Jini™ Technology Executive Overview

THE NETWORK IS THE COMPUTER™

901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300
fax 650 969-9131

Revision 1.0
January 1999

Jini™ Technology Executive Overview

Picture this: *Three strangers from three companies—on the road to jointly bid a job. Among them they have a laptop, a project disk drive with data & applications, a PDA, a cell phone, and a clamshell pager—and their presentation is not ready. Their hotel suite has a small network: scattered net jacks, a couple of infrared & short-range RF transceivers, and though they never worked together before, they have snapped into an impromptu working community. The suite has an Internet gateway & low-resolution printer. The TV, the VCR, and the set-top box are connected to the same network. As they work they use all the services in the room, including those on each other's devices. Plug in the disk & PDA, turn on the laptop & TV—total setup time: 2 minutes.*

While one of them edits the presentation, the others watch it unfold, mirrored on the TV. From time to time one of them takes the laptop to another hotel room to concentrate on a section while the others keep working, running simulations from the project disk on the gateway & viewing results on the TV. Sometimes one of them prints a slide or graph to proof. With the clamshell they fetch files from offices & email questions. Using the cell phone as a remote, they order room service through the TV & check their flights.

When they're done, they need a high-resolution set of overheads. From the pager they reserve the hotel's high-res color printer for 15 minutes—it's in a small room down the hall. Using the suite key programmed by the reservation service to open the room, one of them goes into the printer room, loads foils bought from online room service, and phones into the suite's network to start printing.

Vision or Reality?

This is a great vision, but aren't computers and software too brittle to pull it off anytime soon? No—it's about to happen. Jini Technology makes computers and devices able to quickly form impromptu systems unified by a network. Such a system is a federation of devices, including computers, that are simply connected. Within a federation, devices are instant on—no one needs to install them. The network is resilient—you simply disconnect devices when you don't need them.

In our story the devices and how they communicated were central. The laptop was merely one player that came and went on a larger stage. The key was several smart and flexible independent devices simply connected. This created a work environment where the tools were ready to hand and largely invisible.

How Jini Technology Makes This Work

Jini technology provides simple mechanisms which enable devices to plug together to form an impromptu community—a community put together without any planning, installation, or human intervention. Each device provides *services* that other devices in the community may use. These devices provide their own interfaces, which ensures reliability and compatibility.

In our story, the hotel suite provided a small network and a *lookup service* with which devices and services registered. When the project disk was plugged in, it went through an add-in protocol—called *discovery* and *join*—in which the disk first located the lookup service (discovery) where it then uploaded all its services' interfaces (join). The other devices—the PDA, the clamshell pager, the cell phone, and the laptop—all went through the same process.

To use a service, a person or a program locates it using the lookup service. The service's interface is copied from the lookup service to the requesting device where it will be used. The lookup service acts as a switchboard to connect a client looking for a service with that service. Once the connection is made, the lookup service is not involved in any of the resulting interactions between that client and that service.

It doesn't matter where a service is implemented—compatibility is ensured because each service provides everything needed to interact with it. There is no central repository of drivers, or anything else for that matter.

In our story, the presentation was mirrored on the TV; to do this, the person operating the laptop selected the TV screen display service and plugged it into the presentation software. To reserve the hi-res printer, a service was selected that was built on top of the printer service to control who, how, and when the printer is used.

The Java programming language is the key to making Jini technology work. Devices in a Jini network are tied together using Java Remote Method Invocation (RMI™). By using the Java language, a Jini system is secure. The discovery and join protocols, as well as the lookup service, depend on the ability to move Java objects, including their code, between JVMs.

Jini technology not only defines a set of protocols for discovery, join, and lookup, but also a leasing and transaction mechanism to provide resilience in a dynamic networked environment. The underlying technology and services architecture is powerful enough to build a fully distributed system on a network of workstations. And the Jini infrastructure is small enough that a Jini device community can be built out of the simplest devices. For example, it is entirely feasible to build a Jini device community out of home entertainment devices or a few cellular telephones with no "computer" in sight.

Smart Devices

Devices permeate our lives. Look around: TVs, VCRs, DVDs, cameras, phones, PDAs, radios, furnaces, disk drives, printers, air conditioners, CD players, pagers, and the list goes on. A device performs a simple task, and only that task: Today devices are unaware of their surroundings—they are rigid and cannot adapt. When you buy a disk drive, you expend a lot of effort to install it or you need an expert to do it for you.

Now, devices of even the smallest size and most modest capabilities can affordably contain processors powerful enough for them to self-organize into communities that provide the benefits of multi-way interactions. A device can be flexible and negotiate the details of its interaction. We no longer need a computer to act as an intermediary between a cell phone and a printer. These devices can take care of themselves—they are flexible, they adapt.

A device that can take charge of its own interactions can self-configure, self-diagnose, and self-install. When computers were the size of large rooms, it made sense to have a staff of people to take care of them. As computers became smaller and shared by fewer people, each sys admin took responsibility for more computers. But now the cost of a computer is low, and Jini technology creates the possibility of impromptu device communities popping up in all kinds of places far from any sys admin. Self-managing devices reduce further the need for expert help, and this should lower the total cost of ownership for Jini-technology–based systems.

Historical Trends

How have we arrived at a place where connected devices are the locus for the next wave of computing?

The most significant reason is our better understanding of physics, chemistry, the physical bases for computation, and chip manufacturing process. Today, a significantly powerful computer can be built from one or two small chips and an entire computer system can built on one small board.

There were three dimensions of improvement: size, cost, and computational power. Since the 1960s, size and cost of computers have decreased dramatically while computational power has gone through the roof.

The mainframe of the 1960s was a collection of boxes in a large room—it cost millions of dollars and set the bar for computational power. Only a company could afford one.

The minicomputer became possible when the functionality of a mainframe could be put in a few boxes. It had the computational power of the previous mainframe generation, and could be bought by a single department. Most minicomputers were connected to interactive terminals—the beginnings of computer-based culture, a community.

When a computer the power of a mini shrank to a box that fit beside a desk, we got the workstation. A department could afford to buy one for a couple of professionals. A workstation had enough computational power to support sophisticated design, engineering, and scientific applications, and to provide the graphical support for them.

The personal computer was small enough to fit on a desk and powerful enough to support intuitive graphical user interfaces, individuals could afford them, and companies bought them for every employee.

Eventually processors became small enough and cheap enough to put one in a car in place of an ignition system, or in a TV instead of discrete electronics. Today's cars can have fifty or more processors, the home over a hundred.

The computational power dimension has another fallout. The overall trend toward smaller, faster, cheaper processors meant that fewer people had to share a CPU, but it also meant that people in the organization could become isolated. When a tool is shared, it creates a community; as the tool shrinks, fewer people use it together, and the community disperses. But, a community is hard to give up. Fortunately, computational power kept pace with the shrinking processor, and as the community served by a single computer system shrank, there was enough power to support communication between systems. Thus for example, workstations became successful once they could communicate and exchange data.

The final stretch of the computational power dimension is that now processors are powerful enough to support a high-level, object-oriented programming language in such a way to support moving objects between them. And such a processor is small enough and cheap enough to sit in the simplest devices.

Once there is sufficient computational power, the ability to connect and communicate is the dominant factor determining value. Today for most people, a computer runs only a few applications and mainly facilitates communication: email, the Web. Recall how fast Internet popularity soared first with email and more recently once the Web and browsers became prevalent.

Sun Community Source Licensing

When the Internet was developing, there were two essential activities: defining and perfecting the underlying protocols and infrastructure, and creating applications and services on top of that infrastructure. Internet infrastructure includes TCP/IP, HTTP, SMTP, and FTP—protocols and their implementations. On top of these were built email composers and readers, file fetching programs, Web browsers, and the Web itself. No single company or organization did all the work, and none could, if the venture was to be successful, because underlying it all is a standard protocol, and a protocol can be successful only if it is widely adopted.

For Jini technology to succeed, the underlying protocols and infrastructure must become pervasive, and to accomplish this requires a strong community of participants and partners.

The Sun Community Source License (SCSL) is a mechanism to build such a community around Jini technology. The SCSL opens the source code for the Jini infrastructure to the community of Jini technology licensees, who are free to use it, extend it, improve it, and repair it by following an open process that insures both fairness and stable evolution of the technology. Community members may add to this common body of source code while still maintaining, if they wish, proprietary implementations, though interfaces must be published so other community members can build their own implementations.

There are three levels of participation in the SCSL:

- Research Use: This enables researchers and students to use the source for any non-deployment purpose, and provides a way for organizations and individuals to examine and evaluate Jini technology.
- Internal Deployment: This enables organizations and individuals to deploy products based on Jini technology within their organization.
- Commercial Use: This is for commercial distribution and is based on a branding license model.

Fees are associated only with commercial use, and only in for-profit situations. The research use and internal deployment licenses are Web-based click-throughs, so that joining the community is simple and immediate, just as is Jini technology itself.

The SCSL anticipates an emergent, self-organizing community coalescing into interest groups surrounding different sorts of services, such as printing and digital cameras. Such interest groups would define, refine, and standardize interfaces for their category of service, providing useful community source and verification suites for new community members to get started.

What Our Community is Doing

A Jini device network can be built from many diverse types of devices. Today there are initiatives underway by several major manufacturers to enable their devices with Jini technology. These devices include printers, storage devices such as disks, personal digital assistants, digital cameras, cell phones, residential gateways, digital video cassette recorders, TV sets, set-top boxes, DVD players, industrial controls, and every sort of imaginable consumer electronics device. There are several technologies in preparation for connecting devices within a particular domain, such as home entertainment or industrial control, and bridges are being planned to connect these other types of networks to Jini communities.

Close

The right architecture makes it all simple—and the results can startle. Jini technology is only what is needed to gather a group of devices into an impromptu community that is simply connected: a simple protocol to discover a lookup service and join it, a lookup service which acts as a switchboard connecting clients to services, the Java language to provide the underlying object model, and RMI technology to provide federation and to move objects.

By looking to simple devices as the archetype of how to design devices and by creating a way to simply connect them, we may at last begin to see advanced computer technology simplify our lives.

Just before bed, the man with the laptop reads his email, half-watching the local news. The anchor mentions a brief piece on the local zoo's binturong to air after the break. His daughter is doing a project on them, so he records the piece on his laptop's drive. Just before bed, he decides to email it to her as a surprise.