Jini: Decentralized Control in a Federated Network

Charles Lamb
Sun Microsystems, Inc.
Goals of Jini

- Network plug-and-work
- Erase the hardware/software distinction
- Allow spontaneous networking
- Enable a service-based architecture

Above all, to provide the simplest system possible
Java™ Technology Makes It Work

- Java VM = Homogeneous Network
- Portable object code
- Safety through verification
- A single type system for the network and the implementation
- Dynamic environment
Structure of Jini

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>Programming Model</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java VM</td>
<td>Beans</td>
<td>Servlets</td>
</tr>
<tr>
<td>Java RMI</td>
<td>Swing</td>
<td>EJB</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>JNDI</td>
</tr>
<tr>
<td>Discovery</td>
<td>Leasing</td>
<td>JavaSpaces</td>
</tr>
<tr>
<td>Lookup Service</td>
<td>Transactions</td>
<td>Transaction Manager</td>
</tr>
<tr>
<td></td>
<td>Distributed Events</td>
<td>...</td>
</tr>
</tbody>
</table>
Spontaneous Networking

• Services join a federation by discovering a Lookup Service

• Registering with a Lookup Service = advertising your service

• Clients find services by Java type

• Code moves from the service to the Lookup Service to the client

• Code needed to use the service is dynamically loaded when needed by the client
Registering
Finding a Service

Network Client

Service Request

Java Interface Code

Lookup Service

Java Interface Code
Using the Service
Finding Services

- Lookup is done by Java type; returning a subtype is ok
- Other attributes can be associated with a proxy
- Attribute matching is exact only
- Event notifications can be generated when contents of the Lookup Service changes
Leaving a Jini Network

- Registration with the Lookup Service can be cancelled
- Registrations are leased; if the lease is not renewed, registration is dropped
- Service Proxies can also be leased
- No human intervention is required for joining or leaving
Federation vs. Central Control

- Jini is *not* an operating system
- Jini sets the minimal set of interaction rules
- Services inject themselves into network clients
- All else is left to the local authorities
- Jini is designed to work with, not replace, operating systems
Service Architectures

- No distinction between hardware and software
- Services can be local, remote, or a combination
- Object-orientation centers on what, not how
- Jini (and Java RMI) extends this to the network
Jini and Legacy

• Legacy services fit well into the framework

• Legacy languages (C, C++, COBOL) fit behind Java wrapping

• Jini moves the cross-language membrane from the network to the address-space
The Future of Jini

- Network federation replaces the operating system
- Polymorphism abstracts over network protocols
- Networks become ubiquitous and invisible
- Services replace applications
But We Still Need

• More bandwidth

• Recognition of the goal of simplicity

• Interfaces that enable user understanding

• Identity in the network world