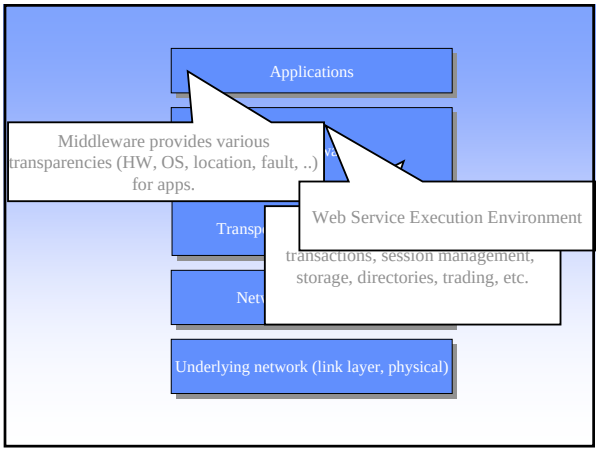
Services and Identity Management

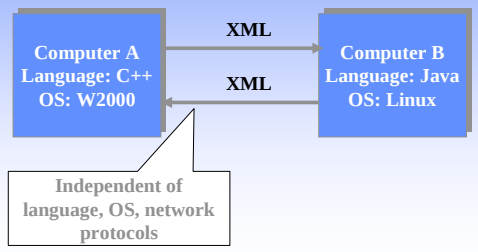
3.12.2007

Dr. Sasu Tarkoma

Contents



A Basic Web Service



Applications and Services

- We have become to rely on a number of Internet services
  - Basic connectivity and data transport (TCP/UDP)
  - SMTP
  - Web sites
  - Web services (SOAP / REST)
  - Signalling (SIP, VoIP)
- We expect many things from services
  - Availability
  - Trustworthiness
  - Usability
  - Interoperability
  - …

Web applications

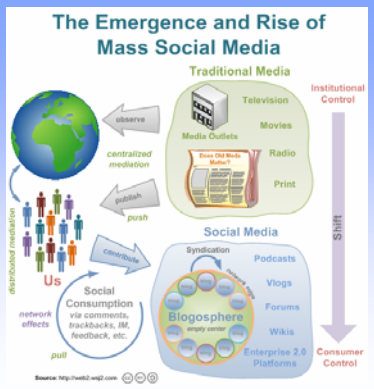
- Recent trend has been to develop web applications
  - Traditional applications on Internet (office suites,..)
  - Search (Google, Yahoo, ..)
  - Instant communications and presence
  - Social collaboration and networking sites
  - Data sharing sites and video sharing
  - Data storage services
  - Blogging
- Another recent trend is to simplify signing to services
  - Single Sign-On, federated identity, OpenID
- And creating mashups
  - Combining services in new ways
  - Custom experience and personalization

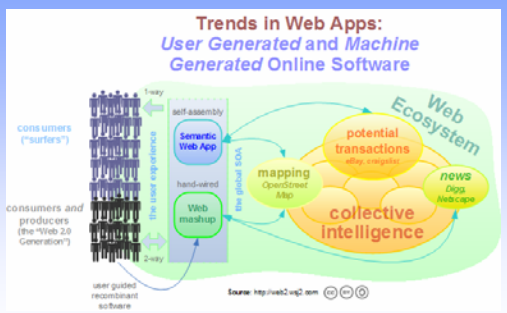
Well-known Services

- Social networking
  - Myspace, Facebook, ..
- Start pages
  - Windows live, Netvibes, ..
- Social bookmarking
  - Del.icio.us, trailfire, ..
- Peer production news
  - Netscape, digg, newsvine,..
- Social media sharing
  - YouTube, jumpcut, ..
- Online storage
  - Amazon S3, Jungle Disk, omnidrive, ..

Challenges

- There are many challenges for Internet services
  - Availability
    - Denial of service attacks
    - How ensure mashups work and are available
  - Trustworthiness
    - Phishing and spam in its many forms
    - Identity theft
    - Privacy
    - How to enforce security
  - Federation
    - How to identify users and authenticate and authorize
    - How to scale mashups





Challenges

- There are many challenges for Internet services
  - Availability
    - Denial of service attacks
    - How ensure mashups work and are available
  - Trustworthiness
    - Phishing and spam in its many forms
    - Identity theft
    - Privacy
    - How to enforce security
  - Federation
    - How to identify users and authenticate and authorize
    - How to scale mashups

Middleware

- Widely used and popular term
- Fuzzy term
- One definition
  - "A set of service elements above the operating system and the communications stack"
- Second definition
  - "Software that provides a programming model above the basic building blocks of processes and message passing" (Colouris, Dollimore, Kindberg, 2001)

Why Middleware?

- Application development is complex and time-consuming
  - Should every developer code their own protocols for directories, transactions, ..?
  - How to cope with heterogeneous environments?
    - Networks, operating systems, hardware, programming languages
- Middleware is needed
  - To cut down development time
    - Rapid application development
  - Simplify the development of applications
  - Support heterogeneous environments and mask differences in OS/languages/hardware

Middleware cont.

- Middleware services include
  - directory, trading, brokering
  - remote invocation (RPC) facilities
  - transactions
  - persistent repositories
  - location and failure transparency
  - messaging
  - Security
- Network stack (transport and below) is not part of middleware

Transparencies

- Location transparency
  - RPC and RMI used without knowledge of the location of the invoked procedure / object
- transport protocol transparency
  - RPC may be implemented using any transport protocol
- transparency of OS and hardware
  - RPC/RMI uses external data representation
  - Presentation is important
  - XML is becoming increasingly important
- transparency of programming languages
  - language independent definition of procedures: CORBA IDL

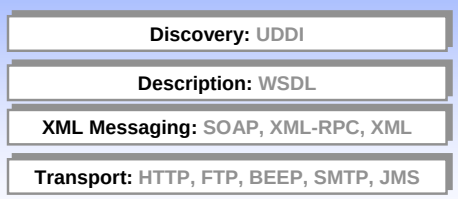
Web Service Architecture

- Motivation
  - Machine readable content on the Web
  - Programming API for the Web
  - Access independent of the computing environment
- The three major roles in web services
  - Service provider
    - Provider of the WS
  - Service Requestor
    - Any consumer / client
  - Service Registry
    - logically centralized directory of services
- A protocol stack is needed to support these roles

Web Services Protocol Stack

- Message Transport
  - Responsible for transporting messages
  - HTTP, BEEP
- XML Messaging
  - Responsible for encoding messages in common XML format
  - XML-RPC, SOAP
- Service Description
  - Responsible for describing an interface to a specific web service
  - WSDL
- Service discovery
  - Responsible for service discovery and search
  - UDDI

WS Protocol Stack



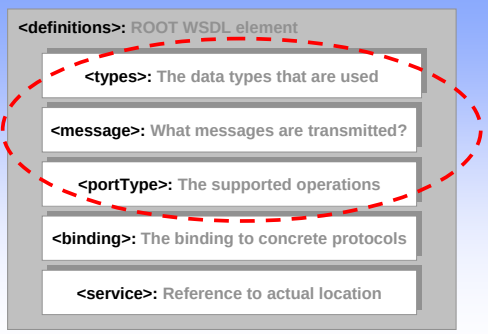
Standardization

- W3C Web Services Activity
  - XML Protocol Working Group
    - SOAP
  - Web Services Description Working Group
    - WSDL
  - Web Services Addressing Working Group
  - Web Services Choreography Working Group
- OASIS
  - Organization for the Advancement of Structured Information Standards
  - E-business standards, UDDI
- WS-I (Web Service Interoperability Org.)
  - Binding profiles,..

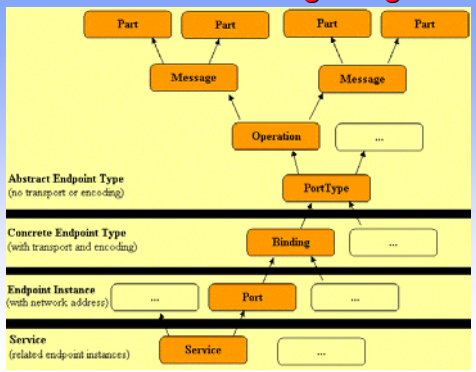
What is WSDL?

- WSDL: Web Service Description Language
- An XML language used to describe and locate web services
  - location of web service
  - methods that are available
  - data type information and XML messages
- Commonly used to describe SOAP-based services
- W3C standard (work in progress)
  - Initial input: WSDL 1.1 as W3C Note
  - Current version 2.0 (Candidate Recommendation)
  - Some differences between 1.1 and 2.0
- WSDL 1.1 in WS-I Basic Profile 1.0 and 1.1.

WSDL Overview



Putting it together

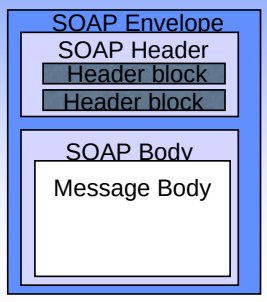


Source: http://msdn.microsoft.com/

What is SOAP?

- Fundamentally stateless one-way message exchange paradigm
  - More complex interactions may be implemented
- Exchange of structured and typed information
  - Between peers in decentralized fashion
  - Using different mediums: HTTP, Email, ..
- Request-reply and one-way communication are supported
- Note that XML infoset is an abstract specification
  - On-the-wire representation does not have to be XML 1.0!
- SOAP 1.2 "HTTP Subset". SOAP as HTTP extension
- Specifications
  - SOAP Version 1.2 Part 0: Primer
  - SOAP Version 1.2 Part 1: Messaging Framework
  - SOAP Version 1.2 Part 2: Adjuncts
  - SOAP Version 1.2 Specification Assertions and Test Collection

SOAP Message Structure



Optional header contains blocks of information regarding how to process the message:

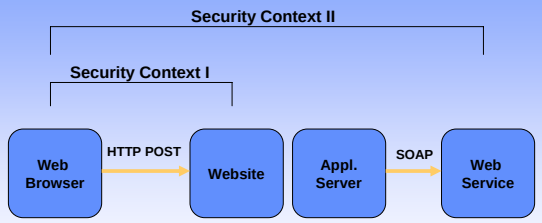
- Routing and delivery settings
- Authentication/authorization assertions
- Transaction contexts

Body is a **mandatory** element and contains the actual message to be delivered and processed (and fault information)

What is UDDI?

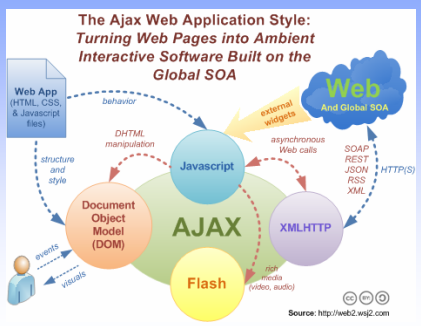
- Universal Description Discovery and Integration
- A mechanism for clients to dynamically find Web services
- White pages, yellow pages (categories), green pages (technical information)
- Developed on industry standards
  - Applies equally to XML and non-XML web services
- Implementation
  - Public web service registry and development resources
  - SOAP-based programming protocol for registering and discovering Web services
    - XML schema for SOAP messages
    - a description of the API
- UDDI does not directly specify how pricing, deadlines, etc. are handled/matched
  - Advanced discovery via portals and marketplaces

WS Security Contexts



Main Point: We need security within AND between security contexts!

AJAX



REST

- REST (Representational State Transfer) (Roy Fielding, PhD thesis)
  - Architectural style of networked systems
  - Applications transfer state with each resource representation
    - Representations of the data are transmitted
  - State is a property of a resource
- Resources
  - Any addressable entity
  - Web site, HTML page, XML document, ..
- URLs Identify Resources
  - Every resource uniquely identifiable by a URI

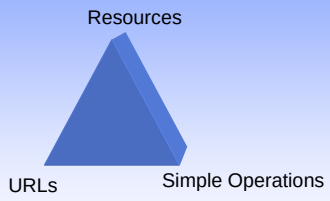
REST II

- Uses standards
  - Addressing and naming: URI
  - Generic resource interface: HTTP GET, POST, PUT, DELETE
  - Resource representations: HTML, XML, GIF,..
  - Media types: MIME
- Loose coupling
- Stateless transactions
- Self-descriptive messages
- **Hypermedia is the engine of application state**
  - **Just resources and URIs**

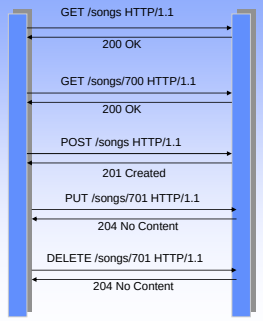
REST Goals

- Scalability of component interactions
- Generality of interfaces
- Independent deployment of components
- Intermediaries to reduce latency, improve security, and encapsulate legacy systems

REST Design Pattern



RESTful Example



Characteristics

- Client-Server pull style interactions
- Stateless, minimal stored context on a server
- Uniform interface: GET POST PUT DELETE
- Named resources (URIs)
- Resource caching (HTTP header)
- Comparison to SOAP
  - For REST all decisions are made upon the URL and HTTP method
  - SOAP uses header and possibly content as well
  - Hypothesis: REST+XML is applicable for all SOAP/WSDL?
    - SOAP offers headers, encoding, extensions

Examples

Amazon Web Services

- Web Services = APIs + Business Models
- Data as a service
  - E-Commerce service (product data)
  - Historical pricing
- Infrastructure as a service
  - Simple Queue Service, Simple Storage Service, Elastic Compute Cloud
- Search as a service
  - Alexa
- People as a service
  - Amazon Mechanical Turk

Amazon Web Services II

- Cross platform support: libraries in Python, Java, ..
- REST and SOAP, XML, XSLT
- Pay as you go
- Simple Queue Service
  - Made for communicating to your various Web services
  - Message storage
- S3 Simple Storage System
  - Not a file system
  - You store your data in buckets
  - Bucket names are globally unique (4KB long, max chunk 5GB)
    Files can be accessed over HTTP or BitTorrent

Amazon Web Services III

- EC2: Elastic Compute Cloud
  - On-demand Linux cluster
  - Uses XEN virtualization technology
  - Instances are stored on S3
  - You get root privileges, can install software

More Web Services

- Google
  - Search API
  - Google Maps
  - Advertisement APIs
  - ..
- Yahoo!
  - Answers, Local, Mail, Maps, Photos, Search, Shopping, Travel, Utilities, ..

programmableweb.com

Papers

- Google
  - The Anatomy of a Large-Scale Hypertextual Web Search Engine
  - The Google File System
  - Web Search for a Planet: The Google Cluster Architecture
- Amazon
  - Amazon Recommendations: Item-to-Item Collaborative Filtering
  - Analyzing Customer Behaviour at Amazon.com
- Yahoo
  - Experience with Personalization of Yahoo!

Security and Trust

- We are going towards identity-based service access
  - A number of identities per host
  - Pseudonyms, privacy issues
  - Delegation and federation are needed
- Decentralization: the user has the freedom of choosing who manages identity and data
- Solutions for authentication
  - Web-based standard (top-down)
    - ID-FF
  - Web-based practice (bottom-up)
    - OpenID and oAuth
  - Web services
    - SAML 2.0

Liberty Alliance ID-FF

- Liberty Alliance Identity Federation Framework (ID-FF)
- Basic case: Web direction
  - Redirect to IDP for credentials, redirect back to service, verification with IDP
  - Uses SAML requests and assertions
- Mandatory features for an identity provider
  - Single sign on and federation
  - Single sign out
  - Federation termination
  - Affliliations
  - Dynamic proxying of Identity Providers
- Circle of trust implemented using
  - SAML assertions, requests, redirection, and validation

ID-FF specs

- Liberty ID-FF
  - Identity Federation Framework
  - A forerunner to the SAML 2.0 specification. All of the functionality in ID-FF has been incorporated into SAML 2.0
- Liberty ID-WSF
  - Identity Web Services Framework
  - Builds on WS-Security and SAML 2.0
- Liberty ID-SIS
  - Identity Services Interface Specifications
  - High-level web service interfaces that support particular use cases like data/profile, geolocation, contact book, and presence services.

Passport and Live ID

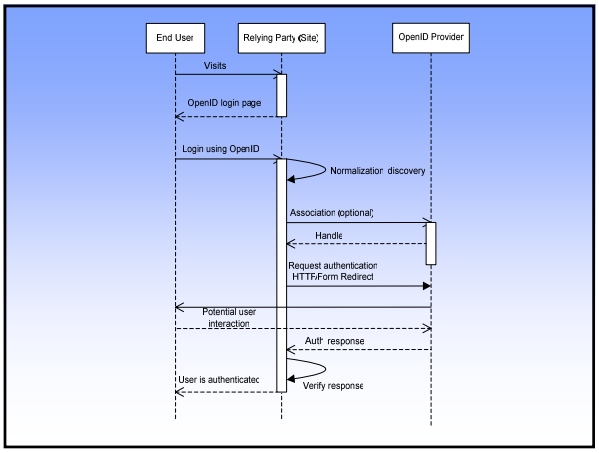
- Intended to solve two problems
  - to be an identity provider to MSN
  - identity provider for the Internet
- First goal
  - over 250 million active Passport accounts and
  - 1 billion authentications per day
- Second goal
  - What is the role of the identity provider in transactions?
  - Passport no longer stores personal information other than username/password credentials
- Authentication service for sites
- Proprietary technology
- Roadmap: towards identity card (CardSpace)
  - Interface for identity based authentication and authorization
  - Identity cards that people can choose (Identity Metasystem)
  - Integration with Web sites
  - Consistent user interface
- Windows Live ID
  - Unified login service for Microsoft sites such as Hotmail, MSNBC, MSN, ..
  - Used also for ad targeting with adCenter
  - Has been opened for Web site developers (August, 2007)

OpenID

- OpenID is a decentralized sign-on system for the Web
  - Not a real single sign-on solution, does not support authorization
- Instead of usernames and passwords, users need to have an account with some identity provider
- The user has the choice of selecting a suitable identity provider
- Support: AOL, Orange, FireFox, Microsoft planning support in Vista, LiveJournal, Wikitravel, Zooomr, Ma.gnolia
- Estimated 120 million OpenIDs on the Internet
- OpenID 2.0 supports discovery
  - Yadis provides a mechanism for determining the services that are available with a given identifier
- Identity aggregation: ClaimID
  - Claim Web resources under your OpenID (must have write permission)

OpenID URL

- There are two ways to obtain an OpenID-enabled URL that can be used to login on all OpenID-enabled websites.
  - To use an existing URL that one's own control (such as one's blog or home page), and if one knows how to edit HTML, one can insert the appropriate OpenID tags in the HTML code following instructions at the OpenID specification.
  - The second option is to register an OpenID identifier with an identity provider. They offer the ability to register a URL (typically a third-level domain) that will automatically be configured with OpenID authentication service.

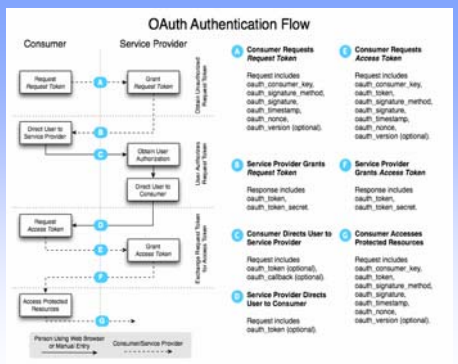


oAuth

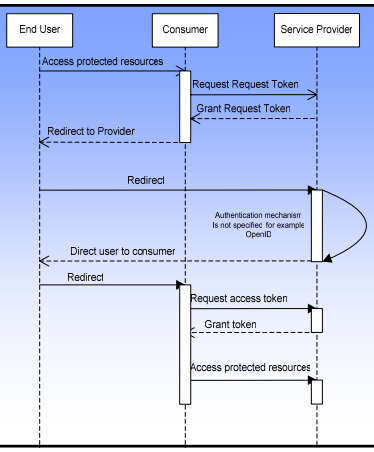
- oAuth is an open protocol to allow clients to access protected data
- Intended for desktop and web applications
- Example: a printing service printer.example.com, oAuth provides mechanisms for the printer to access user photos on photos.org without requiring users to provide credentials to printer.example.com.
- A solution for publish and interact with protected data
- Does not require a specific user interface or pattern, nor does it specify how service providers authenticate users
  - Can be used with OpenID
- Attempt to collect best practices from existing protocols
  - BBAuth (Yahoo), FacebookAuth, FlickrAuth, AuthSub (Google), OpenAuth (AOL) ..
- Contributors from many Web companies: Google, Flickr, Ma.gnolia, sixapart, Jaiku
- oAuth 1.0 Draft 3 was released September 28, 2007
- More information: http://oauth.net

Authentication with oAuth

- Entities: User, Consumer (accessing data), Service Provider (keeps the data)
- Tokens:
  - Request token: used by the consumer to ask the user to authorize access
  - Access token: used by the consumer to access the protected resources on behalf of the user
- OAuth Authentication is done in three steps:
  - The Consumer obtains an unauthorized Request Token.
  - The User authorizes the Request Token.
  - The Consumer exchanges the Request Token for an Access Token.



Source: http://oauth.googlecode.com/svn/spec/branches/1.0/drafts/3/spec.html



Summary

- Web services address easy and flexible service provision on top of the basic networking stack
  - High-level programming API for the Web
  - XML is becoming the presentation format for the Internet
  - Standards-based to ensure interoperability
- A middleware stack is needed
  - WSDL, SOAP, UDDI, messaging protocols
  - REST+XML as a lightweight alternative
    - Many services support both SOAP and REST
- We are going towards identity-based authentication and authorization
  - Centralized vs. decentralized

Thank You!