The Jini™ Architecture

Bruno Souza
Java Technologist, Sun Microsystems
Why Jini™ Technology

- Network plug and work
- Enables a service-based architecture
- Spontaneous networking
- Erase the distinction between hardware and software
- A distributed computing infrastructure to making writing distributed programs easier
<table>
<thead>
<tr>
<th>Feature</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java virtual machine</td>
<td>Homogeneous network</td>
</tr>
<tr>
<td>Portable object code</td>
<td>Architecture independence</td>
</tr>
<tr>
<td>Downloadable code</td>
<td>Dynamic environment</td>
</tr>
<tr>
<td>Unified type system</td>
<td>No impedance mismatch</td>
</tr>
</tbody>
</table>
Simple, Reliable, and Flexible

- Based on the Java programming language
- Interfaces designed for robustness
- Services come and go without administration
- Federation, not central control
## Structure of Jini™ Technology

| Jini Services | • JavaSpaces™  
| • Transaction Managers  
| • Printing, Storage, Databases... |
|---|---|
| Jini Infrastructure | • Discovery  
| • Lookup Service |
| Jini Programming Model | • Leasing  
| • Distributed Events  
| • Transactions |
| Java 2 Platform | • Java RMI  
| • Java VM |
Registering a Service

Discovery Request

Discovery Response

Service Registration

Jini Service

Service Proxy Object

Lookup Service
Finding a Service

- Discovery Request
- Discovery Response
- Lookup Request
- Lookup Response
Using a Service

Jini Client

Service Proxy Object

Jini Service
Wire Protocol Independence

- Communication is between the service and its proxy
- Wire protocol is a private matter
  - Standard protocol
  - Custom, one time protocol
- Protocol can change
- RMI semantics core to functionality
Federations of Services

- **Minimal agreement**
  - Discovery protocol
  - Lookup interface
  - Interfaces define a service

- **Simple interaction patterns**
  - Time-based resource allocation
  - Distributed event delivery
  - Voting for consensus

- **Everything else is left to the participants**
More Than Devices

• Services can be hardware or software
• Dynamic lookup allows service introduction
  – When services become available
  – When services are updated
• Services are network-available components
Enabling Change Over Time

- Base interfaces can be extended
  - Add new functionality
  - Combine functionality
- Old clients can use base interface
- New clients can use extended interface
- Allows local evolution
Objects on the Network

- Polymorphism allows change
- Moving code is key
  - Separates what from how
  - Allows changing behavior
- Object oriented design patterns apply
- Merges agent technology and more standard approaches to distributed computing
What About Legacy?

• **We live in the real world**
  – Not every program is written in the Java programming language
  – Not every device can support a Java virtual machine

• **Most legacy code has yet to be written**
Objects to the Rescue

- **Java technology is only required**
  - For the proxy object
  - On the network

- **Legacy code can offer a proxy object**
  - Proxy is Java technology bytecodes
  - Speaks a private protocol to the legacy code

- **Legacy code can be wrapped**
  - Java programming language calling via Java Native Interface (JNI)
Network Proxies

- **Devices can proxy via another device**
  - Simple device provides service type, code location
  - More competent device provides Jini technology interaction
  - Proxy object can talk to either device

- **One network proxy for multiple devices**

- **Network proxy can also bridge networks**
  - TCP/IP to firewire/Bluetooth/IR/...
  - Jini technology-based system to HAVI
Interface Definition

- **The Jini™ Technology Community**
  - Deciding on base interfaces and classes
  - Driven, shepherded, or independent
  - Currently groups for printing and storage

- **Other standards groups**
  - If it has a Java programming language interface, it has a Jini service interface

- **De facto**
  - Let the market decide
Where Next

• First services coming soon
  – Some shown in the exhibition

• Security

• Scaling up
  – Most scale issues are user issues

• Build the community
For More Information

- http://sun.com/jini
- http://jini.org
- The Jini™ Technology Series
  - *The Jini™ Specification*
  - *JavaSpaces™: Principles, Patterns, and Practice*