David Clark
dwclark@earthlink

Spotlight

Network nirvana and the intelligent device

If Sun's Jini technology becomes widely adopted by manufacturers and developers, children of the future might sing new words to an old song: "Your cell phone's connected to your printer. Your printer's connected to your TV. Your microwave's connected to your dashboard." (They might even figure out how to make it rhyme.) However, whatever you call it—spontaneous networking, Web dialtone, automatic network resource coordinator—the new Java-based architecture that can make networking your computer devices as easy as plugging your telephone into a wall-jack could bring a dramatic shift to the existing ideas of distributed computing.

Jini is technology that can make a network look like one large computer. The technology will let developers and manufacturers create a whole range of computerized "appliances" that can instantly connect—wired or wireless—into a network to share information and services regardless of the underlying operating system or hardware. Once connected, every other computer, device, and user on the network will know that the new device (or resource) has been added and is available. If you want to use or access the resource, your, ah . . . computerized device will be able to download the necessary programming to communicate with that resource.

In a classic example of both over- and understatement, Mark Tolliver, president of Sun's new consumer and embedded division, gives this description of Jini: "At the core, it's nothing more than a small piece of Java code and nothing less than a road map for the future of networking technology."

No doubt, if Sun succeeds in getting developers to implement and improve Jini, the technology will have a tremendous impact on networking—especially for wireless and mobile users. It could provide overnight delivery for what business publications and technology prognosticators call the new generation of smart appliances. "What Jini brings from a value proposition," says Samir Mitra, director of business development for Jini, "is it extends Java by giving the ability to simply connect, and then to easily deliver any services that the network happens to provide."

Put another way, Mitra says, "The model we're trying to promote with Java and Jini is what we call a client services architecture." Until recently, he says, the industry's model was the Net being served on thin clients. Now, what we have is a "cloud," which is a network hosting a bunch of services. Each connected device (ranging from cell phones and desktop boxes to kiosks in the airport to hotel fax machines and printers to laptops and PDAs) is essentially a client and a service.

If you doubt the potential scenario of networked appliances and services, consider that Bill Gates regularly claims that more non-PCs than PCs will be attached to the Internet in 10 years. A recent *Business Week* cover story on information appliances suggests that we should "think divergence, not convergence." Gates and other industry leaders point out that the PC is so general purpose that few of us use more than 5% of its capacity. Furthermore, market analysts point out that after 10 or 15 years of availability, less than 50% of our homes contain a PC and that figure is projected to top out at only 60%. Information appliances, with their ease of use and single-purpose designs, will make them more appealing to consumers.

The growth and maturation of the Internet, the Web, e-mail, e-commerce, and "e-ntertainment" are certainly fueling this "divergence" of networked devices. Companies increasingly want to capitalize on Web-based services, and they want to reach consumers who don't—or won't—use traditional computers. Another important factor in this trend is that advances in the development of software and chip technology make the so-called information appliance feasible. It's now possible to build inexpensive devices with enough memory, storage, and display size to be useful. The key is to enable these devices to connect instantly and easily, both to each other and to the Internet. Jini could be that key.

How it works

Devices in a Jini distributed system provide their own interfaces, which ensures reliability and compatibility. Jini provides the distributed-system services for lookup, registration, and leasing. It consists of four program layers: directory service; JavaSpace; remote-method invocation (RMI); and the boot, join, and discover protocol. Any device with an operating system that supports a Java virtual machine (JVM) can be plugged into the network. And for some devices, of course, the operating system will consist of only the necessary Java/Jini functions and will be embedded on a chip that provides storage and other device-specific functions.

When a device is plugged into the Jini network, the directory service layer registers it as a member of the network. The device's program objects are placed in the JavaSpace layer so that other network members can "discover" the device's availability and its capabilities (for example, the *services* the device provides). The directory lookup service acts as a switchboard to connect a client looking for a service with a device that provides 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. The layer supporting the boot, join, and discover protocol enables devices, users, and applications to announce and register themselves and to discover others. The discovery and join protocols, as well as the lookup service, depend on the ability to move Java objects, including their code, into the JavaSpace and then between JVMs. Communication between objects in different JavaSpaces occurs through the RMI interface and layer, which includes security. (Jim Waldo, Jini's chief architect, says that although Jini leverages Java's already strong security model, more features will be added in future releases of Jini. By mid-1999, Sun expects to publish a draft specification that adds full-fledged authentication, nonrepudiation, and privacy to Jini and Java's RMI.)

The JavaSpace layer is the key to Jini; it manages object processing, sharing, and migration—essentially, it's a mechanism for client applications to make contact with a JavaSpaces server. However, don't think of JavaSpaces as an existing application or interface—JavaSpaces are simply a set of instructions developers use to create applications. Neither is Jini or JavaSpaces middleware—it's not used to create communication links between commercial databases such as Oracle, Sybase, DB2, or between object models such as CORBA, DCOM, and SOM. JavaSpaces (with Jini) is simply the underlying technology that lets developers build applications that can be distributed across a network of computers.

As such, one of the most important elements of the JavaSpace concept is in its simplicity (see Figure 1) and small footprint—both required by the next generation of information appliances. However, says Waldo, don't underestimate it just because it's simple and small. Jini's underlying technology and services architecture are powerful enough to build a fully distributed system on a network of workstations. The beauty of the design, he says, is that the JavaSpaces component model brings distributed-transaction processing forward in a way that most programmers can understand and use. Also, because it's Java-based, it should be quick and relatively easy for the existing community of Java developers to develop applications that incorporate Jini.

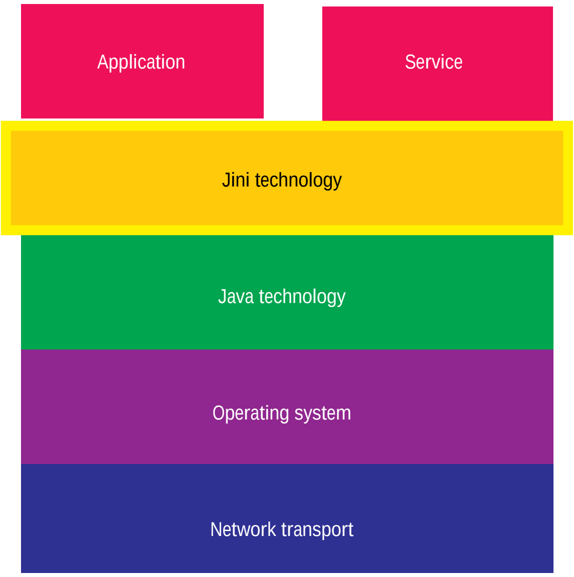

Figure 1. Jini technology provides simple mechanisms which enable devices to plug together to form a "federation." The components work together but are identified as separate components on a network. The federation provides services that are accessed by the various devices and applications. Members of a Jini federation share their services to get a job done. They communicate with each other through a set of Java interfaces known as the service protocol, and they locate each other using a special look-up service. The discovery and join protocols, as well as the lookup service, depend on the ability to move Java objects, including their code, between Java Virtual Machines.

WHO DID IT FIRST?

However beautiful the concept, the origin of Jini's JavaSpaces is contested. Everyone agrees the original ideas came from a 10-year-old Yale University project called "Linda," which David Gelernter initiated. The technology behind Linda is a set of powerful C functions that implement a distributed-computing model. Linda is also acknowledged by IBM in its experimental "TSpaces" technology, which resembles Jini but with some database and XML capabilities.

A small commercial developer, called Scientific Computing Associates, has supported and extended Linda since its early days at Yale. SCA claims Sun is using commercial parts of Linda in Jini without recognition (or payment). However, rather than going to court, SCA plans to redouble its marketing and development (among other products, a complex real-time trading system developed for Lehman Brothers uses Paradise, SCA's extension of Linda). According to Wayne Threatt, vice president of sales and marketing for SCA, his company's goal is to play in the Jini community. "We've believed in this kind of thinking for a long time. Our expertise and business experience can help turn Jini-related initiatives into a reality much faster."

Turning Jini initiatives into reality—getting developers and manufacturers to *use* it—will determine Jini's success. Quoting Jini's Web site: "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." Sun is using what it calls Sun Community Source License, which "opens" the source code for the Jini infrastructure to developers in an attempt to insure Jini's ubiquity. Developers are free to "use it, extend it, improve it, and repair it—they may add to the common body of source code and still maintain proprietary implementations."

Sun has announced that almost 40 hardware and software companies have agreed to license the Jini source code. Vendors include major consumer electronics manufacturers and office equipment companies as well as the computer manufacturers. In addition, Sun recently formed an alliance with 14 other computer, phone, and electric utility companies to use Java and Jini to network consumer and small business devices. Members of the group—called the Open Service Gateway Initiative—will develop standards intended to facilitate development of home gateway appliances.

What can Jini do?

A network of devices employing Jini technology can be built from any device that can run the JVM. That includes printers, storage devices, cell phones, digital cameras, digital VCRs, televisions, set-top boxes, DVD players, and industrial controls, not to mention just about any conceivable electronic toy or gadget.

We've heard for years how we'll be able to turn on our house lights and start the oven from our cars on the way home. Jini will bring that closer. Waldo says Sun is talking to auto companies where the applications for Jini are "fascinating and kind of easy," he says, "once you realize that what you're really driving around is a network base."

Furthermore, application developers can use Jini's technology to build virtual parallel supercomputers. By linking large numbers of PCs, they can use idle processing power to execute computationally intensive applications at blazing speed. This is already being done without Jini in projects such as the Avalon supercomputer that uses 68 PCs with Alpha processors, or in various search engine systems that use clusters of workstations. The difference is that with Jini, these systems can be joined much more easily and very dynamically—machines can join or leave without human intervention or failure.

"In Jini, the technology is built with the notion that nets are sometimes going to fail," says Sun's Mitra, "The fundamental software construct of Jini is inherently resilient to network failure." That key aspect of Jini is one reason why the mobile community is interested in the technology—mobile nets have reliability issues, he says. For example, Sun recently announced a partnership with NTT Mobile Communications Network to study the feasibility of using Java, Jini, and Java card technologies on the Japanese carrier's new mobile computing platform.

Another mobile project, Symbian, is a venture by the world's top three cellular phone makers and UK palmtop manufacturer Psion. Mitra says the research in both projects will enable mobile devices to have fully interactive communication with the network on a variety of different services above the telephony services such as e-mail, online banking, shopping, and the ability to download games and update software.

Jim Waldo says, "mobile computing is one of the areas [in which] we are already seeing people start to build applications." He describes a Sun flexible field office, used by sales and sales-support engineers, where nobody actually has an office. According to Waldo, Sun discovered that when the itinerant workers came in and grabbed a workspace, it took up to several hours to configure their laptops to get e-mail and use printers and fax machines. He says this is one of the things Sun actually designed Jini to do, and Sun plans to install Jini in that field office. Waldo says Jini will change how we see our laptops and portable devices. Instead of thinking of our mobile computers as the computer we carry around with us, we'll start thinking of them as a portable network.

Value added to commodities

Another important application for Jini is in what Samir Mitra calls the area of intelligent agents—especially in the office automation market. He says printers and office automation equipment, which are today mostly just peripherals to a PC or other computer, increasingly have full-fledged computers in them: microprocessors and memory. These devices are increasingly becoming networked, and their price points are coming down. "Because of this, there's an interest in the office automation market because Jini allows these devices to become intelligent—to become full network citizens with very little amount of software overhead," Mitra notes.

The same holds true for disk drives, he says. "Drive manufacturers are very interested in Jini, because now the disk drive actually becomes a storage service provider to the network. The disk drive can say, 'Hey I have a piece of software in here which is an intelligent agent that can do backup services for you so that you don't have to worry about it,' and therefore Quantum and Seagate can charge more because now they're delivering a backup service on their commodity hardware device." Adding intelligence to peripherals and office automation devices with Jini, according to Mitra, increases their value; consumers get more benefit, and manufacturers can get out of the commodity spiral they're in.

The competition

For Jini to accomplish network nirvana, it must become widely adopted. Although Sun has the impetus of a growing Java development community that should be able to take Jini in stride, competition exists from the usual suspects.

Microsoft recently announced Universal Plug and Play, an initiative designed to let a broad range of devices connect as peers over a home network and share resources. Sound familiar? UPP extends Microsoft's earlier plug-and-play initiative and applies it to a home network: peripherals can be attached to a network, "announce" their presence, and have the ability to interoperate with other devices on the network.

Microsoft's partners in UPP include Intel and Hewlett-Packard. The technology will be based on open standards, primarily TCP/IP and XML, and won't require Java. Industry analysts have speculated that UPP is more of an announcement than a reality at this point, but Microsoft's marketing power combined with its use of IP and XML technology could make UPP a strong competitor to Jini.

Other competitors include Hewlett-Packard, IBM, and Lucent Technologies. HP has recently announced Chai-Appliance Plug and Play, a technology that will allow Java-based Web appliances to be "discovered" on a UPP network. Like UPP, the ChaiAppliance communications is based on Web standards such as HTTP and XML, rather than on Java. However, ChaiAppliance was developed in the Chai Virtual Machine, which is HP's clean-room version of Java. This could possibly make ChaiAppliance Plug and Play useful as a bridge between Jini and UPP. Like Jini (and unlike UPP), ChaiAppliance is available to developers now.

Another HP technology, JetSend, is both complementary and competitive to Jini. The two-year-old platform-independent data exchange protocol is available on certain HP printer platforms now, but is expected in other devices soon. It will be compatible with UPP, but HP is also making it compatible with Jini, and JetSend licensees include several of the companies signed up with Jini.

Meanwhile, IBM has developed TSpaces, a component very similar to JavaSpaces. The technology provides for easy data exchange between heterogeneous Java-enabled devices. TSpaces are also complementary to Jini, which suggests that either or possibly both technologies could be eventually adopted without significantly changing Jini's development. Finally, Lucent has announced InfernoSpaces, a part of the company's Inferno venture that would function much like Jini or UPP, but is capable of running in both the Windows NT and Solaris environments.

Although Jini's architecture and specifications are in place, and real companies are developing real Jini-enabled applications, there's a lot left to do. Jini needs more specification and development—Sun hopes the licensee community will provide much of it. In addition, developers and software engineers need to learn how to program the new distributed applications. Finally, there's Jini, the Next Generation. When everybody's device is potentially connected to every other device in the world, how do you filter the noise? Waldo says, "yeah, figuring how to scale this stuff up from the human point of view is going to be tough. We know how to make the technology scale up, but being able to represent it in a way that humans can understand at that scale is something that I think is a major interesting area of research."

The good news about this technology is that we've seen the last of device drivers and many incompatibility issues. Before long, we'll check our e-mail with a phone device and print it out on the hotel's printer while we stand in the hotel's lobby, using the same device to check seating and book reservations at the restaurant next door. The bad news, at least for Sun, is that it might not be Jini that makes our device so powerful. The competition will eventually benefit the consumer, according to industry analysts such as the Gartner Group's Carolyn DiCenzo, but watch out for those competing standards along the way.

David Clark is a freelance writer based in Torrance, California, specializing in technology. Contact him at dwclark@earthlink.net.