Global Justice Information Sharing Initiative

United States
Department of Justice

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

Technical Note RESTful Web Services

Version 1.0

September 2011

Global Infrastructure/Standards
Working Group

This project was supported by Grant No. 2009-DB-BX-K105 awarded by the Bureau of Justice, Office of Justice Programs in collaboration with the U.S. Department of Justice's Global Justice Information Sharing Initiative. The opinions, findings, and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the U.S. Department of Justice.

Table of Contents

Acknowledgements 1

Executive Summary 2

REpresentational State Transfer (REST) Overview 3

Global Reference Architecture Service Interaction Profile Requirements 4

RESTful Resource Access Web (RAW) Services 6

Conclusion 9

Acknowledgements

This Technical Note does not reflect a normative policy position from the Global Justice Information Sharing Initiative (Global) community of stakeholders. Rather, it is intended to inform participants and practitioners on architectural compatibility issues known to exist between the Global Reference Architecture and REpresentational State Transfer (REST) services.

Global would like to thank the Global Infrastructure/Standards Working Group (GISWG), Services Implementation Committee and Mr. Bob Slaski, principal author of this technical note, for their continued volunteer efforts as dedicated participants in the Global initiative. Any questions or comments should be directed to the chair of the GISWG Services Implementation Committee, Mr. Scott Came.

Executive Summary

REpresentational State Transfer (REST) is a movement or architecture that is not defined or specified in a manner that conforms to Global Reference Architecture service requirements. It is conceptually simple, but part of that simplicity is achieved by compromising foundational requirements, such as security. While REST is built on well-defined transport services, all additional requirements, such as data representation, reliable exchange acknowledgements, message-level security, message exchange patterns, etc., are not standardized. Substantial effort would be required to adequately specify the use of REST services, and the resulting specification would not realize the benefits and leverage associated with industry standards. As a result, the development of a Global Reference Architecture REST Service Interaction Profile (SIP) is not planned.

Efforts are under way to clearly document the common framework of the Global Reference Architecture Web Services SIPs and the Global Federated Identity and Privilege Management (GFIPM). The GFIPM System-to-System Profiles are based exclusively on the use of Web services standards. The Logical Entity eXchange Specification (LEXS) is increasingly adopting Web service standards with LEXS version 4.0. While not strictly required, Global Reference Architecture Web services will likely serve as the infrastructure for the vast majority of LEXS implementations. The use of REST is not recommended for organizations that wish to adopt Global products, such as the Global Reference Architecture Web Services SIPs and GFIPM System-to-System Profiles, in conjunction with LEXS 4.0 or later.

There may be circumstances in which an exchange does not need to meet many of the Global Reference Architecture requirements. For example, there may be an existing secure network, and the network may meet security requirements. Likewise, the requirements of a law enforcement exchange may not require protocol-based message reliability when the overall system has a very high reliability and there are application-level mechanisms in place to ensure proper receipt. In these situations, the use of REST may be adequate and appropriate.

REpresentational State Transfer (REST) Overview

REpresentational State Transfer (REST) is a "movement" to promote the development of services on the World Wide Web using the same architecture that is currently used by users to interact with Web resources. REST has been appropriately referred to as Resource Oriented Architecture. The notion of REST is derived from the early work of Roy Fielding and his dissertation, *Architectural Styles and Design of Network-based Software Architecture*.

There are no REST specifications. Rather, REST relies on the use of the World Wide Web HyperText Transfer Protocol (HTTP) as the underlying resource access model. Advocates of REST tend to view "big" Web services as too complex and unnecessary. "Big" Web services refers to traditional Web services, often referred to as "WS-*." The paper, *RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision*, 17th International World Wide Web Conference (WWW2008), Pautasso, Cesare; Zimmermann, Olaf; Leymann, Frank (April 2008), provides guidance on the appropriate architectures for using REST.

Since there are no definitive specifications for REST, much of this Technical Note is based on Internet research. The book *RESTful Web Services*, by Leonard Richardson and Sam Ruby, was used extensively as a reference.

The REST architecture holds that the resource-oriented model of the World Wide Web can and should be applied to most services. The definition of a resource is very broad and essentially identifies anything that has an address, a representation, and a means of linking to other resources. The address is the Uniform Resource Indicator (URI), and the representation can be essentially any media type. Linking is provided by the ability to include URIs within the resource representation.

The key principles of the Resource Oriented Architecture (ROA) are:

- ✓ Uniform interface
- ✓ Statelessness
- ✓ Addressability
- ✓ Connectedness (links)

The uniform interface for REST services consists of the basic methods supported by the HTTP protocol: POST, GET, PUT, DELETE, plus the lesser-used methods of HEAD and OPTION. The HTTP methods—PUT, GET, POST, DELETE—correspond to the notions of create, read, update, and delete.

Because the definition of resource is so broad, any addressable item could be defined as a resource and managed using the uniform interface. For example, a record within a file could be considered a resource, and that resource could be retrieved using REST GET.

Statelessness is provided by the atomic nature of REST services. Any method applied to a resource is viewed as a single autonomous operation. The World Wide Web provides statelessness using autonomous methods and by providing repeatability. Repeatability is more specifically defined in the notions of safety and idempotence. Safety means a GET or read transaction can be repeated with the same results. There are no side effects. Idempotence means that an operation that performs a change can be repeated and the results will be the same. The REST concept works well for applications such as simple file record operations. However, it is inadequate for more complex operations such as debiting an account.

There are two major problems in attempting to use REST as a formal service interaction profile. First, it is not standardized. There are no formal specifications, and implementations vary. Second, REST does not address more complex interaction requirements such as guaranteed reliability or security—a fundamental component and requirement of the justice community of users who the Global Reference Architecture serves.

Global Reference Architecture Service Interaction Profile Requirements

The Global Reference Architecture defines the potential requirements for service interactions. Specific interactions may not need to support all of the potential Global Reference Architecture requirements. While there are no formal standards for REST services, the table below summarizes how REST services might accomplish the requirements associated with a Global Reference Architecture SIP. For reference purposes, the table also shows the associated WS-* specifications.

Potential Global Reference Architecture Requirement	Potential REST Approach to Requirement	WS-* Specification
Service Consumer Authentication	✓ Nonstandard ✓ HTTP Authorization with extensions, e.g., X-WSSE	✓ WS-I Security Profile 1.1 ✓ WS-SecureConversation ✓ GFIPM w/SAML 2.0 ✓ WS-Trust

Potential Global Reference Architecture Requirement	Potential REST Approach to Requirement	WS-* Specification
Service Consumer Authorization	✓ Nonstandard ✓ HTTP Authorization with extensions, e.g., X-WSSE	✓ WS-I Security Profile 1.1 ✓ WS-SecureConversation ✓ GFIPM w/SAML 2.0
Identity Attribute Assertion Transmission	✓ Open Authorization	✓ GFIPM w/SAML 2.0
Service Authentication	✓ Nonstandard	✓ WS-I Security Profile 1.1
Non-Repudiation	✓ Nonstandard using timestamp	✓ Timestamp w/XML Signature
Reliability	✓ Nonstandard ✓ Stateless, idempotent transactions	✓ WS-ReliableMessaging
Message Integrity	✓ Nonstandard using digital signature	✓ WS-I Security Profile 1.1 ✓ XML Signature
Message Confidentiality	✓ Transport Layer Security ✓ HTTPS ✓ Nonstandard XML Encryption ✓ FIPS 140-2	✓ WS-I Security Profile 1.1 ✓ XML Encryption ✓ FIPS 140-2 ✓ Transport Layer Security
Message Addressing	✓ Uniform Resource Identifier (URI)	✓ WS-Addressing
Transaction Support	✓ Nonstandard	✓ WS-AtomicTransaction ✓ WS-BusinessActivity ✓ WS-Coordination
Service Metadata Availability	✓ Nonstandard ✓ HTTP OPTION	✓ WS-MetadataExchange
Interface Description	✓ WSDL 2.0 ✓ WADL	✓ WSDL 1.1, 2.0

Potential Global Reference Architecture Requirement	Potential REST Approach to Requirement	WS-* Specification
Message Exchange Patterns	✓ Resource-oriented ✓ HTTP GET,PUT,DELETE, POST,OPTION,HEAD	✓ Request-Response, One-Way ✓ WS-Notification
Simple Message	✓ XML or non-XML	✓ XML ✓ SOAP
Composite Message	✓ MIME	✓ XML Infoset
Binary Data	✓ Media Type—MIME, linked multimedia (mashup)	✓ XML-Binary Optimized Packaging ✓ Message Transmission Optimization Package

This table is not meant to be exhaustive. Rather, it is intended to highlight that many of the requirements would need to be satisfied in a *nonstandard* manner. While it is possible to define these custom approaches, the viability and cost of using, and possibly maintaining these specifications would be prohibitive and counter to the efficiencies and benefits the Global Reference Architecture community strives to realize.

The absence of a formal interface specification for REST interfaces would be difficult for implementers. Web services descriptions (WSDLs) provide a well-defined interface for services. While the use of NIEM makes WSDLs more complex, they still provide a standard service description which is not available with REST interfaces.

The simple, stateless nature of REST interfaces allows applications to operate with high reliability and can mitigate the need for guaranteed reliability.

No standard security framework exists for REST. Major corporations, such as Amazon, that use REST have implemented their own unique security specifications. The Open Authorization (OAuth) Core specification is one specification that could be used, but there is no clear consensus with respect to REST security.

RESTful Resource Access Web (RAW) Services

The W3C Web Services Resource Access Working Group (http://www.w3.org/2008/11/ws-ra-charter.html) is in the process of finalizing Web

Services for Resource Access, also known as Resource Access Web Services or RAW Services. The earlier Resource Access efforts were focused on management services. However, there is an increasing interest in RAW Services as a means to implement RESTful services. RAW Services are essentially traditional Web services constrained and adapted to a resource view. The specifications assume the use of the WS-I Basic Profile and WS-Addressing. RAW Services can be developed that are fully compliant with the Global Reference Architecture Web Services Service Interaction Profiles.

RAW Services specifications are currently in candidate recommendation working draft. The specifications include WS-Transfer, WS-ResourceTransfer, WS-Enumeration, WS-Eventing, and WS-MetadataExchange. These specifications are useful even as candidate recommendations to adapt Web service implementations to resource orientation. The standardization efforts are being supported by major vendors including Microsoft, IBM, and Oracle (Sun).

A high-level description of each RAW Service standard is provided below.

Standard	Description
WS-Transfer	Create, Read, Update, Delete (CRUD) access to Resources
WS-ResourceTransfer	Field-level (subresource) access using Xpath
WS-Enumeration	“Result set” access using a cursor
WS-Eventing	Subscriptions/notices with both push and pull options
WS-MetadataExchange	Service metadata

How do RAW Services work? RAW Services use SOAP Request-response message exchange patterns to implement basic “CRUD” operations; Create, Read (Get), Update (Put), Delete. Resources are defined via WS-Addressing Endpoint References (EPR). Because SOAP[1] is specified using XML, resource address and resource value data is consistently delivered (in XML). A typical example of RAW Services might be the reading or updating of file records or table rows specified as resources. The table below contrasts REST and RAW concepts.

[1] SOAP is the underlying XML format for defining a message header and body.

RAW Concept	REST Concept
Resource	Resource
WS-Transfer operation	HTTP Action
WS-Addressing Endpoint Reference	URI
XML w/binary using XOP/MTOM	Media type
Stateless or “stateful” as negotiated	Stateless (client-based state)

A RAW Get identifies the resource in the SOAP header. The GetResponse provides the resource content in the SOAP body. A RAW Put identifies the resource in the SOAP header and the SOAP body has resource values to be updated. The PutResponse provides acknowledgement. An example is shown below with a RAW Service getting vehicle data and a corresponding REST service. The RAW Service standards include WS-Eventing, which defines rich push or pull subscription services with delegation. The broad corporate representation on the Web Services Resource Access Working Group would appear to indicate that the long-standing split between WS-Eventing and WS-Notification will be resolved in favor of the new WS-Eventing standard. The WS-Enumeration specification provides for client-based “cursor” control of a “result set” access, e.g., next 10. This has broad applicability in searches which result in multiple “hits” that must then be paged for subsequent refined searches.

RAW Service	REST HTTP
HTTP POST <S11:Header> <wsa:To>/NCIC/transfer/Get <\wsa:To> <VIN>ABCDEF0123456789<\VIN> <\S11:Header> <S11:Body\>	HTTP GET /NCIC/QV/VIN/ ABCDEF0123456789
HTTP 200 OK <S11:Header> <wsa:To>/NCIC/transfer/GetResponse <\wsa:To> <VIN>ABCDEF0123456789<\VIN> <\S11:Header> <S11:Body> <VehicleTag>Vehicle data<\VehicleTag> <\S11:Body>	HTTP 200 OK <Message> <VIN>ABCDEF0123456789<\VIN> <VehicleTag>Vehicle data<\VehicleTag> <\ Message >

RAW Services are specified for the widely supported WS-I Basic Profile 1.1 and composable with other WS-* specs. In particular, RAW Services are fully composable with WS-Security and with WS-ReliableMessaging. Stateless operation and application-level acknowledgement may provide adequate reliability for many applications. RAW Services cannot fully leverage WSDL because resource identifiers (parameter equivalents) are specified in the SOAP header. Also, like REST services, RAW Services are not suitable for more complex or long-running transactions.

Conclusion

REST is an architectural style, not a specification. It is a popular commercial alternative to SOAP-based Web services. It is based on HTTP and the concept of uniform methods to access information as resources. REST can be used in conjunction with a number of best practices to meet Global Reference Architecture requirements, but these solutions are not standardized and represent an ad hoc approach. As a result, it is not possible to adopt REST without very substantial work to develop interoperable specifications. Development of a REST Service Interaction Profile would require considerable effort and would likely have limited adoption. We are not aware of any major justice information sharing initiative based on REST with the exception of the FBI CJIS Division, which is planning to use REST for internal exchanges.

There are no consistent REST specifications that allow REST implementations to meet the service interaction requirements of the Global Reference Architecture. Further, the Global Federated Identity and Privilege Management (GFIPM) System-to-System Profile is based entirely on the use of SOAP and Web services standards. Efforts are under way to clearly document the common framework of the Global Reference Architecture and GFIPM. Further, the Logical Entity eXchange Specification (LEXS) is increasingly adopting SOAP-based Web service standards with LEXS version 4.0. LEXS 4.0 adopts WS-Addressing, WS-Notification and other SOAP-based Web services standards. While not strictly required, SOAP-based Web services will likely serve as the infrastructure for the vast majority of LEXS implementations.

Two common rationales are often given for the use of REST. The theoretical rationale is that the resource view provides greater benefits than a service view. While there is a reasonable basis for this argument, there would need to be considerable effort to model current exchanges as access to resources, and much of the current effort defining and developing standard services would be lost. In addition, it would seem more difficult to adopt a resource view where the resource (e.g., person) is not readily identified, as is often the case in an investigation.

The second rationale, which is often put forward by technical staff, is that REST is easier. REST appears easier because it is not constrained by interoperability specifications and does not address more difficult issues such as security. However, because of this lack of specification, REST solutions are typically proprietary, noninteroperable, and inconsistently secured. Major commercial organizations, such as Amazon, are large enough to dictate proprietary REST interfaces and may even derive some business benefit from having a proprietary interface. Some government systems, such as Federal Bureau of Investigation (FBI) CJIS systems, may have the clout and reach to justify nonstandard interfaces, but state and local agencies will need to rely on well-defined standards-based solutions to achieve interoperability.

Many organizations are implementing security solutions that provide security and privacy enforcement using standard commercial gateway products such as the IBM DataPower appliance. Such products can process security and privacy elements automatically when Web service standards are used. Since REST implementations do not follow a standard security approach, gateways may not support the specific security approach used for the REST implementation. This is a significant limitation of REST use in the criminal justice community.

Most criminal justice and public safety organizations will adhere to the Global Reference Architecture and use Global Reference Architecture Web Services Service Interaction Profiles (SIPs) in conjunction with GFIPM federated identity and Web services LEXS federated queries. Since these organizations will already need to support SOAP-based Web services in accordance with the Global Reference Architecture SIP, any REST offering will be an addition to the SOAP-based Web services, not a replacement. Most organizations will also need to support a legacy interface and will not have the resources to support two different new interfaces—SOAP-based Web service and REST. The legacy interface will likely suffice for those consumers who want a simpler but less secure interface such as REST offers. This same approach is likely to prevail when considering REST for internal transactions. Most likely, internal transactions will remain in legacy formats until applications can be transitioned to new services. Since new services will need to support Global Reference Architecture Web services, there will likely be resistance to implementing a second interface using REST.

Where desired, it is also possible to build "RESTful" applications using Global Reference Architecture-compliant Web services. The core of REST is stateless services, which are defined as access to resources using a simple uniform interface (e.g., Get). Existing Web services specifications, referred to as Resource Access Web (RAW) Services, provide secure, reliable transacted services using proven Web services standards within the broad framework of a resource-oriented approach. RAW Services leverage the benefits of SOAP and XML but also incorporate many of the benefits of the REST architecture, including uniform interface and statelessness. RAW Services allow for more automated, consistent service and client development because of the uniform service interface and well-defined operations.

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/globalgra

Bureau of Justice Assistance
U.S. Department of Justice