



RESTful Web Services

XML AUSTIN USERS GROUP
APRIL 9, 2003

Oliver Schmelzle - Tonic Software, Inc.

Presentation Overview

- Web service definition
- Introduction to the REST architectural style
 - What it is in a nutshell
 - History and evolution
 - Recent discussions
 - How to build “better” Web applications
- REST in context with RPC-based Web services
 - Highlight the “Web services trinity”: SOAP, WSDL, UDDI
 - REST influence on future standards
- Web services in the wild
 - Google SOAP API
 - Amazon REST and SOAP API

What Is A Web Service?

- This is the official W3C take on a Web service definition:

“A Web service is a software system identified by a URI, whose public interfaces and bindings are defined and described using XML. Its definition can be discovered by other software systems. These systems may then interact with the Web service in a manner prescribed by its definition, using XML based messages conveyed by internet protocols”

from “Web Services Architecture” at <http://www.w3.org/TR/ws-arch/>

- Common sense approach
 - Geared towards machine-to-machine vs. machine-to-user communication
 - Often described as XML-over-HTTP style communication
 - Screen scraping back in the old days

REpresentational State Transfer (REST)

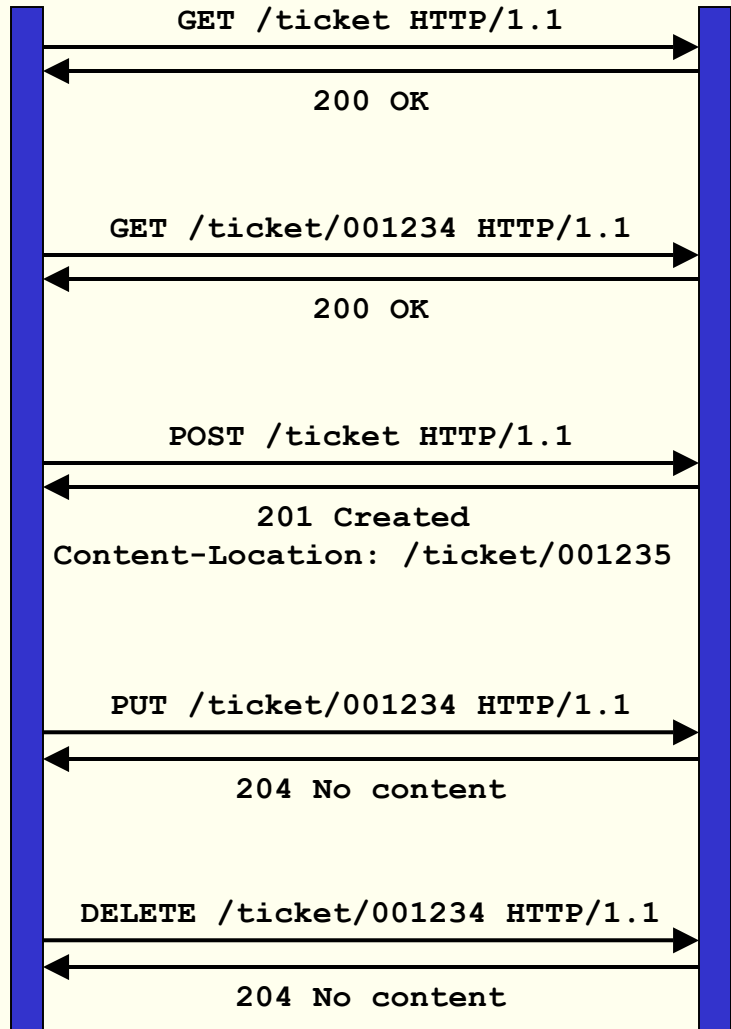
- REST was articulated by Roy Fielding in his dissertation as an underlying concept for today's web architecture
 - A set of constraints to model the modern Web architecture
 - REST principles were applied to HTTP and URI specs
 - Visible in the evolution of HTTP
- An architectural style is
 - Not a protocol
 - Not a specification
- Who is Roy Fielding?
 - Co-author of the HTTP and URI RFCs
 - HTTP 1.0 (RFC 1945)
 - HTTP 1.1 (RFC 2616, RFC 2145, RFC 2068)
 - Uniform Resource Identifiers (RFC 2396, RFC 1808)



What Are RESTful Web Services?

- Applied XML, HTTP and URI
 - Follow “nature of the Web”
 - HTTP semantics as building blocks for applications (e.g. CRUD)
 - Simple approach, often unintentionally used
- Every resource has a URI
 - Unified namespace
 - Every resource is addressable
- Leverage the semantics and functionality of HTTP
 - HTTP actions: GET, POST, PUT, DELETE
 - Status Messages: 200 Ok, 201 Created, ...
 - Content Negotiation (“conneg”)
- Leverage full feature of XML
 - Namespaces
 - Schemas

RESTful Example Application



Retrieve a list of all current tickets

Retrieve a particular ticket

Create a new ticket. Retrieve new ticket URI from HTTP Headers.

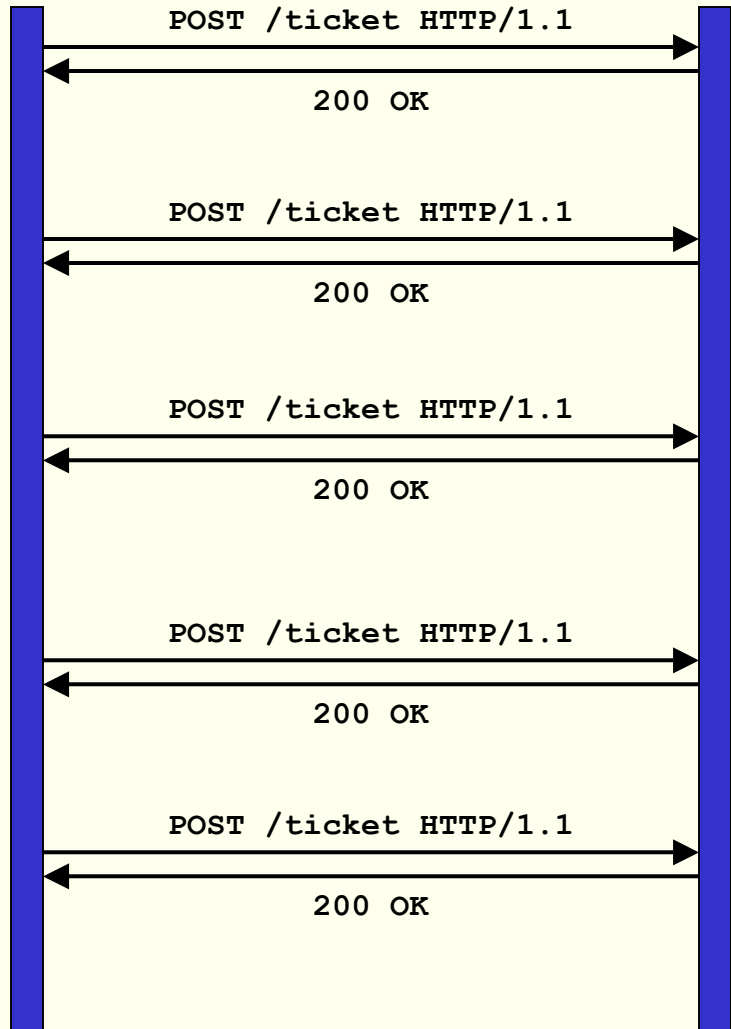
Update a ticket.

Delete a ticket.

Simple Object Access Protocol – SOAP 1.1

- Generic XML-based RPC protocol
 - Initial specification only covers HTTP-binding
 - Use of XML Schema
 - Multiple protocol bindings available today (SMTP, FTP, JMS, etc.)
 - RPC-style and document-style communication
 - Major protocol improvements for SOAP 1.2
- XML-RPC was precursor to SOAP
- Estranged use of HTTP in SOAP 1.1
 - Breaks certain HTTP semantics – e.g. caching
 - Breaks URI concept
 - Resources are not directly addressable
 - Only endpoints have URIs

SOAP Example Application



Retrieve a list of all current tickets

Retrieve a particular ticket

Create a new ticket.

Update a ticket.

Delete a ticket.

Web Service Definition Language – WSDL 1.1

- From the spec: “XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information.”
- Independent of message format or underlying protocols
- Not exclusively tied to SOAP
- Several bindings, e.g HTTP GET/POST
 - REST-style interface can be described with WSDL
- Upcoming WSDL 1.2 specification

Web Services At Google And Amazon

- In 2002 Google and Amazon introduced Web services APIs
 - Google followed a SOAP-only strategy
 - Amazon offers both SOAP and REST style APIs
- Examples of large scale Web services deployment
- Heated discussions in the community
- Recent Tim O'Reilly remark
 - “85% of the users of Amazon's web services API choose the REST interface over SOAP.”

Google Web Services API

- SOAP API for accessing Google
 - Requires registration for developer token
 - Limit of 1000 queries per day
 - According to license terms only way to access Google programmatically
- Three available SOAP methods
 - doGoogleSearch
 - doSpellingSuggestion
 - doGetCachedPage
- Hidden/non-public XML-over-HTTP interface
 - <http://www.google.com/xml?q=xml%20austin>
 - DTD can still be downloaded: <http://www.google.com/google.dtd>

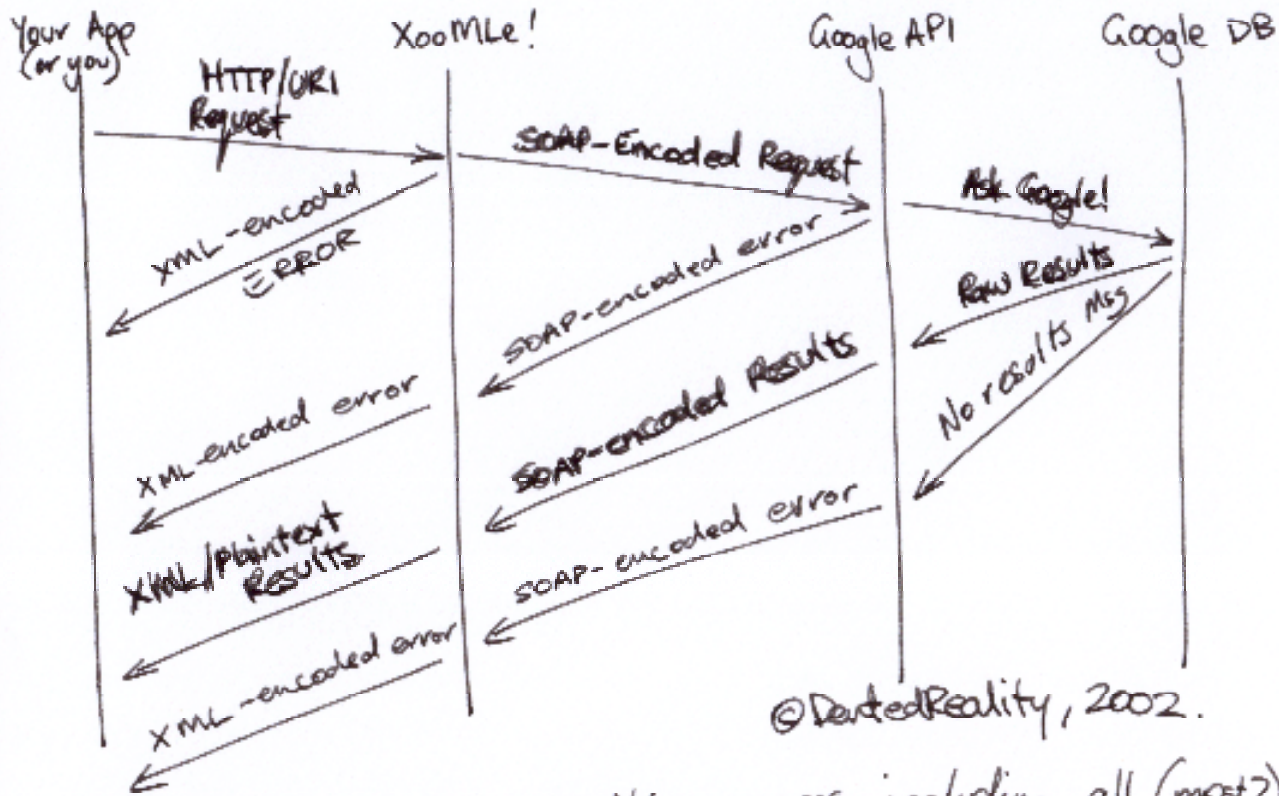
Enter XooMLe



- RESTful interface to Google API
- Translates REST-style to SOAP-style and back
- Hosted gateway service
<http://www.dentedreality.com.au/xoomle/>
- Full expressiveness of original SOAP API
- Google Method *doGoogleSearch* becomes
<http://xoomle.dentedreality.com.au/search/?key=devkey&q=xml%20austin>
- Additional features
 - Automatic looping for larger result sets
 - XSLT transformation within XooMLe

XooMLe Architecture

http://www.dentedreality.com.au/xoomle/docs/media/system_diagram.gif



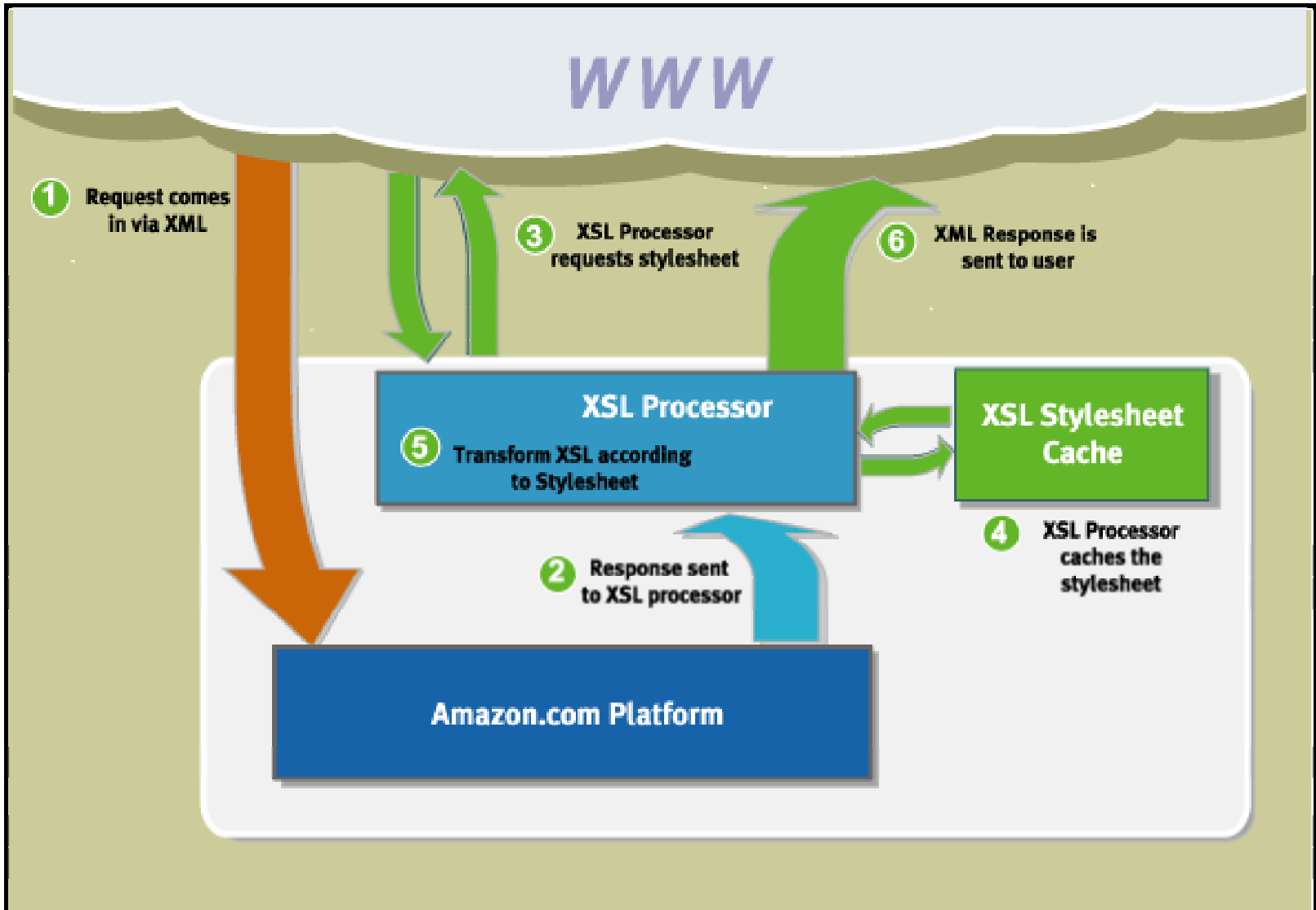
©DentedReality, 2002.

Moving L→R, T→B, describing process, including all (most?) possible process-flows.
Successful Request-Response is "bold"

Amazon Web Services APIs

- Powerful Web services API for accessing Amazon services
 - Dual approach: REST-style and SOAP-style
 - Dual approach: DTDs and Schemas
 - Many available methods
- Complex functionality
 - Remote shopping cart
 - XSLT transformations

Amazon REST Architecture



What The Future Might Hold..

- Towards RESTful SOAP
 - SOAP 1.2 to be finalized summer of 2003
 - Full support for HTTP 1.1 binding
- SOAP implementations will have to follow
- Continued lively community discussions to be expected

REST Resources

- RESTWiki - <http://internet.conveyor.com/RESTwiki/moin.cgi/>
 - Discussion and development of REST ideas
- Paul Prescod - <http://www.prescod.net/rest/>
 - Many excellent articles and presentations on the topic
 - Some examples in this presentation are inspired by Paul's writing
- REST mailing list - <http://groups.yahoo.com/group/rest-discuss/>
 - Active discussions on REST and its application to real world problems
- Roy Fielding's Dissertation
 - http://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf
- Google API
 - <http://www.google.com/apis/>
- Amazon API
 - <http://associates.amazon.com/exec/panama/associates/ntg/browse/-/567634>

Tools Used To Create This Presentation

- jEdit 4.1
- Ethereal 0.9.11
- Apache Axis 1.1RC2
- Apache Tomcat 4.1.24
- Mozilla 1.3 and Phoenix 0.5+
- The Mind Electric GLUE 4.0
- Google API Code
- Amazon API Code