



1

Service Oriented Architecture Reference Model

2

3

4

Working Draft 06, 03 May 2005

5 **Document identifier:**

6 wd-soa-rm-06

7 **Location:**

8 <http://www.oasis-open.org/>

9 **Editors:**

10 *C. Matthew MacKenzie, Adobe Systems Incorporated, mattm@adobe.com*

11 *Christopher Bashioum, Mitre Corporation, cbashioum@mitre.org*

12 *Ken Laskey, Mitre Corporation, klaskey@mitre.org*

13 *Wesley McGregor, GoC(ed: ?), McGregor.Wesley@tbs-sct.gc.ca*

14 *Francis McCabe, Fujitsu (Soft? Consult?), fgm@fla.fujitsu.com*

15 *Don Flinn, Individual, flinn@alum.mit.edu*

16 *Peter Brown, Individual, peter@justbrown.net*

17 *Vikas Deolaliker*

18

19

20 **Abstract:**

21 This Service Oriented Architecture Reference Model is an abstract framework for
22 understanding significant entities and relationships amongst them within a service-
23 oriented environment, and for the development of consistent standards or specifications
24 supporting that environment. It is based on unifying concepts of SOA and may be used
25 by architects developing specific services oriented architectures or for education and
26 explaining SOA. A reference model is not directly tied to any standards, technologies or
27 other concrete implementation details, but it does seek to provide a common semantics
28 that can be used unambiguously across and between different implementations.

29

30 While service-orientation may be a popular concept found in system a broad variety of
31 applications, this reference model scopes itself to the field of software architecture.

32 **Status:**

33 This document is updated periodically on no particular schedule. Send comments to the
34 editor(s).

35 Committee members should send comments on this specification to the [soa-
36 rm@lists.oasis-open.org](mailto:soa-rm@lists.oasis-open.org) list. Others should visit the SOA-RM TC home page at
37 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm, and record
38 comments using the web form available there.

39

40 For information on whether any patents have been disclosed that may be essential to
41 implementing this specification, and any offers of patent licensing terms, please refer to
42 the Intellectual Property Rights section of the SOA-RM TC web page at:
43 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm
44
45 The errata page for this specification is at:
46 http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

Table of Contents

48	1 Introduction	5
49	1.1 Audience	5
50	1.2 How to Use the Reference Model	5
51	1.3 Notational Conventions	6
52	1.4 Relationships to Other Standards	6
53	2 The Reference Model	8
54	2.1 Services.....	9
55	2.1.1 Service Composition.....	10
56	2.1.2 Service Description.....	11
57	2.2 Policies and contracts	11
58	2.2.1 Service Policy	11
59	2.2.2 Service Contract	12
60	2.3 Semantics.....	14
61	2.3.1 Data/Information Model	15
62	2.4 Discovery, Presence and Availability	15
63	3 Conformance Guidelines	17
64	4 References.....	18
65	4.1 Normative.....	18
66	Appendix A. Glossary	19
67	Appendix B. Use Cases and Examples (Non-Normative).....	23
68	Appendix C. Acknowledgments.....	24
69	Appendix D. Notices	25
70		

71 **1 Introduction**

72 The service-oriented architecture (SOA) paradigm has received significant attention within the
73 software design and development industry in recent times resulting in many conflicting definitions
74 of service-oriented architecture. The goal of this reference model document is to define the
75 essence of the service oriented architecture paradigm, and emerge with a vocabulary and a
76 common understanding of SOA.

77

78 This document explicitly avoids defining implementation detail, as doing so would unnecessarily
79 constrain and date the reference model. The goal is to provide a document that can stay relevant
80 through the various technology evolutions that we experience in this industry.

81

82 A reference model cannot be implemented, nor should it be. A reference model is a foundational
83 work that can and should be used to develop architectural patterns and promote effective
84 discourse on derived works.

85 **1.1 Audience**

86 The intended audiences of this document non-exhaustively include:

- 87 • Architects and developers designing, identifying or developing a system based on the
88 service-oriented paradigm.
- 89 • Standards architects / analysts developing specifications that relate to or make use of the
90 service-oriented paradigm.
- 91 • Chief Information Officers and other decision makers seeking a "consistent and common"
92 understanding of service oriented architecture.

93

94 **1.2 How to Use the Reference Model**

95

96 New readers are encouraged to read this reference model in its' entirety, from beginning to end.
97 Concepts are presented in an order that the authors hope promote understanding, quickly.

98

99 Section 0 introduces the conventions and sets the stage for the rest of the document. Non -
100 technical readers are encouraged to read this information as it provides background material
101 necessary to understand the nature of reference models and their use.

102

103 Section 2 introduces the service oriented reference model. A brief overview of the components
104 and their relationships is given. The following subsections delve into greater detail on each
105 component, including their externally visible properties and relationships to each other. This
106 section is provided for the benefit of multiple audiences. Non-technical readers may use this
107 section to gain an explicit understanding of the core principles of SOA.

108

109 Architects are encouraged to use this section as guidance for developing specific service oriented
110 architectures. Section 2 and its subsections are designed to provide guidance for consistent
111 logical divisions of components within architectures. It also helps architects adhere to the basic
112 principles of service-oriented design.

113

114 Section 3 aims to provide guidelines for conformance with the reference model and is aimed at
115 those who wish to explicitly state that their architectures are conformant with this reference
116 model.

117 The appendixes provide several non-normative examples and a glossary to provide clarity of
118 terms whose use may otherwise be ambiguous.

119

120 **1.3 Notational Conventions**

121 The key words *must*, *must not*, *required*, *shall*, *shall not*, *should*, *should not*, *recommended*, *may*,
122 and *optional* in this document are to be interpreted as described in **[RFC2119]**.

123

124 References are surrounded with **[square brackets and are in bold text]**.

125 **1.4 Relationships to Other Standards**

126

127 Due to its nature, this reference model may have an implied relationship with any group that:

- 128
- Considers its' work "Service Oriented"; and/or

129 • Makes (publicly) an adoption statement to use this SOA Reference Model of this TC as a
130 base or inspiration for their work when complete.

131

132 Additionally, there are a large number of standards and technologies that are related by the fact
133 they claim to be or are “service oriented”.

134 Any work that aligns with the functional areas of SOA such as the service, service description,
135 advertising mechanism, service data model or service contract are likely to be directly related.

136

137 The reference model does not endorse any particular service-oriented architecture, or attest to
138 the validity of third party reference model conformance claims.

139

140

2 The Reference Model

141

[ed: This text was provided by Duane, and does require some stylistic rework.]

142

Figure 1 - SOA Architectural Model introduces the core service oriented architecture reference model and its high level components. Services are the fundamental base component of service-oriented architectures. Each Service has a Service Description. A Service Description is a set of metadata declaring all aspects of a service necessary for a Service Consumer to understand the service's externally inspect-able aspects. A Policy is a set of assertions that must be adhered to when a service is invoked. A Contract is implied when a Service Consumer makes and invocation request to a service, in substantial alignment with the Policy declaration.

149

150

A Data Model is the abstract paradigm used in the invocation and consumption of a Service. A Data Model will likely manifest itself within a concrete architecture as a set of concrete Messages.

151

152

153

Semantic agreement on what entities mean with respect to their roles in a system is necessary for service-oriented architecture. Many of the components (Service Descriptions, Policies, Contracts and Data Models) need to be available for discovery by potential service consumers to determine both the suitability of a service and their ability to invoke and/or consume the service. The concept of Discovery is to gain awareness of the Presence of the elements and details of their availability.

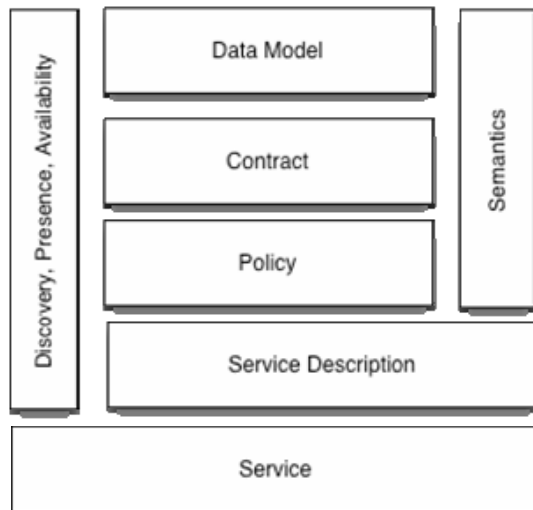
154

155

156

157

158



159

160 *Figure 1 - SOA Architectural Model*

161

162 **2.1 Services**

163

164 A Service is a set of functionality provided by one entity for the use of others.

165

166 There is no need to make architectural distinctions between services that are consumed as part
 167 of a process vs. ones that are not.

168

169 There is not a one to one correlation between requests to invoke a service and instances of a
 170 service being consumed.

171

172 Opacity is a core component of services.

173

174 **2.1.1 Service Composition**

175

176 Since services are opaque, a Service Consumer cannot see anything beyond it. If one service is
177 actually consuming and aggregating two other services, the Service Consumer cannot and
178 should not know such. Whether a Service's functions are mapped to a set of classes in some
179 native language or another service is not important or relevant (other than the service metadata
180 stating what invoking the service means or does)

181



182

183 *Figure 2 - Service Composition*

184

185 Examining Figure 2 - Service Composition above, the service function (for service A) is described
186 in the service description specific to that service. If completing the function depends on two or
187 more serial or parallel paths of execution successfully completing behind the service interface
188 (like calling services B and C) within a certain time frame, that is not relevant to state in the
189 service description for service A. The service consumer is only concerned with the service's
190 ultimate success or failure. Mapping the functionality to success and failure is the responsibility of
191 the service provider. This is necessary to preserve the axiom of opacity.

192

193 The functionality described above is mandatory to comply with the notion of service autonomy. A
194 service alone must determine whether an invocation request succeeds or fails.

195

196 Note (non-normative) If a service consumer can see any specifics behind the service, this violates
197 several of the core principles of SOA. If visibility beyond the offered service is required, then the
198 service does not meet the demand of the service consumer. Accordingly, the service provider and
199 consumer should discuss and re engineer the service.

200

201 Note

202 When implementing, more complex patterns of service invocation can be
203 facilitated while keeping these three axioms. If a transaction sequence is needed,
204 a service interface can offer two services - a put() and a commit().

205

206 **2.1.2 Service Description**

207 Each logical Service has exactly one canonical Service Description.

208

209 A Service Description is comprised of three parts:

210

- 211 a. Data Model - The logical expression of a set of information items associated with the
212 consumption of a service or services;
- 213 b. Policy - Assertions and obligations that service consumers and/or providers must adhere
214 to or provide; and
- 215 c. Contract (and/or offer thereof) - the syntactic, semantic and logical constraints governing
216 on the use of a service.

217

218

219 **2.2 Policies and contracts**

220

221 Broadly speaking, a policy represents some form of constraint or condition on the use,
222 deployment or description of an owned entity. Policies are inherently unilateral – any participant
223 may have policies about issues that are important to them. A contract, however, is a policy that
224 has been agreed to.

225 Where a contract can refer to everything from the detailed description of the service interface to
226 the legal contract entered into when two or more parties use a service. However, the SOA RM
227 focuses on those agreements necessary for a successful interaction with a service.

228 **2.2.1 Service Policy**

229 Abstractly, a policy is an assertion that expresses intent on the part of a participant.

230 Policies apply to many aspects of SOAs: to security, to privacy, manageability, Quality of Service
231 and so on.

232

233 Policy assertions may be, but need not be, written down in a formal machine processable form.
234 Languages that permit policy assertions also range in expressivity from simple propositional
235 assertions to modal logic rules. However, the SOA RM is neutral to how a policy is represented.

236

237 A natural point of contact between service participants and policies associated with the service is
238 in the service description. It would be natural for the service description to contain references to
239 the policies associated with the service.

240

241 Associated with policies is the concept of enforcement. Enforcement is the realization of the
242 policy: an un-enforced policy is simply an abstract logical proposition. However, how a policy is
243 enforced, or even whether a policy is enforced is not a relevant part of the reference model.

244

245 A policy always represents a participant's point of view. For example, a provider of a service may
246 have a policy that all users of the service must be authenticated prior to their access to certain
247 functions. This policy is one that may be enforced by the service provider independently of any
248 agreement from potential users of the service. Similarly, someone's agent may embody a privacy
249 policy independently of any services the agent interacts with.

250 **2.2.2 Service Contract**

251

252 Where a policy represents an assertion from the point of view of a participant, a contract
253 represents an agreement between two or more participants. Like policies, contracts can cover a
254 wide range of aspects of services: quality of service agreements, interface and choreography
255 agreements and commercial agreements. However, the concept of a service contract within the
256 SOA RM applies primarily to the requirements for the successful use and provision of services.

257

258 A contract may be, but need not be, expressed in a machine process-able form. It seems
259 significantly likely that an executed contract will not be in a machine process-able form; especially
260 for commercial agreements. However, languages that can express policies, especially the more
261 powerful variants can often also be used to express machine process-able contracts.

262

263 Each contract may be associated with a life-cycle. This life-cycle has three main phases: a
264 negotiation phase, an active phase and a completion phase.

265

266 While it is possible that a specific negotiation phase precedes an agreement to a contract, often it
267 is more implicit. For example, merely attempting to interact with a service may represent an
268 agreement to follow the prescribed procedures for using the service.

269

270 Often a contract specifies policies that are assumed to be in force during the active phase of the
271 contract. As such, those policies are subject to enforcement in a similar that unilateral policies
272 are.

273

274 Enforcement of an agreement will depend on the nature of the agreement: violating an
275 infrastructure-level agreement is likely to lead to errors and unexpected results. Violating a
276 commercial agreement is likely to lead to loss of service or other legal remedies.

277

278 While there may be many kinds of contract, we envisage three main kinds of contract that may
279 apply in service oriented architectures: the contracts that represent the valid use and provision of
280 services, the contracts that represent the permitted uses of services and the contracts that result
281 from using services.

282 For example, the service description may contain descriptions of the interfaces of a service – the
283 kinds of data entities expected and the names of the operations supported – and may also
284 contain choreographic descriptions of the order of interactions. Such descriptions may range from
285 simple identifiers implying a mutually understood protocol to a complete description of the
286 vocabularies, expected behaviors and so on.

287

288 However, a valid use of a service is not equivalent to a permitted use of the service. For example,
289 one may present a syntactically correct request to a service for withdrawing money from an
290 account. If that request is not accompanied by a suitable authentication, then that request is
291 typically denied – it is not permitted. Many security considerations and quality of service
292 considerations lie in this realm of agreement.

293

294 Often the purpose of interacting with a service is to effect a further agreement. For example, one
295 use of a book-selling service is to cause a book to be purchased and delivered.

296 This kind of contract is an important aspect of the rationale for deploying Service

297 Oriented Architectures; however, such contracts are beyond the scope of this SOA RM.

298 2.3 Semantics

299

300 The semantics of a service are the shared expectations about the service. Fundamentally, we
301 expect that all services deployed in a SOA have an intended purpose. That purpose is the
302 linchpin by which we measure the expectations for a service and is the basis of its semantics.
303 The purpose of a service is the highest level semantic characterization of the service.

304 In principle, the semantics of a service many aspects of its establishment – from the format and
305 structure of any data communicated between the participants of a service interaction to the
306 stateful requirements on the participants to the expected effects of successfully interacting with
307 the service.

308

309 One of the hallmarks of a Service Oriented Architecture is the degree of documentation
310 associated with it. The purpose of this metadata is to facilitate integration, particularly across
311 ownership domains. By providing descriptions, the task of designing client applications that make
312 use of a service is considerably enhanced.

313

314 In this spirit, we might also expect that the semantic aspects of a service may also be
315 documented. Such documentation will, in principle, be layered into several levels:

316

- 317 • The metadata required to reliably contact the service and to establish communication
318 with it. In Web Services, this role may be filled by descriptions using the WS-Reliability
319 specification.
- 320 • The metadata required to reliably format data for interchange between service
321 participants.
- 322 • In Web Services, this role may be filled by WSDL documents.
- 323 • The metadata required to reliably sequence operations of the service. Documents using
324 specifications such as WSBEPL and CDL are oriented towards such requirements.
- 325 • The metadata required to adequately measure the effect of using a service and of the
326 requirements of the participants. Often, this is the kind of description labeled as
327 semantic, although, in reality, all the above documents represent descriptions of the
328 semantics of the service – albeit at different levels of abstraction.

329 • There may also be documents that relate to any policies governing the service and to
330 any agreements and contracts associated with the service. Such documents may range
331 in scope from simple technical policies to legal contracts valid in international law.

332 If documented in metadata, a service's semantics has many possible uses: it can be used as a
333 basis of discovery in dynamic systems, it can assist in managing a service, validating and
334 auditing uses of services may also be simplified by rich metadata.

335 However, it is not essential to the concept of SOAs that the semantics of a service be so
336 completely described.

337

338 **2.3.1 Data/Information Model**

339

340 The goal of SOA Data/Information Model is to specify an abstract interface and data model for
341 exchange of data among SOA entities. Entities in SOA need a standard way to (de)serialize data,
342 extract and/or construct metadata, and infer service semantics.

343

344 At the highest level data in SOA can be classified as private and public. Private data includes the
345 data used by a service. The data model of this data is private to the service implementation. This
346 data model does not provide a mapping or bridge to private data. In an exchange this type of data
347 is carried as payload inside of an exchange data unit.

348

349 Public data includes data that embodies the state, property and parameters of an SOA. Public
350 data should be available at standard interfaces and in standard formats.

351

352 The data model should also define standard mechanisms for services to extract metadata that
353 may be serialized with data.

354

355 **2.4 Discovery, Presence and Availability**

356

357 [ed: title was changed, text below needs to jive.]

358

359 The main concept is a methodology or mechanism to convey awareness of (the existence of) a
360 service(s) to all consumers on a fabric.
361
362 Advertising makes discovery possible.
363
364 A Service Description is advertised to consumers on a fabric to make it discoverable.
365
366 Discovery does not constitute authorization to execute against the service.
367
368 [from W3C WSA] Discovery is the act of locating a resource description
369
370 Discovery involves matching a set of functional and other criteria with a set of resource
371 descriptions.
372
373 Discovery may be performed by an agent, or by an end-user
374
375 Discovery may be realized using a discovery service [end W3C WSA]
376
377

378

3 Conformance Guidelines

379

380 The authors of this reference model envision that architects may wish to declare their architecture
381 is conformant with this reference. In order to be conformant to this reference model, a mapping
382 must be made from each core element of this reference model to components of the conformant
383 architecture.

384 **4 References**

385 **4.1 Normative**

386 **[RFC2119]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
387 <http://www.ietf.org/rfc/rfc2119.txt>, IETF RFC 2119, March 1997.
388

389

Appendix A. Glossary

390

Several terms are used within this Reference Model are also used in other specifications. This glossary locally scopes the semantics of those terms where ambiguity exists or overrides those definitions.

391

392

393

394

Advertising

395

396

A methodology to convey awareness of (the existence of) a service(s) to all consumers on a fabric. Advertising makes discovery possible.

397

398

399

Agent (requester or provider)

400

401

An entity acting on behalf of another entity to fulfill a task.

402

403

Architecture

404

405

Software architecture for a system is the structure or structures of the system, which consist of elements and their externally visible properties, and the relationships among them.

406

407

408

Service Consumer

409

410

An entity which makes use of a service.

411

412

Contract

413

414

The syntactic, semantic and logical constraints governing on the use of a service.

415

416

Data Model

417

418 The logical expression of a set of information items associated with the consumption of a service.

419

420 **Discovery**

421

422 The act of gaining knowledge of a logical service, its existence and details of how to use it.

423

424

425 **Interface**

426

427 A named set of operations that characterize the behavior of an entity.

428

429 **Message**

430

431 A serialized set of data that is used to convey a request or response from one party to another.

432

433 **Policy**

434

435 Policy is a statement of obligations, constraints or other conditions of service use.

436

437 A contract is formed when a specific set of entities accept a policy.

438

439 **Requester or provider**

440

441 Person or organization involved in an SOA transaction an agent that interacts with a service in
442 order to achieve a goal

443

444 **Security**

445

446 Computer security is the effort to create a secure computing platform, designed so that agents
447 (users or programs) can only perform actions that have been allowed. This involves specifying
448 and implementing a security policy. The actions in question can be reduced to operations of
449 access, modification and deletion. Computer security can be seen as a subfield of security
450 engineering, which looks at broader security issues in addition to computer security. (from
451 Wikipedia)

452

453 **Semantics**

454

455 Shared conceptualization of the implied meaning of information. Represents a contract governing
456 the meaning and purpose.

457

458 **Service**

459

460 A behavior, or set of behaviors provided for use by another entity.

461

462

463

464

465 **Service description**

466

467 A specification of the information necessary to a) allow a potential consumer to determine
468 whether or not this service is applicable, and b) facilitate invocation.

469

470 **Service Oriented Architecture (SOA)**

471

472 A form of Enterprise Architecture. The difference between Enterprise Architecture and SOA lies
473 mostly in the fact that EA is specific to an enterprise, while SOA can be abstracted out of a given
474 Enterprise, and collected along with other SOA components so abstracted to form a registry of
475 available services SOA is potentially a specialization of a combination of many things - interface
476 based design (IBD), component architecture (CA), OO methodology etc.

477

478 **Service Oriented Architecture Reference Model (SOA-RM)**

479

480 A reference model is an abstract framework for understanding significant relationships among the
481 entities of some environment, and for the development of consistent standards or specifications
482 supporting that environment. A reference model is based on a small number of unifying concepts.
483 A reference model is not directly tied to any standards, technologies or other concrete
484 implementation details, but it does seek to provide a common semantics that can be used
485 unambiguously across and between different implementations. Is not architecture for a single
486 implementation. Is a model for developing a range of Service Oriented Architectures and
487 analysis/comparison thereof. Is a framework for understanding significant relationships among
488 the entities in an SOA environment. DISCUSSION POINT: should the word "elements" be used in
489 place of "entities" above? Is based on a small number of unifying concepts of all SOAs. A
490 Reference Model is the best mechanism to define SOA.

491

492 **Appendix B. Use Cases and Examples (Non-**
493 **Normative)**
494

495 **Appendix C. Acknowledgments**

496 The following individuals were members of the committee during the development of this
497 specification:

498 [TODO: insert cte. Members]

499

500

Appendix D. Notices

501 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
502 that might be claimed to pertain to the implementation or use of the technology described in this
503 document or the extent to which any license under such rights might or might not be available;
504 neither does it represent that it has made any effort to identify any such rights. Information on
505 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
506 website. Copies of claims of rights made available for publication and any assurances of licenses
507 to be made available, or the result of an attempt made to obtain a general license or permission
508 for the use of such proprietary rights by implementers or users of this specification, can be
509 obtained from the OASIS Executive Director.

510 OASIS invites any interested party to bring to its attention any copyrights, patents or patent
511 applications, or other proprietary rights, which may cover technology that may be required to
512 implement this specification. Please address the information to the OASIS Executive Director.

513 **Copyright © OASIS Open 2005. All Rights Reserved.**

514 This document and translations of it may be copied and furnished to others, and derivative works
515 that comment on or otherwise explain it or assist in its implementation may be prepared, copied,
516 published and distributed, in whole or in part, without restriction of any kind, provided that the
517 above copyright notice and this paragraph are included on all such copies and derivative works.
518 However, this document itself does not be modified in any way, such as by removing the
519 copyright notice or references to OASIS, except as needed for the purpose of developing OASIS
520 specifications, in which case the procedures for copyrights defined in the OASIS Intellectual
521 Property Rights document must be followed, or as required to translate it into languages other
522 than English.

523 The limited permissions granted above are perpetual and will not be revoked by OASIS or its
524 successors or assigns.

525 This document and the information contained herein is provided on an "AS IS" basis and OASIS
526 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO
527 ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE
528 ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
529 PARTICULAR PURPOSE.