CONCEPT OF OPERATIONS

FOR THE

SWIM INTER-REGISTRY FRAMEWORK (SIRF)

Version 1.0.0

U.S. FEDERAL AVIATION ADMINISTRATION

SYSTEM WIDE INFORMATION MANAGEMENT (SWIM)

OCTOBER 11, 2018

Preface

The objective of the proposed System Wide Information Management (SWIM) Inter-Registry Framework (SIRF) is to enable discovery of SWIM services across independently developed and managed registries. This Concept of Operations (ConOps) document outlines an approach for developing a virtual network of interoperable service registries to facilitate the exchange of service-related information among national and international SWIM communities.

The SIRF draws upon well-established works published by various industry standards bodies, as well as research conducted by SWIM-implementing programs. It presents recommendations for the conditions and engineering approaches needed to enable information exchange among independently developed and autonomously operated SWIM registries. The SIRF is flexible enough to adapt to current SWIM organizational and technological constraints, as well as to requirements that are expected to emerge in the future.

Table of Contents

List of Figures

List of Tables

1 Scope

1.1 Identification

This document is identified as "Concept of Operations for the SWIM Inter-Registry Framework (SIRF)."

1.2 Purpose

This Concept of Operations (ConOps) document provides a conceptual overview of the proposed SWIM Inter-Registry Framework (SIRF).

The purpose of this document is:

1) To provide a clear vision of the intended use and resulting benefits of the SIRF;
2) To build consensus among sponsors, decision makers, stakeholders, and integrators participating in SIRF-based projects;
3) To initiate the effort of defining the requirements and architecture for the future development of the SIRF.

The format of this document is consistent with the outline of a concept of operations document defined in the Institute of Electrical and Electronics Engineers (IEEE) Standard 29148:2011 [1].

This document contains the following sections:

Section 1 - Provides an overview of this document and describes the general nature and concept of the framework to which this document applies.
Section 2 - Lists the references mentioned in this document.
Section 3 - Describes the current situation and the issues that led to the proposed SIRF.
Section 4 - Justifies the proposed SIRF based on the most current information available.
Section 5 - Describes and discusses the concepts of the proposed SIRF.
Section 6 - Describes various operational scenarios.
Section 7 - Summarizes operational, organizational, and other impacts of implementing the proposed framework.

1.3 Overview

The civil aviation community has embraced System Wide Information Management (SWIM) as a platform for effective information sharing. SWIM has changed the way information is provided and managed by Air Traffic Management (ATM) stakeholders by making available a wide range of information resources available through a network of reusable and shared services [2].

In early adaptations (EUROCONTROL/SESAR, FAA/NEXTGEN), SWIM has been implemented as a self-contained, independently governed collection of infrastructures and services within boundaries that represent a state or a state-sponsored organizational entity. Successful demonstration by the SWIM implementations of the ability to share information has led to increasing interest in the international community, and other countries, regions, and international entities have begun to adopt and further evolve the SWIM paradigm [11] [12] [13].

One of the critical factors in the overall usability of services is *service discoverability*, that is, the ability of a service to be located through a uniformly interpretable set of service metadata that is accessed by a service consumer through some form of retrieval mechanism (e.g., a service registry)

[14]. Because the concept and architecture of a service registry is already described in many service-oriented architecture (SOA) and SWIM-specific works, in this ConOps we will only offer a high-level depiction of a registry in a service-centric environment (see Figure 1).

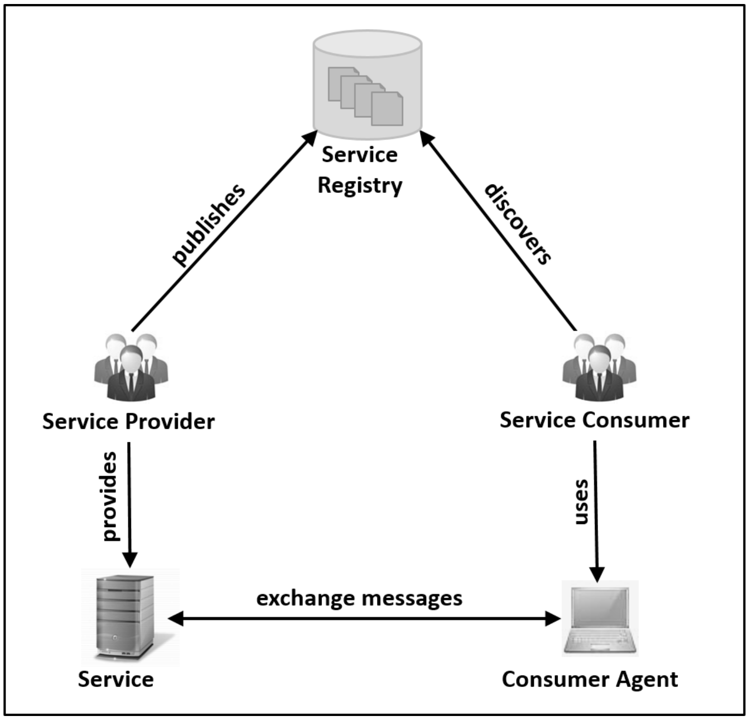


Figure 1 Service registry in the context of SOA

A proliferation of SWIM implementations by international communities has introduced new requirements for service discoverability. These requirements call for a registry to make service meta-information available not just to its human users but also to other registries or discovery mechanisms affiliated with other agencies or states. In essence, *a global SWIM service discovery capability* is a needed and necessary precondition for achieving *global information exchange* among SWIM-enabled services.

2 Referenced Documents

[1] ISO/IEC/IEEE 29148:2011(E) - ISO/IEC/IEEE International Standard - Systems and software engineering - Life cycle processes -Requirements engineering, IEEE Computer Society, December 1, 2012
http://standards.ieee.org/findstds/standard/29148-2011.html

[2] SWIM Common Registry: Concept, Architecture, and Implementation; FAA/SESAR; Pedro Fernandez-Sancho, Mark Kaplun, Eric Roelants, Carol Uri; June 2014
https://www.faa.gov/air_traffic/technology/swim/governance/outreach/media/SWIM%20Common%20Registry%20Concept%20Architecture%20and%20Implementation.pdf

[3] SWIM Controlled Vocabulary, FAA, May 2013
https://www.faa.gov/air_traffic/technology/swim/vocabulary/

[4] OASIS Reference Model for SOA 1.0, 12 October 2006
http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

[5] Service Description Conceptual Model (SDCM) 2.0; SESAR CP 2.1; June 3, 2016
http://swim.aero/sdcm/2.0.0/sdcm-2.0.0.html

[6] Concept of Operations for the SWIM Common Registry (SCR), SJU CP 2.1, April 2015
https://www.faa.gov/air_traffic/technology/swim/governance/international_collaboration/media/SCR%20CONOPS%20FINAL%20DRAFT%2004042015.pdf

[7] Web Services Architecture; W3C; Working Group Note; 11 February 2004
http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[8] UDDI Version 3.0.2, UDDI Spec Technical Committee Draft, 20041019
http://www.uddi.org/pubs/uddi-v3.0.2-20041019.htm

[9] IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries, IEEE, 1990

[10] The Levels of Conceptual Interoperability Model: Applying Systems Engineering Principles to M&S; Wenguang Wang, Andreas Tolk, and Weiping Wang; In Proceedings of the 2009 Spring Simulation Multiconference (SpringSim '09). Society for Computer Simulation International, San Diego, CA, USA; 2009

[11] The Research Synopsis about SWIM in China; Zhao, Milong, et al; Proceedings of the 2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems. IEEE Computer Society, 2015.

[12] Long-term Vision for the Future Air Traffic Systems (CARATS), Japan Civil Aviation Bureau, 2010

[13] A SWIM Architecture Proposal for Brazilian Scenarios; Li Weigang, José Alexandre Tavares Guerreiro Fregnani, Italo Romani de Oliveira, Glaucia Balvedi; Rio de Janeiro SITRAER 2017, Conference Paper; October 2017

[14] Aviation Domain: Quality and Discovery; Open Geospatial Consortium (OGC), Testbed 13; 2017
http://www.opengeospatial.org/pub/Testbed13/t14.html

[15] Net-Centric Service Framework, Version 2.1; Robert Palmqvist (Saab), Lars Schylberg (Saab), Allen Jones (Boeing), Jerry Sonnenberg (Harris); Network-Centric Operations Industry Consortium (NCOIC(TM)); 2009

[16] Architecture Constraints for Interoperability and Composability in a Smart Grid; Andreas Tolk; IEEE Xplore, Conference Paper; August 2010

[17] Registry Integration Module (RIM) 1.0; SESAR CP 2.1; October 2017
http://swim.aero/rim/1.0.0/rim-spec-1.0.0.html

[18] Service Description Model for XML (SDM-X), version 1.0; FAA/SESAR, October 2017
http://swim.aero/sdm-x/1.0.0

[19] semantics.aero, Collection of Semantic Artifacts; last modified at 2018-01-05
http://semantics.aero

[20] RFC 5246, The Transport Layer Security (TLS) Protocol, Version 1.2; Network Working Group; August 2008
https://tools.ietf.org/html/rfc5246

[21] IETF RFC-3986, Uniform Resource Identifier (URI): Generic Syntax; T. Berners-Lee, R. Fielding, L. Masinter, January 2005
https://tools.ietf.org/html/rfc3986

[22] Extensible Markup Language (XML),1.0; W3C Recommendation; November 2008
https://www.w3.org/TR/xml/

[23] NATO Interoperability Standards and Profiles, Verion 1.0; AdatP-34(J); October 2017
https://nhqc3s.hq.nato.int/Apps/Architecture/NISP/

[24] Levels of Information Systems Interoperability (LISI);DOD AWG; March 1998
http://web.cse.msstate.edu/~hamilton/C4ISR/LISI.pdf

[25] European Interoperability Framework for pan-European eGovernment Services; IDA Working Document; Version 4.2; January 2004
http://ec.europa.eu/idabc/servlets/Doc4ee4.pdf?id=18060

[26] RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1; R. Fielding et al; IETF, Network Working Group; June 1999
https://www.ietf.org/rfc/rfc2616.txt

3 Current Situation

The current situation on which the proposed changes are based can be envisioned as a combination of two parts: 1) the current set of operational registries, and 2) the registries that are either being developed or expected to be developed in the foreseeable future. Naturally, there are some assumptions made with respect to the projected environments and associated registries; however, these assumptions are based on extensive experience obtained while developing and operating existing SWIM registries.

3.1 Background, Objectives, and Scope

Currently there are two functioning implementations of SWIM: EUROCONTROL's and FAA's. Each has been historically established and operated as a self-contained, independently governed collection of services within boundaries that represent a state-sponsored organizational entity.

To support service discovery in the context of their respective service inventories, each of these SWIM implementations has also established its own registry: the FAA NAS Service

Registry/Repository (NSRR) [3] at https://nsrr.faa.gov/ and the EUROCONTROL European SWIM Registry at https://eur-registry.swim.aero/.

Since their original inception, both SWIM programs started and continue to work on various aspects of interoperability and integration between the two systems, including collaboration in the area of service discovery. Noteworthy results from these efforts are:

- Service Description Conceptual Model (SDCM) [5] – a graphical and lexical representation of the properties, structure, and interrelationships of all service metadata elements, collectively known as a service description. SDCM is the architectural foundation for a service description as well as a structured representation of service metadata in the context of a SWIM registry. Later, an XML exchange model called the Service Description Model for XML (SDM-X) [18] was derived from the second version of SDCM.
- SWIM Common Registry (SCR) Concept of Operations [6] – a high level document that describes an architectural approach for the integration of two SWIM registries: FAA and EUROCONTROL.[1] Subsequent prototyping of this concept demonstrated that implementation of SCR would require a number of changes to the way both registries operated.
- Registry Integration Module (RIM) [17] – a specification that defines an interface for machine-to-machine interactions among SWIM registries. The RIM is de facto a proof of concept for the architectural approaches articulated in this ConOps. The swim.aero/rim site maintains the most current versions of artifacts used in the context of RIM.
- Common Taxonomies – a set of semantic artifacts developed a) to support semantic mediation among SWIM registries, and b) to manage a set of Semantic Web artifacts capable of supporting service discovery. These artifacts are currently maintained at semantics.aero [19].

There are also emerging implementations of SWIM registries by other state-sponsored aviation agencies, such as Luftfartsverket (LFV)[2] and NAV CANADA[3]. (For the purpose of this document, we consider these instances as parts of the “current” system or situation discussed in Section 3).

All of these existing and planned registries share the same information domain (i.e., SWIM services meta-information) and are very similar in terms of basic functionalities. For example:

- They register only services affiliated with their sponsoring organizations.
- They are registries/repositories, that is, they maintain not only services metadata, but also artifacts and documentation related to the services.
- They are used to facilitate SWIM governance policies and procedures throughout the service life cycle.

---

[1] The work on SDCM and SCR originally contributed to Single European Sky ATM Research Programme’s (SESAR) organization, a subcomponent of EUROCONTROL.

[2] Swedish Civil Aviation Administration

[3] Nav Canada is a privately run, not-for-profit corporation that owns and operates Canada's civil air navigation system.

However, because these registries were designed independently from each other and under different sets of organizational and architectural constraints, information stored in them cannot easily be integrated, correlated, or compared.

Currently, if a user wishes to locate services in different registries that match a specific set of criteria, the user must:

a) Log into a registry,
b) Search for a service using the set of criteria,
c) Log into another registry (usually using another set of credentials),
d) Search for the service using the same criteria,
e) Repeat steps a) through d) as many times as there are available registries,
f) Correlate results received from all queries.

Figure 2 illustrates this scenario.

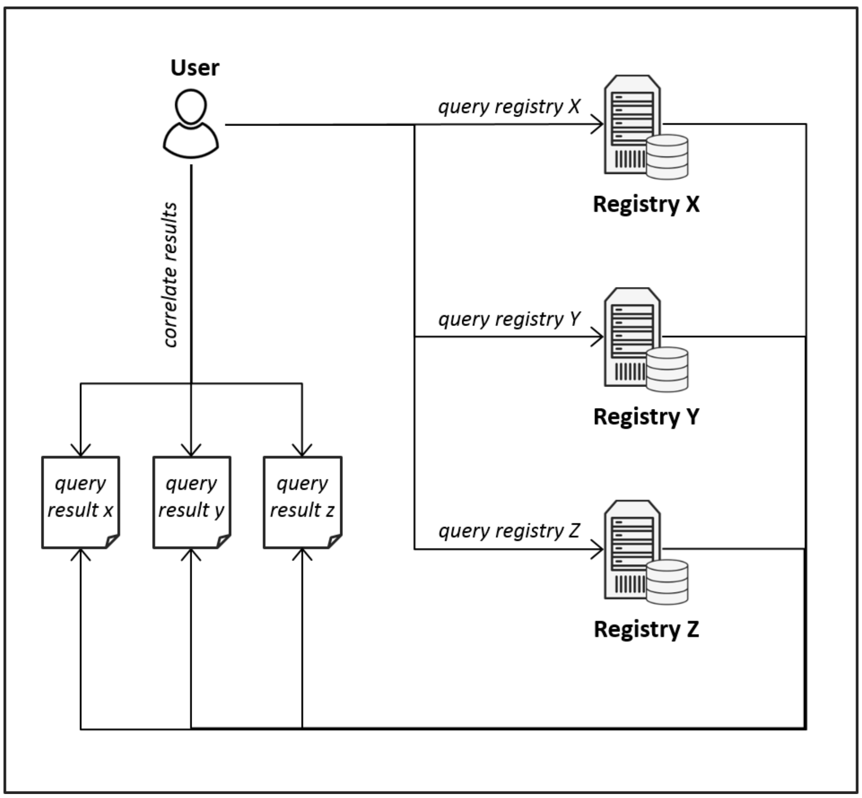


Figure 2 Search in multiple registries (current situation)

It is easy to see that this process is logistically complex and tends to produce inconsistent results. From a user's perspective, the most effective solution is having a mechanism capable of sequencing or simultaneously making several calls to registries and packaging the returned responses into a single semantically consistent result.

4 Justification for and Nature of Changes

As explained above, each SWIM registry essentially operates as a *disparate system* without exchanging data or interacting with other registries. All SWIM registries have been or will be developed independently and under different sets of organizational and technological constraints. And although all registries contain information specific to the same domain – service metadata of SWIM-enabled services – this information, if not harmonized, cannot always be correlated and integrated easily. In other words, the SWIM registries are unable to *interoperate*, which precludes SWIM stakeholders from discovering services in the context of "global" SWIM.

4.1 Description of Desired Changes

To achieve global SWIM objectives, all SWIM registries must work together, i.e., they must be *interoperable*. At the most general level, *interoperability* is understood as the ability of two or more systems or components to exchange information and to use the information that has been exchanged [9].

When assessing the interoperability requirements for a SWIM registry, several levels of interoperability must be considered, including technical, semantical, and data exchange. Each of these levels imposes specific requirements that need to be addressed by a service-oriented integration solution. Typically, these requirements are addressed by conforming all interacting components to a shared set of standards or architectural solutions.

However, making all participating registries interoperable by adopting a shared set of standards is necessary but not sufficient to reach the SIRF's objective. The registries also need to be "assembled" in a way that will make them appear to a user as a single application capable of spanning multiple registries and returning a consolidated result.[4]

It should be pointed out that the SIRF should not require changing the ways in which the participating registries are operated and governed. All participating registries should be able to continue to function independently from each other while supporting their respective organization-specific responsibilities and practices, as well as continue to function regardless of whether other participating registries are available or accessible.

In summary, the SIRF proposes that the following changes be applied to the current environment:

a) All registries should adopt a common set of standards to be able to interoperate with each other.
b) A technological solution by which all participating registries perform as components of a single composite should be implemented by all registries.

[4] An assembly of components and references designed and deployed together in a single application is referred to by system engineers as a *composite*. A design principle of developing services out of independently existing services -- or in our case, registries -- is known within the service-oriented architectural paradigm as *service composition*, described in more detail in section 5.1.3.

c) Assembling registries into a single application should not preclude them from functioning autonomously and should not change the ways each registry currently operates or is designed to operate in the future.

The concepts of interoperability and service composition are discussed in more detail in Section 5.

4.2 Changes Considered but Not Included

a) *Semantic mediation* is the process of relating multiple organization-specific vocabularies and taxonomies to support information consolidation. It allows registry developers to share taxonomies and vocabularies while retaining control of organization-specific implementations. While some successful prototyping of semantic meditation has taken place, a detailed description of this subject is deferred to future SIRF documentation.

b) Considering the small number of current SWIM registries, a discussion about creating and distributing a list of registries that participate in inter-registry exchanges is deferred to future SIRF documentation.

c) While this ConOps does not prescribe particular standards for encryption, authentication or authorization, we expect that all participating registries will conform to recognized security standards and enforce security policies based on their organizational needs. A successful prototyping of the SIRF approach has demonstrated that Transport Layer Security (TLS) protocol [20] and HTTP Authentication [21] provides a sufficient level of security for a SIRF-compliant setting.

5 Concepts for the Proposed Framework

5.1 Key Concepts

For better understanding of the proposed framework, some basic concepts should be examined and further discussed.

5.1.1 Service Registry

A service registry is the main topic of this ConOps. The concept of a registry is well defined and sufficiently described in many SOA works, e.g., [4].

At the most general level, a registry is a mechanism for making information about services available [2]. However, at an implementation level, a registry can be realized in more than one way. To avoid confusion that may arise from using an overloaded term, we are extending the definition of a registry in two special cases:

1) A registry is an application that exposes its functionality through an interface designed explicitly for interaction with a human agent, usually in the form of a graphical user interface (GUI). Current SWIM registries in FAA and EUROCONTROL were originally implemented as this type of registry. Henceforth, this type of registry will be referred to as a *registry-application*.

2) A registry is a service designed to support interoperable machine-to-machine interaction over a network. It has a well-defined application programming interface (API). Other systems interact with the registry by exchanging messages typically conveyed using an XML serialization [7]. This vision of a registry as a service was effectively described in a number of well-recognized works; for example, the Web Service Architecture document [7] describes the concept of "Discovery Service" as being "a service that enables agents to retrieve Web service-related resource descriptions." We will refer to this type as a *registry-service*.

Summarizing: to adequately describe the SIRF's architecture we define two subtypes of the concept registry: a *registry-application* and a *registry-service*. The discriminating factor for these two concepts is the type of user; a registry-application is designed to be accessed by human agents, while a registry-service is accessed by other registries. Appropriately, the former exposes a GUI, and the latter defines an application-programming interface (API) for communicating with software agents.

Figure 3 describes the concepts presented in this section.

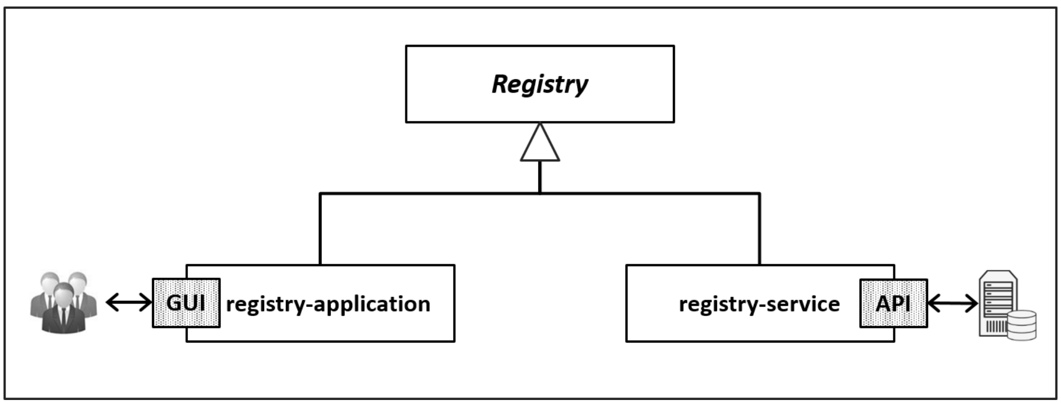


Figure 3 Subtyping of the concept "registry" in the SIRF

In the remainder of this ConOps, we will be using the terms "registry" and "registry-service" interchangeably. We will always use the term "registry-application" where appropriate to avoid potential ambiguity.

5.1.2 Interoperability

Interoperability is the ability of two or more systems or components to exchange information and to use the information that has been exchanged [9]. Interoperability is generally described as a model constructed of multiple levels with each level representing a specific aspect of interoperability. There are several interoperability levels models in technical and information domain, e.g., NATO Interoperability Model [23], Levels of Information Systems Interoperability (LISI) [24], and European Interoperability Framework (EIF) [25]. In this ConOps document, we follow the Levels of Conceptual Interoperability Model (LCIM) [10]. The LCIM has been designed for evaluating and managing interoperation and composability tasks for complex Modeling and Simulation (M&S) systems.

Because the objectives of SIRF are different from M&S systems, we use only the first five of the seven levels[5] described by LCIM, as depicted in Figure 4.

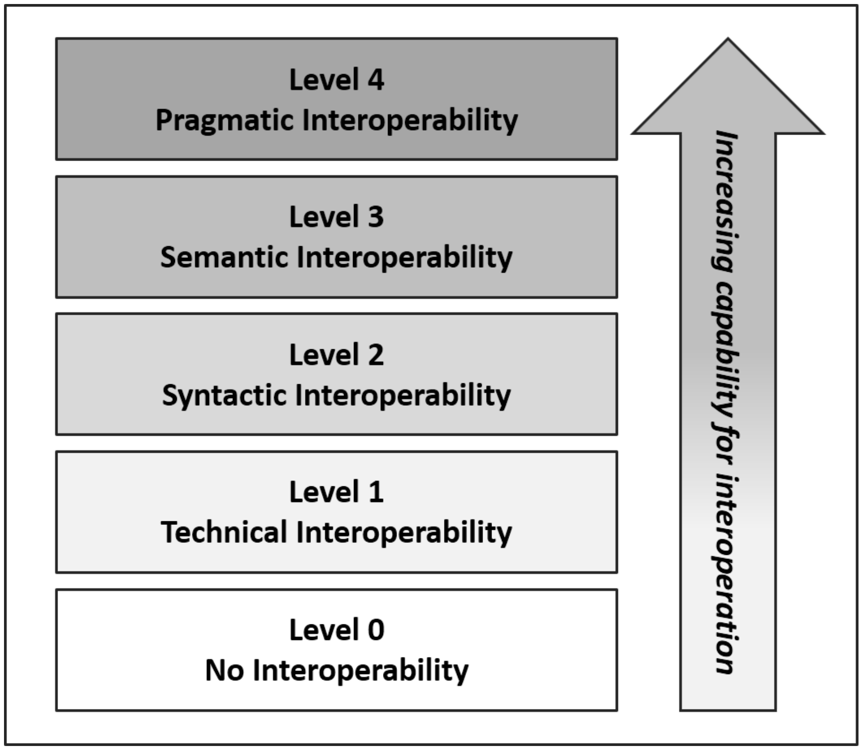


Figure 4 SIRF Levels of Interoperability

Table 1 provides descriptions for each level (from lowest to highest) [10], [16].

Table 1 Descriptions of interoperability levels

Level 0 No Interoperability Stand-alone systems have no interoperability.
Level 1 Technical Interoperability This level deals with infrastructure and network challenges, enabling systems to exchange bits and bytes. At this level, a communication protocol for exchanging data between participating systems exists, a communication infrastructure and a network protocol is established.
Level 2 Syntactic Interoperability This level introduces a common structure to exchange information; i.e., a common data format. On this level, there is an agreed protocol to exchange the right forms of data, but the meaning of data elements is not established.

[5] The other levels are Dynamic Interoperability ("Systems are able to comprehend the state changes that occur in the assumptions and constraints that each is making over time, and they are able to take advantage of those changes") and Conceptual Interoperability ("Interoperating systems are completely aware of each other's information, processes, contexts, and modeling assumptions").

Level 3 Semantic Interoperability This level provides means to capture a common understanding of the information to be exchanged, often in the form of a model, common vocabulary, or ontology. This level is reached when a meaning of the data is shared.
Level 4 Pragmatic Interoperability This level recognizes the patterns in which data are organized for the information exchange, which are in particular the inputs and outputs, often in form of messages, replication patterns, etc.

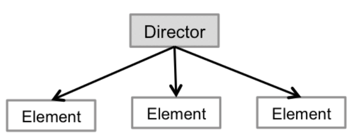
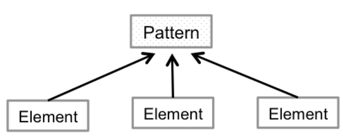
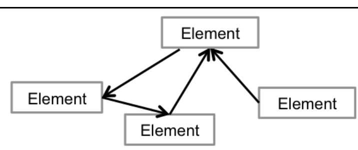
Similar to the LCIM, all levels are mutually supportive from the bottom up. When higher levels are reached, the lower levels must have been satisfied.

5.1.3 Service Composition

One of the main benefits of a SOA is the ability to compose applications, processes, or more complex services from less complex services. This activity, often called *service composition*, allows developers to compose applications and processes using services from heterogeneous environments without regard to details and differences of those environments [15].

A composition can be realized using three common patterns or styles: orchestration, choreography, and collaboration.

Table 2 Composition patterns

	*Orchestration* is a type of composition where one particular element, referred to as Director, is used by the composition to oversee and direct the other elements.
	*Choreography* is a type of composition whose elements interact in a non-directed fashion with each autonomous part knowing and following an observable predefined pattern of behavior for the entire (global) composition.
	*Collaboration* is a type of composition whose elements interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior.

Each of these patterns can be used for creating a *registry-services composite*, a final product of a SIRF implementation. This will be demonstrated in the next section.

5.2 Proposed Approach

The proposed approach, SIRF, consists of two sequential steps: a) asserting a state of interoperability among *registry-services* and b) executing a shared service composition pattern.

5.2.1 Assertion of SIRF Interoperability

Interoperability typically is enabled by establishing standard protocols and data and information models to which all connected components must adhere [16].

Table 3 presents a set of artifacts (i.e., standards, protocols, models) that support interoperability along with associated layers.

Table 3 Standards and protocols used for asserting interoperability in SIRF

Interoperability Level	***Artifacts required to achieve interoperability at this level***	***Note***
Level 1 ***Technical Interoperability***	TCP/IP, Public Internet, HTTP [26]	Each participating registry should be accessible from a public Internet.
Level 2 ***Syntactic Interoperability***	XML [22], URI [21], SDM-X [18]	All registries should support the set of ubiquitous Web protocols. A data structure for information exchange is defined by the SDM-X schema [18].
Level 3 ***Semantic Interoperability***	SDCM [5], Common ontologies/taxonomies Taxonomies: SWIM Service Product [19] Service Availability Status [19] Service Interface Type [19]	The meaning (semantics) of all information elements exchanged by participating registries is described in the SDCM [5]. The set of common taxonomies defines shared criteria for search across all registries.
Level 4 ***Pragmatic Interoperability***	RIM Specification [17]	The RIM Specification describes the patterns in which messages are exchanged, as well as inputs and outputs for a registry pluggable model built according to principles outlined in the SIRF.

5.2.2 Point-to-Point (P2P) Service Discovery Composition Pattern

The SIRF adopts a *P2P Service Discovery* design approach for implementing a *registry-service* composite.

As the name suggests, the P2P service discovery design pattern has coalesced from a well-known networking architecture: Peer-to-peer (P2P). The notion of applying P2P to service discovery was originally described by W3C in its Web Service Architecture [7].

A P2P distributed application architecture describes a network where every node ("peer") is an equally privileged, equipotent participant, that is, has the same capabilities and responsibilities.

Following the P2P architectural vision, P2P service discovery is designed around the notion of equal peer-registries (registry-services), with each registry being capable of simultaneously functioning as both a "client" and a "server" to the other nodes on the network [7].

This design leads to a solution where a registry-services receiving a request not only generates and returns a response, but also sends the request to one or more different registries, receives responses from these registries, and subsequently adds these responses to its own response and returns the consolidated response to the originator of the request.

As Figure 5 shows, a user sends a request to Registry X. Registry X generates a response to the user's request and at the same time sends the same request to Registry Y and Registry Z. After receiving responses from Registries Y and Z, Registry X combines all responses and returns a consolidated response to the user.

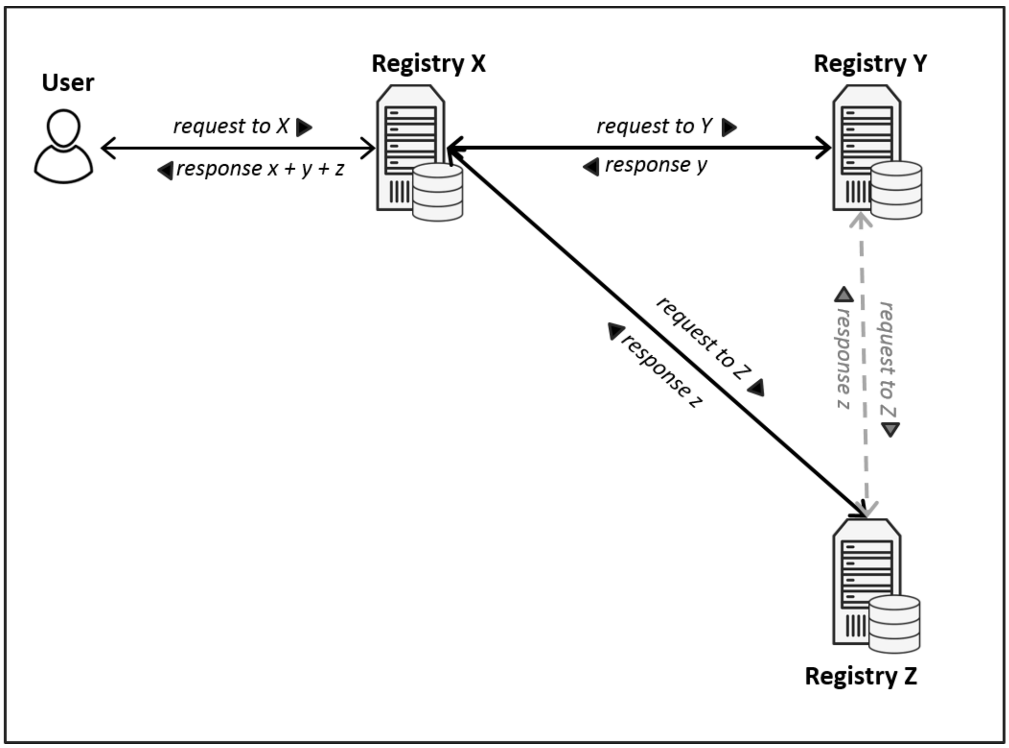


Figure 5 P2P Service Discovery model

It should also be noted that currently the SIRF does not prescribe any path for subsequent requests. As Figure 5 depicts, Registry X can invoke Registry Y, and Registry Y in its turn can "ask" Registry Z.

The P2P discovery model implements a *choreography* type of composition, that is, participating registry-services interact in a non-directed fashion with each autonomous part knowing and following an observable predefined pattern of behavior.

5.3 Analysis of Alternative Approaches

To evaluate the effectiveness of the proposed P2P discovery approach, we will review how other realizations of composition patterns could be used to retrieve information collected by multiple registries while accessing a single registry.

5.3.1 Replicating Registries

The *replicating registries* approach describes the way for a registry-services to publish its content, in whole or in part, in another registry.

The concept of replicating registries was developed and fully formalized by the Organization for the Advancement of Structured Information Standards (OASIS) and specified in the Universal Description, Discovery, and Integration (UDDI) specification [8].

In the replicating registries design, participating registries, also referred to in the UDDI API as *affiliated registries*[6], import and export their respective contents to and from each other. This is illustrated in Figure 6.

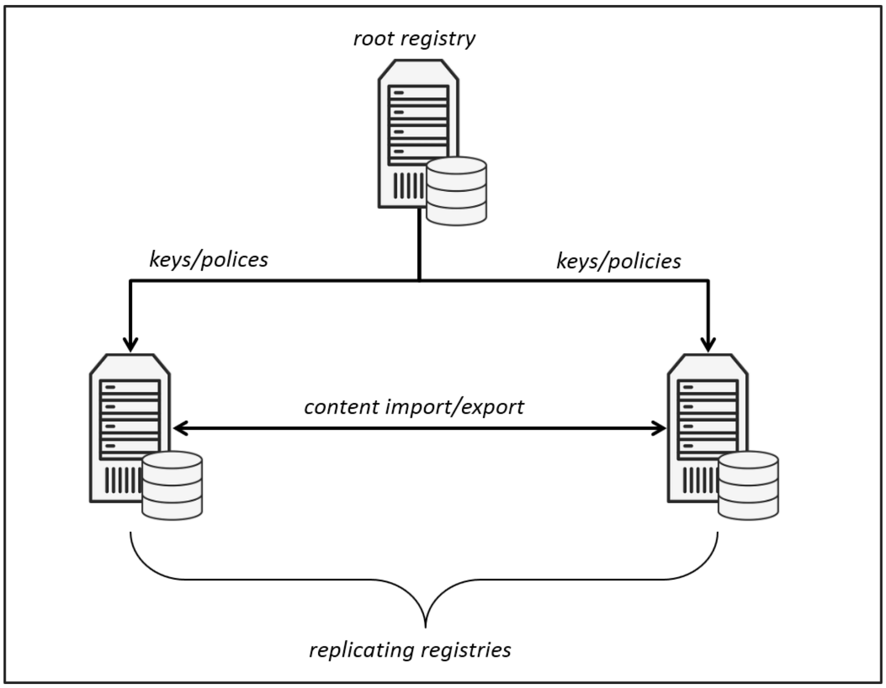


Figure 6 Registries Replication

To avoid potential key[7] collisions during a replication, this design defines a *root registry* as a mechanism for generating keys for all *affiliated* registries. The root registry may delegate key generation to affiliated registries, although this process should be compliant to a policy defined by the root registry.

From a service composition perspective, the replicating registries design implements an orchestration service composition pattern, with a root registry serving as a "director".

[6] In the UDDI specification, "affiliated registries are sets of registries that share compatible policies for assigning keys and managing data" [8].

[7] A key is a unique token used to identify and refer to a service description stored in a registry. In UDDI a key is a URI.

Like P2P service discovery, the replicating registries approach offers a convenient way for a user to access data across multiple registries; however, it also presents significant challenges for implementing.

Among the most notable challenges are:

1) Maintaining copies of a specific registered service in multiple registries naturally leads to potential collisions between keys.
2) Entering and updating data in another registry requires a publisher to adhere to rules and constraints established by the "receiving" registry, which may differ from rules adopted by the "publishing" registry.
3) To make it possible for a service description to be imported from or exported to other registries, all affiliated registries should share a common data model.

This approach was also researched and prototyped by SWIM registry implementers as part of the SWIM Common Registry (SCR) effort [6]. However, due in large part to the challenges described above, the SCR was never implemented.

5.3.2 Service Index

A *service index* is not a registry per se but a compilation of information that exists in registries [7].

There are a few important distinctions[8] between a service index and a registry that should be noted:

- The information about services presented in an index may not be published by a service provider; the index can be created by a third party that collects information exposed by various registries (so-called *passive publishing*).
- Different indexes can present different kinds of information – some richer, some sparser; some indexes may present full service descriptions as published in the "original" source (registry), and some may provide only references[9].
- Anyone can create their own index; when descriptions are exposed, they can be harvested using spiders and arranged into an index. The index may be implemented as a public Web document and can be accessed by anyone on the Semantic or "traditional" Web.

Figure 7 depicts implementation of a service index.

[8] These considerations are inspired by W3C's Web Services Architecture [7].
[9] Google search engine is often cited as an example of the index approach.

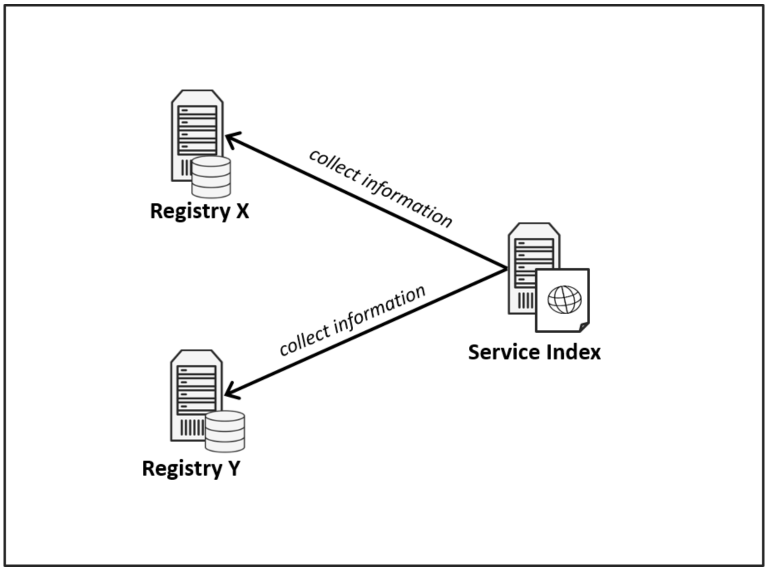


Figure 7 Service Index

5.3.3 Alternative Approaches Comparison Matrix

Approach	*Advantage*	*Disadvantage*
P2P Service Discovery	Registries can function independently from each other; if one or more registries are off-line, the rest of the registries may continue working without interruptions. Funding and administrative issues are addressed independently.	Query results may vary depending on the number of registry-services being on-line. Because discovery is executed "as needed," query response time can be affected by performance of network(s) along a request route.
Replicating Registries	All information is stored in a "home" registry, which makes query implementation easier and faster.	Having a root registry shared by all registries requires collaborative development and/or funding which could be logistically challenging. All registries should adhere to a common set of policies for modifying and entering data in a registry.
Service Index	The information presented as part of the "open" Web is easily accessible by all users.	Service descriptions in an index are not generally published by a service provider and thus the service information is not authoritative. The information contained in an index could be out of date.

6 Operational Scenarios

6.1 Adding Integration Module to a SWIM Registry

This scenario describes a design for applying SIRF principles to an existing SWIM registry-application. It has been successfully prototyped as part of the NSRR (FAA SWIM registry) registry integration effort. This prototype, Registry Integration Module (RIM), has been fully described in a specification document [17] along with other artifacts available at swim.aero/rim.

6.1.1 Actors

There are three interacting entities for this operational scenario:

1. *Registry X* - “home” registry, the starting point of a query, from a user perspective,
2. *Human Users* - users who have accounts in Registry X,
3. *Registry Y* - peer registry to Registry X.

6.1.2 Premise

Registry X was originally implemented as a registry-application. The registry deployed a collection of HTML pages for receiving input from users (almost always service providers) and enabled users (usually but not always service consumers) to browse or query the information. The registry exhibited a GUI for interaction with the users. The service descriptions supplied by the users were persisted in a registry database (see Figure 8).

6.1.3 Implementation

This scenario is implemented by way of adding (“plugging in”) a registry-service to Registry X. Registry X has been developed as a registry-application (similar to existing SWIM registries) and is modified to allow the exchange of data with other registries, represented here by Registry Y.

This registry-service should follow all engineering and architectural guidelines described in the ConOps (see Section 5.2.1, Table 3); that is, all registries should:

1. Reside on the Internet (TCP/IP) network,
2. Implement common Web protocols, (XML, URI, HTTP),
3. Use SDM-X as a shared data model,
4. Be semantically compliant with SDCM,
5. Use semantics artifacts, such as SWIM Service Product, Service Availability Status, and Service Interface Type taxonomies, as clauses for request queries.

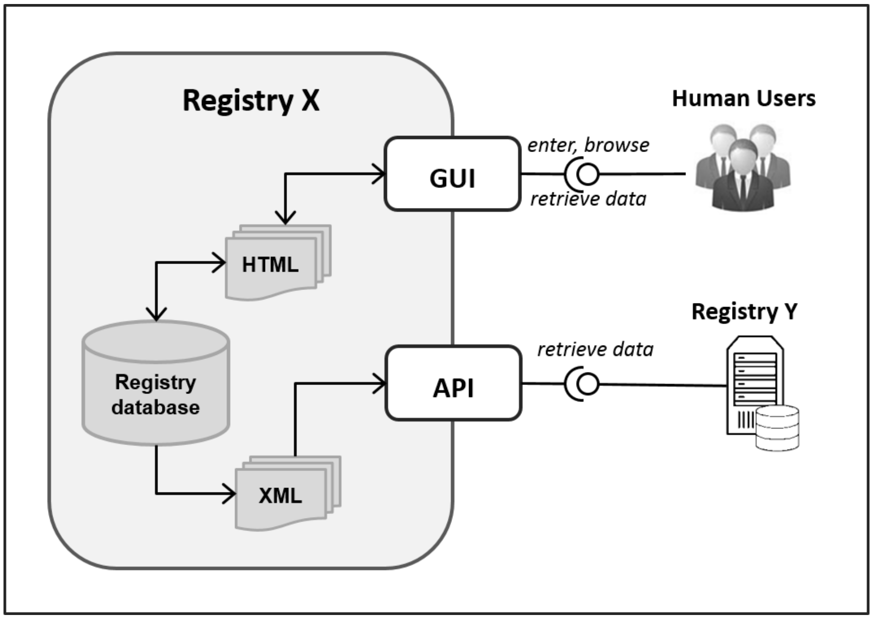


Figure 8 Conceptual vision of SIRF operational scenario

6.1.4 Execution Flow

1. An interaction is initiated by Human Users which deploys any HTTP client to post a query to Registry X via the API interface.
2. Upon receiving the request, Registry X:
    a. Processes the request by accessing its database and query mechanism,
    b. Sends a request to Registry Y with the same query.
3. Registry Y receives the request from Registry X and generates a response and sends it back to Registry X. (Note: it is possible, although not elaborated in this operational scenario, that Registry Y in its turn will "ask" yet another registry by sending the same query.)
4. Registry X augments its own response with the response received from Registry Y and returns the combined response to Human Users who initiated the query.

From the perspective of a service composition, both Registry X and Registry Y should execute a P2P Service Discovery design (see section 5.2.2).

7 Summary of Impacts

Implementing the SIRF will realize key tenets of service orientation, namely discoverability and composability. It will have the following impacts:

- It will assist in enabling global discoverability, that is, the ability of SWIM stakeholders to find services, regardless of which SWIM registry originates and maintains the service metadata.
- It will promote a technological means for presenting all aspects of a service's metadata in a manner suitable for both human-readable and machine-processable representations.

- It will serve as a foundation for further advancement of a shared vision of services, and leverage development of commonly-shared artifacts and elements such as vocabularies, specifications, and practices in the international aviation community.

Acronyms

ATM	Air Traffic Management
API	Application Programming Interface
ConOps	Concept of Operations
EIF	European Interoperability Framework
FAA	Federal Aviation Administration
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
LCIM	Levels of Conceptual Interoperability Model
NAS	National Airspace System
NEXTGEN	Next Generation Air Traffic Management
NSRR	NAS Service Registry/Repository
OASIS	Organization for the Advancement of Structured Information Standards
P2P	Peer-to-Peer
RIM	Registry Integration Module
SCR	SWIM Common Registry
SIRF	SWIM Inter-Registry Framework
SDCM	Service Description Conceptual Model
SDM-X	Service Description Model for XML
SESAR	Single European Sky ATM Research Programme
SOA	Service-Oriented Architecture
SWIM	System Wide Information Management
TCP/IP	Transmission Control Protocol/Internet Protocol
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
XML	eXtensible Markup Language