

# WHITE PAPER

Jabber, Inc. :: 1899 Wynkoop Street, Suite 600 :: Denver, CO 80202 :: P: 303.308.3231 :: F: 303.308.3219 :: E: info@jabber.com :: W: www.jabber.com

## THE POWER OF PRESENCE™

### INTRODUCTION

Presence is a key technology powering the real-time Internet. By providing information about the availability of people, devices, and applications, presence data enables organizations to deploy innovative real-time services and achieve significant revenue opportunities and productivity improvements. Forward-thinking businesses, service providers, and government agencies are already reaping the benefits of presence-enabled information exchange. This paper explores the nature of presence data, its value to the organization, and methods for leveraging presence to create strategic value and competitive advantage.

### THE NATURE OF PRESENCE

What is presence? At the most basic level, presence is simple, on-off information about the network availability of a person, device, or application. Presence is, therefore, a core component of an entity's real-time identity. But presence does not exist for its own sake. Instead, it functions as a catalyst for communication—the purpose of presence is to signal availability for interaction over a network.

#### Pres-ence (noun)

The availability of a person, device, or application for interaction over a network.

What kinds of entities have presence? Certainly people—presence technologies enable a wide variety of real-time interactions, including instant messaging (IM), multi-user chat (text conferencing), Voice over Internet Protocol (VoIP), whiteboarding, gaming, and application sharing. Presence information about people makes organizations run more efficiently through faster workflow processing, reduced voicemail and email delays, and the ability to find the right person at the right time. It also enables service providers to deploy applications that drive network adoption, such as push-to-video (e.g., when a teenager at a party can see that her best friend is online, she may be more likely to share a video clip of what's happening).

But presence is not limited to people. Phones, conference rooms, applications, Web services, routers, firewalls, servers, appliances, buildings, devices, and more can be networked for presence-enabled interaction. Is the email server up and running? Has a crucial piece of data changed? Are sensors detecting movement? Is the technical support "bot" able to answer questions? Is the syndication service delivering content feeds? Has business intelligence software triggered an important event? Has the shipment reached its destination? Is my friend available on her mobile phone? Is the restaurant around the corner still open? These are all examples of presence in action.

The concept of presence is also being extended to capture more subtle information about availability, such as a person's current activity, mood, location (e.g., GPS coordinates), or preferred communication method (e.g., phone, email, IM, etc.). While these extensions are interesting and important, they mainly serve to supplement information about an entity's network connectivity, which remains the core of presence.

## The Value of Presence

To understand the value of presence, consider a few scenarios.

### INCIDENT RESPONSE

#### Presence | DEFICIENT

It's rush hour in Washington, D.C., and the Woodrow Wilson Bridge is bumper-to-bumper as usual. Suddenly, a lone figure appears teetering on the rail, faltering for balance, and threatening to jump. As drivers react in horror and disbelief, hordes of police cars, ambulances, and transportation officials from the three jurisdictions surrounding the bridge converge on the scene. Traffic comes to a complete standstill. The situation is compounded by lack of any means for these first responders—from Maryland, Virginia, and the District of Columbia—to effectively and quickly communicate across their jurisdictions. As a result, drivers and responders are stuck on the bridge for more than seven hours.

#### Presence | ENABLED

When word arrives that someone is threatening to jump from the Woodrow Wilson Bridge, officials at the Capital Wireless Integrated Network (CapWIN) create a dedicated text conferencing "incident room" and invite representatives from the relevant jurisdictions whose presence indicates that they are available for interaction. These users gather and share information as it comes in and plan a coordinated response to the incident. Front-line responders also enter the room based on their proximity to the incident and then review the complete discussion history to quickly get up to speed. Afterwards, officials analyze the room log to prepare an "after action" report and improve cross-agency response to future incidents. As a result of the communication enabled by presence-driven services, the incident is resolved in less than an hour.

### COMMODITY TRADING

#### Presence | DEFICIENT

As an unexpected heatwave grips western North America, officials at an electric utility in Colorado are searching for additional power to keep industry humming. Information is hard to come by, but they call their contacts at several utilities in Texas and the upper Midwest, frantically completing a few trades for peak electricity (though not enough to prevent early closings at government offices throughout the state). Unfortunately, they later learn that cheaper baseload electricity had been available at the time.

#### Presence | ENABLED

When the heatwave arrives, utility officials log into the online trading network run by Calgary-based Net Energy. They receive real-time pricing updates, which enable them to quickly source enough baseload electricity to meet the needs of their customers. And because presence enables them to see other energy traders who are using the system, they don't have to make last-minute phone calls—instead, they seamlessly negotiate with multiple traders at once, enabling them to find the lowest price and complete more transactions in less time.

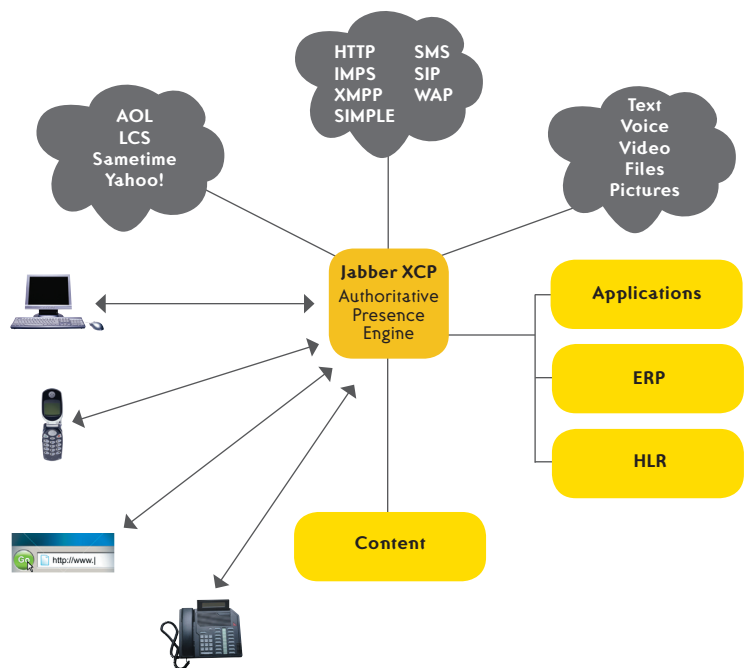
As these brief examples show, presence can be extremely valuable. But what are the technical requirements for presence-enabling applications and organizations? Are some presence technologies better than others?

### THE TECHNOLOGY OF PRESENCE

Because presence is primarily an enabling technology for interaction, it first emerged as one aspect of communication systems, especially IM systems such as ICQ ("I seek you"), which enabled users to see the availability of their friends. The historical connection between IM and presence is evident even today in the main presence protocols: Instant Messaging and Presence Service (IMPS), originally developed by the Wireless Village initiative and now maintained by the Open Mobile Alliance (OMA); Session Initiation Protocol (SIP) for Instant Messaging and Presence Leveraging Extensions (SIMPLE), an ongoing effort within the Internet Engineering Task Force (IETF); and the Extensible Messaging and Presence Protocol (XMPP), first developed in the Jabber open-source community and subsequently ratified as an Internet standard by the IETF.

Essentially, presence follows a software design pattern known as publish-and-subscribe (pub/sub): a user or application publishes information about its network availability, and that information is broadcast to all entities that are authorized to receive it. The authorization usually takes the form of a subscription, and in IM implementations the authorized entities are often called "buddies" or "contacts."

### THE PUB/SUB PATTERN



Presence technologies first came to wide attention as a crucial aspect of consumer-oriented IM services. These services familiarized millions of people with the concept of availability for communications and provided a good start toward ubiquitous presence. However, the limits of consumer IM services quickly become clear when enterprises and service providers attempt to integrate presence into business-critical systems and services. Such organizations require a great deal of control and flexibility over the technologies they deploy; in particular, they need presence technologies that meet the following criteria:

1. A logical separation between the presence service and the communication mechanisms (e.g., IM or VoIP) that presence enables.
2. A scalable, distributed architecture wherein any given organization or provider controls its own presence domain.
3. A presence format that is lightweight enough to not overload the network once every entity, application, and device is presence-enabled.
4. An extensible framework capable of intelligently handling future deployment of new presence-powered applications, systems, and services.
5. Strong security management, including system authentication and granular subscription authorization.
6. The ability for any device or application to publish and subscribe to presence information.
7. The ability to integrate presence information from multiple sources (e.g., each of a person's devices).
8. The ability to access presence data via many different methods, such as direct network connections and Web services application program interfaces (APIs).
9. The ability to federate numerous cross-protocol presence sources.
10. The ability to integrate presence information with existing organizational infrastructure such as business directories.
11. The ability for other sources of information (e.g., location, personal blogs/photo albums, auction sites, etc.) to publish content and allow other people and/or applications to subscribe to that information so that updates and changes are notified in real time based on the presence/availability of those people/applications.

Closed, consumer-oriented IM services do not enable enterprises or institutions to meet these requirements and leverage the power of presence. A smarter approach is to use one of the standard presence protocols: SIMPLE (an ongoing effort within the IETF supported by several large commercial vendors) and XMPP (formalized within the IETF and widely deployed). As discussed in the following sections, these two technologies are the main options to consider when developing and deploying presence applications that address both the fixed and mobile Internet ecosystems.

#### SIMPLE at a Glance

- IETF technology
- Based on SIP
- Still under development
- Not widely deployed

#### SIMPLE

Development of SIMPLE technologies began in 2001 based on work on the Session Initiation Protocol (SIP) within the IETF, and the first SIMPLE presence specification was published in August 2004. SIP is a sophisticated but fairly complex rendezvous protocol for negotiating and managing communication sessions, mainly used to manage voice and video calls. In general, SIP interactions result in a hand-off to a dedicated streaming protocol such as Realtime Transport Protocol (RTP). However, SIP has also been extended to carry presence and messaging traffic over the rendezvous channel. These extensions are known collectively as SIP for Instant Messaging and Presence Leveraging Extensions or SIMPLE.

SIMPLE presence is implemented through customary SIP semantics such as the SUBSCRIBE and NOTIFY methods. Because SIP is mainly used to subscribe to relatively short-lived events, such as a conference call, SIP presence subscriptions are not persistent and must be renewed periodically (e.g., once an hour). The presence information itself is provided when a user sends a special Extensible Markup Language (XML) file to the server

that maintains its registration state on the network. This XML file may contain information about any number of devices or connections maintained by the user, and it is up to the presence server or subscriber to parse that complete information into pieces. The presence server may also maintain a list of authorized subscribers and send notifications to those subscribers whenever the publishing entity updates its XML file.

As of this writing, various SIP-related working groups within the IETF continue to define associated technologies for extended presence, geographical location, contact lists, and the like. Once the SIP extensions for presence are finalized, it is likely that more presence-enabled SIP applications will be implemented and deployed. To date, SIMPLE presence has been used mainly by devices that already contain a full SIP protocol stack (e.g., SIP-based softphones) as well as early versions of several IM applications, including Microsoft Live Communications Server (LCS) and IBM Lotus Sametime. However, SIMPLE systems have not yet been widely deployed and their behavior in production environments is not yet known, especially given the large size of SIMPLE presence packets and the short-lived nature of SIMPLE presence subscriptions.

#### XMPP at a Glance

- IETF core, open extensions
- Based on XML streams
- Open-source heritage
- Widely deployed

#### XMPP

The Extensible Messaging and Presence Protocol (XMPP) is the IETF's formalization of the core XML messaging and presence protocols originally developed by the open-source Jabber community in 1999. These protocols have been in wide use on the Internet for over five years. In a manner similar to Hypertext Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP), the IETF's work on XMPP mainly focused on improving an existing protocol rather than designing a new protocol from scratch. The non-profit Jabber Software Foundation (JSF) contributed XMPP to the IETF in 2002 and continues to define XMPP extensions through its well-regarded Jabber Enhancement Protocol (JEP) series ([www.jabber.org/jeps/](http://www.jabber.org/jeps/)). A wide variety of XMPP-based technologies are currently deployed at tens

of thousands of organizations, including leading financial services firms, technology companies, mobile and fixed telephony operators, and critical national security infrastructures.

The XMPP transport layer is an XML streaming technology that includes strong security, channel encryption, authentication, and internationalization using best-of-breed standards such as Simple Authentication and Security Layer (SASL), Transport Layer Security (TLS), Unicode, and Stringprep. Various application-layer technologies are built on top of XML streams, and one key XMPP application is presence. The client-server model enables multiple clients to connect to a presence server on behalf of any given presence account, for example to handle multiple user devices. Information from these connections is typically "broadcast" to all authorized subscribers and may be aggregated by the server so that subscribers do not need to parse through presence information from each publishing resource. In addition, presence extensions for geographical location and other advanced features have been added at higher layers in the XMPP protocol stack. As described below, these extended presence features are normally implemented on top of the XMPP pub/sub framework.

#### MAKING A CHOICE: SIMPLE OR XMPP?

SIMPLE is a relatively early-stage presence technology that is still being defined, whereas XMPP is a stable technology that has been widely deployed on the Internet since 1999. In general, early adopters of presence-enabled technologies (such as Wall Street banks and national security organizations) have chosen to deploy XMPP-based solutions in order to generate immediate competitive advantage and organizational efficiency. The two technologies are likely to co-exist for many years to come, and there already are gateways that enable the new SIMPLE systems to federate with deployed XMPP systems.

#### EXTENDING PRESENCE

Basic presence indicates an entity's network availability, sometimes with supplemental information such as a status message (e.g., "In a meeting") or availability details (e.g., "Away"). The concept of presence has been

**Where Are You?**

**One popular extension to presence is geographical location.**

For example, knowing if a friend is nearby or being informed of special events close to your current location. This information builds virtual communities and can drive adoption of wireless services.

extended further to incorporate aspects of an entity's real-time identity such as geographical location, mood, activity, even the music to which a person is listening.

The XML formats used by both SIMPLE and XMPP make it relatively easy to include such "extended presence" information. The challenge is to manage extended presence. There are several factors to consider. First, loading presence with frequent changes in things like geolocation and musical tunes can result in considerable traffic volumes that pose serious challenges to the bandwidth consumption and scalability of the presence infrastructure (especially in SIMPLE systems, which have much larger presence packets than XMPP). Second, the permissions model for extended presence will be quite different from the permissions model for basic presence. For example, not everyone who needs to know an individual's network availability may be interested in the person's mood or authorized to know the person's geographical location.

In general there are two ways to address these issues. The SIMPLE approach is generally to treat both basic and extended presence as conceptually the same, but to institute filtering and authorization rules so that presence engines can control which entities are allowed to publish and receive particular kinds of extended presence data. This approach keeps basic presence and extended presence together but tends to introduce quite a bit of complexity into presence engine design and implementation.

The XMPP approach is generally to treat basic presence and extended presence as conceptually similar but distinct by handling extended presence through dedicated pub/sub engines rather than core presence engines (on the theory that although presence is a kind of pub/sub, not everything that exhibits pub/sub-like qualities is presence). This approach separates extended presence from basic presence and enables greater flexibility and granularity in managing extended presence services.

Right now it is too early to say which is the right approach to handling extended presence. However, production experience with other forms of rapidly changing information such as Really Simple Syndication (RSS) feeds indicates that the pub/sub approach used in XMPP has inherent advantages with regard to scalability, filtering, and control.

**FROM PRESENCE TO INTERACTION**

Presence is not the only catalyst for interaction. Another key catalyst is knowledge of capabilities, which includes answers to questions such as: Does my contact have a video camera or microphone attached to her computer? Is she able to easily input text on her phone? Does her workgroup application support multi-user chat?

**Are You Capable?**

**Presence alone is not enough.**

An authoritative presence solution also includes capabilities data to maximize communication opportunities.

The old-fashioned way to determine another entity's capabilities is for one human to ask another human, but that does not always work well because people tend to be slow and error-prone. The modern, reliable method is called service discovery, wherein applications and devices directly exchange information about their capabilities without human involvement.

While no generalized framework for service discovery has been produced by a standards development organization such as the IETF, there exists a capabilities extension for SIP/SIMPLE and there is also a robust, stable service discovery extension for XMPP.

However, static capabilities information is not always enough, especially if applications need to repeatedly "poll" for that information. Because capabilities can change in real time (e.g., when a user plugs a camera into her computer), it can be helpful to embed capabilities information within broadcasted presence information, often in a shorthand form to preserve network resources. The SIMPLE Working Group is

working on such a technology and one already exists in a widely-deployed XMPP extension. Together, service discovery and capabilities broadcast enable users and applications to gain knowledge about the capabilities of other entities on the network, providing a real-time "trigger" for additional usage of presence-enabled systems.

## PUTTING PRESENCE TO WORK

Talk of protocols and underlying technologies is all well and good, but the real challenge is figuring out how to leverage the power of presence within an organization or service offering. There are really three parts to the equation: the ability to publish presence information from a wide range of data sources, the ability to receive or embed presence information in just about any platform or application, and a robust presence engine to tie ubiquitous publishers and subscribers together.

### On Ubiquity

**How widespread can presence become?** Consider the example of a logistics company that scans packages at each stage in the shipping process, triggering presence notifications to the sender, recipient, and customer service staff. That's ubiquity!

Just about any entity that can communicate over a network might have presence. In general, the requirements for functioning as a presence publisher are fairly minimal. As a result, SIP software stacks are available for C, C++, Java, and Python, and XMPP code libraries are available for C, C++, C#, COM, Delphi, Erlang, Flash, Java, JavaScript, Mono, Objective-C, Perl, PHP, Ruby, and Tcl. Given the wide range of languages supported, it is relatively easy to add native presence publishing capabilities to most applications and devices.

Of course, the question arises: How many devices and applications really need to publish presence and capabilities information? Just as Internet users today increasingly face the prospects of "information overload" from messaging media such as email and IM, soon people may also complain about "presence overload" as they struggle to manage the presence information published by their many devices and applications. Software developers and service providers are working on better ways to aggregate and manage presence information in order to make it a useful enabler rather than yet another burden of the digital age.

Yet enabling a wide range of devices and applications to publish presence information is only half of the battle. The other half is delivering the right presence information to the right subscribers (whether they are people, devices, or applications) at the right time.

Already, presence information is being added to a wide range of devices and applications to streamline and accelerate communication and information exchange. Phones, personal digital assistants (PDAs), address books, documents, directories, email messages, Web pages, search engines, workflow applications, collaboration environments, customer service centers, and much more are all being presence-enabled.

What does it mean to be "presence-enabled"? The basic idea is to show the availability of a person, document author, message sender, machine, or other "presentity" in an appropriate interface (often using a special icon). More sophisticated applications aggregate presence information about all of a person's connections (e.g., phone, PDA, and laptop) or even construct aggregate presence for an entire group (e.g., technical support).

For communication devices such as phones and applications such as IM, presence information is often built in. For less communication-centric applications such as a document or Web page, presence may be gathered by means of a Web services API or channeled through a presence daemon on a local machine (e.g., to distribute availability information to any application on a personal computer). Providing presence data through as many avenues as possible is in large measure the responsibility of a presence engine, as described below.

### CHOOSING AN AUTHORITATIVE PRESENCE SOLUTION

A presence engine acts as a central clearinghouse for presence publishers and subscribers. As with real-life information brokers, whose purpose is to connect buyers and sellers, a presence clearinghouse provides aggregation of information from many sources, abstraction of that information into open and flexible formats, and distribution of that information to a wide variety of interested parties. In the realm of presence, these key qualities of aggregation, abstraction, and distribution imply that the ideal presence clearinghouse has the following features:

Checklist—The Big Three
<p><b>Make sure your presence solution can:</b></p> <ol style="list-style-type: none"> <li>1. Aggregate from any presence producer</li> <li>2. Abstract from multiple presence formats</li> <li>3. Distribute to any presence consumer</li> </ol>

- **Trustworthy.** As presence becomes the new dial tone for Internet communications, presence engines need to provide strong authentication, channel encryption, explicit authorization and access control policies, high reliability, and the consistent application of aggregation rules.
- **Open.** Speaking multiple protocols including IMPS, SIMPLE, and XMPP is a basic requirement in order to distribute presence information as widely as possible, and a presence engine needs to support any multiple connection methods (e.g., Transmission Control Protocol (TCP), Wireless Application Protocol (WAP), HTTP), inter-domain federation, Web services APIs, and directory integration.
- **Extensible.** Flexibility is needed to provide basic availability data and extended presence information, as well as to adjust to the “market realities” of new features such as service discovery and capability advertisement.
- **Intelligent.** Aggregating information from a wide variety of sources requires pluggable presence rules that enable subscribers to get the right information at the right time, even if presence needs to be combined across multiple devices or people.

### JABBER XCP: THE PRESENCE ENGINE OF CHOICE

The Jabber Extensible Communications Platform (Jabber XCP) is a highly-scalable, widely-deployed presence and messaging framework that meets all of the key requirements for an authoritative presence solution. Jabber XCP’s open architecture supports a full range of presence protocols (IMPS, SIMPLE, XMPP) and access methods while ensuring maximum reliability and security. Driven by a core pub/sub engine, Jabber XCP has the flexibility to handle any kind of real-time notifications, from basic availability to high-volume applications such as geolocation services. Data aggregation and pluggable rules are standard, enabling an organization to provide customized presence services to its particular environment.

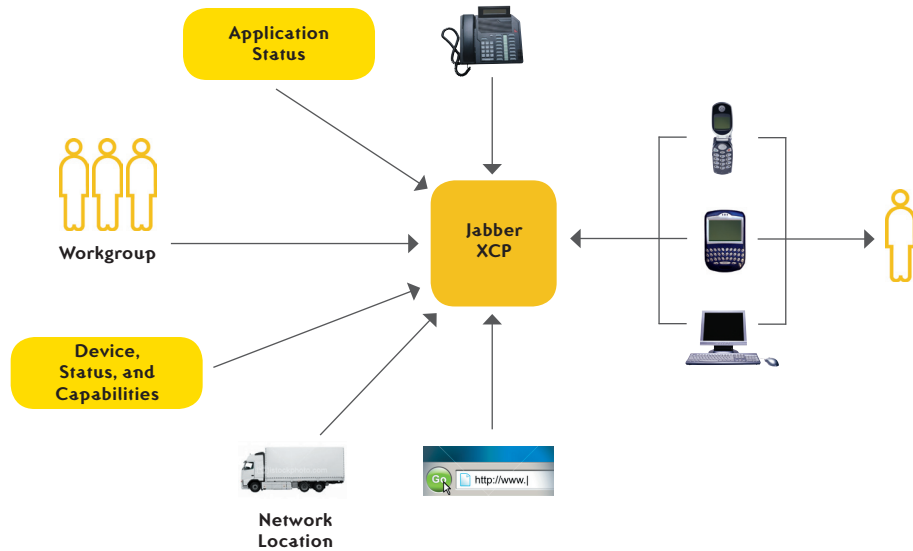
Because of its ability to address a diverse set of network, device, and application requirements, Jabber XCP has been the choice of innovative organizations in an incredibly wide range of industries:

Wall Street	Government	Service Provider
Jabber XCP powers mission-critical availability and communication services to the world’s leading financial services firms.	Jabber XCP provides a model presence and communications infrastructure for multiple emergency management agencies.	Jabber XCP is the basis for presence-driven services such as mobile push-to-talk and push-to-video.
<b>THE RESULT</b> Improved information sharing, quicker decisions, and more transactions completed in less time.	<b>THE RESULT</b> Cross-agency interoperability and faster response times—just when they’re needed most.	<b>THE RESULT</b> Improved customer retention and increased usage of high-value services.

As shown in the following diagrams, Jabber XCP provides an unmatched combination of presence aggregation, abstraction, and distribution services to a broad array of networks, devices, and applications.

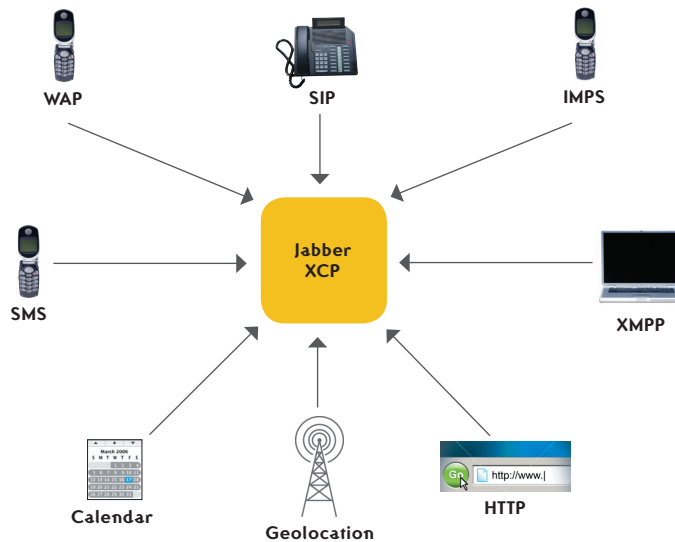
**AGGREGATION: PRESENCE PRODUCERS**

Jabber XCP aggregates information from the full spectrum of presence producers—stand alone computers, existing business applications, and a wide range of mobile phones and handheld devices.



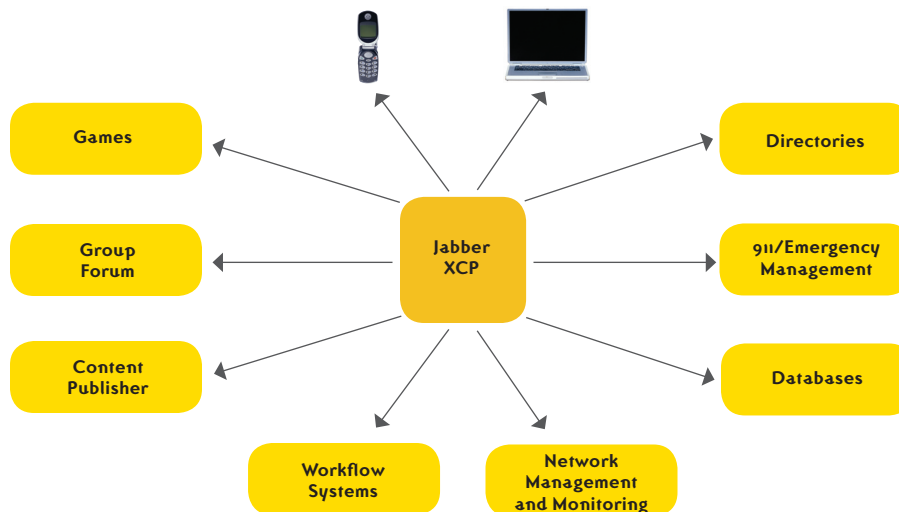
**ABSTRACTION: PRESENCE FORMATS**

Jabber XCP can handle presence information in all existing formats. It then abstracts from any particular format by providing common interfaces and APIs for maximum flexibility.



**DISTRIBUTION: PRESENCE CONSUMERS**

Jabber XCP covers the “last mile” of presence delivery by pushing information out to any presence consumer—people, devices, and business applications—to maximize the value of your presence investment.



Jabber XCP is at the center of the world's most innovative presence infrastructures. Contact Jabber Inc. today to find out how the power of presence can be leveraged to drive competitive advantage and transform your organization into a real-time enterprise.

**References**

The following specifications provided background information for this white paper and may be of interest to technical readers.

IETF RFC 2778: A Model for Presence and Instant Messaging. URL: <<http://www.ietf.org/rfc/rfc2778.txt>>  
 IETF RFC 2779: Instant Messaging / Presence Protocol Requirements. URL: <<http://www.ietf.org/rfc/rfc2779.txt>>  
 IETF RFC 3261: SIP: Session Initiation Protocol. URL: <<http://www.ietf.org/rfc/rfc3261.txt>>  
 IETF RFC 3840: Indicating User Agent Capabilities in the Session Initiation Protocol (SIP). URL: <<http://www.ietf.org/rfc/rfc3840.txt>>  
 IETF RFC 3856: A Presence Event Package for the Session Initiation Protocol (SIP). URL: <<http://www.ietf.org/rfc/rfc3856.txt>>  
 IETF RFC 3863: Presence Information Data Format (PIDF). URL: <<http://www.ietf.org/rfc/rfc3863.txt>>  
 IETF RFC 3920: Extensible Messaging and Presence Protocol (XMPP): Core. URL: <<http://www.ietf.org/rfc/rfc3920.txt>>  
 IETF RFC 3921: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence. URL: <<http://www.ietf.org/rfc/rfc3921.txt>>  
 JSF JEP-0030: Service Discovery. URL: <<http://www.jabber.org/jeps/jep-0030.html>>  
 JSF JEP-0060: Publish-Subscribe. URL: <<http://www.jabber.org/jeps/jep-0060.html>>  
 JSF JEP-0080: User Geolocation. URL: <<http://www.jabber.org/jeps/jep-0080.html>>  
 JSF JEP-0107: User Mood. URL: <<http://www.jabber.org/jeps/jep-0107.html>>  
 JSF JEP-0108: User Activity. URL: <<http://www.jabber.org/jeps/jep-0108.html>>  
 JSF JEP-0112: User Physical Location. URL: <<http://www.jabber.org/jeps/jep-0112.html>>  
 JSF JEP-0115: Entity Capabilities. URL: <<http://www.jabber.org/jeps/jep-0115.html>>  
 JSF JEP-0119: Extended Presence Protocol Suite. URL: <<http://www.jabber.org/jeps/jep-0119.html>>