysis or prototyping would identify the optimal approach quickly, but the distinction is not material for the purpose of this scenario. The result of this registration is that we have an end-to-end management path from a SOSA management module to SCA OE components.

3.1.2.3 Instantiate and Configure Waveform Application

The WFAs can then be instantiated after the OE has been established. The SCA two-step process is highlighted in this scenario whereby the WFA is first installed and then it is constructed. Practically speaking, it is not necessary to perform both of these steps after the WFA has been installed. At that point, the identity of the factory component that can instantiate the WFA has been created and incorporated within the SCA infrastructure and can be reused without need of the additional step. A critical part of the WFA creation process occurs just prior to the execution of the WFA process. The SCA infrastructure attempts to allocate the necessary resources from the underlying infrastructure to ensure that the WFA can function properly. In the SOSA ←→ SCA interoperability approach, it is important to realize that some of those resources may exist organically outside of the SCA environment, so it may be necessary for SCA OE proxy to delegate the resource request to its corresponding SOSA module. This paper doesn't dictate how the resource allocation might be performed is SOSA. Many SCA implementations utilize an approach where resources are claimed when a WFA comes into existence and released when it is terminated. This strategy works, but it is not tailored to optimize resource utilization which could be an important characteristic of a system with SOSA modules that could have multiple sensors that are sharing resources operating in parallel.

3.1.3 Enable WF Application

<table>
<tr><th colspan="6">Use Case: Enable WF Application</th></tr>
<tr><td>Objectives</td><td colspan="5">• Transitions SCA card-based waveform application into an operational state</td></tr>
<tr><td>Actors</td><td colspan="5">• Client – human or NPE that interacts with the communications system</td></tr>
<tr><td>Assumptions</td><td colspan="5">• The SOSA Task Manager is operating within the SOSA Ecosystem and linked to the SCA WFA
• The SOSA Infrastructure is capable of providing a suitable executing environment for the SCA WFA</td></tr>
<tr><td>Pre-conditions</td><td colspan="5">• WF Application is installed and configured
• SCA Infrastructure is operational
• WF Application is connected to the SMA</td></tr>
<tr><td>Initiating Event</td><td colspan="5">• Request enable message sent from SOSA management entity</td></tr>
<tr><td rowspan="4">Sequence of Events</td><td></td><td>Event</td><td>Event</td><td>Input(s)</td><td>Outputs(s)/Outcomes(s)</td></tr>
<tr><td>1.</td><td>Send enable message to SCA WFA</td><td>Request Enable</td><td></td><td>Direct enable request to specified SCA WFA</td></tr>
<tr><td>2.</td><td>Enable WFA</td><td>Enable WFA</td><td></td><td>Transition WFA into an operational state</td></tr>
<tr><td colspan="5">• WFA is able to provide communications services to system clients</td></tr>
<tr><td>Post Conditions</td><td colspan="5">• SOSA infrastructure is aware of the fact that the WFA is providing service
• SOSA audit log has received a record that indicates WFA has been brought into service</td></tr>
<tr><td>Expected Outcomes</td><td colspan="5">• Much of what needs to occur within a WFA to bring it into service is implementation specific so it is difficult to identify a one size fits all description of what needs to take place; however, the specifics are largely self-contained within the SCA container, so as long as the “SOSA infrastructure” has enough capability to support the WFA operational requirements the details of what occurs during start up should not be material.</td></tr>
<tr><td>Notes and Issues</td><td colspan="5"></td></tr>
</table>

Figure 10 contains the sequence diagram for the Enable WF Application use case, and the subsections that follow provide additional information related to the interactions within the diagram.

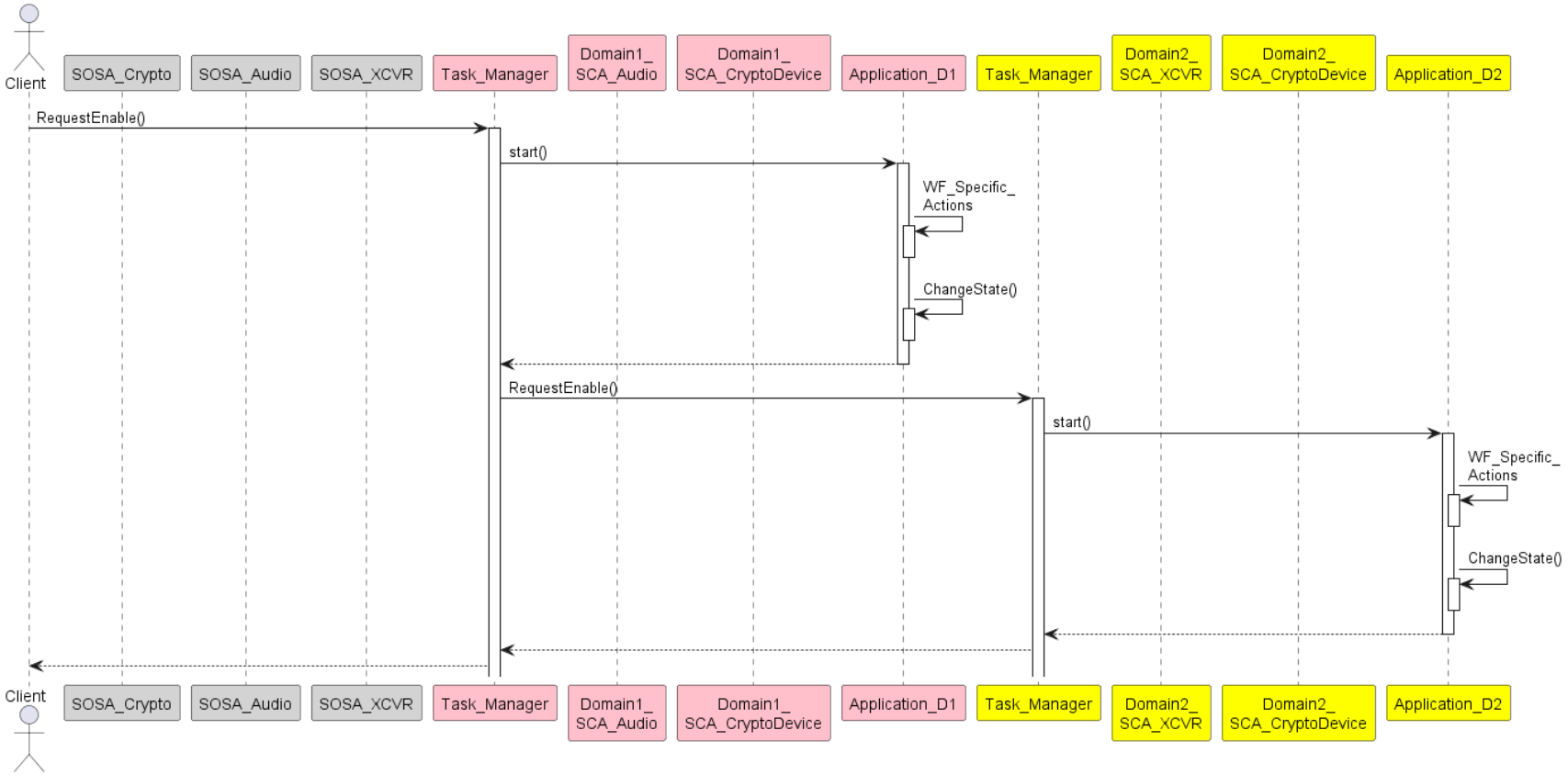

Figure 10 – Enable WF Application Sequence Diagram

3.1.3.1 Request Enable

A client initiates the process of transitioning the communications system into an operational state. The physical stimulus of bringing the WFA into an operational capability. The action could be one of several actions such as hitting a button on a screen, tapping a button on a device, or an API call being invoked from a cooperating application. The client initiated enable request will be directed to the SOSA Task Manager. The Task Manager subsequently redirects the enable request to the WFA.

3.1.3.2 Enable WFA

When the WFA receives the enable request, mapped to an SCA start(), several functions are initiated that result in bringing the WFA into operations. A key design parameter of the SCA and this approach is the focus on providing access to the edges of the communications applications and not focusing too much on the architecture of how the service needs to be developed. As a result, this document treats that step as a blob where the WFA developer does whatever is necessary to bring the capability into existence. Within the implementation, there are a couple of specific actions that need to occur to show that progress is being made such as transitioning a component attribute to a value that indicates the component is operating. Events such as this transition or other milestones could provide opportunities that WFA might leverage to notify the system that progress is being made. Event notifications like those identified in the prior systems might leverage the SCA / SOSA notification/event services to send those messages to the SOSA Management Modules.

3.1.4 Update WF Configuration

Use Case: Update WF Configuration					
Objectives	• Provides a method to dynamically reconfigure the properties and attributes of the waveform application				
Actors	• Client – human or NPE that interacts with the communications system				
Assumptions	• Supporting system physical infrastructure and services are in an operating condition and necessary resources are allocated to the WFA • Command input media are accessible by the client				
Pre-conditions	• SDR is able to accept commands from the client • WFA is in an operating condition				
Initiating Event	• Configuration message sent from an SDR client				
Sequence of Events		**Event**	**Event**	**Input(s)**	**Outputs(s)/Outcomes(s)**
	1.	Send configuration request to WFA	Reconfigure WFA		Appropriate WFA component receives the client reconfiguration request

	2.	WFA sends configuration request to SOSA Transceiver	Reconfigure Transceiver		Shared system Transceiver device has received requested reconfiguration
	3.	Transceiver validates and then performs configuration update	Update configuration		Reconfiguration has been performed, and the system is operating with the new parameters
	4.	Return to normal WFA operations	Continue operations		
	5.	Send configuration request to WFA	Reconfigure WFA (2)		Appropriate WFA component receives the client reconfiguration request
	6.	WFA sends configuration request to SOSA Transceiver	Reconfigure Transceiver (2)		Shared system Transceiver device has received requested reconfiguration
	7.	Transceiver successfully performs configuration update	Update configuration (2)		Reconfiguration has been performed but the system is not in an operational state
	8.	Transceiver not able to perform configuration update	Reconfiguration failure		The transceiver properties have been restored to their pre-reconfiguration values and the WFA is not in an operational state
	9.	Return to normal WFA operations	Continue operations (2)		All activities associated with the attempted reconfiguration have been performed and the WFA is in an operational state
Post Conditions	• WFA repository is updated with newly configured settings				

Expected Outcomes	• WF Application is operating with updated settings (if successful) or the original settings if the configuration update could not be performed.
	• Depending on the implementation, the WFA might send a notification to the system that reflects the configuration change. This change may or may not be required based on customer requirements. • There are several "value added" implementation strategies that could be applied to this scenario to make this more robust. For example, a fault recovery type of extension could be incorporated that would seek out alternative frequency ranges if the one provided could not be obtained.

Figure 11 contains the sequence diagram for the Update WF Configuration use case, and the subsections that follow provide additional information related to the interactions within the diagram.

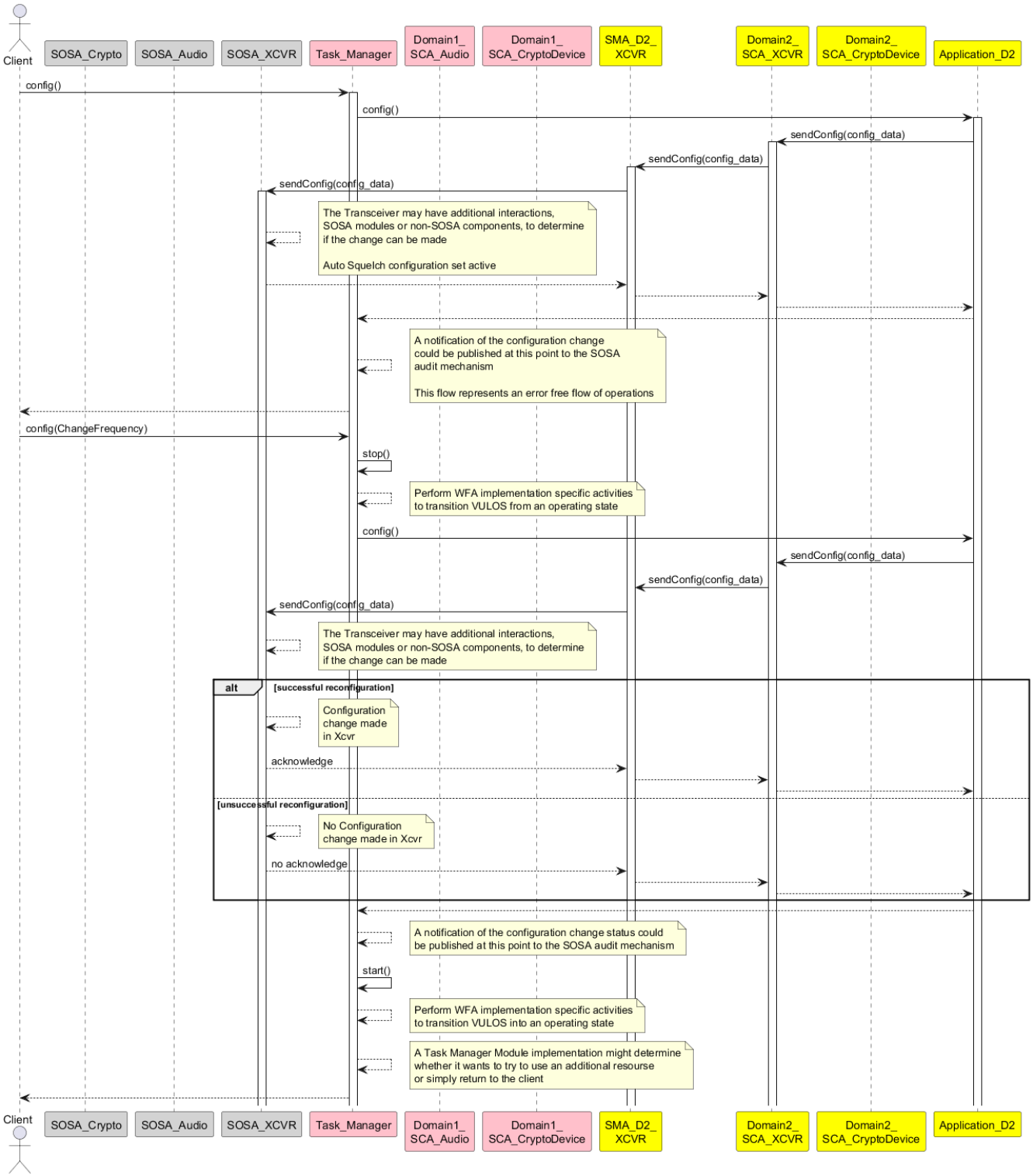

Figure 11 – Update WF Configuration Sequence Diagram

3.1.4.1 Reconfigure WFA

The client initiates a "configuration event" to modify a WFA setting or property. The config call moves through the system until it reaches the WFA. The path of calls from client to WFA was established as part of the WFA and OE instantiation portions of the system operation. While enroute to the WFA the configuration call passes through the SMA where it is converted from an SDM to a request aligned with the SCA WFA middleware.

3.1.4.2 Reconfigure Transceiver

The WFA is responsible for sending the configuration request to the appropriate target location. There are two components associated with preforming a WFA configurations change. First, the property value assignment must be updated within the WFA "repository". The repository could be a value maintained in memory or stored in some type of non-volatile storage area. The second aspect of the configuration change would be for the effect of the configuration change to be reflected in the appropriate location within the system of systems. In this example, the Auto Squelch setting would be updated within the WFA component and the configuration change would be reflected within the shared system Transceiver. The config change would be sent to the SCA Transceiver Device within the container where it would be transported to the SMA, converted to an SDM, and sent in that format to the Transceiver over the system control bus.

3.1.4.3 Update Configuration

The physical config update, i.e., the second aspect of the configuration change, occurs within the Transceiver. The activities that take place during this phase of the scenario are beyond the scope of the SOSA ←→ SCA interoperability activities, but worth mentioning because it is an area that needs to be addressed within the system infrastructure. Steps should be taken within the Transceiver to ensure that the configuration change can be realized prior to the update being performed. Concerns that must be accounted for are related to questions regarding what other services or applications are using the shared resource or how this change might impact other sensors operating within the system of systems. Once the validation has been performed, the configuration change can be made within the device, and it becomes effective within the system.

3.1.4.4 Continue Operations

After the configuration change has been made, the flow of control unwinds and the operation of the SDR returns to the WFA being in an operational state, albeit with Auto Squelch being activated. The flow of control follows the path from the Transceiver device through the SMA to the SCA Transceiver Devices the Domain 2 WFA, and finally back to the client.

3.1.4.5 Reconfigure WFA (2)

At this point in the scenario, VULOS is operating within the SDR. Although it is somewhat contrived, we will assume that the client decides to make another configuration change within the system. In this instance, the user determines that they want to modify the operating frequency of the WFA. The fundamental activity of this step is like that of Section 3.1.4.1 with one major exception. Changing

the frequency of the WFA requires a two-step process, so before the config call is sent to the WFA it must first be stopped so the change can be performed. Stopping a WFA is an implementation dependent process, but it typically includes activities such as disallowing some new client requests from being sent to the WFA and allowing existing WFA processing to proceed to a graceful termination.

3.1.4.6 Reconfigure Transceiver (2)

This step is identical to that of Section 3.1.4.2, with the exception being that the reconfigure request is to change the frequency.

3.1.4.7 Update Configuration (2)

This step is like that of Section 3.1.4.3 with one distinction. After the configuration change has been made, the WFA does not return to normal operations. Recall, that the first step of this configuration change was to stop the WFA. As a result, the act of implementing the modification does not automatically restart the WFA. The final step of the successful frequency modification is to send a completion acknowledgement back to the SCA Transceiver.

3.1.4.8 Reconfiguration Failure

When the Transceiver reconfiguration is not successful, a completion non-acknowledgement is sent to the SCA Transceiver. In addition, the Transceiver implementation must ensure that it backs out any configuration modifications that may have occurred prior to the failure being encountered.

3.1.4.9 Continue Operations (2)

After the configuration change has been made, the flow of control unwinds and the operation of the SDR returns with the requested frequency at the new (if the change was successful) or the original value. The final step of this process is for the Task Manager to send a request to the WFA for it to be started and returned to normal operations.

3.1.5 Report Status

Use Case: Report Status	
Objectives	• Enables the waveform application to share information about its status and operations with the system management entity or other clients
Actors	• Client – human or programmatic entity that interacts with the communications system
Assumptions	• The requisite SOSA modules are instantiated and operating within the SOSA Ecosystem • The WFA is operating within the communications card • SOSA clients can receive OSA event messages • The SCA Transceiver device has subscribed to receive Transceiver notifications

<table>
<tr><td>Pre-conditions</td><td colspan="5">• SDR WFA and OE in an operational state
• SCA OE implements that Event Channel, and the Event Service is operating
• SMA has registered as an SCA Event Channel consumer
• The SOSA Security Module if configured to receive SYSLOG messages
• SCA Transceiver has registered as a SOSA Transceiver notification client</td></tr>
<tr><td>Initiating Event</td><td colspan="5">• Client initiates an action to reconfigure the WFA
• SOSA Transceiver experiences a fault and reports an error</td></tr>
<tr><td rowspan="7">Sequence of Events</td><td></td><td>Event</td><td>Event</td><td>Input(s)</td><td>Outputs(s)/Outcomes(s)</td></tr>
<tr><td>1.</td><td>Configure WFA</td><td>Client reconfigures WFA settings</td><td>New property value</td><td>Client configuration change request propagated through the system to the WFA</td></tr>
<tr><td>2.</td><td>Raise Event</td><td>WFA generates informational message</td><td>Event type</td><td>WFA has completed configuration change and issued SCA event to communicate change</td></tr>
<tr><td>3.</td><td>Receive Event</td><td>SMA receives configuration change status message</td><td>Config message</td><td>SMA receives event notification from SCA, generates SYSLOG notification, and SOSA module has received message</td></tr>
<tr><td>4.</td><td>Notify Error</td><td>Transceiver device recognizes and reports error</td><td>Error type</td><td>Physical device experienced error condition and propagated it to the WFA</td></tr>
<tr><td>5.</td><td>Raise Event</td><td>WFA generates error message</td><td>Event type</td><td>WFA has completed event response and issued SCA event to communicate fault</td></tr>
<tr><td>6.</td><td>Receive Event</td><td>SMA receives error report message</td><td>Error message</td><td>SMA receives event notification from SCA, generates SYSLOG notification, and SOSA module has received message</td></tr>
</table>

Post Conditions	• SOSA clients have received WFA notifications
Expected Outcomes	• After the messages have been reported and forwarded to the client(s), the system elements continue operating in accordance with the nature of the incident. Events such as a configuration change message are informational in nature and after they have been issued, the WFA typically continues operating normally. In many instances, these messages are received by the client and stored in a log or audit repository. When an error condition is reported, the resulting actions are usually implementation-dependent based on the nature of the error. Three alternative results are that the message could be ignored, and the system could continue operating in a degraded state, upon identification of the error the device could initiate some sort of self-recovery activity, or the receipt of the message by SOSA could initiate some sort of fault management activity to diagnose and repair the issue.
Notes and Issues	• N/A

Figure 12 contains the sequence diagram for the Report Status use case, and the subsections that follow provide additional information related to the interactions within the diagram.

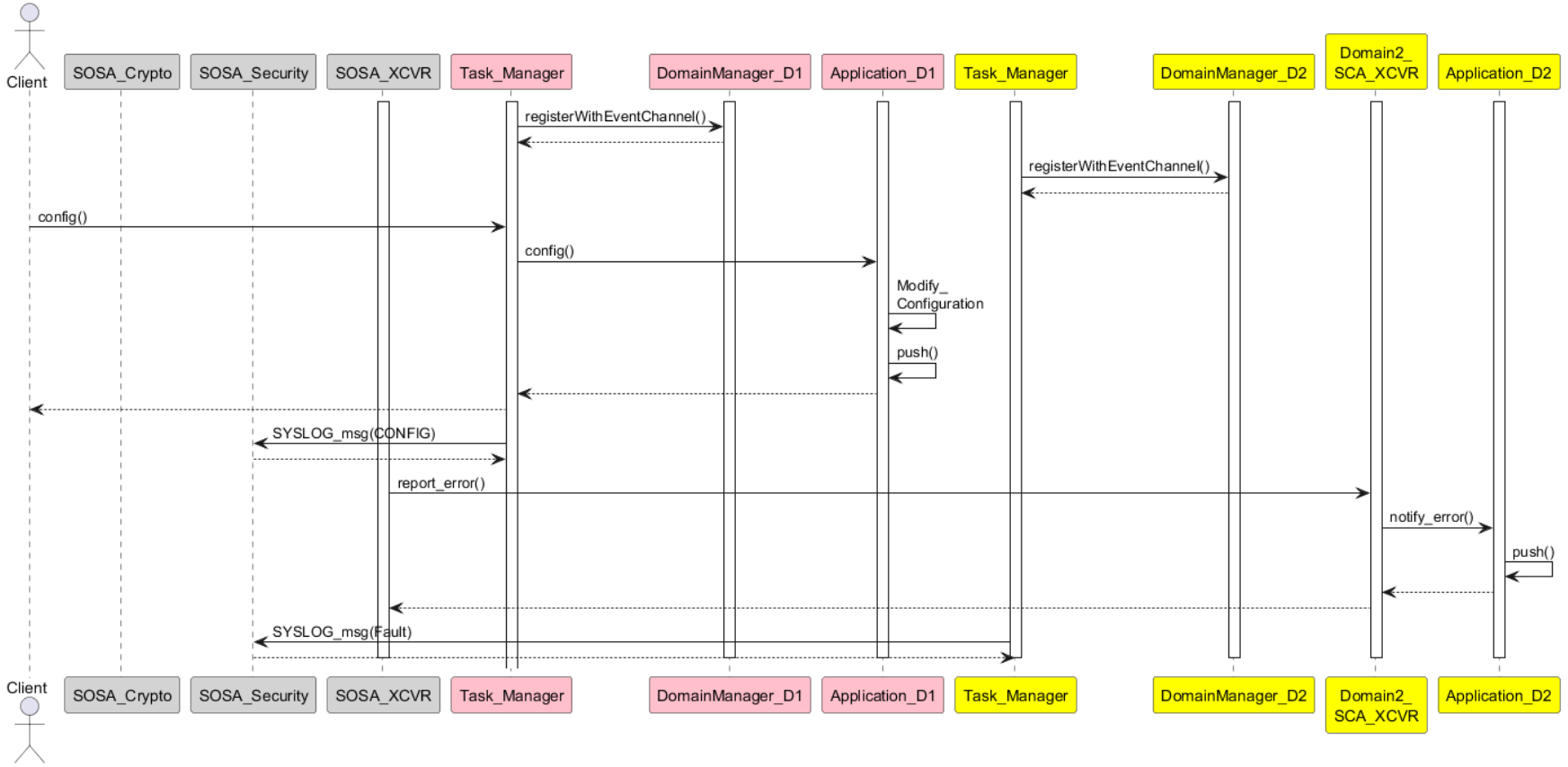

Figure 12 – Report Status Sequence Diagram

A precondition to this initial step is the registration of the SMA with the SCA event channel. Registration allows the system component to be alerted about, and receive, notifications originating from the SCA ecosystem. The event service registration is something that only occurs once over the OE lifespan, so it would not need to occur prior to each configuration change or fault encountered.

3.1.5.1 Configure WFA

The client initiates a "configuration event" to modify a WFA setting or property. The config call moves through the system until it reaches the WFA. The path of calls from client to WFA was established as part of the WFA and OE instantiation portions of the system operation.

3.1.5.2 Raise Event (2)

Upon receipt of the client command, the WFA implementation follows its internal sequence of commands to affect the change of the WFA operational status. Once the change has been finalized, the WFA uses its identified Event Channel to share the fact that the change occurred with all subscribers for that event type.

3.1.5.3 Receive Event (3)

Because the SMA registered for the appropriate event type, it receives a copy of the WFA initiated event. Upon receipt of the message, the SMA is and generates an equivalent SYSLOG message which it sends to the SOSA Security module. It is beyond the scope of this document, and the interoperability approach to dictate what happens to the message information after is has been received by the SOSA module, but there should be enough coordination between the SCA developers, SOSA developers, and/or system integrators to ensure that none of information about the generated event information is lost and not preserved for the purpose of audits or post mortems.

3.1.5.4 Notify Error

A precondition for this step is that the SCA Transceiver has registered to receive notifications from the SOSA Transceiver. Within this scenario, the initiating event is the SOSA transceiver's recognition of an abnormal event. The transceiver then goes through its normal processing and publishes an event notification using the mechanism as specified by the SOSA Technical Standard [3].

3.1.5.5 Raise Event (5)

Upon receipt of the error notification, the WFA implementation follows its implementation of specific sequence of commands to respond to the situation. This step represents another opportunity for an architectural decision to be made regarding how the WFA should respond. For this CONOPS the WFA makes a first pass to react to and resolve the situation within the SCA environment. If a successful resolution is performed, then the generated message from the WFA will be more informational in nature. However, if the WFA is not able to resolve the situation, then the WFA is forced to generate a message that can be consumed and addressed within the SOSA modules. Once the appropriate message has been identified, the WFA uses its identified Event Channel to share the fact that the change occurred with all subscribers for that event type. An alternative approach for this

series of events would be for the WFA to cede all fault response / management responsibilities to SOSA and not attempt to make any adjustments locally.

3.1.5.6 Receive Event (6)

Because the SMA is registered for the appropriate event type, it receives a copy of the WFA initiated event. Upon receipt of the message, the SMA generates an equivalent SYSLOG message which it sends to the SOSA Security module. It is beyond the scope of this document, and the interoperability approach to dictate what happens to the message information after it has been received by the SOSA module, but there should be enough coordination between the SCA developers, SOSA developers, and/or system integrators to ensure that an appropriate response can be initiated, none of information about the generated event information is lost, and any recovery action or WFA anomaly is preserved for the purpose of audits or post mortems.

3.1.6 Send Voice Message

<table>
<tr><th colspan="6">Use Case: Send Voice Message</th></tr>
<tr><td>Objectives</td><td colspan="5">• Allows the waveform application to transmit a voice message to an external recipient</td></tr>
<tr><td>Actors</td><td colspan="5">• Client – human user that interacts with the communications system
• Antenna – physical device that transmits the signal generated by the WFA over the air</td></tr>
<tr><td>Assumptions</td><td colspan="5">• The network infrastructure to transmit the OTA messages has been established
• The necessary system infrastructure for the WFA to send the message is operational
• The WFA implementation includes all the necessary cryptographic elements for secure operations
• The client has performed the WFA specific action to initiate the SDR communications capability</td></tr>
<tr><td>Pre-conditions</td><td colspan="5">• The WFA has been instantiated, connected, and is in an operating state
• The WFA that is going to transmit the OTA message has been selected and configured</td></tr>
<tr><td>Initiating Event</td><td colspan="5">• Client speaks into the SDR microphone</td></tr>
<tr><td rowspan="3">Sequence of Events</td><td></td><td>Event</td><td>Event</td><td>Input(s)</td><td>Outputs(s)/Outcomes(s)</td></tr>
<tr><td>1.</td><td>Send to Audio Device</td><td>Client initiated message acquired by system and sent to system Audio device</td><td></td><td>Data associated with user message collected within physical audio device</td></tr>
<tr><td>2.</td><td>Send to WFA (1)</td><td>Collected audio data routed to WFA and</td><td></td><td>Voice message has been digitized, encoded, and</td></tr>
</table>

			processed within domain 1		otherwise processed by the WFA
	3.	Send to WFA (2)	Voice message transported to domain 2 portion of WFA		Digitized voice message has been encrypted and prepared for additional processing
	4.	Send to Transceiver	Voice message sent to Transceiver for RF Chain processing		Digitized voice message has been upconverted, modulated, and is ready for transmission
	5.	Send Over the Air	Voice message sent to recipient(s)		Voice message sent to WFA specific antenna and transmitted OTA
Post Conditions	• SDR reverts to its operational state awaiting its next action				
Expected Outcomes	• For this sunny day case, the message has been transmitted, the system is in a steady state awaiting its next stimulus or "initiating action", and the client is free to take whatever next step is deemed appropriate. It is important to note that transmitting voice messages is a stateless / discrete event which, technically speaking, should not have any residual impact on the system or the WFA after the message has been sent. The act of sending a message will not have any impact on the WFA resource allocations or associated priorities.				
Notes and Issues	• A very sophisticated WFA or management infrastructure implementation might attempt to reallocate resources between messages, but that is beyond the scope of the documented SCA capacity management approach.				

Figure 13 contains the sequence diagram for the Send Voice Message use case, and the subsections that follow provide additional information related to the interactions within the diagram.

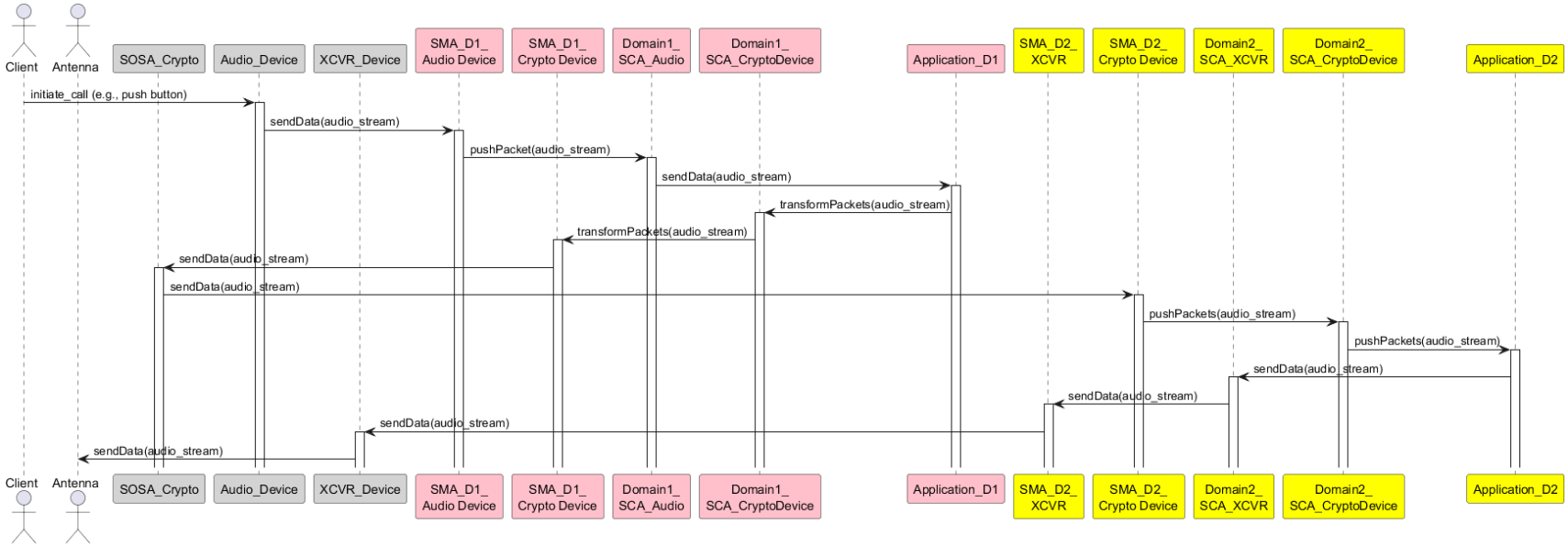

Figure 13 – Send Voice Message Sequence Diagram

3.1.6.1 Send to Audio Device

The human client performs an action to initiate the process of sending a voice message. This action is a WFA implementation dependent activity, but it is typically something similar to pressing a button on the SDR microphone. The client then speaks into the microphone, and the spoken words are transformed into their corresponding digital representation and routed to a system Audio Device. The Audio Device is a physical element, not a SOSA module, that resides within the system of systems and is an essential component of the SDR.

3.1.6.2 Send to WFA (1)

The result of this activity is that the data stream that represents the voice message is transported to the domain 1 portion of the WFA. However, the transport of the data must go through several steps to reach its destination. The Audio device in this example incorporates both audio processing and vocoding functionality and upon exit from the device, the voice message has been transformed into an encoded digital representation of the message that has been inserted within an SDM for transport to the Audio device SMA. The SMA receives the over the wire message and forwards the message to its corresponding SCA AudioDevice using a native SCA message exchange protocol. Lastly, the message is sent to the WFA from the AudioDevice. The connection from the AudioDevice to the WFA was established as part of the WFA installation and configuration.

3.1.6.3 Send to WFA (2)

The purpose of this action is to send voice messages over the cryptographic boundary to the domain 2 portion of the WFA. The domain 1 portion of the WFA sends the voice message to the domain 1 side SCA crypto device. After some intermediate processing, the message is transported to the platform (shared) crypto device within the SOSA infrastructure where the content is encrypted using WFA specific algorithms and software. After the encryption is complete, the message is sent to the domain 2 SCA crypto device (the path between the crypto devices is established as part of WFA instantiation). The final step of the process has the voice message conveyed using the Security API to the domain 2 WFA.

3.1.6.4 Send to Transceiver

The domain 2 portion of the WFA focuses on processing the voice message and readying it for broadcast. Functions such as encoding and quantization may occur within the WFA before the message is sent to the transceiver. Since the transceiver is a shared platform resource, the standard flow of control will need to occur for the message to be transported to the device. The domain 2 WFA sends the data to the SCA Transceiver device, which in turn sends the message to the Transceiver SMA where it is converted to an SDM and sent to the platform Transceiver.

3.1.6.5 Send Over the Air

The SOSA platform transceiver encapsulates the WFA RF chain processing, e.g., low noise amplifiers, power amplifiers, digital to analog conversion, and frequency converters, and prepares the voice message for delivery to the antenna where it can be broadcast OTA.

3.1.7 Receive Voice Message

Use Case: Receive Voice Message					
Objectives	• Allows the waveform application to receive a voice message from an external communications device				
Actors	• Client – human user that interacts with the communications system • Antenna – physical device that receives the OTA WFA signal				
Assumptions	• The network infrastructure to receive the OTA messages has been established • The necessary system infrastructure for the WFA to receive the message is operational • The WFA implementation includes all the necessary cryptographic elements for secure operations • The client has performed the WFA specific action to initiate the SDR communications capability				
Pre-conditions	• The WFA has been instantiated, connected, and is in an operating state • The physical communications infrastructure is ready and able to receive signals				
Initiating Event	• OTA communications signal arrives at the SDR antenna				
	1.	Send to Transceiver	Received OTA message routed to Transceiver for RF chain processing		Received signal has been forwarded to the SDR for RF chain processing
	2.	Send to WFA (2)	OTA message transported to domain 2 portion of WFA		Received voice message has been downconverted, demodulated, and processed by the WFA
	3.	Send to WFA (1)	OTA message transported to domain 1 portion of WFA		Received voice message has been decrypted, decoded, and otherwise processed by the WFA
	4.	Send to Audio Device	OTA message sent to Audio Device for final conveyance		Digitized voice message has been upconverted, modulated, and is ready for broadcast

	5.	Send to User	OTA message undergoes final processing and is sent as audible message to Client		Analog voice message emitted from Audio Device and can be heard by Client
Post Conditions	• SDR reverts to its operational state awaiting its next action				
Expected Outcomes	• The message received has been broadcast by the system designated Audio device and hopefully heard by the human system user. Similar to sending a voice message, receipt of a message does not have any bearing on any future operations of the WFA or the SDR.				
Notes and Issues	• The extent of what is included or affected as part of the message receipt process can vary on a WFA basis. This scenario represents a simplistic approach which aims to identify a model of the "major muscle movements" of what needs to occur.				

Figure 14 contains the sequence diagram for the Receive Voice Message use case, and the subsections that follow provide additional information related to the interactions within the diagram.

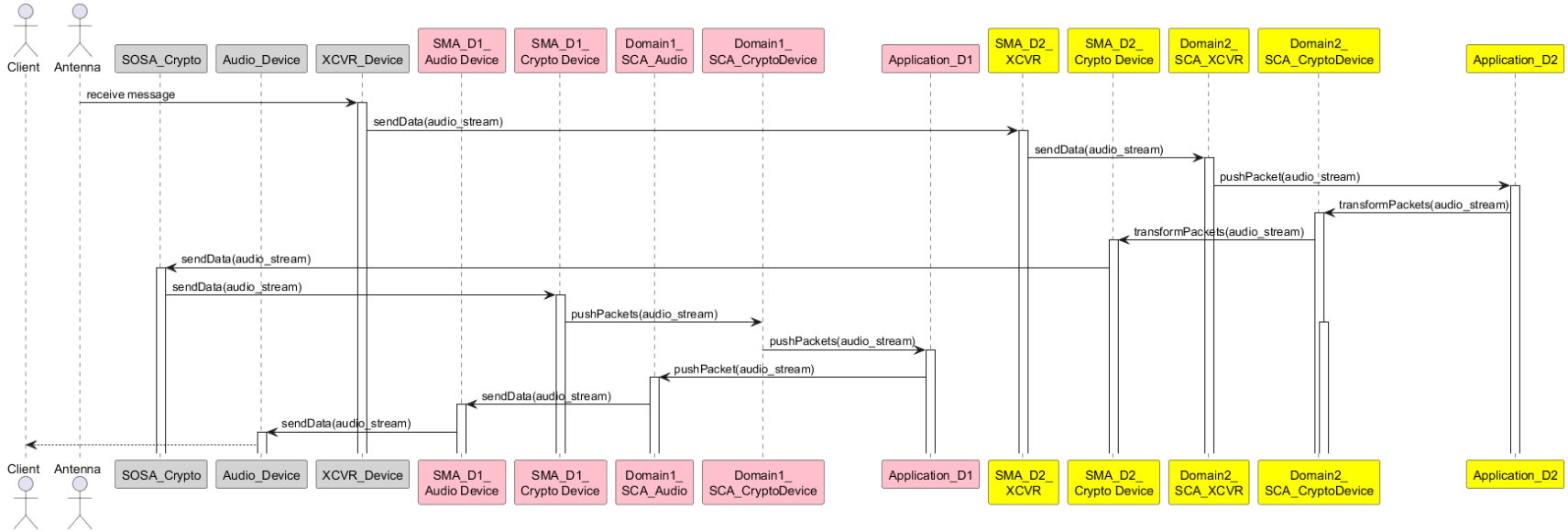

Figure 14 – Receive Voice Message Sequence Diagram

3.1.7.1 Send to Transceiver

The signal in space is sensed and received by an antenna of the SDR where it is routed to the SOSA Transceiver. In this context, the SOSA Transceiver is equivalent to the RF processing chain. The processing that occurs during this initial phase of receiving a voice message is largely outside the scope of SOSA to SCA interoperability and the WFA functionality.

3.1.7.2 Send to WFA (2)

The result of this activity is that the data stream that represents the voice message is transported to the domain 2 portion of the WFA. The data transport will proceed through all the steps of the RF chain processing, e.g., analog to digital conversion, amplification, filtering, while within the realm of the Transceiver. Once the processing is complete, the digitized voice message data will be sent as an SDM to the Transceiver SMA. The SMA will transform the SDM to an equivalent message that can be sent by the SCA Transceiver Device using the platform specific, i.e., CORBA in this instance to the domain 2 portion of the WFA.

3.1.7.3 Send to WFA (1)

The purpose of this action is to send voice messages over the cryptographic boundary to the domain 1 portion of the WFA. The domain 2 portion of the WFA sends the voice message to the domain 2 SCA crypto device. After some intermediate processing the message is transported to the platform (shared) crypto device within the SOSA infrastructure where the content is decrypted using WFA specific algorithms and software. After the decryption is complete the message is sent to the domain 1 SCA crypto device (the path between the crypto devices is established as part of WFA instantiation. The final step of the process has the voice message conveyed using the Security API to the domain 1 WFA.

3.1.7.4 Send to Audio Device

The domain 1 portion of the WFA focuses on processing the voice message and readying it for delivery to the system user. Functions such as decoding may occur within the WFA before the message is sent to the Audio Device. Since the Audio Device is a shared platform resource, the standard flow of control will need to occur for the message to be transported to the device. The domain 1 WFA sends the data to the SCA Audio device, which in turn sends the message to the Audio SMA where it is converted to an SDM and sent to the platform Audio Device.

3.1.7.5 Send to User

The SOSA platform Audio device encapsulates data stream functions such as decoding the message and sending it to a physical Audio Device such as a speaker where it can be broadcast to the human system user.

3.1.8 Send Data Message

Use Case: Send Data Message					
Objectives	• Allows the waveform application to transmit a data message to an external communications device				
Actors	• Client – human or NPE that interacts with the communications system • Antenna – physical device that transmits the signal generated by the WFA over the air				
Assumptions	• The network infrastructure to transmit the OTA messages has been established • The necessary system infrastructure for the WFA to send the message is operational • The WFA implementation includes all the necessary cryptographic elements for secure operations • The client has performed the WFA specific action to initiate the SDR communications capability				
Pre-conditions	• The WFA has been instantiated, connected, and is in an operating state • The WFA that is going to transmit the OTA message has been selected				
Initiating Event	• Client enters message into the SDR system				
Sequence of Events		**Event**	**Event**	**Input(s)**	**Outputs(s)/Outcomes(s)**
	1.	Send to Digital Device	Client initiated message acquired by system and sent to system Digital device		Data associated with user message collected within physical Digital device
	2.	Send to WFA (1)	Collected data message routed to WFA and processed within domain 1		Data message has been encoded, and otherwise processed by the WFA
	3.	Send to WFA (2)	Data message transported to domain 2 portion of WFA		Data message has been encrypted and prepared for additional processing
	4.	Send to Transceiver	Data message sent to Transceiver for RF Chain processing		Data message has undergone final signal processing and is ready for transmission
	5.	Send Over the Air	Data message sent to recipient(s)		Data message sent to WFA specific antenna and transmitted OTA

Post Conditions	• SDR reverts to its operational state awaiting its next action
Expected Outcomes	• The message has been transmitted, the system is in a steady state awaiting its next stimulus or "initiating action", and the client is free to take whatever next step is deemed appropriate. The act of sending a message will not have any impact on the WFA resource allocations or associated priorities.
Notes and Issues	• A very sophisticated WFA or management infrastructure implementation might attempt to reallocate resources between messages, but that is beyond the scope of the documented SCA capacity management approach.

Figure 15 contains the sequence diagram for the Send Data Message use case, and the subsections that follow provide additional information related to the interactions within the diagram.

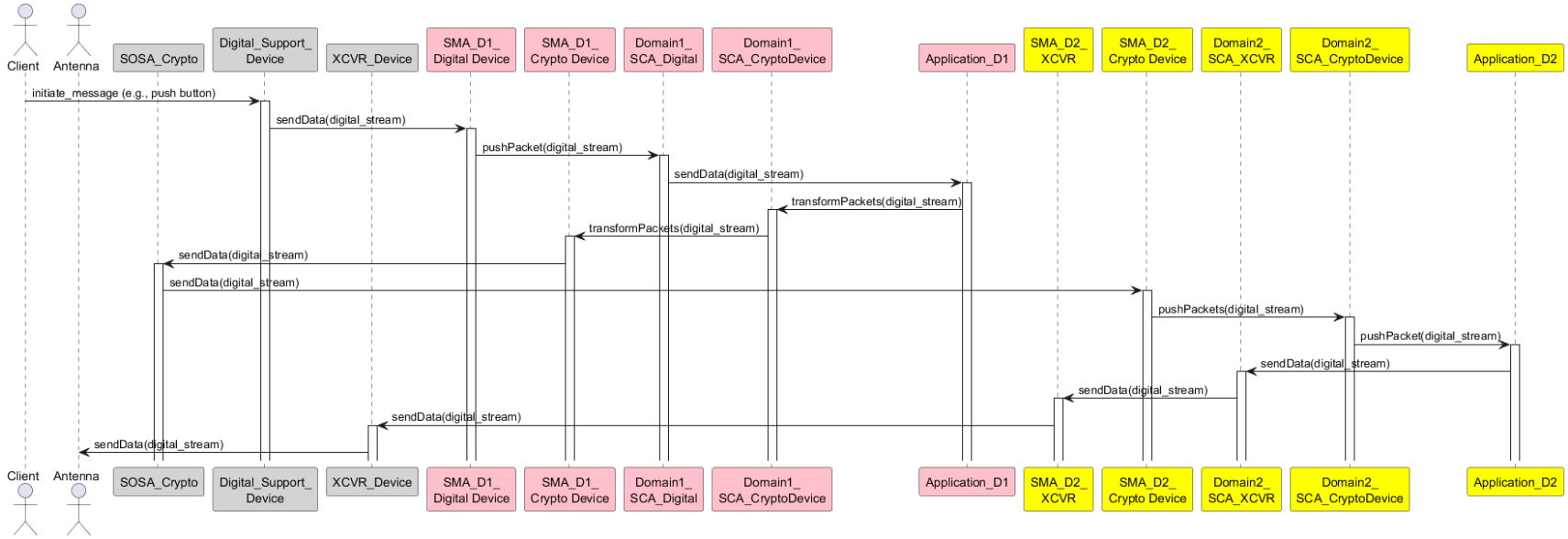

Figure 15 – Send Data Message Sequence Diagram

3.1.8.1 Send to Digital Device

This step is equivalent to the functions performed in Section 3.1.6.1 of the Send Voice message scenario with the exceptions being that the message is generated via a keypad, touch screen, or some other input mechanism and the system routing is directed to a system Digital Device.

3.1.8.2 Send to WFA (1)

This step is equivalent to the functions performed in Section 3.1.6.2 of the Send Voice message scenario with the exception being that a Digital Device, and its associated functionality, rather than an Audio Device is responsible for directing the data message to the WFA.

3.1.8.3 Send to WFA (2)

This step is equivalent to the functions performed in Section 3.1.6.3 of the Send Voice message scenario with the exception being that the message being transported is a data message.

3.1.8.4 Send to Transceiver

This step is equivalent to the functions performed in Section 3.1.6.4 of the Send Voice message scenario with the exception being that the message being transported is a data message.

3.1.8.5 Send Over the Air

This step is equivalent to the functions performed in Section 3.1.6.5 of the Send Voice message scenario with the exception being that the message being transported is a data message.

3.1.9 Receive Data Message

Use Case: Receive Data Message	
Objectives	• Allows the waveform application to receive a data message from an external communications device
Actors	• Client – human or NPE that interacts with the communications system • Antenna – physical device that receives the OTA WFA signal
Assumptions	• The network infrastructure to receive the OTA messages has been established • The necessary system infrastructure for the WFA to receive the message is operational • The WFA implementation includes all the necessary cryptographic elements for secure operations • The client has performed the WFA specific action to initiate the SDR communications capability
Pre-conditions	• The WFA has been instantiated, connected, and is in an operating state • The physical communications infrastructure is ready and able to receive signals

Initiating Event	• OTA communications signal arrives at the SDR antenna

Sequence of Events

	Event	Event	Input(s)	Outputs(s)/Outcomes(s)
1.	Send to Transceiver	Received OTA message routed to Transceiver for RF chain processing		Received signal has been forwarded to the SDR for RF chain processing
2.	Send to WFA (2)	OTA message transported to domain 2 portion of WFA		Received data message has been downconverted, demodulated, and transported to the WFA
3.	Send to WFA (1)	OTA message transported to domain 1 portion of WFA		Received data message has been decrypted, decoded, and otherwise processed by the WFA
4.	Send to Digital Device	OTA message sent to Digital Device for final conveyance		Data message has undergone final processing and is ready for delivery
5.	Send to User	OTA message undergoes final processing and is sent as text message to Client		Data message displayed on SDR device and can be viewed by Client

Post Conditions	• SDR reverts to its operational state awaiting its next action
Expected Outcomes	• The message received has been delivered by the system designated Digital device and hopefully viewed by the human system user. Similar to sending a data message, receipt of a message does not have any bearing on any future operations of the WFA or the SDR.
Notes and Issues	• The extent of what is included or affected as part of the message receipt process can vary on a WFA basis. This scenario represents a simplistic approach which aims to identify a model of the “major muscle movements” of what needs to occur.

Figure 16 contains the sequence diagram for the Receive Data Message use case, and the subsections that follow provide additional information related to the interactions within the diagram.

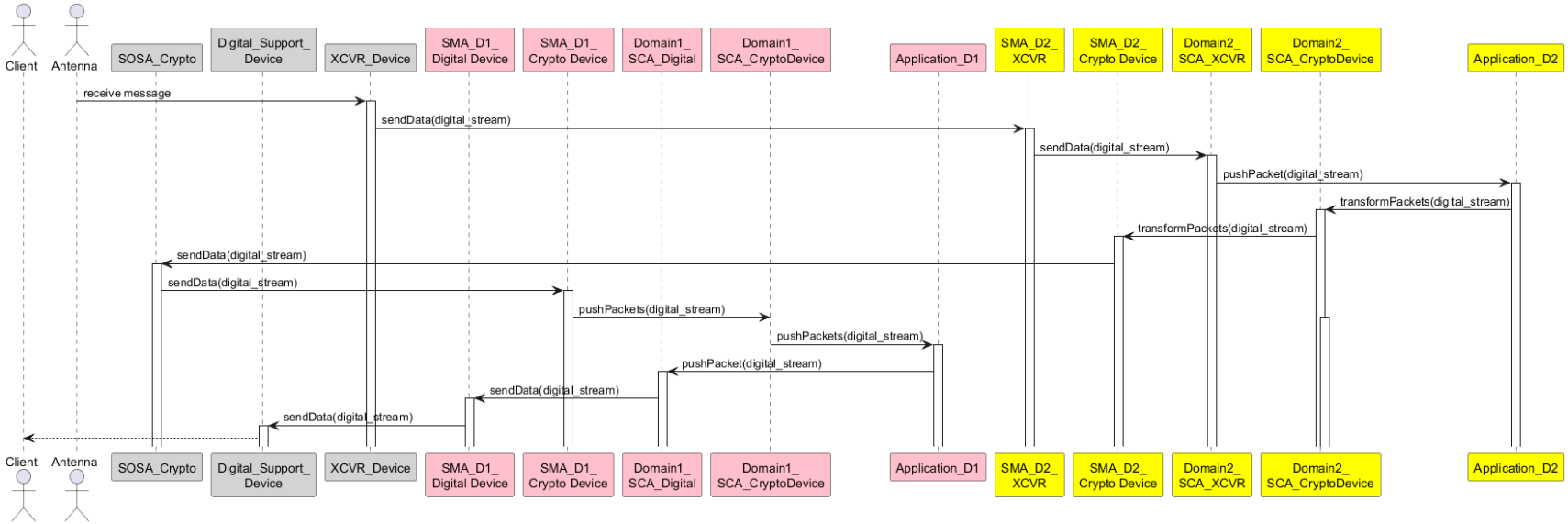

Figure 16 – Receive Data Message Sequence Diagram

3.1.9.1 Send to Transceiver

This step is equivalent to the functions performed in Section 3.1.7.0 of the Receive Voice message scenario with the exception being that the message being transported is a data message.

3.1.9.2 Send to WFA (2)

This step is equivalent to the functions performed in Section 3.1.7.2 of the Receive Voice message scenario with the exception being that the message being transported is a data message.

3.1.9.3 Send to WFA (1)

This step is equivalent to the functions performed in Section 3.1.7.3 of the Receive Voice message scenario with the exception being that the message being transported is a data message.

3.1.9.4 Send to Digital Device

This step is equivalent to the functions performed in Section 3.1.7.4 of the Receive Voice message scenario with the exception being that a Digital Device, and its associated functionality, rather than an Audio Device is responsible for directing the data message to the Client.

3.1.9.5 Send to User

This step is equivalent to the functions performed in Section 3.1.7.5 of the Receive Voice message scenario with the exception being that the message target is a screen, NPE Client, or some physical attribute of the SDR.

3.1.10 Disable WF Application

Use Case: Disable WF Application	
Objectives	• Terminates functional operation of the SCA card-based waveform application, but preserves the application instantiation
Actors	• Client – human or NPE that interacts with the communications system
Assumptions	• Disabling a WFA must be propagated to all domains of the operating WFA • The sensor platform will continue to function normally after a WFA has been disabled • Disabling a WFA does not destroy the SCA container or release system resource allocations
Pre-conditions	• SDR WFA and OE in an operational state • Dependent and collaborative SOSA Modules are operating and potentially providing service to additional SOSA modality instances • SMA has registered with the SOSA infrastructure

Initiating Event	• Client initiates an action to disable the WFA				
Sequence of Events		**Event**	**Event**	**Input(s)**	**Outputs(s)/Outcomes(s)**
	1.	Request Disable	Client Requests that WFA be disabled		Client message to disable WFA has been received by all of its components
	2.	Disable WFA	WFA disables itself		WFA has been transitioned to a non-operational state. However, the WFA remains instantiated, and the WFA container is still spun up
Post Conditions	• WFA is no longer operational • WFA container exists within the PIC • Task manager (and client) knows the WFA is not operational				
Expected Outcomes	• The WFA is no longer operational and all clients, human or NPE, have been notified that it is not providing service. In addition, the Task Manager also knows that the WFA is not providing service				
Notes and Issues	• Disabling a WFA should not be thought of as a pause. The primary distinction between the two concepts is that state or context is not preserved after a disable and if the WFA is returned to the service there are no prior activities that will resume.				

Figure 17 contains the sequence diagram for the Disable WF Application use case, and the subsections that follow provide additional information related to the interactions within the diagram.

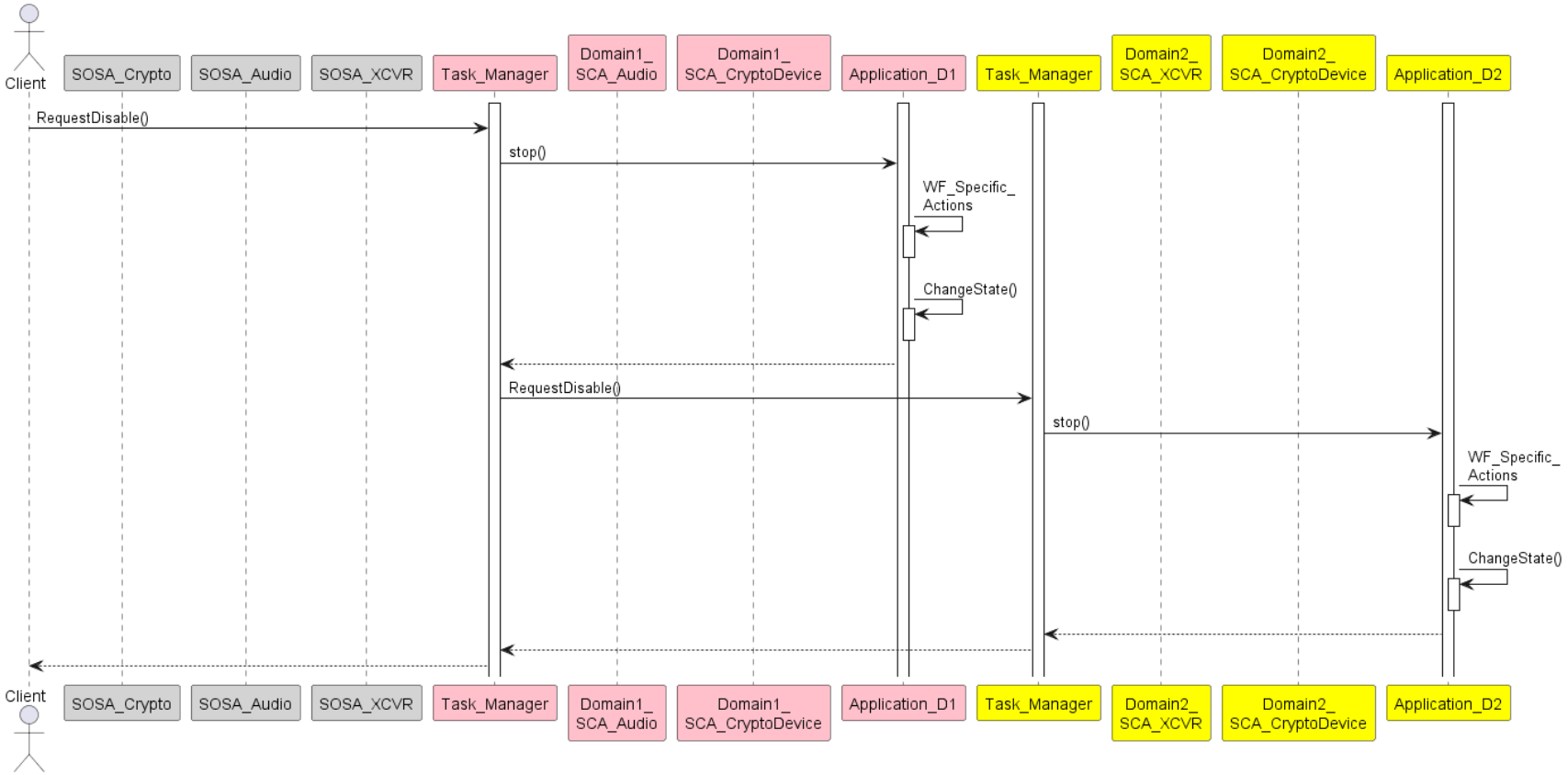

Figure 17 – Disable WF Application Sequence Diagram

3.1.10.1 Request Disable

The client initiates a "disable event" to transition a WFA from an operational state. It is important to note that this transition must be communicated to all domains where WFA components exist. A compliant SCA WFA can be designed and implemented in several approaches. The decision choice reflected in this sequence diagram assumes the existence of two WFAs, one for each domain, that constitutes this VULOS instance. While this design is not required, it was included to show the diversity of implementations that could exist within SCA and that could be accommodated by this interoperability approach. The disable request utilizes the communications paths established earlier in the WFA lifespan to have command routed to the WFA.

3.1.10.2 Disable WFA

Upon receipt of the client command, the WFA follows an implementation sequence of operations to transition the application. The SCA native commands are very general in nature, "should disable all current component operations and put it in a non-operating condition" and "should be ignored if the component is already in a non-operating condition", so it is challenging to dictate what needs to happen when the command is received. However, the WFA implementation might communicate the fact that it is not providing service to the Devices and Services that it collaborates with. Such communication would entail the requisite messages to be sent across the SCA ←→ SOSA boundary. If these notifications were communicated across the boundary it could result in a much more flexible system, as it could create that opportunity for other system resources to get a higher priority when accessing shared resources.

3.1.11 Deconstruct WF Application

Use Case: Deconstruct WF Application					
Objectives	• Destroys the container which hosts the based waveform application and deallocates its assigned system resources				
Actors	• Client – human or NPE that interacts with the communications system				
Assumptions	• Disabling a WFA must be propagated to all domains of the WFA • Once the destruction process begins, no additional requests will be accepted by the WFA • WFA must be Disabled prior to Deconstruction				
Pre-conditions	• WFA is in a non-operating state				
Initiating Event	• Client initiates an action to deconstruct the WFA				
Sequence of Events		**Event**	**Event**	**Input(s)**	**Outputs(s)/Outcomes(s)**
	1.	Request Deconstruct	Client Requests that WFA be torn down		Client message to deconstruct WFA has

					been received by all of its components
	2.	Deconstruct WFA	WFA removes all of its resources		All WFA resources within the container have been released and it has released its system Device and Service resource allocations
	3.	Remove WFA	WFA eliminates awareness of its existence from the system		WFA registration has been removed from the SOSA management modules and the WFA container has been destroyed
Post Conditions	• WFA no longer instantiated on the system (it may still be downloaded to the platform) • WFA must be reinstalled to be used again by a client • SOSA Devices and Services no longer hold resource reservations for the removed WFA				
Expected Outcomes	• WFA ceases operations • Clear the SCA WFA container • Remove any SOSA Management infrastructure elements created in tandem with the WFAs instantiation • Release all SOSA Module Resource allocations				
Notes and Issues	• Deconstructing a WFA is the only comprehensive method to remove it from existence and reclaim its associated resources, i.e., it is the culmination of the WFA lifecycle				

Figure 18 contains the sequence diagram for the Deconstruct WF Application use case, and the subsections that follow provide additional information related to the interactions within the diagram.

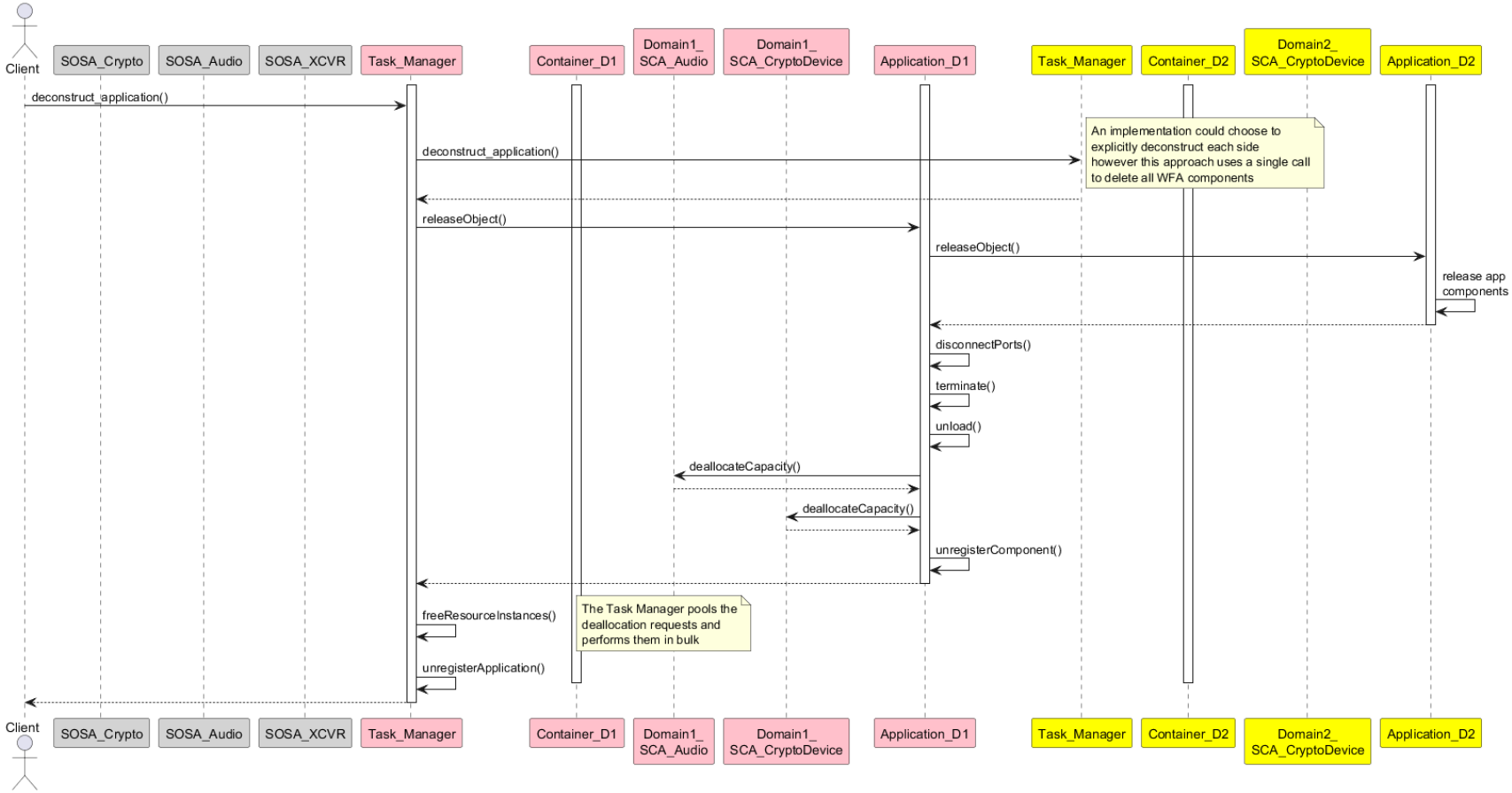

Figure 18 – Deconstruct WF Application Sequence Diagram

3.1.11.1 Request Deconstruct

When a WFA is deconstructed, the call must be distributed to all the application's components. Given that we have established that WFA components may exist in multiple domains, the initial call in this diagram is intended to show the implementation specific aspect of deconstruction. The system management infrastructure could send the deconstruct call to each domain explicitly and the deconstruction could proceed from that point. However, for the purpose of this paper, we have decided to have the call directed to a single control point within Domain 1 and allow that component to distribute that call to the remainder of the WFA components, irrespective of their location. In SCA terminology, the deconstruct call is directed to the WFA application controller, which is the entry point to the WFA implementation, which is accessed by the SCA CF Framework Control Components.

3.1.11.2 Deconstruct WFA

WFA deconstruction is an orderly process that occurs within the bounds of the SCA implementation. The application controller delegates the deconstruct call to the other WFA components regardless of their location. The destruction process for each of those second level components involved in delegating the deconstruct call to any of their dependent application components. The hierarchical process continues until the WFA leaf components are reached and the process of deconstruction, i.e., disconnecting ports, removing processes, and unloading modules can occur. After the leaf nodes are removed, the deconstruction process can pop up the stack until the deconstruction activities are finalized within the application controller. The sequence diagram associated with this scenario does not show any interactions with SOSA modules during the deconstruction process; however, there are a few instances where interactions across the boundary might occur. First, an implementation might decide to publish notifications as the deconstruct activities are being executed. Secondly, and something that will definitely occur, is that as the WFA is deconstructed, any system resources allocated or reserved by the WFA will be released. Consequently, when the WFA deallocates its SCA infrastructure components, that call will be relayed to the corresponding SOSA module of the SCA Device or Service and any allocated resources will be freed and returned to the system.

3.1.11.3 Remove WFA

Once the WFA has been removed and its consumed resources have been returned, a final set of activities can be performed depending on the implementation. The common activity that will occur is that the WFA will be removed from the SOSA management modules. The deconstruct implementation will unregister itself from its associated SOSA module, e.g., the Task Manager, and it will be the responsibility of that module to take any other action necessary to ensure that any of the other SOSA modules know the WFA in question is no longer available or accessible within the system. The other activity is that the communications card might take the opportunity to spin down the container which hosted that WFA that was just removed. Deconstructing the container is only advisable if there are no other WFAs or essential activities that are still executing within the container boundary.

4 Conclusion

The eleven scenarios explored within this provide, for the most part, a "sunny day" scenario of the interactions required to leverage SCA-based waveform applications within an operating platform populated by SOSA modules. The scenarios establish the foundation for how the applications may interoperate with one another but do not provide enough information for a development team to produce a robust, comprehensive design and implementation. The hope is that subsequent initiatives can expand upon this work to address more focused topics such as the following:

- SOSA systems allow for automatic component/module discovery which facilitates more of a plug and play capability. What interactions and extensions to the existing SCA-based WFAs would be required to provide that capability?

- What would be the specific design for the SMA or SMA sub agents? Would this capability have to be agent specific, or could elements such as the message protocol translations or exception handling capability be realized as reusable components?

- Investigating whether it would be beneficial to standardize the SMA or its sub agents. This activity could define common interfaces for the agents and establish an infrastructure or toolkit that would facilitate their development.

- Identifying if or how it would be appropriate to standardize this concept. Pursuing development of a standard would require multiple activities such as requirements elicitation, selecting a venue for standardization, and identifying conformance criteria.

- A more comprehensive analysis of System and Task Manager functions could be performed to determine the extent to which the SCA-based WFAs might require additional development to provide the functionality required by the managing modules.

- The SOSA ←→ SCA Interoperability Approach brief [14] provides a pathway for native SCA component to component communication across the SOSA Module / container boundary on the SMI. The need for this capability was not investigated as part of this effort and it was not incorporated within any of the scenarios. The likelihood, value, and implications of including this capability could be investigated to see if it should be included.

The hope is that the product provides a compelling enough case for development teams to consider using this approach as a method to provide SOSA with a "quick win" solution for a MOSA communications capability. Depending on the success of the approach, this interoperability approach might prove to be an enduring way to allow SCA-based WFAs to be leveraged within products that utilize SOSA module or a model for how other sensor frameworks or models could be integrated within a SOSA development approach.

5 References

5.1 Referenced documents

[1] Tri-Service Memorandum for Service Acquisition Executives and Program Executive Officers: "Modular Open Systems Approach for Department of Defense Weapon Systems," December 17, 2024
https://www.cto.mil/wp-content/uploads/2024/12/Tri-Service-Memo-Signed-17Dec2024.pdf

[2] International Radio Security Services API Specification, The Wireless Innovation Forum, WINNF-09-S-0011, Version V2.0.1 13 June 2013
https://winnf.memberclicks.net/assets/Coord_SCA_Standards-Portfolio/winnf-09-s-0011-v2.0.1%20int%20radio%20security%20services%20api.pdf

[3] Technical Standard for SOSA® Reference Architecture, The Open Group, Edition 2.0 (Snapshot 3), May 2025

[4] Software Communications Architecture Specification, Joint Tactical Networking Center, Version 4.1, 20 August 2015
https://www.jtnc.mil/Resources-Catalog/Resource-Catalog-Article-view/Article/2083253/comlplets-sca-41-release/

[5] Runtime Spec Principles, The Open Containers Initiative, April 2024
https://specs.opencontainers.org/runtime-spec/principles/?v=v1.0.2

[6] SPD-driven Smart Transmission Layer Based on a Software Defined Radio Test Bed Architecture, Kresimir Dabcevic, Lucio Marcenaro, and Carlo S. Regazzoni, 30 September 2014

[7] Microwave and RF Design, M. Steer - Scitech Publ., Inc., N.C., 2010, also from Yes Dee Publ., India, 2016

[8] Modular Open RF Architecture (MORA) Specification, Version 2.5, U.S. Army Combat Capabilities Development Command (CCDC) C5ISR Center, December 2022

[9] Vehicular Integration for C4ISR/EW Interoperability (VICTORY) Standard Specifications, Version 1.11, TBD
https://jtnc.experience.crmforce.mil/JTNC/s/cmoss

[10] Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[11] JTNC Standards API Release 1.1, Joint Tactical Networking Center, Version 1.1, 20 August 2015
https://www.jtnc.mil/Resources-Catalog/Resource-Catalog-Article-View/Article/2084586/complete-jtnc-standards-api-release-11/

[12] Transceiver Facility PIM Specification, The Wireless Innovation Forum, WINNF-TS-0008, Version V2.1.1, 22 January 2022
https://winnf.memberclicks.net/assets/work_products/Specifications/WINNF-TS-0008-V2.1.1.pdf

[13] Interface Definition Language, Object Management Group, formal/18-01-05, Version 4.2, March 2018
https://www.omg.org/spec/IDL/4.2/PDF

[14] Strategy Proposal for SOSA™ SCA Interoperability, Joint Tactical Networking Center, Version 1.80, 19 September 2022
https://www.jtnc.mil/Resources-Catalog/Resource-Catalog-Article-View/Article/4207777/strategy-proposal-for-sosa-sca-interoperability/

The provided URLs were successfully accessed at the release date of the document.

6 Acronyms list

Acronym	Description
AM	Amplitude Modulation
ANT	Antenna
API	Application Programming Interface
BIT	Built in Test
bps	Bits per second
CF	Core Framework
CODEC	Coder-Decoder
Comms	Communications
CONOPS	Concept of Operations
CORBA	Common Object Request Broker Architecture
CSS	Cryptographic Subsystem
DARE	Data at Rest Encryption
DEWS	Directed Energy Weapon Systems
DoD	Department of Defense
EO/IR	Electro-Optical/Infra-Red
EW	Electronic Warfare
FM	Frequency Modulation
FPGA	Field Programmable Gate Array
GIOP	General Inter-ORB Protocol
GPP	General Purpose Processor
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
IIOP	Internet Inter-ORB Protocol
JSON	JavaScript Object Notation
JTNC	Joint Tactial Networking Center
JTRS	Joint Tactical Radio System
kHz	Kilohertz
MHz	Megahertz
MORA	Modular Open RF Architecture
MOSA	Modular Open Systems Approach
NPE	Non-Person Entity
OAS	OpenAPI Specification
OCI	Open Container Initiative
OE	Operating Environment
ORB	Object Request Broker
OSA	Open Systems Architecture
OTA	Over the Air
PIC	Plug in Card
PNT	Position Navigation and Timing
POST	Power On Self Test
PSM	Platform Specific Model
RCD	RF Conditioning and Distribution Device
REST	Representational State Transfer
RF	Radio Frequency
RFD	RF Distribution
RTE	Run Time Environment

SCA	Software Communications Architecture
SDM	SOSA Data Message
SDR	Software Defined Radio
SIGINT	Signals Intelligence
SMA	SOSA Module Agent
SMI	SOSA Message Interconnect
SOSA®	Sensor Open Systems Architecture
SWLLI	SOSA Wideband Low Latency Interconnect
SYSLOG	System Logging Protocol
TCP	Transmission Control Protocol
TCS	Tactical Communications Standards
UDP	User Datagram Protocol
UHF	Ultra-High Frequency
VHF	Very High Frequency
VICTORY	Vehicular Integration for C4ISR/EW Interoperability
VULOS	Very High Frequency/Ultra High Frequency Line of Sight
WF	Waveform
WFA	Waveform Application
WInnF	Wireless Innovation Forum
XML	Extensible Markup Language

Table 2 Acronyms list

END OF THE DOCUMENT