



Grid Computing:

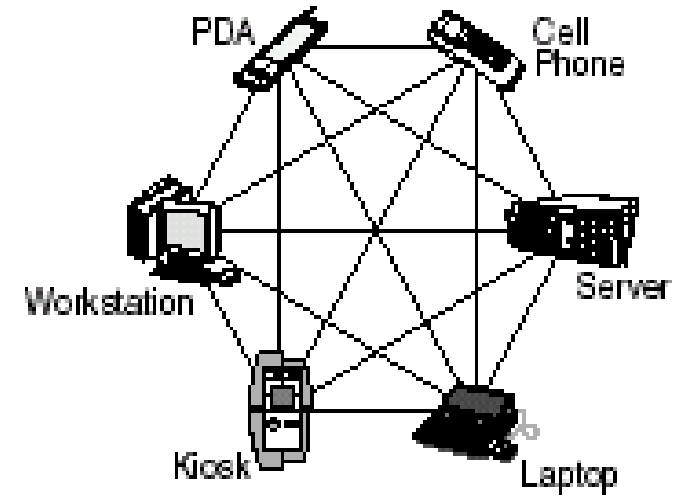
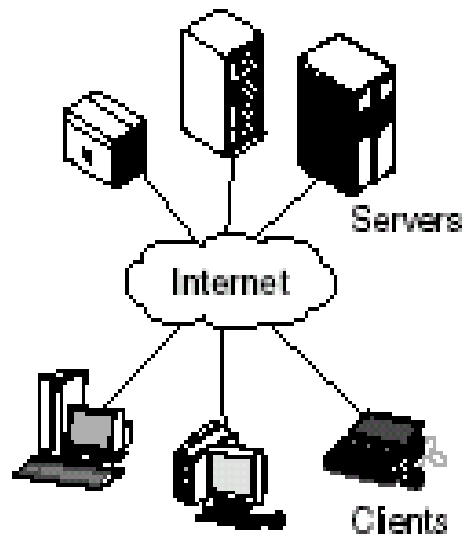
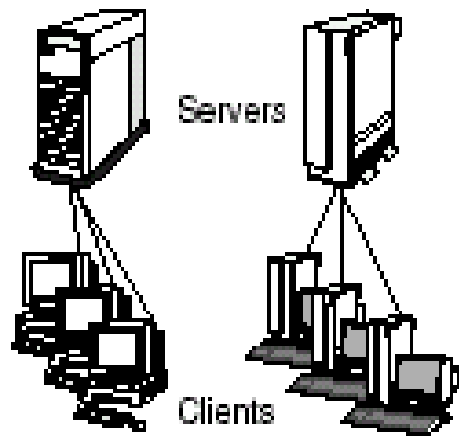
SGE, Jxta and Jini

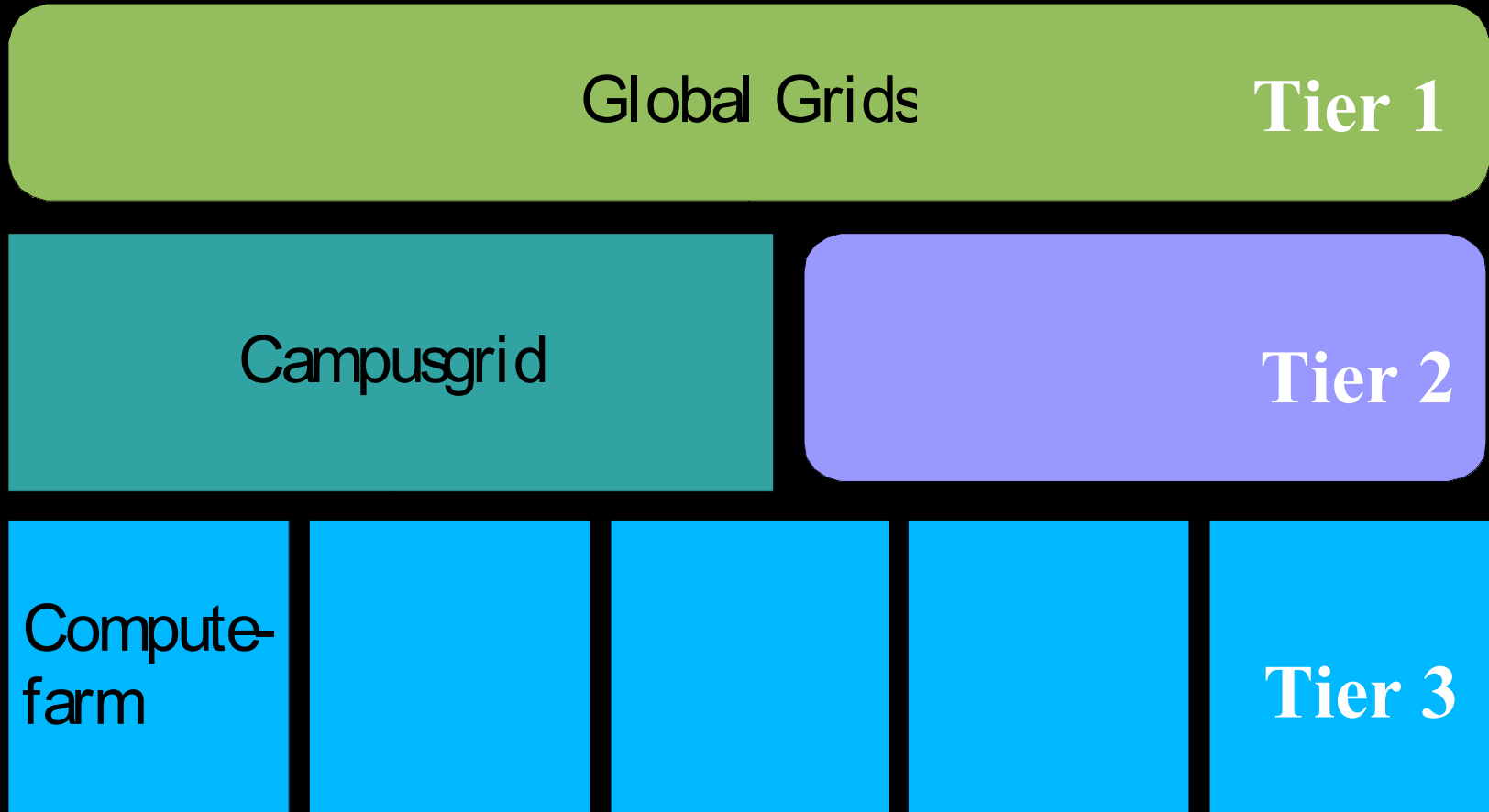
Dr Simon See

Director High Performance Computing Technology

Adjunct Associate Professor , Nanyang Technological

University





Grid Type Complexity

Department Grid

Cluster load balancing

Campus Grid

Heterogeneity, HPC, priorities, ...

Research Grid

Internet, multi-tier, collaboration, ...

Enterprise Grid

Security, policies, RAS, ...

Access Grid

Imersive collaboration, training, ...

Public Grid

Resource discovery, reservation, ...

Broker Grid

Level of service, economy,
accounting

Other Grids:

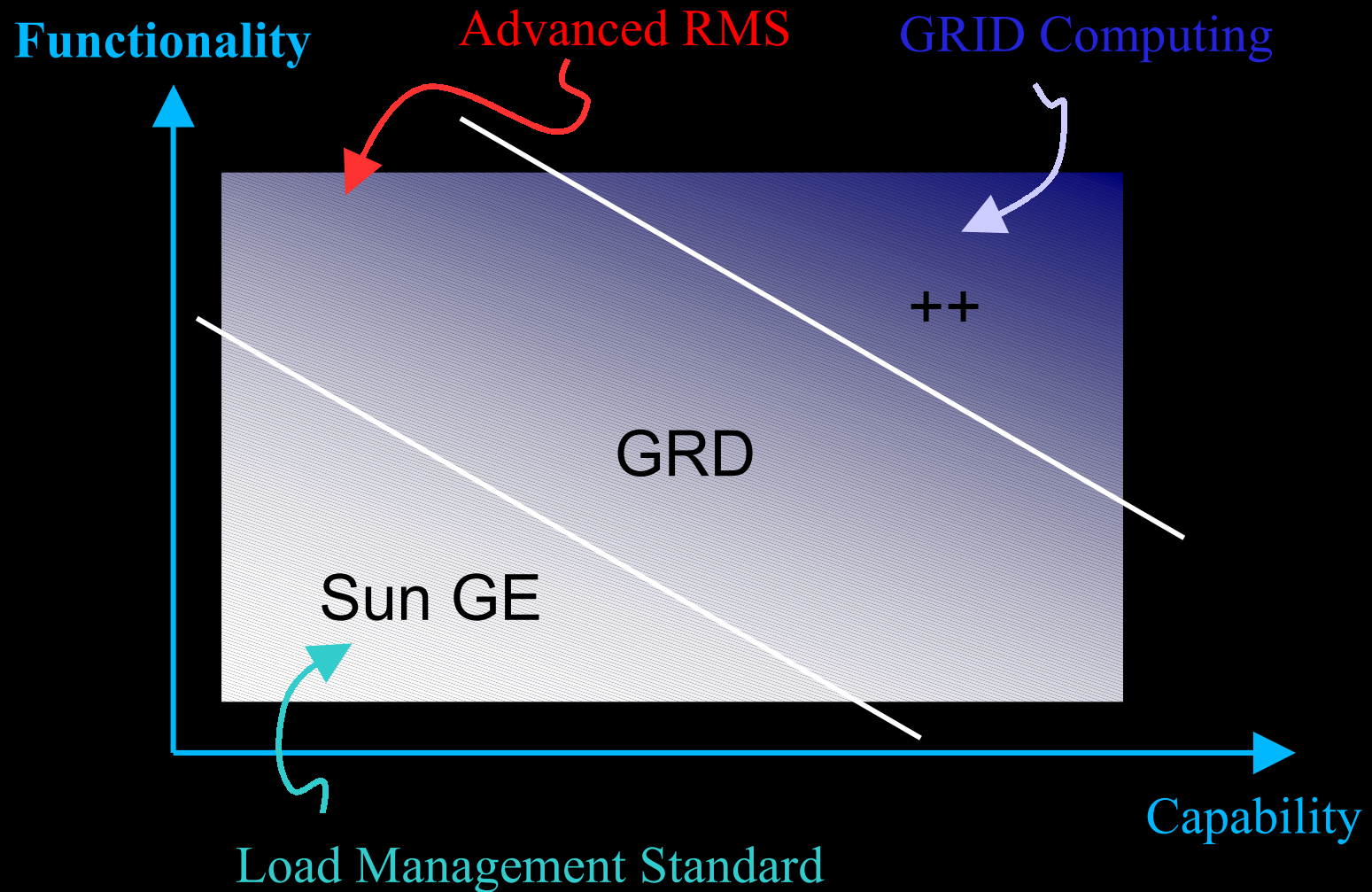
Application / Storage / Network Service Providers, Peer-to-Peer, Collaboration Grid, Data Grid, HealthGrid, BioGrid, Entertainment Grid, Information Grid, ...

Grid Computing Challenges

Extends cluster computing toward LANs and WANs for parallel and distributed computing. Problems to be solved:

- bandwidth and latency
- reliability, availability, fault tolerance
- wide area parallel processing
- interoperability
- heterogeneity
- a single global name space
- protection, security, authentication
- efficient scheduling
- and comprehensive resources management

Distributed Resource Management Product Space



Sun Distributed Resource Management

- Load balancing maximizes resource utilization
- Transparent job submission & machine selection
- Monitoring and accounting
 - ==> SGE Sun Grid Engine, free, open source
- Guaranteeing required resources
- Full control over resource utilization
- Fair and share based resource usage
- Implementation of management policies
 - ==> SGE/GRD Global Resource Director



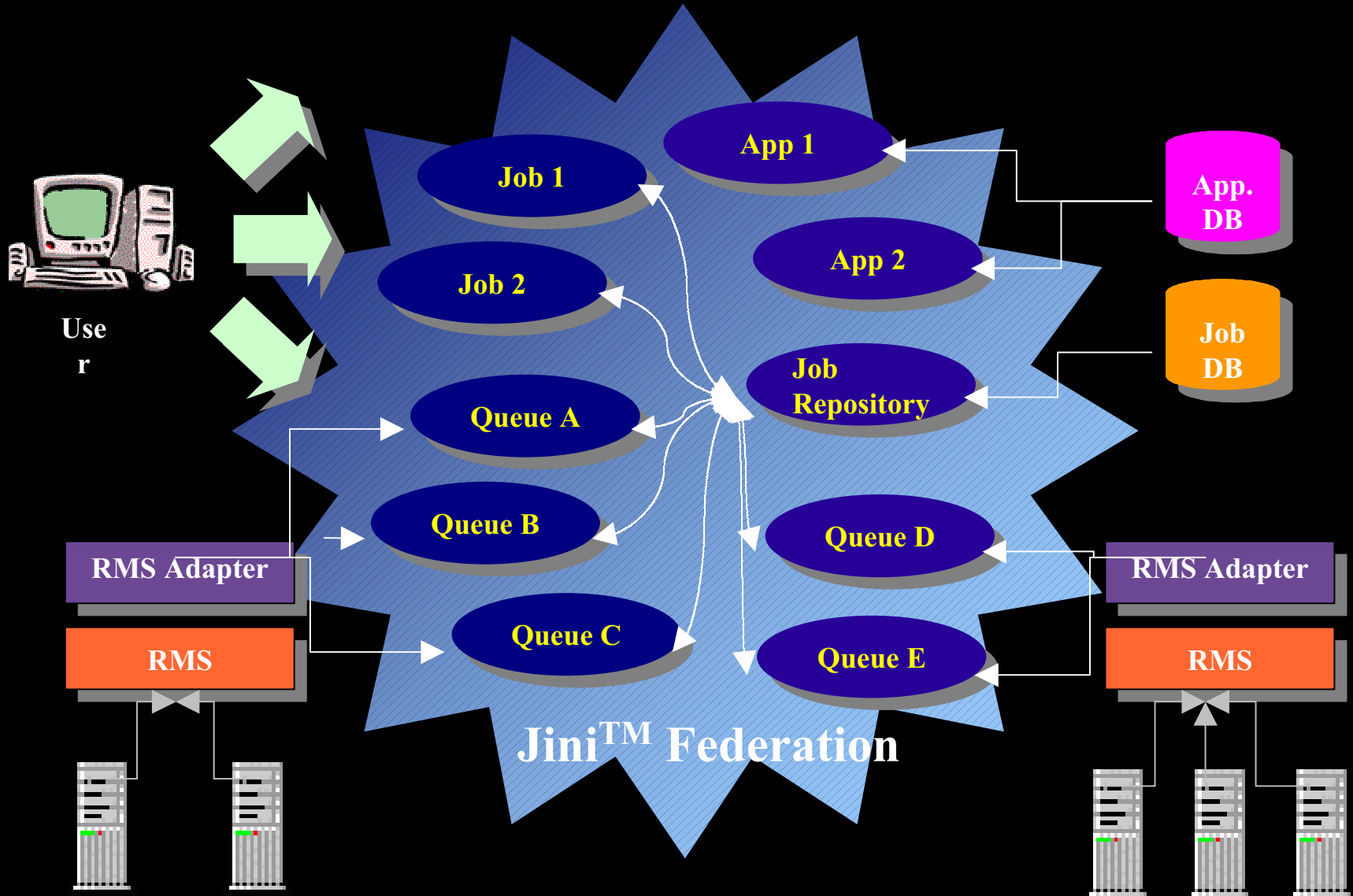
nth
ready

- ∇ Infrastructure for providing services in a network
 - Key attributes
 - Distributed
 - Location independence
 - Spontaneous
 - Supports dynamic service attributes
 - Reacts as services join and leave
 - Robust
 - Handles partial failures

- Key protocols
 - Discovery
 - Locate a (set of) Jini lookup service(s)
 - Join
 - Register your service with discovered lookup services
 - Lookup
 - Find and download a desired service object (proxy) by:
 - Java (interface) type
 - Service attributes
 - ServiceID
 - Client codes to service interface (via service proxy)

- Sun Microsystems
- Swiss Center for Scientific Computing (CSCS)
- Research! Not a current or future Sun product
 - Technology Exploration: Java™ in HPC
 - Domain: Distributed Resource Management
- Technologies employed
 - Java™, Jini™, JavaSpaces™
 - Sun™ Grid Engine

- Resources
 - Applications
 - Existing, unmodified HPC codes
 - Compute cycles
 - Local resource manager queues
- Geographically Distributed
 - Intranet domain
- Integrated
 - Application selection & execution framework





- **Search** the entire Web — and all its connected devices — for information or resources (deeper Web)
- **Share** compute services, files, music, etc. with anyone, regardless where they're located or what type of device they're using (pervasive networking)
- **Store** files and data anywhere on the network, not just to local hard drives (distributed networking)

Find it...Get it...Use it

Jun 1999: Bill Joy's Mantra "Pervasive, Simple & Secure "

Jan 2000: First Design (Protocols, XML, Messaging, Space)

Jun 2000: First Demo Prototype (Palm)

Nov 2001: New Design (Protocols, XML, Peers, Pipes, PeerGroups)

Committers (5 Sun)

Apr 2001: J2SE Reference Implementation on www.jxta.org

Jun 2001: 50,000 Downloads!

Start Platform Community Projects

Committers (14 Sun, 8 Non-Sun)

New V1.0 Protocols Spec (Revision 1.1)

- Java-specific
- A networking API
- A new way to browse the web
- Another file-sharing or chat tool
- Jini repackaged
- A Sun product

- Peer-to-peer platform
- Platform, language, and transport agnostic
- A way for peers to find each other and share information
- A set of protocols that enable P2P applications and services
- A spec and a couple of reference implementations

1. It's All about Protocols (platform & transport & Service agnostics)
2. Pipes (XML Messaging – Structured Data – Pipeline)
3. Core Principles
 1. Provide Essential Building Blocks (Pipes, PeerGroups, Monitoring, Security)
 2. 100 % Pure P2P (No centralized services DNS)
 3. Information flows towards the edges
 4. Performance improve as system ages

- Peer Groups
 - Manages identity of peers and peer groups
 - Join, Leave, Create
- Peer Pipes
 - Send any content (code, data, etc.) to peers
 - Messages in XML format
 - Secure pipes supported
- Peer Monitoring
 - Check peers' status remotely, control remote peer behavior
 - Meter peers' usage, set usage or bandwidth limits
- Security
 - Pluggable architecture supports a variety of standard security solutions
 - Control group membership
 - Encrypt data sent over pipes



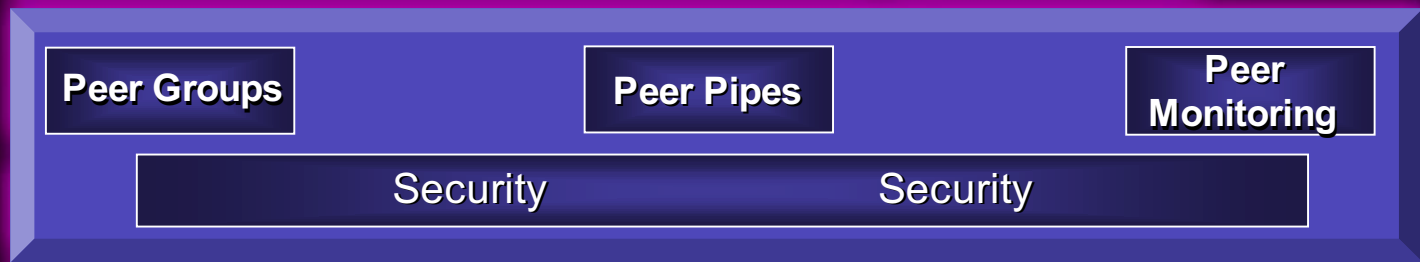
JXTA Applications



JXTA Services



JXTA



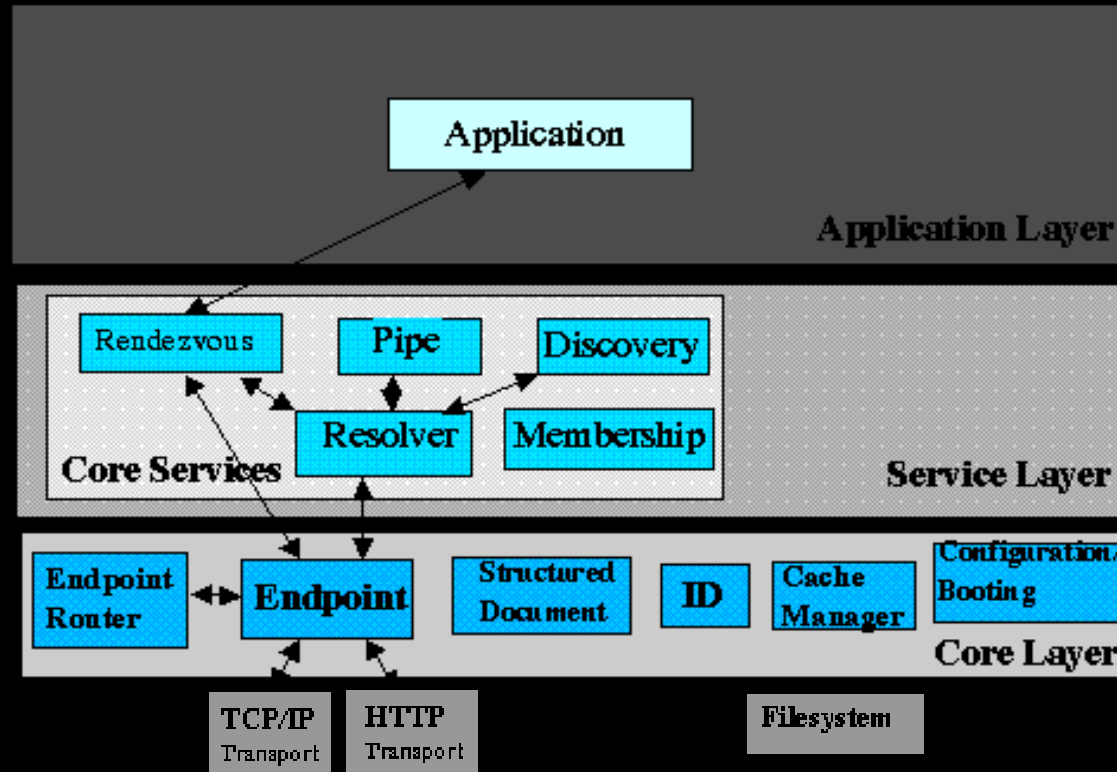
Any Peer on the Expanded Web



- ∇ Define a domain of trust
- ∇ Define a set of common services (protocols)
- ∇ Define a scope of peer interaction
- ∇ Define an administration and monitoring domain

- ∇ Virtual Connection Channel between Peers
- ∇ Dynamic Pipe Endpoint Binding
- ∇ Structured Data Messages
- ∇ Pipes:
 - ∇ point-to-point
 - ∇ Propagate (1-to-Many)

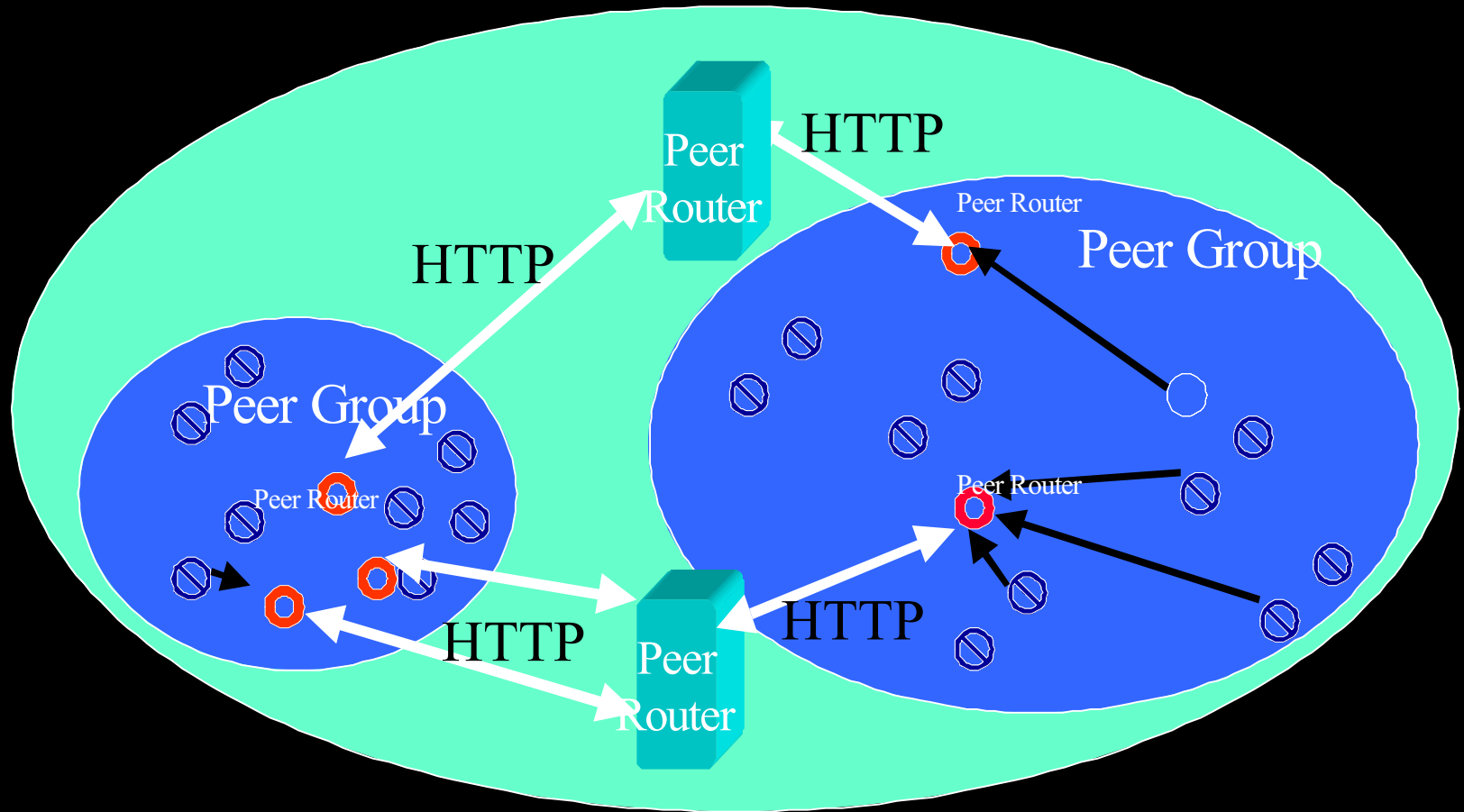
- ∇ Membership
 - Peer Membership Protocol
- ∇ Discovery
 - Peer Discovery Protocol
 - Peer Resolver Protocol
- ∇ Pipe
 - Pipe Binding Protocol
 - Peer Endpoint Protocol
- ∇ Monitoring
 - Peer Information Protocol



- ∇ Leverage surrounding peers, and groups
- ∇ Meetings points for far away peers, and groups
- ∇ No centralized naming
- ∇ Asynchronous protocol
- ∇ Pre-defined meeting points (bootstrapping)
- ∇ Reverse Discovery
- ∇ Learn about other peer's abilities

- ∇ Route messages between multi-hops peers
- ∇ Every peer can be a router
- ∇ HTTP Routing Server
 - ∇ Traverse Firewalls
 - ∇ NAT Support
 - ∇ Message Gate
 - ∇ ways (TTL)

JXTA Peer Routers



- Consumer file sharing (like Gnutella)
- Distributed processing (like SETI@home)
- Distributed data storage
- Communications (like IM)

Both:

- Are distributed P2P technologies
- Are platform-independent
- Allow the creation of spontaneous, self-healing networks
- Define how to find other peers, but not how to communicate with them

Project JXTA is:

- Protocol spec
- Language-independent
- Inherently secure
- Uses published advertisements
- Concerned with natural grouping of peer services
- Provides a way for peers to monitor each other

Jini is:

- API
- Java only
- Security now being added
- Uses service registries
- Concerned with dynamic service delivery
- Requires no prior knowledge of services

- Project JXTA's Technology open source available for download now (based on the Apache open source model)
- Developer tools, specifications, and documentation available at <http://www.jxta.org>
- Applications using Project JXTA's technologies in development
- Multiple implementations in development (Java, C, etc.)

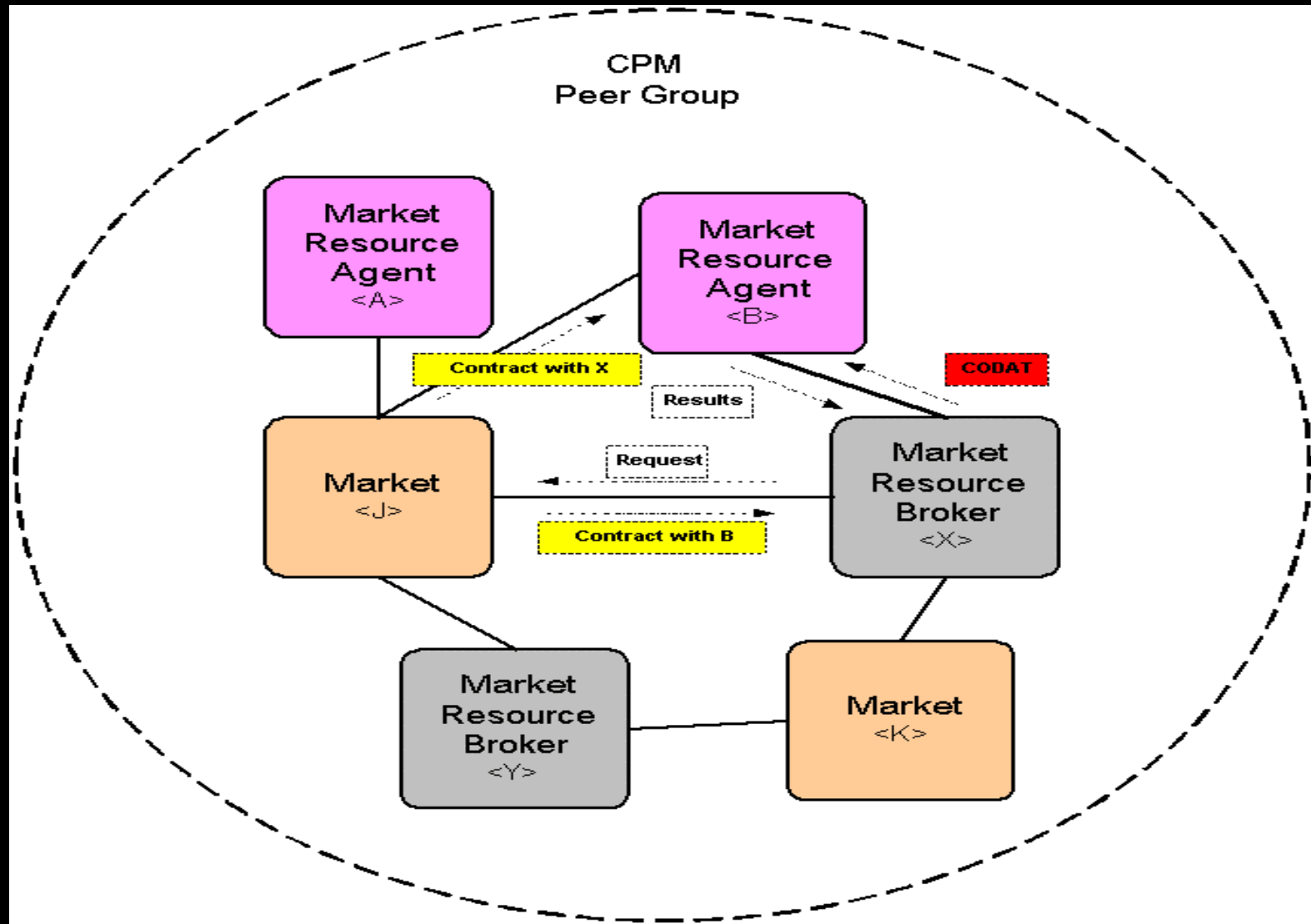
- jxta.org
 - All source, projects, docs, examples
- Apache-style software license
 - No barriers to getting started
- Meritocracy
 - The more you've done, the more you are allowed to do



- JXME – JXTA binding for J2ME
- JXTA–Wire – Many-to-many pipes
- iPeers – AI for JXTA
- P2P Email – Decentralized email
- Instant P2P – JXTA instant messenger
- CMS – Content management system
- JXTASpaces – JavaSpaces for JXTA

Base on pricing mechanism

- Decentralized
- Markets
- Market Resource Agents (Providers)
 - Act on behave of providers ex. end-users, labs
- Market Resource Brokers (Consumers)
 - Act on behave of consumers ex. scientists, enterprises.



Join us!!!

<http://compute-power-market.jxta.org/servlets/ProjectMemberList>

The screenshot shows a Microsoft Internet Explorer browser window displaying the Project JXTA member list page. The browser's address bar shows the URL <http://compute-power-market.jxta.org/servlets/ProjectMemberList>. The page content includes a navigation menu on the left, a main header for Project JXTA, and a table of members with their roles and assigned issues.

Project JXTA

Project Info
[Background](#)
[News](#)
[Downloads](#)
[FAQ](#)
[Help](#)

Developer Resources
[Getting started](#)
[View projects](#)
[Join Project JXTA](#)
[Login](#)
[Mailing lists](#)
[Report bugs](#)

Documentation
[Project JXTA docs](#)
[Tool docs](#)
[License](#)
[Governance](#)

Weekly Stats
11-AUG-2001
Members 5221
Posts 197
CVS Commits 407

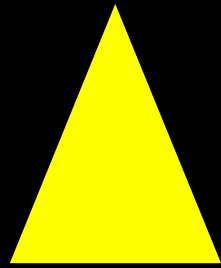
Project [Project Home](#) | [News](#) | [Members](#)
Resources: [Mailing Lists](#) | [Source Code](#) | [Issue Tracking](#)

Members: compute-power-market **If you Register, you could join this Project**

| User | Real Name | Role(s) | Assigned Issues |
|-------------|----------------|-------------------------|-----------------------------|
| buyya | Rajkumar Buyya | Committer Project Owner | View Issues |
| cheesun | Liew Chee Sun | Contributor | View Issues |
| fazilah | Fazilah Haron | Committer Project Owner | View Issues |
| feefoon | Kong Fee Foon | Contributor | View Issues |
| hychan | Chan Huah Yong | Committer Project Owner | View Issues |
| jshuang | hjdfhd | Committer | View Issues |
| krakesh | Rakesh Kumar | Contributor | View Issues |
| lolaella | gary timms | Contributor | View Issues |
| madhavmj | Mahesh Madhav | Committer | View Issues |
| mysky | MySKY Ong | Contributor | View Issues |
| OmerRana | Omer F. Rana | Contributor | View Issues |
| PHinker | Paul Hinker | Contributor | View Issues |
| rahulvaidya | rahul vaidya | Contributor | View Issues |
| siewsee | siew see | Contributor | View Issues |
| skymars | Gobinathan | Contributor | View Issues |

Simulation Space

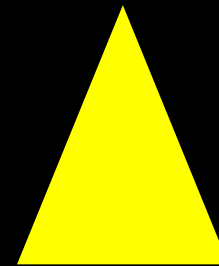
Client-Simulation
Administration



CA Daemon

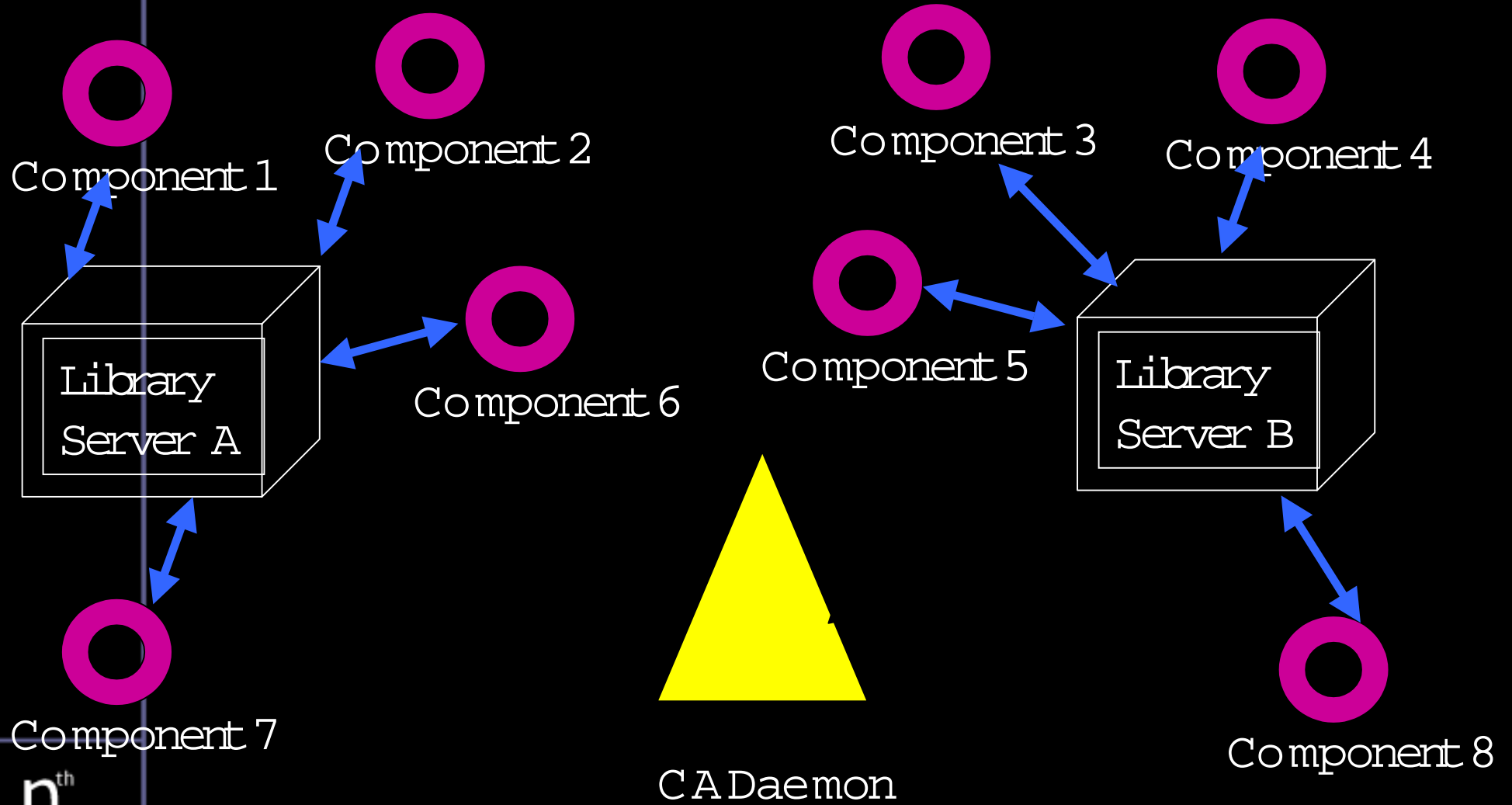
HPC Space

Complex Computation



Broker

Both spaces require a point of reference which other entities within the network can relate to.



- <http://www.onjava.com/pub/a/onjava/2001/04/25/jxta.html>
- http://www.clip2.com/jxta_chat.html
- <http://www.jxta.org>
- <http://www.jxta.org/project/www/getstart.html>
- <http://onestop.eng/javatech/JXTA>



Simon See

Simon.see@sun.com