Web Services Architecture Directions

Rod Smith, Donald F Ferguson, Sanjiva Weerawarana
IBM Corporation

Overview

- Today's Realities
- Web Services Architecture Elements
- Web Services Framework
- Conclusions & Discussion

Today's Realities

- HTML forms are a dominant form of client-server computing
- *Ad hoc* B2B occurs today via XML over HTTP
- Traditional B2B has focused on well-defined, standard message formats and protocols (e.g., RosettaNet, cXML)
- Common problems: messaging, message driven processing, message brokering
- Vertical groups are re-inventing the entire stack daily

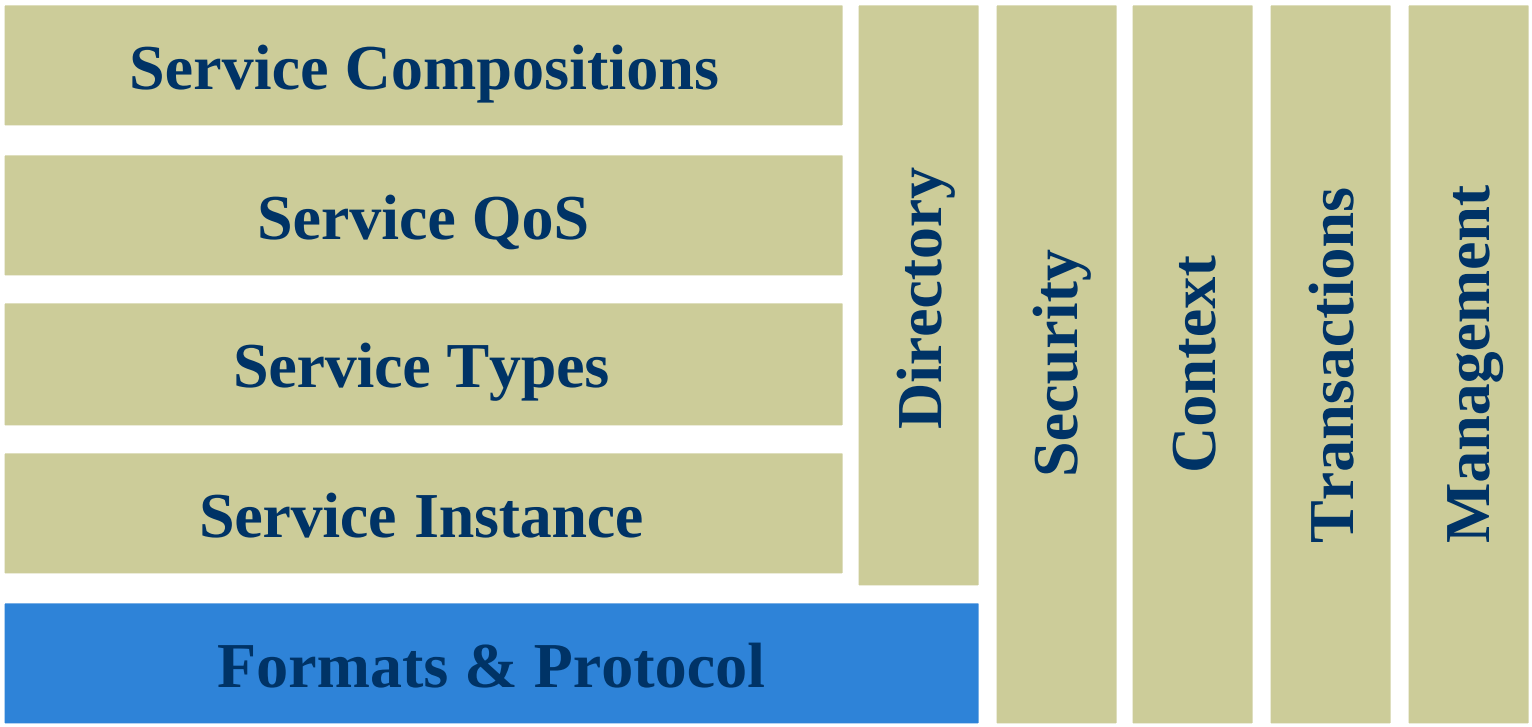
Web Services

- Web services initiative is attempting to provide a common base to build on
- Web services is a new adaptive distributed computing platform built on deployed network infrastructure including XML & HTTP
- **Composing & choreographing *application* components on a large scale**

What's new about Web services?

- Finding services *by-what* and *by-how*
  - Traditional distributed computing searched by name
- Looser coupling via less reliance on pre-defined interfaces
  - The growing role of the "service broker" to mediate between requestors and providers
  - No single point of failure
- Does not assume single implementation technology
  - Same concepts must scale from simple HTTP POST of XML to robust, reliable MOM
- Web services are starting with an ad hoc view instead of systematic approach
  - Focus is more on shorter term partnerships and collaborations instead of long term relationships

Web Services Framework



What do we need to do?

- Define a Web services architecture consisting of several complimentary layers
  - Message Exchange and Function Calls
  - Security
  - Conversations and Activities
  - Business Processes and Agreements
  - Manageability
  - Intermediaries
  - Context Awareness
  - Directory Binding

XML Protocol

- *Lingua franca* of the Web services stack
- Simple, reliable messaging is a key feature
  - Different QoS: best-effort, at-least-once, at-most-once, exactly-once
- XML Protocol must be available over multiple transports with different native QoSs

Security

- Web services require an end-to-end security story
  - E.g., Web service request originates inside enterprise, travels via TIBCO, HTTP, MSMQ to service provider.
- Requirements:
  - End to End Authentication
  - End to End Authorization
  - End to End Integrity
  - End to End Confidentiality.
  - Audit & Non-repudiation

Conversations and Activities

- Concept of “transactions” is a key part of application processing
- Web services require incremental QoS for “transactions”
  - An *activity service* which allows one to define the operational context of one or a series of requests, controlling duration and participants
    - See OMG Extended Structuring Mechanism (Activity Service JSR)
  - A *conversation service* which provides interaction styles for Web services
    - Request atomicity
    - Conversations
    - …

Business Processes and Agreements

- We see three levels of functions support multi-party business processes and "workflows"
  - Operational descriptions of services (WSDL)
  - Non-operational, behavioral descriptions of services (WSEL)
  - Composing and choreographing services to build larger business processes (WSFL)

- Agreements are a way to augment end-point definitions with
  - Roles that can invoke services/operations in their organization
  - Concrete values for parameters of end-point definition
  - SLAs

Web Services Description Language (WSDL)

- WSDL addresses the problem of providing a machine-readable description of a service
- More than an IDL – has everything needed to access service
- Model:
  - Types, Messages, Operations, Interfaces
  - Bindings, Ports, Services
- WSDL is a component definition language for Web service components.

WSDL Goals

- Provide a unifying way of describing new and existing services
- The language must be open with respect to
  - service platforms (multiple type systems)
  - service interaction types: message/procedural
  - protocols: WSDL is extensible to allow new protocol bindings
    - XML Schema, SOAP-XMLP will be dominant choices.
- Provide the base for a Web services component composition framework

Web Services Endpoint Language (WSEL)

- An “end-point” is more than a WSDL port(type)
  - QOS characteristics
  - Sequencing of operations (open, read/write, close…)
  - Cost characteristics
  - Security characteristics
- All of this is needed for composition and choreographing services into larger business processes
- (Under development)

Web Services Flow Language

- WSFL supports two types of composition and choreography:
  - *Flow models*: describes business processes
  - Global models: describe overall partner interactions
- Flow models
  - Describe how to choreograph the functionality provided by a collection of Web services to achieve a particular business need
- Global models
  - Describe how a set of Web services interact with each other
- (Under development)

Manageability

- Services need basic management interfaces
- Services have full autonomy over their infrastructure and management
  - Should publish basic reporting and recovery interfaces
- Management interfaces should be described in WSDL with appropriate bindings
  - E.g.: JMX for Java platform

Intermediaries

- Web services has intrinsic support for intermediaries
  - E.g.: An in-network 3rd party non-repudiation service
- Service providers would specify intermediaries support / required
  - WSFL-based “public flow” is specified as part of service description
- Intermediaries open the door for value added services

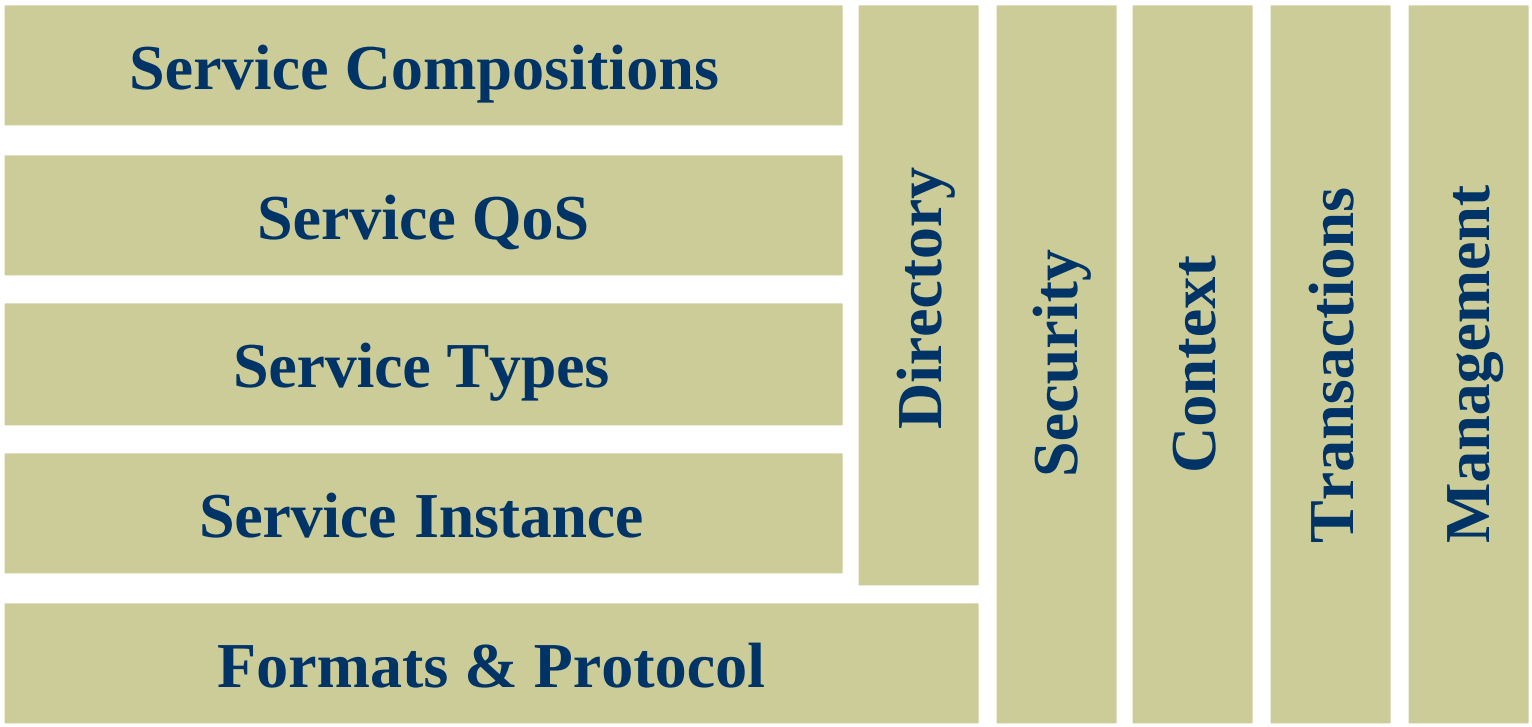
Context Awareness

- "Intelligent" Web services refer to being aware of the context of the user / system / provider:
  - Device type
  - User profiles
  - Temporal information
  - Geographic information
- Protocols must support context propagation
- Contexts themselves will be defined by verticals

Directory Binding

- A key value of the Web services platform is support for by-what and by-how service discovery and binding
  - Directories are key enabler
- Service providers describe their services using WSDL, WSEL and WSFL and requestors query the directory to find matching services
- Directory-driven delayed binding within Web services (compositions) enables dynamic e-business

Web Services Framework



Position Papers

- The 62 position papers are a rich base to start filling in the pieces of the web services platform
  - Several “framework” papers
  - Several “component” papers

- Attempt at classifying “component” papers:
  - Formats & protocols: 39, 40, 41
  - **Service descriptions, QoS, compositions: 10, 19, 20, 26, 28, 30, 49, 55,**
  - Directory: 9, 37, 43, 57,
  - **Security: 16, 22, 23, 47, 50**,
  - Context: 29,
  - Transactions: 1, 46,
  - Management: 32,

Discussion and Conclusions

- To realize the Web services stack, the following standards efforts are needed
  - Reliable messaging support over XMLP
  - End-to-end security model
  - Service descriptions, end-point descriptions and business process compositions
  - Transactional support via activities and conversations
  - Extending directory functionality in UDDI and enabling flexible query and “look up” of services
  - Propagating application context in service interactions

If we build it, will they come?

- Will these differences make Web services the distributed computing platforms that truly achieves ubiquity?
- Right technology, at the right time?
  - KISS