Project JXTA:
An Open, Innovative Collaboration

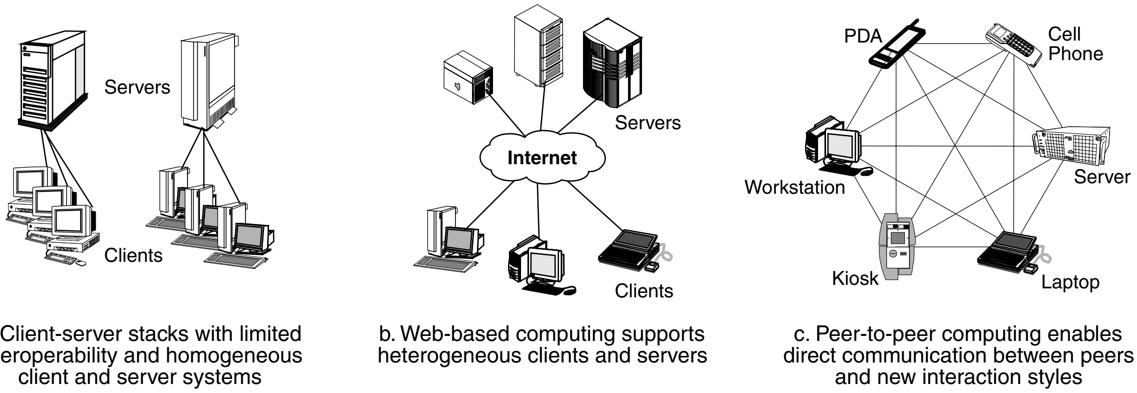
Introduction

Almost overnight, users everywhere have adopted innovative technologies that enable them to participate in a larger vision of the Internet. Distributed computing applications like SETI@home have empowered millions of users to contribute their computing resources to work on a common computational analysis. Instant messaging services have enabled users to communicate and collaborate with their peers in real time. And true peer-to-peer (P2P) computing, embodied by applications like Napster, Gnutella, and Freenet, has offered a compelling and intuitive way for Internet users to find and share resources directly with each other, often without requiring a central authority or server. As much as these diverse applications have broken new ground, they most often address only a single function, run primarily only on a single platform, and are unable to directly share data with other, similar applications.

Seeing the potential of P2P computing to enable access to a broader, deeper Web — and the need to free today's model from its current limitations — Sun's Chief Scientist Bill Joy has started a small research effort called *Project JXTA*. Working with developers in an open, collaborative manner, Project JXTA is helping to create a common platform that makes it simple and easy to build a wide range of distributed services and applications in which every device is addressable as a peer, and where peers can bridge from one domain into another. Project JXTA's technologies will enable developers to focus on their own application development while easily creating distributed computing software that is flexible, interoperable, and available on any peer on the expanded Web.

Evolution of Internet Use

In contrast to the Internet's design as a flat interconnection of networks, the primary way in which network applications have been built during the last two decades is primarily hierarchical, following the client-server model (Figure 1a). First used in local-area networks, client-server applications once required homogeneous client and server systems and provided limited or no interoperability between applications.



a. Client-server stacks with limited interoperability and homogeneous client and server systems

b. Web-based computing supports heterogeneous clients and servers

c. Peer-to-peer computing enables direct communication between peers and new interaction styles

Figure 1: Evolution of network computing from client-server, to the World-Wide Web, to peer-to-peer computing. Logical, not physical connections are illustrated.

The World Wide Web made client-server computing commonly available through the acceptance of a universal client (Web browser) that uses a standard communication protocol (HTTP), that can display information described in a standard format (HTML), and that can execute applications written using Java™ and Extensible Mark-up Language (XML) technologies (Figure 1b). Anyone with Web-enabled devices ranging from personal computers to cell phones can use this model — as long as they connect to a Web server with a known location and name.

Today's peer-to-peer file sharing programs have tapped into simple user needs — and the success of these applications has shifted the way people think about how they use the Internet. Returning to the roots of how the Internet is built, users can interact with each other directly, without involving a Web server, chat room arbitrator, or bulletin board system. The Web suddenly appears broader and deeper as users have access to a whole new style of computing that doesn't necessarily require a set of clients and servers in a hierarchical relationship (Figure 1c).

The rapidly-growing set of distributed applications and services suggest a model that *complements* the client-server model while emphasizing direct communication between Internet users, not mainly from users at the Internet's edge to servers located at its core. Rather than clients and servers having a vertical relationship, both can exist as equal peers on the network — despite different performance characteristics. With so much computing power, disk storage, and network bandwidth located outside of data centers and in the hands of users everywhere, an acceptance of sharing personal resources for common benefit is hastening the adoption of these technologies.

Peer-to-Peer Enables Complete Access

The widespread acceptance of first-generation peer-to-peer applications has brought distributed computing into the limelight — and suggests that the Internet is at the next inflection point in its growth. The overnight adoption of peer-to-peer file sharing software is testimony to the power of this model to expand access to resources, enhance the richness and depth of content, and propel the growth of the Web.

Searching and Sharing

Peer-to-peer computing is generating excitement because it offers an intuitive model for the most fundamental Internet activities: *searching and sharing*. Although today's applications are primarily used for finding, retrieving, and using media files, they hint at what complete access to the Web can deliver in the future:

- Resources — including information and processing power — can be shared directly from those who *have* them to those who *need* them. Documents can be managed and encrypted so that they are anonymous, likely to exist somewhere in the network at all times, and more secure.

- Peer-to-peer searches are distributed, parallel, and asynchronous, enabling 'deep' searches of Internet content that quickly yields up-to-the-minute results. Today's look-ups conducted by Internet search engines are limited by the months it takes their crawlers to traverse the sites that they index. In contrast, peer-to peer search results are more relevant, and can be more focused. For example, by qualifying the peer group, and qualifying the search, more precise results can be obtained over a much larger search space.

- Instant messaging systems can locate users quickly with distributed searches, enabling direct user-to-user communication and collaboration independent of any service provider.

- Buyers and sellers can be matched directly through P2P auction and transaction services, and new generations of distributed resource-sharing tools like today's Search for Extraterrestrial Intelligence (SETI@home) will begin to appear.

- Peer shells will enable users and developers to experiment and prototype new applications using simple scripting languages. Command-line administrative tools can be used to manage applications and their peer groups.

A New Style of Distributed Computing

Peer-to-peer computing enables applications that are collaborative and communication-focused; more probabilistic than deterministic. Applications that are well-suited to this model are those which can tolerate the coming and going of individual peers. In this model, if the desired result is not obtained, users can try again later. The information exchanged in P2P environments is timely and accurate, yet results may vary from time to time depending on which peer group members are available.

High availability comes through the existence of multiple peers in a group, making it likely that at any time there is a peer in the group able to satisfy a user request. This stands in stark contrast to traditional computing models, where high availability comes through complex load-balancing and application fail-over schemes. Peer-to-peer computing leverages available computing performance, storage, and bandwidth found on systems around the globe, and works because people realize that there is value in sharing their power with others in order to reap the benefits when they need it themselves.

In many ways, the world of peer-to-peer computing is juxtaposed to the hierarchical client-server model, hence the name *Project JXTA*. By supporting applications that are collaborative and communication-oriented, Internet use can be more natural and intuitive.

Vision for Project JXTA

Project JXTA is extending P2P computing to enable a wide range of distributed computing applications and overcome the limitations found in many of today's P2P applications. Project JXTA should eventually enable new applications to run on any device that has a digital heartbeat — including desktop computers and servers; PDAs, cell phones, and other connected devices.

Sun believes that an open, collaborative development process works best for creating and stimulating the rapid adoption of new software technologies. This approach admits that there are vast resources of programming expertise outside of any company's doors, and that no company has a monopoly on smart programmers.

Project JXTA is involving companies that are committed to work with the open source model, invest in P2P technology, and share results that can enable developers to leverage their domain-specific knowledge and create rich, new peer services and applications. The project's Web site (http://*www.jxta.org*) promotes the results of this effort and enables sharing of community-built software. The project's outputs are licensed through an Apache-style software license, facilitating access by developers. Interested contributors with similar visions for P2P computing are invited to join the site's 'peer group,' monitor progress, and contribute ideas.

Peer-to-peer is an important network computing technology that complements both client-server and Web-based paradigms — and presents a similar set of problems to solve, including isolated client-server application stacks, platform independence, security, and administration tools.

A Peer-to-Peer System

Project JXTA is building core network computing technology to provide a set of simple, small, and flexible mechanisms that can support P2P computing on any platform, anywhere, and at any time. The project is first generalizing P2P functionality and then building core technology that addresses today's limitations on P2P computing. The focus is on creating basic *mechanisms* and leaving *policy* choices to application developers.

JXTA technology leverages open standards like XML, Java technology, and key concepts that make the UNIX® operating system so powerful and flexible — including the ability for shells to connect commands together using pipes to accomplish complex tasks. By using existing, proven technology and concepts, the result will be a peer-to-peer system that is familiar to developers and easy to build upon.

Project JXTA is creating a peer-to-peer system by identifying a small set of basic functions necessary to support P2P applications and providing them as building blocks for higher-level functions (Figure 2). At the core, capabilities must exist to create and delete peer groups, to advertise them to potential members, to enable others to find them, and to join or leave them. At the next layer, the core capabilities can be used to create a set of peer services, including indexing, searching, and file sharing. Peer applications can be built using these facilities. In addition, peer commands and a peer shell have been created to create a window into the JXTA technology-based network.

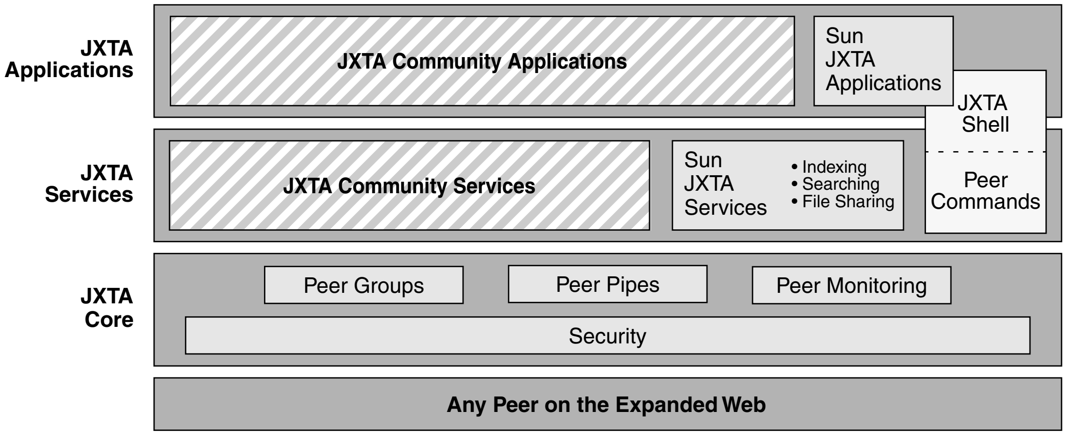


Figure 2: JXTA software is layered with core functionality developed through an open, collaborative effort, and higher-level services developed by P2P community developers.

JXTA Core

The *JXTA Core* provides core support for peer-to-peer services and applications. In a multi-platform, secure execution environment, the mechanisms of *peer groups*, *peer pipes*, and *peer monitoring* are provided:

- *Peer groups* establish a set of peers and naming within a peer group with mechanisms to create policies for creation and deletion, membership, advertising and discovery of other peer groups and peer nodes, communication, security, and content sharing.

- *Peer pipes* provide communication channels among peers. Messages sent in peer pipes are structured with XML, and support transfer of data, content, and code in a protocol-independent manner — allowing a range of security, integrity, and privacy options.

- *Peer monitoring* enables control of the behavior and activity of peers in a peer group and can be used to implement peer management functions including access control, priority setting, traffic metering, and bandwidth balancing.

The core layer supports choices such as anonymous vs. registered users, and encrypted vs. clear text content without imposing specific policies on developers. Policy choices are made, or when necessary, implemented, at the service and application layers. For example, administration services such as accepting or rejecting a peer's membership in a peer group can be implemented using functions in the core layer.

JXTA Services

Just as the various libraries in UNIX operating systems support higher-level functions than the kernel, *JXTA Services* expand upon the capabilities of the core and facilitate application development. Facilities provided in this layer include mechanisms for searching, sharing, indexing, and caching code and content to enable cross-application bridging and translation of files.

Searching capabilities include distributed, parallel searches across peer groups that are facilitated by matching an XML representation of a query to be processed with representations of the responses that can be provided by each peer. These facilities can be used for simple searches — like searching a peer's repository — to complex searches of dynamically-generated content that is unreachable by conventional search engines.

P2P searches can be conducted across a company's intranet, quickly locating relevant information within a secure environment. By exercising tight control over peer group membership, and enabling encrypted communication between peers, a company can extend this capability to its extranet, including business partners, consultants, and suppliers as peers. The same mechanisms that facilitate searches across the peer group can be used as a bridge to incorporate Internet search results, and to include data outside of the peer's own repository, for example searching a peer's disk.

The peer services layer can be used to support other custom, application-specific functions — for example a secure peer messaging system could be built to allow anonymous authorship and a persistent message store. The peer services layer provides the mechanisms to create such secure tools; specific tool policies are determined by the application developers themselves.

JXTA Shell

Straddling the boundary between peer services and applications is the *JXTA Shell*, which enables both developers and users to experiment with the capabilities of JXTA technology, prototype applications, and control the peer environment. The peer shell contains both built-in functions that facilitate access to core-level functions through a command-line interface, and external commands which can be assembled using pipes to accomplish more complex functions — just like a UNIX operating system shell, but extended into the peer-to-peer environment. For example, a shell command `peers` lists the accessible peers in the group. Other facilities include command-line access to peer discovery, joining and leaving peer groups, and sending messages between peers. In the future, shell-level commands will enable administrative control of peer groups, including who can join a given group and what resources they can access.

JXTA Applications

JXTA Applications are built using peer services as well as the core layer. The project's philosophy is to support the fundamental levels broadly, and rely on the P2P development community to provide additional peer services and applications.

Peer applications enabled by both the core and peer services layers include P2P auctions that link buyers and sellers directly — with buyers able to program their bidding strategies using a simple scripting language. Resource-sharing applications like SETI@home can be built more quickly and easily, with heterogeneous, world-wide peer groups supported from day one. Instant messaging, mail, and calendaring services can facilitate communication and collaboration within peer groups that are secure and independent of service provider-hosted facilities.

Enabling a New Generation

JXTA technology provides fundamental mechanisms for solving many of the issues of today's distributed computing applications, enabling a new generation of ubiquitous, secure, interoperable, heterogeneous applications. It currently supports Java technology-based platforms and systems without Java technology. In the future JXTA technology will support small, mobile devices with limited memory footprints. JXTA software's combination of Java technologies with XML data representations provides flexibility, power, and the ability for vertical applications to interoperate in ways that overcome the limitations of P2P software today. And with small, simple building blocks that are easy to build upon, JXTA technology frees developers from having to invent their own framework — enabling them to focus on building new, innovative, distributed computing applications.

References

Sun Microsystems, Inc. posts complete information on Sun's hardware and software products and service offerings in the form of data sheets, specifications, and white papers on its Internet Web page at *http://www.sun.com/*.

More information on Project JXTA may be found at *http://www.jxta.org/*.

SUN MICROSYSTEMS, INC., 901 SAN ANTONIO ROAD, PALO ALTO, CA 94303-4900 USA
PHONE: 650-960-1300 FAX: 650-969-9131 INTERNET: *www.sun.com*

Draft 1.0 April 25, 2001