

Java™ Dynamic Management™ Architecture for Intelligent Networks



THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303
1 (800) 786.7638
+1.512.434.1511

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any. Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun logo, Java, JavaBeans, Java Dynamic Management, Java Studio, and Java WorkShop are trademarks, registered trademarks, or service marks of Sun Microsystems, Inc. in the U.S. and other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc. Information subject to change without notice.

The OPEN LOOK and Sun™ Graphical User Interface was developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.

Copyright 1998 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303-4900 Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Des parties de ce produit pourront être dérivées des systèmes Berkeley BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

Sun, Sun Microsystems, le logo Sun, Java, JavaBeans, Java Dynamic Management, Java Studio, et Java WorkShop sont des marques de fabrique ou des marques déposées, ou marques de service, de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC sont utilisées sous licence et sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" ET AUCUNE GARANTIE, EXPRESSE OU IMPLICITE, N'EST ACCORDEE, Y COMPRIS DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DE LA PUBLICATION A REpondre A UNE UTILISATION PARTICULIERE, OU LE FAIT QU'ELLE NE SOIT PAS CONTREFAISANTE DE PRODUIT DE TIERS. CE DENI DE GARANTIE NE S'APPLIQUERAIT PAS, DANS LA MESURE OU IL SERAIT TENU JURIDIQUEMENT NUL ET NON AVENU.



Please
Recycle



Adobe PostScript

Java™ Dynamic Management™ Architecture for Intelligent Networks

By executing computer applications identically on different computing platforms, the Java™ programming technology developed by Sun Microsystems, Inc. is unlocking the potential for industry-wide network computing. Similarly, by separating service creation and control from the underlying network infrastructure, Intelligent Networks promise to break the current monopoly on service provisioning held by service carriers and switch makers, and to provide the possibility for improved service differentiation. However, incompatibilities between different hardware and software protocol solutions are inhibiting Intelligent Networks from delivering on its full potential.

Java Advanced Intelligent Networks

Just as it has already enhanced the Internet, Java programming technology is now poised to introduce a new dimension in service deployment, portability, distribution, and management for Intelligent Networks.

Simply by combining Java programming technology with Intelligent Networks, the Java Advanced Intelligent Network immediately addresses the common problems of incompatibility, portability, and scalability that are currently slowing the expansion of Intelligent Networks throughout the telecommunications industry. It also serves to draw the traditionally rival worlds of smart telephony and the Internet together, providing a blend of high availability and accessibility combined with increased flexibility.

However, the Java Advanced Intelligent Network represents much more than a set of programming interfaces to hide the complexities of differing hardware platforms and protocol stacks, and offers more than a simple Java implementation of Intelligent Networks.

The underlying infrastructure of the Java Advanced Intelligent Network is provided by the Java™ Dynamic Management™ environment. It is this infrastructure which resolves the problem of resource management in Intelligent Networks and allows the Java Advanced Intelligent Network to claim true portability and scalability in a distributed network. In addition, the Java Dynamic Management environment further dissolves the unnecessary distinction between Intelligent Networks and the Internet because exactly the same management framework is used in both domains.

The Need for a Management Solution in Intelligent Networks

Until now, telecommunications service providers have been able to maintain extremely tight control over the types of equipment and services attached to their networks. However even within this closed environment, up to 90% of the deployment cost of a specific service can be attributed to the cost of managing it. Despite this fact, Intelligent Networks still fail to address the problem of service management concretely, and do not provide any definition for a management interface.

Furthermore, the service-driven network and the increasing convergence of Intelligent Networks and the Internet requires an environment that supports software distribution and platform independence, without placing constraints on the design and implementation of services. Contrary to the currently closed environment of Intelligent Networks, this new, open environment engenders its own set of challenges for the service supplier. Paramount of these challenges are *secure delivery* and *entity management*.

Secure delivery is inherent in the Java programming technology itself because it was specifically developed to support the distribution of secure network applications across the Internet. In addition the Java Dynamic Management framework makes provisions for authentication and login mechanisms to verify user identity.

Entity management for the Intelligent Network can be divided into four areas:

- Management of service applications pushed to the network
- Service provisioning
- Service creation and deployment
- Service operations and maintenance

The Java Dynamic Management architecture provides a framework that is shared by all Intelligent Network entities and within which all entities are manageable. The Java Advanced Intelligent Network combines this unique distributed intelligence and cohesive management services that restores the promise of Intelligent Network service portability.

This proposed architecture blends Intelligent Network and Internet technologies to provide state-of-the-art telecom services. Intelligent Network services could be assembled “on the fly” in a plug-and-play fashion, drastically reducing the time and effort to develop services.

What benefits would result?

Under such a Intelligent Network architecture, it is possible to propagate intelligence into Webphones, personal digital assistants (PDAs), PCs, and mobile handsets. A further positive result of this Internet/Intelligent Network integration is that standard Web user interfaces, such as browsers and HTML links, will find their way into the Intelligent Network environment.

By combining Intelligent Network and Internet technology, the architecture will allow carriers to deliver the content and services their customers have been waiting for. They would have the ability to deploy Intelligent Network services with an Internet interface on any device at the edge of the network, including any Java technology-enabled end-user devices.

Combining Java Dynamic Management with Intelligent Networks

The Java Dynamic Management environment is the first Java-based solution for building management intelligence directly into network devices and applications, thus distributing the management workload throughout the network. In this environment, *smart* agents are capable of performing much of the management processing locally, which reduces the overhead of communication between clients and managers.

Management services, implemented as JavaBeans™ components, can be propagated throughout the network and updated dynamically as the network evolves. The Java Dynamic Management architecture is fully compliant with the JavaBeans component model.

The Java Dynamic Management environment provides the Java classes and Java interfaces required to develop smart agents, and the compilers and tools required to create them. It includes:

- A dynamic management architecture, which provides Web-based mechanisms for propagating management services throughout the network.
- An agent framework, which provides a library of reusable core agent services. For example, an object repository, dynamic class loading, native library loading, and basic notification, filtering, and monitoring services.
- A master agent framework, which provides a library of master agent services. For example, the cascading service that allows a master agent to access the managed objects of the agents they control.

The core services are implemented as JavaBeans components called *m-beans* and *m-lets*. An *m-bean* is either dedicated to management tasks (a *management* bean) or provides a manageable interface (a *manageable* bean). An *m-let* is a package of *m-beans* that can be manipulated for downloading new services or upgrading existing services.

Additional *m-beans* and *m-lets* can be developed using any available bean builder (for example, Java™ WorkShop™ or Java™ Studio™ software) to extend the basic services provided by the Java Dynamic Management Kit framework and to fully integrate existing Intelligent Network applications.

The Java Dynamic Management framework does not impose management APIs on the developer; the only constraint is that the developer must develop JavaBeans components. This means that the developer doesn't need to adapt the code to work specifically with the Java Dynamic Management architecture.

Conclusion

For carriers and service providers, this Java Dynamic Management Kit/Intelligent Network framework offers different layers of benefits. On one level, the resulting rich portfolio of platform-independent Intelligent Network applications will help carriers control costs and protect their investments. But it also gives them the tools to compete effectively by speeding time-to-market because of the portability, rapid application development cycles and reusability of these new Intelligent Network applications. Ultimately, against the backdrop of the new era of telecom competition, carriers who embrace these new Intelligent Network capabilities will grow and prosper, leveraging their ability to rapidly create new services as a means to differentiate and distinguish themselves from less nimble competitors.

Frequently Asked Questions

1. *The Java Advanced Intelligent Network is a new solution, which is still currently evolving. Does that mean that it will be at least 3 years before its really available and useful?*

No. The underlying architecture for the Java Advanced Intelligent Network is the Java Dynamic Management framework, which exists and is available today.

2. *The Java Advanced Intelligent Network is yet another new architecture for Intelligent Networks. Does that mean I will have to discard all my existing investment in devices and software?*

No. The Java Dynamic Management architecture allows existing systems to be fully integrated with the evolving intelligent network, so you can protect your current investments while taking advantage of the latest Intelligent Network architecture.

3. *By choosing the Java Advanced Intelligent Network, am I simply committing myself to yet another Intelligent Network vendor?*

No. The Java Dynamic Management architecture is fully compliant with the JavaBeans component model, and does not rely on any proprietary protocols.

4. *Is performance an issue in the Java Advanced Intelligent Network architecture?*

Performance is always a consideration. Java achieves superior performance by adopting a scheme by which the interpreter can run at full speed without needing to check the run-time environment. The automatic garbage collector runs as a low-priority background thread, ensuring a high probability that memory is available when required, leading to better performance.

Applications requiring large amounts of compute power can be designed such that compute-intensive sections can be rewritten in native machine code as required and interfaced with the Java platform. In general, users perceive that interactive applications respond quickly even though they're interpreted.

While these performance numbers for interpreted bytecodes are usually more than adequate to run interactive graphical end-user applications, situations may arise where higher performance is required. In such cases, Java bytecodes can be translated on the fly (at run time) into machine code for the particular CPU on which the application is executing. This process is performed by the Just-In-Time (JIT) compiler. For those accustomed to the normal design of a compiler and dynamic loader, the Just-In-Time compiler is somewhat like putting the final machine code generator in the dynamic loader.

5. *How about security and Java Dynamic Management Kit/Java Advanced Intelligent Network?*

Security commands a high premium in the growing use of the Internet for products and services ranging from electronic distribution of software and multimedia content to "digital cash". The area of security with which we're concerned here is how the Java compiler and run-time system restrict application programmers from creating subversive code.

The Java language compiler and run-time system implement several layers of defense against potentially incorrect code. The environment starts with the assumption that nothing is to be trusted, and proceeds accordingly.

The Java language is designed for creating highly reliable software. It provides extensive compile-time checking, followed by a second level of run-time checking. Language features guide programmers towards reliable programming habits.

Java technology is designed to operate in distributed environments, which means that security is of paramount importance. With security features designed into the language and run-time system, Java lets you construct applications that can't be invaded from outside. In the network environment, applications written in the Java language are secure from intrusion by unauthorized code attempting to get behind the scenes and create viruses or invade file systems.



THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303

1 (800) 786.7638
+1.512.434.1511

<http://www.sun.com/software/>

September 1998