A Networking Primer for Grid Computing

Supplement to Module 1

Basic Networking and related Definitions

Application Programming Interface (API) defines the interface.

- Refers to definition, not implementation
  - For example, there are many implementations of MPI
- Specification often language-specific (or IDL)
  - Routine name, number, order and type of arguments; mapping to language constructs
  - Behavior or function of routine
- Examples
  - GSS API (security), MPI (message passing)

Application Programming Interface

- A specification for a set of routines to help application development
  - Refers to definition, not implementation
  - E.g., there are many implementations of MPI
- Spec often language-specific (or IDL)
  - Routine name, number, order and type of arguments; mapping to language constructs
  - Behavior or function of routine
- Examples
  - GSS API (security), MPI (message passing)

A Software Development Kit (SDK) is a particular instantiation of an API

- An SDK consists of libraries and tools
  - Provides implementation of API specification
- One API can have multiple SDKs
- Examples of SDKs
  - MPICH

Software Development Kit

- A particular instantiation of an API
- SDK consists of libraries and tools
  - Provides implementation of API specification
- Can have multiple SDKs for an API
- Examples of SDKs
  - MPICH, Motif Widgets

Network Protocol

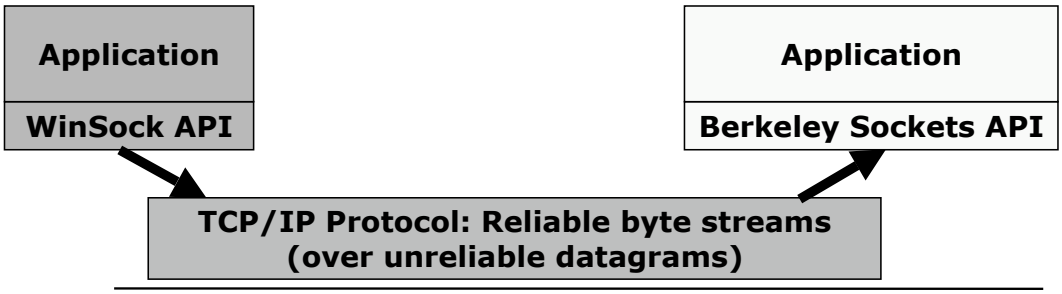
- A formal description of message formats and a set of rules for exchanging messages
- Good protocols are designed to do just one thing
  - Protocols can be layered
- Examples of protocols
  - IP, TCP, TLS (SSL), HTTP, Kerberos

Syntax

- Rules for encoding information, e.g.
  - XML, Condor ClassAds, Globus RSL
  - X.509 certificate format (RFC 2459)
  - Cryptographic Message Syntax (RFC 2630)
- Distinct from protocols
  - One syntax may be used by many protocols (e.g., XML); & useful for other purposes
- Syntaxes may be layered
  - E.g., Condor ClassAds -> XML -> ASCII

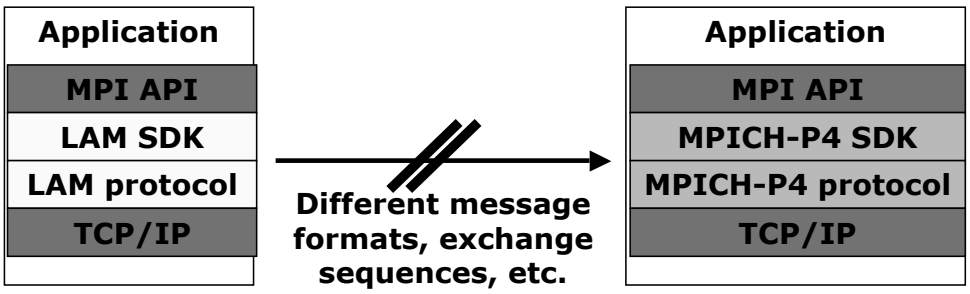
Protocols can have multiple APIs.

- TCP/IP APIs include BSD sockets, Winsock, System V streams, …
- The protocol provides *interoperability*
  - Programs using different APIs can exchange information
  - I don't need to know remote user's API



An API can have multiple protocols

- MPI provides portability: any correct program compiles & runs on a platform
- Does not provide interoperability: all processes must link against same SDK
  - E.g., MPICH and LAM versions of MPI



APIs and protocols are both important

- Standard APIs/SDKs are important
  - They enable application *portability*
    - Can move application to different places
  - But w/o standard protocols, interoperability is hard
  - Example: MPI
- Standard protocols are important
  - Between computers
  - Enable *interoperability*
    - Applications can talk to each other
  - Enable shared infrastructure – example: the internet
  - But w/o standard APIs/SDKs, application portability is hard (different platforms access protocols in different ways)

Communication Elements

- Links, routers, switches, name servers, protocols
- Infrastructure evolves slowly (politics, large scale changes, money)
- Gilder's Law: total bandwidth of communication systems doubles every six months
- Change in LAN to desktops
  - 100 mbps shared
  - 100 mbps switched
  - 1 gbps
  - 10 gbps
- Clusters: Gigabit ethernet (TCP/IP and MPICH/LAM) standard, Myrinet (own MPI drivers) better performance

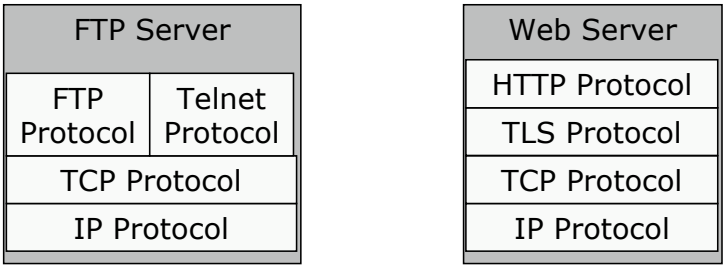
Network Speeds

- Analog modem: 57 kbps
- GPRS: 114 kbps
- Bluetooth: 723 kbps
- T-1: 1.5 Mbps
- Eth 10Base-X: 10Mbps
- 802.11b (WiFi) 11 Mbps
- T-3: 45 Mbps
- OC-1: 52 Mbps
- Fast Eth 100Base-X: 100 Mbps
- OC-12: 622 Mbps
- GigEth 1000Base-X: 1 Gbps
- OC-24: 1.2 Gbps
- OC-48: 2.5 Gbps
- OC-192: 10 Gbps
- 10 GigEth: 10 Gbps
- OC-3072: 160 Gbps

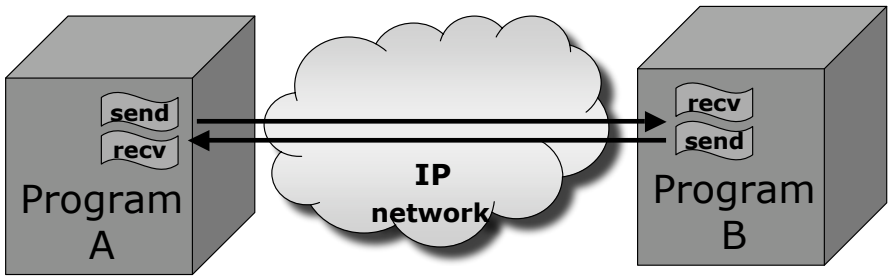
- Home internet
  - Upload: 35 KB/s
  - Download 250 KB/s

Network Enabled Services

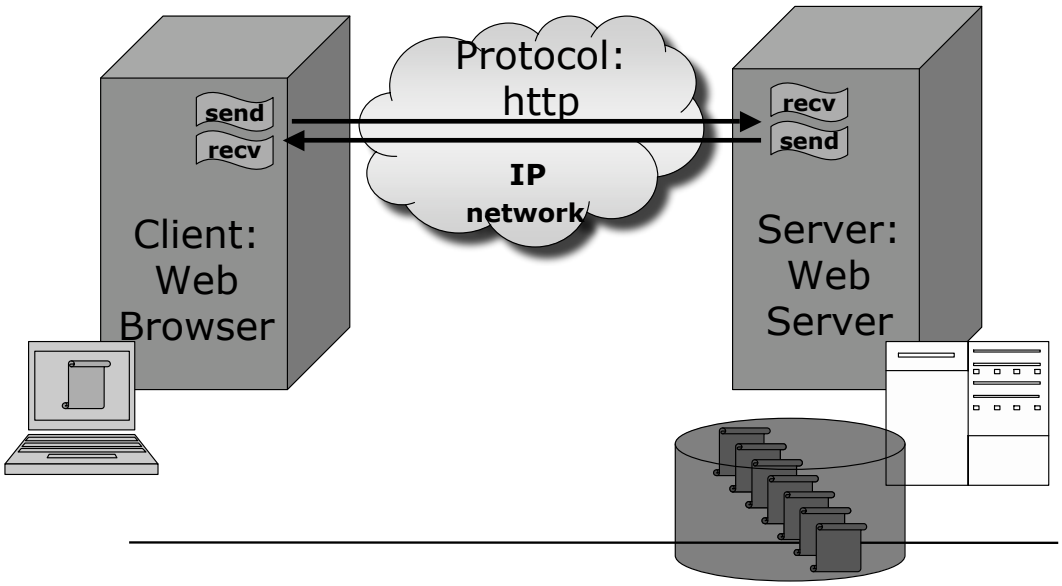
- Implementation of a protocol that defines a set of capabilities
  - Protocol defines interaction with service
  - All services require protocols
  - Not all protocols are used to provide services (e.g. IP, TLS)
- Examples: FTP and Web servers



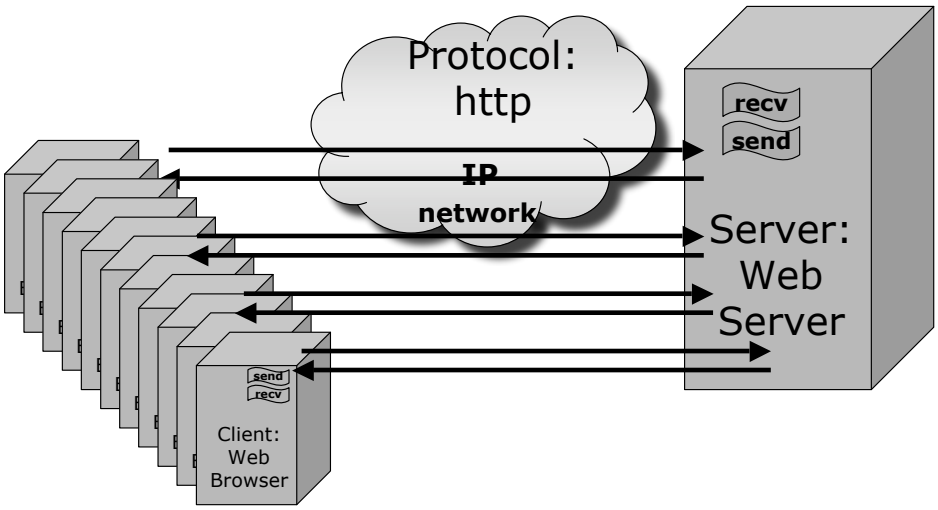
Sockets – the basic building block



Services are built on Sockets



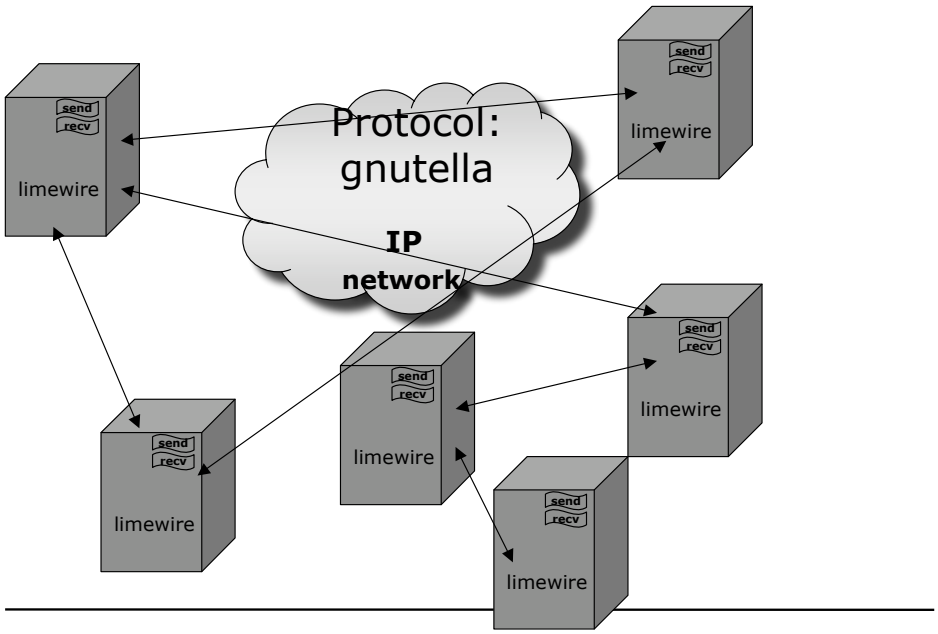
Client-Server Model



Familiar Client-Server Apps

- Email
  - Protocols: POP, SMTP
- File Copying
  - Protocol: FTP
- Logging in to remote computers
  - Protocol: Telnet

Peer-to-Peer Model



Familiar Peer-to-Peer Apps

- File (music) Sharing
  - Protocols: Napster, Gnutella
- Chat (sort of)
  - Protocols: IRC, Instant Messenger
- Video Conferencing
  - Protocols: H323

Basic Networking and related Definitions

Basic Networking Skills

- Grid usage requies a basic level of UNIX and networking skills
- Remember:
  - Find out about Unix commands or tools using “man”
    - > man nslookup
  - You should read up and practise basic networking, security, linux, skills.
  - There is lots of information on the web
    - “what is” with Google, www.wikipedia.org, www.webopedia.com

IP Addresses

- All computers on the Internet use TCP/IP.
  - IP: responsible for moving packets of data from node to node
  - TCP: responsible for correct delivery of data
- TCP/IP assigns a unique number or "IP address" to every computer on the Internet
  - 32-bit number
  - Written as four numbers, like: 128.105.3.61
- An IP Address identifies a network interface, not a computer.
  - A computer can have multiple IP addresses.

Domain Name Server (DNS)

- DNS maps IP addresses to names, and vice-versa
  - www.amazon.com ←→ 72.21.206.5
  - Discover this with "host" or "nslookup" or "dig"
  - Try all three—how do they differ?

DNS

- host www.amazon.com

  www.amazon.com has address 207.171.163.90

- nslookup -sil www.amazon.com

  Server: 10.0.1.1
  Address: 10.0.1.1#53
  Non-authoritative answer:
  Name: www.amazon.com
  Address: 207.171.166.102

- dig www.amazon.com

  ;; ANSWER SECTION:
  www.amazon.com. 20 IN A 207.171.166.102

  ;; Query time: 4 msec
  ;; SERVER: 10.0.1.1#53(10.0.1.1)
  ;; WHEN: Sat Jun 24 08:59:00 2006
  ;; MSG SIZE rcvd: 48

Ping!

- Is a computer on the network?
- Use ping to find out

```
% ping cu.ncsa.uiuc.edu
PING cu12.ncsa.uiuc.edu (141.142.30.77): 56 data bytes
64 bytes from 141.142.30.77: icmp_seq=0 ttl=233 time=53.663 ms
64 bytes from 141.142.30.77: icmp_seq=1 ttl=233 time=55.615 ms
64 bytes from 141.142.30.77: icmp_seq=2 ttl=233 time=55.153 ms
64 bytes from 141.142.30.77: icmp_seq=3 ttl=233 time=57.184 ms
^C
--- cu12.ncsa.uiuc.edu ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 53.663/55.404/57.184/1.256 ms
```

- Some sites block the use of ping :(

Internet routes

- Between you and a computer on the network, there is an often complex route.

```
% traceroute www.cs.uwm.edu
traceroute to miller.cs.uwm.edu (129.89.143.24), 30 hops max, 40
   byte packets
1  svi-121.cisco1.cs.wisc.edu.105.128.in-addr.arpa
(128.105.121.248)  0.423 ms  0.242 ms  0.227 ms
2  rh-cssc-b280c-2-core-vlan-492.net.wisc.edu (144.92.128.186)
0.404 ms  4.985 ms  0.489 ms
… snip…
6  r-uwmilwaukee-isp-atm1-0-1.wiscnet.net (140.189.8.2)  2.730
ms  2.603 ms  2.689 ms
7  space-needle-mke.csd.uwm.edu (216.56.1.194)  2.836 ms  2.718
ms  2.748 ms
8  miller.cs.uwm.edu (129.89.38.24)  2.754 ms *  2.796 ms
```


Internet routing

- To be supplied.

Port numbers

- A port number indicates which program to talk to on a computer.
- Some port numbers are standard:
  - HTTP (web): port 80
  - SMTP (mail): port 25
  - Ping: port 7
- Some port numbers are assigned dynamically when you run a server.

Netstat

- Netstat can answer the question: is a program running on a port on the local computer.

```
netstat --protocol=inet -l
tcp  0 0 *:finger  *:* LISTEN
```

- -l meant "listening for connections". Look for active connections:

```
netstat --protocol=inet | grep ssh
% netstat --protocol=inet | grep ssh
tcp  0 0 chopin.cs.wisc.edu:ssh  ppp-67-38-160-
108:20715 ESTABLISHED
tcp  0 0 chopin.cs.wisc.edu:ssh
68.185.181.47:1176      ESTABLISHED
…
```

Telnet

- Telnet isn't just for remote access to a computer
- Telnet can tell you if remote services are running correctly and "listenening" for "connections".
- Is ssh running?
  - Find ssh port number in /etc/services. It's 22.
  - telnet <host> 22. Example:

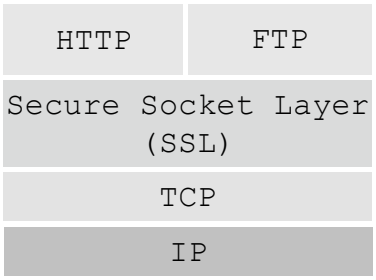
```
telnet beak.cs.wisc.edu 22
Trying 128.105.146.14...
Connected to beak.cs.wisc.edu (128.105.146.14).
Escape character is '^]'.
SSH-1.99-OpenSSH_3.6.1p2
^]    (That is control-right bracket)
telnet> quit
```


APIs and Protocols are Both Important

- Standard APIs/SDKs are important
  - They enable application *portability*
  - But w/o standard protocols, interoperability is hard (every SDK speaks every protocol?)
- Standard protocols are important
  - Enable cross-site *interoperability*
  - Enable shared infrastructure
  - But w/o standard APIs/SDKs, application portability is hard (different platforms access protocols in different ways)

Secure Sockets Layer: SSL (TLS)

- Protocol that transmits communications over the Internet in an encrypted form
  - SSL ensures that the information is sent, unchanged, only to the server you intended to send it to.
- SSL uses a private key to encrypt data
  - Netscape and Internet Explorer support SSL
  - Web sites use SSL to obtain confidential user information, such as credit card numbers.
  - By convention, URLs that require an SSL connection start with https: instead of http:.
- Newest version of SSL is called Transport Later Security (TLS)



OpenSSL

- An Open Source implementation of the SSL (Secure Sockets Layer) and TLS (Transport Layer Security) protocols
- OpenSSL is used by Apache HTTP Server for https support and by MySQL to provide secure database access.

OpenSSH

- OpenSSH is an implementation of the SSH protocol suite of tools
- OpenSSH encrypts all traffic (including passwords)
- OpenSSH provides a variety of authentication methods.
- The OpenSSH suite includes
  - ssh program which replaces rlogin and telnet,
  - scp which replaces rcp,
  - sftp which replaces ftp.
  - Also other basic utilities like ssh-add, ssh-agent, ssh-keygen

Very Basic Web Services

Web Services

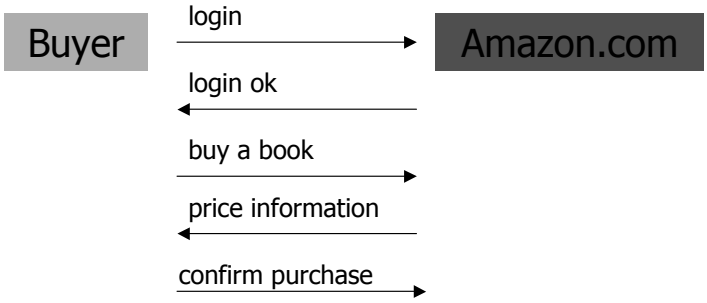
A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. (W3C definition)

In plain words, they provide a good mechanism to connect heterogeneous systems with WSDL, XML, SOAP, XML, UDDI.

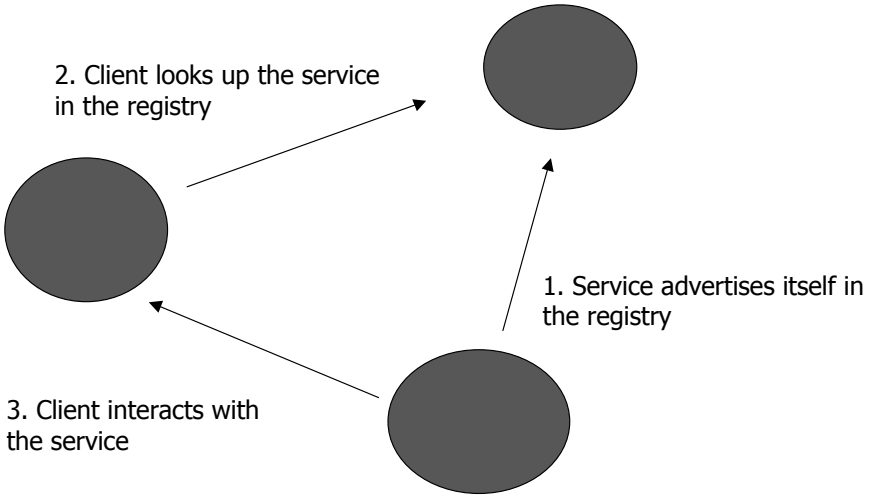
A Sample interaction

Key things to note:

Request/response



The Web Service state machine

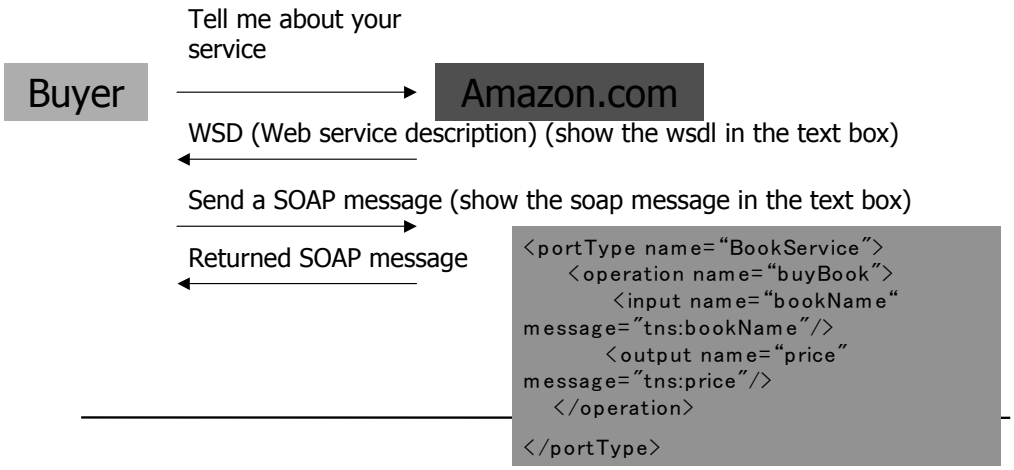


XML (Extensible Markup Language)

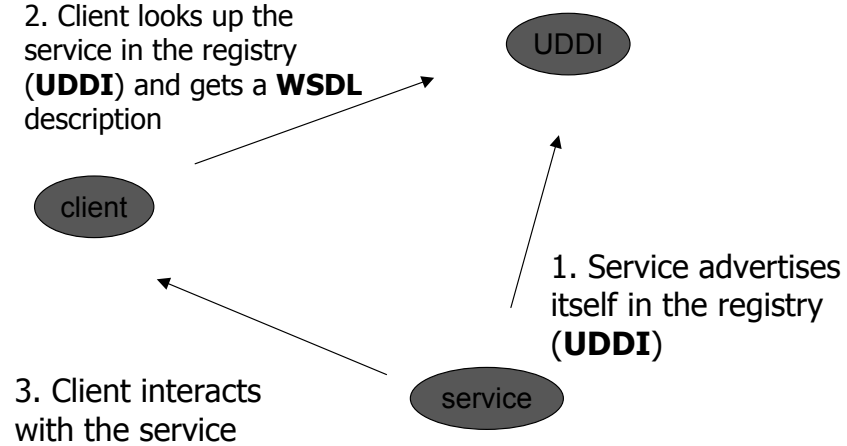
- A language for describing data
- Platform independent and self-describing
- Good for distributed computing where heterogeneous systems co-exist

```
<?xml version="1.0"?>
<contact-info>
  <name>John Smith</name>
  <company>University of Florida</company>
  <phone>352-392-1200</phone>
</contact-info>
</xml>
```

An example scenario



The Web Service state machine



The technologies

- WSDL
  - to describe the basic format of web service requests
- SOAP
  - defines a uniform way of passing XML-encoded data
- XML and
- UDDI concepts
  - for finding web services

SOAP (Simple Object Access Protocol)

- SOAP is a protocol specification that defines a uniform way of passing XML-encoded data
- In also defines a way to perform remote procedure calls (RPCs) using HTTP as the underlying communication protocol
- It is the underlying protocol for all web services

WSDL (Web Service Description Language)

- WSDL tells us about the web service interface.
- WSDL provides a way for service providers to describe the basic format of web service requests over different protocols or encodings
- It provides the following information about the service
  - What the service can do
  - Where it resides
  - How to invoke it
- Provides a platform and language independent abstraction of the service
- It specifies the port types, messages exchanged and how data is encoded etc.

UDDI: Universal Description, Discovery and Integration

- A protocol for finding web services
- Registries of web services can be maintained
- The primary purpose is to find services with certain qualities

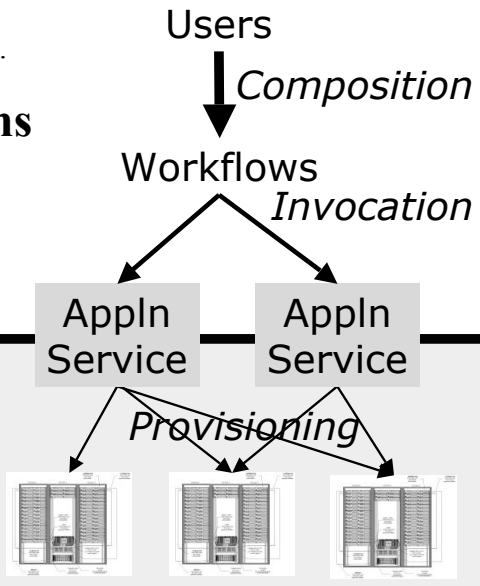
Web Services vs Grid Services

- Though web services are great, some key things that are required on the grid are missing
  - State management
  - Global Service Naming
  - Reference resolution
  - more …
- Grid services are web services that are customized to grid environment
  - Similar to web services they provide the glue to interact with heterogeneous systems
  - Grid computing needs a few additional features, e.g. state
  - Lots of development & politics here!

Grid architecture is evolving to a Service-Oriented approach.

...but this is beyond our workshop's scope.
See "Service-Oriented Science" by Ian Foster.

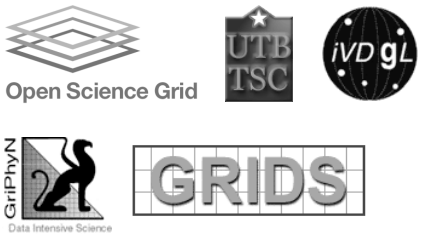
- Service-oriented **applications**
  - Wrap applications as services
  - Compose applications into workflows
- Service-oriented **Grid infrastructure**
  - Provision physical resources to support application workloads



"The Many Faces of IT as Service", Foster, Tuecke, 2005

Based on:

Grid Intro and Fundamentals Review



Dr Gabrielle Allen
Center for Computation & Technology
Department of Computer Science
Louisiana State University
gallen@cct.lsu.edu

Grid Summer Workshop
June 26-30, 2006