



United States
Department of Justice

U.S. Department of Justice's Global Justice Reference Architecture (JRA)

Specification

JRA

Version 1.7

November 18, 2008

Global Infrastructure/Standards
Working Group



This project was supported by Grant No. 2007-NC-BX-K001 awarded by the Bureau of Justice Assistance, in collaboration with the U.S. Department of Justice's Global Justice Information Sharing Initiative. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the U.S. Department of Justice.

Table of Contents

1		
2	Acknowledgements	ii
3	How to Use This Document	iii
4	Policymakers, Executives, and Decision Makers	iii
5	Project Managers, Architects, and Technologists.....	iii
6	Document Conventions.....	iv
7	Executive Summary	v
8	1. Introduction.....	1
9	1.1. Global’s SOA Initiative	1
10	1.2. An Interoperability Strategy	2
11	1.3. Consensus on the OASIS Reference Model for SOA	3
12	1.4. Creating the JRA.....	4
13	1.5. What Is the JRA?	5
14	1.6. What the JRA Is Not.....	5
15	2. Architecture Requirements	6
16	3. The JRA.....	12
17	3.1 Graphical Overview	12
18	4. Concepts and Relationships	14
19	<i>OASIS Reference Model for Service-Oriented Architecture</i>	14
20	<i>Core Concepts—Services, Service Consumers, Capabilities, and Real-World Effects</i> ...	15
21	<i>Supporting Concepts</i>	16
22	<i>Interaction, Visibility, Service Models, and Service Interfaces</i>	16
23	<i>Design and Description of Service Interfaces</i>	21
24	<i>Capabilities in Detail</i>	22
25	<i>Service Policy, Service Contract, and Service Agreement</i>	24
26	<i>Execution Context</i>	25
27	<i>Provisioning Model</i>	25
28	5. Reconciliation of Architecture With Principles.....	26
29	6. Elaboration of Service Interaction Requirements	29
30	7. Glossary	31
31	8. References.....	37
32	9. Document History	38

Acknowledgements

The Justice Reference Architecture (JRA) was developed through a collaborative effort of the Global Justice Information Sharing Initiative (Global), Office of Justice Programs (OJP), U.S. Department of Justice (DOJ).

Global aids its member organizations and the people they serve through a series of important initiatives. These include the facilitation of Global Working Groups. The Global Infrastructure/Standards Working Group (GISWG) is one of five Global Working Groups covering critical topics such as intelligence, privacy, security, outreach, and standards. The GISWG is under the direction of Tom Clarke, Ph.D., National Center for State Courts. The GISWG consists of three committees—Management and Policy, Services Implementation, and Enterprise Architecture.

Although this document is the product of Global and its GISWG membership, it was adapted primarily from the technical reference architecture developed by the state of Washington, and sincere appreciation is expressed to Mr. Scott Came, state of Washington and SEARCH, The National Consortium for Justice Information and Statistics, for his guidance and leadership. In addition, parts of the architecture were derived from the Organization for the Advancement of Structured Information Standards (OASIS) Reference Model for Service-Oriented Architecture 1.0 (SOA-RM). Other major contributors include the OASIS Court Filing Technical Committee, OASIS SOA-RM Technical Committee, and the Messaging Focus Group.

Although all members of the GISWG are recognized for their contributions and for volunteering their time to the development of the architecture, Global would also like to recognize the members of the GISWG Executive Architecture Committee.

Mr. Scott Came—State of Washington and SEARCH, The National Consortium for Justice Information and Statistics, GISWG Services Implementation Committee

Dr. Tom Clarke—National Center for State Courts, Chair, GISWG

Mr. Scott Fairholm—National Center for State Courts, Chair, GISWG Services Committee (2005–2008)

Mr. Dale Good—Judicial Council of California, Chair, GISWG Management and Policy Committee

Mr. Kael Goodman—IJIS Institute, Chair, GISWG Services Interaction Committee (2005–2007)

Mr. Ron Hawley—SEARCH, The National Consortium for Justice Information and Statistics, GISWG Management and Policy Committee Chair (2005–2006)

Mr. Eric Sweden—National Association for State Chief Information Officers, Vice Chair, GISWG (2005–2008)

How to Use This Document

Policymakers, Executives, and Decision Makers

Global is committed to providing Service-Oriented Architecture (SOA) resources, such as this document, to local, state, regional, tribal, and federal justice and public safety organizations. As additional resources become available, these materials will demonstrate the value of the architecture to the stakeholders in a way that is targeted to their particular needs. Other planned resources include strategy, executive summary, case studies from early implementers, management and policy, and other planning briefings, which will target managers, chiefs, and executives.

For the purposes of this document, Global has selected a distinguished group of technical and domain representatives from a group of skilled peers who have volunteered to develop this material as a starting point in establishing the Justice Reference Architecture (JRA).

Keep in mind that the sections in this document referencing the conceptual diagram, high-level components, and relationships establish definitions that are intended for use by technical architects and project managers who are responsible for identifying all the elements necessary within their jurisdictions to implement SOA. **This document is intended as a formal and complete architectural specification for people with previous knowledge of technical architecture, service-oriented architecture, and supporting industry standards (such as Web services).**

Project Managers, Architects, and Technologists

This report is intended as a resource for a technical audience, including Global Justice XML Data Model (Global JXDM) and National Information Exchange Model (NIEM) implementers, architects, developers, system integrators, and other justice and public safety technical practitioners.

It provides the background and concepts—a strong foundation—required for the implementation of SOA. The JRA is a new term coined for the justice community, and it is derived from the OASIS Reference Model for Service-Oriented Architecture 1.0 [SOA-RM]. The reader should refer to the SOA-RM for more detailed information about many of the concepts in this document. JRA is intended to facilitate your SOA implementation by establishing a common language that can be used to exchange data with partner organizations.

106

Document Conventions

107 In this document, use of a bold small-caps typeface, as in this **EXAMPLE**, indicates an
108 important concept or a term defined either in the glossary or in the body of the text
109 at the point where the term or concept is first used.

110

111 In this document, use of a bold caps typeface, as in this **[EXAMPLE]**, indicates an
112 important resource document noted in the Reference Section of this document.

113

Executive Summary

114

115 In 2004, Global endorsed service-oriented architecture (SOA) as a recommended
116 strategy for integrating justice information systems. This document—the Justice
117 Reference Architecture Specification—is a first step towards achieving this vision.

118 SOA promises many benefits to state, local, and tribal justice partners. It promotes
119 the sharing of information in a manner that maximizes agility—the ability of partners
120 to change business processes and technology solutions rapidly at minimum cost. In
121 today’s dynamic justice business environment, this is more important than ever. It
122 also gives justice partners a set of tools that allow them to share infrastructure by
123 identifying where interoperability is important, thus enabling them to make smart
124 investments that stretch every dollar. Finally, SOA offers the promise of an over-
125 arching umbrella framework that demonstrates how all of Global’s work products fit
126 together as a cohesive approach to improving information sharing.

127 While recognizing these benefits, it is also important to recognize that SOA is not
128 trivial to implement, especially if practitioners do not share lessons learned and best
129 practices across jurisdictions. The cost of reimplementing SOA from scratch in every
130 state, county, municipality, and tribal organization in the United States would be
131 overwhelming. The JRA aims to solve this problem by providing practitioners with a
132 set of documents that represent the national justice community’s very best practices,
133 experiences, and lessons learned from implementing SOA. A state, local, or tribal
134 integration architect or project manager can start with these documents rather than
135 starting from nothing, dramatically accelerating his or her jurisdiction’s path to SOA.
136 Along the way, the JRA will lead the jurisdiction to adoption of the other products
137 that Global and its partners have developed.

138 This document—the JRA Specification—is a conceptual framework for SOA that is
139 based on an industry standard, the OASIS SOA Reference Model, which was
140 developed by a committee of industry and government SOA experts, including some
141 of the GISWG members who authored the JRA. The Specification defines a set of
142 key concepts in a standard way, so that across the country, justice practitioners and
143 their industry partners can adopt a consistent vocabulary for communicating about
144 SOA. The framework also provides a jumping-off point for the rest of the broader
145 reference architecture, by identifying areas where the community needs more
146 thorough standards and guidelines. Separate documents within the JRA elaborate
147 these concepts, which include:

- 148 • A methodology for identifying what services—exchange points—a
149 jurisdiction should develop to solve some identified business
150 problem
- 151 • A standard for describing services so they can be used, understood,
152 and consumed across jurisdictions

- 153 • Recommended requirements for infrastructure necessary to support
154 SOA
- 155 • Technical communications protocols, based on industry standards
156 such as Web services and XML, for transmitting information as
157 messages between justice partners and their systems
- 158 • Guidelines for governing and managing an SOA in a jurisdiction—
159 how to assign decision rights and responsibilities for implementing
160 elements of an SOA

161 If you are an executive-level decision-maker without direct day-to-day management
162 responsibilities over technology, you should view this document (and the remainder
163 of the JRA) as important guidance for your technology staff to follow as you plan (or
164 participate in planning) information sharing in your jurisdiction. Even if you are not
165 technically oriented, you still have ultimate accountability for the wise investment of
166 public funds in your community, and you should be aware of the JRA's power to
167 lead you and your partners to an agile, standards-based, shared approach to
168 information sharing.

169 If you are a chief information officer, architect, senior project manager, or other
170 technology leader responsible for implementation of information sharing solutions,
171 the JRA holds the promise of saving you a great deal of time, effort, and money in
172 implementing the best practices inherent in SOA. This document is primarily for
173 you.

174

175

176

177

178

179

180

181

182

183

184

185

1. Introduction

1.1. Global's SOA Initiative

On September 29, 2004, the Global Justice Information Sharing Initiative (Global) Advisory Committee (GAC) unanimously adopted **SERVICE-ORIENTED ARCHITECTURE** (SOA) and the recommendations in the report titled *A Framework for Justice Information Sharing: Service-Oriented Architecture (SOA)*. **[SOA-REC]**

Global provides support for SOA by:

- Recognizing SOA as the recommended **FRAMEWORK** for development of justice information sharing systems
- Promoting the utility of SOA for the justice community
- Encouraging the members of the justice community to take these recommended incremental steps in the development of their own systems

Global's approval was based on the understanding that SOA is the approach most likely to result in an infrastructure that will support its vision of how information should be shared within the justice community. If SOA is to be used successfully as the framework for justice information sharing **ARCHITECTURE**, Global must play a proactive leadership role in several areas. The development of the **JUSTICE REFERENCE ARCHITECTURE** was based on the following actions recommended by Global:

- Incorporate SOA into the activities of all Global Working Groups. SOA raises issues for security, privacy and information quality, and intelligence that will be given explicit attention and treated as part of a broad initiative.
- Encourage the creation of a mechanism for drawing together the experiences and lessons from the field.
- Reach out to existing national systems to incorporate their efforts into the design of an overall strategy.
- Address the following six issues as priorities—services, standards, interagency agreements, registries, security, and privacy and data quality—because they will be a major part of the agenda for the next set of Global activities.
- Develop a multitiered strategy for the public sector to influence standards. It will include encouraging the creation of a public process (as it did with XML), taking part in industry groups that are developing standards relevant to justice (e.g., OASIS), and developing partnership processes with industry and other public entities.

227 1.2. An Interoperability Strategy

228 Solving interoperability challenges continues to be a significant problem and a high
229 priority for the justice and public safety community. Approximately 100,000 justice
230 agencies have the critical need to share information across their various information
231 systems, and this variety creates multiple layers of interoperability problems because
232 hardware, software, networks, and business rules for data exchange are different.
233 The need for information sharing has led to this interoperability strategy and the
234 JRA.

235 The strategy for developing JRA involves many steps. This paper details some
236 highly technical and abstract concepts. Understanding these concepts may require
237 significant effort from the reader. Though it may seem strategically questionable to
238 place such a high hurdle at the beginning of a multistep process, doing so actually
239 creates a flexible vocabulary and a conceptual framework that will enable the
240 desired interoperability to flourish. Additionally, subsequent steps that will build
241 from this framework will be incrementally more concrete and will ultimately lead to
242 actual implementation specifications that can be used by practitioners in the field.
243 Global believes that this dynamic interoperability strategy will help to prevent
244 incompatibilities, guide vendors and organizations on how to fit components
245 together, and facilitate communication and interoperability among disparate
246 communities.

247 Global's strategy for JRA, like other work that has preceded it, follows a five-step
248 process:

- 249 Step One: Agree on common concepts
- 250 Step Two: Agree on the relationships and deliverables
- 251 Step Three: Assign the work
- 252 Step Four: Produce the deliverables
- 253 Step Five: Revise the deliverables

254 As an example, when the Global JXDM project started, it had a small set of limited
255 solutions. Through much iteration, Global JXDM has been expanded and refined
256 and addresses a successively larger set of justice domains.

257

258 **1.3. Consensus on the OASIS Reference Model for SOA**

259 One of the justice requirements is to create a common language for talking about
260 architecture across major domains. For instance, it is currently difficult for
261 emergency management personnel to talk to justice personnel about how their
262 respective systems might share data beyond the content standards issue because
263 their ways of communicating about architecture are so different.

264 After considerable discussions among the stakeholders, Global adopted the
265 Organization for the Advancement of Structured Information Standards (OASIS)
266 Reference Model for Service-Oriented Architecture 1.0 **[SOA-RM]**. OASIS has
267 approved this standard reference model for describing different architectures using
268 comparable, vendor-neutral language. Global is adopting the OASIS framework for
269 describing its architecture and holding conversations with other domains.

270

271 **1.4. Creating the JRA**

272 It is important to note that SOA-RM provides a conceptual foundation not only for
273 the justice community but also for any other domain to create a **REFERENCE**
274 **ARCHITECTURE**. JRA builds on the SOA-RM concepts by specifying additional
275 relationships and defining and specifying these adopted concepts.

276 Although there is no perfect solution and since there is a need to start somewhere,
277 SOA-RM is recommended as the best place to start Global's SOA work efforts.
278 Global began by mapping the SOA components, documenting, and leveraging the
279 work that has been done already—like the Global JXDM—and finally, worked to
280 identify and fill the gaps.

281
282
283
284
285
286
287

Justice Reference Architecture is derived from the OASIS Reference Model for Service-Oriented Architecture 1.0. The OASIS work was developed to provide a conceptual foundation for creating a reference architecture. As intended by OASIS, the JRA builds on or expands from the OASIS model.

288 Specifically, Global is developing a modular architecture that clearly and
289 appropriately identifies and separates technical and governance layers so that
290 standards can be developed to improve interoperability.

291

292 1.5. What Is the JRA?

293 This section defines the JRA and explains why a reference architecture is useful.
294 Keep in mind that there are many potential justice reference architectures but that
295 the JRA focuses entirely on SOA for the justice and public safety community.

296 **JRA is an abstract framework for understanding significant**
297 **components and the relationships between them within a**
298 **Service-Oriented Architecture. It lays out common concepts**
299 **and definitions as the foundation for the development of**
300 **consistent SOA implementations within the justice and public**
safety communities.

301 The JRA is a description of the important concepts in a justice information sharing
302 architecture and of the relationships between those concepts. The JRA also
303 identifies, at a high level, the kinds of components (software systems, hardware
304 infrastructure, policies, practices, intersystem connections, and so on) necessary to
305 bring those concepts to life in a particular context. The JRA is generally not specific
306 enough to govern the implementation of any individual software system
307 implementation. Rather, it is a framework for guiding implementations in general,
308 with the aim of standardizing or harmonizing certain key aspects of those
309 implementations to support reusability or interoperability.

310 It is important to note that at this time, the JRA is not complete. Many sections of
311 this document are still under development, but the document does attempt to
312 identify the necessary concepts, relationships, and components that will require
313 further elaboration and/or implementation.

314 1.6. What the JRA Is Not

315 The JRA is a reference architecture for information sharing and, as such, does not
316 address the following:

- 317 • Detailed specifications for justice agencies' operational systems
318 (e.g., police records management systems, court case management
319 systems)
- 320 • Detailed specifications of information exchanges or services
- 321 • Recommendations or standards for integration infrastructure
322 products

323

2. Architecture Requirements

This section documents the business requirements to be addressed and satisfied by the JRA. These requirements are stated in the form of principles, the intent of which is to guide and constrain the choices made in developing the architecture.

Principle: Independence of Information Sharing Partners

A reference architecture for justice information sharing should accommodate a large number of independent information sharing partners at the federal, state, local, and tribal levels of government.

Rationale

It is a plain fact that organizations responsible for functions in the criminal justice process are independent and autonomous from other organizations playing roles in that process. In general, it is not possible for one partner or set of partners to dictate to others how they conduct their business, what information systems they use, how they store information, and so on.

It is also true—especially at the state, regional, and national levels—that the number of partners that need to share information is large and growing. To make agreement on information sharing possible, it is necessary to reduce or eliminate the need to agree on how partners' systems and business processes function and to move towards open industry standards instead of proprietary approaches.

While partners may readily agree on the need to share information, their individual objectives and incentives for doing so may differ.

Any information sharing architecture that does not accommodate these facts will face difficulty in its adoption and implementation by the community. Where adopted and implemented, an architecture that does not accommodate these facts will likely fail to scale to include the large number of involved partners.

Note: This principle also summarizes the first two requirements for SOA established by the Global Infrastructure/Standards Working Group in its 2004 paper, *A Framework for Justice Information Sharing: Service-Oriented Architecture* [SOA-REC, pages 2–5].

Implications

This principle implies the following about the JRA:

- The JRA should encourage the definition of system interfaces that focus only on what system functionality or information is to be

357 shared, not on how organizations design, deploy, or operate their
358 systems

- 359 • The JRA should encourage information sharing mechanisms and
360 approaches based on open industry standards rather than on
361 approaches proprietary to one vendor, one domain, one level of
362 government, or one specific partner
- 363 • The JRA should identify issues on which justice information
364 sharing partners will typically need to reach and enforce
365 agreement, which conversely will identify issues on which they can
366 continue to take independent approaches

367 **Principle: Scalability**

368 A reference architecture for justice information sharing should provide useful
369 guidance to integrated justice enterprises of all sizes, from small operations with a few
370 participants, to national processes that reach across local, state, tribal, federal, and
371 even international boundaries.

372 Rationale

373 The national justice community consists of enterprises large and small, from the
374 smallest rural county to the largest metropolitan areas and most populous states. To
375 enable sharing of justice information within and among these jurisdictions, a
376 consistent set of technical standards, guidelines, and infrastructure requirements is
377 necessary. An information sharing architecture that addresses only one size of
378 jurisdiction will fall short of the goal of fulfilling a truly national scope.

379 In addition, experience and practical considerations indicate that information sharing
380 architecture is most often implemented in an incremental fashion. Jurisdictions
381 should be able to implement modest standards and infrastructure at first, with
382 confidence that as their scope and capabilities grow, there will be minimal rework
383 and reinvestment. This principle promotes an architecture that will satisfy the needs
384 of an initial implementation and that will retain its relevance as the implementation
385 expands.

386 Note: This principle also summarizes the third requirement for SOA established by
387 the Global Infrastructure/Standards Working Group [[SOA-REC](#), pages 5–6].

388 Implications

389 This principle implies the following about the JRA:

- 390 • The JRA should adopt a modular approach that allows jurisdictions
391 to implement a subset of the full architecture, achieving some initial

392 benefit while retaining the option of adopting more of the
393 architecture later

- 394 • The JRA should encourage the adoption of industry standards with
395 a broad range of implementations available in the marketplace,
396 from less expensive implementations with modest capabilities, to
397 larger investments that support an increased volume of information
398 sharing
- 399 • The JRA should encourage the clear description, the
400 straightforward discovery, and ultimately the reuse of services
401 across jurisdictions to provide information more economically
402 (particularly to smaller jurisdictions)

403 **Principle: Diversity of Data Source Architectures**

404 A reference architecture for justice information sharing should accommodate data
405 sources and partner systems that differ widely in software, hardware, structure, and
406 design.

407 Rationale

408 There is not now—nor will there be in the foreseeable future—a single solution or
409 system for any particular domain within criminal justice. Because of the
410 independence and autonomy of jurisdictions (and organizations within jurisdictions),
411 it will in general be impossible for the sharing of justice information to rely on a
412 single vendor system, application platform, or database. Even if it were possible to
413 achieve, implementing a single vendor’s solution across all the partners within a
414 jurisdiction introduces interdependencies that reduce agility and impede technical
415 and policy innovation.

416 In addition, today’s optimal choice of systems and platforms will likely be different
417 than tomorrow’s. When a partner wishes to swap out old software or hardware for a
418 new solution, that ought not to cause chaos for its information sharing partners.

419 Note: This principle also summarizes the fourth requirement for SOA established by
420 the Global Infrastructure/Standards Working Group [[SOA-REC, page 6](#)].

421 Implications

422 This principle implies the following about the JRA:

- 423 • The JRA should encourage the sharing of information and
424 functionality between systems in a way that minimizes the
425 implementation dependencies between them

- 426 • The JRA should encourage communication between systems using
427 open industry standards rather than proprietary approaches
- 428 • The JRA should use vendor-neutral terminology and concepts in
429 defining the architecture
- 430 • The JRA should adopt a modular approach to intersystem
431 communication mechanisms and protocols so that the entire
432 architecture need not change when improved protocols are
433 developed in the future

434 **Principle: Agility**

435 A reference architecture for justice information sharing should accommodate
436 changes in policy, information flow, and partner system implementation without
437 forcing investments or changes in unrelated systems or exchanges.

438 Rationale

439 While the events in the justice community that trigger information exchange remain
440 fairly constant (arrests, bookings, charging decisions, case filing, disposition,
441 supervision, etc.), the policy responses and the flow of information following these
442 events are in constant change. This principle promotes an architecture that allows
443 information sharing practitioners to respond to—and even to thrive in—this dynamic
444 environment.

445 Technologies within partner organizations change frequently as well. The days of
446 purchasing a line of business system, such as a records system or a case
447 management system, and leaving it untouched for years at a time are long past.
448 New capabilities available from vendors and improvements in internal operations
449 both compel a more rapid rate of change. This principle promotes an architecture
450 that separates partners' system implementations from one another, reducing the
451 impact of change to one on the others.

452 Note: This principle also reflects the sixth requirement for SOA established by the
453 Global Infrastructure/Standards Working Group [[SOA-REC](#), pages 7–8].

454 Implications

455 This principle implies the following about the JRA:

- 456 • The JRA should encourage the sharing of information and
457 functionality between systems in a way that minimizes the
458 implementation dependencies between them
- 459 • The JRA should encourage the definition of system interfaces that
460 reflect what the interfaces do, as opposed to how they work

- 461 • The JRA should provide mechanisms to separate the logic of
462 information exchange (e.g., the routing and transforming of
463 messages that flow between partners) from the logic of line-of-
464 business systems

465 **Principle: Reuse and Sharing of Assets**

466 A reference architecture for justice information sharing should promote the use of
467 existing system interfaces, information exchanges, and infrastructure to support new
468 business requirements.

469 Rationale

470 Organizations responsible for criminal justice are, like many public sector
471 organizations, being asked by citizens to do more with less. In addition, reusing
472 system interfaces and information exchange implementations can improve
473 consistency and reliability of information by having all information consumers draw
474 from the same source. This principle reflects these factors by encouraging an
475 architecture that supports reuse of interfaces and infrastructure.

476 Implications

477 This principle implies the following about the JRA:

- 478 • The JRA should encourage the definition of system interfaces that
479 do not require usage in particular contexts
- 480 • The JRA should provide mechanisms to separate the logic of
481 information exchange (e.g., the routing and transforming of
482 messages that flow between partners) from the logic of line-of-
483 business systems

484 **Principle: Alignment With Best Practices and Experience**

485 A reference architecture for justice information sharing should reflect concepts and
486 mechanisms that have proven viable in actual, real-world information exchange
487 scenarios; the architecture should reflect the experiences of both public- and private-
488 sector information exchange implementation projects.

489 Rationale

490 There is considerable experience, both in the private and public sectors, with
491 implementing information sharing architecture. This principle encourages the JRA to
492 help future implementers avoid the pitfalls of the past, while adopting those practices
493 that have proven effective.

494 Note: This principle also reflects the fifth requirement for SOA established by the
495 Global Infrastructure/Standards Working Group [[SOA-REC](#), pages 6–7].

496 Implications

497 This principle implies the following about the JRA:

- 498 • The JRA should base proposed standards and infrastructure
499 requirements on practices that have proven effective

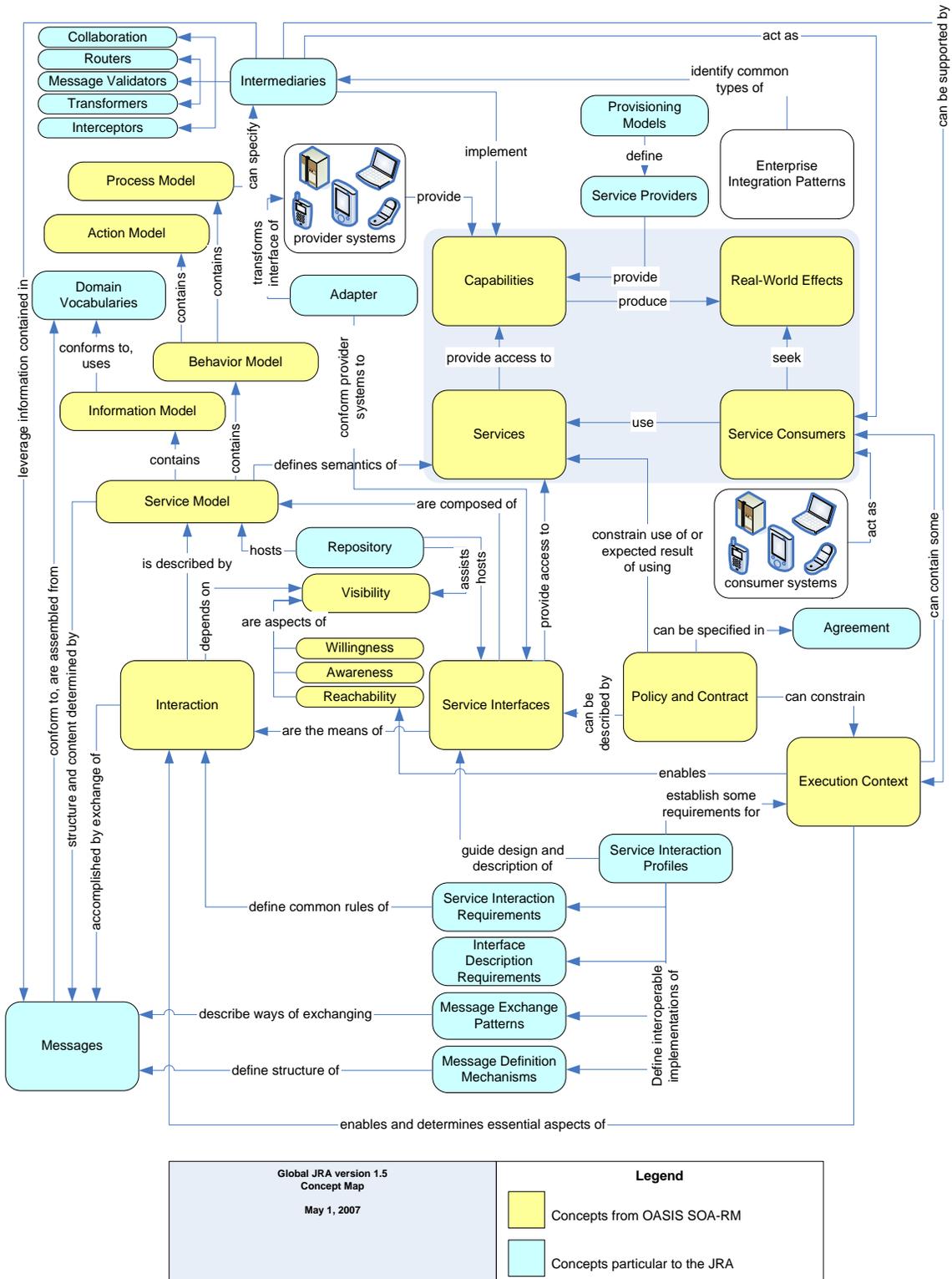
500

501

502 **3. The JRA**

503 **3.1. Graphical Overview**

504 The following diagram depicts the concepts, high-level components, and
505 relationships in the JRA specification Version 1.7. These elements are described in
506 detail in the following sections.



507

508

4. Concepts and Relationships

The following sections describe the concepts, components, and relationships depicted in the diagram on the previous page.

OASIS Reference Model for Service-Oriented Architecture

The JRA depicted in the diagram above (and defined in this document) adopts and builds on the OASIS SOA-RM.

The SOA-RM defines its purpose as follows:

“A **REFERENCE MODEL** is an abstract framework for understanding significant relationships among the entities of some environment. It enables the development of specific reference or concrete architectures using consistent standards or specifications supporting that environment. A reference model consists of a minimal set of unifying concepts, axioms and relationships within a particular problem domain, and is independent of specific standards, technologies, implementations, or other concrete details.” [SOA-RM, p. 4]

“The goal of this reference model is to define the essence of service-oriented architecture, and emerge with a vocabulary and a common understanding of SOA. It provides a normative reference that remains relevant for SOA as an abstract and powerful model, irrespective of the various and inevitable technology evolutions that will influence SOA deployment.” [SOA-RM, p. 4]

While the SOA-RM is a powerful model that provides a vendor-neutral, open-standard definition of service-oriented architecture, its abstract nature means that further work must be done to create a reference architecture. This work should include the definition of specific standards and guidelines for information sharing and should define minimum requirements for infrastructure necessary to enable information sharing while supporting those standards and guidelines. It should do this in a way that satisfies the goals and requirements of the enterprise creating the reference architecture.

The JRA is just such a reference architecture, intended to satisfy the goals and requirements of justice information sharing by identifying specific standards, guidelines, and infrastructure requirements for any group of justice partners interested in sharing information among themselves.

In the JRA diagram, OASIS SOA-RM concepts are shaded yellow. Concepts and components particular to the conceptual architecture defined by this document are shaded cyan. Relationships between concepts (indicated by arrows) are defined in

545 the SOA-RM if the arrows connect concepts shaded yellow. Relationships between
546 cyan-shaded concepts or between cyan-shaded and yellow-shaded concepts are
547 particular to the JRA.

548 The descriptions of SOA-RM concepts provided in the following sections are
549 intended to be brief summaries; consequently, they omit certain details that appear
550 in the SOA-RM. The SOA-RM itself is the primary source for full exposition of
551 SOA-RM concepts and the relationships between them.

552 **Core Concepts—Services, Service Consumers, Capabilities, and Real-** 553 **World Effects**

554 *These four concepts make up the core of the JRA. All other concepts support these*
555 *concepts.*

556 The JRA begins from the premise that a group of justice partners have **CAPABILITIES**
557 that they provide to one another. These capabilities “solve or support a solution for
558 the problems [businesses] face in the course of their business.” [SOA-RM, p. 8] That
559 is, capabilities are the things organizations have to solve problems and therefore add
560 value, directly or indirectly, to their stakeholders.

561 Note that the JRA is generic enough to support virtually any kind of capability.
562 However, the purpose of the JRA is to describe an approach to achieving
563 interoperability among automated, computer software-based information systems.
564 Therefore, the JRA considers only those business capabilities that are provided by
565 information systems. The JRA calls these systems **PROVIDER SYSTEMS**.

566 Each capability produces one or more **REAL-WORLD EFFECTS**, each of which is an
567 outcome of the business value sought by one of the partners. A real-world effect can
568 be either the obtaining of information, the changing of something of business
569 relevance to the participating partners, or both. Because the JRA establishes that
570 capabilities are implemented by provider systems, real-world effects consist of the
571 functional business requirements of provider systems. That is, real-world effects in
572 the JRA are essentially the information made available by provider systems or the
573 outcomes resulting from business processes and workflows automated by provider
574 systems, or both.

575 In a service-oriented architecture, a **SERVICE** is the way in which one partner gains
576 access to a capability offered by another partner. A partner that uses a service to
577 gain access to another partner’s capability is called a **SERVICE CONSUMER**. As with
578 capabilities, the architecture is generic enough to support virtually any kind of service
579 consumer. However, since the purpose of the JRA is to describe an approach to
580 information systems interoperability, the JRA narrows the SOA-RM definition of
581 service consumer to information systems that interact with services directly through

582 an interface that conforms to a service interaction profile (as defined below). The
583 JRA calls such systems **CONSUMER SYSTEMS**.

584 One of the most important features of the JRA is the separation of consumer systems
585 from provider systems by services in the middle. This is the defining characteristic of
586 a service-oriented architecture and is the key to minimizing the implementation
587 dependencies between systems, which is identified as part of the rationale of several
588 of the JRA principles listed above.

589 The fact that information sharing is one kind of real-world effect allows the
590 architecture to support the traditional view of system integration as “data exchange”
591 or “information sharing.” The JRA improves this view by encouraging systems to
592 share information in a way that minimizes the dependencies of each system on the
593 implementation of other systems.

594 ***Supporting Concepts***

595 Beyond the four core concepts of real-world effects, capabilities, services, and service
596 consumers, the remainder of the concepts in the JRA deal with the following three
597 important concerns:

- 598 • How consumers may find out that a service exists
- 599 • Once they find the service, how consumers may understand what
600 the service does and what information flows in and out of it
- 601 • How a consumer may reach and interact or communicate with the
602 service

603 The remaining concepts that address these concerns are called “supporting
604 concepts” and are defined in this section.

605 ***Interaction, Visibility, Service Models, and Service Interfaces***

606 Services define what features of a provider system the system owner makes
607 accessible to business partners. Services also provide a logical description of the
608 information exchanged between consumer and provider systems as the consumer
609 accesses the capability.

610 ***Interaction***

611 The JRA refers to a consumer’s accessing the features of a capability through a
612 service as **INTERACTION**, defined as “the performing [of] actions against a service.”
613 [SOA-RM, p. 15] Service interaction generally involves the exchange of information
614 between the consumer and the service.

615 Interaction depends on two things. First, the designers of potential consumers need
616 to be able to find services and, once found, establish a physical interaction
617 mechanism with them. These needs are addressed by the concept of **VISIBILITY**.
618 Second, the designers of potential consumers need a description of the actions that
619 can be performed on a service, as well as the structure and meaning of information
620 exchanged during the interaction. These needs are addressed by the concept of a
621 service's **INFORMATION MODEL** and **BEHAVIOR MODEL**, collectively called **SERVICE**
622 **MODELS** in the JRA.

623 **Visibility**

624 Visibility, as the name implies, defines how service consumers and the providers of
625 capabilities “see” each other in a way that enables interaction between them. The
626 JRA identifies three aspects of visibility.

- 627 • A service consumer must have information that makes it aware of
628 the existence of a service; the possession of this information is
629 called **AWARENESS**.
- 630 • The service (or capability accessed through the service) must be
631 willing to interact with the consumer; this is called **WILLINGNESS**.
- 632 • The consumer and service must be able to communicate with one
633 another through some kind of communication path or channel; the
634 existence of such a communication path is called **REACHABILITY**.

635 In the JRA, a **REPOSITORY** will support awareness by hosting service models and
636 service interfaces. “Hosting” in this context means storing models and interface
637 descriptions in a central location that is accessible to appropriate stakeholders. A
638 repository will permit searching for models and interface descriptions based on a
639 range of identifying criteria. A repository will also map logical service identifiers with
640 physical addresses. When a consumer wishes to communicate with a service
641 (identified by a logical identifier), the consumer queries the repository for the
642 physical address associated with the service’s logical identifier. This decouples the
643 consumer from the physical location of a service at any point in time, thereby
644 permitting the physical relocation of the service without affecting the implementation
645 of the consumer.

646 The concept of willingness is related to authorization and access control policies, in
647 that a common reason for lack of willingness to interact is that the consumer is not
648 authorized to conduct the requested interaction. Willingness often manifests in
649 service descriptions, as well as policies, contracts, and agreements (discussed on
650 page 24). A **SERVICE MODEL** is defined as the information needed in order to use,
651 or consider using, a service.

652 The concept of reachability is closely related to the concept of execution context
653 (discussed on page 25).

654 Service Models

655 Service models, consisting of a service’s behavior and information models, define the
656 semantics of interaction with the service.

657 The behavior model of a service consists of two parts—the action model, which
658 defines the operations available to consumers (in effect, what the service does) and
659 the process model, which defines how consumers may invoke the service’s actions
660 together or in sequence to accomplish some larger business process.¹

661 The information model of a service describes the structure and meaning of data that
662 consumers send to and receive from the service in the course of interaction.

663 In general, service models will be described at conceptual and logical levels of detail.
664 (Service models have a physical manifestation as well, in the form of the service
665 interface discussed in the next section.) A conceptual description of a service model
666 will typically describe, in prose text form, the capability to which the service provides
667 access, a listing and brief textual description of each action, and a brief textual
668 description of the information model (e.g., key information entities, key properties on
669 those entities, and brief definitions). A logical description of a service model will
670 describe the actions and information structures in detail but independent of any
671 physical implementation mechanism. Often this description will be graphical and
672 follow a standard diagramming or modeling technique, such as Uniform Modeling
673 Language (UML).

674 A **MESSAGE** is defined as the entire “package” of information sent between service
675 consumer and service (or vice versa), even if there is a logical partitioning of the
676 message into segments or sections. For instance, if an interface expresses actions as
677 operations or functions that take arguments, and a particular operation has two
678 arguments, both arguments would be considered part of the same message, even
679 though they may be logically separated within the message structure. A message
680 also includes the concept of an “attachment,” in which there are several additional
681 sections (attachments) that relate to a distinct, “primary” section.

682 In the JRA, the exchange of messages is the only way in which consumers and
683 services can communicate. This establishes a linkage between the Federal Enterprise
684 Architecture Data Reference Model (FEA DRM) and the JRA—a message in the JRA
685 equates to an Information Exchange Package (IEP) in the FEA DRM. In the JRA, all
686 service interaction is accomplished via message (information) exchange, and each
687 message triggers the invocation of an action in the service’s action model.

¹The OASIS SOA-RM term “process model” is consistent with the JRA definition given here; however, it is somewhat at odds with the popular notion of “Business Process Modeling,” which generally refers to documenting/modeling the interactions between many services or capabilities. The JRA remains consistent with the OASIS SOA-RM, but readers are cautioned not to confuse the two definitions of this term.

688 The concept of **DOMAIN VOCABULARIES** in the JRA includes canonical data models,
689 data dictionaries, and markup languages that standardize the meaning and structure
690 of information for a topical or business domain. Domain vocabularies can improve
691 the interoperability between consumer and provider systems by providing a neutral,
692 common basis for structuring and assigning semantic meaning to information
693 exchanged as part of service interaction. Domain vocabularies can usually be
694 extended to address information needs specific to the service interaction or to the
695 business partners integrating their systems.

696 The information model for a service generally should be built from components in
697 one or more domain vocabularies to promote semantic interoperability. In the
698 justice domain, the information model for services should be built from components
699 in the National Information Exchange Model (NIEM) when NIEM components exist
700 that satisfy the semantic requirements of the model.

701 **SERVICE DESIGN PRINCIPLES**² provide consistent guidance regarding the overall
702 partitioning of capabilities into services and the relationships between services. For
703 instance, service design principles may call for services to represent one concise, self-
704 contained function and may also suggest that services should completely hide the
705 implementation details of the capabilities to which they provide access.

706 There is a wide variety of ways in which a service can provide access to a capability.
707 In some cases, the provider system that implements the capability may already
708 expose all or some of its functionality as services (through one or more service
709 interfaces, described on page 20). In other cases, the business partner that
710 provisions the capability can purchase an off-the-shelf adapter from the provider
711 system vendor (or a third party) that exposes the system's functionality as a set of
712 services. Finally, the provider system may require reimplementing or custom
713 adaptation to expose functionality as services. This is often expensive and risky, and
714 the desire to avoid this situation should be addressed in the service design guidelines.

715 In general, a given information system can be both a provider system and a
716 consumer system. Similarly, a particular business organization may offer capabilities
717 to its partners and, at the same time, be a consumer of the capabilities offered by
718 others. This has important implications for how the organization should conceive
719 and describe its information systems assets and how it assigns responsibilities for the
720 maintenance and support of those assets. For example, in the past, it was common
721 to think of systems as having "client" and "server" components (or "browser" and
722 "server" components), which in turn influenced thinking about systems deployment,
723 networking, security, support, and a range of other issues. These issues deserve
724 reconsideration in an architecture in which a system or system component can be
725 both a "client" (consumer of services) and a "server" (provider of services) at the

²Principles and guidelines are important components of the conceptual JRA; however, these principles and guidelines are not illustrated on the diagram because they will exist for most of the components.

726 same time. The discussion of service interaction on page 16, and the subsequent
727 elaboration of interaction mechanisms in future iterations of the JRA, will reflect the
728 impact of these issues.

729 Note that the concept of a service in the JRA does not equate to a Web service. The
730 term “Web services” is a label for a family of standards and an associated technical
731 approach to communicating between service consumers and services. The
732 architecture supports flexibility in how this communication happens through the
733 notion of service interaction profiles (discussed on page 22). A Web service profile
734 has been developed for the Web services family of standards; however, the JRA will
735 include additional profiles that adopt other communication mechanisms, such as
736 MQ, JMS, and ebXML. [\[WSSIP AND ebXMLSIP\]](#)

737 As previously stated, a repository should contain service model description artifacts
738 for each level of detail. The availability of service model descriptions to consumer
739 system designers, implementers, and purchasers is a key factor in establishing
740 visibility and the reuse of services.

741 **Service Interface**

742 Service models describe the actions available from a service and the information
743 exchanged between a consumer and the service during the performance of those
744 actions. In this way, the service models describe the “what” of interaction.

745 A **SERVICE INTERFACE** “is the means for interacting with a service. It includes the
746 specific protocols, commands, and information exchange by which actions are
747 initiated [on the service].” [\[SOA-RM, p. 22\]](#) A service interface is what a system
748 designer or implementer (programmer) uses to design or build executable software
749 that interacts with the service. That is, the service interface represents the “how” of
750 interaction.

751 In many cases, the capability to which a service provides access is some kind of
752 information system. The JRA calls such a system a provider system, as discussed
753 above on page 15. However, in general, a provider system will not conform to or
754 satisfy the constraints imposed by the service interface through which consumers
755 access the capability. A software component called an **ADAPTER** is required to
756 transform interactions with the provider system into interactions that conform to the
757 service interface. Depending on the type of provider system, adapters may be
758 available from the system vendor or a different vendor; in other cases, the service
759 provider may need to build a custom adapter.

760 The JRA considers the service interface to be the physical manifestation of the
761 service models. Best practices call for a service interface to be described in an open-
762 standard, referenceable format (that is, a format whose contents are capable of
763 automated processing by a computer).

764 Many service interaction profiles use the term “endpoint” or “endpoint interface” to
765 refer to the physical point that receives a message sent by a consumer. There is a
766 one-to-one correspondence between a service interface and an endpoint interface,
767 and the JRA considers the two terms to be synonymous.

768 A given service may have multiple interfaces that conform to the same service
769 interaction profile, where the multiple interfaces expose different sets of the service’s
770 actions. For instance, a service may have one “query” action and three “update”
771 actions; the query action may be exposed by one Web services interface, while the
772 three update actions may be exposed by a separate Web services interface. These
773 interfaces would reside at separate endpoints.

774 Note that at least some policies and contracts can be described in a service’s
775 interface.

776 The format, structure, and allowable contents of a service interface are established by
777 **INTERFACE DESCRIPTION REQUIREMENTS**, described in the following section.

778 ***Design and Description of Service Interfaces***

779 The JRA identifies four architectural elements that guide the design and description
780 of service interfaces.

781 **SERVICE INTERACTION REQUIREMENTS** define common rules of service interaction.
782 Typically, these requirements are not directly related to the capability used by the
783 service consumer, nor are they related to the real-world effect resulting from use of
784 that capability. Rather, the requirements enforce (or support the enforcement of)
785 policies or contracts or otherwise protect the interests of particular business partners
786 or the business organization overall.

787 Common service interaction requirements address areas such as security, reliability,
788 and availability. An initial elaboration of service interaction requirements appears on
789 page 29.

790 **INTERFACE DESCRIPTION REQUIREMENTS** establish common characteristics of
791 service interface descriptions. These requirements address areas such as required
792 interface contents, naming rules, documentation rules, and specification of a
793 standard structure and format for descriptions.

794 **MESSAGE EXCHANGE PATTERNS** identify common sequences of message
795 transmission between service consumers and services. They provide a label to a
796 series of message transmissions that have some logical interrelationship.

797 **MESSAGE DEFINITION MECHANISMS** are closely related to interface description
798 requirements, described above. Unlike interface description requirements, message
799 definition mechanisms establish a standard way of defining the structure and

800 contents of a message. Note that since a message includes the concept of an
801 “attachment,” the message definition mechanism must identify how different sections
802 of a message (for example, the main section and any attachment sections) are
803 separated and identified and how attachment sections are structured and formatted.

804 **Service Interaction Profiles**

805 A **SERVICE INTERACTION PROFILE** defines a family of industry standards or other
806 technologies or techniques that together demonstrate implementation or satisfaction
807 of:

- 808 • Service interaction requirements
- 809 • Interface description requirements
- 810 • Message exchange patterns
- 811 • Message definition mechanisms

812 Service interaction profiles are included in the JRA to promote interoperability
813 without forcing the organization to agree on a single way of enabling service
814 interaction. Each service interface will support a single profile; a service will have
815 multiple interfaces if it supports multiple profiles. By supporting a profile, an
816 interface establishes the mode of interoperation it allows from service consumers;
817 any consumer that also supports that profile can “reach” the service.

818 The JRA explicitly recognizes that a service interaction profile may be further
819 constrained by an implementer to require specific techniques, technologies, or
820 mechanisms, as long as the additional constraints remain consistent with the original
821 profile.

822 **Capabilities in Detail**

823 The JRA identifies several types of capabilities to assist decision makers in
824 understanding where certain capabilities should be deployed in the organization and
825 what relationships they may have to other capabilities and services.

826 **Intermediaries**

827 An **INTERMEDIARY** is any capability that receives messages from a consumer and
828 subsequently, as a service consumer itself, interacts with another service. The term
829 “intermediary” indicates that these capabilities sit between other services and
830 “mediate” the interaction by managing, controlling, brokering, or facilitating the
831 transmission of messages between them. An intermediary is the mechanism by
832 which the JRA separates the logic of integration from the logic of line-of-business
833 systems, which is a key feature of SOA.

834 The JRA identifies five types of intermediary but recognizes that other types are
835 possible. The five identified types are orchestrations, routers, message validators,
836 transformers, and interceptors.

837 An **ORCHESTRATION** is a capability that coordinates interaction with multiple
838 services. It is a declarative technique used to compose hierarchical and self-
839 contained service-oriented business processes that are executed and coordinated by
840 a single conductor [SOA-RA, p. 69]. An orchestration is often implemented using an
841 open industry standard implementation mechanism such as Business Process
842 Execution Language (BPEL) that allows the implementation to be shared across
843 tools and platforms.

844 It is often possible to design and model orchestrations using a graphical approach, in
845 which the implementer diagrams business processes and work flows, the steps of
846 which are services that already exist. After the diagram is complete, the implementer
847 generates a standards-based artifact that is deployed into a software component that
848 exposes the work flow as a service through a service interface. The promise of this
849 approach is that less technical implementers with greater business expertise can be
850 responsible for the implementation of orchestrated capabilities.

851 Note that the execution of the steps described in a business process model can be
852 considered a capability in and of itself. In addition, each of the steps in a business
853 process model can unfold into yet another business process model at a more focused
854 level of detail. In this way, each step in a series of service interactions can itself be a
855 series of service interactions. And, in theory, this recursion of models can go on
856 forever, though in practice it rarely exceeds three or four levels of containment. So,
857 services and capabilities form a hierarchy, where a service provides access to a
858 capability whose real-world effect is to accomplish the coordination of multiple
859 services at a lower level of detail.

860 As a side effect, each of the steps in a business process model provides a contextual
861 justification for service interaction between a particular consumer and a particular
862 provider. It is often useful to capture this information in a taxonomy to promote a
863 better understanding of where services are being used and to add value.

864 Note that an orchestration is different from a choreography, in that a choreography
865 is a description of how a group of business peers coordinate a service-oriented
866 business process without the direction of a controller.

867 **ROUTERS** are capabilities that receive a message, examine it, and transmit it to one
868 or more destinations based on the contents. In general, routers can be designed to
869 operate on any of the information contained within the message; they may use
870 information about the origin of the message, routing directive information contained
871 within the message or the main content of the message itself.

872 **TRANSFORMERS** are capabilities that receive a message and transform it into another
873 format before transmitting it to another destination.

874 **MESSAGE VALIDATORS** are capabilities that examine a message to ensure that the
875 contents adhere to established business rules.

876 **INTERCEPTORS** are capabilities that receive a message and use the message content
877 to trigger a secondary action; generally, the interceptors pass the message unaltered
878 to the next step in a process. Most interceptors capture information from the
879 message for reporting or analytical purposes.³

880 Routers and transformers are useful mechanisms for decoupling the senders and
881 recipients of messages. They tend to centralize and share certain kinds of logic so
882 that the logic can be maintained independently of the provider and consumer
883 capabilities at the edges; sharing also improves the likelihood of reuse, since it is
884 easier to reuse functionality if it encapsulates a single task.

885 Support for router, transformer, and collaboration capabilities is a common feature
886 in many integration platforms; therefore, support for these capabilities is a
887 consideration in choice of execution context (discussed on page 25).

888 Routing, transformation, and collaboration capabilities are well understood and well
889 documented in the integration architecture literature. The most common flavors of
890 these capabilities have been collected into pattern form as **ENTERPRISE**
891 **INTEGRATION PATTERNS**. [PATTERNS] The JRA incorporates these patterns by
892 reference.

893 Intermediaries are a key component in implementing business process models and
894 also lead to the formation of service/capability hierarchies.

895 ***Service Policy, Service Contract, and Service Agreement***

896 **SERVICE POLICIES** and **SERVICE CONTRACTS** express rules that govern the
897 interaction between a service consumer and a service. A policy is an assertion by
898 either a consumer or a service provider of that participant's requirements for
899 willingness to interact. A policy also has an enforcement aspect and must be stated
900 in such a way as to permit enforcement. A **SERVICE CONTRACT** is an agreement by
901 the parties involved, and there is a process associated with the agreement action.
902 Whereas a policy is an assertion by one participant in the interaction, a contract is an
903 agreement between the participants that expresses some expectation or requirement
904 of the interaction. And whereas policy enforcement is generally the responsibility of

³The concept of interceptor defined here is similar to, but separate and distinct from, the notion of an interceptor as defined in the SOAP protocol [reference needed to SOAP standard]. The definition of this concept in JRA is not intended to imply any implementation technique or technology.

905 the participant who asserts the policy, contract enforcement may involve resolution
906 of disputes that arise between the parties.

907 A **SERVICE AGREEMENT** is a document that establishes policies and contractual
908 elements for a given interaction or set of interactions (that is, for one or more
909 services).

910 **Execution Context**

911 **EXECUTION CONTEXT** is “the set of infrastructure elements, process entities, policy
912 assertions, and agreements that are identified as part of an instantiated service
913 interaction.” [SOA-RM, p. 24]

914 Execution context is the primary enabler of the reachability aspect of visibility.
915 Execution context includes the set of infrastructure elements that provide a physical
916 communication path between service consumers and services.

917 The JRA considers execution context to be primarily the supporting infrastructure
918 elements that permit service consumers and services to interact. These infrastructure
919 elements consist of:

- 920 • Data networks used by service consumers and services to exchange
921 information.
- 922 • Integration infrastructure (hardware and software) that makes
923 service interfaces available and handles higher-level message
924 routing, transformation, and collaboration.
- 925 • Infrastructure technology to support service interaction; examples
926 include access control, policy decision-making and enforcement,
927 public key infrastructure (PKI), and metering.

928 Execution context can implement (or support the implementation of) some service
929 interaction requirements, such as reliability and availability. Service interaction
930 profiles, contracts, and policies can constrain the behavior of execution context
931 elements by requiring particular technologies or techniques or establishing service
932 level policies, for example.

933 Finally, execution context can support intermediary capabilities (as defined above)
934 directly in the integration infrastructure.

935 **Provisioning Model**

936 A **PROVISIONING MODEL** determines the organizational (perhaps contractual or legal)
937 responsibility for providing a capability, via services, to achieve consumers’ desired
938 real-world effect. The entity identified in a provisioning model as responsible for
939 providing a capability is called a **SERVICE PROVIDER**.

940

5. Reconciliation of Architecture With Principles

The JRA seeks to support and encourage the set of principles identified earlier in this document.

Principle: Independence of Information Sharing Partners

Principle: Diversity of Data Source Architectures

Principle: Agility

These three principles are all interrelated. What ties them together is the notion that in the justice business domain, partners who exchange information and collaborate in business processes must remain autonomous, separately governed organizations. They must retain the ability to establish policy and practice in their own organizations, while at the same time establishing common policy and practice for the common enterprise in which they all participate. They will maintain different information systems from different vendors (in some cases, building critical systems in-house); these systems will be written in diverse programming languages and will leverage diverse database management systems and application servers. An architecture that required uniformity in these areas would be doomed to failure.

To maintain this autonomy and yet be effective, partners must adopt an architecture that gives them agility, or the ability to be responsive to changing circumstances. These circumstances could involve the factors just mentioned—changing internal policies, changing system vendors, or changing technologies. But the circumstances could originate from external forces that affect all participants equally—changes in citizen needs and expectations, changes in legislation, changing requirements for sharing information with federal partners, and so on. The architecture must support a responsive, flexible approach to information sharing between partners.

The JRA promotes business agility in those organizations that adopt it by encouraging systems, agencies, information exchanges, and business process to have minimal dependencies on one another. It accomplishes this in several ways:

- It encourages the conceptualization of information exchanges as actions on services, which introduce a layer between systems that exchange information. This allows one agency to change anything about its internal operations and system behavior without disrupting partners' businesses. This in turn increases the rate at which partners can change, which is agility.
- It introduces a service identification methodology (in a separate document) that establishes principles and techniques for service design that minimize the dependency of one service on another.

- 977 • It introduces the concept of a service interaction profile, which
978 encourages justice partners to adopt standards-based, vendor-
979 neutral approaches to the transmission and encoding of
980 information between agencies.

981 **Principle: Reuse and Sharing of Assets**

982 The JRA encourages the reuse and sharing of assets in several ways:

- 983 • It introduces as one of its core concepts the notion of visibility for
984 services. The concept of visibility recognizes that potential
985 consumers must be aware of the existence of services and, once
986 aware of them, must have clear documentation regarding how to
987 access them.
- 988 • It includes service modeling guidelines, which establish clear,
989 consistent rules for the information contained in a service
990 description and how that information must be presented so that
991 potential consumers understand what the service does and how to
992 interact with it.
- 993 • It introduces the concept of execution context and guidelines for
994 how to share execution context infrastructure across a group of
995 partners. Thus, instead of each project or pair of partners
996 provisioning its own infrastructure, partners share common
997 infrastructure elements where it is possible to do so.
- 998 • It introduces, as part of shared execution context, registries and
999 repositories that store service descriptions and support searches
1000 that allow potential consumers to find the services they need
1001 quickly. The easier it is for consumers to find services, the more
1002 likely they are to reuse them.

1003 **Principle: Scalability**

1004 The conceptual framework, standards, and guidelines within the JRA apply to
1005 enterprises of varying sizes, from pairs of partners with a handful of exchanges to
1006 large, multiagency efforts with dozens of exchanges, to a national environment with
1007 potentially hundreds of participants and thousands of exchanges.

1008 It is possible to implement basic components of the JRA, such as the conceptual
1009 framework, service interaction profiles, service identification methodology, and
1010 service modeling guidelines, without significant investments in infrastructure
1011 (middleware, registries, etc.) Enterprises with a few services representing point-to-
1012 point information exchanges can often move forward with infrastructure already in
1013 place.

1014 At the same time, the guidelines and standards in the JRA are well-aligned with
1015 industry direction and product offerings, so larger enterprises can also leverage the
1016 same standards within the enhanced capabilities of sophisticated infrastructure.

1017 **Principle: Alignment With Best Practices and Experience**

1018 The JRA aligns with best practices and the experiences of innovative organizations in
1019 the following ways:

- 1020 • It has been developed by a group of practitioners and technologists
1021 from the public sector, national associations, and industry who
1022 have gained experience working with service-oriented architecture.
1023 It is the result of this group of experienced individuals collaborating
1024 and consolidating the lessons learned from SOA implementation
1025 projects.
- 1026 • It leverages accepted standards that have been developed by
1027 industry standards bodies, representing a diversity of technologies
1028 and vendors. The conceptual framework is based on (and
1029 conforms to) the OASIS SOA-RM. Individual JRA deliverables,
1030 such as service interaction profiles and service modeling guidelines,
1031 further leverage open industry standards such as the Web services
1032 stack and UML.
- 1033 • It builds on and provides linkages between national justice
1034 community standards such as NIEM, GFIPM, security, privacy
1035 guidelines, etc.

1036

1037

6. Elaboration of Service Interaction Requirements

1038

The following is an initial list of candidate service interaction requirements. Note that when these requirements refer to **SERVICE CONSUMER**, this is not a human being but an information system that interacts with a service. This is consistent with the JRA usage of the term, as defined on page 15.

1039

1040

1041

1042

- **Service Consumer Authentication:** Information provided with messages transmitted from service consumer to service that verifies the identity of the consumer.

1043

1044

1045

- **Service Consumer Authorization:** Information provided with messages transmitted from service consumer to service that documents the consumer's authorization to perform certain actions on and/or access certain information via the service.

1046

1047

1048

1049

- **Identity and Attribute Assertion Transmission:** Information provided with messages transmitted from service consumer to service that asserts the validity of information about a human or machine, including its identity.

1050

1051

1052

1053

- **Service Authentication:** The ability of a service to provide a consumer with information that demonstrates the service's identity to the consumer's satisfaction.

1054

1055

1056

- **Message Nonrepudiation:** Information provided in a message to allow the recipient to prove that a particular authorized sender in fact sent the message.

1057

1058

1059

- **Message Integrity:** Information provided in a message to allow the recipient to verify that the message has not changed since it left the control of the sender.

1060

1061

1062

- **Message Confidentiality:** Information provided in a message to prevent anyone except an authorized recipient from reading the message or parts of the message.

1063

1064

1065

- **Message Addressing:** Information provided in a message that indicates where a message originated, the ultimate destination of the message (beyond physical end point), a specific recipient to whom the message should be delivered (this includes sophisticated metadata designed specifically to support routing), and a specific address or entity to which reply messages (if any) should be sent.

1066

1067

1068

1069

1070

- **Reliability:** Information provided with messages to permit message senders to receive notification of the success or failure of message transmissions and to permit messages sent with specific sequence-related rules either to arrive as intended or fail as a group.

1071

1072

1073

1074

1075

1076

- **Transaction Support:** Information provided with messages to permit a sequence of messages to be treated as an atomic transaction by the recipient.

1077

1078

1079

1080

1081

1082

1083

- **Service Metadata Availability:** The ability of a service to capture and make available (via query) metadata about the service. Metadata is information that describes or categorizes the service and often assists consumers in interacting with the service in some way.

1084

1085 7. Glossary

1086 Architecture

1087 A set of artifacts (that is: principles, guidelines, policies, models, standards,
1088 and processes) and the relationships between these artifacts that guide the
1089 selection, creation, and implementation of solutions aligned with business
1090 goals.

1091 Awareness

1092 A state whereby one party has knowledge of the existence of the other party.
1093 Awareness does not imply willingness or reachability.

1094 Behavior Model

1095 The characterization of, and responses to, temporal dependencies between
1096 the actions on a service.

1097 Business Process Models

1098 A description (usually formal and often graphical) of a series of activities that
1099 culminate in the achievement of some outcome of business value. Some (but
1100 not necessarily all) of the steps in this series of activities involve producing a
1101 real-world effect provided by a capability, and some of the steps require a
1102 consumer to use a service. Each one of these steps, then, provides the
1103 contextual justification for service interaction between a particular consumer
1104 and a particular provider.

1105 Capabilities

1106 Real-world effect(s) that service provider(s) are able to provide to a service
1107 consumer.

1108 Collaboration

1109 A capability that coordinates interaction with multiple services. A
1110 collaboration is often implemented using an open industry standard
1111 implementation mechanism, which allows the implementation to be shared
1112 across tools and platforms.

1113 Consumer Systems

1114 The information system that gains access to another partner's capability
1115 offered by means of a service.

1116 Domain Vocabularies

1117 Includes canonical data models, data dictionaries, and markup languages that
1118 standardize the meaning and structure of information for a domain. Domain
1119 vocabularies can improve the interoperability between consumer and
1120 provider systems by providing a neutral, common basis for structuring and
1121 assigning semantic meaning to information exchanged as part of service

1122 interaction. Domain vocabularies can usually be extended to address
1123 information needs specific to the service interaction or to the business
1124 partners integrating their systems.

1125 **Enterprise Integration Patterns**

1126 Enterprise integration has to deal with connecting multiple applications
1127 running on multiple platforms in different locations. Enterprise integration
1128 patterns help integration architects and developers design and implement
1129 integration solutions more rapidly and reliably. Most of the patterns assume
1130 a basic familiarity with messaging architectures. However, the patterns are
1131 not tied to a specific implementation.

1132 **Execution Context**

1133 The set of technical and business elements that form a path between those
1134 with needs and those with capabilities and that permit service providers and
1135 consumers to interact.

1136 **Framework**

1137 A set of assumptions, concepts, values, and practices that constitutes a way of
1138 viewing the current environment.

1139 **Information Model**

1140 The characterization of the information that is associated with the use of a
1141 service. The scope of the information model includes the format of
1142 information that is exchanged, the structural relationships within the
1143 exchanged information, and the definition of terms used.

1144 **Interaction**

1145 The activity involved in making use of a capability offered, usually across an
1146 ownership boundary, to achieve a particular desired real-world effect.

1147 **Interface Description Requirements**

1148 Establishes common characteristics of service interface descriptions. These
1149 requirements address areas such as required interface contents, naming rules,
1150 documentation rules, and specification of a standard structure and format for
1151 descriptions.

1152 **Interceptors**

1153 Interceptors are capabilities that receive a message and use the message
1154 content to trigger a secondary action; generally, the interceptors pass the
1155 message unaltered to the next step in a process.

1156 **Intermediaries**

1157 Routers and transformers are collectively called intermediaries. This term
1158 indicates that routers and transformers generally sit between other services

1159 and “mediate” the interaction by managing the transmission of messages
1160 between them or by reformatting messages in transit.

1161 **Justice Reference Architecture**

1162 The JRA is an abstract framework for understanding significant components
1163 and relationships between them within a service-oriented environment. It
1164 lays out common concepts and definitions as the foundation for the
1165 development of consistent service-oriented architecture (SOA)
1166 implementations within the justice and public safety communities. The term
1167 refers to the modular architecture that clearly and appropriately identifies and
1168 separates technical and governance layers so that standards can be
1169 developed to improve interoperability. The JRA is being developed by
1170 Global; it leverages the work of others, such as the state of Washington, and
1171 builds upon the work of OASIS.

1172 **Messages**

1173 The entire “package” of information sent between service consumer and
1174 service (or vice versa), even if there is a logical partitioning of the message
1175 into segments or sections.

1176 **Message Definition Mechanisms**

1177 Establishes a standard way of defining the structure and contents of a
1178 message; for example, Global JXDM- or NIEM-conformant schema sets.
1179 Note that since a message includes the concept of an “attachment,” the
1180 message definition mechanism must identify how different sections of a
1181 message (for example, the main section and any attachment sections) are
1182 separated and identified and how attachment sections are structured and
1183 formatted.

1184 **Message Exchange Patterns**

1185 Identifies common sequences of message transmission between service
1186 consumers and services. They provide a label to a series of message
1187 transmissions that have some logical interrelationship.

1188 **Message Validators**

1189 An intermediary that examines a message to ensure that the contents adhere
1190 to established business rules.

1191 **Orchestration**

1192 A capability that coordinates interaction with multiple services. It is a
1193 declarative technique used to compose hierarchical and self-contained
1194 service-oriented business processes that are executed and coordinated by a
1195 single conductor.

1196

1197 Process Model

1198 The characterization of the temporal relationships between and temporal
1199 properties of actions and events associated with interacting with the service.

1200 Provider Systems

1201 The information system that offers the use of capabilities by means of a
1202 service.

1203 Provisioning Models

1204 The responsibility/models for making a service available to customers in a
1205 manner consistent with formal (or occasionally informal) customer
1206 expectations.

1207 Reachability

1208 The ability of a service consumer and a service provider to interact.
1209 Reachability is an aspect of visibility.

1210 Real-World Effects

1211 The actual result(s) of using a service, rather than merely the capability
1212 offered by a service provider.

1213 Reference Architecture

1214 A reference architecture is an architectural design pattern that indicates how
1215 an abstract set of mechanisms and relationships realizes a predetermined set
1216 of requirements.

1217 Reference Model

1218 A reference model is an abstract framework for understanding significant
1219 relationships among the entities of some environment that enables the
1220 development of specific reference or concrete architectures using consistent
1221 standards or specifications supporting that environment.

1222 A reference model consists of a minimal set of unifying concepts, axioms, and
1223 relationships within a particular problem domain and is independent of
1224 specific standards, technologies, implementations, or other concrete details.

1225 Repository

1226 Stores models and interface descriptions in a central location that is accessible
1227 to appropriate stakeholders. A repository will permit searching for models
1228 and interface descriptions based on a range of identifying criteria. A
1229 repository will also map logical service identifiers with physical addresses.

1230 Routers

1231 A capability that receives a message, examines it, and transmits it to one or
1232 more destinations based on the contents. In general, routers can be designed
1233 to operate on any of the information contained within the message; they may

1234 use information about the origin of the message, routing directive information
1235 contained within the message or the main content of the message itself.

1236 **Services**

1237 The means by which the needs of a consumer are brought together with the
1238 capabilities of a provider.

1239 **Service Agreements**

1240 A document that establishes policies and contractual elements for a given
1241 interaction or set of interactions (that is, for one or more services).

1242 **Service Consumers**

1243 An entity that seeks to satisfy a particular need through the use of capabilities
1244 offered by means of a service.

1245 **Service Contracts**

1246 An agreement by two or more parties regarding the conditions of use of a
1247 service.

1248 **Service Design Principles**

1249 The documentation to provide consistent guidance regarding the overall
1250 partitioning of capabilities into services and the relationships between
1251 services.

1252 **Service Interaction Profiles**

1253 Defines a family of industry standards or other technologies or techniques that
1254 together demonstrate implementation or satisfaction of:

- 1255 ○ Service interaction requirements
- 1256 ○ Interface description requirements
- 1257 ○ Message exchange patterns
- 1258 ○ Message definition mechanisms

1259 Service interaction profiles are included in the JRA to promote interoperability
1260 without forcing the organization to agree on a single way of enabling service
1261 interaction. Each service interface will support a single profile; a service will
1262 have multiple interfaces if it supports multiple profiles.

1263 **Service Interaction Requirements**

1264 Define common rules of service interaction. Typically, these requirements are
1265 nonfunctional in nature in that they are neither directly related to the
1266 capability used by the service consumer nor related to the real-world effect
1267 resulting from use of that capability. Rather, the requirements enforce (or
1268 support the enforcement of) policies or contracts or otherwise protect the
1269 interests of particular business partners or the business organization overall.

1270 **Service Interfaces**

1271 The means by which the underlying capabilities of a service are accessed.

1272 **Service Model**

1273 Interaction depends on two things. First, the designers of potential consumers
1274 need to be able to find services and, once found, establish a physical
1275 interaction mechanism with them. Second, the designers of potential
1276 consumers need a description of the actions that can be performed on a
1277 service, as well as the structure and meaning of information exchanged during
1278 the interaction. These needs are addressed by the concept of a service's
1279 information model and behavioral model, collectively called service models in
1280 the JRA.

1281 **Service-Oriented Architecture (SOA)**

1282 Service-Oriented Architecture is a paradigm for organizing and utilizing
1283 distributed capabilities that may be under the control of different ownership
1284 domains. It provides a uniform means to offer, discover, interact with, and
1285 use capabilities to produce desired effects consistent with measurable
1286 preconditions and expectations.

1287 **Service Policies**

1288 A statement of obligations, constraints, or other conditions of use,
1289 deployment, or description of an owned entity as defined by any participant.

1290 **Service Providers**

1291 An entity (person or organization) that offers the use of capabilities by means
1292 of a service.

1293 **Transformer**

1294 A capability that receives a message and transforms it into another format
1295 before transmitting it on to another destination.

1296 **Visibility**

1297 The capacity for those with needs and those with capabilities to be able to
1298 interact with each other.

1299 **Willingness**

1300 A predisposition of service providers and consumers to interact.

1301

1302

8. References

- 1303
- 1304 **ebXMLSIP** GISWG. The JRA ebXML Messaging Service Interaction
1305 Profile Version 1.0, October 1, 2007.
1306 http://it.ojp.gov/documents/ebXML_SIP_v01_Final_Version_10
1307 [0407.pdf](http://it.ojp.gov/documents/ebXML_SIP_v01_Final_Version_10).
- 1308 **Erl** Erl, Thomas. *Service-Oriented Architecture: Concepts,*
1309 *Technology, and Design*. Prentice-Hall, 2005.
- 1310 **GISWG** GISWG. *A Framework for Justice Information Sharing:*
1311 *Service-Oriented Architecture*. Global, December 9, 2004.
- 1312 **JRA** GISWG. <http://it.ojp.gov/globaljra>.
- 1313 **Patterns** Hohpe, Gregor, and Woolf, Bobby. *Enterprise Integration*
1314 *Patterns: Designing, Building, and Deploying Messaging*
1315 *Solutions*. Addison Wesley, 2004.
1316 <http://www.eaipatterns.com>.
- 1317 **Sholler** Sholler, Daniel. *Demystifying Service-Oriented Architecture,*
1318 *META Practice*, June 9, 2004.
- 1319 **SOA-RA** *Reference Architecture for Service-Oriented Architecture 1.0,*
1320 *Public Review Draft 1*. OASIS, April 23, 2008.
1321 <http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf>.
- 1322 **SOA-REC** GISWG. *A Framework for Justice Information Sharing:*
1323 *Service-Oriented Architecture*. Global, December 9, 2004.
1324 http://it.ojp.gov/documents/20041209_SOA_Report.pdf.
- 1325 **SOA-RM** *Reference Model for Service-Oriented Architecture 1.0, Oasis*
1326 *Standard*. OASIS, October 12, 2006.
1327 <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>.
- 1328 **WSSIP** GISWG. The Global JRA Web Services Service Interaction
1329 Profile Version 1.1, August 1, 2007.
1330 <http://it.ojp.gov/documents/WS->
1331 [SIP_Aug_31_version_1_1_FINAL\(3\).pdf](http://it.ojp.gov/documents/WS-SIP_Aug_31_version_1_1_FINAL(3).pdf).
- 1332
- 1333

1334

9. Document History

Date	Version	Editor	Change
March 25, 2006	1.0	Scott Came	Initial draft.
March 28, 2006	1.0	Tish Cunningham Kim Geer	Editorial changes and IIR Quality Control.
May 1, 2006	1.1	Monique La Bare	Integrated comments from EAC, glossary, introduction, acknowledgements; inserted scenario; edited page numbers.
June 1, 2006	1.1	Tom Clarke	Elaboration of concepts and principles.
June 28, 2006	1.1		Reordered elaboration of concepts, added warrant scenario.
November 2, 2006	1.2	Scott Came	Consistency edits. Edits resulting from October GISWG meetings. Reflected comments of Iveta Topalova and Martin Smith.
December 6, 2006	1.3	Kim Geer Dolores Parker	Formatting. Editorial changes and IIR Quality Control.
February 14, 2007	1.4	Scott Came	EAC revisions.
February 11, 2008	1.6	Scott Came	EAC revisions.
July 6, 2008	1.7 candidate	Scott Came	Added concepts of relationships between actions, messages, and the action/process models of a service.

Date	Version	Editor	Change
October 30, 2008	1.7 candidate	Monique La Bare	Verified references, added service interaction requirements; editing.
November 18, 2008	1.7	Scott Came	New service interface language; Executive Summary update

www.it.ojp.gov/globaljra