



Jini™ Architecture Overview

Jim Waldo

A Jini system is a Java™-centric distributed system designed for simplicity, flexibility, and federation. The Jini architecture provides mechanisms for machines or programs to enter into a federation where each machine or program offers resources to other members of the federation and uses resources as needed. The design of the Jini architecture exploits the ability to move Java language code from machine to machine and unifies under the notion of a *service* everything from the user of a Jini system to the software available on the machines to the hardware components of the machines themselves.

LGE Exhibit-1010/Page 1 of
LGE v. Uni



© 1998 Sun Microsystems, Inc.
901 San Antonio Road, Palo Alto, CA 94303 U.S.A.
All rights reserved. Copyright in this document is owned by Sun Microsystems, Inc.

Sun Microsystems, Inc. (SUN) hereby grants to you at no charge a nonexclusive, nontransferable, worldwide limited license (without the right to sublicense) under SUN's intellectual property rights that are essential to use this Specification for internal evaluation purposes only. Other than this limited license, you acquire no right, title, or interest in or to the Specification and you shall have no right to use the Specification for productive commercial use.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/8) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-1(a).

This software and documentation is the proprietary information of Sun Microsystems, Inc. You shall use it only in accordance with the terms of the license agreement you entered into with Sun.

SUN MAKES NO REPRESENTATIONS OR WARRANTIES ABOUT THE SUITABILITY OF THE SOFTWARE, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. SUN SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES.

TRADEMARKS

Sun, the Sun logo, Sun Microsystems, JavaSoft, JavaBeans, JDK, Java, HotJava, HotJava Views, Visual Java, Solaris, NEO, Joe, Netra, NFS, ONC, ONC+, OpenWindows, PC-NFS, EmbeddedJava, PersonalJava, SUN, SunNet Manager, Solaris sunburst design, Solstice, SunCore, SolarNet, SunWeb, Sun Workstation, The Network Is The Computer, ToolTalk, Ultra, Ultracomputing, Ultraserver, Where The Network Is Going, Sun WorkShop, XView, Java WorkShop, the Java Coffee Cup logo, and Visual Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. THE PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.

Jini™ Architecture Overview

LGE Exhibit-1010/Page 2 of 2

Introduction

This document describes the high level architecture of a Jini software system, defines the different components that make up the system, characterizes the use of those components, and discusses some of the component interactions. This document identifies those parts of the system that are necessary infrastructure, those that are part of the programming model, and those that are optional services which can live within the system. This document also discusses the reasons behind particular design choices.

1.1 Goals of the System

A Jini system is a distributed system based on the idea of federating groups of users and the resources required by those users. The overall goal is to turn the network into a flexible, easily administered tool on which resources can be found by human and computational clients. Resources can be implemented as either hardware devices, software programs, or a combination of the two. The focus of the system is to make the network a more dynamic entity that better reflects the dynamic nature of the workgroup by enabling the ability to add and delete services flexibly.

A Jini system consists of the following parts:

- a set of components that provide an infrastructure for federating services in a distributed system
- a programming model that supports and encourages the production of reliable distributed services

The technology disclosed herein may be covered by patents or patents pending

- services that can be made part of a Jini federation and which offer functionality to any other member of the federation

While these pieces are separable and distinct, they are interrelated, which can blur the distinction in practice. The components that make up the Jini infrastructure make use of the Jini programming model; services that reside within the infrastructure also use that model; and the programming model is well supported by components in the infrastructure.

The end goals of the system span a number of different audiences; these goals include the following:

- enabling users to share services and resources over a network
- providing users easy access to resources anywhere on the network while allowing the network location of the user to change
- providing programmers with tools and programming patterns which allow the development of robust and secure distributed systems
- simplifying the task of building, maintaining, and altering a network of devices, software, and users.

The Jini system extends the Java application environment from a single virtual machine to a network of machines. The Java application environment provides a good computing platform for distributed computing because both code and data can move from machine to machine. The environment has built-in security that allows the confidence to run code downloaded from another machine. Strong typing in the Java application environment enables identifying the class of an object to be run on a virtual machine even when the object did not originate on that machine. The result is a system in which the network supports a fluid configuration of objects which can move from place to place as needed and can call any part of the network to perform operations.

The Jini architecture exploits these characteristics of the Java application environment to simplify the construction of a distributed system. The Jini architecture adds mechanisms that allow fluidity of all components in a distributed system, extending the easy movement of objects to the entire networked system.

The Jini infrastructure provides mechanisms for devices, services, and users to join and detach from a network. Joining into and leaving a Jini grouping is an easy and natural, often automatic, occurrence. Jini groups are far more dynamic than is currently possible in networked groups where configuring a network is a centralized function done by hand.

1.2 Environmental Assumptions

The Jini system federates computers and computing devices into what appears to the user as a single system. It relies on the existence of a network of reasonable speed connecting those computers and devices—10mbps in the general case. Some devices require much higher bandwidth and others can do with much less—displays and printers are examples of extreme points. We assume the latency of the network is reasonable, measured, at most, in seconds rather than minutes.

We assume that each Jini-connected device has some memory and processing power. Devices without processing power or memory may be connected to a Jini system, but those devices are controlled by another piece of hardware and/or software, called a *proxy*, that presents the device to the Jini system and which itself contains both processing power and memory. The architecture for devices not equipped with a Java virtual machine is discussed more fully in a separate document.

The Jini system is Java-technology centered. The Jini architecture gains much of its simplicity from assuming that the Java programming language is the implementation language for components. The ability to dynamically download and run code is central to a number of the features of the Jini architecture. However, the Java-centric nature of the Jini architecture depends on the Java application environment rather than on the Java programming language. Any programming language can be supported by a Jini system if it has a compiler that produces compliant bytecodes for the Java programming language.

1.3 Related Documents

This document does not provide a full specification of the Jini system. Each of the Jini components is specified in a companion document. In particular, the reader is directed to the following documents (note: Please check <http://www.java.sun.com/Products/Jini> for availability):

The technology disclosed herein may be covered by patents or patents pending

LGE Exhibit-1010/Page 5 of 5

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts



Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research



With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips



Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.