
1 Introduction

Service Oriented Architecture is an architectural paradigm that has gained significant attention within the information technology (IT) and business communities. The OASIS Reference Model for SOA provides a common language for understanding the important features of SOA but does not address the issues involved in constructing, using or owning a SOA-based system. This document focuses on these aspects of SOA.

The intended audiences of this document include non-exhaustively:

- Architects will gain a better understanding when planning and designing enterprise systems of the principles that underlie Service Oriented Architecture.
- Standards architects and analysts will be able to better position specific specifications in relation to each other in order to support the goals of SOA.
- Decision makers will be better informed as to the technology and resource implications of commissioning and living with a SOA-based system; in particular, the implications following from multiple ownership domains.
- Users will gain a better understanding of what is involved in participating in a SOA-based system.

1.1 What is a Reference Architecture?

The SOA Reference Model defines reference architecture as “an architectural design pattern that indicates how an abstract set of mechanisms and relationships realizes a predetermined set of requirements.” More precisely, a reference architecture can be described as an architectural pattern that provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them [TOGAF v8.1].

It is possible to define reference architectures at many levels of detail or abstraction, and for many different purposes. In fact, the reference architecture for one domain may represent a further specialization of another reference architecture, with additional requirements over those for which the more general reference architecture was defined.

A reference architecture need not be a concrete architecture; i.e., depending on the requirements being addressed by the reference architecture, it may not be necessary to completely specify all the technologies, components and their relationships in sufficient detail to enable direct implementation. Such a concrete architecture may be valuable and necessary to ensure a successful implementation; however, the detail necessary in concrete architectures may force technology choices that are not forced by the requirements per se, but by the technology choices available at the time.

1.1.1 What is this Reference Architecture?

This Reference Architecture is an abstract realization of SOA, focusing on the elements and their relationships needed to enable SOA-based systems to be used, realized and owned; while avoiding reliance on specific concrete technologies.

When designing systems that are intended to be used across ownership boundaries over extended periods of time it is necessary to address not only how the system is to be constructed, but also how it integrates with the life of users of the system and what is involved in owning such a system. In effect, we take a total cost of ownership stance on the architecture of SOA-based systems.

While requirements are addressed more fully in Section 2, the key assumptions that we make in this Reference Architecture is that SOA-based systems involve:

- resources that are distributed across ownership boundaries³;

³ Even in contexts that apparently have no ownership boundaries, such as within a single organization, the reality is that different groups and departments often behave as though they had ownership boundaries between them. This

- 43 • people and systems interacting with each other, also across ownership boundaries;
- 44 • security, management and governance is similarly distributed across ownership boundaries; and
- 45 • interaction between people and systems is primarily through the exchange of messages with
- 46 reliability that is appropriate for the intended uses and purposes.

47 Below, we talk about such an environment as a SOA ecosystem. Informally, our goal in this Reference
48 Architecture is to show how Service Oriented Architecture fits into the life of users and stakeholders in a
49 SOA ecosystem, how SOA-based systems may be realized effectively, and what is involved in owning
50 such a SOA-based system. We believe that this approach will serve two purposes: ensuring that the true
51 value of a SOA meeting the stated requirements can be realized using appropriate technology, and
52 permitting the audience to focus on the important issues without becoming over-burdened with the details
53 of a particular implementation technology.

54 **1.1.2 Relationship to the Reference Model**

55 The primary contribution of the Reference Model is that it identifies the key characteristics of SOA, and it
56 defines many of the important concepts needed to understand what SOA is and what makes it important.
57 This Reference Architecture takes the Reference Model as its starting point in particular in relation to the
58 vocabulary of important terms and concepts.

59 The Reference Architecture's goes a step further than the Reference Model in that we try to show how we
60 might actually have SOA-based systems. As noted above, SOA-based systems are better thought of as
61 ecosystems rather than stand-alone software products. Consequently, how they are used and managed
62 is at least as important architecturally as how they are constructed.

63 In terms of approach, the primary difference between the Reference Model and this Reference
64 Architecture is that the former focuses entirely on the distinguishing features of SOA; whereas this
65 document introduces concepts and architectural elements as needed in order to fulfill the core
66 requirement of realizing SOA-based systems.

67 **1.1.3 Relationship to other Reference Architectures**

68 It is fully recognized that other SOA reference architectures have emerged in the industry, both from the
69 analyst community and the vendor/solution provider community. Some of these reference architectures
70 are at a sufficient level of abstraction away from specific implementation technologies while others are
71 based on a solution or technology "stack." Still others use emerging middleware technologies such as the
72 Enterprise Service Bus (ESB) as the architectural foundation.

73 As with the Reference Model for SOA, the Reference Architecture for SOA is primarily focused on large-
74 scale distributed IT systems where the participants may be legally separate entities. While it is quite
75 possible for many aspects of the Reference Architecture to be realized on quite different platforms, we do
76 not dwell on such opportunities.

reflects good organizational practice; as well as reflecting the real motivations and desires of the people running those organizations.

77 **1.1.4 Expectations set by this Reference Architecture**

78 This Reference Architecture is not a complete blueprint for realizing SOA-based systems. Nor is it a
79 technology map identifying all the technologies needed to realize SOA-based systems. It does identify
80 many of the key aspects and components that will be present in any well designed SOA-based system.

81 In order to actually use, construct and manage SOA-based systems many additional design decisions
82 and technology choices will need to be made. For example, we identify in this Reference Architecture a
83 mode of interaction between service participants based on some form of message communication. The
84 particular style of message communication, the transport technologies and the message encoding
85 technologies are all important issues that are beyond the scope of this document. Similarly, the particular
86 governance models used in a given application will need to be elaborated on and made concrete – for
87 example, the exact committees and their jurisdictions would have to be set.

88 We believe that our approach will serve two purposes: ensuring that the true value of the SOA approach
89 can be realized on any appropriate technology, and permitting our audience to focus on the important
90 issues without becoming over-burdened with the details.

91 The primary contribution of this Reference Architecture is to make clear which technology and design
92 choices are needed and what their purpose is. For example, we identify the role of participants and their
93 relationships in terms of social structures. The specific organizations involved; how roles are designed
94 and how the service interaction mechanisms determine the rights and responsibilities of the participants is
95 also beyond our scope: we identify the need for the determination but not the specifics.

96 **1.2 Service Oriented Architecture – An Ecosystems perspective**

97 Many systems cannot be understood by a simple decomposition into parts and subsystems. There are
98 too many interactions between the parts. For example, a biological ecosystem is a self-sustaining
99 association of plants, animals, and the physical environment in which they live. Understanding an
100 ecosystem often requires a holistic perspective rather than one focusing on the system's individual parts.

101 From a holistic perspective, a SOA-based system is a network of independent services, machines, the
102 people who operate, affect, use, and govern those services as well as the suppliers of equipment and
103 personnel to these people and services. This includes any entity, animate or inanimate, that may affect or
104 be affected by the system. With a system that large, it is clear that nobody is really "in control" or "in
105 charge" of the whole ecosystem; although there are definite stakeholders involved, each of whom has
106 some control and influence over the community.

107 Instead of visualizing a SOA as a single complex machine, it is perhaps more productive to think of it as
108 an ecosystem: a space where people, machines and services inhabit in order to further both their own
109 objectives and the objectives of the larger community. In certain situations this may be a difficult
110 psychological step for owners of so-called enterprise systems to take: after all, such owners may rightly
111 believe that since they own the system they should also have complete control of it.

112 This view of SOA as ecosystem has been a consistent guide to the development of this architecture.

113 Taking an ecosystems perspective often means taking a step back: for example, instead of specifying an
114 application hierarchy, we model the system as a network of peer-like entities; instead of specifying a
115 hierarchy of control, we specify rules for the interactions between participants.

116 The three key principles that inform our approach to a SOA ecosystem are:

- 117 • a SOA is a *medium* for *exchange of value* between independently acting *participants*;
- 118 • participants (and stakeholders in general) have legitimate claims to *ownership* of resources that are
119 made available via the SOA; and
- 120 • the behavior and performance of the participants is subject to *rules of engagement* which are
121 captured in a series of policies and contracts.