Programming the Internet

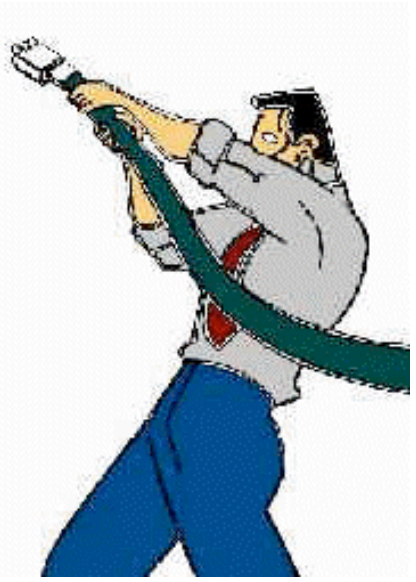
Phillip J. Windley
phil@windley.com
www.windley.com

Trending Keywords

- What keywords describe current trends in computing?
  - Personalized
  - Peer-based
  - Decentralized
  - Collaborative
  - Connected
  - Converged
  - Presence-enabled

Internet Programming

- Programming on the Internet means*
  - integrating network *services*
  - that are *very far away*
  - and *owned by strangers*
- Internet is *not* a slow LAN
- Loose Coupling



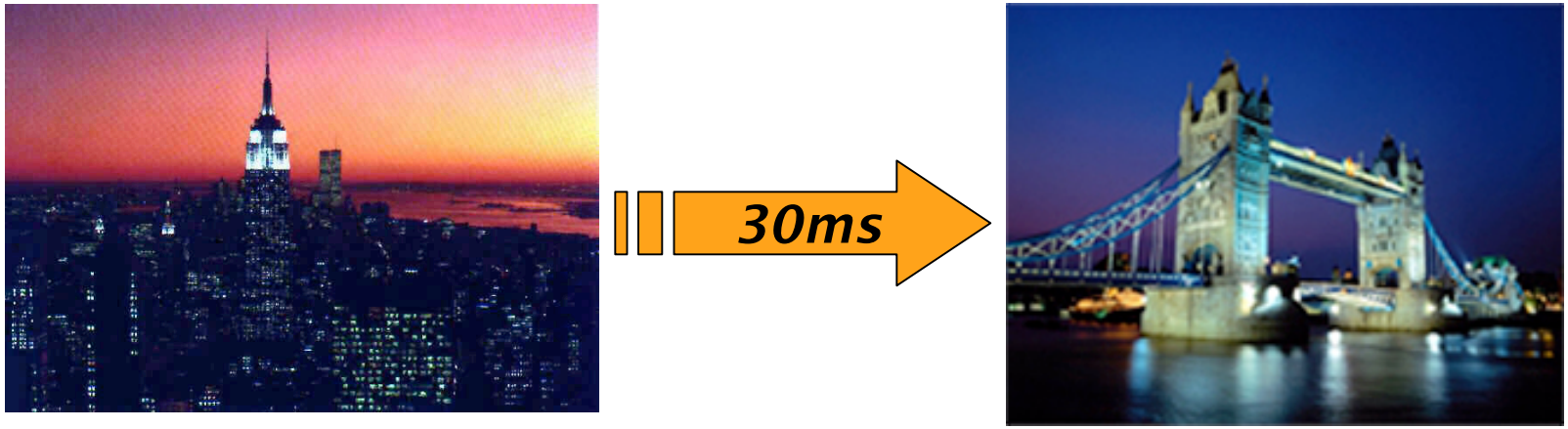
*Rohit Khare, UCLA

Network Services

- The abstract unit of software is a ***network service***
  - Contrast with subroutines, libraries, processes, objects, …
  - Message traffic becomes primary point of view
- Devices on the network provide operating system-like services

Dealing with Latency

- ***Latency*** is an absolute limit of system architecture
- One of the few physical limits on computation
- London will always be 30ms from New York, regardless of Moore's Law



Decentralized Computing

- ***Decentralized computing*** is different than distributed computing
- ***Decentralization*** means crossing organizational boundaries
- Extending a system to achieve consensus and cooperation between several separate organizations
- Every computing and communication device is owned by someone.

Web Services

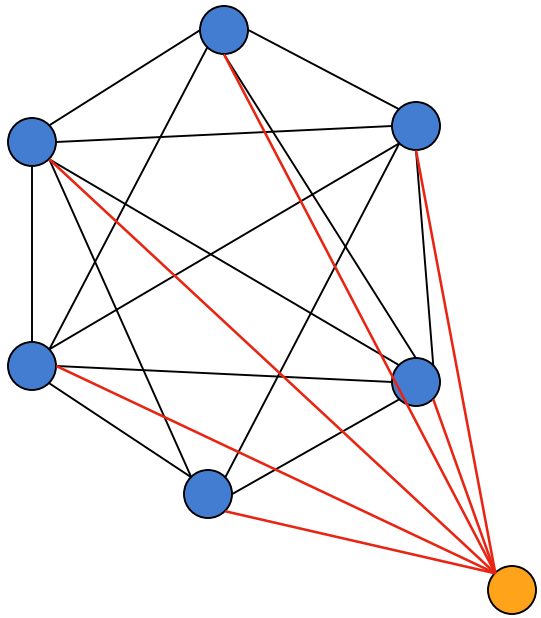
- Web services are self-contained pieces of code with three distinguishing properties:
  1. Communicate in an interoperable XML protocol, such as SOAP.
  2. Describe themselves in an interoperable XML meta-format, such as WSDL.
  3. Federate globally through XML based registry services, such as UDDI.
- Not defined in terms of SOAP, WSDL, and UDDI.

Scaling in the Internet

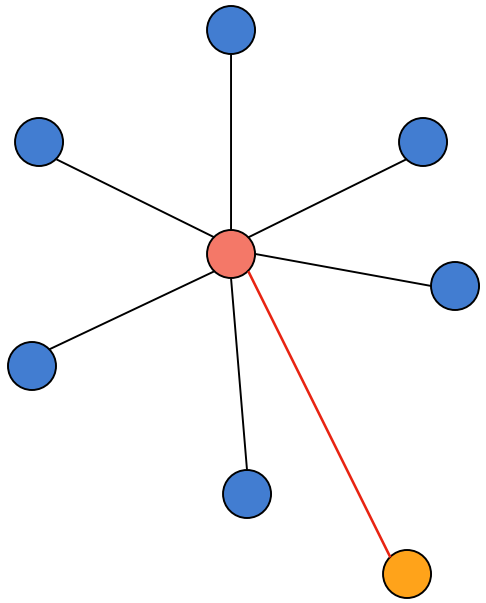
- The Internet scales in three ways:
  - Scaling across time
  - Scaling across space
  - Scaling across organizations
- Level-3 routers solve these problems for connectivity

Brokered Transactions

- Decouple provider from requestor



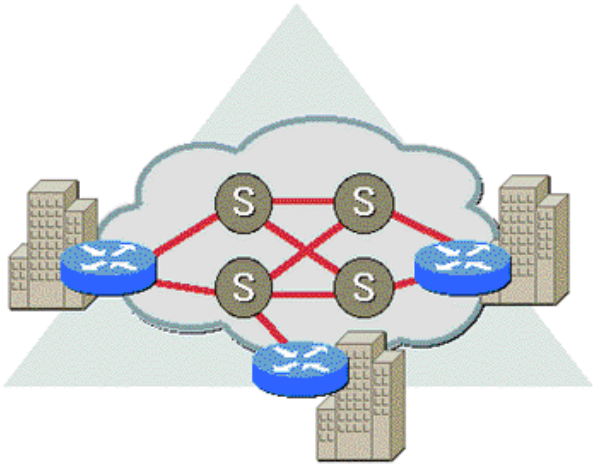
Without Broker:
N^2 connections



With Broker:
N connections

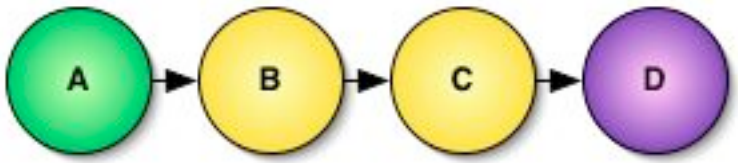
Application Layer Internetworking

- Network services as building blocks
- Exposed APIs create opportunity
- Active intermediaries are interesting
  - Switches
  - Routers
  - Proxies



GXA Routing

- GXA is XML in the SOAP Header:
  - "from" element for the message originator (A),
  - a "to" element for the final destination (D),
  - a "fwd" element to contain the forward message path for the intermediaries, and
  - a "rev" element to contain the reverse message path.



GXA Routing Example

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2001/06/soap-envelope">
  <SOAP-ENV:Header>
    <wsrp:path
       xmlns:wsrp="http://schemas.xmlsoap.org/rp/">
      <wsrp:action> http://www.im.org/chat< ;/wsrp:action>
      <wsrp:to> soap://D.com/some/endpoint< ;/wsrp:to>
      <wsrp:fwd>
         <wsrp:via> soap://B.com< ;/wsrp:via>
         <wsrp:via> soap://C.com< ;/wsrp:via>
      </wsrp:fwd>
      <wsrp:from> soap://A.com/some/endpoint< ;/wsrp:from>
      <wsrp:id>
      uuid:84b9f5d0-33fb-4a81-b02b-5b760641c1d6</wsrp:id>
    </wsrp:path>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>       ...   </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

GXA Routing

- Routing is contained in the message
- Routing is transport neutral
- The <rev> element is constructed as the message is routed
- This example shows a static route, dynamic routing is possible (WS-Referral)
- Orthogonal to security, reliability, retransmission, transactions, etc.

Message Store and Forward

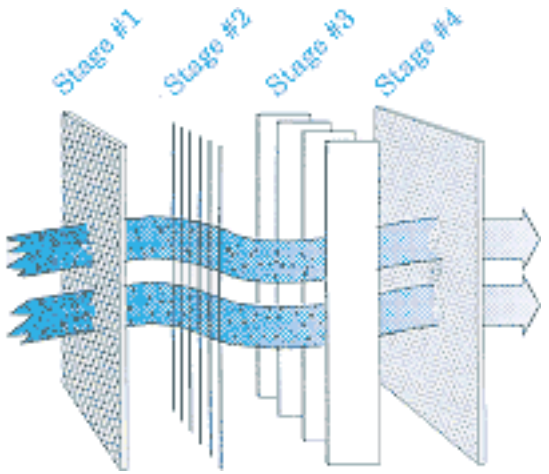
- Internet protocol sets up store and forward network for packets
- Same thing is possible for web services messages
- Guaranteed deliver
  - Once and only once
  - In order delivery
- Addresses latency and reliability

Service Call Switching

- Load balancing for web services
- Shuttling high load jobs to special servers
- Service cut-over

Context Sensitive Filtering

- Most obvious reason is for authentication and authorization
- Information could be selectively filtered based on access levels, caller, etc.



Event Monitoring

- Alarms
  - Inventory low
- Notifications
  - Big customer order
- Error handling
  - Re-routing or retry

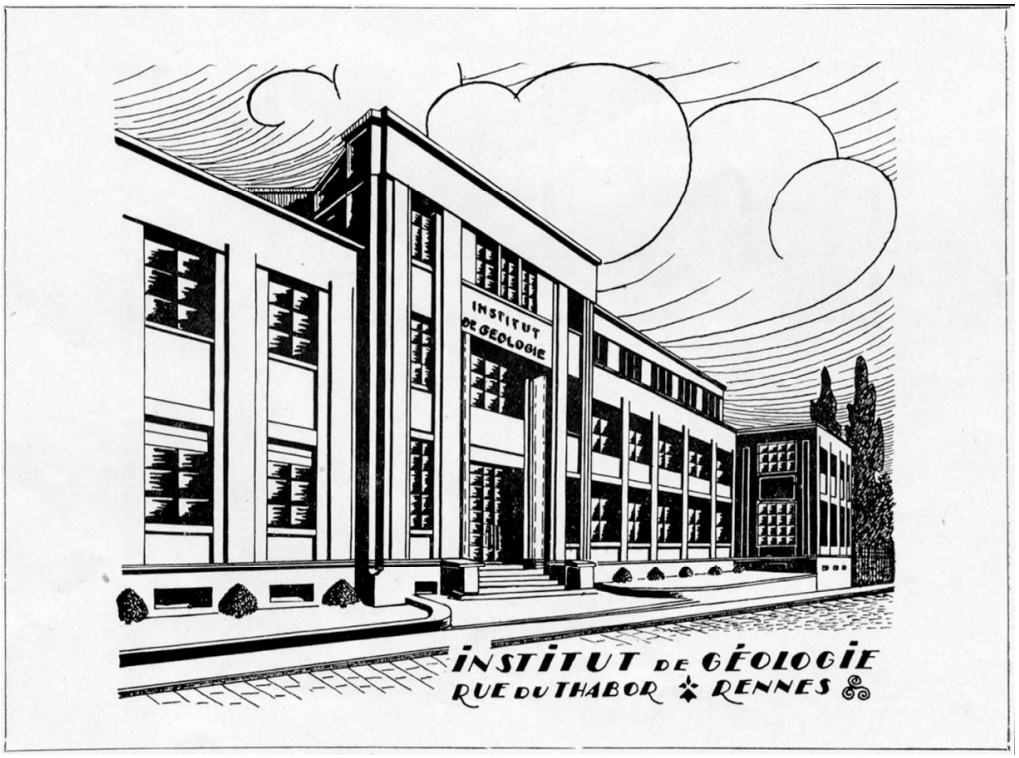
Logging

- Debugging
- Auditing
- Transaction tracking
- Selective logging
- One-way messaging
- Trust

© ERIC HOTZ "THE SCRIBE"
(PAINTED PRINT - 6" x 7") CLIENT: COLUMBIA GAMES INC

Service Facades

- Proxies
- Content delivery, routing and storage
- API facades

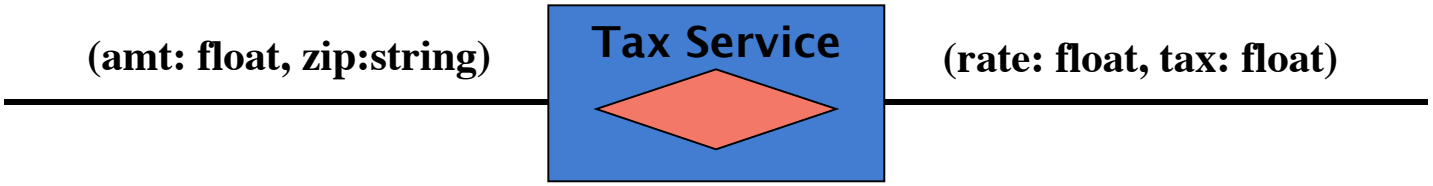


Business Rules Repositories

- Store business rules for transactions
- Router now aggregate services
- Intelligent routing
- On the fly
  - Transformation
  - Updating

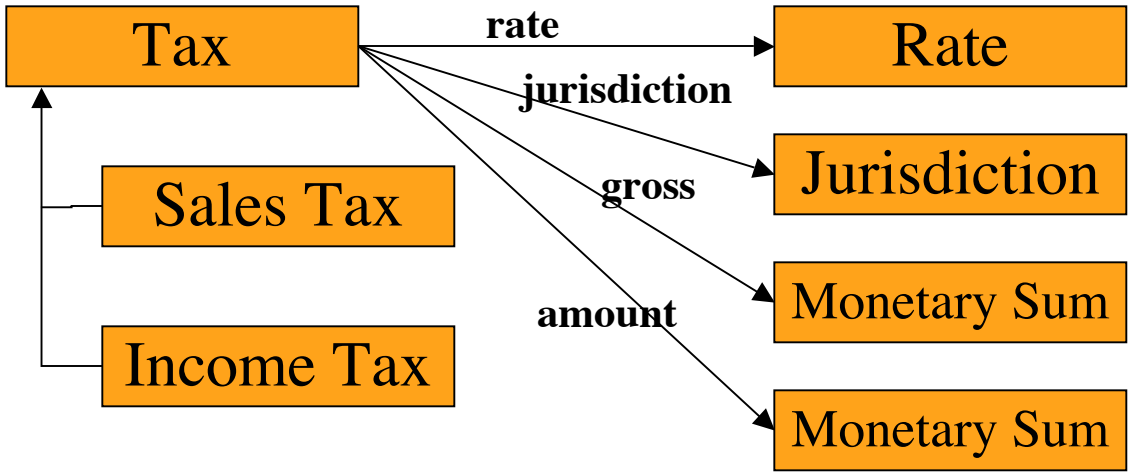
Semantic Mapping

- The problem:
  - Poorly defined semantics
  - Shared syntax, different semantics
  - Shared semantics, different syntax



- Semantic mapping provides a bridge
  - Types
  - Properties
  - Business Rules

Mapping Example

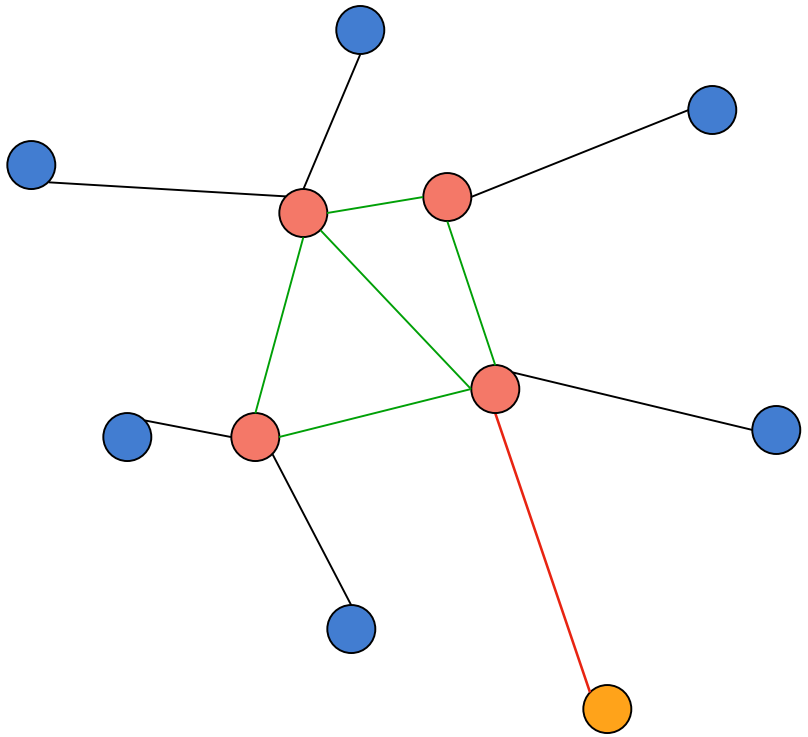


- Rate: % or multiplier
- Jurisdiction: set of zip codes
- 9-digit zip is subtype of 5-digit zip
- Monetary sum has quantity and currency

Example from Semantic Discovery for Web Service, WSJ, Vol 3, Issue 4

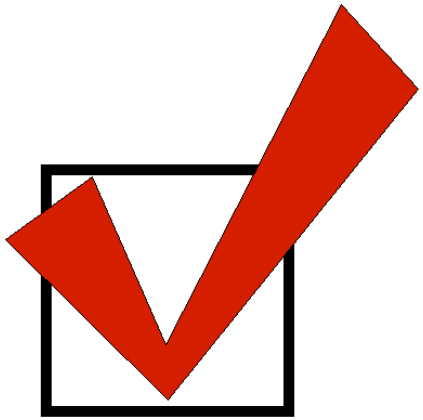
Reliability

- Multiple switches, routers and proxies
- Self-organizing
- Name-space aware



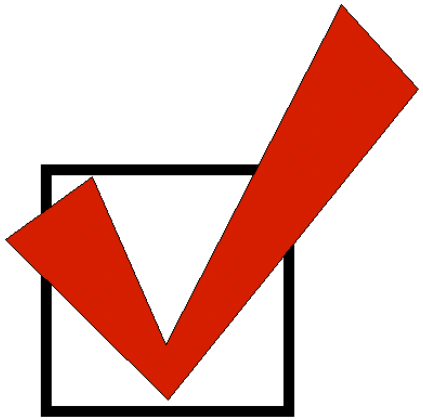
Advantages

- Security
- Reliability
- Availability
- Scalability
- Interoperability
- Interactivity



Advantages

- Separation of concerns
  - Policy-based configuration
  - Business logic
- Reliability
  - Professionals develop difficult code
- Access
  - You don't own it



Content Pipelines

- RSS, Exchange folders, etc. as sources
- Route through web services for annotation
  - Google
  - Amazon
  - Weather.com
  - Dictionary.com

- Deliver personalized, annotated information

Clemens Vasters, http://radio.weblogs.com/0108971/2003/04/11.html - a154

Digital Identity

- Why digital identity matters
- Driver's license
  - Identity
  - Authentication
  - Authorization
- Three phases (Andre Durand):
  1. Their identity – marketing
  2. Our identity – corporate
  3. My identity – individual

Resources

- Middleware Conference
- ACM Transactions on Internet Technology
- Journal of Computer Resource Management
- IEEE Internet Computing
- *and others…*

Contact Information

Contact me

- phil@windley.com
- www.windley.com

Questions?

Auxilliary Slide

- The slides past here are for reference only

Design Principles

1. *Every* data element and collection is a resource
2. *Every* resource should have a URI
3. Cool URI's *don't* change
4. Preserve the structure of data until the *last possible moment* (*i.e.* return XML)
5. Make XML Schemas available *online* for your XML
6. Data queries on existing resources should be done with a GET
7. Use POST to create new resources

Design Principles (cont)

8. Document your service API using WSDL, WRDL, or some other standard
9. Advertise the presence of the data using WSIL
10. Adhere to data standards such as RSS where available
11. Use Metadata (RDF) for XML
12. Use HTTP authentication as much as possible
13. Make data available in multiple flavors