

White Paper

# Sprint Enterprise Application Messaging<sup>SM</sup>

## A Secure Messaging Framework



July 13, 2003

**An Insight Into Technology**

iQknowledge

*Prepared by:*  
**InQuest Corporation**

*Sponsored by:*  
**Sprint Mobile Computing Services Group**

© InQuest Corporation

## Table of Contents

<b>EXECUTIVE SUMMARY .....</b>	<b>1</b>
<b>1. INTRODUCTION .....</b>	<b>2</b>
<b>2. RISKS ASSOCIATED WITH CONSUMER MESSAGING PRODUCTS.....</b>	<b>4</b>
<b>3. THE URGENCY TO SECURE REAL-TIME MESSAGING CONNECTIONS .....</b>	<b>7</b>
3.1. PRESENCE.....	8
3.2. VERTICAL INDUSTRY REGULATORY CONSIDERATIONS .....	8
3.3. FILE TRANSFER AND FILE SHARING .....	9
<b>4. A SECURE ENTERPRISE APPLICATION MESSAGING FRAMEWORK .....</b>	<b>9</b>
4.1. DIRECTORY INTEGRATION AND ACCESS CONTROL .....	10
4.2. AUTHENTICATION AND IDENTIFICATION .....	10
4.3. ENCRYPTION, AUTHORIZATION AND PRIVACY .....	11
4.4. CENTRAL MANAGEMENT .....	12
4.5. ARCHIVING, AUDIT, AND REPORTING CAPABILITY .....	12
4.6. INTEROPERABILITY & INTEGRATION WITH BUSINESS APPLICATIONS .....	13
4.7. POLICIES & PROCEDURES .....	14
<b>5. COMPONENTS OF THE APPLICATION MESSAGING FRAMEWORK .....</b>	<b>15</b>
5.1. INFRASTRUCTURE COMPONENTS .....	16
5.2. SPRINT'S SECURE FRAMEWORK.....	18
<b>6. A PHASED APPROACH TO ENTERPRISE APPLICATION MESSAGING .....</b>	<b>20</b>
6.1. PHASE 1: ASSESS AND CONTROL.....	20
6.2. PHASE 2: DEVELOP & DEPLOY .....	20
6.3. PHASE 3: EXTEND THE IMPLEMENTATION.....	20
<b>7. CONCLUSION .....</b>	<b>21</b>
<b>8. GLOSSARY: COMMUNICATION SECURITY CONCEPTS – A NEED TO KNOW</b>	<b>22</b>
8.1. PUBLIC KEY INFRASTRUCTURE.....	23
8.2. SYMMETRIC-KEY CRYPTOGRAPHY .....	23
8.3. STREAM CIPHERS .....	23
8.4. BLOCK CIPHERS.....	23
8.5. SSL PROTOCOL.....	24
8.6. IPSEC PROTOCOL.....	24
<b>9. REFERENCES:.....</b>	<b>24</b>

---

## Highlights

---

***Application Messaging is used to denote an instant messaging (IM) architecture that in addition to person-to-person communication extends to support real-time person-to-application as well as application-to-application messaging.***

***Enterprises who wish to integrate real-time messaging into their infrastructure need to ensure that they meet all corporate and regulatory security requirements.***

***Sprint's Enterprise Application Messaging<sup>SM</sup> framework is robust in its implementation of best forward-looking practices that ensure a secure conversation between clients and satisfies all regulatory requirements***

## Executive Summary

*Application messaging* is used to denote an instant messaging (IM) architecture that in addition to person-to-person communication extends to support real-time person-to-application as well as application-to-application messaging. Application messaging extends the existing web and application architectures to simple text- or forms-based IM clients that can be used from almost any device. This capability makes application messaging a “killer” application that wireless carriers, such as Sprint, have been looking for to introduce new customer-facing business solutions that complement their significant investments in next-generation wireless services.

*Presence awareness* is a key feature of application messaging. The real-time information about the availability of a person, application, or process will prove very valuable in speeding the enterprise's ability to respond to a situation. Presence awareness is a strong proposition that enables alerting and collaboration functionalities that the traditional web technologies alone cannot provide.

Application messaging vastly improves the speed at which data gets delivered throughout the enterprise. Enterprises are just beginning to understand the rewards of real-time messaging in terms of user productivity and customer experience -- a reward that can be significantly increased once extended to business applications and mobility. Application messaging can be integrated with many types of corporate applications, including CRM, ERP, telephony, system & network management, directory, order entry, and inventory systems. For example, a solution that instantly alerts available system administrators to problems with infrastructure components will prove extremely valuable in helping IT groups maintain service level agreements.

Enterprises that wish to integrate real-time messaging into their infrastructure need to meet corporate and regulatory security requirements. Enterprises must develop, adopt, and implement security policies that specifically address real-time messaging. Sprint's Enterprise Application Messaging<sup>SM</sup> framework is robust in its implementation of best forward-looking practices that ensure a secure conversation between clients and satisfy regulatory requirements. Sprint does this by incorporating robust authentication, end-to-end encryption, audit and reporting, access control, and central administration in its implementation of the solution.

Sprint has recognized that the success of its foray into the application messaging market is contingent on the capability of its application messaging framework to securely integrate with other third-party Enterprise Instant Messaging (EIM) solutions. While Sprint's application messaging framework provides a complete real-time messaging solution for customers that are new to EIM, it does not require replacement of existing EIM solutions. The flexibility to securely integrate other EIM solutions into its application messaging architecture has placed Sprint on the path to forge new partnerships with companies such as Microsoft for delivery of comprehensive EIM and enterprise application messaging solutions.

This white paper examines the security issues associated with many consumer real-time messaging products and provides insight into how Sprint Enterprise Application Messaging addresses these issues within the confines of satisfying corporate and regulatory requirements for a secure real-time messaging architecture.

---

**Highlights**

---

***The challenge faced by Sprint's MCS team was to extend applications to pervasive devices in a secure and managed fashion...***

## 1. Introduction

Sprint's achievements in the mobile telecommunications landscape has presented an opportunity to Sprint as well as other service providers and Internet developers to securely extend corporate applications to pervasive devices. To address this opportunity Sprint has organized a team of talented software developers and consultants in its Mobile Computing Services (MCS) group. Recognizing that corporate partners, employees, and customers present a heterogeneous wireless environment, Sprint's MCS team has focused its attention on mobility solutions that are secure and interoperable with other carrier networks as part of the overall Sprint solution suite.

The challenge faced by Sprint's MCS team was to extend applications to pervasive devices in a secure and managed fashion while making the application delivery experience consistent from device to device. The conceived solution had to avoid loss of application functionality and introduction of new risk exposures to the enterprise. Sprint responded to this challenge by introducing the Sprint Enterprise Application Messaging framework and architecture. Sprint's application messaging architecture extends real-time messaging to enterprise applications and mobility. Suitability of real-time messaging for application delivery and mobility stems from several considerations that include:

- Presence awareness, which enables real-time collaboration and alerting,
- Simplified text- or forms-based user interface, which can be extended consistently to many device form-factors,
- Ease of integration with business applications,
- Polymorphism, or the ability to leverage the same infrastructure for multiple purposes, which helps provide a consistent method for application delivery and management across all platforms at a manageable cost.

Real-time messaging has been around for many years as a consumer product that was introduced by Internet Service Providers (ISPs). However, due to lack of security and corporate control, Consumer Instant Messaging (CIM) is not suitable for business or corporate communications. An enterprise real-time messaging solution requires a framework capable of traversing both public networks and private messaging domains while satisfying the requirements and traits that IT organizations are looking for in an enterprise class solution. These requirements include:

- Ease of integration with business applications
- Mobility while avoiding security gaps
- Centralized administration
- Message encryption
- Directory and infrastructure integration
- Archiving, logging, and compliance with regulatory requirements
- Interoperability and access to third party consumer IM networks
- Reporting and analysis
- Ability to support policies for access and usage
- Virus protection
- Billing and usage tracking

---

## Highlights

---

These enterprise requirements have motivated development of Enterprise Instant Messaging (EIM) solutions by independent software vendors (ISVs), some of which have only recently been announced. Examples include Microsoft's® Real-Time Communication Server (RTC Server), AOL® Enterprise AIM® Services, Lotus® Sametime, and Yahoo!® Messenger Enterprise Edition. These EIM solutions extend real-time person-to-person instant messaging to the enterprise by incorporating additional security, archiving, reporting, and management functions. However, in their rush to introduce enterprise solutions, ISVs have limited their attention primarily to the same set of real-time messaging features that has made consumer IM solutions a success, leaving a significant gap in the areas of enterprise *application messaging* and *mobility* – two areas where Sprint sees significant demand and opportunity.

Synopses of some EIM developments follow:

- AOL's Enterprise AIM Services. AOL uses a gateway that can be installed behind a firewall, or hosted as a subscription service. Enterprise AIM Services allows businesses to log IM message sessions and set up naming conventions. The user's identity can also be tied back to the corporate directory.
- Yahoo! Messenger Enterprise Edition. This product enables users to have the freedom to communicate with users outside the corporate firewall as well as the 20 million users of the consumer Yahoo! Messenger product. Yahoo! Messenger Enterprise Edition employs 128-bit encryption and secure socket layer (SSL) so that only intended recipients can read messages. This product also authenticates users against the corporate directory.
- Microsoft Real-Time Communications (RTC) Server. This product offers enhanced security, manageability and logging of real-time messaging. RTC Server integrates with Microsoft's Active Directory for user identification and authentication.

***Sprint in its continued pursuit of business critical applications for mobile devices, has successfully formed the foundation of an "executable internet" that brings people and applications together in an anywhere-anytime paradigm.***

Sprint's application messaging framework incorporates most features found in the above products. However, Sprint's architecture does not require replacement of EIM solutions where they already exist. Rather, it provides the flexibility to securely integrate other EIM solutions into its application messaging architecture. This has placed Sprint on the path to forge new partnerships with EIM solution providers for delivery of comprehensive enterprise messaging solutions that extend to applications and mobility.

In its continued pursuit of business critical applications for mobile devices, Sprint has successfully formed the foundation of an "executable internet" that brings people and applications together in an anywhere-anytime paradigm.

The details of Sprint's application messaging architecture are presented in a companion white paper entitled, "*Sprint Enterprise Application Messaging, A Reference Architecture and Framework*," which provides a detailed discussion of the concept of application messaging and the issues related to extensibility to applications, mobility, interoperability, reliability and scalability. This document refrains from similar discussions and instead focuses on how Sprint's architecture eliminates many of the security risks that are associated with consumer messaging products.

---

**Highlights**


---

***If unmanaged, the use of CIM poses severe risks that can eventually be deleterious to an enterprise's productivity, security and public image.***

***"Real-time messaging security was identified as one of the "Top 11 Security Issues for 2003."***

## 2. Risks Associated with Consumer Messaging Products

Since most corporations have not yet implemented EIM solutions, corporate employees are taking individual initiatives to use consumer IM products for communication with colleagues and business partners. This ad-hoc introduction of CIM in the enterprise could result in serious security consequences. If unmanaged, the use of CIM poses severe risks that can eventually be deleterious to an enterprise's productivity, security and public image.

During a recent Gartner Symposium/ITxpo, real-time messaging security was identified as one of the "Top 11 Security Issues for 2003." With more than 25 million business users of real-time messaging in the United States, its security is fast becoming a top priority for corporate IT departments.

Most consumer real-time messaging architectures depend on a client-server architecture for authentication but a peer-to-peer architecture to send and receive messages. The client is installed on a computer by an end-user and is the interface that is used to communicate with others. The server manages and relays all client/user communication and is maintained by a service provider. The server is responsible for delivering messages to the intended recipients, and is also responsible for authenticating users and verifying their online status. Consumer IM products could create great vulnerabilities in the defensive perimeter set up around corporate networks. For example, unlike corporate e-mail that remains on corporate servers and never leaves the enterprise network, a conversation over a CIM network requires that unencrypted messages be sent to the CIM server hosted by a service provider and then forwarded to the intended recipient. This holds true even if both users are within the same enterprise and protected by the same firewall. Once the conversation leaves the corporate firewall, it could become subject to eavesdropping.

Following are some of the key risk exposures of consumer IM products:

- Unencrypted messages (messages sent in plain text)
- Copyright and intellectual property infringement (company knowledge distributed without corporate permission)
- File transfers and file sharing
- Known worms and viruses
- Known client buffer overflows
- Social engineering (seeking privileged information through people's behavior or habits)

**Table 1 – IM Network Threat Matrix<sup>1</sup>**

<b>RISKS</b>	Unencrypted Messages	Copyright Infringement	File Transfers & Sharing	Desktop Remote Control	Known Worms & Viruses	Known Buffer Overflows	Social Engineering
AIM	☑	☑	☑		☑	☑	☑
.NET	☑	☑	☑	☑	☑		☑
Yahoo!	☑	☑	☑				☑
ICQ	☑	☑	☑			☑	☑

<sup>1</sup> "Risk Exposure Through Instant Messaging And Peer-To-Peer (P2P) Networks," Internet Security Systems, April 2002

---

## Highlights

---

***Clients for certain IM networks are designed to help evade filtering and policy control. Users of these services are unknowingly putting information about themselves or their enterprise at risk.***

These risks and others associated with consumer real-time messaging clients are outlined above, in Table 1, as they apply to common real-time messaging networks.

Clients for certain consumer real-time messaging networks are designed to help evade filtering and policy control. Users of these services are unknowingly putting information about themselves or their enterprise at risk. The three major consumer real-time messaging vendors (AOL, Yahoo!, and Microsoft) have had issues with privacy violations and well-publicized security holes that leverage architectural and programming flaws. Architectural flaws, for example, include the pervasiveness of real-time messaging solutions and their ability to bypass firewalls, and anti-virus scanners. Programming flaws include susceptibility to buffer overflows, Trojans, and worm-like propagation.

- **Opening holes in firewalls**

Any user who has Internet access could route his traffic through an external proxy server to bypass a firewall on any non-restricted port from the internal corporate network. Since many CIM solutions allow users to select the ports to use (such as port 80), it's next to impossible to restrict unauthorized outbound traffic by port number. One potential solution is to deploy a network-based intrusion detection system (NIDS) to watch for and reset unauthorized IM traffic.

- **Circumventing gateway Anti-Virus**

Similar to the way e-mail transmits attachments, many IM applications can sneak files past perimeter security devices as attachments. Since the peer-to-peer tunnel goes directly to the desktop, infected files riding on consumer IM clients slip past gateway anti-virus scanners. Unless the desktops have active scanning, consumer IM clients can easily introduce viruses and worms.

- **Hacker Intrusions**

Hackers can use consumer IM vulnerabilities to gain access to workstations, using the host as a jumping-off point to other parts of the network. For example, a recent AIM buffer-overflow flaw enabled hackers to take remote control of a target. From there, the hacker could do basically anything the workstation owner could do on the network.

- **Weak or deficient encryption**

Authentication credentials and session data must always be encrypted. While many vendors say they secure authentication credentials, far fewer are addressing session security. All four of the most popular consumer IM applications transmit messages in plaintext. This offers plenty of opportunity for unauthorized viewing via network sniffers and special-purpose data capture programs.

Various consumer real-time messaging solutions will claim different features from their competitors, but all work relatively the same way and share many of the same security issues. Given the pervasiveness of the big four, here is a look at their architectures and inherent security issues as well as a table on how to block particular services offered by the real-time messaging clients:

- **Yahoo! Messenger**, in terms of security, is the weakest of the major IM solutions. No encryption or hashing is used to protect user names and passwords during the login/authentication or session processes. If proxy servers are used, users' logon credentials are stored unencrypted in the proxy logs, making them freely available to anyone with log file access. Yahoo! Messenger

---

## Highlights

---

also maintains local logs by default, so anyone with local file-system access to a user's desktop may be able to view a user's chat dialogues.

- AIM** is a client/server application that includes real-time messaging, voice/video chat, file transfers and sharing, remote assistance (remote control of someone's system), application sharing and chat room capabilities. The AIM network comprises two kinds of servers: Open System for Communications in Real-time (OSCAR), which handles user authorizations; and Basic OSCAR Service (BOS), which provides the search tools for users to find each other. AIM uses FLAP when authenticating to OSCAR servers. FLAP is AOL's proprietary protocol that uses the open-source algorithm XOR to encrypt user screen names and passwords. Upon successful login, a cookie is issued that grants users access to the various BOS servers for the life of the session. When users sign out, the cookie is invalidated and all of the user sessions are disconnected. After the session is initiated, communications are transmitted in plaintext. AIM users can make use of various proxy solutions-including SOCKS 4, SOCKS 5, HTTP and HTTPS--to secure IM sessions.

**Table 2 Real-time Messaging Solutions - How To Block Them?**

Real-time Messaging Solution	Services Offered	How To Block The Service?
AOL Messenger	Real-time Messaging, Voice / Video chat, File transfer & File sharing	Default use TCP port 5190, but cannot be blocked since the app uses any available port
	Sending/Receiving images	Block in- and outbound TCP port 4443
	Block All Services of this App	Block access to login.oscar.aol.com
MS .NET Messenger	File transfer and File sharing	Block in- and outbound TCP port 6891
	Real-time Messaging, Voice / Video chats	Block UDP ports 13324 and 13325
	Application sharing	Block TCP port 1503
	Block All Services of this App	Block all TCP port 1863 access to hosts within the msgr.hotmail.com sub-domain
Yahoo! Messenger	Instant Messaging	Block in- and outbound TCP port 5010 for file sharing and file transfers
	Block All Services of this App	Block access to all hosts within *.msg.yahoo.com sub-domain
ICQ Messenger	Real-time Messaging	Block in- and outbound TCP port 5190
	File transfers	Block TCP port 3574
	File sharing	Block TCP port 7320
	Block All Services of this App	Block all TCP port 5190 access to login.icq.com

- Microsoft .NET Messenger** makes use of a seemingly universal suite of server farms--all of which can authenticate .NET clients. Users can direct their traffic through various proxies to secure sessions. During the user logon session, the

---

## Highlights

---

.NET server generates a seed value (unique string of letters and numbers), which it sends to the client. The client then appends its unique 16-character value to the server's seed, hashes this 32-character result via MD5 and sends it back to the server for authentication. After authentication, all messages are transmitted in plaintext. As with the other popular consumer IM solutions, any direct connections such as file sharing, file transfers and voice/video chats expose one's client IP address and domain name.

- **ICQ** offers relatively the same services as the more popular solutions. The ICQ protocol and encryption algorithm are proprietary and, for the most part, cannot be easily assigned to specific ports. During the authentication sequence, the client sends its Unique Identification Number (UIN) and the user's password, encrypted by ICQ's proprietary algorithm, to the authentication server. Once authenticated, all peer-to-peer communications flow directly between the clients in plaintext. Like many of its competitors, ICQ can use proxy servers to secure IM sessions.

### 3. The Urgency to Secure Real-Time Messaging Connections

***By the end of 2003, there will be nearly 200 million unique real-time messaging users in the workplace.***

A recent Gartner Group research stated that in the United States over fifty-percent of businesses are using real-time messaging, but more importantly less than two percent are managing its use. By the end of 2003, there will be nearly 200 million unique real-time messaging users in the workplace. This rapid adoption and pervasive use of real-time messaging in the enterprise suggests that it will become a business-critical communications tool that is widely accepted by CxOs and IT departments. Predictably, the majority of workplace real-time messaging use today occurs via consumer IM networks, rather than managed and secured enterprise-class solutions. The security and manageability challenges introduced by consumer IM networks into the corporation must be addressed immediately. This is especially true for regulated entities, where security, reporting, privacy or auditing regulations necessitate a more aggressive security-focused approach to the use of real-time messaging systems.

Securing the corporate collaboration environment from end to end is not a simple or straightforward task. But it is critically important for any organization that hopes to survive and succeed in this volatile and security-conscious post-9/11 world. That's why many enterprises have stepped up their evaluations of secure messaging products. Proactive organizations aren't waiting for lawyers, regulators, and politicians to tell them they need to get serious about security.

Is there harm associated with consumer IM usage in the enterprise? Potentially yes, and the risk increases significantly not managed properly. The biggest concern revolves around consumer products inability to provide archives and indexes of messages. That's particularly important in industries where business is done electronically. In fact, in industries such as financial services, telecommunications, healthcare, and energy, agencies such as the Securities and Exchange Commission (SEC) have already begun to alert companies that they need to retain, archive and index real-time message transcripts in much the same manner that they manage e-mail.

That kind of scrutiny is causing some enterprises to ban real-time messaging use, at least for now. IT managers are beginning to take a harder look at the limitations of consumer messaging products and demanding features such as audit trails. Therefore, companies are turning to EIM products tailored for enterprise users from vendors such as Sprint, AOL,

---

**Highlights**


---

***A robust instant messaging framework, must assume that security threats from within the organization are just as likely as the external ones.***

Microsoft, Yahoo!, WiredRed Software Corp., NetLert Communications Inc., FaceTime, and Lotus Development Corp. Such products solve some of consumer IM's security problems by moving the messaging server inside the enterprise firewall and keeping messages off the public Internet. Most add encryption, while some also add administrative features such as archiving and indexing of messages.

Security of an enterprise messaging architecture begins with its hosting options. Most security professionals will urge that the core real-time messaging infrastructure be housed entirely inside the corporate firewall, or hosted wholly in a secure data center. Each of these scenarios presents varying security needs that must align with organizational goals. In addition to business issues, technical considerations include integration with other IT infrastructure components such as RADIUS, LDAP, SMTP, and VPNs. Isolation from critical applications to enforce internal segregation policies, server certificate use, firewall configuration, and dedicated network access options are also among the requirements to consider. A flexible application messaging framework and architecture must address the secure handling of sensitive corporate information with the use of appropriate port handling, hardened environments, and associated policies and procedures.

Following are other areas of concern that all enterprises utilizing real-time messaging or considering integration of an application messaging infrastructure need to be keenly aware of:

### **3.1. Presence**

The fundamental questions presence answers is

- Who is online?
- Who is off-line?
- Who can answer my questions right now?
- Who can I interact with right now?

There is real value in knowing, in advance, whether a person or application is available to respond to a question. Imagine the example of an unscheduled call to someone, they aren't there and you have to leave voicemail for a call back. Westsphere Equity, a financial firm with locations in the US and South America, has a policy which states that employees must check their "buddy" lists to see if a colleague is on-line before placing long-distance telephone calls. This policy is saving the firm ten thousand dollars per year in reduced long distance phone charges.

As presence awareness and real-time messaging become a core component of a company's communications infrastructure it could allow hackers to exploit social engineering tactics (conversation or requests) to gain user-ids and passwords. This threat should be addressed through appropriate corporate usage policies and training.

### **3.2. Vertical Industry Regulatory Considerations**

To meet government mandated regulatory criteria, Sprint's Enterprise Application Messaging framework captures and stores internal and external messages sent from or received by users, whether between two Sprint application messaging clients, or between Sprint clients and other IM network clients. Sprint's framework also records the sender and recipient addresses (i.e., IM account names) and times/dates of correspondence in their original format. All of the information is then indexed by

---

## Highlights

---

time/date and user name. In order to comply with regulatory specifications, Sprint's framework includes an administrative tool that allows authorized users to search the message archives by user name, time/date, and keyword parameters. These search results can be used to create standard or customized reports as needed. The administrative tool is accessible to authorized users through a secure interface.

### **3.2.1. Finance**

The real-time messaging systems of financial services firms must comply with the Gramm-Leach-Bliley Act (GLBA) and the Patriot Acts, SEC Rules 17a-3 and 17a-4, as well as the National Association of Securities Dealers (NASD) Conduct Rule 3010 that specify how electronic communications must be retained, reviewed, and supervised. These regulations are designed to ensure that correspondence between financial service firms and the public are recorded and stored in a way that guarantees integrity and accessibility by governing bodies for inquiries and investigations.

### **3.2.2. Healthcare**

The health care industry is adapting to ensure compliance with the Privacy Rule, a key component of the Health Insurance Portability and Accountability Act of 1996, or HIPAA. The HIPAA Privacy Rule requires all organizations that handle patient health information (PHI) to put in place policies, procedures and technical measures to ensure that only authorized individuals have access. Any occurrence where there is an unauthorized leak of information making any patient identifiable as an individual is a breach of the Privacy Rule.

Enterprise messaging solutions need to ensure extensive visibility to PHI violations generated by employees or contractors.

### **3.2.3. Government**

Real-time messaging now faces the same retention requirements as e-mail. This will be a compliance mandate for government usage regulated by the Freedom of Information Act, Open Records Act, Patriot Act and widely applicable directives such as DITSCAP, NIACAP and GISRA.

## **3.3. File Transfer and File Sharing**

File sharing is a key collaborative feature of application messaging. Sending the document by e-mail is subject to delivery lag time caused by workload of the e-mail servers involved. With application messaging, files can be shared or transferred more flexibly and conveniently. But this poses a problem from a security perspective. Similar to e-mail, real-time messaging must provide a mature and robust environment from which logging and rule sets on the server can monitor and log content including documents, thereby limiting an enterprise's exposure to viruses. The Sprint Enterprise Application Messaging framework provides a robust mechanism for enabling virus scanning during file transfers and just as importantly logs and archives all file transfer records to ensure compliance with all regulatory requirements.

## **4. A Secure Enterprise Application Messaging Framework**

Security is about managing risk, not eliminating it. Secure collaboration clearly depends on a pervasive security framework within the application and networking infrastructure, including strong authentication, role-based access control, content confidentiality, message non-repudiation, message content filtering, and activity/message logging.

---

## Highlights

---

A robust application messaging framework must assume that security threats from within the organization are just as likely as external ones, and should address rudimentary areas of security concern from an administrator's perspective. These areas include:

- Directory Integration and Access Control,
- Authentication and Identification,
- Encryption,
- Authorization & Privacy,
- Central Management,
- Archiving Auditing and Reporting,
- Policies and Procedures.

Few vendors provide products or services that specifically focus on securing mobile messaging environments. Rather, security on mobile messaging is generally a byproduct of security features integrated with the underlying messaging, presence, application, and networking environments that users are accessing from handheld devices over wireless connections. Sprint has taken pains to ensure all understood security risks are addressed in the Sprint Enterprise Application Messaging solution.

### ***4.1. Directory Integration and Access Control***

Directory Integration solutions should automatically integrate, consolidate and synchronize disparate user identity data into a single centralized repository, enabling the enterprise to leverage existing directory investments and maintain accurate, consistent data across the organization. Sprint's messaging architecture can integrate with industry standard directory services such as Microsoft's Active Directory<sup>TM</sup> or SUN ONE Directory Server<sup>TM</sup> and directory access protocols, such as the Lightweight Directory Access Protocol (LDAP), for managing and accessing user profiles and authentication information.

This level of integration provides the capability to:

- Authenticate against existing directory infrastructure
- Eliminate multiple passwords and facilitate single sign-on
- Create, maintain, and delete user profiles
- Perform lookups for inclusion of individuals or applications in "buddy lists"

### ***4.2. Authentication and Identification***

The application messaging frameworks must provide the ability to tie into enterprise authentication mechanisms. For example, secure authentication and access control frameworks such as RADIUS are in use in many organizations and supported by Sprint's architecture. Other authentication and identification requirements that are also supported by Sprint's architecture include:

- The ability to identify a user to others by a public identifier, such as the user's email address. Corporate usernames should be hidden from other users, even if they are used for authentication. Client software or devices

---

## Highlights

---

must not store private information such as corporate usernames and passwords.

- User passwords should not be stored in any form on messaging servers. If such storage is required, encryption or one-way hash functions must be used.
- User passwords, or hashes that may be substituted for them in authentication schemes, should not be transmitted in unencrypted form. A challenge-response authentication scheme whereby passwords are never transmitted is preferred.
- A provision should exist for profiles used to set user parameters and preferences to be stored on corporate directories. An independent check of the mapping between profile and user should be performed at login, e.g., lookup of profile by email address, and match to username (used for authentication) as listed in profile.

### 4.3. Encryption, Authorization and Privacy

The premise behind message encryption is to maintain the privacy of messages by preventing unauthorized access from both inside and outside the corporate firewall. Message encryption and privacy is completely absent in the consumer IM frameworks, leaving messages susceptible to eavesdropping and interception.

In general a conversation between two application messaging clients can be encrypted in two ways.

*...conversation between two application messaging clients can be encrypted in two ways.*

- *Peer-to-Peer*
- *Hop-by-Hop*

- Peer-to-Peer – where data is encrypted between the two clients and can only be decrypted by the clients.
- Hop-by-Hop – where data is encrypted between the client and the message server. In this model a message received by the server is decrypted using the sender's key and re-encrypted in the recipient's key before being transmitted to its final destination.

In order for the message server to archive a message it must have access to a clear-text copy of the message. It can then encrypt the message using its own public key or symmetric ciphers for archival. If the clients were to establish a peer-to-peer encrypted session, the messaging server would not be able to archive a usable copy of the message unless extraordinary methods were employed such as saving a second encrypted copy of the message using the server's key. This is similar to two e-mail clients using PGP (Pretty Good Privacy) to exchange messages.

With respect to privacy needs, we can generalize real-time messages into three basic categories.

- Secured – no one, not even the administrative and security personnel should have access to secured messages. Examples include, user and application passwords.
- Private – data exchanged between users and/or applications that are meant for private consumption of those entities. When needed, authorized corporate personnel or application agents should be able to access the information to ensure compliance with regulatory as well as corporate policy requirements. Preferably, any private data should also be retrievable by

---

## Highlights

---

those involved in the conversation. Private messages should be transmitted and stored in an encrypted format.

- Public – These are messages that are intended for consumption by the corporate community, and therefore may not require privacy protection. Examples include training sessions, or general corporate communications.

Peer-to-peer encryption is suitable for “secure” messages such as application passwords, since this information must be protected at all cost. For “private” and “public” messages, hop-by-hop encryption is desirable, since it would facilitate access to messages for archiving, policy enforcement, and reporting.

It is possible for an application messaging infrastructure to include more than one message server, each responsible for a domain or a group of applications and users. The message handoff between messaging servers should also be encrypted, as should be any communication between application and messaging servers.

For “authorization” and “privacy”, message traffic between users could be controlled through user and group permissions. The ability to manage group information from a central point, such as the corporate directory, is a basic feature that gives administrators the ability to:

- Authorize or restrict access to presence information on an individual or group basis.
- Restrict other users or groups from sending messages.
- Restrict viewing of non-required profile information by users or groups.

### **4.4. Central Management**

Unlike consumer-oriented messaging products, Sprint’s architecture provides the administrator with full control over all aspects of user communication such as a definable recipients list for each group of users, sending and receiving permissions, attachment control, and other vitally important settings and restrictions. Administrators also have the capability to monitor logs, receive and review incident reports, and implement corporate usage policies.

### **4.5. Archiving, Audit, and Reporting Capability**

Maintaining archives and logs of application messaging conversions is mandatory for most organizations. Just as e-mail archives allow for enforcement of e-mail usage policies, as the user population transitions to real-time messaging, similar policies must be implemented.

Archiving should take place because:

- Industry specific federal mandates such as HIPAA for healthcare institutions and the GLB Act for financial organizations must be complied with as well as the Patriot Act which is applicable to all organizations.
- As IM conferencing becomes more popular, it is vital that the conferencing sessions be captured to retain corporate knowledge.
- Compliance with corporate usage policies must be enforced. For example, sessions established by suspected employees or contractors could be monitored, or logs could be scanned for questionable keywords or phrases.

---

## Highlights

---

Since messages may contain private or sensitive data, the archive engine should log the information in an encrypted format so that it is only retrievable through the appropriate administrative and security channels. This requires a tight integration of the archive engine with the “reporting and analysis” and “message encryption” engines.

Messaging servers must log messages and usage statistics both anonymously and by user. All “secure” and “private” messages must be encrypted before archiving, and any such encryption feature should represent a genuine security improvement (e.g., decryption key should not be stored on same server with encrypted messages). File transfer events must also be logged.

A robust application messaging framework should provide the necessary reporting and analysis application agents to replay, manipulate, analyze, and summarize the archives. Real-time messaging significantly improves a company’s ability to meet reporting and compliance requirements as compared to other modes of communication such as telephone conversations. Reporting and analysis is a defensive strategy that ensures compliance with corporate usage policies and regulatory requirements.

When used for policy enforcement, the reporting and analysis engine must be as automated as possible to minimize human interaction, which is the weakest link in any security framework. For example, Sprint’s application messaging architecture utilizes robots or application agents to scan the logs for phrases and generate reports only when violations are detected. In general reporting tools include:

- Application agents for scanning the logs
- Report generators for capturing application messaging sessions
- Administrative tools for customizing and analyzing reports

***For Sprint, interoperability meets two objectives. First, it enables Sprint’s customers to communicate with their partners and clients who may be using a different EIM platform. Second, it facilitates coexistence of Sprint Enterprise Application Messaging with other EIM platforms within the same enterprise, therefore allowing customers to leverage the enhanced application messaging and mobility capabilities of its architecture to complement any existing or planned EIM solutions.***

#### ***4.6. Interoperability & Integration with Business Applications***

For Sprint, interoperability meets two objectives. First, it enables Sprint’s customers to communicate with their partners and clients who may be using a different EIM platform. Second, it facilitates coexistence of Sprint Enterprise Application Messaging<sup>SM</sup> solution with other EIM platforms within the same enterprise, therefore allowing customers to leverage the enhanced application messaging and mobility capabilities of its architecture to complement any existing or planned EIM solutions. Sprint’s application messaging architecture utilizes interoperability gateways that facilitate presence awareness, session initiation, protocol conversion, and encryption for connecting to third party IM platforms. To facilitate protocol conversion and translation, the message gateways will act as security checkpoints where the payloads will be decrypted, re-encapsulated into a new protocol and encrypted again for transmission to the messaging server.

All of the above becomes extremely important as the application messaging framework is securely extended to real world business applications such as Help Desk, Customer Service and Sales Force Automation.

---

**Highlights**

---

#### **4.7. Policies & Procedures**

Mobile computing opens up a host of new possibilities for companies, but it also poses a unique set of risks. According to experts, the best way to mitigate those risks is through careful attention to mobile security policy. Such policies allow companies to ensure mobile devices are used in a safe and appropriate manner. Policies that can be enforced either with technology, such as tools that ensure that strong passwords are used, or through employee conduct, will need an approach that ensures user education and compliance.

Application messaging and mobile security compliance should be monitored, maintained, and enforced by a policy management framework. Initially policy management may not appear to be a critical issue, however, once real-time messaging is extended to corporate applications and customer support its importance will become apparent.

The application messaging policy management framework should support three classes of policies that include:

- Individual Policies – These are user communication preferences, “buddy lists”, presence visibility to others, disallowed lists, mobile preferences such as weekdays and times that mobile notification is allowed, and more.
- Security Policies – These are access control, encryption, group segregation, archive & logging, and other security related policies. For example, consider a healthcare or financial institution where the customer support representative requires the ability to query customer information while at work, but not outside the work hours. The policy management framework should curtail access to customer information outside the scheduled work hours.
- Usage policies – These are for monitoring compliance with corporate “usage policies” for conduct, and enforcement of communication standards with customers and partner. Real-time monitoring of customer support sessions is an example of enforcing usage policies. Policy management application agents could in real-time monitor conversations for phrases and comments that may not be in the best interest of the organization. Once a situation is detected a supervisor could be invited to join and monitor the conversation. Most users wish to respect corporate “usage policies”. To help educate users about “usage policies”, alerts caused by violation of these policies could be immediately brought to the users’ attention through the application messaging clients.

A robust policy management framework requires at least three levels of granularity for application of policies. These include:

- Individual users and applications
- Groups and Subgroups of users or applications
- The application messaging domain

The organization controls the domain, group, and subgroup policies. For example, a corporate policy may establish that all members of the “morning shift” of the “teller” group should have access to the banking applications from 7:30 AM to 5:00 PM. In general each lower level of granularity will inherit the policies established for its

---

**Highlights**


---

parent. However, the parent should have the flexibility to delegate the policy to a lower level. For example, a domain policy may establish that all communication between users must be encrypted. It could however provide the flexibility to allow groups to override that policy. A group may also allow individuals to override that policy. This level of flexibility in policy management enables organizations to balance their need for control and convenience.

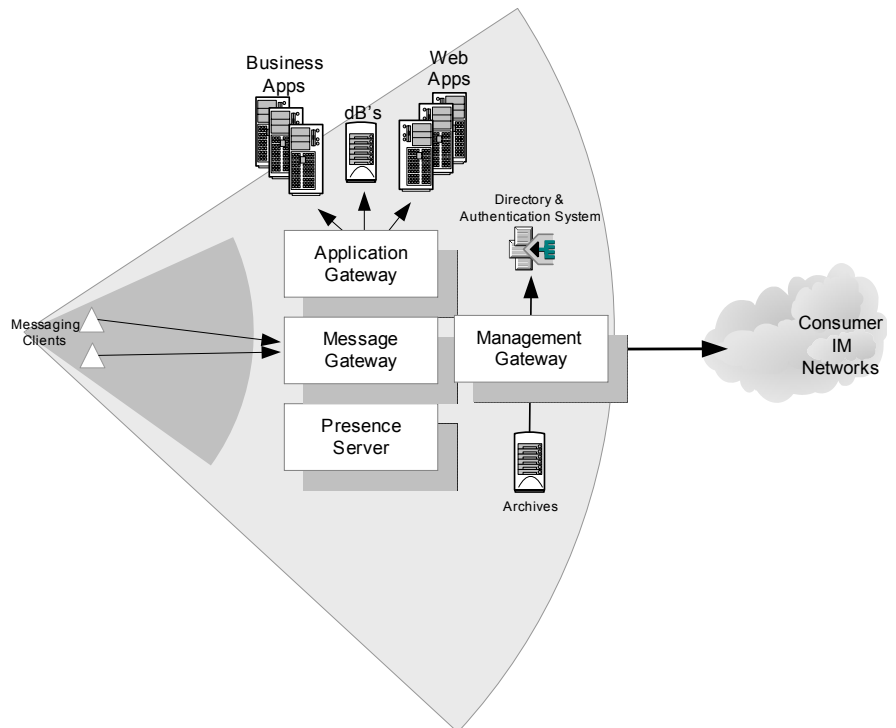
Individual users and applications will always inherit the policies applied to their groups or subgroups. At the discretion of the corporation, groups may have the flexibility to delegate some policies to the users.

Since each application messaging domain could consist of multiple servers, the policy information should be stored in a data repository and be accessible by all servers via LDAP. This architecture facilitates scalability of the environment without the need to configure each individual server. Another benefit of leveraging LDAP is that any existing directory infrastructure could be used as a repository for the policy information.

## 5. Components of the Application Messaging Framework

Figure 1 provides a high-level architectural view of the components used to create a secure application messaging solution like Sprint's Enterprise Application Messaging. This high-level view consists of seven broad categories of components: Management Gateway, Message Gateway, Directory & Authentication System, Client Software, Presence Server, Application Server and the Messaging Protocols. These components integrated into an enterprise IT infrastructure, along with strong security and user policies and procedures define a good conceptual start for a secure messaging framework.

***A managed IM infrastructure meets enterprise IT requirements for security, administration, and integration.***



**Figure 1 - Key Components of a Secure Application Messaging Framework**

---

**Highlights**

---

## **5.1. Infrastructure Components**

As the enterprise considers implementation of an application messaging environment it must approach the exercise recognizing both the user's need to connect to third-party IM networks, and more importantly IT requirements for security, manageability and integration. The following sections provide an overview of the primary components required to meet these objectives.

### **5.1.1. Management Gateway**

Enterprise real-time messaging is enabled in large part by the multiple functions performed by a management gateway, which provides a critical link between application messaging framework and the organization's administrative and IT personnel. The management gateway can be used to activate or disable features of the real-time messaging system, such as file transfer, file sharing, and encryption. It can also be used to set or change permissions, and establish "individual", "security", and "usage" policies. Another function commonly performed by management gateway is the screening of real-time messages for acceptable use policies, viruses, or inappropriate content.

### **5.1.2. Messaging Gateway**

Consumer IM traffic often travels peer-to-peer or in uncontrolled ways across an enterprise network. This poses significant management challenges for security and control, as many consumer IM clients connect on non-standard ports using proprietary protocols. A managed application messaging infrastructure centralizes all real-time messaging traffic through a single routing point, controlling the behavior of real-time messaging clients and capturing all real-time messages on a corporate network. Once messages are captured and controlled, a message gateway can increase application messaging security by directing real-time messaging traffic over particular routes, such as VPNs. Due to lack of standards for real-time messaging ID's and message formats, a message gateway is necessary to translate these items from one format to another should an enterprise decide to connect internal application messaging and external real-time messaging networks.

### **5.1.3. Directory and Authentication System**

Most enterprises have deployed a directory and authentication system for e-mail, VPN and application access. A managed application messaging infrastructure leverages the enterprise directory to provide a single point for registering application messaging screen names, and user authentication and preferences.

### **5.1.4. Presence Server**

A presence server allows an enterprise to host its own application messaging network, thereby providing presence information only to authorized entities. A real-time database keeps track of the current presence status of every person, device, or software application and provides that information to other users who may have included those entities in their "buddy lists".

---

**Highlights**

---

**5.1.5. Application Gateways**

As application messaging and presence technologies migrate more deeply into the enterprise, they will be increasingly used as a critical productivity tool for a wider range of business processes. Enterprises will extend real-time messaging beyond employee communications and into customer relationship management (CRM), supply chain management (SCM), and enterprise resource planning (ERP) systems via application gateways. Application gateways are critical components of a managed application messaging infrastructure, enabling real-time messaging to be leveraged by the existing application infrastructure.

**5.1.6. Messaging Protocols and Security**

Currently a universal real-time messaging protocol standard does not exist, though the IETF's SIP for real-time messaging and Presence Leveraging Extensions (SIMPLE) protocol, promoted by IBM and Microsoft, is a contender, as is the Extensible Messaging and Presence Protocol (XMPP), an open standard proposed by the developers of Jabber, an XML-based IM application.

Neither SIMPLE nor XMPP addresses security effectively today, and require integration of add-on security tools to complement their capabilities.

**5.1.7. Enterprise User Client**

Application messaging clients are compact text- or forms-based user interface to the real-time messaging network. Most typically, a messaging client serves as a message input and display window and allows users to set various preferences for how that client will operate. Users can view their "buddy lists" and launch a variety of communications functions on the messaging client itself.

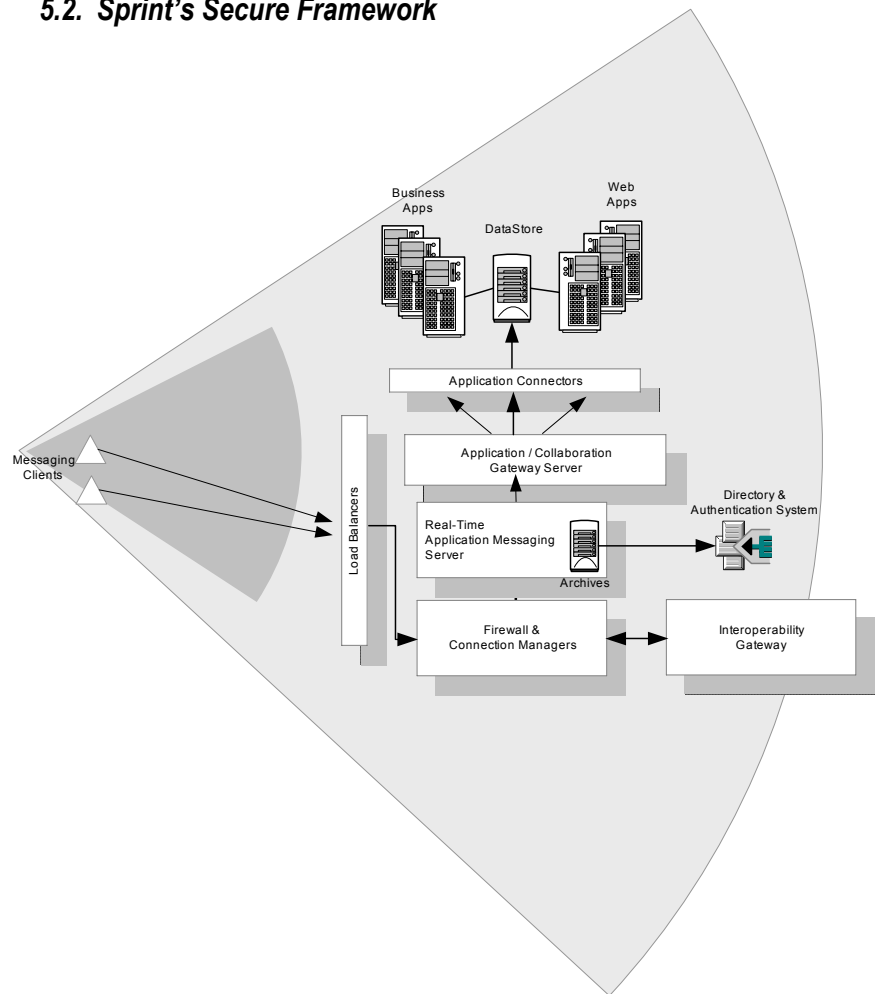
---

**Highlights**


---

- ***Sprint Enterprise Application Messaging framework uses XML switching, which can easily be extended to many applications and clients.***
- ***The core XML switch provides the ability to interact with most common protocols and APIs through component libraries and software development kits (SDKs).***
- ***Rather than creating a propriety solution, Sprint has adopted Jabber and XMPP (eXtensible Messaging and Presence Protocol) as the core building blocks for its application messaging architecture.***
- ***The biggest risk to corporations is losing the richness of application functionality. Sprint has effectively addressed this issue by inclusion of natural language processing and workflow capabilities in the core application messaging architecture.***

## 5.2. Sprint's Secure Framework



**Figure 2 – Sprint's Secure Enterprise Application Messaging**

Sprint's Enterprise Application Messaging platform for the corporate environment is a robust enterprise-class platform designed specifically to ensure user interoperability regardless of device or network. Sprint's solution offers flexible deployment configurations, so a company can install and manage a "behind the firewall" solution or outsource to a Sprint hosted site. Sprint's application messaging framework works with desktops, PDAs, and cellular telephones. Sprint's platform provides state of the art encryption of data whether on desktop, PDA, or a mobile phone. The platform will easily integrate with existing corporate network and security architectures, authentication schemas, and internal directories, allowing seamless integration. For standalone installations, the platform will be able to provide its own authentication and internal directory structures.

## Highlights

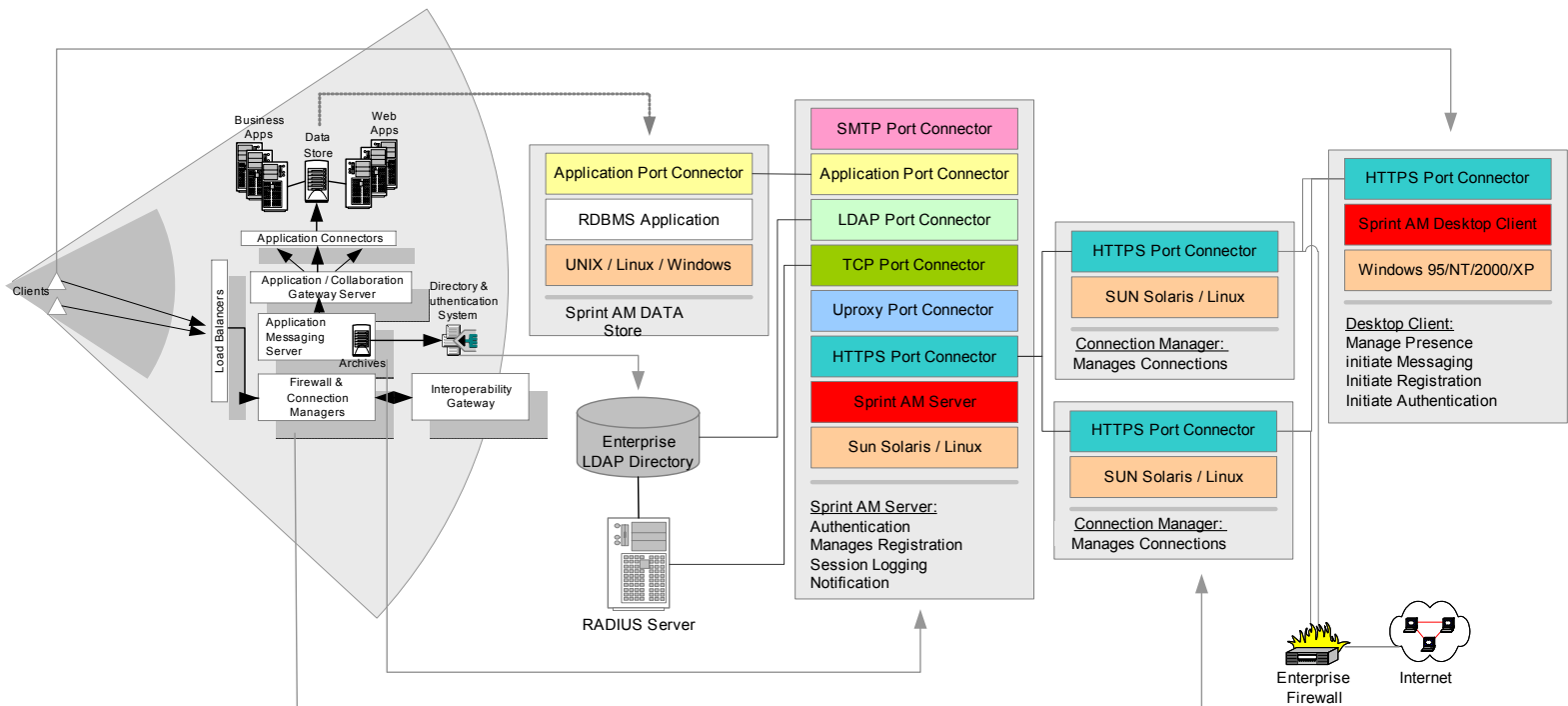


Figure 3 – A Typical Enterprise Application Messaging Configuration

Sprint Enterprise Application Messaging is based on a client-server architecture, where the predominant responsibility for extensibility to business applications lies with the messaging server. Sprint Enterprise Application Messaging framework uses XML switching, which can easily be extended to many applications and clients. The core XML switch provides the ability to interact with most common protocols and APIs through component libraries and software development kits (SDKs). Rather than creating a propriety solution, Sprint has adopted Jabber and XMPP as the core building blocks for its application messaging architecture. The biggest risk to corporations is losing the richness of application functionality. Sprint has effectively addressed this issue by inclusion of natural language processing and workflow capabilities in the core application messaging architecture.

Sprint's application messaging architecture utilizes a component-based model to simplify the integration and provide interoperability with existing enterprise systems. This model allows for easy consolidation and distribution of the various functions required for registration, authentication, administration, and messaging. This also provides for scalability as user demand grows.

---

**Highlights**


---

## 6. A Phased Approach to Enterprise Application Messaging

Enterprises that recognize the vast opportunity that application messaging provides should consider a three-phase deployment approach to minimize any undesired risks. Sprint Enterprise Application Messaging integration methodology, which includes “Assess & Control”, “Develop and Deploy”, and “Extend the Implementation” phases, is a coherent and methodical approach for extending real-time messaging to business applications and mobility.

### 6.1. Phase 1: Assess and Control

Phase 1 includes assessing and controlling existing real-time messaging usage. The objectives of this phase include:

- Establishing policies & procedures for employee access to real-time messaging
- Providing regular reports of real-time messaging network traffic and usage

Enterprises can rapidly deploy a Phase 1 solution with minimal cost, and create a managed real-time messaging infrastructure without significant deployment or administrative resources as a pre-cursor to an application messaging infrastructure. These solutions immediately decrease security risks, and allow organizations to embrace real-time messaging as an enterprise-class communications medium.

### 6.2. Phase 2: Develop & Deploy

Phase 2 includes developing and deploying a secure and managed enterprise application messaging solution with the following objectives:

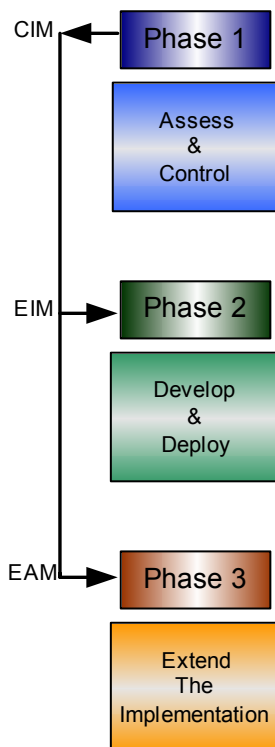
- Establishing real-time messaging infrastructure standards
- Identifying client requirements for mobile, desktop, or browser use
- Providing integration with existing directories and authentication systems to establish enterprise namespace control across all application messaging domains, and to provision user identities across private and public networks
- Establishing control over enterprise application messaging permissions and access to consumer IM services
- Logging and auditing messages
- Establishing virus protection and file transfer controls
- Providing content filtering and monitoring

Enterprises should incorporate a solution that ensures managed naming conventions across all application messaging connections.

### 6.3. Phase 3: Extend the Implementation

Phase 3 will extend the secure real-time messaging to business applications and mobility. The objectives for this phase include:

#### ***A Phased Approach To Secure Enterprise Application Messaging***



---

## Highlights

---

- Developing gateways to corporate applications that require mobilization. These gateways will allow business applications to appear as "application buddies" that represent transactions, processes, or accounts.
- Developing business solutions that are "presence-aware".
- Incorporating natural language processing and workflow capabilities into the solutions.

## 7. Conclusion

Application messaging as an evolution of instant messaging is becoming a viable business productivity tool for companies of all sizes. The core components, which include management gateway, messaging gateway, directory and authentication systems, presence server, application gateways, user clients, and, messaging protocols are now available for developers to create custom real-time messaging applications. The ability to add real-time presence to legacy applications can be a powerful enhancement, which ensures immediate communication between employees, and ultimately benefits the customers they serve.

Enterprises are being to embrace real-time messaging as a mission-critical business tool. Businesses depend on productivity gains to offset costs and competitive pressures. They simply cannot ignore one of the fastest growing technology tools that they can deploy to bring people and applications together in real-time from any device. To unleash the true power of presence awareness and real-time messaging, businesses should embark on implementing a well-designed secure messaging architecture and framework that can extend real-time messaging to applications and mobility.

When planning a real-time messaging solution, organizations should ask the following questions:

- What business, legal, or regulatory requirements must be responded to in addressing encryption, digital signatures, privacy protection, content filtering, usage monitoring, and other collaboration security concerns?
- Is there a need to establish a general-purpose identity and security environment that spans all collaborations?
- How should application messaging services integrate with identity and security infrastructures at various tiers of the organization?
- How should messaging services leverage various identity and security features in the infrastructure, including directories, PKI, user management, authentication, authorization, roles, encryption, digital signatures, smartcards, single-sign on, virtual directories, and VPNs?
- How will the organization provide physical security for all of the components of the application messaging environment, including servers, gateways, and clients (and especially mobile clients that are vulnerable to being lost and stolen)?

Sprint has made concerted efforts to ensure all identified security risks are addressed in its application messaging solution. Sprint Enterprise Application Messaging architecture provides the forethought to securely integrate application messaging into an enterprise ecosystem.

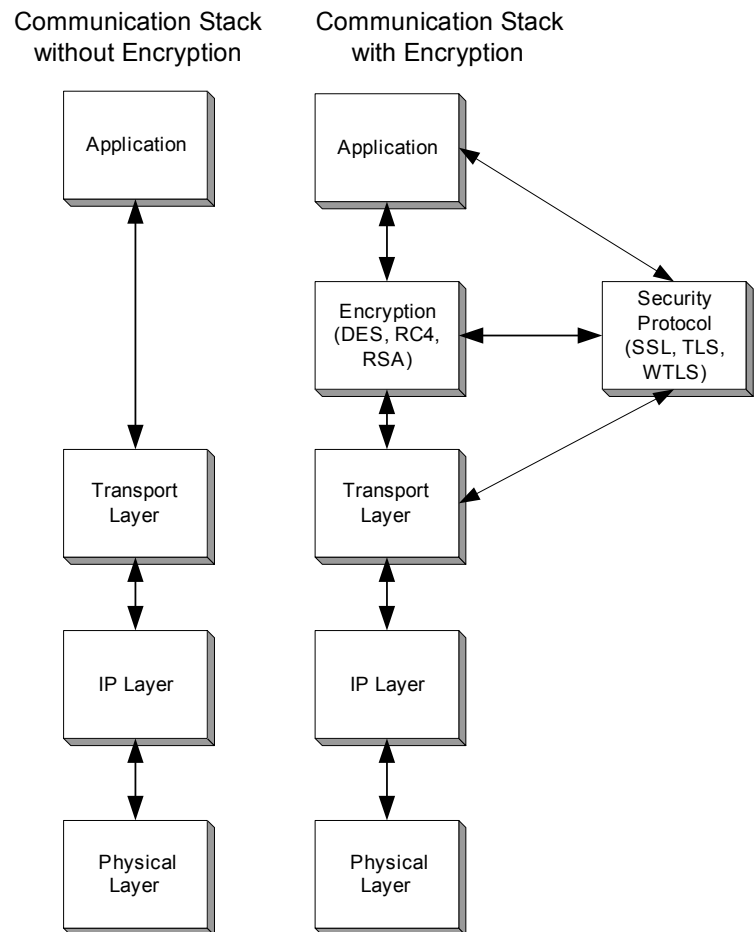
---

**Highlights**


---

## 8. Glossary: Communication Security Concepts – A Need to Know <sup>2</sup>

The communication stack isolates the different functions needed for reliable data transfer. Each layer of a protocol stack treats information passed to it by the layer above merely as data, labeling that data in such a way as to be identified and deciphered by the equivalent layer on the other computer. Only the physical layer is responsible for actually placing data onto the wire or over the air—all other layers provide some well-defined level of functionality, such as error detection, correction, and encryption and so on. Figure 4 shows a typical communication stack and the effects of added security on the architecture.



**Figure 4 – Communication Stack**

When an application needs to encrypt the data that it is sending, it is necessary to have a security protocol to establish a secure connection. Security protocols are a negotiation

---

<sup>2</sup> "Risk Exposure Through Instant Messaging And Peer-To-Peer (P2P) Networks," Internet Security Systems, April 2002

---

**Highlights**

---

***Wireless Application Messaging is a particularly compelling application for wireless users because it fully leverages the mobility aspect of wireless and inherent immediacy of communications with a text-based messaging application that closely emulates the natural language rhythms of human speech.***

(often called a “handshake”) of security parameters required to securely establish an encrypted communication session. Generally, they also provide authentication. Examples of security protocols are Transport Layer Security (TLS) and Secure Sockets Layer (SSL).

### **8.1. Public Key Infrastructure**

In Public Key Infrastructure (PKI), a certificate authority creates a certificate for the client or the server and signs it with the authority’s private key. The public key is distributed. For example, a browser may have the public key, allowing it to verify the identity and accept secure communications from anyone who has a certificate signed with the corresponding private key. The software gets the certificate from the server to verify its authenticity. In browser-based connections, only server-side certificates are supplied. The client does not have to supply a certificate to the server. The identity information must contain something that is explicitly verified by the client. For example, in a browser’s case, this is the fully qualified domain name.

In a full PKI system, there are also certificates associated with the client so that the server can verify the identity of the client, in addition to the server having certificates that the client can use to verify the server’s identity. In a PKI system there are management functions that must be considered, such as how to get certificates out to all the clients and servers that require them and what procedures should be taken if the private key is compromised (for example, preparing a revocation list of all the certificates that are compromised).

### **8.2. Symmetric-Key Cryptography**

In symmetric-key cryptography, the same key is used to both encrypt and decrypt data. This method is considerably faster than public-key cryptography. In SSL, to make communication more efficient, the client and server agree upon and exchange another key and switch to a symmetric key cipher. They use this key and cipher for the rest of their communication because the symmetric cipher allows data to be encrypted and decrypted more efficiently.

### **8.3. Stream Ciphers**

If a stream cipher is used, the cipher produces a continuous, random stream from the key that is XORed with the plain text. On the other end, the receiver produces the same random stream generated from the key and XORs the encrypted text to get the plain text. When using stream ciphers, the key should not be re-used because multiple messages encrypted with the same key afford attackers much more information to break encryption. Examples of stream ciphers include RC4 and SEAL. In SSL, a new symmetric key is generated during the handshake negotiation at the start of a session. The rest of the session is generally a continuous stream in the stream cipher sense.

### **8.4. Block Ciphers**

When using a block cipher, the cipher transforms a block of data into a seemingly-unrelated block of data of the same size. The same key can be used in different blocks because the algorithm is encrypted in ways that deliberately change for each block. Examples of block ciphers include DES, Blowfish, Twofish, AES (Rijndael), and MDSR. Block ciphers are not typically used in session-based communication. They are used for message-based communication and to encrypt data in one place, such as a database file.

---

**Highlights**

---

**8.5. SSL Protocol**

SSL is the Internet standard for secure data transfer. It allows SSL-enabled servers and SSL-enabled clients to authenticate each other in order to establish an encrypted connection. TLS is a newer and standardized version of SSL. The process of establishing a secure connection with SSL has two components:

- A handshake for protocol negotiations
- A messaging definition for data exchange

During the “handshake”, the client and the server negotiate an algorithm using public key cryptography to exchange the information: they exchange certificates in order to verify each other’s identity. (In server authentication mode, only the server has a certificate that must be verified). Up to six packets are sent before any data is transmitted. Part of the handshake involves sending random bytes from both the client and the server. These random bytes are used to generate symmetric keys and they also ensure that the session cannot be replayed because replaying the stream against the same server results in a different set of random bytes. The public key algorithm is used to exchange the symmetric key information. Once the symmetric key is exchanged, it is used to encrypt all the packets that go back and forth between the client and the server. Each message that is sent maybe signed to prevent modification of the packets that are being exchanged because even though the packets are encrypted, no one should be able to modify the packets.

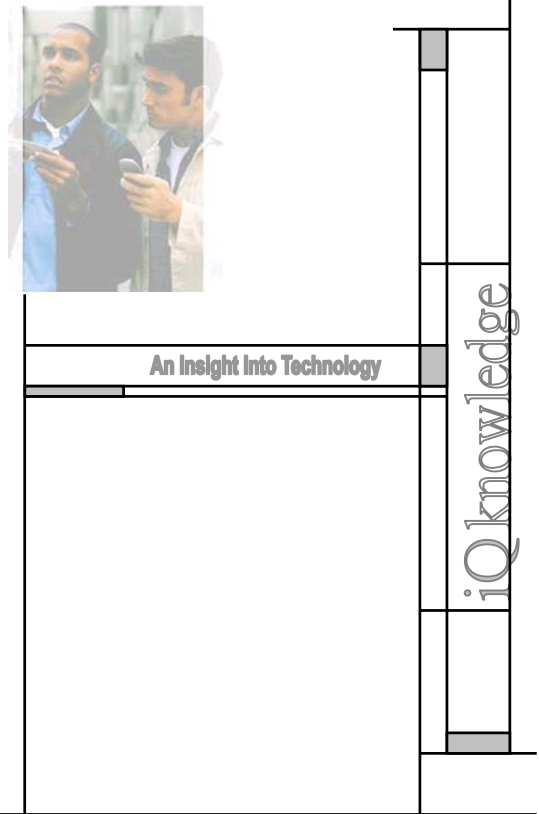
**8.6. IPSEC Protocol**

IPsec is a suite of standard protocols that provides security services for Internet communications. It protects the entire IP datagram in an “end-to-end” fashion; no intermediate network node in the public Internet can access or modify any information above the IP layer in an IPsec-protected packet. However, recent advances in Internet technology introduce a rich new set of services and applications, like traffic engineering, TCP performance enhancements, or transparent proxying and caching, all of which require intermediate network nodes to access a certain part of an IP datagram, usually the upper layer protocol information, to perform flow classification, constraint-based routing, or other customized processing. This is in direct conflict with the IPsec mechanisms.

**9. References:**

- “Risk Exposure Through Instant Messaging And Peer-To-Peer (P2P) Networks,” Internet Security Systems, April 2002
- “Mobile Insecurity, A Practical Guide To Threats and Vulnerabilities,” Bluefire Security Technologies, January 2003
- “What’s New – Instant Messaging,” IBM Corporation, Fall 2002
- “Mobile Messaging platforms Scope Out What’s Ahead,” InfoWorld, March 7, 2003
- “Mobile Data Security,” Sybase Corporation, August 2002
- “The CIO’s Guide to Wireless,” Synchrologic, April 2003

InQuest Corporation is pleased to present iQknowledge®, a series of whitepapers to assist IT Professionals in making good technology decisions to support their business needs.



**InQuest Corporation**  
5137 Belle Drive  
Metairie, LA 70006 USA  
Tel: 504-456-7380  
[www.inquest-corp.com](http://www.inquest-corp.com)

This document may be reproduced and distributed in whole only when it includes the cover page and this notice. Any reproduction, use, appropriation, or disclosure of this information, in part, without the specific prior written authorization of InQuest Corporation is strictly prohibited.

Copyright © 2003 InQuest Corporation. All rights reserved. Unpublished rights reserved under U.S. copyright laws. InQuest, iQ Knowledge, and InQuest logo are trademarks of InQuest Corporation. All other trademarks are property of their respective owners.