

LogicomUSA[®]'s SoftSwitch Overview and Architecture

LogicomUSA[®]'s flagship product, is a softswitch built to provide advanced PBX, voice, data and messaging services to millions of users globally. Our PBX software platform provides a complete PBX solution – without the inflated cost and management headaches of old phone systems. Calls are sent as data packets over the computer data network (VoIP) instead of the traditional phone network.

The Softswitch platform was created for LogicomUSA[®] partners/resellers to be able to create their own telecom service strategy and be able to provide whatever unique service packages they want to their customers. Resellers are empowered to assign whatever permissions they desire and to create their own cost structures paying LogicomUSA[®] only a set fee for services.

Furthermore, resellers are able to customize the Softswitch as they wish. Whether this means enabling existing LogicomUSA[®] features or building and integrating their own via the RESTful APIs, LogicomUSA[®]'s mission is to revolutionize communications by allowing endless possibilities.

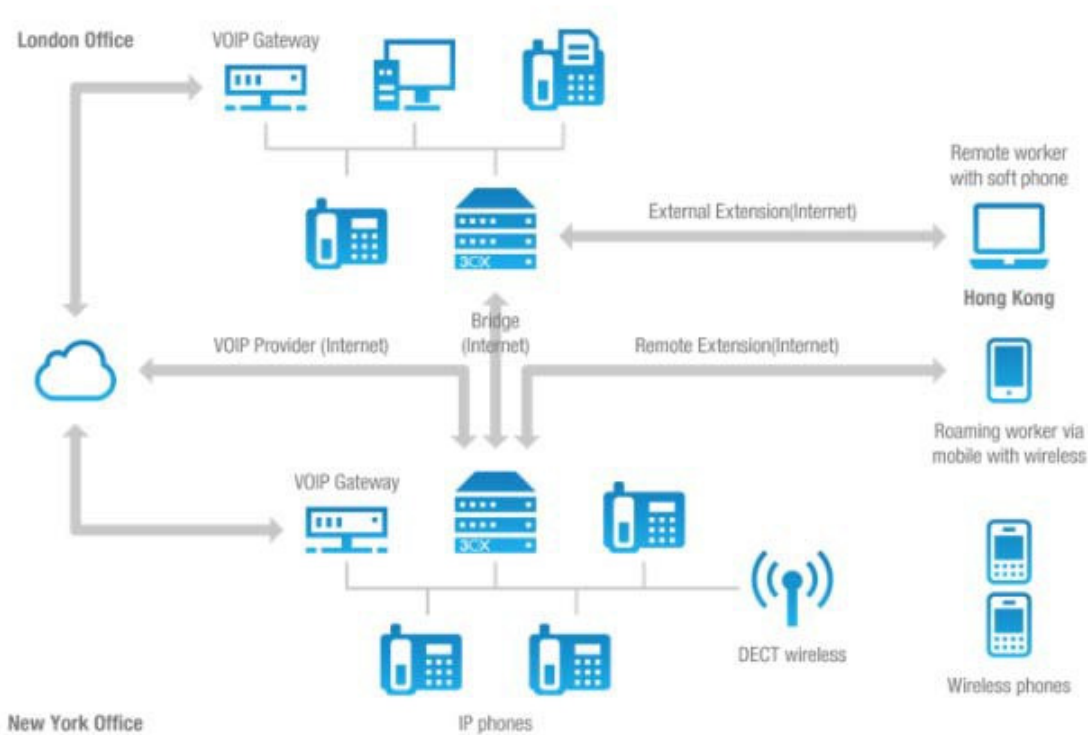
Private Branch Exchange (PBX) Systems

A private branch exchange (PBX) is a telephone exchange or switching system that serves a private organization, performs concentration of central office lines or trunks and provides intercommunication between a large number of telephone stations in the organization. The central office lines provide connections to the public switched telephone network (PSTN) and the concentration aspect of a PBX permits the shared use of these lines between all stations in the organization.

Each PBX-connected station, such as a telephone set, a fax machine or a computer modem, is often referred to as an extension and has a designated extension telephone number that may or may not be mapped automatically to the numbering plan of the central office and the telephone number block allocated to the PBX.

How an IP Phone System Works

A VoIP Phone System generally consists of the IP PBX server, one or more SIP based phones and a VoIP/PSTN Gateway or a VoIP service provider. The IP PBX server is similar to a proxy server. SIP clients, being either softphones or hardware based phones, register with the IP PBX server. When they wish to make a call they ask the IP PBX to establish the connection. The IP PBX has a directory of all phones/users and their corresponding SIP address, and connects an internal call or routes an external call via either a VoIP/PSTN gateway or a VoIP service provider. The image below illustrates how an IP PBX integrates with the network and how it uses the PSTN and/or the Internet to connect calls.



Hosted PBX Systems

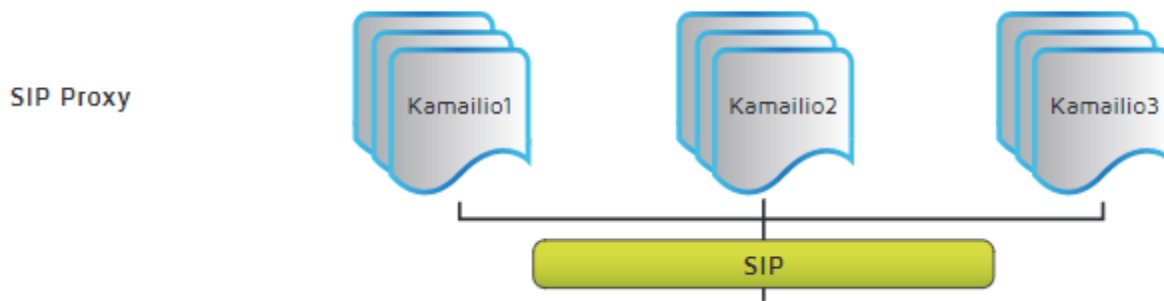
Hosted PBX systems deliver PBX functionality as a service, available over the PSTN or the Internet. Hosted PBXs are typically provided by a telephone company or service provider, using equipment located in the premises of a telephone exchange or the provider's data center. This means the customer does not need to buy or install PBX equipment. Generally the service is provided by a lease agreement and the provider can use the same switching equipment to service multiple hosted PBX customers. The LogicomUSA[®]'s Softswitch Hosted Cloud product is a hosted PBX system.

Some of the benefits of a hosted PBX system include:

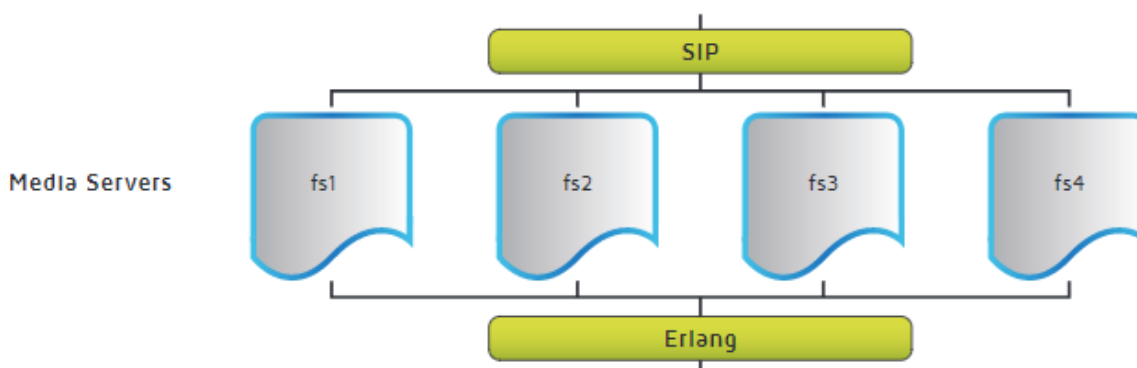
1. A single number can be presented for the entire company, despite it being geographically distributed. A company could even choose to have no physical premises with workers connected from home using their domestic telephones but receiving the same features as any PBX user.
2. Multimodal access where employees access the network via a variety of telecommunications systems, including POTS, ISDN, cellular phones and VOIP. This allows one extension to ring in multiple locations (either concurrently or sequentially).
3. Allows scalability so that a larger system is not needed if new employees are hired thus helping to make sure that resources are not wasted if the number of employees is reduced.
4. Eliminates the need for companies to manage or pay for on-site hardware maintenance.

LogicomUSA[®]'s Softswitch Architecture

The first layer of technology deployed in the Softswitch Stack is Kamailio which serves as our Session Border Controller. In the LogicomUSA[®] architecture, Kamailio provides load balancing functionality as well as all of the normal services a SIP Proxy provides. Kamailio acts as a load balancer for the SIP Requests (much the way nginx or HAProxy are used to load balance Web Requests). We minimize the number of public interfaces needed to inform clients and carriers by pointing them to the load balancers. Adding capacity becomes as easy as informing Kamailio of the new Switch. We utilize SIP as a method of passing calls from Kamailio towards our media server.

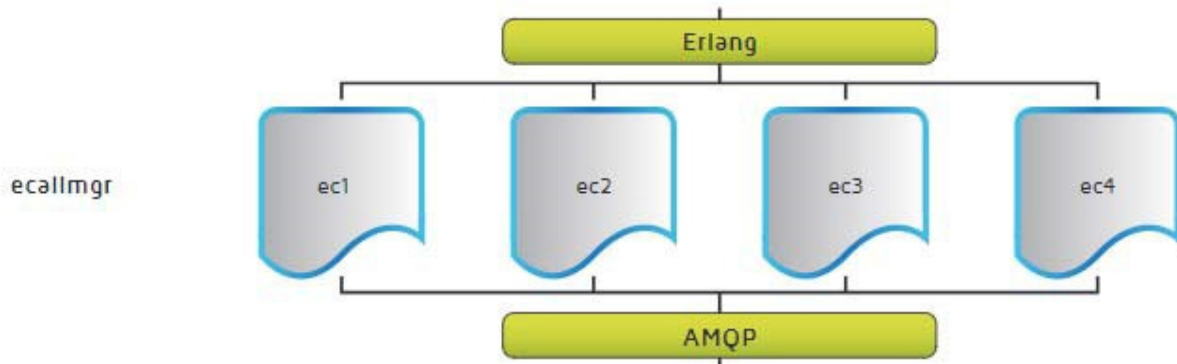


Once the call has been authenticated to our proxy in the prior step, we pass the call to FreeSWITCH, which serves as our media server. In our environment, we leverage a streamlined version of the FreeSWITCH system. FreeSWITCH, in our setup, serves only to provide trunking connection services; that is to say, FreeSWITCH only does ‘Call Pickup’, ‘Call Hangup’ and ‘Transcoding’. No logic beyond those basic functions is performed within those boxes. We perform this logic in another layer via tight integration with the `mod_erlang_event` module. FreeSWITCH is an amazingly resilient and powerful switch. Its design has allowed us to abstract the logic away from FreeSWITCH allowing these servers to do what they do best (transcode calls) while freeing our architecture to scale each resource independently. In other architectures, when one is scaling, the core processor contains all of the logic as well as the database. The result is that the user scales all components instead of only the necessary ones which usually is a huge waste of time and resources.

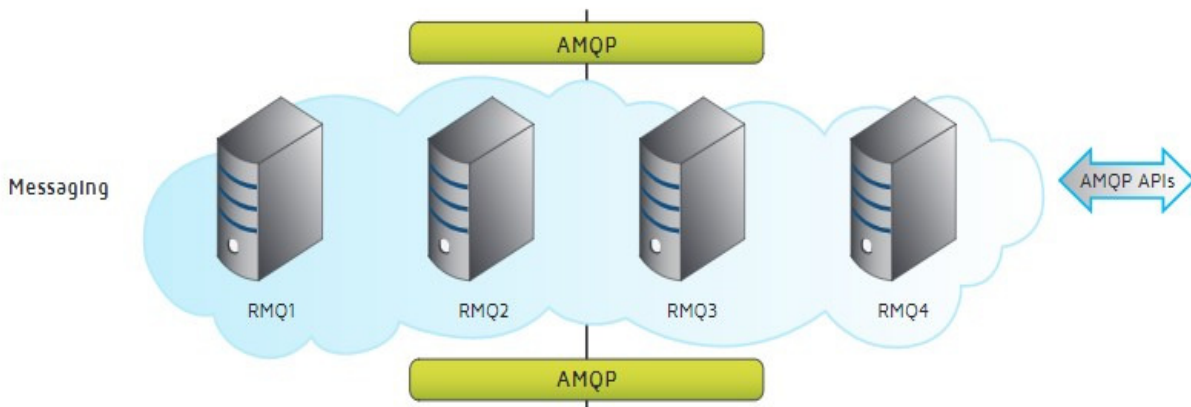


LogicomUSA[®] leverages the functional Language Erlang in order to provide massively scalable logic as a function of our system. Erlang has been deployed in some of the most stable, and massive, architectures on Earth, including the AXD 301 (the only system to reach 99.9999999, 9 9’s, of reliability, which is 4 mins of downtime every 20 years), Goldman Sachs High-speed trading models, and Github. It excels where other architectures struggle, particularly when dealing with massive concurrent media streaming. Erlang was developed by Sony Ericsson in

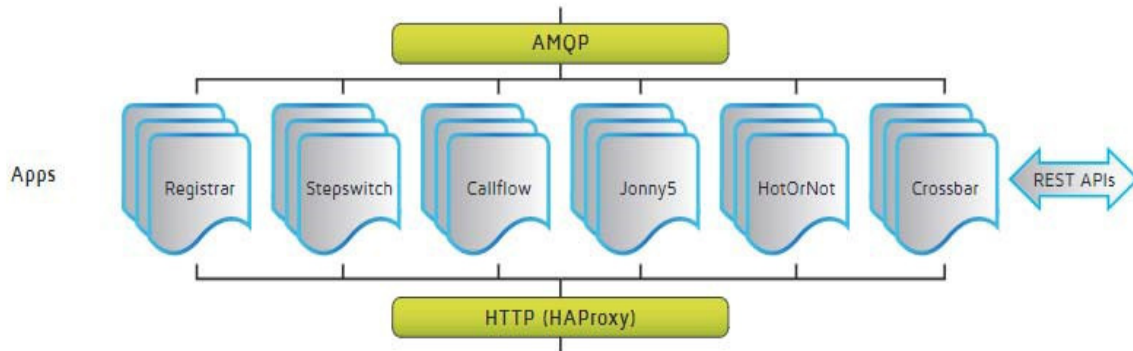
the Mid 80's for Cell phone Tower Control, so it's easy to understand why this language performs the way it does. ECallManager is the name of our Erlang Logic program and it's where all of the applications in the Softswitch architecture engage with Users. It serves as the primary part of what users imagine when they think of the Softswitch. Essentially, every voicemail, every IVR, and every menu is referenced in ECallManager when users interact with the Softswitch.



We leverage AMQP which is a messaging queuing system. Specifically, we leverage Rabbit, primarily because Rabbit has all of the brokers built-in, and works amazingly well over the WAN. Message passing with proper queuing is extremely important to our design. In the Softswitch, we care most about media delivery completion, and so we arrange for events related to those topics to be escalated in the queue. CDR delivery, for example, is relegated to delayed status as CDRs (as long as they're eventually correct) are not temporally sensitive. If you have to choose between having a call arrive on time, and having a record of the call arrive on time, you're definitely choosing the call, and our design takes that into consideration. RabbitMQ is the entry point for AMQP based API commands. If you are writing applications using our AMQP APIs, this is where the commands enter the Softswitch.



The Apps Layer is where all of the applications that have been developed for the Softswitch and are stored, and where they reside in our architecture. Applications can invoke Softswitch commands at this layer using our REST APIs. Apps can control what happens at all stages of a call (even initiating calls on their own). Authentication, routing, in-call applications (like IVRs and voicemail), and more, are all exposed via the Softswitch APIs. We provide a set of APIs via a REST interface, implemented as an app called CrossBar. With CrossBar, configuration of PBX functionality, conferencing or other services is exposed.



We take data management very seriously. We have seen database technology advance considerably over the last decade and we implement many of those innovations by combining NoSQL database architecture with sensible schema and proven resilient design. We take advantage of Database Sharding, Global Distribution and a number of other killer features using BigCouch, which takes the best elements from Google's BigTable and CouchDB to form a powerful but stable distributed architecture. Our implementation of CouchDB is a key-value store using JSON schema. It is truly multi-tenant and scalable.

