CPET 581 Cloud Computing: Technologies and Enterprise IT Strategies

Lecture 7
SOA and Distributed Computing

Spring 2015

A Specialty Course for Purdue University's M.S. in Technology Graduate Program: IT/Advanced Computer App Track

Paul I-Hai Lin, Professor

Dept. of Computer, Electrical and Information Technology

Purdue University Fort Wayne Campus

References

1. Chapter 5. Service-Oriented Architectures for Distributed Computing of the Text Book "Distributed and Cloud Computing," by Kai Hwang, Geoffrey C. Fox a,d Jack J. Dongarra, published by Mogan Kaufmman/ Elsevier Inc.
2. CPET 545 Service-Oriented Architecture & Enterprise Applications, Paul I-Hai Lin, Fall 2008, Purdue University, M.S. in Technology IT/Advanced Computer App Track, http://www.etcs.ipfw.edu/~lin/CPET545_SOA/cpet545-F08/cpet545home_F08.html

CPET 545 Course Description

CPET 545 Service-Oriented Architecture & Enterprise Applications, Class 3, Cr. 3

This course provides an intensive and comprehensive introduction to all essential aspects of the Service-Oriented Architecture (SOA) and enterprise applications including modeling, methodologies, processes, enabling technologies in support of enterprise computing applications. Topics include XML and Web services, enterprise application and data integration issues, service-oriented architecture, SOA analysis and design (SOAD), enterprise SOA realization, SOA-driven project management, and case studies. Student participation in presenting technical papers from the recent literature, class discussion, and team-based mid-term and final projects and presentation are expected.

Evolution of Enterprise Applications & Data Integration

- Enterprise Applications
  - Communications and networking infrastructure
  - 2-tier and N-tier applications
  - Distributing computing
  - E-business and E-commerce
  - Enterprise Resource Planning
  - Supply-Chain Management
  - Customer Relationship Management

Evolution of Enterprise Applications & Data Integration

- Enterprise Applications Integration
  - Web services for applications integration
  - Middleware, Web
- Data integration
  - Linking databases to applications and other databases
  - Database
  - Data warehouse
  - Data center
  - Data mining
- Knowledge Management & Business Intelligence

Services and Service-Oriented Architecture

- Technologies and System Architectures used resources
  - Collecting, Accessing, Processing, Integrating, Exchange, storage, Programming
  - Resources: web pages, business info, messages, pictures, video files, etc
- Distributed, Loosely coupled, and Heterogeneous information and computer systems
- Web services
  - Internet access
  - Web based
  - Multi-tier
  - Service producer, broker, consumer

Courtesy of Geoffrey Fox, 2010

Services and Service-Oriented Architecture (SOA)

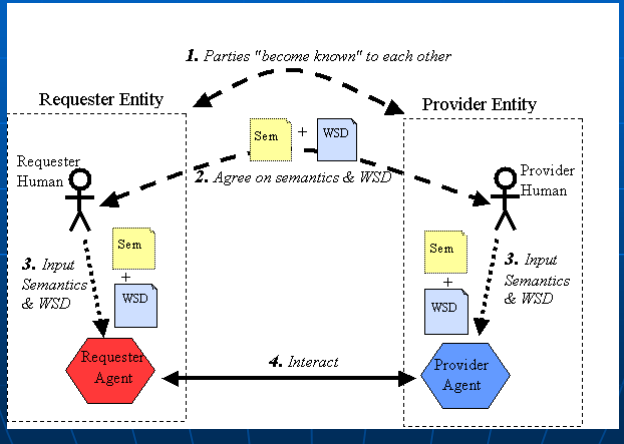
The W3C defines SOA as a form of distributed systems architecture characterized by the following properties, http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

- Logic View:
  - The SOA is an abstracted, logic view of actual programs, databases, business processes and so on, defined in terms of what it does, typically carrying out a business-level operation.
  - The service is formally defined in terms of the messages exchanged between provider agents and request agents.
- Message Orientation
- Description Orientation
  - Granularity
  - Network orientation
  - Platform-neutral

Service-Oriented Architecture

2004, http://www.w3.org/TR/ws-arch/

- Figure 1-1 The General Process of Engaging a Web Service



Services and Service-Oriented Architecture

- SOAP-based Web Services
- A definition, http://www.w3.org/2005/Talks/1115-hh-k-ecows/#(10)v

"A Web service is a software system designed to support **interoperable machine-to-machine interaction** over a network. It has an interface **described in a machine-processable format** (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description **using SOAP messages**, typically conveyed using HTTP with an XML serialization in conjunction with other **Web-related standards**."

Service-Oriented Architecture: The Evolution

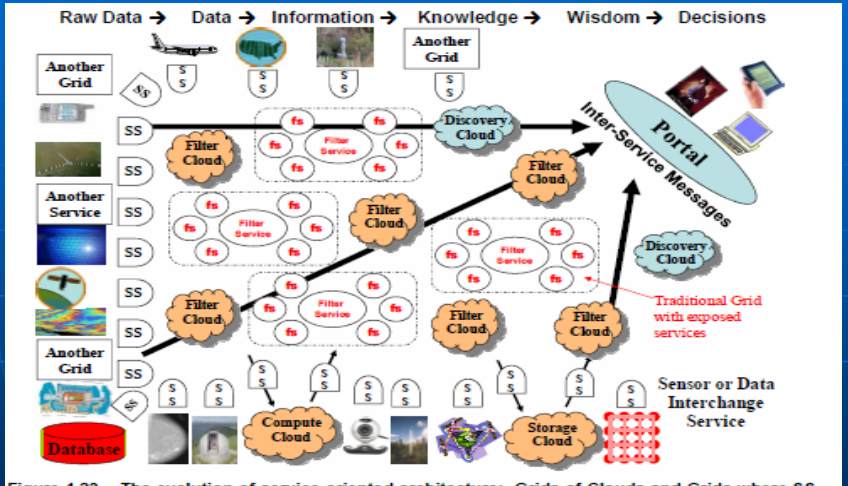


Figure 1.23 The evolution of service-oriented architecture: Grids of Clouds and Grids where SS refers to Sensor Service and fs to a filter or transforming service.

Courtesy of Geoffrey Fox, 2010

Service-Oriented Computing (Knowledge & Skills Inventory)

- Web Service Architecture, 2004, http://www.w3.org/TR/ws-arch/
- Programming languages, design, & development tools: C/C++, Java, C#, UML, NetBeans, Visual Studio, etc
- Computer systems, application servers, Web servers
- Network & communications: TCP/IP protocols, IP, network programming
- Web programming tools and protocols: HTTP, HTTPS, HTML, XML, Javascript, SAP.NET, Java Applet and Servlet, Perl, etc
- Databases: MySQL, IBM DB2, Oracle, MS SQL

Service-Oriented Computing

- Web Service Protocols:
  - XML RPC (Remote Procedure Call)
  - CORBA (Common Object Request Broker Architecture), http://www.omg.org/gettingstarted/history_of_corba.htm
  - SOAP 1.2: Simple Object Access Protocol, 2007, http://www.w3.org/TR/soap/
  - WSDL 1.1: Web Service Description Language, 2001, http://www.w3.org/TR/wsdl
  - UDDI OASIS, Universal Description, Discovery, Integration, http://uddi.xml.org/
  - REST

General Layered Architecture for Distributed Entities

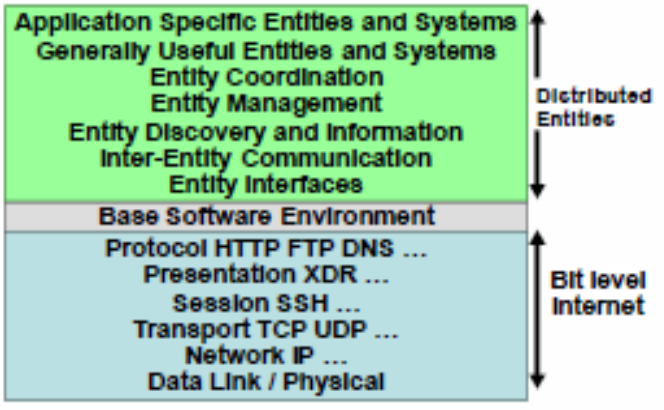


Fig. 1.21 General layered architecture for distributed entities

Layered Architecture for Web Services and Grids

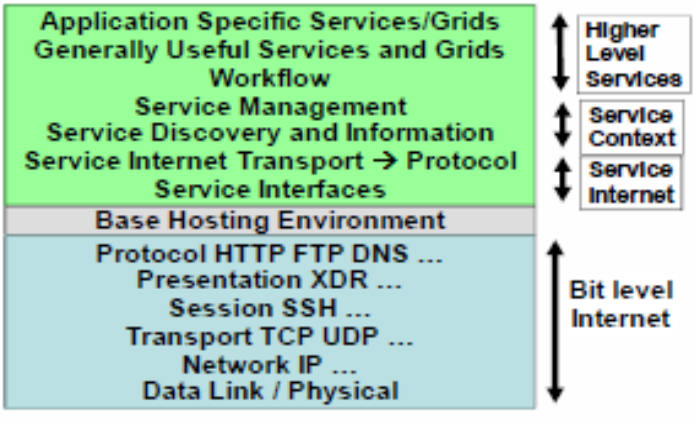


Figure 1.22 Layered architecture for web services and grids

REST (Representational State Transfer) and Systems of Systems

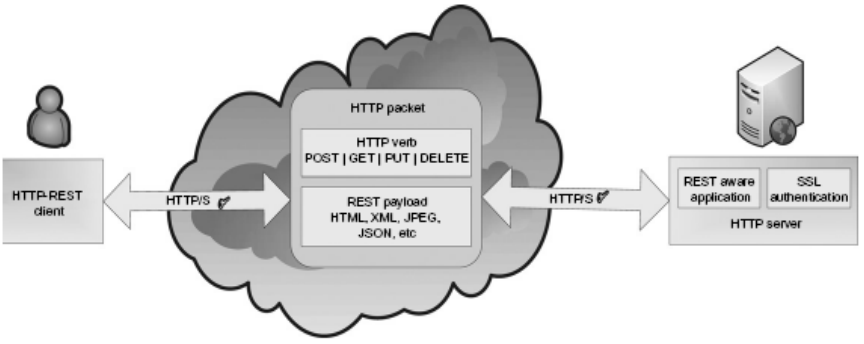


FIGURE 5.1

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

REST Web Services

- Web Service Protocols:
  - REST (Representational State Transfer)
  - Originally, a **software architectural style** for networked hypermedia applications in distributed systems
  - Primarily used to build HTTP-based web services that are lightweight, maintainable, and scalable.
  - A service based on REST is called RESTful services
- References
  - REST Architecture Style, http://www.w3.org/2001/sw/wiki/REST
  - Reconsiling Web Services and Rest Services, 2005, http://www.w3.org/2005/Talks/1115-hh-k-ecows/#(7)

RESTful Services

- Features of REST (Representational State Transfer)
  - Representations, Messages, URIs, Uniform interface, Stateless, Links between resources, Caching
- Four principles
  - Resource Identification through URIs
  - Uniform, Constrained Interface
  - Self-Descriptive Message
  - Stateless Interactions

RESTful Services

- Representation of a resource
  - JSON (JavaScript Object Notation) , http://json.org/
  - XML based
- JASON Example, http://www.drdobbs.com/web-development/restful-web-services-a-tutorial/240169069

{ “ID”: “1”, “Name”: “M Vaqqas”, “Email” :m.vaqqas@gmail.com”, “country”: “India”}

- XML Example
  - <Person> <ID> 1 </ID> <Name> M Vaqqas </Name> <Email> m.vaqqas@gmail.com </Email> <Country> India </Country> <Person>

Table 5.1 REST Architectural Elements

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

Example 5.1 RESTful Web service in Amazon S3 Interface (S3 - Simple Storage Service)

- S3 keep fundamental entities
  - "Object" which are named pieces of data accompanied by some metadata to be store in containers called "bucket"
  - Unique key
- S3 provides three types of resources accessible through https://s3.amazonaws.com/{name-of-bucket}/{name-ofobject}
  - A list of user buckets
  - A particular bucket
  - A particular S3 object
- HTTP standard operations: GET, HEAD, PUT, and DELETE to create, retrieve, or manipulate S3 resources
- S3 APIs

Example 5.1 RESTful Web service in Amazon S3 Interface (S3 - Simple Storage Service)

Sample REST Request-Response for Creating an S3 Bucket

REST Request	REST Response
PUT /[bucket-name] HTTP/1.0 Date: Wed, 15 Mar 2010-14:45:15 GMT Authorization: AWS [aws-access-key-id]: [header-signature] Host: s3.amazonaws.com	HTTP/1.1 200 OK x-amz-id-2: VjzdTviQorQtSjcgLshzCZSzN+7CnewvHA +6sNxR3VRcUPyO5fmSmo8bWnIS52qa x-amz-request-id: 91A8CC60F9FC49E7 Date: Wed, 15 Mar 2010 14:45:20 GMT Location: /[bucket-name] Content-Length: 0 Connection: keep-alive

SOAP: Simple Object Access Protocol
UDDI: Universal Description Discovery & Integration
WSDL: Web Service Description Language

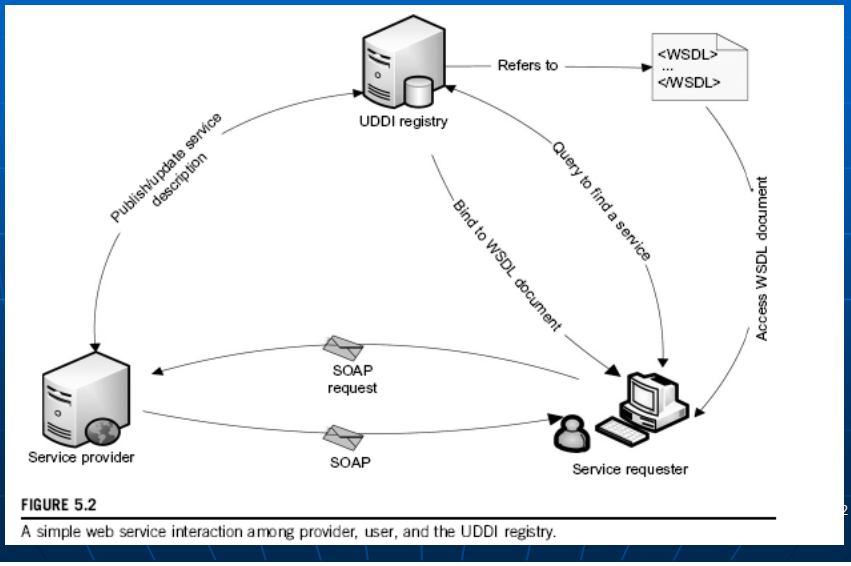


FIGURE 5.2

A simple web service interaction among provider, user, and the UDDI registry.

WS-I (Web Service – Interoperability) Protocol Stack and Its Related Specification

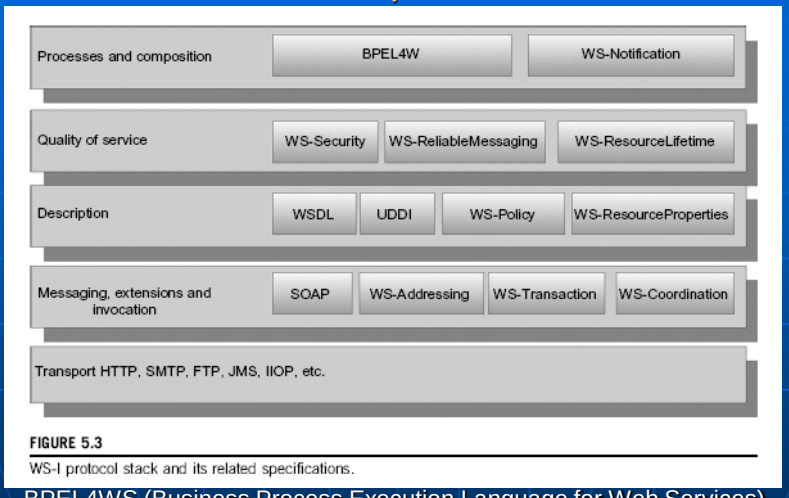


FIGURE 5.3
WS-I protocol stack and its related specifications.

- BPEL4WS (Business Process Execution Language for Web Services), XML-based
- WS – I, http://www.ws-i.org/
- JMS – Java Message Service, CORBA's IIOP (Internet Inter-ORB) protocol

Sample SOAP Request-Response for Creating an S3 Bucket

SOAP Request	SOAP Response
<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>	<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>

Ten 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 Multi-tier Architecture: from 3-tier to N tier

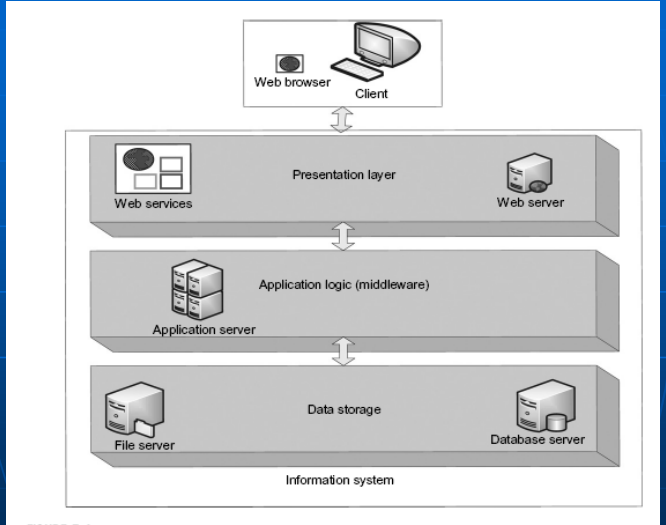


FIGURE 5.4
Three-ter system architecture.

OGSA: Open Grid Services Architecture - Globus
OGSI: Open Grid Services Infrastructure

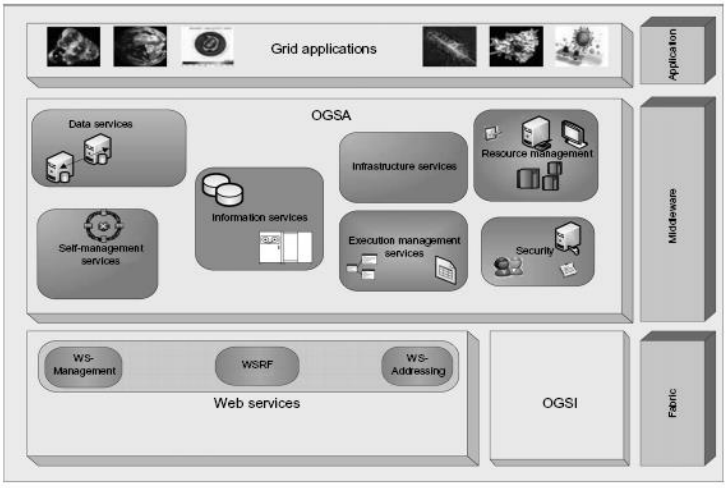


FIGURE 5.5
The OGSA architecture.

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

OGSA (Open Grid Services Architecture) Service Areas

- Service Areas
  - Infrastructure Services
  - Execution Management Services
  - Data Management Services
  - Resource Management Services
  - Security Services
  - Information Services
  - Self-Management Services
- WSRF (Web Service Framework) Related Specifications

WSRF (Web Services Resource Framework - OASIS) and Its Related Specifications

Specification			Description
WSRF Specifications	WS-ResourceProperties		Standardizes the definition of the resource properties, its association with the WS interface, and the messages defining the query and update capability against resource properties
	WS-ResourceLifetime		Provides standard mechanisms to manage the life cycle of WS-resources (e.g., setting termination time)
	WS-ServiceGroup		Standard expression of aggregating Web services and WS-Resources
	WS-Basefault		Provides a standard way of reporting faults
WSRF-Related Specifications	WS-Notification	WS-Base Notification	Proposes a standard way of expressing the basic roles involved in Web service publish and subscribe for notification message exchange
		WS-BrokeredNotification	Standardizes message exchanges involved in Web service publish and subscribe of a message broker
		WS-Topics	Defines a mechanism to organize and categorize items of interest for a subscription known as "topics"
	WS-Addressing		Transport-neutral mechanisms to address Web service and messages

Example 5.3 US DoD Net-Centric Services

- Global Information Grid (GIG), http://www.globalsecurity.org/intell/systems/gig.htm
- Dept. of Defense Global Information Grid Vision, 2007, *www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA484389*

Table 5.6 Core Global Information Grid Net-Centric Services

Service or Feature	Examples
Enterprise services management	Life-cycle management
Security; information assurance (IA)	Confidentiality, integrity, availability, reliability
Messaging	Publish-subscribe important
Discovery	Data and services
Mediation	Agents, brokering, transformation, aggregation
Collaboration	Synchronous and asynchronous
User assistance	Optimize Global Information Grid user experience
Storage	Retention, organization, and disposition of all forms of data
Application	Provisioning, operations and maintenance
Environmental control services	Policy

Example 5.4 Services and Standards used in CICC – A Chemical Informatics Grid

Service Name	Description
Workflow/Monitoring/ Management services	Uses Taverna from the UK e-Science Program/OMII or mashups written in scripting languages.
Authentication/ Authorization	Currently all services are openly available.
Registry and Discovery	Will inherit registry services from other grids.
Portal and portlets	Use a JSR 168-based portal.
File Services	No specialized service. URLs are used for naming files and simple remote download. Services developed for grids can be used for uploads.
NIH DTP Database Services	Access to the NIH Developmental Therapeutics Program (DTP)'s database of molecular screens against 60 cancer cell lines, a free service by Chembiogrid.
PubMed Search Service	Searchable online database of medical journal articles. CICC develops harvesting services of the abstracts combined with text analysis applications such as OSCAR3.
SPRESI Services	Clients/service proxies to the commercial SPRESI service (www.spresi.com/). This scientific database houses molecular and reaction data and references and patents.

Services and Standards used in CICC (Continued)

Varuna Database Service	Molecular structure and more detailed information (such as force fields).
VOTables Data Tables Web Service	CICC-developed Web service based on the National Virtual Observatory's VOTables XML format for tabular data.
Specific applications: BCI, OpenEye, Varuna, AutoGEFF	CICC inherits job management services from other grids (including one based on Apache Ant) for managing the execution of both commercial and in-house developed high-performance computing applications.
Condor and BirdBath	Examine the use of Condor and its SOAP interface (BirdBath) as a super-scheduler for Varuna applications on the TeraGrid.
ToxTree Service	Wraps an algorithm for estimating toxic hazards in a particular compound. Useful in combination with other clustering programs in a workflow.
OSCAR3 Service	Based on OSCAR3 by the WWMM group, performs text analysis on journal articles and other documents to extract (in XML) the chemistry-specific information. SMILES assigned to well-known compounds. Works with traditional database and clustering algorithms.
CDK Services	CICC has developed a number of simple services based on the Chemistry Development Kit (CDK). These include similarity calculations, molecular descriptor calculations, fingerprint generators, 2D image generators, and 3D coordinate molecular generators.
OpenBabel Service	Converts between various chemical formats (such as between InChI and SMILES).
InChIGoogle	For a given InChI (a string specification of a molecular structure), performs a Google search to return a page-ranked list of matches.
Key interfaces/ standards/software used	WSDL, SOAP (with Axis 1.x). CML, InChI, SMILES, Taverna SCUFl, JSR-168 JDBC Servlets, VOTables
Unused interfaces/software	WS-Security, JSDL, WSRF, BPEL, OGSA-DAI

5.2 Message-Oriented Middleware

- Enterprise Bus
  - Enterprise Service Bus (ESB), http://www.oracle.com/technetwork/articles/soa/ind-soa-esb-1967705.html
- Publish-Subscribe Model
- Queuing and Messaging Systems

Two Message Bus Implementation

- Message Styles: SOAP, REST, Java RMI
- MQSeries, WebSphereMQ

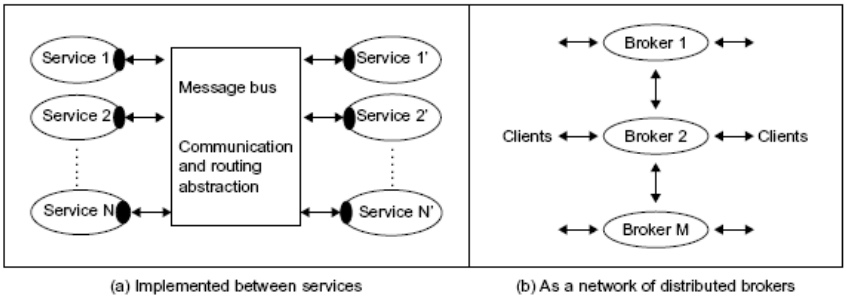


FIGURE 5.6
Two message bus implementations between services or using a broker network.

Queuing and Messaging Systems

- JMS (Java Message Service)
- AMQP (Advanced Message Queuing Protocol)
- WebSphereMQ
- MuleMQ
- Cloud-based
  - Amazon Simple Queue
  - Azure Queue

Table 5.8 Comparison of Messaging and Queuing Systems

System Features	Amazon Simple Queue	Azure Queue	ActiveMQ	MuleMQ	WebSphere MQ	Narada Brokering
AMQP compliant	No	No	No, uses OpenWire and Stomp	No	No	No
JMS compliant	No	No	Yes	Yes	Yes	Yes
Distributed broker	No	No	Yes	Yes	Yes	Yes
Delivery guarantees	Message retained in queue for four days	Message accessible for seven days	Based on journaling and JDBC drivers to databases	Disk store uses one file/channel, TTL purges messages	Exactly-once delivery supported	Guaranteed and exactly-once
Ordering guarantees	Best effort, once delivery, duplicate messages exist	No ordering, message returns more than once	Publisher order guarantee	Not clear	Publisher order guarantee	Publisher- or time-order by Network Time Protocol
Access model	SOAP, HTTP-based GET/POST	HTTP REST interfaces	Using JMS classes	JMS, Adm. API, and JNDI	Message Queue Interface, JMS	JMS, WS-Eventing
Max. message	8 KB	8 KB	N/A	N/A	N/A	N/A
Buffering	N/A	Yes	Yes	Yes	Yes	Yes
Time decoupled delivery	Up to four days; supports timeouts	Up to seven days	Yes	Yes	Yes	Yes
Security scheme	Based on HMAC-SHA1 signature Support for WS-Security 1.0	Access to queues by HMAC SHA256 signature	Authorization based on JAAS for authentication	Access control, authentication, SSL for communication	SSL, end-to-end application-level data security	SSL, end-to-end application-level data security, and ACLs
Support for Web services	SOAP-based interactions	REST interfaces	REST	REST	REST, SOAP interactions	WS-Eventing
Transports	HTTP/HTTPS, SSL	HTTP/HTTPS	TCP, UDP, SSL, HTTP/S, Multicast, in-VM, JXTA	Mule ESB supports TCP, UDP, RMI, SSL, SMTP, and FTP	TCP, UDP, Multicast, SSL, HTTP/S	TCP, Parallel TCP, UDP, Multicast, SSL, HTTP/S, IPsec
Subscription formats	Access is to individual queues	Access is to individual queues	JMS spec allows for SQL selectors; also access to individual queues	JMS spec allows for SQL selectors; also access to individual queues	JMS spec allows SQL selectors; access to individual queues	SQL selectors, regular expressions, <tag, value> pairs, XQuery and XPath

Example 5.5 Environmental Monitoring and Internet Conference using NaradaBrokering

- The GOAT project at Clemson University, http://www.clemson.edu/public/impacts/07summer/env/environmental_sensors_track_impact_of_development.html
  - Environmental sensor track impact of development
- The Program of Integrated Study for Coastal Environmental Sustainability (PISCES)
- The NaradaBrokering Project, Indiana University, http://www.naradabrokering.org/
- Anabas.com (Internet meeting software), http://www.anabas.com/

Example 5.6 QuakeSim Project for Earthquake Science

- http://quakesim.org

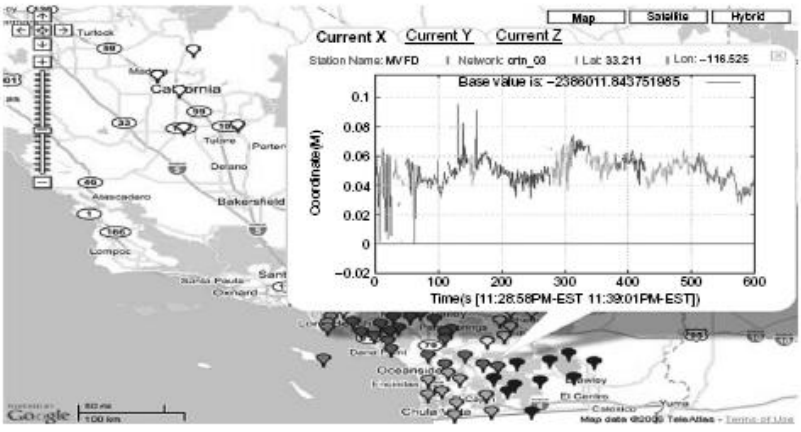


FIGURE 5.7

Display of GPS sensors managed by NaradaBrokering in Southern California; the map displays the time series produced by one of the GPS stations. (http://quakesim.jpl.nasa.gov/).

5.3 Portals and Science Gateways

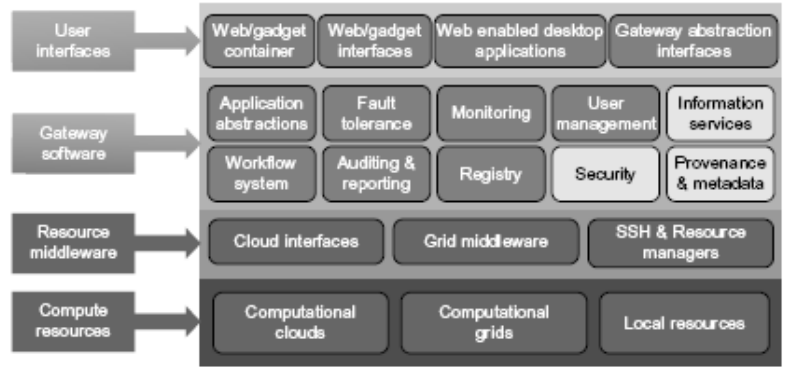


FIGURE 5.8

A gateway component software stack for scientific applications.

The HUBzero Architecture

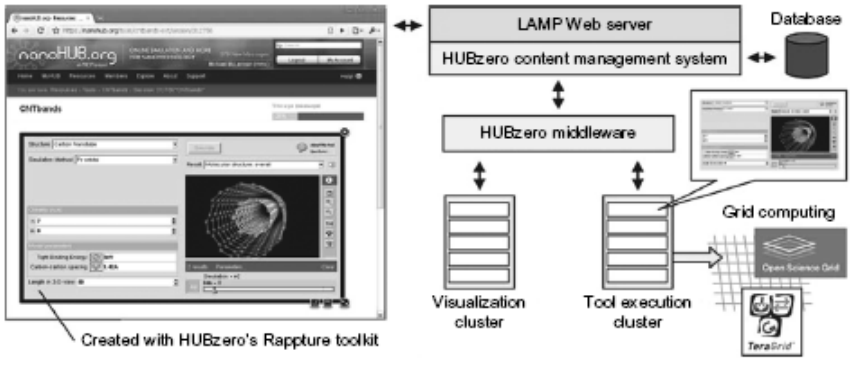


FIGURE 5.9

The HUBzero architecture and its major functional components.

5.4 Discovery, Registries, Metadata, and Databases

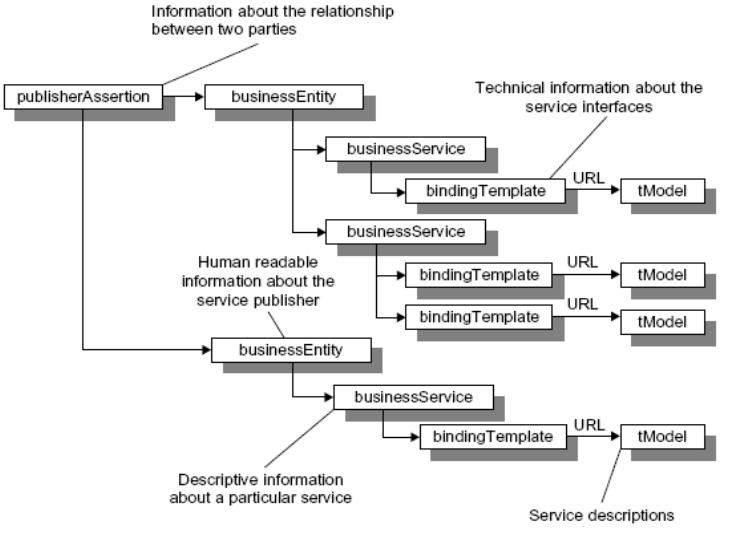


FIGURE 5.10

UDDI entities and their relationship.

41

Databases and Publish-Subscribe Model

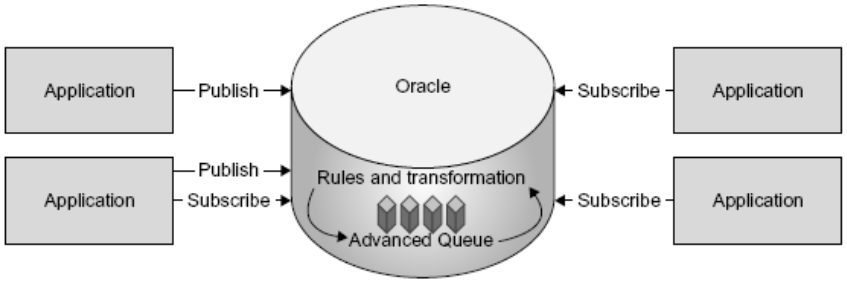


FIGURE 5.11

Oracle publish-subscribe model.

Semantic Web and Grid

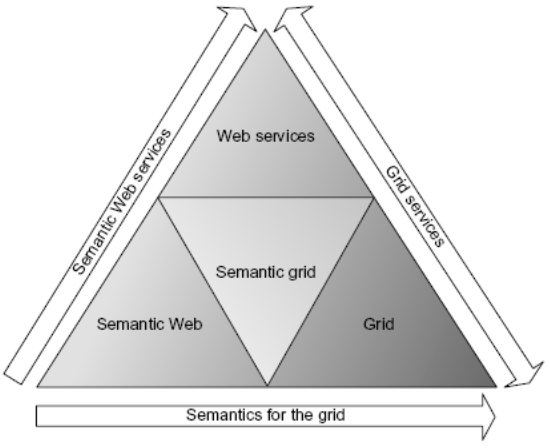


FIGURE 5.12

Semantic grid-related concepts and technologies.

(Courtesy of Goble and Roure, 16th European Conference on Artificial Intelligence (ECAI-2004), Valencia, Spain, 2004 [95])

Semantic Grid Architecture

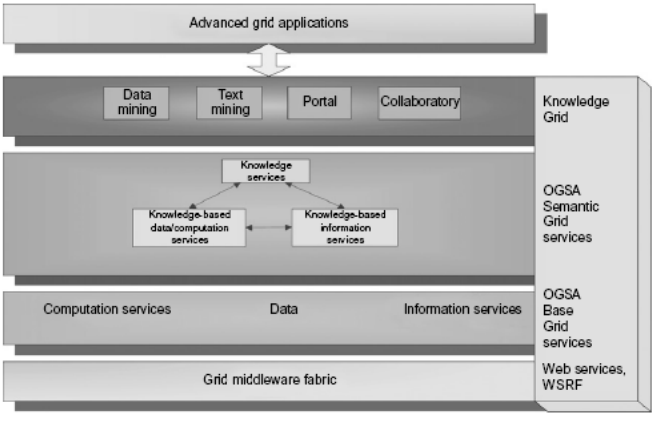


FIGURE 5.13

Semantic grid architecture.

(Courtesy of Goble and Roure, 16th European Conference on Artificial Intelligence (ECAI-2004), Valencia, Spain, 2004 [95])

5.5 Workflow in SOAs
The Grid of Grids of Services

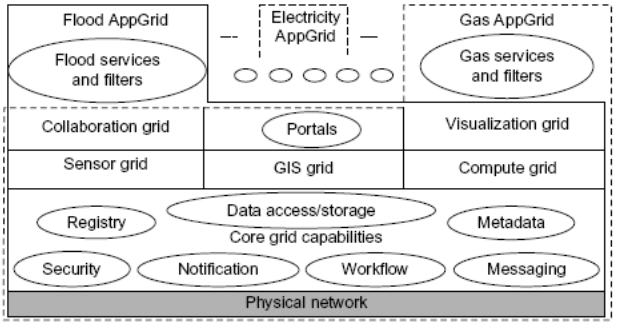


FIGURE 5.14

The concept of the grid of grids of services.

Basic Workflow Concepts

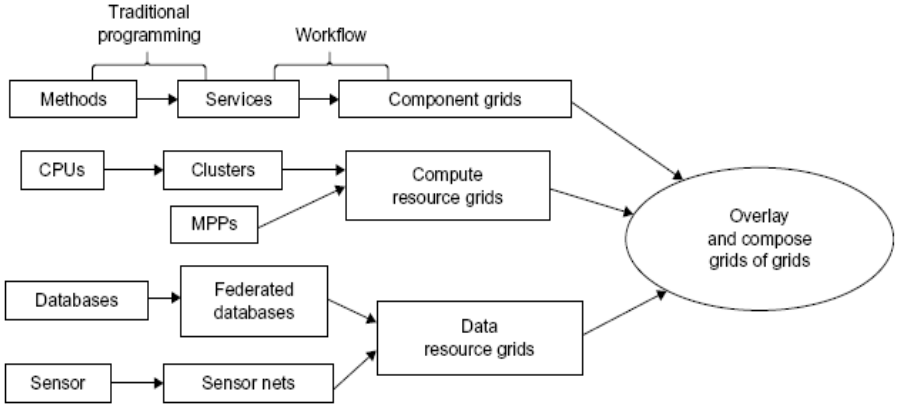


FIGURE 5.15

Hierarchical computing, data, and programming abstraction.

Table 5.9 Workflow Standards, Links, and Status

Standard	Link	Status
BPEL Business Process Execution Language for Web Services (OASIS) V 2.0	http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html; http://en.wikipedia.org/wiki/BPEL	April 2007
WS-CDL Web Service Choreography Description Language (W3C)	http://www.w3.org/TR/ws-cdl-10/	November 2005, not final
WSCI Web Service Choreography Interface V 1.0 (W3C)	http://www.w3.org/TR/wsci/	August 2002, note only
WSCL Web Services Conversation Language (W3C)	http://www.w3.org/TR/wscl10/	March 2002, note only
WSFL Web Services Flow Language	http://www.ibm.com/developerworks/webservices/library/ws-wsfl2/	Replaced by BPEL
XLANG Web Services for Business Process Design (Microsoft)	http://xml.coverpages.org/XLANG-C-200106.html	June 2001, replaced by BPEL
WS-CAF Web Services Composite Application Framework including **WS-CTX, WS-CF,** and **WS-TXM**	http://en.wikipedia.org/wiki/WS-CAF	Unfinished
WS-CTX Web Services Context (OASIS Web Services Composite Application Framework TC)	http://docs.oasis-open.org/ws-caf/ws-context/v1.0/OS/wsctx.html	April 2007

Table 5.9 Workflow Standards, Links, and Status

Standard	Link	Status
WS-Coordination Web Services Coordination (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wscoor/2006/06	February 2009
WS-AtomicTransaction Web Services Atomic Transaction (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wsat/2006/06	February 2009
WS-BusinessActivity Framework (BEA, IBM, Microsoft at OASIS)	http://docs.oasis-open.org/ws-tx/wsba/2006/06	February 2009
BPMN Business Process Modeling Notation (Object Management Group, OMG)	http://en.wikipedia.org/wiki/BPMN; http://www.bpmn.org/	Active
BPSS Business Process Specification Schema (OASIS)	http://www.ebxml.org/; http://www.ebxml.org/specs/ebBPSS.pdf	May 2001
BTP Business Transaction Protocol (OASIS)	http://www.oasis-open.org/committees/download.php/12449/business_transaction-btp-1.1-spec-cd-01.doc	Unfinished

A Typical (Load and Merge) Workflows

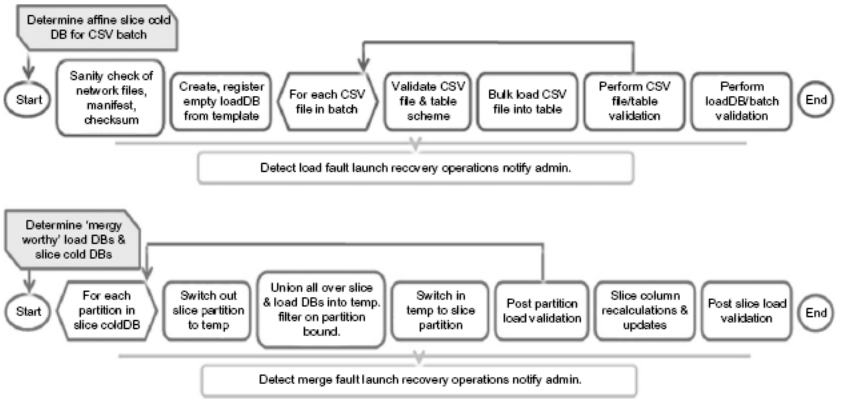


FIGURE 5.16

Two typical (Load and Merge) workflows from the Pan-STARRS astronomy data processing area.

(Courtesy of Barga, et al. [129])

Example 5.10 Pan-STARTS Workflow

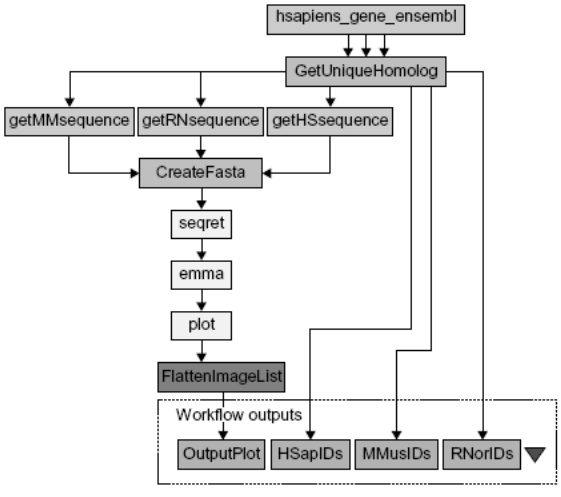


FIGURE 5.17

The workflow in the Taverna System showing.

(Courtesy of C. Goble, European Conf. on Research and Advanced Technology for Digital Libraries, 2008 [132])

A Workflow Graph

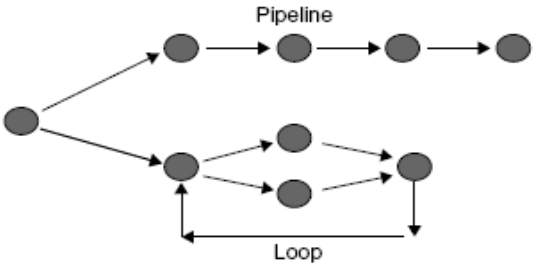


FIGURE 5.18

A workflow graph that includes subgraphs illustrating pipelines and loops.

Example 5.11 Swift Workflow System Architecture

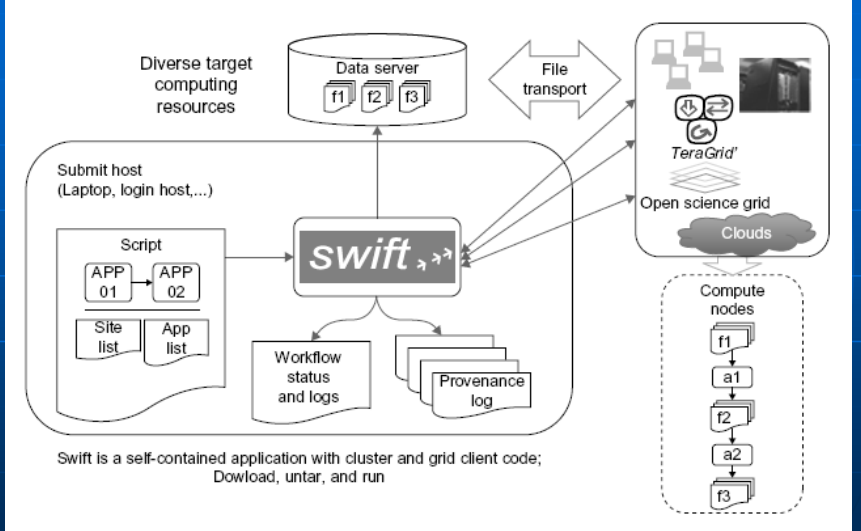


FIGURE 5.19

Swift workflow system architecture.

(Courtesy of www.ci.uchicago.edu/swift/ [138])

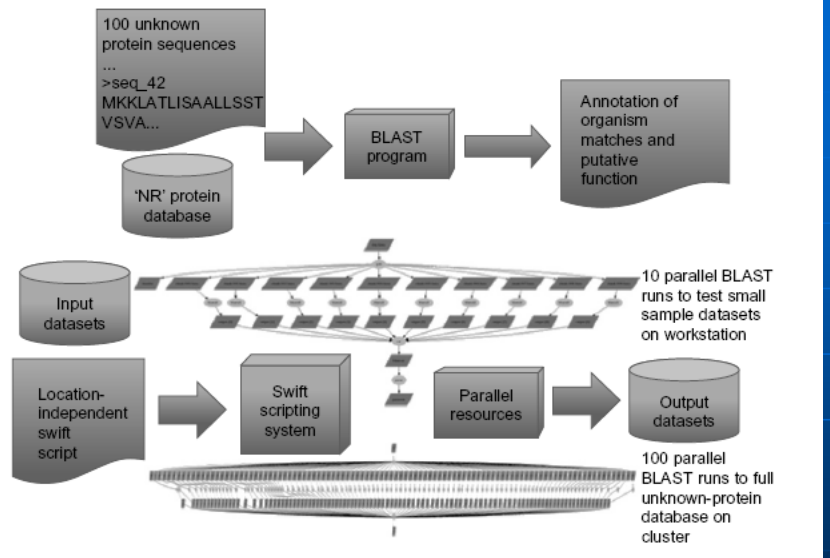


FIGURE 5.20

Application of the Swift to bioinformatics.

(Courtesy of Wikipedia, http://en.wikipedia.org/wiki/Many-task_computing, [154])

Cloud Computing Service Provider Priorities

- **Ensure confidentiality, integrity, and availability in a multi-tenant environment.**
- **Effectively meet the advertised SLA, while optimizing cloud resource utilization.**
- **Offer tenants capabilities for self-service, and achieve scaling through automation and simplification.**

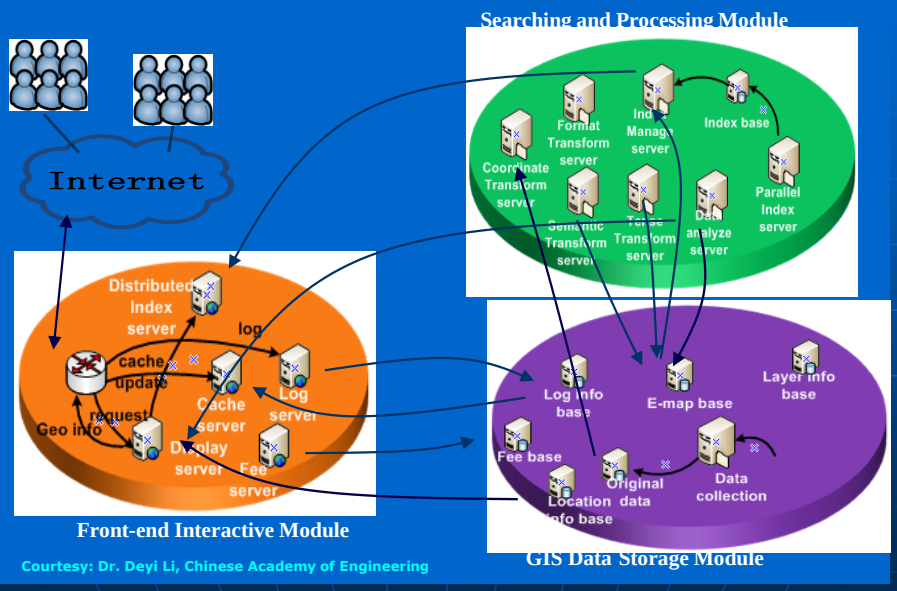
Guiding Principles in Using Clouds for Data-Intensive Applications (G. Fox, 2011)

- Clouds may not be suitable for everything but they are suitable for majority of data intensive applications
  - Solving PDEs on 100,000 cores needs classic MPI clusters
- Cost effectiveness, elasticity and quality programming model will drive use of clouds in many areas such as genomics
- Need to solve issues of
  - Security-privacy-trust for sensitive data
  - How to store data – "data parallel file systems" (HDFS), Object Stores, or classic HPC approach with shared file systems with Lustre etc.
- Programming model which is likely to be MapReduce based
  - Look at high level languages
  - Compare with databases (SciDB)
  - Must support iteration to do "real parallel computing"
  - Need Cloud- HPC Cluster Interoperability

SOA Security

- SOA security is essentially is the web services security
- Three core specifications
  - WS-Security, XML-Signature, XML-Encryption
  - WS*-Security offers a new technology for SOA security
- Single sign-on (SSO) is a form of centralized security mechanism
- Related specifications for SOA security
  - WS-Security, WS-SecurityPolicy, WS-Trust, WS-SecureConversation, WS-Federation, XACML, Extensibe Rights Markup Language, XML Key Management, XML, Signature, SAML, .NET Passport, Secure Socket Layer, WS-I Basic Security Profile

Example : Geo-Information Service (GIS) Cloud



Courtesy: Dr. Deyi Li, Chinese Academy of Engineering

Major Tasks to Build a GIS Cloud
for Community Services

- Must build fast mining engine to support data-mining and data aggregation
- Community or virtual community discovery
- Support of bandwidth demand for video, voice, data, etc.
- Providing GISs and mashup services with GIS
- Handling of massive registered users and their payments
- Interoperability among different GISs in the industry
- Centralized, domain-oriented, and multi-granularity services, make GIS centers public and specialized.

Conclusion