Grid Standards & Architecture

Reference: Grid Computing – on Demand Series,

Joshy Joseph & Craig Fellenstein, Pearson, IBM Press, 2011

Overview

- Grid Standards
- Grid Architecture

Grid Computing

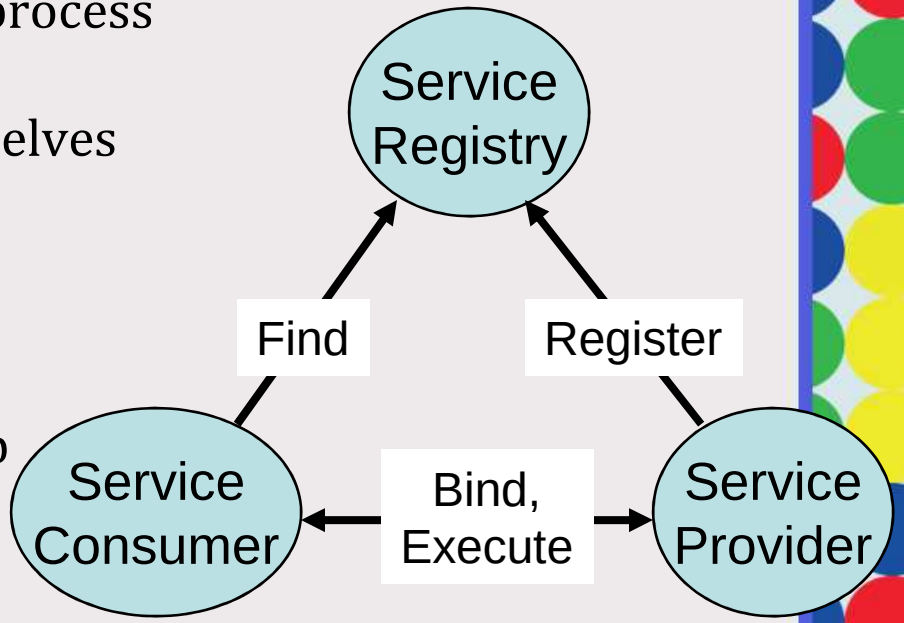
- **Grid computing** is a form of distributed computing whereby a "super and virtual computer" is composed of a cluster of networked, loosely coupled computers, acting in concert to perform very large tasks.
- Grid computing (Foster and Kesselman, 1999) is a growing technology that facilitates the executions of large-scale resource intensive applications on geographically distributed computing resources.
- Facilitates flexible, secure, **coordinated large scale resource sharing** among dynamic collections of individuals and institutions.
- Enable communities ("virtual organizations") to share geographically distributed resources as they pursue common goals

Standards of Grid

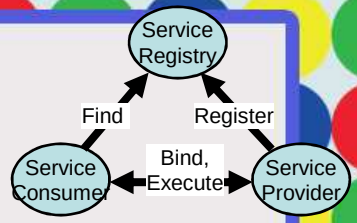
- **Web Services**
  - XML - (Extended Markup Language)
  - SOAP - (Simple Object Access Protocol)
  - WSDL - (Web Service Description Language)
- **OGSA (Open Grid Service Architecture)**
  - It helps to standardize the services provided by grid such as resource discovery, management, security through standard web service interface.
  - OGSA does not provide implementation
- **OGSI (Open Grid Service Infrastructure)**
  - OGSI provides details of implementation
  - It provides formal and technical specification needed for implementation of grid services through WSDL.
- **WSRF (Web Service Resource Framework)**
  - Defines generic and open framework for modeling and accessing stateful resources using web services.
- **OGSA- DAI (OGSA- Data Access and Integration)**
  - Middleware to provide access and integration of distributed data sources using grid.

What is Service Oriented Architecture (SOA)?

- An SOA application is a composition of services
- A “service” is the atomic unit of an SOA
- Services encapsulate a business process
- Service Providers Register themselves
- Service use involves: Find, Bind, Execute
- Most well-known instance is Web Services

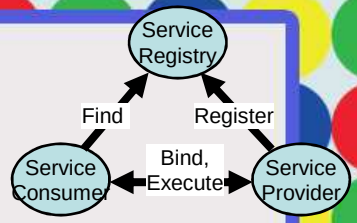


SOA Actors



- **Service Provider**
  - Provides a stateless, location transparent business service
- **Service Registry**
  - Allows service consumers to locate service providers that meet required criteria
- **Service Consumer**
  - Uses service providers to complete business processes

SOA Benefits



Business Benefits

- Focus on Business Domain solutions
- Leverage Existing Infrastructure
- Agility

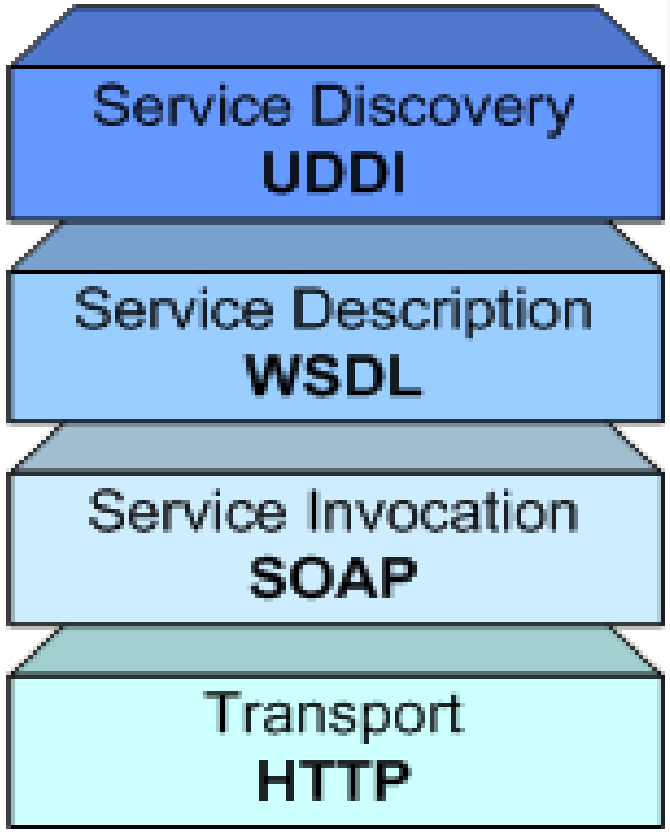
Technical Benefits

- Loose Coupling
- Autonomous Service
- Location Transparency

Services in the Web and the Grid
Web services

- Define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of relevant service providers
- A distributed computing technology (like CORBA, RMI…)
- They allow us to create loosely coupled client/server applications.

Services in the Web and the Grid
Web Services Architecture



Find Web services which meet certain requirements
(Universal Description, Discovery and Integration)

Services describe their own properties and methods
(Web Services Description Language)

Format of requests(client) and responses (server)
(Simple Object Access Protocol)

Message transfer protocol

(Hypertext Transfer Protocol)

Grid Architecture

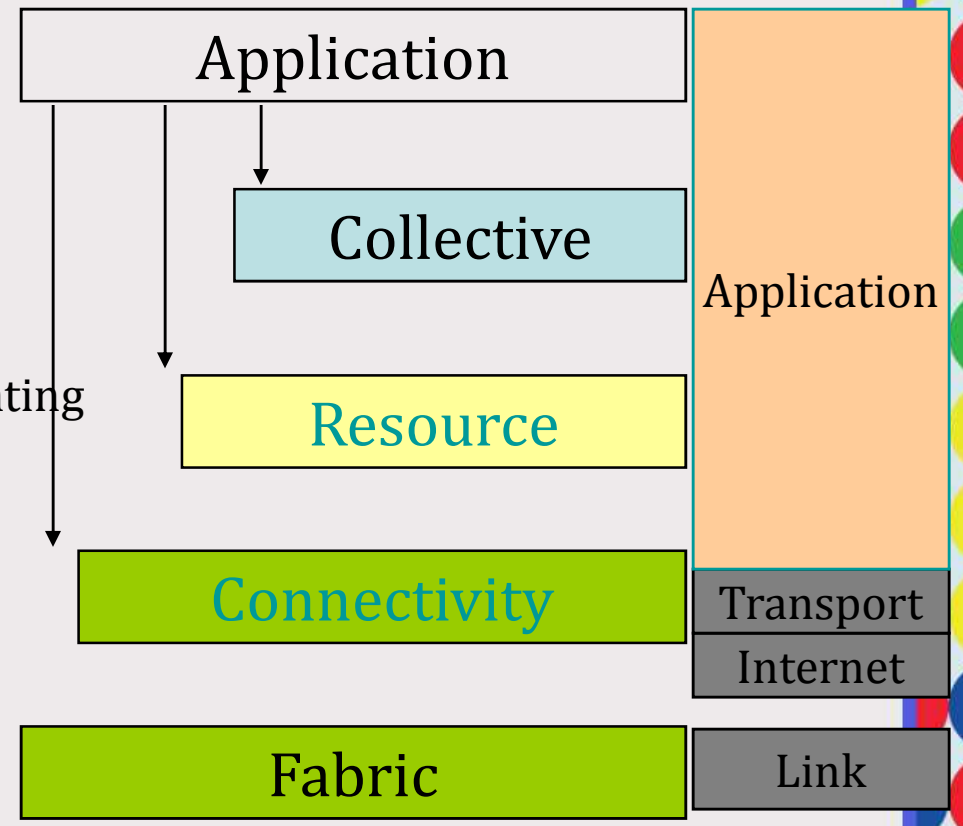
Layered Grid Architecture (By Analogy to Internet Architecture)

"Coordinating multiple resources": ubiquitous infrastructure services, app-specific distributed services

"Sharing single resources": negotiating access, controlling use

"Talking to things": communication (Internet protocols) & security

"Controlling things locally": Access to, & control of, resources



Example: Data Grid Architecture

App	Discipline-Specific Data Grid Application
Collective (App)	Coherency control, replica selection, task management, virtual data catalog, virtual data code catalog, …
Collective (Generic)	Replica catalog, replica management, co-allocation, certificate authorities, metadata catalogs,
Resource	Access to data, access to computers, access to network performance data, …
Connect	Communication, service discovery (DNS), authentication, authorization, delegation
Fabric	Storage systems, clusters, networks, network caches, …

Fabric Layer: Interface to Local Resources

- The Fabric layer defines the resources that can be shared. This could include computational resources, data storage, networks, catalogs, and other system resources.
- These resources can be physical resources or logical resources by nature.

Connectivity Layer: Manages Communications

- The Connectivity layer defines the core communication and authentication protocols required for grid-specific networking services transactions. Communications protocols, which include aspects of networking transport, routing, and naming, assist in the exchange of data between fabric layers of respective resources.
- The most commonly used Network layer protocol is the TCP/IP Internet protocol stack

Connectivity Layer: Manages Communications

- **Single sign-on:** This provides any multiple entities in the grid fabric to be authenticated once.
- **Delegation:** This provides the ability to access a resource under the current users permissions set.
- **Integration with local resource specific security solutions:** This may include (for example) Kerberos security methods, Windows security methods, Linux security methods, and UNIX security methods.
- **User-based trust relationships:** trust relationship between users and multiple service providers is very critical.
- **Data security:** important in order to provide data integrity and confidentiality

Resource Layer: Sharing of a Single Resource

- The Resource layer utilizes the communication and security protocols defined by the networking communications layer, to control the secure negotiation, initiation, monitoring, metering, accounting, and payment involving the sharing of operations across individual resources.
- Two main Protocols
  - **Information Protocols:** collects structure and the operational state of a single resource
  - **Management Protocols :** Negotiation, performing operations on resources, monitoring status of operation, accounting and payment management

The Collective Layer: Coordinating Multiple Resources

- Collective layer is responsible for all **global resource management** and **interaction** with a **collection** of **resources**.
- Example Collective services.
  - Discovery Services
  - Coallocation, Scheduling, and Brokering Services
  - Monitoring and Diagnostic Services
  - Data Replication Services
  - Grid-Enabled Programming Systems
  - Workload Management Systems and Collaborative Frameworks
  - Software Discovery Services
  - Community Authorization Servers
  - Community Accounting and Payment Services

Application Layer: User-Defined Grid Applications

- These are user applications, which are constructed by utilizing the services defined at each lower layer.
- Application can directly access the resource, or can access the resource through the Collective Service interface APIs (Application Provider Interface).
- Each layer in the grid architecture provides a set of APIs and SDKs (software developer kits) for the higher layers of integration.
- These user-defined grid applications are (in most cases) domain specific and provide specific solutions.

Summary

- Grid Standards
- Grid Architecture