Middleware for Distributed Systems

Old pals and the new kid(s) on the block

Walter Kriha

What is Middleware?

software that helps two separate systems **communicate seamlessly**.
(www.knownow.com/middleware/lexicon.html)

In a strict sense middleware is **transport software** that is used to move information from one program to one or more other programs, **shielding the developer** from dependencies on communication protocols, operating systems and hardware platform ("**plumbing**")
(www.talarian.com)

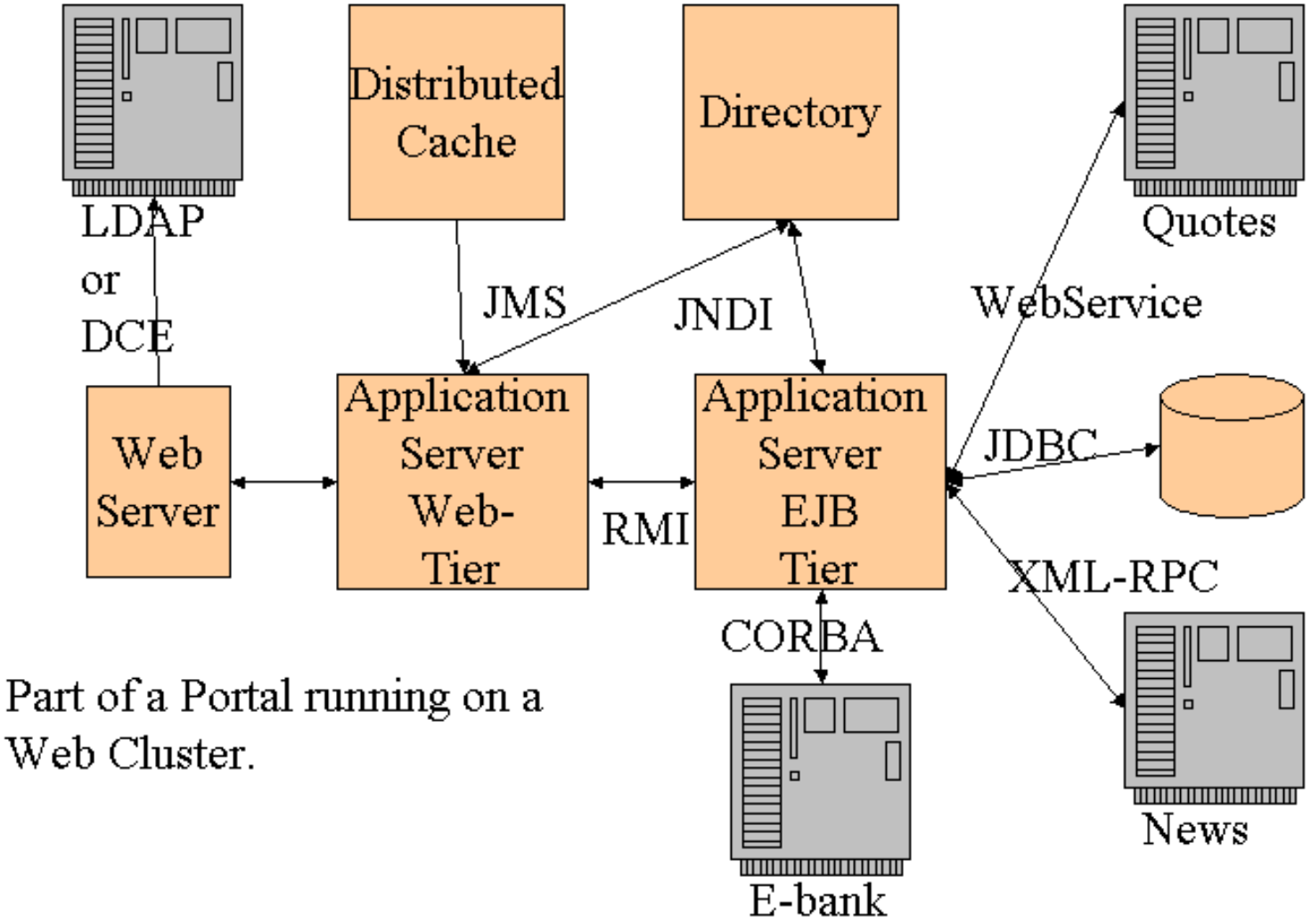
The Transparency Dogma

- Middleware is supposed to hide remote-ness and concurrency by hiding distribution behind local programming language constructs

Critique: Jim Waldo, SUN

Full transparency is impossible and the price is too high

Where do we find Middleware?



Part of a Portal running on a Web Cluster.

Classification

- Remote Procedure Calls (RPCs)
- Object Request Brokers (ORBs)
- Message Oriented Middleware (MOMs)
- Others

The “ilities”

- Reliability
- Availability
- Security
- Scalability
- Quality
- Performance
- Maintainability

Before using a specific middleware, always make sure that the “ilities” aka non-functional requirements are met. Middleware almost always differs implementation quality between vendors.

Real-World Problems

- Skills/Understanding: Best practice patterns?
- Single-Point-Of-Failures: replication, load-balancing etc.
- Tooling: generators, deployment tools
- “Brittle-ness” if interfaces change (Compiler “illusion”)

RPC type Middleware

- E.g. Sun-RPC, OSF DCE
- Main idea: distribute functions, use concurrent processing
- On top of it: Distributed Directory, File system, Security (cells, principals)
- XML-RPC over http (www.userland.com)

Layer foundations: UUIDs, value vs. reference, marshaling, versioning etc.

Distributed Objects

CORBA

- Object Request Broker
- Multi-language support (platform independence)
- Interface Definition language
- Wire Protocoll: IIOP, GIOP

RMI

- Java only (e.g.Introspection used)
- Lightweight method call semantics
- Java Implementations
- Wire Protocoll now mostly: RMI over IIOP

Both try to preserve object semantics. Interface/Implementation separation

Distributed Components

- Objects are too granular: performance and maintenance problems
- Programmers need more help: separation of concerns and context

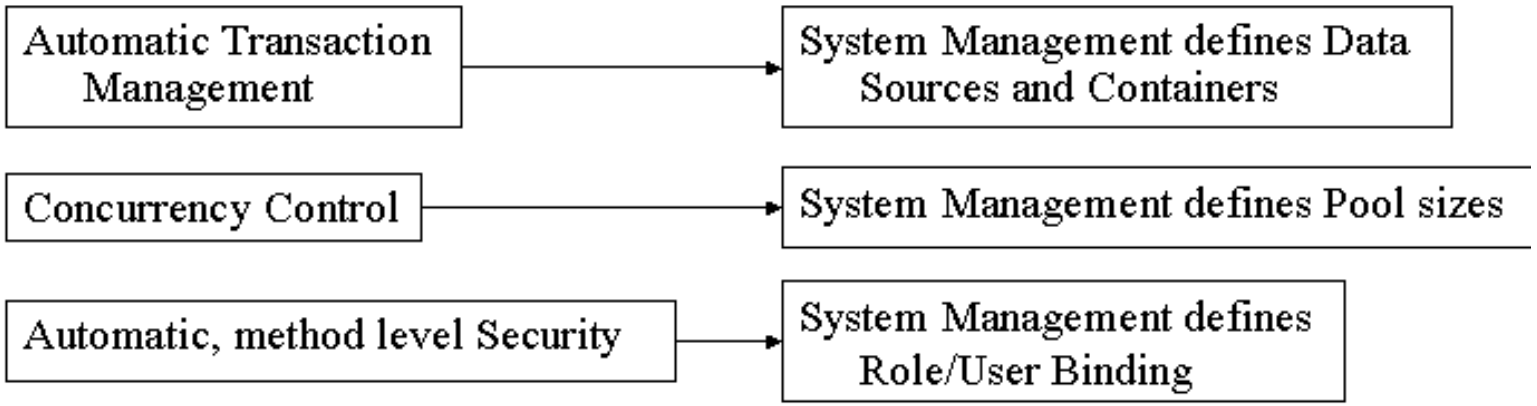
Solutions:

- Enterprise Java Beans
- CORBA Components
- COM+

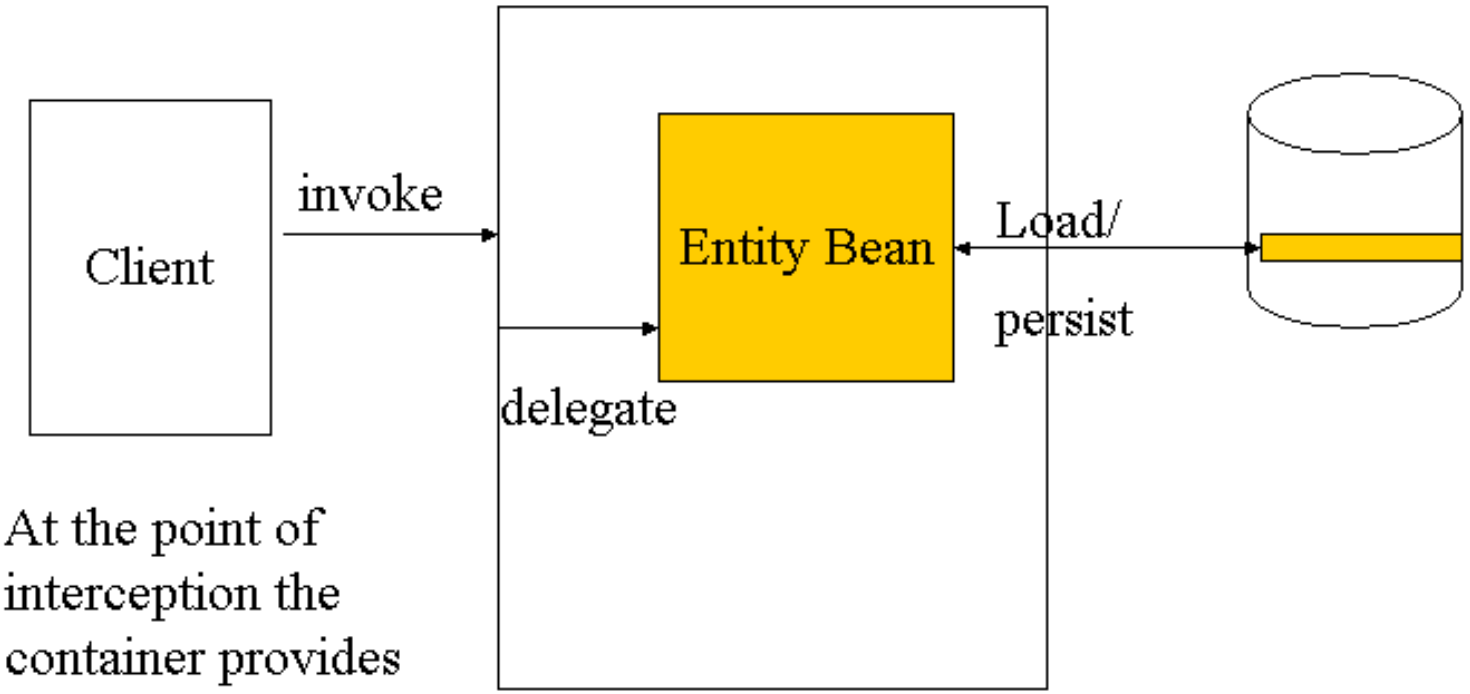
Example: Enterprise Java Beans

EJB Framework (Separation of **concerns**):

Deployment (Separation of **context**):



EJB Container

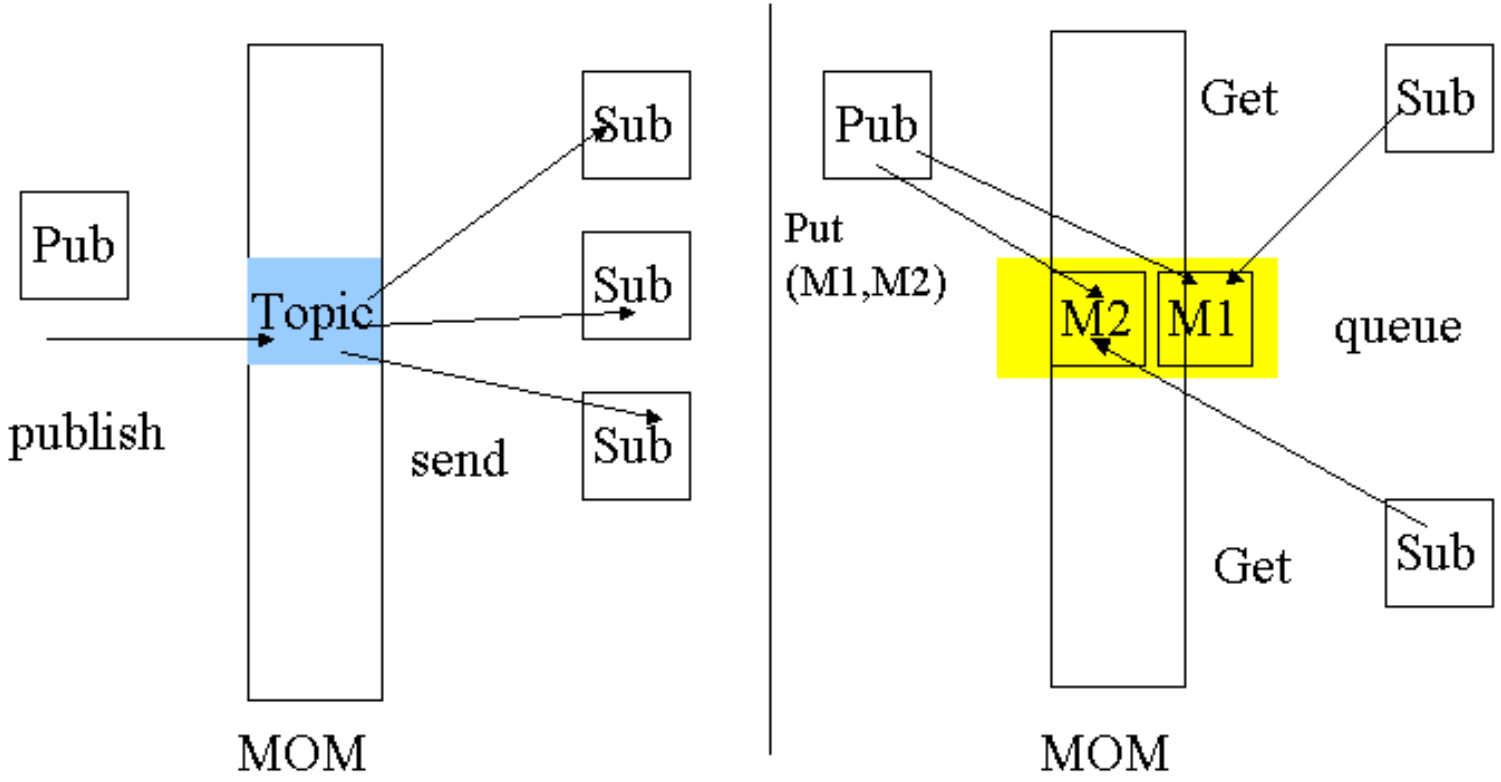


At the point of interception the container provides the following services to the bean:

Resource management, life-cycle, state-management, transactions, security, persistence

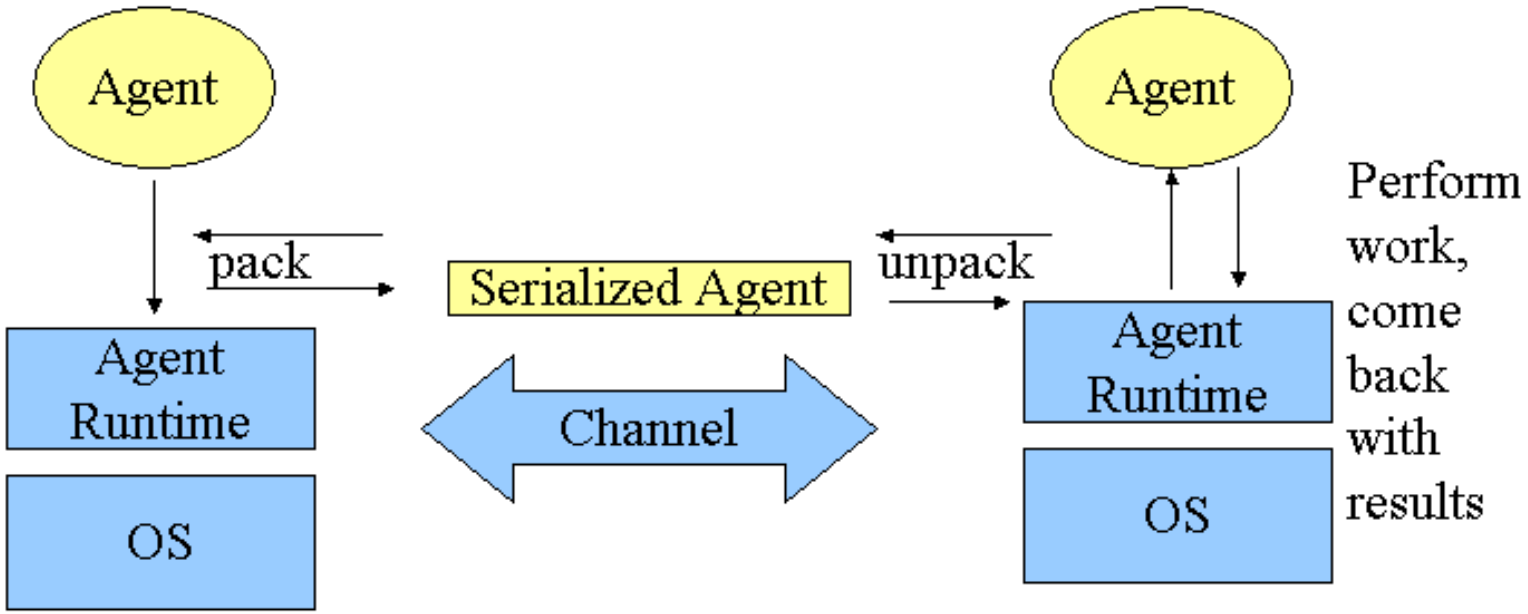
Distributed Messages (MOM)

Asynchronous, loosely-coupled (fault tolerant), persistent messages with either publish/subscribe (topics) or queuing semantics. Scales well. Delivery guarantees differ.

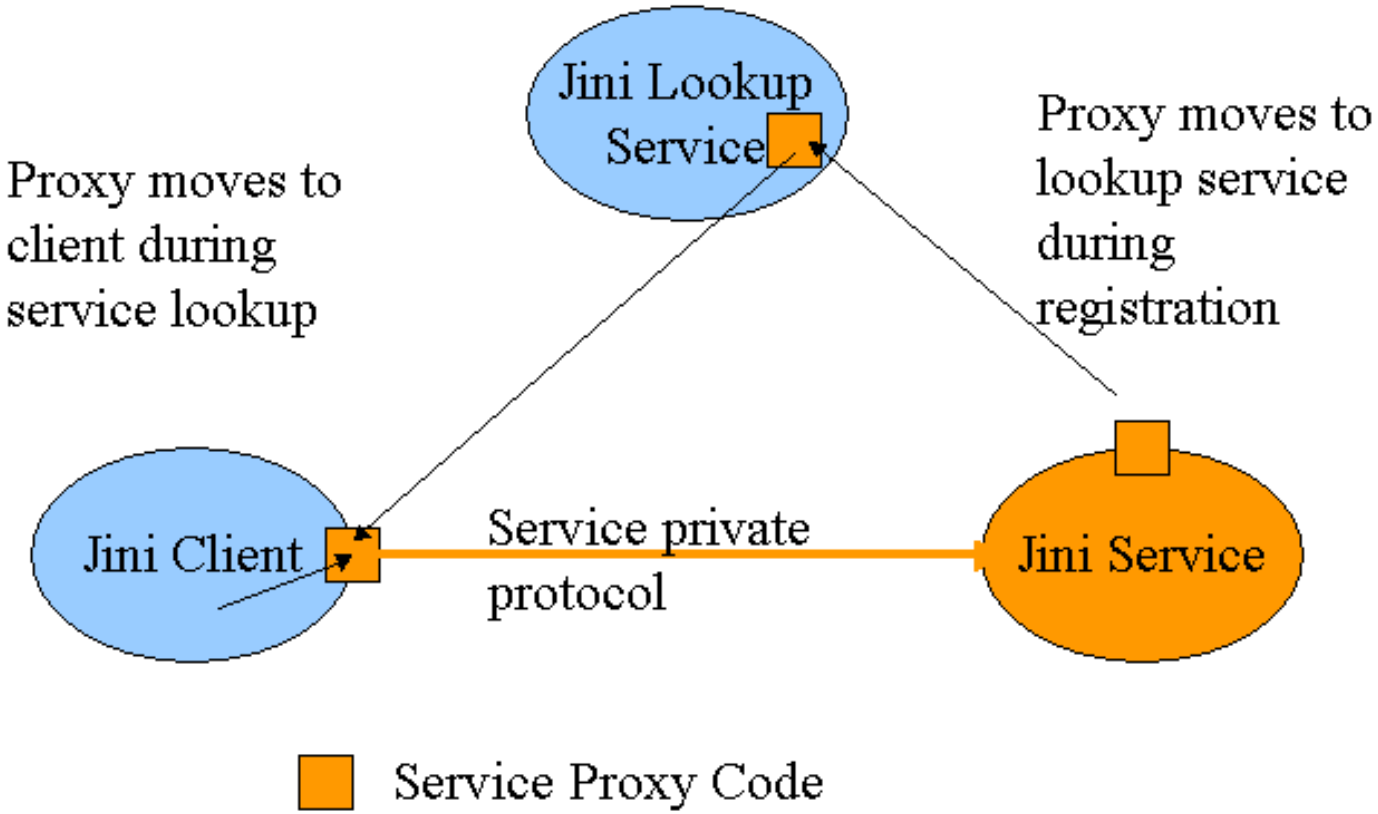


Distributed Code I (Agents, Aglets)

The Problem: who wants a new runtime system?

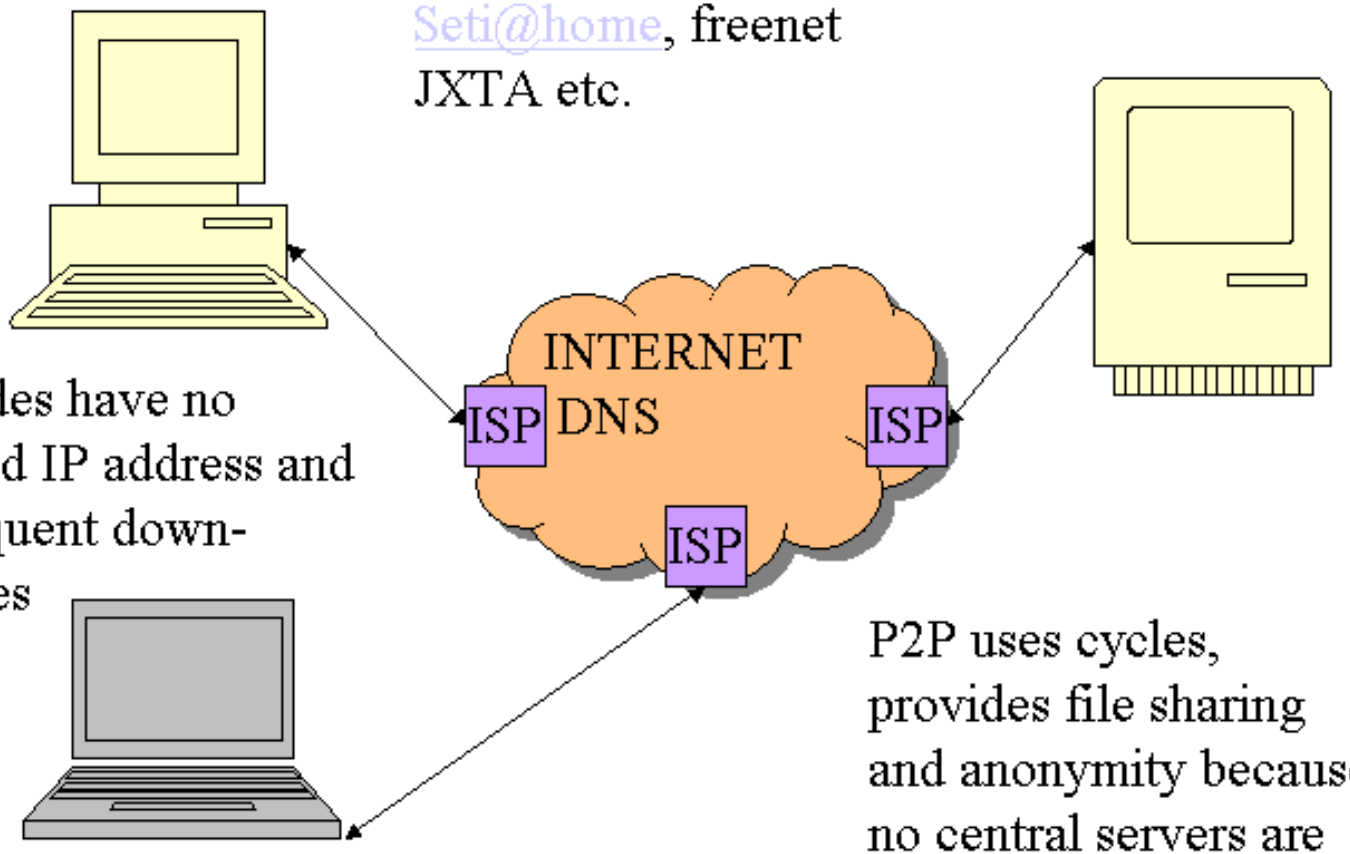


Distributed Code II (Jini) – The End of Protocols?



Peer 2 Peer

Seti@home, freenet
JXTA etc.



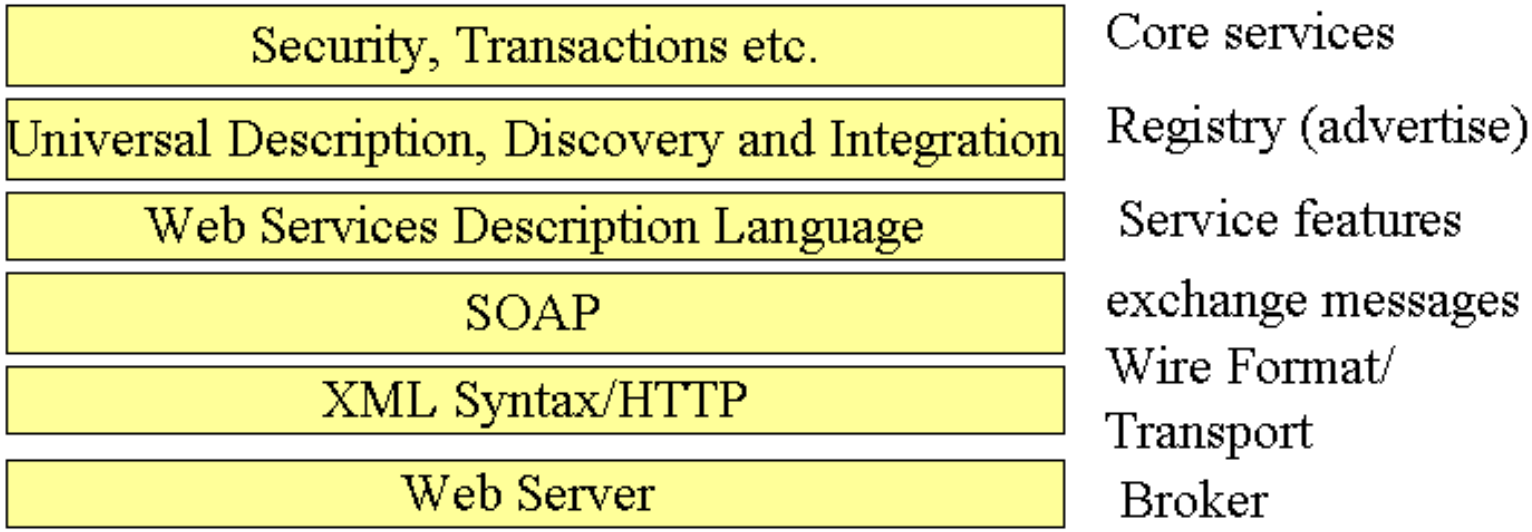
Nodes have no fixed IP address and frequent down-times

P2P uses cycles, provides file sharing and anonymity because no central servers are used

Problems: How do you version files? Overhead?

WebServices

Promises de-coupling of service provider and requester, document interfaces, machine-to-machine communication and ease of use compared to distributed objects. No way to define QOS or conversation yet.



Service Granularity? Application, Component, Object or Request?

Use your “de-hyper” generously!

Others

- Parallel Processing: PVM, MPI (e.g. for Linux Beowulf cluster)
- Wireless:

  Bluetooth
- System Management:

  Jiro/FMA
- Linda Tuple Spaces (Javaspaces, Tspaces)
- Group Computing (virtual synchrony): Horus, iBus

Resources

- Jim Waldo, End of Protocols
- Marco Boger,Java in verteilten Systemen
- Ken Birman, Building secure and reliable Network Application
- ObjectSpectrum 7/2001, WebServices
- Java Magazine 7/2001, Java Message Service
- www.theserverside.com on EJBs
- Clay Shirky, What is P2P and what Isn't (www.openp2p.com)
- S.Tai, I.Rouvellou, Strategies for Integrating Messaging and Distributed Object Transactions