



International Civil Aviation Organization

**THE SECOND MEETING OF SYSTEM WIDE INFORMATION
MANAGEMENT TASK FORCE (SWIM TF/2)**

Bangkok, Thailand, 09 – 12 April 2018

Agenda Item 3: d) Task 1-4: SWIM Governance

POLICY-CENTRIC GOVERNANCE MODEL FOR APAC SWIM

(Presented by USA)

SUMMARY

All SWIM components must be based on common standards and conform to common rules to guarantee interoperability and alignment with business goals. This is achieved by establishing a robust and sustainable SOA governance model that ensures that the common standards, policies and rules are identified, consistently applied and enforced.

This paper examines the concept of a policy-centric governance model and describes its distinct components. It also provides selected examples of deploying this model in the context of APAC SWIM, and formulates conclusions regarding future work in this area.

1. INTRODUCTION

1.1 The ICAO Asia and Pacific Office (APAC) is in the process of evolving their traditional mode of information exchange into a paradigm of service-oriented operations known as System Wide Information Management (SWIM). SWIM effectuates communication and interoperability requirements for highly-distributed, loosely-coupled and platform-independent components through consistent application of principals of Service-Oriented Architecture (SOA).

1.2 One of the most challenging aspects of establishing a SWIM service-centric environment is an effective governance model. In the context of APAC SWIM, governance is defined as a process of enabling a set of enforceable policies, standards, organizational activities and enforcement mechanisms to ensure that all of the independent information exchange efforts come together to meet APAC SWIM business objectives.

1.3 SOA governance is a complex and multi-dimensional phenomenon. It is commonly comprised of various aspects which in SOA industry literature are often described in the form of models [1], [2], [5]. There are multiple models that are used for establishing a SOA governance framework, e.g., business model, stakeholder model, security model, integration model, etc.

1.4 In this paper, we propose a policy-centric model that defines a structure for identifying, defining and enforcing the technological and procedural policies in the context of APAC SWIM.

1.5 It should be noted that the model presented here is not a complete product. Rather, it is intended to be a “living” document that will be complemented by other governance models and constantly updated to ensure alignment with emerging business requirements and technological solutions.

2. DISCUSSION

2.1 SOA Governance is realized through the body of policies that drive the overall behavioral model of the SOA participants and ensures interoperability of service operating in the SOA. Policies are the means by which governance is operationalized [2].

2.2 The Policy-Centric Model outlined in this document consists of three interrelated major concepts. These concepts are:

1. **Standard** – a set of requirements, models or regulations adopted or established by authority or general consent.

Note: artifacts such as open and proprietary standards, specifications, data, exchange and semantic models and taxonomies, protocols and best practices are generalized by this model as standards.

2. **Policy** – a statement that defines constraints on the behavior of a managed resource, a user or an organization.

Note: this document makes no assumptions about format or content; the policies can be both codified (machine-processable) and natural language (human-readable) documents. They can also be technological as well as business or organizational.

3. **Enforcement mechanism** – a process through which compliance with the organizational policies is ensured.

Note: governance enforcement mechanisms may include (but are not limited to): governance boards, working groups, and governance tools and enabling technologies [3].

2.3 The Policy-Centric Model at a high conceptual level is depicted in **Figure 1**.

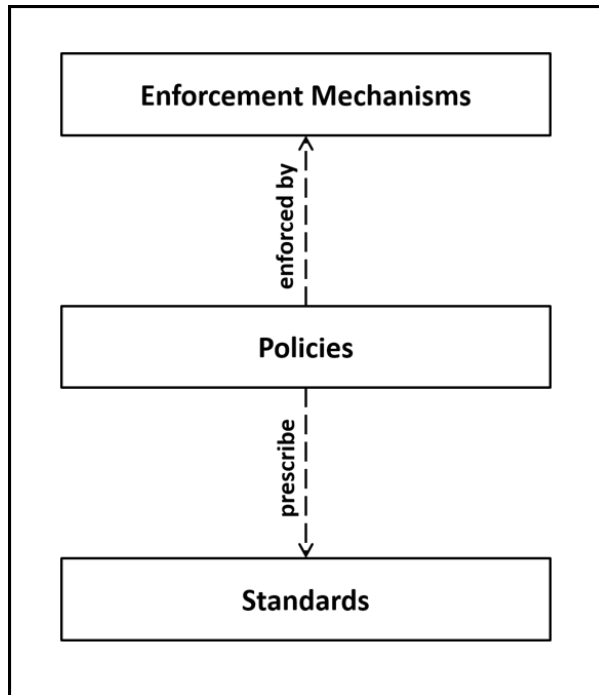


Figure 1: Conceptual policy-centric model

2.4 The relationships between these components are described as follows:

1. Policies *prescribe* or *mandate* usage of identified standards.

Note: policies define the rules for selecting and implementing technological or business solutions commonly conveyed by the standards. In other words, the policies serve as a filter for selecting what standards to use and how these standards should be applied.

2. Policies are *enforced* or *monitored* by enforcement mechanisms.

Note: in order for a governance model to be effective, policies must be processed through appropriate enforcement processes and compliance monitoring mechanisms. These mechanisms should be able to support policy enforcement at both design and run time.

Example of Implementation

2.5 Although the previous section defines a very straightforward model, real-life relationships among various artifacts and processes in the context of this model can be quite complex. Figure 2 illustrates how the policy-centric model can be realized by utilizing well-known standards, policies and enforcement mechanisms.

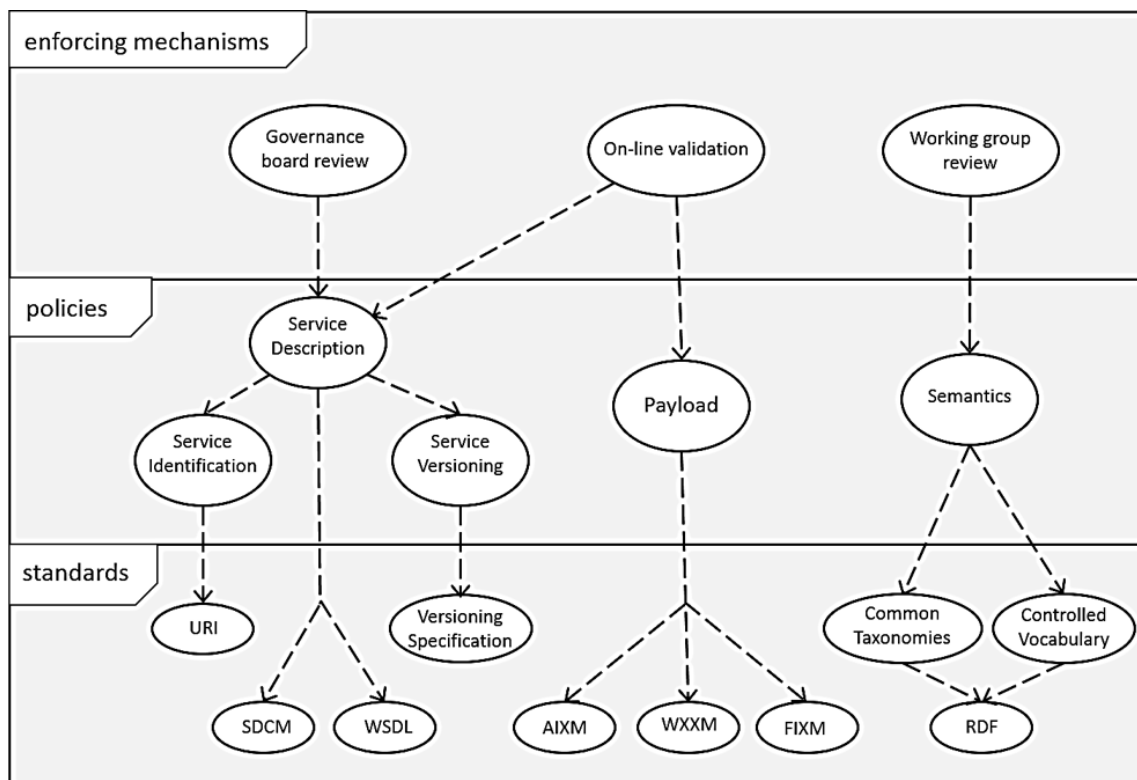


Figure 2: Example of a "populated" Policy-Centric Model

2.6 For example, an open international standard such as RFC 2396 Uniform Resource Identifiers (URI) [4] and a projected SWIM APAC versioning specification may serve as a base for a Service Identification Policy and a Service Versioning Policy respectively (see example of APAC SWIM Versioning Specification in **Appendix A**). A Service Description Policy can be established based on international standards or models, such as the Service Description Conceptual Model (SDCM) [6] and/or the Web Services Description Language (WSDL) [7]. It is also conceivable that the Service Description Policy may reference the already established Service Identification Policy and Service Versioning Policy. For example, the Service Description Policy may say, “Every service description MUST identify a service version as prescribed in the Service Versioning Policy document”.

2.7 Compliance with the Service Description Policy may be enforced by having a (notional) Governance Board or a Working Group review and approve the service description. (If the service description is a machine-processable document, an automatic policy enforcement tool can be deployed¹.)

Conclusion

2.8 SOA Governance – and inherently APAC SWIM governance – is realized through the body of policies that drive the overall behavioural model of participants and ensures interoperability of services operating in the SWIM environment.

¹ The FAA SWIM registry, for example, provides a tool that allows a user to validate a WSDL document against the WSDL standard as well as against an FAA specification that establishes additional constraints and requirements for a WSDL-based document.

2.9 A robust and sustainable governance model needs to be developed and deployed to ensure that a common set of policies, rules, and standards for identifying, designing, implementing, discovering, and operating SWIM-enabling components is consistently applied and enforced throughout the APAC Region.

2.10 The policy-centric model presented in this paper is the first building block of a larger edifice. It is expected to be complemented by other policy models (to be defined) that together should constitute the APAC SWIM governance framework.

References

- [1] SOA Governance Framework, Technical Standard, The Open Group, 2009
<http://www.opengroup.org/soa/source-book/gov/index.htm>
- [2] Service-Oriented Architecture Governance for the Services Driven Enterprise, Wiley, Eric A. Marks, 2015
- [3] SOA Governance: Governing Shared Services On-Premise and in the Cloud; Thomas Erl; The Prentice Hall; 2011
- [4] RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax; Network Working Group; T. Berners-Lee et al; August 1998
<http://www.isi.edu/in-notes/rfc2396.txt>
- [5] Next Generation SOA Governance; Eric A. Marks; AgilePath; 2008
- [6] Service Description Conceptual Model (SDCM), 2.0; FAA/SESAR; June 3, 2016
https://www.faa.gov/nextgen/programs/swim/governance/servicesemantics/media/SDCM_v2.0/SDCM_v2.0.html
- [7] Web Services Description Language (WSDL), version 2.0; W3C; 26 June 2007
<https://www.w3.org/TR/2007/REC-wsdl20-20070626/>

3. ACTION BY THE MEETING

3.1 The meeting is invited to:

- a) Review the contents of this working paper;
- b) Provide feedback on the proposed model;
- c) Advise on the way forward and in particular set priorities for the development of relevant artifacts.

Appendix A. Example of an APAC SWIM Standard and Policy

This Appendix presents an example of a *standard* for defining a version identifier and subsequently a *policy* that effectuates this standard. The standard and policy are shown below as a single specification document. This example is not mandatory and does not make any assumption about the format of a “real life” document.

1. SCOPE

This specification provides requirements and guidelines for the versioning of a service developed and operated in the context of APAC SWIM. The requirements defined in this document are based on (but not limited to) non-proprietary industry-wide common practices and recommended guidelines.

2. TERMS AND DEFINITIONS

<i>Versioning</i>	The process of managing multiple releases of a service or its artifacts for the purpose of managing the service’s evolution.
<i>Version Identifier</i>	A unique name or number that denotes a particular version of a service or service-related artifact.
<i>Major changes</i>	Changes or updates that are not backward-compatible; that is, they force a consumer agent to change in order to use the new version of the service. It is said that these changes "break" a consumer agent.
<i>Minor changes</i>	Changes that allow a consumer agent to continue to use the existing version of the service (they do not "break" the consumer agent), although the consumer agent is unable to use or is unaware of the new features. These changes are considered backward-compatible (e.g., a new capability, new optional request parameters).
<i>Patches</i>	Backward-compatible error corrections that do not affect in any way interaction between service and consumer agent (e.g., fixing a bug in software, correcting a typographical error in an XML schema document).

3. REQUIREMENTS

3.1 A version identifier shall be formatted as three dot-separated positive integers without leading zeroes, e.g., “1.2.3”.

1.1.1 The first digit SHALL represent a major change to the service or service artifact.

1.1.2 The second digit SHALL represent a minor change to the service or service artifact.

1.1.3 The third digit MAY represent a patch to the service or service artifact.

Explanation: To illustrate usage of these requirements, figure 1 presents a scenario wherein a service consumer agent was designed to use version 1.0.0 of some service. After minor changes, which were communicated through the incremented second and third digits in the version identifier (1.1.0 and 1.2.1 respectively), the consumer agent could continue to use each subsequent version of this service. However, after major changes were made (and communicated via the incremented first digit), the consumer agent required modification in order to use version 2.0.0 of the artifact.

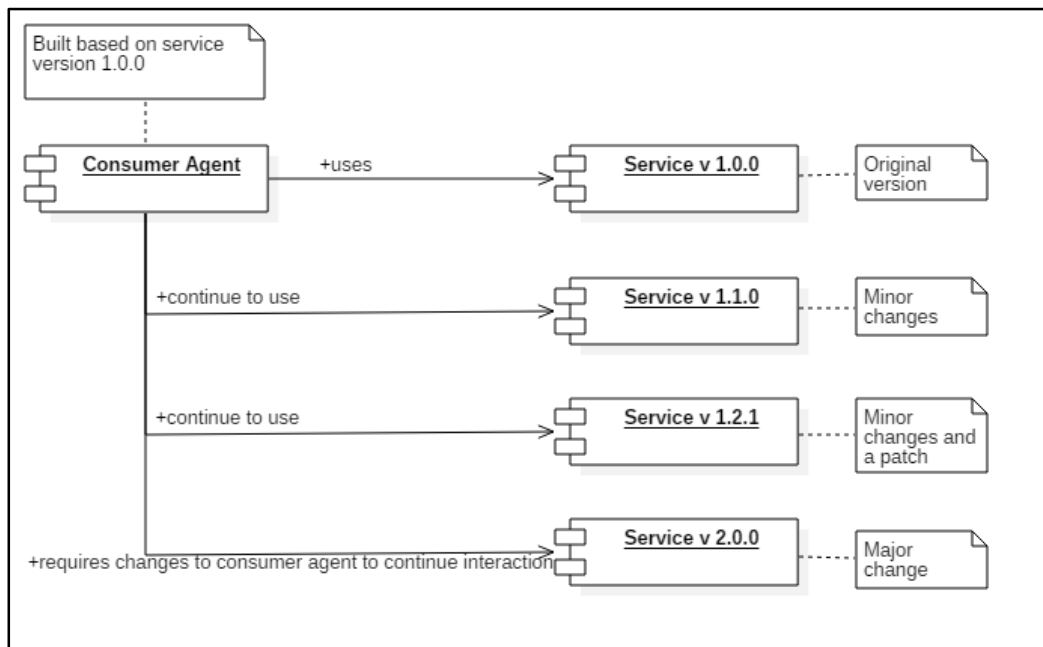


Figure 1 Example of applying described versioning approach

1.1.4 A version identifier SHALL be presented as a part of the versioned artifact without reliance on any additional tooling.

Explanation: Versioning data associated with an artifact should be autonomous; that is, it should not rely on a tool or repository that may store or manage the artifact and should provide consistent versioning identification even when used outside of this tool.

4. APAC SWIM VERSIONING POLICY

4.1 All SWIM services SHALL identify their version as prescribed by the APAC SWIM Versioning Specification.

Explanation: This policy makes no assumptions about artifacts that must provide service versioning information. These artifacts may include a service description in both human-readable and machine processable formats, an entry in a service registry, a Service Level Agreement (SLA), etc .