



SERVICE-ORIENTED ARCHITECTURE

DR. MANJUNATH V HEGDE AND DR. VIDYA RAO

L17-L18



- SOA, or service-oriented architecture, defines a way to make software components reusable via service interfaces. These interfaces utilize common communication standards in such a way that they can be rapidly incorporated into new applications without having to perform deep integration each time.

SERVICES AND SERVICE-ORIENTED ARCHITECTURE

- SOA is about **how to design a software system that makes use of services of new or legacy applications through their published or discoverable interfaces.**
- SOA **aims** to make **service interoperability extensible and effective.**
- World Wide Web Consortium (W3C) defines SOA as distributed systems with the following properties:
 - **Logical view:** The SOA is an abstracted, logical view of actual programs, databases, and business processes,.
 - **Message orientation:** The internal structure of providers and requesters includes the implementation language, process structure, and even database structure.
 - **Description orientation:** A service is described by machine-executable metadata.
 - Granularity Services.
 - Network orientation Services
 - Platform-neutral Messages

REST AND SYSTEMS OF SYSTEMS

- REST is a software architecture style for distributed systems, particularly distributed hypermedia systems, such as the World Wide Web.
- Popular among: Google, Amazon, Yahoo, Facebook and Twitter (due to simplicity)
- The REST architectural style is based on four principles:
 1. Resource Identification through URIs
 2. Uniform, Constrained Interface
 3. Self-Descriptive Message
 4. Stateless Interactions

RESTFUL WEB SERVICE

RESTful web service is simple, lightweight in nature, and integrates with HTTP.

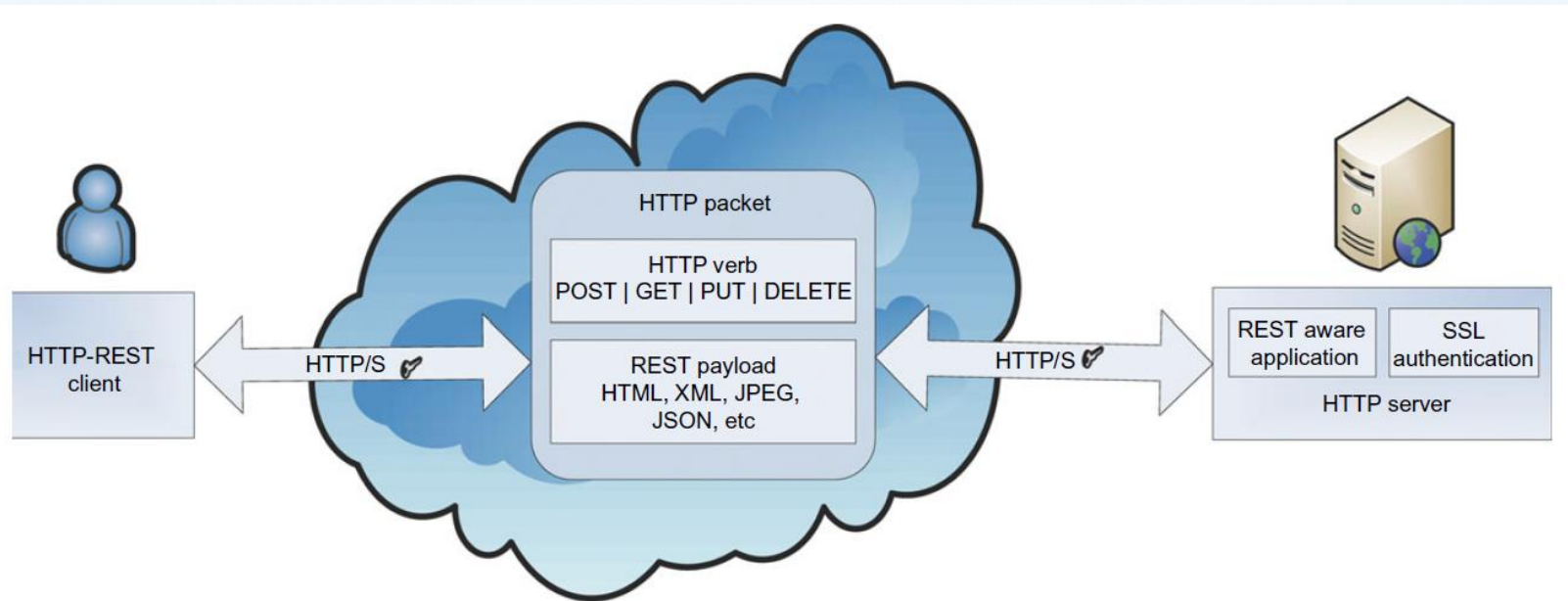


FIGURE 5.1

A simple REST interaction between user and server in HTTP specification.

(Courtesy of Thomas Fielding [2])

RESTFUL WEB SERVICE (CONTD..)

Restlet(Java Framework), implements REST architectural elements such as **resources, representation, connector, and media type.**

Table 5.1 REST Architectural Elements (Adapted from [2])

REST Elements	Elements	Example
Data elements	Resource	The intended conceptual target of a hypertext reference
	Resource identifier	URL
	Representation	HTML document, JPEG image, XML, etc.
	Representation metadata	Media type, last-modified time
	Resource metadata	Source link, alternates, vary
	Control data	If-modified-since, cache-control
Connectors	Client	libwww, libwww-perl
	Server	libwww, Apache API, NSAPI
	Cache	Browser cache, Akamai cache network
	Resolver	Bind (DNS lookup library)
	Tunnel	SSL after HTTP CONNECT
Components	Origin server	Apache httpd, Microsoft IIS
	Gateway	Squid, CGI, Reverse Proxy
	Proxy	CERN Proxy, Netscape Proxy, Gauntlet
	User agent	Netscape Navigator, Lynx, MOMspider

CASE STUDY: RESTFUL WEB SERVICE IN AMAZON S3 INTERFACE

- Amazon S3 (Simple Storage Service) is data storage for Internet applications.
- It provides simple web services to store and retrieves data from anywhere at any time via the web.
- S3 stores metadata in a container called “buckets”

Table 5.2 Sample REST Request-Response for Creating an S3 Bucket

REST Request	REST Response
PUT/[bucket-name] HTTP/1.0 Date: Wed, 15 Mar 2011 14:45:15 GMT Authorization:AWS [aws-access-key-id]: [header-signature] Host: s3.amazonaws.com	HTTP/1.1 200 OK x-amz-id-2: VjzdTviQorQtSjcgLshzCZSzN+7CnewvHA +6sNxR3VRcUPyO5fmSmo8bWnlS52qa x-amz-request-id: 91A8CC60F9FC49E7 Date: Wed, 15 Mar 2010 14:45:20 GMT Location: /[bucket-name] Content-Length: 0 Connection: keep-alive

CASE STUDY: AMAZON S3 (CONTD..)

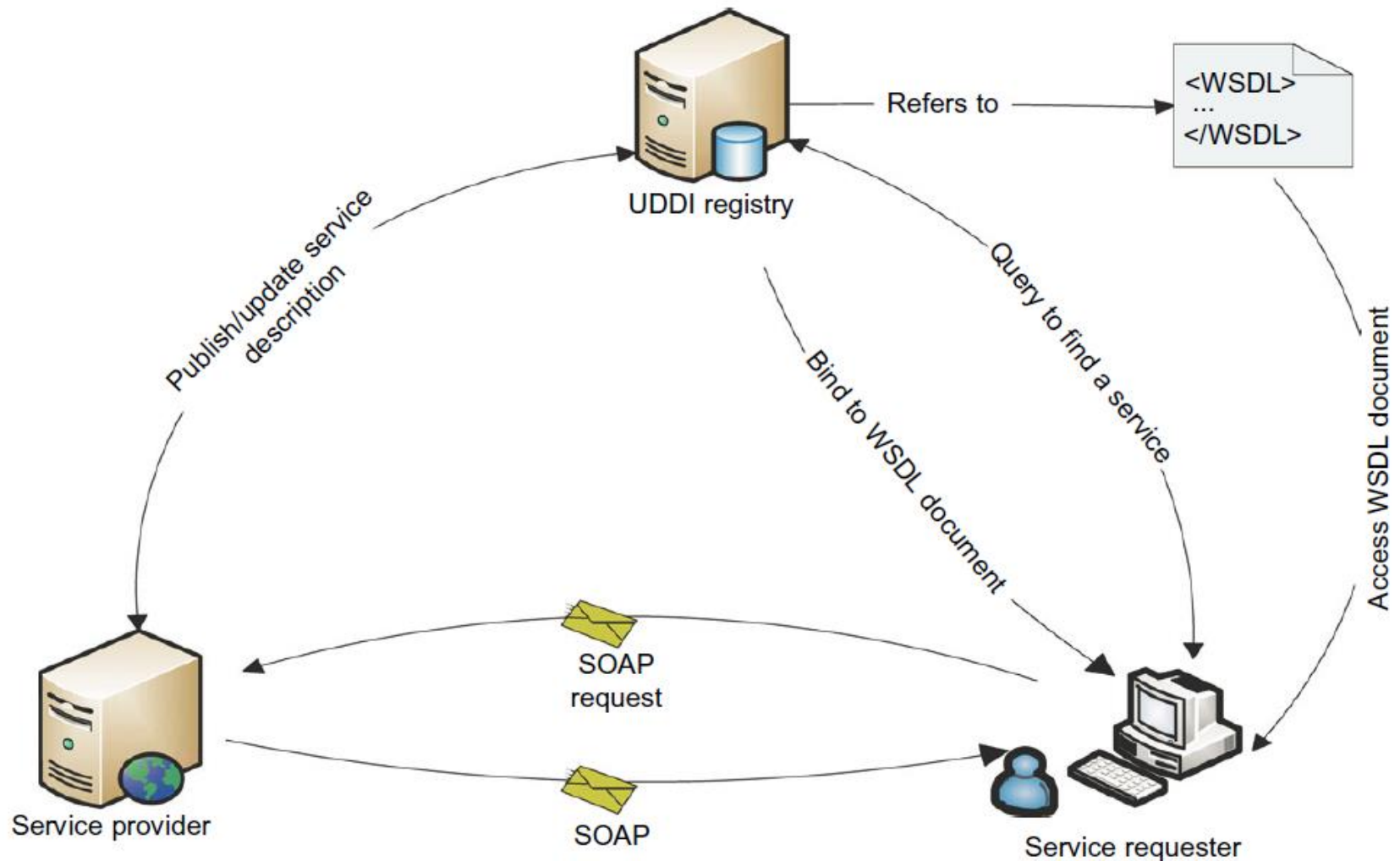
- “buckets” serves several purposes:
 - Organize the Amazon S3 namespace
 - Identify the account responsible for storage
 - Data transfer charges
- Amazon S3 provides three type of resources:
 - a list of user buckets
 - a particular bucket,
 - particular S3 object accessible through:
<https://s3.amazonaws.com/{name-of-bucket}/{name-of-object}>.

SERVICES AND WEB SERVICES

- The term “**web service**” is often referred to a self-contained, self-describing, modular application designed to be used and accessible by other software applications across the web.
- **W3C working group** [1] defines a web service as a software system designed to support interoperable machine-to-machine interaction over a network.
- Some of the technologies are:
 1. Simple Object Access Protocol (SOAP)
 2. Web Services Description Language (WSDL)
 3. Universal Description, Discovery, and Integration (UDDI)

SERVICES AND WEB SERVICES

(CONT..)



Simple object access protocol (SOAP)

- Simple object access protocol (SOAP) SOAP provides a standard packaging structure for transmission of XML documents over various internet protocols, such as SMTP, HTTP, and FTP.
- A soap message consists of
- A root element called envelope,
 - A header: a container that can be extended by intermediaries with additional application-level elements such as routing information, authentication,
 - Transaction management,
 - Message parsing instructions
 - Quality of service (qos) configurations,
 - A body element that carries the payload of the message.

WSDL and UDDI

- **Web services description language (WSDL)** WSDL describes the interface, a set of operations supported by a web service in a standard format. It standardizes the representation of Input and output parameters of its operations as well as the service's protocol binding, the way in which the messages will be transferred on the wire. Using WSDL enables disparate clients to Automatically understand how to interact with a web service.
- **Universal description, discovery, and integration (UDDI)** UDDI provides a global registry for advertising and discovery of web services, by searching for names, identifiers, categories, or The specification implemented by the web service.

SERVICES AND WEB SERVICES

(CONTD..)

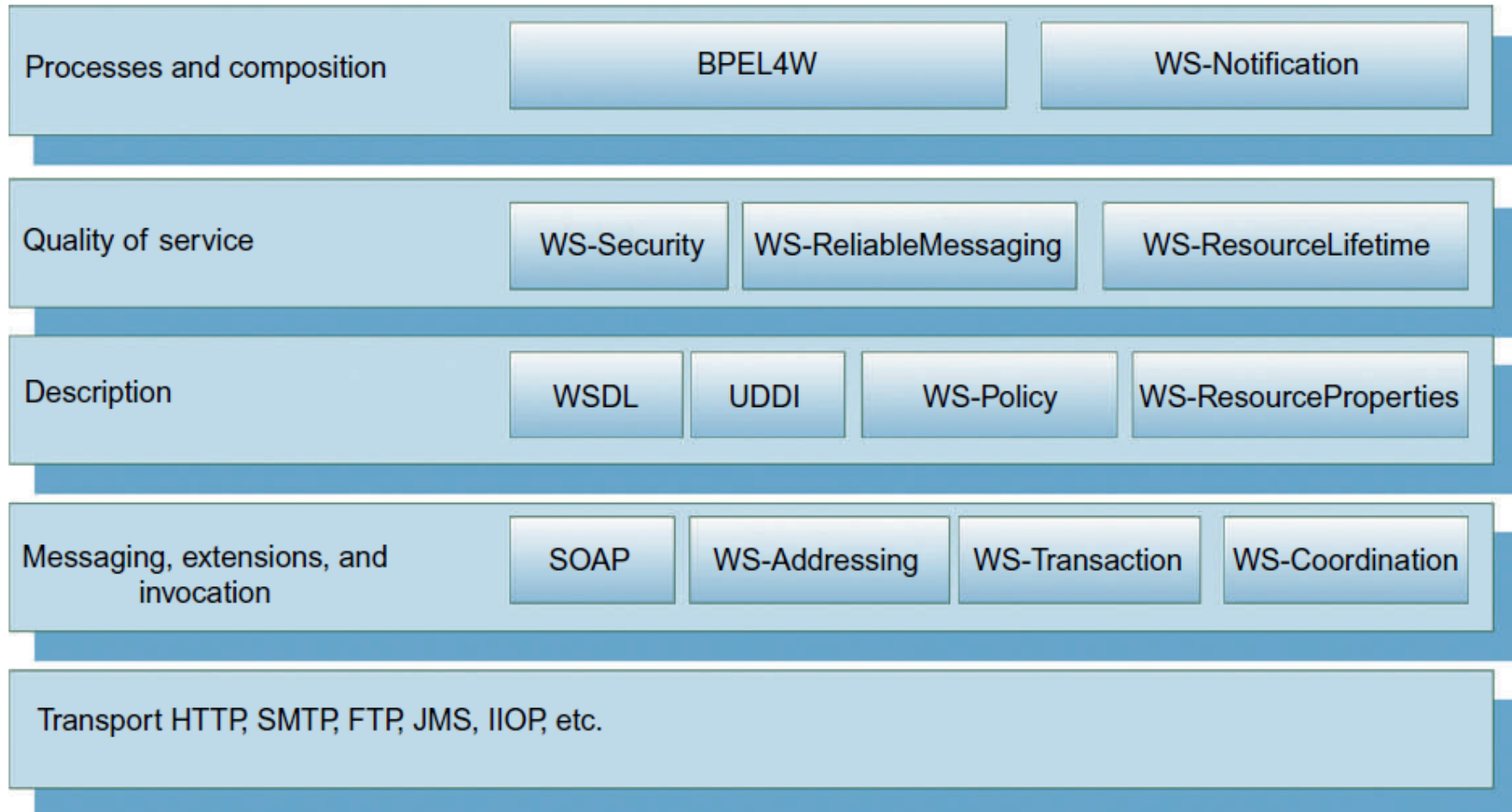


FIGURE 5.3

WS-I protocol stack and its related specifications.

SERVICES AND WEB SERVICES

(CONTD..)

SOAP message consists of an envelope containing a header and a body block

Table 5.3 Sample SOAP Request-Response for Creating an S3 Bucket

SOAP Request	SOAP Response
<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"> <soap:Body> <CreateBucket xmlns="http://doc.s3.amazonaws .com/2010-03-15"> <Bucket>SampleBucket</Bucket> <AWSAccessKeyId> 1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId> <Timestamp>2010-03-15T14:40:00.165Z </Timestamp> <Signature>luyz3d3P0aTou39dzbqaEXAMPLE =</Signature> </CreateBucket> </soap:Body> </soap:Envelope></pre>	<pre><soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap- envelope" soap:encodingStyle= "http://www.w3.org/2001/12/soap-encoding"> <soap:Body> <CreateBucket xmlns="http://doc.s3.amazonaws .com/2010-03-15"> <Bucket>SampleBucket</Bucket> <AWSAccessKeyId>1B9FVRAYCP1VJEXAMPLE= </AWSAccessKeyId> <Timestamp>2010-03-15T14:40:00.165Z </Timestamp> <Signature>luyz3d3P0aTou39dzbqaEXAMPLE =</Signature> </CreateBucket> </soap:Body> </soap:Envelope></pre>

SERVICES AND WEB SERVICES

(CONTD..)

WS-* Core SOAP Header Standards

Table 5.4 The 10 Areas Covered by the Core WS-* Specifications

WS-* Specification Area	Examples
1. Core Service Model	XML, WSDL, SOAP
2. Service Internet	WS-Addressing, WS-MessageDelivery, Reliable WSRM, Efficient MOTM
3. Notification	WS-Notification, WS-Eventing (Publish-Subscribe)
4. Workflow and Transactions	BPEL, WS-Choreography, WS-Coordination
5. Security	WS-Security, WS-Trust, WS-Federation, SAML, WS-SecureConversation
6. Service Discovery	UDDI, WS-Discovery
7. System Metadata and State	WSRF, WS-MetadataExchange, WS-Context
8. Management	WSDM, WS-Management, WS-Transfer
9. Policy and Agreements	WS-Policy, WS-Agreement
10. Portals and User Interfaces	WSRP (Remote Portlets)

ENTERPRISE MULTITIER ARCHITECTURE

```
graph TD; A[ENTERPRISE MULTITIER ARCHITECTURE] --> B[2 tier client/server]; A --> C[3 tier client/server]; C --> D[Presentation layer]; C --> E[Application layer]; C --> F[Storage layer];
```

2 tier
client/server

3 tier
client/server

Presentation layer
Application layer
Storage layer

3 TIER SYSTEM ARCHITECTURE

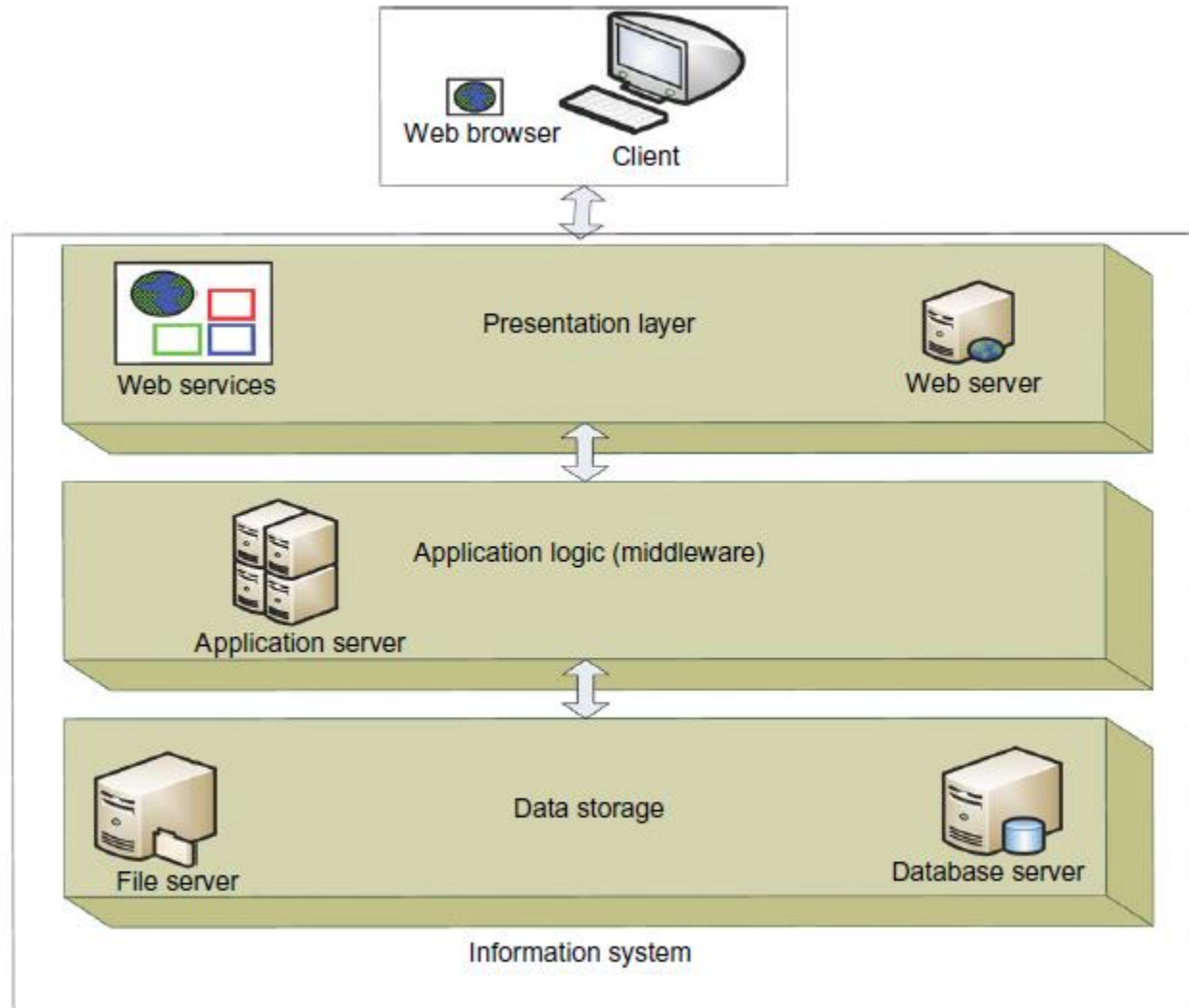


FIGURE 5.4

Three-tier system architecture.

GRID SERVICES AND OGSA

- OGSA is a service-oriented architecture that aims to define a common, standard, and open architecture for grid-based applications.
- “Open” refers to both the process to develop standards and the standards themselves.

Note:

Web Services Description Language

Grid Service Handle (GSH)

Grid Service Reference (GSR),

Open Grid Services Infrastructure (OGSI)

Web Services Resource Framework (WSRF)

GRID SERVICES AND OGSA (CONTD..)

OGSA is intended to:

- Facilitate use and management of resources across distributed, heterogeneous environments.
- Deliver seamless QoS.
- Publishes the interfaces in order to provide interoperability of diverse resources
- Exploit industry-standard integration technologies
- Develop standards that achieve interoperability
- Integrate, virtualize, and manage services and resources in a distributed, heterogeneous environment
- Deliver functionality as loosely coupled, interacting services aligned with industry-accepted web service standards

GRID SERVICES AND OGSA (CONTD..)

OGSA services fall into seven broad areas:

- Infrastructure Services
- Execution Management Services
- Data Management Services
- Resource Management Services
- Security Services
- Information Services
- Self-Management Services

GRID SERVICES AND OGSA (CONTD..)

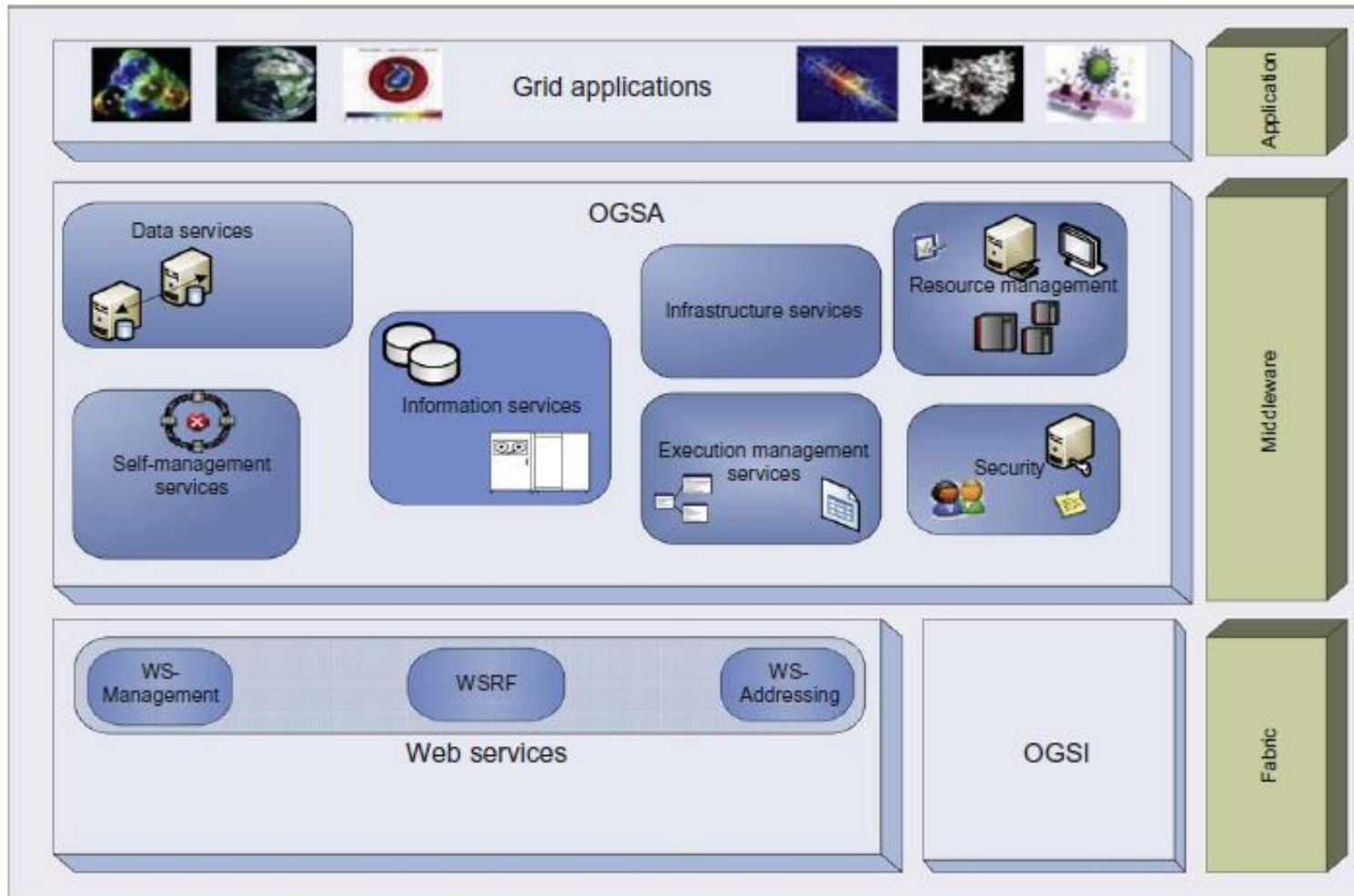


FIGURE 5.5

The OGSA architecture.

(Courtesy of Foster, et al. [24], <http://www.ogf.org/documents/GFD.80.pdf>.)



NEXT CLASS....

ABC