

System Architecture Diagrams

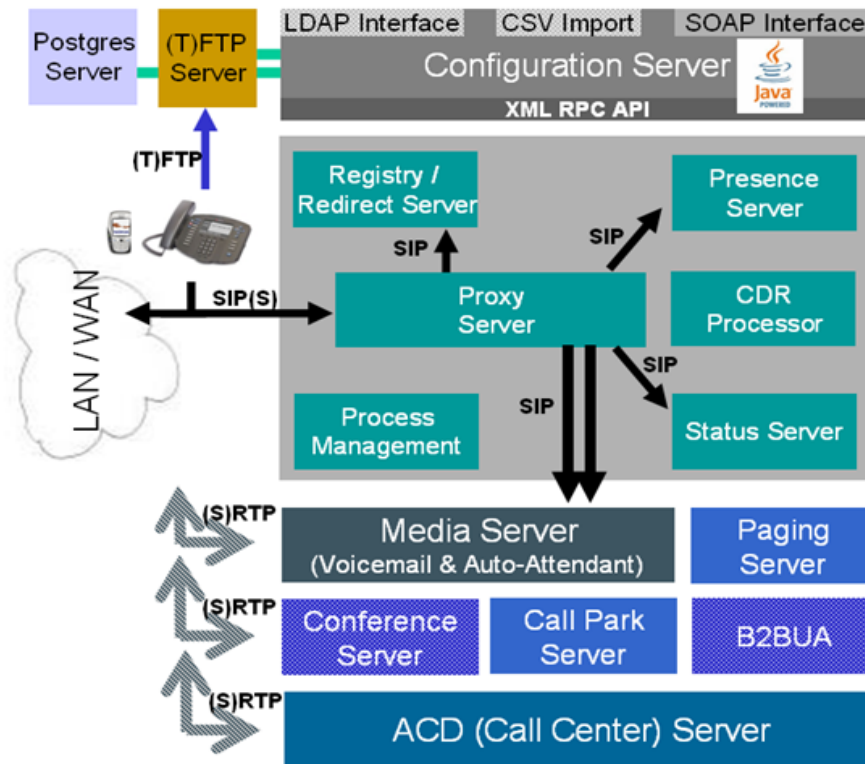
The sipXcom System Architecture Diagram

The sipXcom system is built based on a modular architecture with different components that communicate over TCP/IP for the most part.

The SIP protocol was designed in support of a fully distributed architecture where different features are provided by servers that again communicate among themselves using the SIP protocol. There is a strict separation between Media and Signaling, which allows a globally scalable system to be built with the necessary redundancy and resiliency. Media streams, once setup, flow directly between end-points (phones and gateways) without any requirement for media to go through a central signaling server.

Highlights

- **Modular:** 12 different server components that can coexist on one server HW or be distributed to different systems
- **Resilient:** Offers HA redundancy between call control components
- **Scalability:** Load balancing using the DNS SRV system
- **SIP Proxy based:** sipXcom is not based on a B2BUA but implements a true SIP proxy architecture
- B2BUA server (sipXbridge) for connecting to Internet Telephony Service Providers (ITSPs) (not recommended for HA or production)
- Conference bridge included in package
- Freeswitch based Auto Attendant and Voicemail integrated
- **Better Voice Quality:** Media goes direct point-to-point between the end points and never through the call control system
- **Standard SIP:** sipXcom is all about a standards compliant implementation
- **Plug & Play Management:** Phones are managed plug & play from within the system
- **IT Integration:** LDAP, SOAP, and database integration. System looks like and operates like an IT application
- **No Special HW:** sipXcom does not require any special HW. A standard Intel server will do

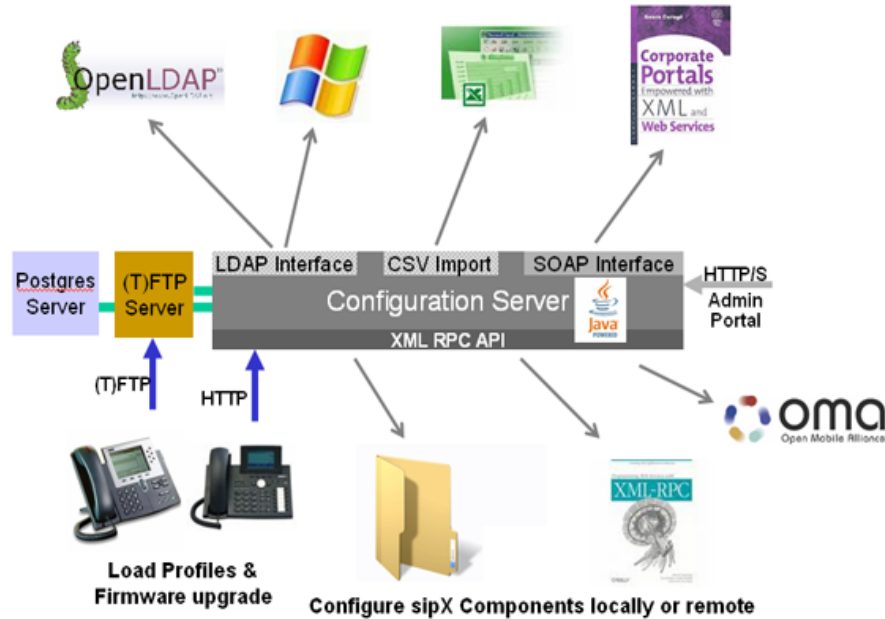


Given its architecture, the sipXcom system is designed to interoperate with other third party feature servers. In addition, as we accomplished our objective of a truly distributed system, several instances of every component can be run on dedicated hardware and in different geographical locations to render a very scalable system. The sipXcom Configuration Server provides plug & play management for core components, all feature servers, as well as connected peripherals such as phones and gateways. An XML-based plug-in framework allows easy inclusion of additional components, both feature servers as well as additional peripherals. Documentation exists about how to add support for additional [phones](#) and [gateways](#).

While the different components of the sipXcom system could be used stand-alone, the focus of the project has been to provide a complete sipXcom IP PBX system out-of-the-box. On Fedora, Red Hat, Debian and SUSE distributions, the sipXcom system can be installed easily using the distributions's respective package management system from our repository.

Architecture of the Configuration Server

The sipXcom Configuration Server is a Web Services based configuration management system that offers plug & play management for all the sipXcom server components including all connected phones and gateways. It connects to the existing IT infrastructure e.g. using LDAP and allows Web Services portals to use its services over a SOAP interface. It also allows integration with Microsoft Active Directory and Exchange 2007.



User data can be uploaded from LDAP or using a comma-delimited CSV file typically generated by a spreadsheet application.

Library components

- **Jetty:** Java HTTP Server and Servlet Container
- **Spring Framework:** Java Application Server Framework
- **Tapestry:** Apache Jakarta Tapestry
- **Hibernate:** Idiomatic Persistence for Java
- **Velocity:** Java based Template Engine
- **Lucene:** Java-based indexing and search

For persistent storage we use the [PostgreSQL](#) database.

A Web browser directly connects to the Jetty HTTP server and the Apache server is no longer required.

Web Services integration is possible using the provided SOAP interface, which can be used for bulk configuration transactions.



Starting with version 4.2 sipXconfig provides a set of REST APIs using [RESTlet](#) library. The plan for next releases is to expose all sipXconfig functionalities in REST calls.

Configuration Server communicates with other sipXcom server components using the file system or its [XML RPC](#) interface. Using the local file system, configuration files are written to the `/etc/sipxpbx` directory.

Two different mechanisms are supported for phones and gateways to get access to their respective configuration files at boot time:

- Both an FTP and TFTP server provide access to the `ftproot` directory on the sipXcom server;
- HTTP Web access is possible to the `docroot` directory. `docroot` access is provided on port 8090 using the Apache Web server.

Phones and gateways can be auto-configured on the LAN, which further simplifies deployment.

