THE OPEN GROUP SERIES

SOA Source Book

THE Open GROUP

www.opengroup.org

SOA Source Book

Other publications by Van Haren Publishing

Van Haren Publishing (VHP) specializes in titles on Best Practices, methods and standards within four domains:

- IT management,
- Architecture (Enterprise and IT),
- Business management and
- Project management

These publications are grouped in series: *ITSM Library, Best Practice and IT Management Topics*. VHP is also publisher on behalf of leading companies and institutions: The Open Group, IPMA-NL, PMI-NL, CA, Getronics, Pink Elephant.

Topics are (per domain):

IT (Service) Management / IT Governance
ASL
BiSL
CATS
CMMI
COBIT
ISO 17799
ISO 27001
ISO/IEC 20000
ISPL
IT Service CMM
ITIL® V2
ITIL® V3
ITSM
MOF
MSF

Architecture (Enterprise and IT)
Archimate
GEA
TOGAF™

Business Management
EFQM
ISA-95
ISO 9000
ISO 9001:2000
SixSigma
SOX
SqEME®

Project/Programme/ Risk Management
A4 Project management
ICB / NCB
MINCE®
M_o_R®
MSP
PMBoK
PRINCE2®

For the latest information on VHP publications, visit our website: www.vanharen.net.

SOA Source Book

THE Open GROUP
www.opengroup.org

Van Haren PUBLISHING

Colofon

Title:	**SOA Source Book**
A publication of:	The Open Group
Publisher:	Van Haren Publishing, Zaltbommel, www.vanharen.net
ISBN:	978 90 8753 503 2
Print:	First edition, first impression, April 2009
Layout and design:	CO2 Premedia, Amersfoort-NL
Copyright:	© 2009, The Open Group

For any further enquiries about Van Haren Publishing, please send an e-mail to: info@vanharen.net

Contents

Chapter 1 Service-Oriented Architecture 1
1.1 What Is SOA? 1
1.1.1 Service-Oriented Architecture 1
1.1.2 SOA Architectural Style 2
1.2 SOA and Enterprise Architecture 2
1.2.1 Enterprise Architecture 3
1.2.2 SOA 4
1.2.3 Overview of SOA 5
1.2.4 Architectural Dimension of SOA 6
1.2.5 Mainstream SOA 7
1.3 SOA and Boundaryless Information Flow 8
1.3.1 The Problem 8
1.3.2 Boundaryless Information Flow through SOA 8
1.4 SOA Features and Benefits 10
1.4.1 Summary of Features and Benefits 11
1.4.2 Service 12
1.4.3 Service Re-Use 13
1.4.4 Messaging 14
1.4.5 Message Monitoring 14
1.4.6 Message Control 14
1.4.7 Message Transformation 15
1.4.8 Message Security 15
1.4.9 Complex Event Processing 15
1.4.10 Service Composition 16
1.4.11 Service Discovery 17
1.4.12 Asset Wrapping 18
1.4.13 Virtualization 19
1.4.14 Model-Driven Implementation 19
1.5 Maturity Model for SOA 20
1.5.1 Overview of the Model 20
1.5.2 Using the Model 20
1.5.3 Dimensions 22
1.5.4 Maturity Levels 23

Chapter 2 The SOA Reference Architecture 27
2.1 The Building Blocks of SOA 27
2.1.1 Services 28
2.1.2 Business Processes 28
2.1.3 Human Actors 29
2.1.4 Events 30
2.1.5 Service Descriptions, Contracts, and Policies 30
2.1.6 Service Compositions 31
2.1.7 Programs 32
2.1.8 Information Items, Data Items, and Data Stores 32
2.2 A High-Level Perspective of the SOA Reference Architecture 33
2.2.1 Overview 33
2.2.2 Operational Systems Layer 35
2.2.3 Service Components Layer 35
2.2.4 Services Layer 36
2.2.5 Business Processes Layer 36
2.2.6 Consumers Layer 37
2.2.7 Integration Layer 37
2.2.8 Quality of Service Layer 37
2.2.9 Information Layer 38
2.2.10 Governance Layer 38
2.3 Detailed Building Blocks of the SOA Reference Architecture 38
2.3.1 Composition 39
2.3.2 Messaging 40
2.3.3 Service Discovery 42
2.3.4 Asset Wrapping 43
2.3.5 Virtualization 43
2.3.6 Event Processing 44
2.4 Infrastructure for SOA 45
2.4.1 Service Repository 46
2.4.2 Messaging Program 46
2.4.3 Activity Monitor 48
2.4.4 PDPs and PEPs 49
2.4.5 Data Translator 49
2.4.6 Encryption Engine 50
2.4.7 Event Processor 51
2.4.8 Composition Engine 51
2.4.9 Service Registry 52

2.4.10 Service Components 53
2.4.11 Model-Implementation Environment 53

Chapter 3 Service-Oriented Infrastructure 55
3.1 Overview 55
3.2 SOI Reference Model 56
3.2.1 Business Requirements 56
3.2.2 Service-Level Requirements 57
3.2.3 BPM 57
3.2.4 IMF 57
3.2.5 Applications 57
3.2.6 Services Exposed to and Consumed by IT 57
3.2.7 Infrastructure Services 58
3.2.8 Physical Services 58
3.2.9 Virtualized Services 58
3.3 Infrastructure Services 58
3.3.1 Pure-Play Infrastructure Services 59
3.3.2 Cross-Application Infrastructure Services 60
3.3.3 Operational Infrastructure Services 61
3.4 Virtualized Services 61

Chapter 4 SOA and TOGAF 63
4.1 Using TOGAF for Enterprise SOA 63
4.1.1 Preliminary Phase 64
4.1.2 Architecture Requirements Management 66
4.1.3 Phase A: Architecture Vision 66
4.1.4 Phase B: Business Architecture 70
4.1.5 Phase C: Information Systems Architectures 71
4.1.6 Phase D: Technology Architecture 74
4.1.7 Phase E: Opportunities and Solutions 75
4.1.8 Phase F: Migration Planning 76
4.1.9 Phase G: Implementation Governance 76
4.1.10 Phase H: Architecture Change Management 76
4.2 Using TOGAF for SOA Solutions 77
4.2.1 Preliminary Phase 77
4.2.2 Architecture Requirements Management 78
4.2.3 Phase A: Architecture Vision 78
4.2.4 Phase B: Business Architecture 78

4.2.5 Phase C: Information Systems Architectures 79
4.2.6 Phase D: Technology Architecture 81
4.2.7 Phase E: Opportunities and Solutions 82
4.2.8 Phase F: Migration Planning 82
4.2.9 Phase G: Implementation Governance 82
4.2.10 Phase H: Architecture Change Management 82
4.3 Addressing Stakeholder Concerns in SOA 83
4.3.1 Stakeholders 83
4.3.2 Concerns and Models 84
4.4 Information Architecture for SOA 87
4.4.1 Importance of Information Architecture for SOA 88
4.4.2 Information Architecture for SOA using TOGAF 89
4.4.3 Preliminary Phase 90
4.4.4 Phase A: Architecture Vision 91
4.4.5 Phase B: Business Architecture 91
4.4.6 Phase C: Information Systems Architectures 92
4.4.7 Phase D: Technology Architecture 93

Chapter 5 SOA Governance 95
5.1 Introduction to SOA Governance 95
5.1.1 Definition of SOA Governance 96
5.1.2 Enterprise SOA Governance Models 97
5.1.3 SOA Governance Framework 98
5.2 SOA Governance Reference Model 99
5.2.1 Governance Principles 99
5.2.2 SOA Activities 100
5.2.3 Governing Processes 101
5.2.4 Roles and Responsibilities 102
5.3 SOA Governance Vitality Method 103
5.3.1 The Plan Phase 104
5.3.2 The Define Phase 104
5.3.3 The Implement Phase 104
5.3.4 The Monitor Phase 105

Index 107

List of Figures

Figure 1: Overview of a Service-Oriented Architecture 5
Figure 2: Boundaryless Information Flow 9
Figure 3: Information Silos 9
Figure 4: SOA for Boundaryless Information Flow 10
Figure 5: SOA Maturity Model Matrix 21
Figure 6: High-Level Perspective of the SOA Reference Architecture 34
Figure 7: Basic Model for Service Composition 39
Figure 8: Model for Scripted Service Composition 39
Figure 9: Basic Messaging Model 40
Figure 10: Detailed Messaging Model 41
Figure 11: Model for Service Discovery 42
Figure 12: Model for Asset Wrapping 43
Figure 13: Model for Virtualization 44
Figure 14: Model for Event Processing 45
Figure 15: Service-Oriented Infrastructure Reference Model 56
Figure 16: Example ODBC Service Stack 60
Figure 17: SOA Governance Relationships 96
Figure 18: SOA Governance Aspects 97
Figure 19: SOA Governance Framework 99
Figure 20: SOA Governance Vitality Method 103

List of Tables

Table 1: SOA Features, Benefits, and Infrastructure 12
Table 2: SOA Concerns, Stakeholders, and Models 87

Preface

The Open Group

The Open Group is a vendor-neutral and technology-neutral consortium, whose vision of Boundaryless Information Flow™ will enable access to integrated information within and between enterprises based on open standards and global interoperability. The Open Group works with customers, suppliers, consortia, and other standards bodies. Its role is to capture, understand, and address current and emerging requirements, establish policies, and share best practices; to facilitate interoperability, develop consensus, and evolve and integrate specifications and Open Source technologies; to offer a comprehensive set of services to enhance the operational efficiency of consortia; and to operate the industry's premier certification service, including UNIX® certification.

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

The Open Group has over 15 years' experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification.

More information is available at www.opengroup.org/certification.

The Open Group publishes a wide range of technical documentation, the main part of which is focused on development of Technical and Product Standards and Guides, but which also includes white papers, technical studies, branding and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

This Document

The Open Group's *SOA Source Book* is a collection of source material for use by enterprise architects working with Service-Oriented Architecture (SOA).

It consists of material that has been considered and in part developed by The Open Group SOA Working Group[1]. The SOA Working Group is engaged in a work program to produce definitions, analyses, recommendations, reference

1 Refer to www.opengroup.org/projects/soa.

models, and standards to assist business and information technology professionals within and outside of The Open Group to understand and adopt SOA. The Source Book does not represent the final output of that work program, which will be published as a collection of Open Group Standards and Guides. It is an interim publication, and its content will not necessarily be reflected in the final output.

The material reflects input from a large number of people from a wide range of Open Group member companies, including product vendors, consultancies, and users of SOA. In some cases, these people have brought concepts developed, not just by themselves, but by groups of people within their organizations. The input has been refined and further developed through discussion within the Working Group. The value in the result is due to the ideas and efforts of the Working Group members.

The material is now published in its current form to make that value available to the wider architecture community.

Chapter 1 discusses SOA in relation to enterprises, and describes how to evaluate SOA features in business terms.

Chapter 2 presents The Open Group SOA Reference Architecture.

Chapter 3 describes how to apply the principle of service-orientation to infrastructure.

Chapter 4 explains how to use TOGAF - the comprehensive architecture framework developed and maintained by The Open Group - for SOA.

Chapter 5 describes SOA governance, and provides an initial explanation of how to define and maintain an SOA governance regimen for an enterprise.

Trademarks

Boundaryless Information Flow™ and TOGAF™ are trademarks and Making Standards Work®, The Open Group®, UNIX®, and the "X" device are registered trademarks of The Open Group in the United States and other countries.

Model Driven Architecture® and MDA® are registered trademarks, and Business Process Modeling Notation™, BPMN™, MOF™, and Unified Modeling Language™ are trademarks of the Object Management Group, Inc. in the United States and/or other countries.

OASIS® is a registered trademark, and Security Assertion Markup Language™ and SAML™ are trademarks of OASIS.

W3C® is a registered trademark (registered in numerous countries), and XML™ and XSL™ are trademarks of the World-Wide Web Consortium (W3C); marks of W3C are registered and held by its host institutions MIT, ERCIM, and Keio.

The Open Group acknowledges that there may be other brand, company, and product names used in this document that may be covered by trademark protection and advises the reader to verify them independently.

Acknowledgements

The Open Group gratefully acknowledges the following people in contributing, either directly or indirectly, to the SOA Source Book.

The material in the book is derived from the work of the Definition of SOA, SOA Reference Architecture, SOA/TOGAF Practical Guide, SOA Governance, Service-Oriented Infrastructure, and SOA Ontology projects of The Open Group SOA Working Group, from work done by The Open Group Service Integration Maturity Model project, and also from work done jointly by the SOA Working Group and the Semantic Interoperability Working Group.

The co-chairs of the SOA Working Group, Tony Carrato (IBM) and Mats Gejnevall (Capgemini), together with former co-chair Chris Greenslade (CLARS), member Jorge Diaz (IBM), and Forum Director Chris Harding (The Open Group), comprise the Working Group's Steering Committee. They are responsible for the overall direction of the Working Group and contribute greatly to the quality of its work. In addition, Tony Carrato took a particular interest in guiding the development of the Source Book. Chris Harding was primary author.

The Definition of SOA project was led by Dave Hornford (Hornford Associates).

The SOA Reference Architecture project is led by Ali Arsanjani (IBM) and Nikhil Kumar (ApTSi). Ali Arsanjani made a particular contribution in providing the base document for the work, having led its development within IBM.

The SOA/TOGAF Practical Guide project is led currently by Awel Dico (Bank of Montreal) and Dave Hornford, and was led formerly also by Steve Bennett (BEA Systems).

The SOA Governance project is led currently by Mats Gejnevall and Jorge Diaz, and was led formerly also by Andrew Hately (IBM), Tony Carrato, and Steve Bennett. In addition, Bill Brown (IBM) provided a substantial part of its base material.

The Service-Oriented Infrastructure project is led currently by Hemesh Yadav (Wachovia), E.G. Nadhan (HP), and Michael Salsburg (Unisys), and was led formerly also by Mark England (HP) and Frank Kroon (formerly Capgemini, now HP). E.G. Nadhan authored the Service-Oriented Infrastructure section of the Source Book.

The SOA Ontology project is led by Chris Harding.

The Open Group Service Integration Maturity Model project is led by Andras Szakal (IBM). He was responsible for providing the project's base document, which resulted from work led by Ali Arsanjani within IBM.

The contribution of the Semantic Interoperability Working Group was led by Arnold Van Overeem (CapGemini) and Ron Schuldt (Lockheed Martin).

Many of the people mentioned above also made contributions to projects of which they were not officers.

The following Working Group members, who were not project or working group officers, made particular contributions to one or more projects: Stuart Boardman (CGI), Kathy Carusone (MIT Lincoln Laboratory), Dave Chapelle (BEA Systems), Bill Estrem (Metaplexity), Ed Harrington (Model-Driven Solutions), Harry Hendrickx (Capgemini), Heather Kreger (IBM), Bob Laird (IBM), Srikanth Inaganti (Wipro), Shreyas Kamat (Infosys), Rich Valentine (Unisys), and Bobbi Young (Unisys).

Finally, over 300 other people have been involved in the SOA Working Group. It is not possible to mention them all individually, but their collective contribution is important.

Referenced Documents

The following documents are referenced in this Source Book:

- The Boundaryless Organization: Breaking the Chains of Organizational Structure, by Ron Ashkenas, Dave Ulrich, Todd Jick, & Steve Kerr; ISBN 0-7879-5943-X.
- Control Objectives for Information and related Technology (COBIT), Version 4.1, available from ISACA; refer to www.isaca.org.
- Interoperable Enterprise Business Scenario (K022), published by The Open Group; refer to www.opengroup.org/bookstore/catalog/k022.htm.
- ISO/IEC 2382-1:1993, Information Technology – Vocabulary – Part 1: Fundamental Terms.
- OECD Corporate Governance Principles, 2004, available from the Organization for Economic Cooperation and Development; refer to www.oecd.org.
- TOGAF; refer to www.opengroup.org/togaf.
- The SOA Solution Stack: A Reference Architecture for Designing SOA Solutions, IBM Corporation.
- The following standards defined by OASIS; refer to www.oasis-open.org:
  - Business Process Execution Language (BPEL)
  - Security Assertion Markup Language (SAML)
  - Universal Description Discovery and Integration (UDDI)
  - Web Services Reliable Messaging (WS-ReliableMessaging)
  - Web Services Security (WS-Security)
  - Web Services Security Policy (WS-Security-Policy)
  - eXtensible Access Control Markup Language (XACML)
- The following standards, defined by the Object Management Group (OMG); refer to www.omg.org:
  - Business Process Modeling Notation (BPMN)
  - Meta Object Facility (MOF)
  - Unified Modeling Language (UML)
- The following standards, defined by the World-Wide Web Consortium (W3C); refer to www.w3.org:
  - Simple Object Access Protocol (SOAP)
  - Web Ontology Language (OWL)
  - Web Services Description Language (WSDL)

- Web Services Policy Framework (WS-Policy)
- eXtensible Markup Language (XML)
- eXtensible Stylesheet Language (XSL) Transformations (XSLT)

Chapter 1

Service-Oriented Architecture

This section discusses Service-Oriented Architecture (SOA) in relation to enterprises, and describes how to evaluate SOA features in business terms. It contains:

- A definition of SOA
- An analysis of the role of SOA in relation to enterprise architecture
- An explanation of how SOA can enable an enterprise to achieve Boundaryless Information Flow
- A description of the features of SOA and the business benefits that they provide
- An SOA maturity model that facilitates the assessment of an organization's current and desired future states in service integration and flexibility, and helps the organization to determine its architectural strategy for adopting service-orientation

1.1 What Is SOA?

This definition of SOA was produced by the SOA Definition team of The Open Group SOA Working Group.

1.1.1 Service-Oriented Architecture

Service-Oriented Architecture (SOA) is an *architectural style* that supports *service-orientation*.

Service-orientation is a way of thinking in terms of services and service-based development and the outcomes of services.

A service:

- Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit, provide weather data, consolidate drilling reports)
- Is self-contained

- *May be* composed of other services
- Is a "black box" to consumers of the service

1.1.2 SOA Architectural Style

An *architectural style* is the combination of distinctive features in which architecture is performed or expressed.

The SOA architectural style has the following distinctive features:

- It is based on the design of the services – which mirror real-world business activities – comprising the enterprise (or inter-enterprise) business processes.
- Service representation utilizes business descriptions to provide context (i.e., business process, goal, rule, policy, service interface, and service component) and implements services using service orchestration.
- It places unique requirements on the infrastructure – it is recommended that implementations use open standards to realize interoperability and location transparency.
- Implementations are environment-specific – they are constrained or enabled by context and must be described within that context.
- It requires strong governance of service representation and implementation.
- It requires a "Litmus Test", which determines a "good service".

1.2 SOA and Enterprise Architecture

SOA provoked hot debate when it burst onto the scene in 2005. Its advocates said that it would replace traditional information technology (IT) architecture. The traditionalists replied that SOA was nothing new; just a rehash of old (but good) ideas about encapsulation and loose coupling.

There is some truth in both of these positions. But in the main they are both wrong. Although SOA does include earlier architectural ideas, it is a distinct style which marks a major step forward. And, to obtain maximum benefit from SOA, an enterprise needs traditional architectural disciplines and methods.

1.2.1 Enterprise Architecture

Why does an enterprise need an SOA – or an architecture of any other kind?

The directing function of an enterprise – the board of directors of a commercial company, or the top-level management of a division or government department, for example – sets objectives for the enterprise, and decides how it should operate in order to achieve them. A clearly articulated architecture describes the desired enterprise organization and manner of operation. By doing so, it provides:

- A definition of the changes that should be implemented to achieve this organization
- A basis for control and governance of its ongoing operation

An enterprise architecture also provides a third benefit. Enterprises change over time. They combine and split, as in commercial mergers and spin-offs, or government department reorganizations. It is easier to combine an enterprise with another, or to split it into component parts, when it has a clearly-defined architecture. This brings significant cost savings, and can increase the value of a commercial enterprise.

Enterprise architecture in its widest sense includes much more than IT. It covers business operations, finance, people, and buildings in addition to technology, and it covers technologies other than IT, such as for manufacturing or transport. The enterprise architect must understand these areas, at least well enough to supervise architects that specialize in them. The IT architect must be able to work in teams with such specialists.

The SOA Source Book focuses on the IT component of enterprise architecture. This is concerned with the strategic development of an enterprise's IT. It looks at the whole of the enterprise, not just a particular system, and it looks at the long-term evolution of the IT, not just at what should be installed today.

The quality of an enterprise's IT architecture can have a major impact on its business performance. Since the 1950s, commercial and government organizations have become increasingly dependent on IT for the conduct of their everyday operations, and that trend looks likely to continue. Companies that use IT effectively prosper. The best of the once-derided .com companies

("When will they ever make a profit?") became household names. Companies with poor IT fall behind their competition, or fail.

Because of its importance to the overall business, enterprise IT architecture has become a profession. No company would think of undertaking the development of a major building without engaging a buildings architect with a professional status that provides a guarantee of competency. Similarly, companies undertaking the development of major IT systems look for professional enterprise IT architects. Their status as professionals indicates that they understand, and have a track record of applying, the best IT architecture methods and techniques.

1.2.2 SOA

An enterprise architect looks at the overall construction of the enterprise. SOA is a particular construction technique that can be used to build enterprise IT.

A particular technique can have a major impact on the overall construction. The introduction of steel-frame techniques in the latter part of the 19th century revolutionized buildings architecture. It made possible the skyscrapers of the 1920s, and the even larger buildings that we have today.

SOA could have a similar impact on IT architecture. It does not increase the size of IT systems, but it does increase their interoperability.

With SOA, the IT systems perform services that are defined and described in the context of the enterprise's business activities. Each service is identified, and what it does is clearly set out in the form of a contract. This principle enables use of techniques such as service composition, discovery, message-based communication, and model-driven implementation, which give fast development of effective and flexible solutions. They are important features of SOA. Their benefits – especially that of enterprise agility – are the most frequently quoted reasons for SOA adoption.

But it is the replacement of large, monolithic applications that have tiny interoperability interfaces, grudgingly provided and not guaranteed, by smaller, modular services that have interface descriptions and contracts, that is the most fundamental effect of SOA. This is the basis for the huge increase

in IT system interoperability that SOA can bring, not only within enterprises, but also between enterprises.

1.2.3 Overview of SOA

The principle of service-orientation can apply throughout the enterprise architecture, but is most commonly applied to the organization of the software that supports the enterprise's business operations. With SOA, this software is organized as a set of software services. The services are supported by an infrastructure that, together with the services, improves information flow within the enterprise and between the enterprise and external enterprises.

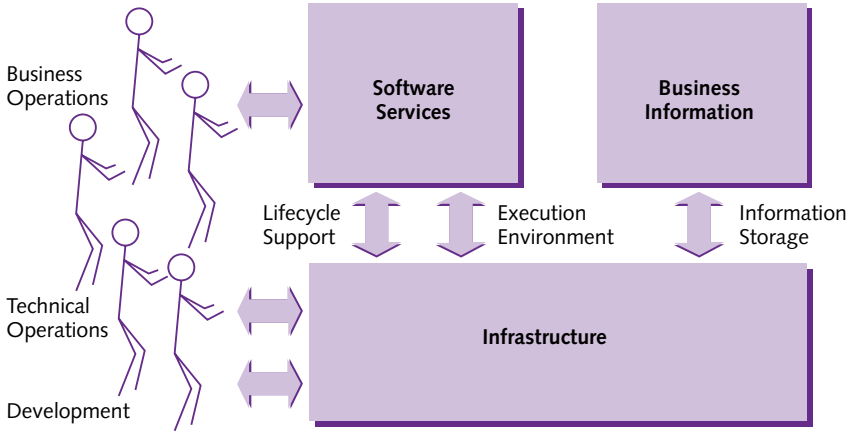

Figure 1: Overview of a Service-Oriented Architecture

The software services are used by the enterprise's business operations. This frequently involves a human-computer interface, often implemented as a web interface using portals, etc., but it may also involve other interfaces, such as machine interfaces for process control.

Specific sets of business processes, services, and interfaces are created in the context of a supporting infrastructure as service-based *solutions*. Each solution solves a particular business problem.

The business operations themselves may be organized on the service-oriented principle. Indeed, there are many people who believe that the greatest benefits of SOA are obtained when it is applied to the business architecture.

The infrastructure provides the execution environment for the software services. This includes the basic operating system and networking, and also includes specific support for software services, such as message passing and service discovery. The infrastructure is managed via human-computer interfaces by technical staff who are responsible for all aspects of operating the enterprise's IT, including its availability, performance, and security.

A major benefit of SOA is that it delivers enterprise agility, by enabling rapid development and modification of the software that supports the business processes. The infrastructure can provide for this by including facilities such as business-oriented scripting languages and model-driven implementation tools. These facilities support not only the creation of new software services, but also the modification and replacement of existing ones: the whole service lifecycle. They are used via human-computer interfaces by development staff.

The infrastructure also provides for storage of enterprise information. SOA can enable easier flow of information within and between enterprises. The information is not locked up in specific services, as it often is in the so-called "silo" applications of earlier architecture styles, but is available to all the software services that need it.

Service-orientation may extend to the design of the infrastructure, and many people advocate this, but it is not essential to service-oriented software architecture.

1.2.4 Architectural Dimension of SOA

It takes far greater knowledge and skill to erect a skyscraper than to build a house. The buildings architect must make complex stress calculations based on an understanding of the properties of the materials involved. Training and experience are essential for success.

Knowledge and skill are also needed for success with SOA. The IT architect must specify the right tools and infrastructure, create the basis for the identification of modular services, and ensure that appropriate implementation governance is in place. Good judgment in these matters is crucial.

Also, just as steel-frame construction is not appropriate for every building, SOA is not necessarily the right approach to solving every IT problem. The IT architect must know when, as well as how, to use SOA.

SOA can be a big investment. Its tools and infrastructure cost money, but that is only one part of what is needed. Development and operation staff must have special skills to create and use SOA, and the overall organization structure and culture must be right if the full benefits of SOA are to be achieved. Staff development and organizational change is often the larger part of the investment. Such an investment can only be justified in the light of a long-term strategy for the enterprise as a whole.

Many enterprises have undertaken small-scale SOA developments as part of a learning process. This is an excellent way for them to introduce SOA, but they often find it hard to extend beyond the initial pilot. Developers complain that they cannot justify the infrastructure that they need. Of course not! Expensive infrastructure cannot be justified on the basis of small projects and, in any case, looking for business justification for technical spend is putting the cart before the horse. The business need should come before the technical solution. SOA should be used where – and only where – it is the best way to meet that need.

This is where enterprise architecture comes in. Enterprise architecture creates long-term IT strategy in the light of business possibilities and needs. Inclusion in such a strategy is the only good justification for large-scale SOA.

1.2.5 Mainstream SOA

SOA is no longer a new toy. It is an established style that architects understand and can use.

The architect does not start by assuming SOA, but considers service-orientation and its associated techniques in the light of the business strategy. Sometimes, the technical possibilities can change that strategy, but the business needs and possibilities are still the main driving force. The architect finishes by specifying a particular combination of SOA techniques because it best realizes the possibilities and meets the needs.

This is the normal architectural approach to IT strategy. SOA and enterprise architecture may have seemed different in the beginning, but SOA is now part of the enterprise architecture mainstream.

1.3 SOA and Boundaryless Information Flow

Why is SOA important to The Open Group?

The Open Group's vision is Boundaryless Information Flow. It has long been a principle of enterprise organization that permeable boundaries between departments, organizational levels, enterprises, and nations deliver productivity and enterprise agility. This was established in the 1980s by pioneers such as Jack Welch of GE (see *The Boundaryless Organization: Breaking the Chains of Organizational Structure*). But traditional IT architectures hinder this! The need for Boundaryless Information Flow – provided by IT architectures that enable information to flow freely across the permeable organizational boundaries – was identified by The Open Group and described in its Interoperable Enterprise Business Scenario. The Open Group took on the mission of driving the creation of Boundaryless Information Flow.

1.3.1 The Problem

Enterprise architecture is the key to achieving Boundaryless Information Flow. The problem, as described in the Interoperable Enterprise Business Scenario, is that enterprises need the kind of architecture shown in Figure 2, in which the business processes are supported by systems that can exchange information freely.

Too often, however, they are faced with a situation where each business process has its own system which has its own particular interfaces and information formats, and is a so-called "information silo", as shown in Figure 3.

1.3.2 Boundaryless Information Flow through SOA

With SOA, the applications are replaced by services that interact with each other. Typically, interactions take place by exchange of messages via an Enterprise Services Bus (ESB) within the enterprise, or across the web in the case of external services, although other forms of interaction, even direct

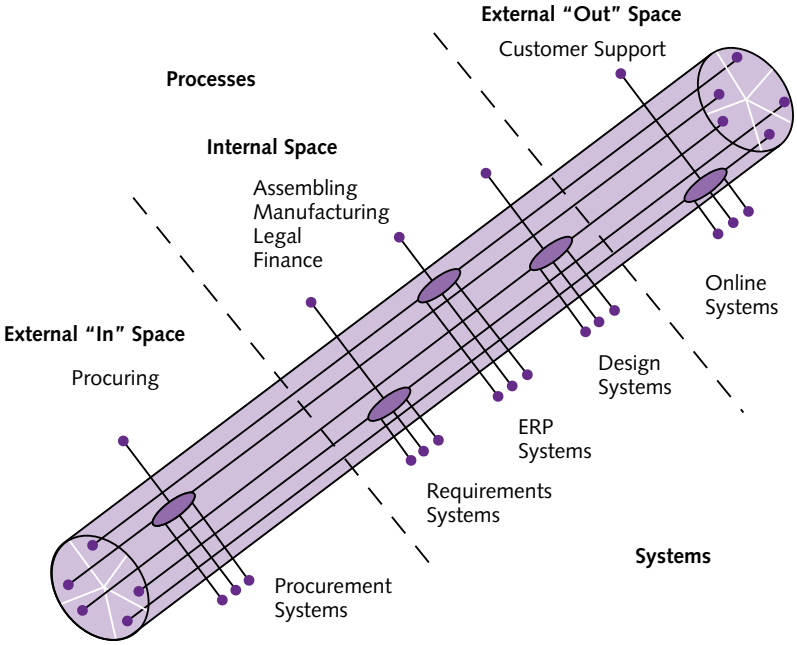

Figure 2: Boundaryless Information Flow

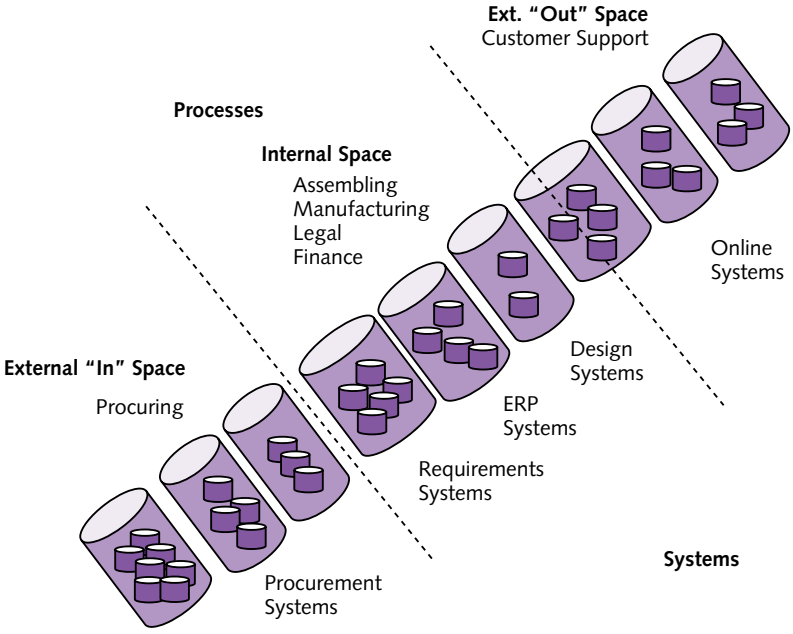

Figure 3: Information Silos

invocation of one service by another (so-called "hard-wiring") may be used. This style of architecture can be the basis of Boundaryless Information Flow, as illustrated in Figure 4.

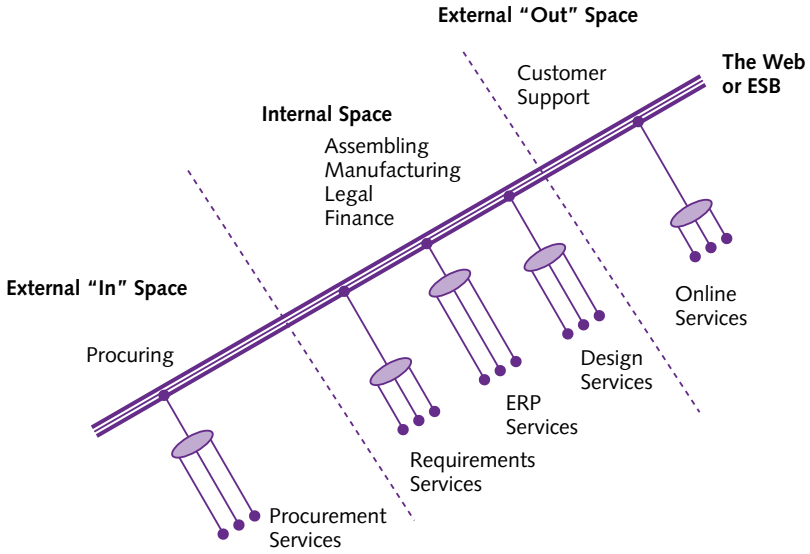

Figure 4: SOA for Boundaryless Information Flow

It is because of the potential for SOA to deliver Boundaryless Information Flow that SOA is critically important to The Open Group.

1.4 SOA Features and Benefits

SOA starts with a simple idea – the concept of *service*. This makes it possible to introduce other ideas, such as *service bus*, *service composition*, and *service virtualization*, each of which can be applied to the architecture of an enterprise to deliver benefits. As an architect, it is your job to evaluate the needs of your enterprise, and the costs of the different potential solutions, to determine which of these ideas should be applied, and how they should be applied, in your SOA.

An architect should always probe into the information given, about both requirements and solutions, to reach a level of understanding that goes deeper than the buzzwords. For example, it is often said that "SOA delivers

enterprise agility". What does "agility" mean for your enterprise? Is it the ability to re-combine existing functions to meet changing customer requirements? Is it the ability to develop new functions rapidly? Is it the ability to scale operations to meet different levels of demand? Within the broad concept of SOA, there are three very different ideas that can help you meet these different agility requirements: service composition, model-driven development, and service virtualization. You can build all of these ideas into your SOA, but they each require different – and expensive – supporting infrastructure. You must choose your solution to fit the requirements.

This section will help you to match the features of SOA to the needs of your enterprise, so that you can determine the kind of SOA that is appropriate.

1.4.1 Summary of Features and Benefits

Table 1 shows the main features and benefits of SOA, together with the infrastructure needed to support them.

Feature	Benefits	Supporting Infrastructure
Service	Improved information flow Ability to expose internal functionality Organizational flexibility	
Service Re-use	Lower software development and management costs	Service repository
Messaging	Configuration flexibility	Messaging service
Message Monitoring	Business intelligence Performance measurement Security attack detection	Activity monitor
Message Control	Application of management policy Application of security policy	PDPs and PEPs
Message Transformation	Data translation	Data translator
Message Security	Data confidentiality and integrity	Encryption engine
Complex Event Processing	Simplification of software structure Ability to adapt quickly to different external environments Improved manageability and security	Event processor
Service Composition	Ability to develop new function combinations rapidly	Composition engine

Feature	Benefits	Supporting Infrastructure
Service Discovery	Ability to optimize performance, functionality, and cost Easier introduction of system upgrades	Service registry
Asset Wrapping	Ability to integrate existing assets	
Virtualization	Improved reliability Ability to scale operations to meet different demand levels	
Model-driven Implementation	Ability to develop new functions rapidly	Model-implementation environment

Table 1: SOA Features, Benefits, and Infrastructure

1.4.2 Service

Service is the essential concept of SOA.

It is not originally a technical concept. The idea of a service was developed in the world of business. Look in any "Yellow Pages" directory, and you will find categories such as "courier services", "garage services", and "roofing services". For each of these, some person or company (the service provider) is offering to do something – carry goods and messages, look after vehicles, install and repair building roofs – that will benefit other people or companies (the service consumers). The providers offer to contract with the consumers to do these things, so that the consumers know in advance what they will get for their money.

The idea has been adopted by technologists. They have established the concept of a *software service*. A software service is performed by a software program. It produces effects that have value to the people or organizations that are its consumers. It has a provider – a person or organization that takes responsibility for running the program to produce those effects. And there is an implicit or explicit contract between the provider and the consumers that the program will produce the effects that the consumers expect.

Software services can be provided over the Internet and the world-wide web. In some countries, for example, the government provides a service by which taxpayers can complete and submit their tax returns via the web. Here,