



Week 8

WebServices, RPC, ReST, Authentication, Authorization

Mastering Cloud Computing
Coleman Kane

(based on material by Paul Talaga)

Web Services

Service Oriented Architecture (SOA)

Standardized description of services

- WSDL
- JSON-WSP description
- OMG IDL

Service discovery

Revision management

Pre-request validation

Response data structure management

Web Services (resources)

WSDL / SOAP

- <https://www.w3.org/TR/wsdl>
- https://en.wikipedia.org/wiki/Web_Services_Description_Language

JSON-WSP & JSON-RPC

- <https://en.wikipedia.org/wiki/JSON-WSP>
- <http://www.simple-is-better.org/json-rpc/jsonrpc20.html>
- <https://wiki.python.org/moin/WebServices>

Many options for wrapping existing UI web-based applications into standardized service frameworks

ReST & Endpoints

ReST - **R**epresentational **S**tate **T**ransfer

RSDL - <https://en.wikipedia.org/wiki/RSDL>

Design approach to expose intuitive machine-interface API to the Internet for a service-offering application

Typically embed functional endpoints in URL path elements, and uses HTTP commands: GET, PUT, POST, PATCH, DELETE, HEAD, OPTIONS

RPC - Remote Procedure Call

A mechanism providing remote services, and hiding specifics of those services behind a library function-call interface

Simplifies application design

Fosters reusability, interoperability

RPC - Remote Procedure Call

In a typical application, execution works as follows:

1. Main code executes forward, encounters a function call
2. Main-local data passed as arguments to the call is organized into a data structure local to the new function
3. Execution transfers to the new function, proceeding linearly
4. Upon completing the function, return data local to the function is organized into a data structure local to the main function
5. Execution returns to the point after function call in the main code

RPC - Remote Procedure Call

In an RPC environment, execution works as follows:

1. Main code executes forward, encounters a function call
2. Main-local data passed as arguments to the call is organized into a data structure local to the new function
3. Execution transfers to the new function
4. Argument data is “marshalled” into a payload to send to RPC API used by the target web service, and is sent to the intended endpoint
5. Current program blocks, waiting for a response from the server
6. When a response is received from the web service, it may contain a data payload that consists of encoded result data from the remote procedure
7. This payload is unmarshalled, so that the data is now available as a local data structure in the new function
8. The return data is organized into a data structure local to the main function
9. Execution returns to the point after function call in the main code

Resource Access Control

Resource access control becomes even more important in an Internet-connected web service.

Access can be managed using the concepts of Authentication (AuthN) and Authorization (AuthZ)

Some parts of the data or application features may need to be public, or “less controlled”, while other parts may need to be “more controlled”.

Ex:

Browsing Amazon catalog versus ordering items online

Authentication (AuthN)

Authentication is the mechanism of identifying a particular user of a system

When you log into a web application using your username/password, you are authenticating yourself

Authentication management also known as **Identity Management**

Authorization (AuthZ)

Authorization is the system of deciding and managing what resource access a particular user should have

Once authenticated, you may have different rights in a system than other users

Under UNIX/Linux, this is commonly addressed using “file modes”, RBAC, Groups, etc.

Also known as **Resource Access Control**

AuthN/AuthZ Examples

Often, it is common to centralize Identity and Access control in a single system, and applications all connect to that system to validate access

- Microsoft Active Directory
- Apple OpenDirectory
- Apache Shiro - <https://shiro.apache.org/>
- Cork - <http://cork.firelet.net/>