CS514: Intermediate Course in Operating Systems

Professor Ken Birman
Vivek Vishnumurthy: TA

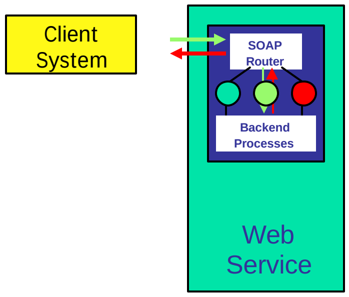
Today

- Web Services – Introduction
- “Remote Procedure Call” in WS
  - Binding, Marshalling…
- Using TCP as the transport for RPCs
  - Connectivity Issues: NAT, Firewall

What are Web Services?

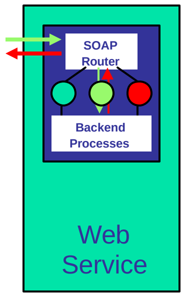
- Today, we normally use Web browsers to talk to Web sites
  - Browser names document via URL (lots of fun and games can happen here)
  - Request and reply encoded in HTML, using HTTP to issue request to the site
- Web Services generalize this model so that computers can talk to computers

What are Web Services?



What are Web Services?

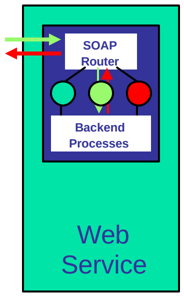
- “Web Services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP.”

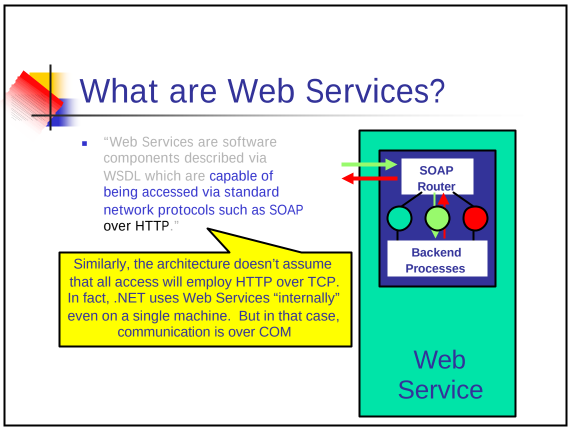


What are Web Services?

- “Web Services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP.”

Today, SOAP is the primary standard. SOAP provides rules for encoding the request and its arguments.

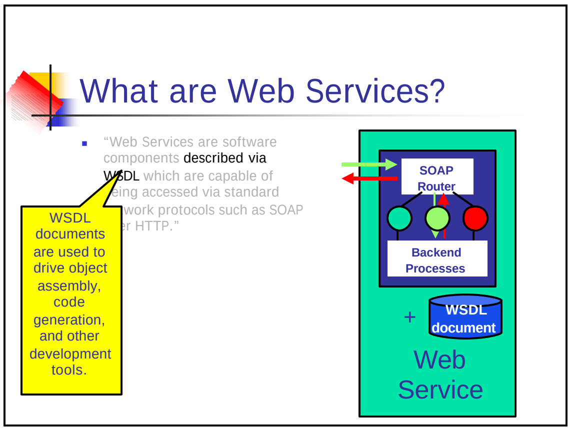




What are Web Services?

- "Web Services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP."

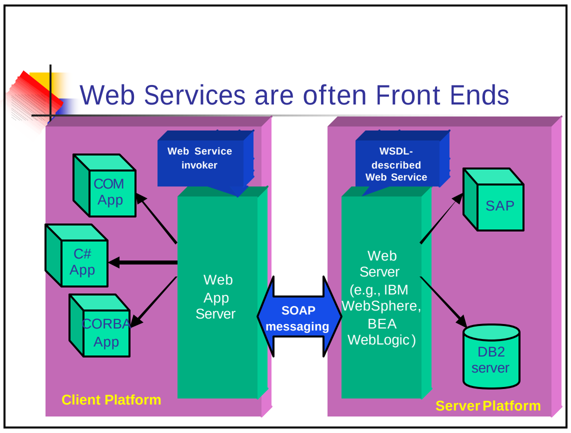
Similarly, the architecture doesn't assume that all access will employ HTTP over TCP. In fact, .NET uses Web Services "internally" even on a single machine. But in that case, communication is over COM



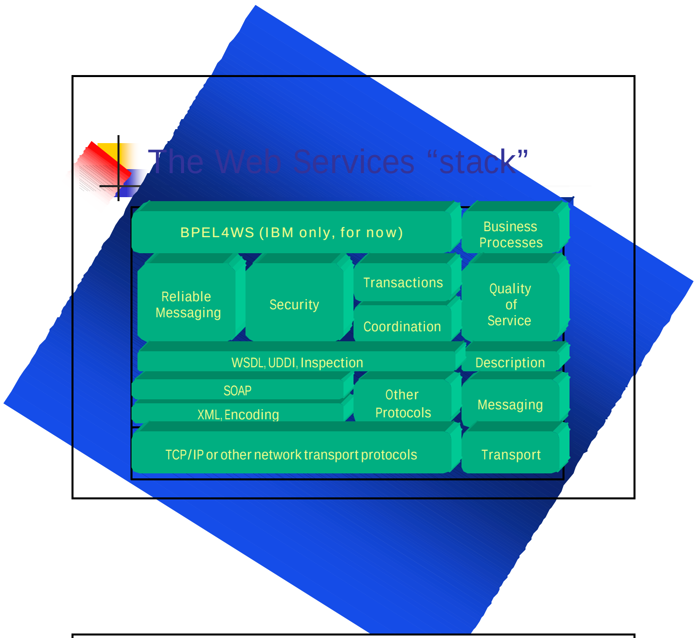
What are Web Services?

- "Web Services are software components described via WSDL which are capable of being accessed via standard network protocols such as SOAP over HTTP."

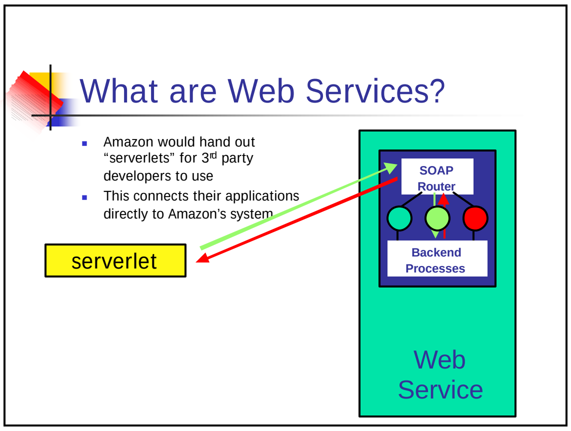
WSDL documents are used to drive object assembly, code generation, and other development tools.



Web Services are often Front Ends

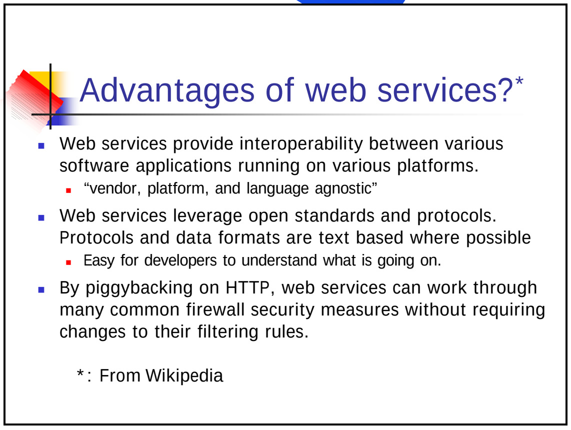


The Web Services "stack"



What are Web Services?

- Amazon would hand out "serverlets" for 3rd party developers to use
- This connects their applications directly to Amazon's system



Advantages of web services?*

- Web services provide interoperability between various software applications running on various platforms.
  - "vendor, platform, and language agnostic"
- Web services leverage open standards and protocols. Protocols and data formats are text based where possible
  - Easy for developers to understand what is going on.
- By piggybacking on HTTP, web services can work through many common firewall security measures without requiring changes to their filtering rules.

*: From Wikipedia

How Web Services work

- First the client *discovers* the service.
  - More in next lecture!
- Typically, client then *binds* to the server.
  - By setting up TCP connection to the discovered address .
  - But binding not always needed.

How it works…

- Next build the SOAP request: (*Marshaling*)
  - Fill in what service is needed, and the arguments. Send it to server side.
  - More details in next lecture
- SOAP router routes the request to the appropriate server(assuming more than one available server)
  - Can do load balancing here.

How it works…

- Server unpacks the request, (*Demarshaling*) handles it, computes result.
- Result sent back in the reverse direction: from the server to the SOAP router back to the client.

Marshalling Issues

- Data exchanged between client and server needs to be in a platform independent format.
  - “Endian”ness differ between machines.
  - Data alignment issue (16/32/64 bits)
  - Multiple floating point representations.
  - Pointers
  - ➢ (Have to support legacy systems too)

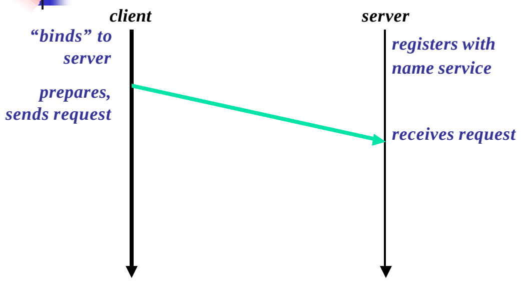
Marshalling…

- In Web Services, the format used is XML.
  - In UNICODE, so very verbose.
  - There are other, less general, but more efficient formats.

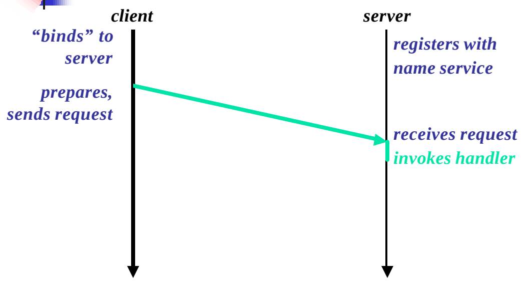
RPC – Remote Procedure Call

- Call a procedure on a remote machine “just” as you would on the local machine.
- Introduced by Birrell and Nelson in 1985
- Idea: mask distributed computing system using a “transparent” abstraction
  - Looks like normal procedure call
  - Hides all aspects of distributed interaction
  - Supports an easy programming model
- Today, RPC is the core of many distributed systems.
- Can view the WS client server interaction as an RPC.

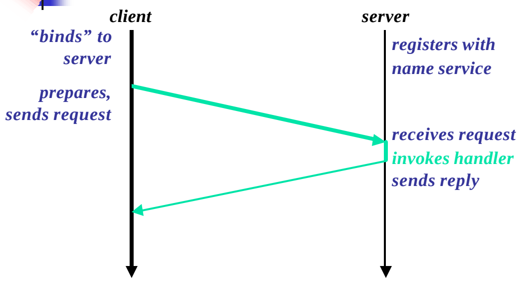
The basic RPC protocol



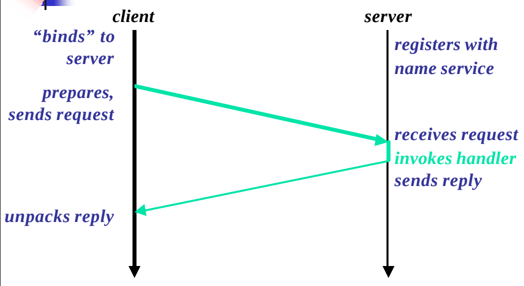
The basic RPC protocol



The basic RPC protocol



The basic RPC protocol



RPC – what can go wrong?

- Network failure, client failure, server failure
- Assuming only network idiosyncrasies for now...
- RPCs use acks to make packet transmission more reliable.
  - If timeout with no ack, resend packet.
  - Leads to the issue of replayed requests.
- Each packet has a sequence number and timestamp embedded to enable detection of duplicates.

RPC Optimization

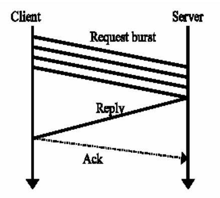


Figure 4.4. RPC using a burst protocol; here the reply is sent soon enough so that an acknowledgement to the burst is not needed.

- Delay sending acks, so that imminent reply itself acts as an ack.
- Don’t send acks after each packet.
- Send ack only at the end of transmission of entire RPC request.
- NACK sent when missing sequence number detected

What happens when machines could fail too?

- What does a failed request mean?
  - Network failure and/or machine failure!
  - Client that issued request would not know if the server processed the request or not.

How about layering RPC on TCP?

- TCP gives reliable in-order delivery, flow control and congestion control.
  - Reliable: Acknowledgments and retransmissions.
  - In-order: Sequence numbers embedded in each message.
  - Flow Control: Max allowed window size.

TCP...

- Congestion Control: the saw tooth curve
  - Ramp up as long as no timeouts.
    - Slow -start phase – exponential increase (until the slow-start threshold is hit)
    - Congestion Avoidance phase – additive increase
  - Multiplicative Decrease on timeout.

TCP optimizations

- Random Early Detection
- Selective Acknowledgments
- Fast Retransmit/Recovery

Back to RPC on TCP:

- Eg: Web Services, CORBA
- TCP gives reliable communication when both ends and the network connecting them are up.
- So the RPC protocol itself does not need to employ timeouts and retransmission.
  - Simpler RPC implementation.

RPC/TCP ...

- Does this mean RPC got more reliable by using TCP?
- NO, since broken connections reported by TCP mean the same thing they did earlier (without TCP)
  - Client still doesn't know whether the server processed its request.
- No standard way of handling timeouts.

RPC Semantics

- "Exactly Once"
  - Each request handled exactly once.
  - Impossible to satisfy, in the face of failures.
  - Can't tell whether timeout was because of node failure or communication failure.

RPC Semantics…

- "At most Once"
  - Each request handled at most once.
  - Can be satisfied, assuming synchronized clocks, and using timestamps.
- "At least Once"
  - If client is active indefinitely, the request is eventually processed (maybe more than once)

WS & RPC Connectivity Issues: Network Address Translation

- IP Address – 32 bits only.
  - Address Space Shortage.
- NATs invented to overcome this problem.
- Have a NAT box in between a private network and the internet.
- Can use locally allocated addresses within private network.
- The NAT router maps the internal IP address:port to the external IP address:port and vice-versa.

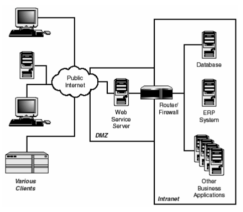
NAT…

- The internal address is not addressable from outside.
  - A measure of security.
  - If RPC server is behind a NAT, trouble!
    - NAT needs the host behind it to start the connection process.
    - Need to configure NAT to let specified traffic through.
    - Generally: (WS traffic)HTTP is let through.
  - Tough to have a connection in between two hosts behind NATs.
    - There are some tricks to bypass this though.

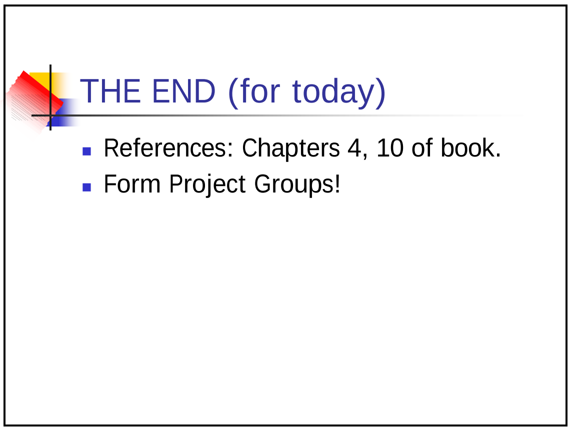
Firewall

- Allow/disallow traffic, depending on source, destination, protocol used, etc.
- Stateful: remember active flows, and disallow unexpected packets (NAT)
  - Again, need to configure to ensure server traffic gets through. (General RPC)
  - Again, (WS)HTTP does not face as much of a restriction.
- Get traffic statistics.
- Spam/virus checking, etc.
- NAT and firewall typically in the same box.

Demilitarized Zone (DMZ)



- DMZ: used to host publicly accessible services like company webpages, ftp, dns.
- Good place to host the Web Service!
- DMZ situated outside the private network.
- No outgoing connections from DMZ.
- If DMZ attacked, damage limited to DMZ.



THE END (for today)

- References: Chapters 4, 10 of book.
- Form Project Groups!