

Jini™ Technology for Stuttgart JUG

Donnerstag 2/6/05

Onno Kluyt

Sr. Director, JCP & Jini Programs, Sun Microsystems

onno.kluyt@sun.com





What is it!?

- A service-based architecture for building networked systems that can adapt to change
 - Erases the hardware/software distinction
- A set of specifications
- **A programming model**
- Contributed implementations
 - Jini Starter Kit
- **A Community**
 - <http://jini.org>

Other Distributed Technologies

For context, not comparison



- Traditional remote procedure call (rpc)
- Java RMI (Java's rpc mechanism)
- Common Object Broker Architecture (CORBA)
- Enterprise JavaBeans (EJB)
- Servlets
- Web services



Common Characteristics

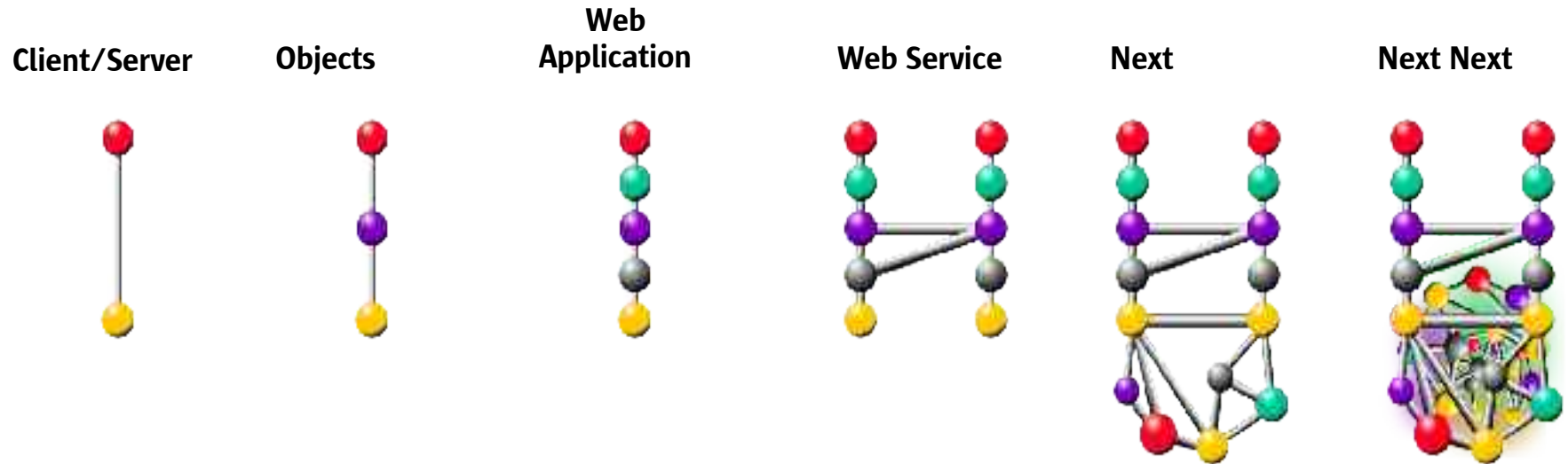
- Mechanism for advertising and finding the system itself
- Mechanism for advertising and finding services provided by the system
- Well defined model for interaction
 - Service identification
 - Acquisition and reclamation of resources
 - Communication
 - Coordination of actions
- All embodied in a **programming model**

The Axioms of Jini technology



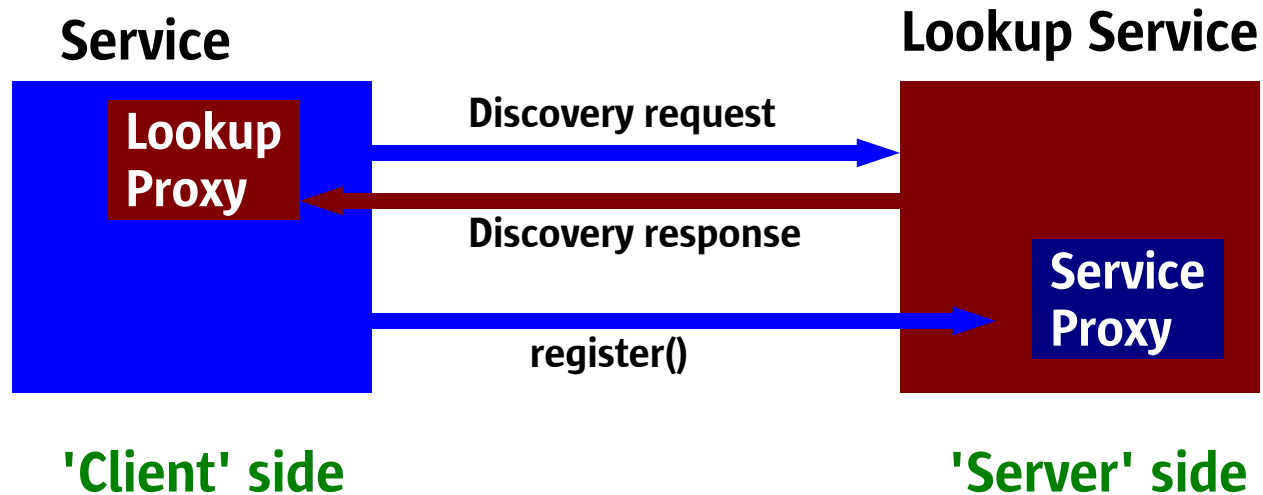
- **Axiom A:** fundamental differences exist between local and remote objects
 - These differences should be acknowledged and dealt with as part of the programming model
 - The network cannot be ignored
 - The fallacies of network computing
- **Axiom B:** agreement occurs in the public interface
 - Not the implementation, and not the protocol
 - The 'remoteness' of an object is part of the contract

Programs are becoming networks



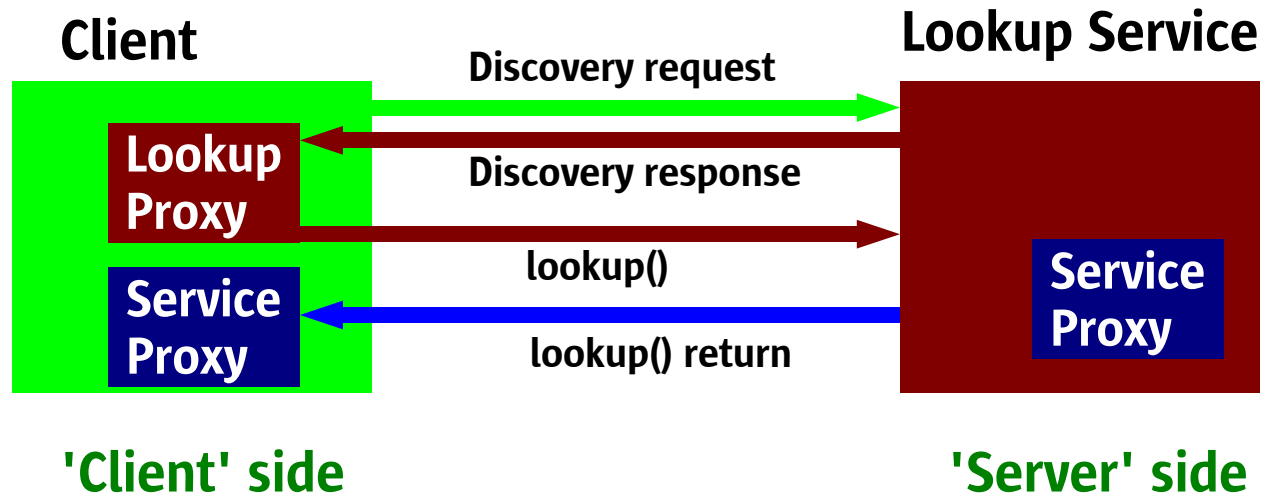
Lookup Discovery and Join

Service and Lookup Service interaction



Service Discovery

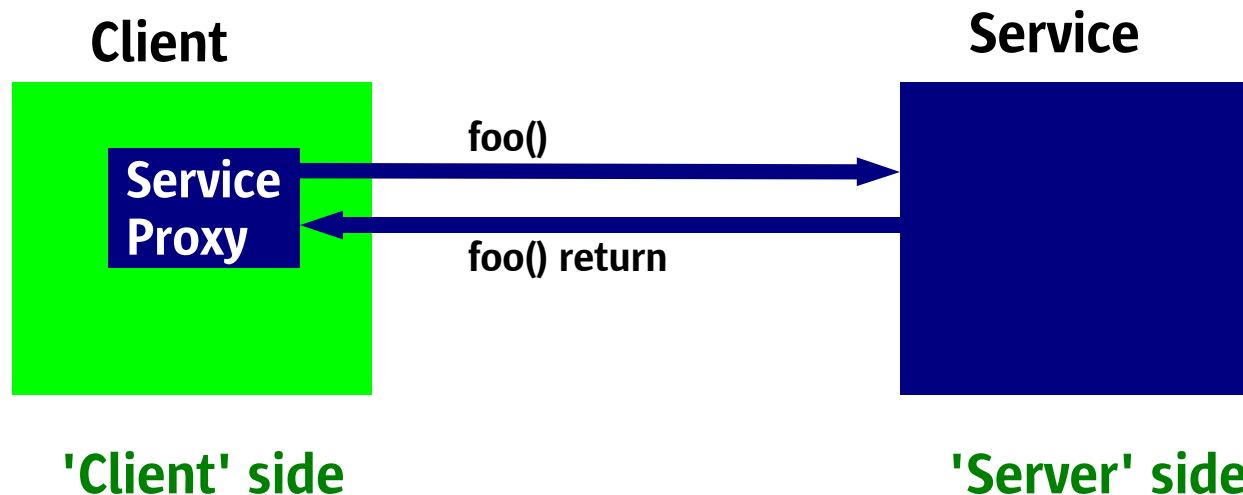
Client and Lookup Service interaction





Client and Service interaction

- Code downloaded into the client side
 - Client only knows about service's public interface
- Execution is **initiated** on the client side
- Execution **occurs** on the server side



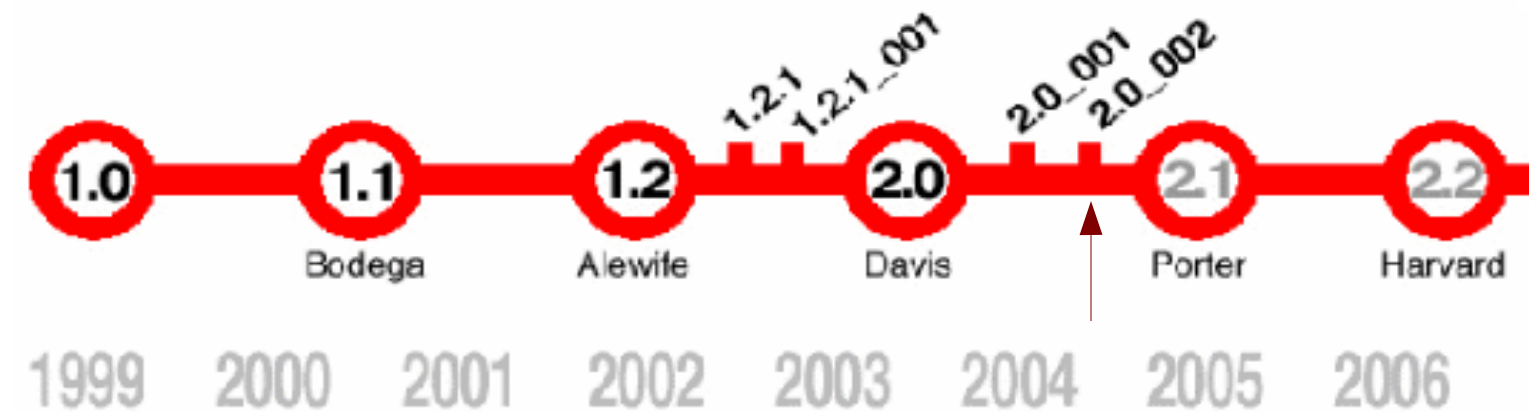


Technology by Community

- The Jini Community is
 - Open, dynamic, diverse, and global
 - Comprised of individuals, corporations, academic institutions, and other organizations
 - Advancing via technology improvements, standards development, developer support, marketing and business development activities
 - Credible reference for technology usage, “My friend recommended...”
- Jini Community is today providing
 - Commercially sold and supported implementations of Jini technology
 - Tools and utilities for Jini technology developers
 - Jini technology education and support services
- Jini Community also offers opportunities for collaboration, sharing, and learning
 - www.jini.org, events, meetings, projects, discussion lists, newsletters, etc.

Release history

Jini technology starter kit



- 1.0 – Original technology release
- 2.0 – Security and configuration capabilities
- 2.1 – Developer experience (build, install, patch) plus other improves
- Harvard – transaction manager updates, continue dev theme, resource/scale limit testing



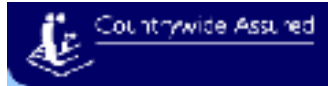
A license for the technology, for the community

- Encourage all who contribute to contribute under a single, common source license
- An existing, commonly accepted source license
- “My source code, my license”
- Initial community feedback
 - “Compatibility” in the sense of “interoperability”
 - SCSL too complex, much misunderstood
 - No “new” license
 - Compatibility with GPL
 - Assurance about patents

The Apache license



- On March 29 we released Jini Starter Kit v2.1 beta under the ASF v2 license
- Over time most, if not all, projects will also migrate
- The Jini specifications are re-licensed under ASF v2 license
- This license because:
 - It is well-known and widely accepted
 - Products based on the technology can be released under (L)GPL if one wishes
 - It addresses concerns about patents



Orbitz

(Dynamic Systems Integration)



- Environment
 - 1000+ servers (1300+ Jini Services – 25% growth in past year)
 - 90+ software developers (only 4 know Jini technology)
 - Small project teams (usually 2 or 3 people)
 - 3.9 million lines of code in the web system
 - 110,000 being added or deleted every week, most is in the application layer rather than the services layer
 - Sent over 775,000 wireless messages in Dec 2003
 - 22 million user accounts, 40-50K concurrent sessions
- Credits technology decisions for its speed to market and for its growth into the number 3 position in marketshare (17%) vs. Travelocity (20%) and Expedia (40%)
- “When SOA became popular, Orbitz had already found it.”

Canadian Astronomy Data Centre

(Compute grid)



- Calibration and analysis processing of large repository of telescope images.
- 350,000 images with more than 25 terabytes of data
- 60 dedicated compute workers plus varying numbers of volunteer workers
- Simplicity of the Jini/Javaspaces model is key
 - The system is about 5k-6k lines of code

Nedapp



- Security, identity management, physical resource management
 - Eiffel Tower (elevators), Rijksmuseum (access control) livestock management, city traffic management
- Hardware devices controlled by networked software
 - New hardware detected automatically
 - Devices represented by software objects
 - Each device object is a Jini service
 - Behavior objects determine behavior of system
 - Each behavior object is a Jini service
- Hardware becomes fully transparent within system

ISVs play a key role in the Jini technology ecosystem...



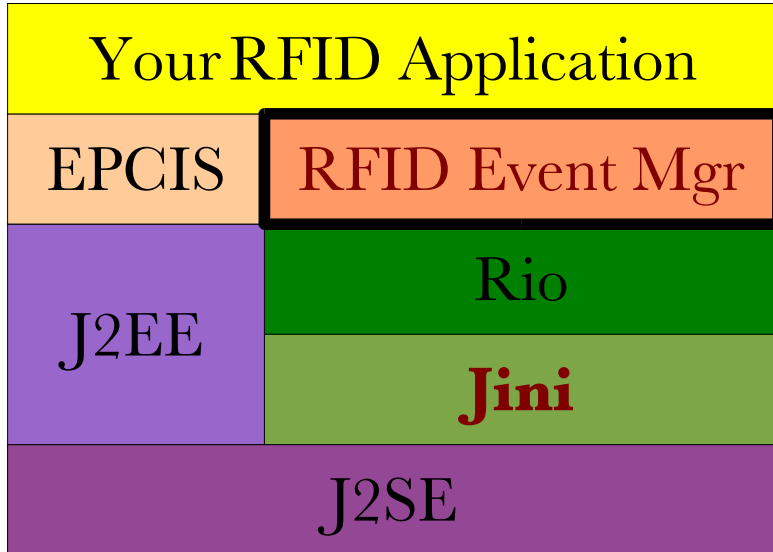
As do individuals...

Mark Brouwer
Calcum Shaw-Mackay
Dale Asberry
Dan Creswell
Bill Venners
Ken Arnold

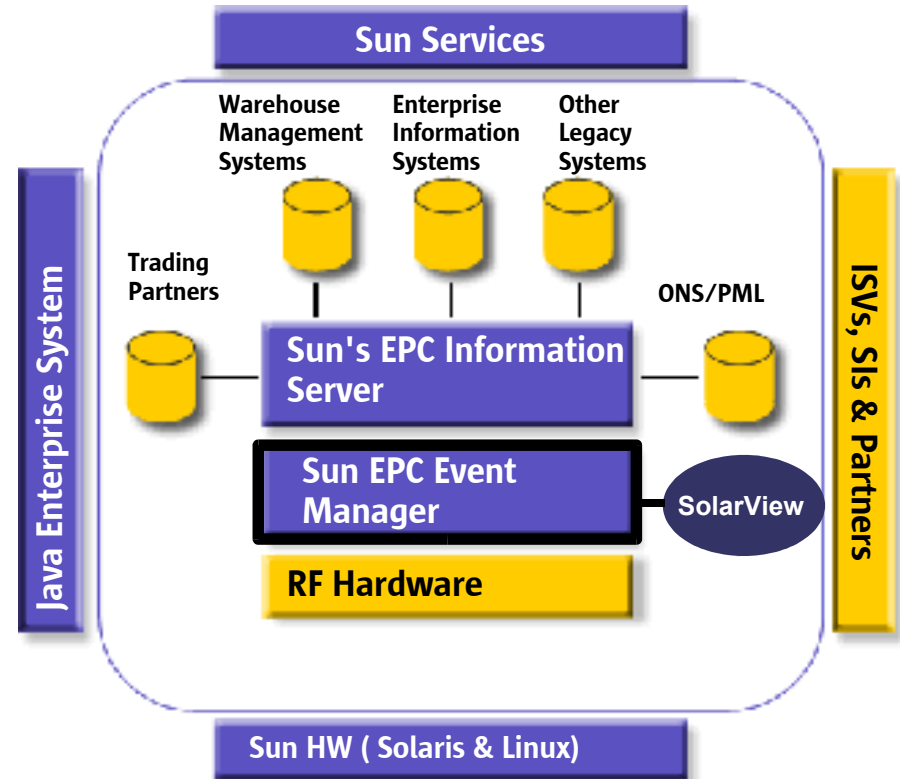
Jan Newmarch
Gregg Wonderly
Geoff Arnold
Jools Enticknap
Bill Rawlings
many more...

Sun's RFID product

(Edge Networking)



Sun EPC Event Manager



Jini technology

Value of the abstraction



- Logical interfaces support more flexible application design
 - Interfaces describe *what* services do, not *how* they do it
- Service code can run on client machine
 - *Where* code runs is an implementation detail
- Lower level networking protocols become interchangeable
 - Clients require no knowledge of lower level networking protocols
 - Remote communications protocols become service implementation details, which can be varied over time and/or space
- This is object-oriented networking

A Computational Community

- <http://jini.org>



- The Jini Networking Architecture
- A true network systems architecture
 - Reliable systems from unreliable parts
 - Constant change, long lifetimes
- How: Confronts the fallacies
 - Resources can vanish (topologies)
 - Implementations vary (latency/bandwidth)
 - Interfaces are everything
- It promotes the “disappearing computer”

Thank you!

Questions?

onno.kluyt@sun.com

