Global Justice Information Sharing Initiative

United States
Department of Justice

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

Specification

Version 1.7

March 2009

Global Infrastructure/Standards
Working Group

Table of Contents

Acknowledgements

The Justice Reference Architecture (JRA) was developed through a collaborative effort of the U.S. Department of Justice (DOJ) Global Justice Information Sharing Initiative (Global) membership and DOJ's Office of Justice Programs (OJP), Bureau of Justice Assistance (BJA). Global's Infrastructure/Standards Working Group would like to express their appreciation to BJA for their support and guidance in the development of JRA resources, such as this document. These resources provide invaluable assistance to local, state, regional, federal and tribal entities toward the goal of improved information sharing.

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.

The creation of this document was a volunteer effort by numerous contributors, and sincere thanks is extended to them for their hard work in the development of the architecture, Global would also like to recognize the following key contributors.

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. Patrick McCreary—Bureau of Justice Assistance, OJP, DOJ

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

Mr. Chris Traver—Bureau of Justice Assistance, OJP, DOJ

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.

Document Conventions

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

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

Executive Summary

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

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

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

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

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

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

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

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

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.

1.2. An Interoperability Strategy

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

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

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

Step One: Agree on common concepts
Step Two: Agree on the relationships and deliverables
Step Three: Assign the work
Step Four: Produce the deliverables
Step Five: Revise the deliverables

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

1.3. Consensus on the OASIS Reference Model for SOA

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

After considerable discussions among the stakeholders, Global adopted the Organization for the Advancement of Structured Information Standards (OASIS)

Reference Model for Service-Oriented Architecture 1.0 **[SOA-RM]**. OASIS has approved this standard reference model for describing different architectures using comparable, vendor-neutral language. Global is adopting the OASIS framework for describing its architecture and holding conversations with other domains.

1.4. Creating the JRA

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

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

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.

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

1.5. What Is the JRA?

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

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

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

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

1.6. What the JRA Is Not

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

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

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 shared, not on how organizations design, deploy, or operate their systems
- The JRA should encourage information sharing mechanisms and approaches based on open industry standards rather than on approaches proprietary to one vendor, one domain, one level of government, or one specific partner
- The JRA should identify issues on which justice information sharing partners will typically need to reach and enforce agreement, which conversely will identify issues on which they can continue to take independent approaches

Principle: Scalability

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

Rationale

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

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

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

Implications

This principle implies the following about the JRA:

- The JRA should adopt a modular approach that allows jurisdictions to implement a subset of the full architecture, achieving some initial benefit while retaining the option of adopting more of the architecture later
- The JRA should encourage the adoption of industry standards with a broad range of implementations available in the marketplace, from less expensive implementations with modest capabilities, to larger investments that support an increased volume of information sharing
- The JRA should encourage the clear description, the straightforward discovery, and ultimately the reuse of services across jurisdictions to provide information more economically (particularly to smaller jurisdictions)

Principle: Diversity of Data Source Architectures

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

Rationale

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

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

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

Implications

This principle implies the following about the JRA:

- The JRA should encourage the sharing of information and functionality between systems in a way that minimizes the implementation dependencies between them
- The JRA should encourage communication between systems using open industry standards rather than proprietary approaches
- The JRA should use vendor-neutral terminology and concepts in defining the architecture
- The JRA should adopt a modular approach to intersystem communication mechanisms and protocols so that the entire architecture need not change when improved protocols are developed in the future

Principle: Agility

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

Rationale

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

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

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

Implications

This principle implies the following about the JRA:

- The JRA should encourage the sharing of information and functionality between systems in a way that minimizes the implementation dependencies between them
- The JRA should encourage the definition of system interfaces that reflect what the interfaces do, as opposed to how they work
- The JRA should provide mechanisms to separate the logic of information exchange (e.g., the routing and transforming of messages that flow between partners) from the logic of line-of-business systems

Principle: Reuse and Sharing of Assets

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

Rationale

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

Implications

This principle implies the following about the JRA:

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

Principle: Alignment With Best Practices and Experience

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

Rationale

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

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

Implications

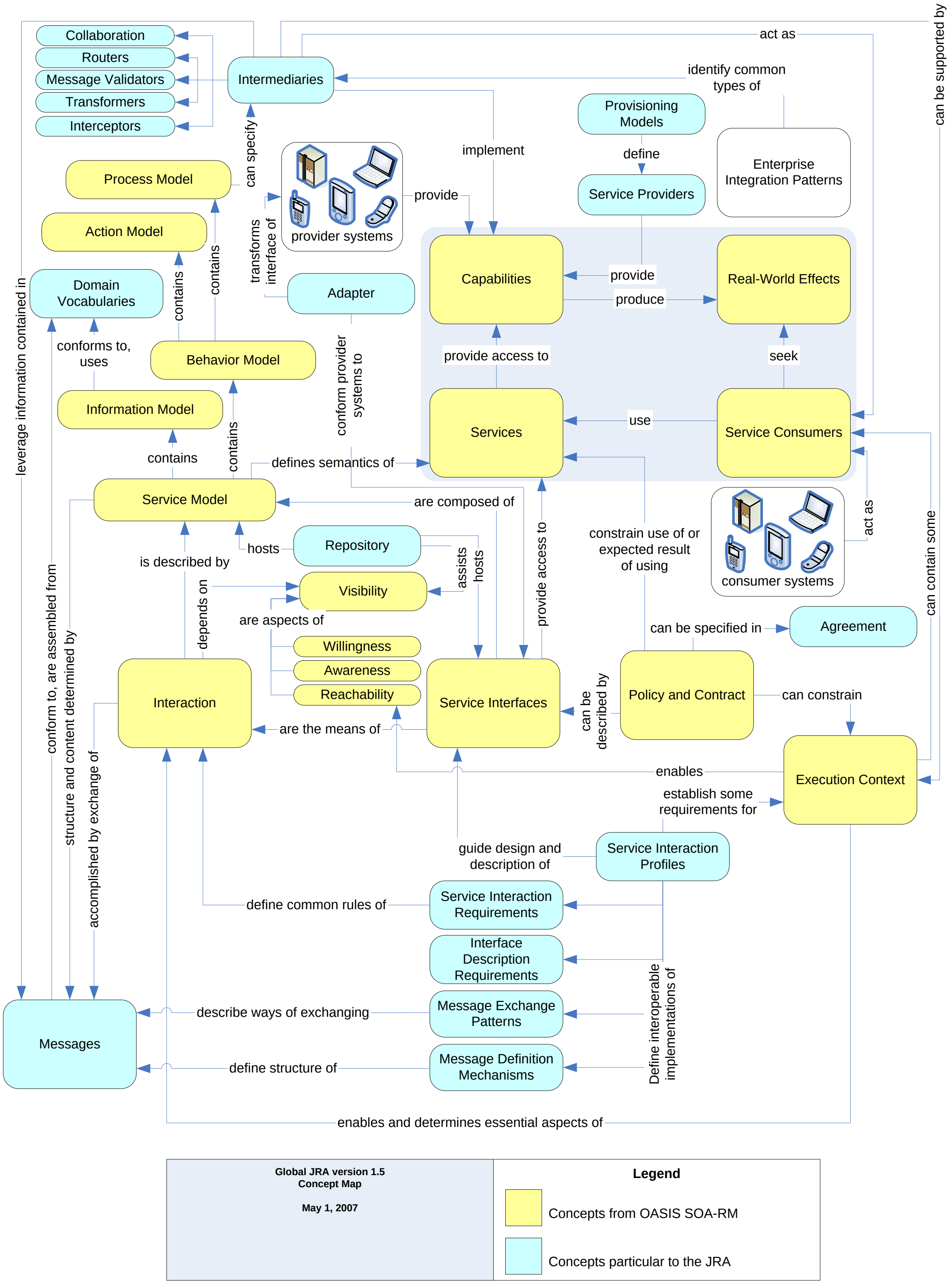
This principle implies the following about the JRA:

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

3. The JRA

3.1. Graphical Overview

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


Collaboration
Routers
Message Validators
Transformers
Interceptors
Intermediaries
act as
can be supported by
identify common types of
Provisioning Models
define
Enterprise Integration Patterns
Service Providers
implement
provide
can specify
provider systems
transforms interface of
Process Model
Action Model
contains
contains
Domain Vocabularies
Adapter
Capabilities
provide
produce
Real-World Effects
leverage information contained in
conforms to, uses
Behavior Model
conform provider systems to
provide access to
seek
Information Model
contains
Services
use
Service Consumers
contains
defines semantics of
Service Model
are composed of
consumer systems
act as
hosts
Repository
assists
hosts
provide access to
constrain use of or expected result of using
is described by
depends on
Visibility
are aspects of
Willingness
Awareness
Reachability
can be specified in
Agreement
can contain some
conform to, are assembled from
structure and content determined by
Interaction
Service Interfaces
can be described by
Policy and Contract
can constrain
are the means of
enables
Execution Context
accomplished by exchange of
establish some requirements for
guide design and description of
Service Interaction Profiles
define common rules of
Service Interaction Requirements
Interface Description Requirements
Define interoperable implementations of
Messages
describe ways of exchanging
Message Exchange Patterns
define structure of
Message Definition Mechanisms
enables and determines essential aspects of
Global JRA version 1.5
Concept Map
May 1, 2007
Legend
Concepts from OASIS SOA-RM
Concepts particular to the JRA

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

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

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

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

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

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

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

In a service-oriented architecture, a **SERVICE** is the way in which one partner gains access to a capability offered by another partner. A partner that uses a service to gain access to another partner's capability is called a **SERVICE CONSUMER**. As with capabilities, the architecture is generic enough to support virtually any kind of service consumer. However, since the purpose of the JRA is to describe an approach to information systems interoperability, the JRA narrows the SOA-RM definition of service consumer to information systems that interact with services directly through an interface that conforms to a service interaction profile (as defined below). The JRA calls such systems **CONSUMER SYSTEMS**.

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

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

Supporting Concepts

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

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

The remaining concepts that address these concerns are called "supporting concepts" and are defined in this section.

Interaction, Visibility, Service Models, and Service Interfaces

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

Interaction

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

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

Visibility

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

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

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

The concept of willingness is related to authorization and access control policies, in that a common reason for lack of willingness to interact is that the consumer is not authorized to conduct the requested interaction. Willingness often manifests in service descriptions, as well as policies, contracts, and agreements (discussed on page 21). A **SERVICE MODEL** is defined as the information needed in order to use, or consider using, a service.

The concept of reachability is closely related to the concept of execution context (discussed on page 22).

Service Models

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

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

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

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

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

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

[1] The 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.

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

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

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

There is a wide variety of ways in which a service can provide access to a capability. In some cases, the provider system that implements the capability may already expose all or some of its functionality as services (through one or more service interfaces, described on page 17). In other cases, the business partner that provisions the capability can purchase an off-the-shelf adapter from the provider system vendor (or a third party) that exposes the system's functionality as a set of services. Finally, the provider system may require reimplementation or custom adaptation to expose functionality as services. This is often expensive and risky, and the desire to avoid this situation should be addressed in the service design guidelines.

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

[2] Principles 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.

same time. The discussion of service interaction on page 13, and the subsequent elaboration of interaction mechanisms in future iterations of the JRA, will reflect the impact of these issues.

Note that the concept of a service in the JRA does not equate to a Web service. The term "Web services" is a label for a family of standards and an associated technical approach to communicating between service consumers and services. The architecture supports flexibility in how this communication happens through the notion of service interaction profiles (discussed on page 19). A Web service profile has been developed for the Web services family of standards; however, the JRA will include additional profiles that adopt other communication mechanisms, such as MQ, JMS, and ebXML. **[WSSIP AND ebXMLSIP]**

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

Service Interface

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

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

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

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

A given service may have multiple interfaces that conform to the same service interaction profile, where the multiple interfaces expose different sets of the service's actions. For instance, a service may have one "query" action and three "update" actions; the query action may be exposed by one Web services interface, while the three update actions may be exposed by a separate Web services interface.

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

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

Design and Description of Service Interfaces

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

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

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

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

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

MESSAGE DEFINITION MECHANISMS are closely related to interface description requirements, described above. Unlike interface description requirements, message definition mechanisms establish a standard way of defining the structure and contents of a message. Note that since a message includes the concept of an "attachment," the message definition mechanism must identify how different sections of a message (for example, the main section and any attachment sections) are separated and identified and how attachment sections are structured and formatted.

Service Interaction Profiles

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

- Service interaction requirements
- Interface description requirements
- Message exchange patterns
- Message definition mechanisms

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

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

Capabilities in Detail

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

Intermediaries

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Service Policy, Service Contract, and Service Agreement

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

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

[3] The 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.

Execution Context

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

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

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

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

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

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

Provisioning Model

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

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.

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

Principle: Reuse and Sharing of Assets

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

- It introduces as one of its core concepts the notion of visibility for services. The concept of visibility recognizes that potential consumers must be aware of the existence of services and, once aware of them, must have clear documentation regarding how to access them.

- It includes service modeling guidelines, which establish clear, consistent rules for the information contained in a service description and how that information must be presented so that potential consumers understand what the service does and how to interact with it.

- It introduces the concept of execution context and guidelines for how to share execution context infrastructure across a group of partners. Thus, instead of each project or pair of partners provisioning its own infrastructure, partners share common infrastructure elements where it is possible to do so.

- It introduces, as part of shared execution context, registries and repositories that store service descriptions and support searches that allow potential consumers to find the services they need quickly. The easier it is for consumers to find services, the more likely they are to reuse them.

Principle: Scalability

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

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

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

Principle: Alignment With Best Practices and Experience

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

- It has been developed by a group of practitioners and technologists from the public sector, national associations, and industry who have gained experience working with service-oriented architecture. It is the result of this group of experienced individuals collaborating and consolidating the lessons learned from SOA implementation projects.

- It leverages accepted standards that have been developed by industry standards bodies, representing a diversity of technologies and vendors. The conceptual framework is based on (and conforms to) the OASIS SOA-RM. Individual JRA deliverables, such as service interaction profiles and service modeling guidelines, further leverage open industry standards such as the Web services stack and UML.

- It builds on and provides linkages between national justice community standards such as NIEM, GFIPM, security, privacy guidelines, etc.

6. Elaboration of Service Interaction Requirements

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.

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

- **Service Authentication:** The ability of a service to provide a consumer with information that demonstrates the service's identity to the consumer's satisfaction.
- **Message Nonrepudiation:** Information provided in a message to allow the recipient to prove that a particular authorized sender in fact sent the message.
- **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.
- **Message Confidentiality:** Information provided in a message to prevent anyone except an authorized recipient from reading the message or parts of the message.
- **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.
- **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.
- **Transaction Support:** Information provided with messages to permit a sequence of messages to be treated as an atomic transaction by the recipient.
- **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.

7. Glossary

Architecture

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

Awareness

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

Behavior Model

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

Business Process Models

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

Capabilities

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

Collaboration

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

Consumer Systems

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

Domain Vocabularies

Includes canonical data models, data dictionaries, and markup languages that standardize the meaning and structure of information for a domain. Domain vocabularies can improve the interoperability between consumer and provider systems by providing a neutral, common basis for structuring and assigning semantic meaning to information exchanged as part of service interaction. Domain vocabularies can usually be extended to address information needs specific to the service interaction or to the business partners integrating their systems.

Enterprise Integration Patterns

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

Execution Context

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

Framework

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

Information Model

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

Interaction

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

Interface Description Requirements

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

Interceptors

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

Intermediaries

Routers and transformers are collectively called intermediaries. This term indicates that routers and transformers generally sit between other services and "mediate" the interaction by managing the transmission of messages between them or by reformatting messages in transit.

Justice Reference Architecture

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

Messages

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

Message Definition Mechanisms

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

Message Exchange Patterns

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

Message Validators

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

Orchestration

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

Process Model

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

Provider Systems

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

Provisioning Models

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

Reachability

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

Real-World Effects

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

Reference Architecture

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

Reference Model

A reference model is an abstract framework for understanding significant relationships among the entities of some environment that 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.

Repository

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

Routers

A capability that receives a message, examines it, and transmits it to one or more destinations based on the contents. In general, routers can be designed to operate on any of the information contained within the message; they may use information about the origin of the message, routing directive information contained within the message or the main content of the message itself.

Services

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

Service Agreements

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

Service Consumers

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

Service Contracts

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

Service Design Principles

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

Service Interaction Profiles

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

- Service interaction requirements
- Interface description requirements
- Message exchange patterns
- Message definition mechanisms

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

Service Interaction Requirements

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

Service Interfaces

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

Service Model

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

Service-Oriented Architecture (SOA)

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

Service Policies

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

Service Providers

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

Transformer

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

Visibility

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

Willingness

A predisposition of service providers and consumers to interact.

8. References

ebXMLSIP GISWG. The JRA ebXML Messaging Service Interaction Profile Version 1.0, March 2009. http://it.ojp.gov/globaljra.

Erl Erl, Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice-Hall, 2005.

GISWG GISWG. *A Framework for Justice Information Sharing: Service-Oriented Architecture*. Global, December 9, 2004.

JRA GISWG. http://it.ojp.gov/globaljra.

Patterns Hohpe, Gregor, and Woolf, Bobby. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison Wesley, 2004. http://www.eaipatterns.com.

Sholler Sholler, Daniel. *Demystifying Service-Oriented Architecture*, META Practice, June 9, 2004.

SOA-RA *Reference Architecture for Service-Oriented Architecture 1.0, Public Review Draft 1*. OASIS, April 23, 2008. http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf.

SOA-REC GISWG. *A Framework for Justice Information Sharing: Service-Oriented Architecture*. Global, December 9, 2004. http://it.ojp.gov/documents/20041209_SOA_Report.pdf.

SOA-RM *Reference Model for Service-Oriented Architecture 1.0, Oasis Standard*. OASIS, October 12, 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf.

WSSIP GISWG. The Global JRA Web Services Service Interaction Profile Version 1.2, March 2009. http://it.ojp.gov/globaljra.

9. Document History

Date	Version	Editor	Change
March 25, 2006	1.0	Scott Came	Initial draft.
July 6, 2008	1.7 candidate	Scott Came	Added concepts of relationships between actions, messages, and the action/process models of a service.
October 30, 2008	1.7 candidate	Monique La Bare	Added service interaction requirements.
November 18, 2008	1.7	Scott Came	New service interface language; Executive Summary update.

Editors

Scott Came	Tom Clarke	Monique La Bare

About Global

The U.S. Department of Justice's Global Justice Information Sharing Initiative (Global) serves as a Federal Advisory Committee to the U.S. Attorney General on critical justice information sharing initiatives. Global promotes standards-based electronic information exchange to provide justice and public safety communities with timely, accurate, complete, and accessible information in a secure and trusted environment. Global is administered by the U.S. Department of Justice, Office of Justice Programs, Bureau of Justice Assistance.

For more information on DOJ's Global and its products, including those referenced in this document, call

(850) 385-0600

or visit

www.it.ojp.gov/globaljra

BJA Bureau of Justice Assistance

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.