

# Voxer Architecture and Feature List

Voxer® is a Walkie Talkie app for smartphones. The app allows users to send instant audio, text, photo and location messages to one subscribers or a group of subscribers. The recipients can listen to the message while live or check it out later.

## Architecture

The Voxer Platform is a high performance, highly distributed, and redundant system that runs across an array of servers. The system is a configuration of tiers of server rings (clusters) that provide the various functions required to support Voxer communications. Servers can be dynamically allocated to the rings allowing linear scaling of all key processes without downtime or service interruption.

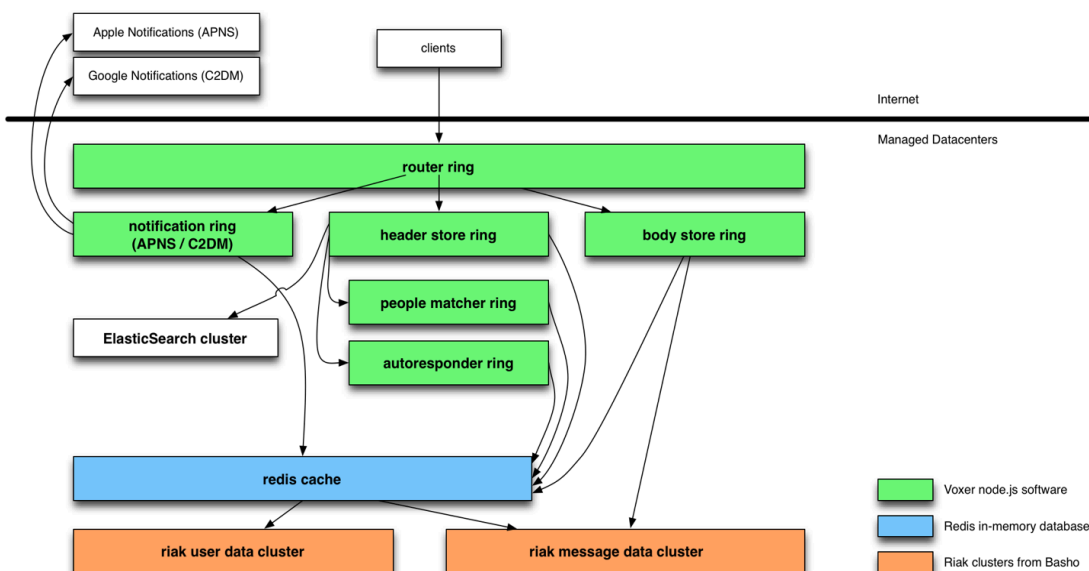
The Voxer Platform is formed primarily of internal services. The key services are tiered on clusters (rings) of server hardware:

- Router Ring
- Header Store Ring
- Body Store Ring
- Riak User Data Cluster
- Riak Message Data Cluster
- Redis Cache

In addition, there are other service components:

- Apple and Google Notification services
- Matching services (not used for contracted system)
- Data collection and reporting services (only some will be part of the contracted system)
- Email services

## Illustration of the Base Architecture



## Voxer Architecture and Feature List

There are two key software platforms on which the Voxer Platform depends:

- Several of the rings are constructed on top of Node.js.
  - “Node.js is a platform for easily building fast, scalable network applications. It uses an event-driven, non-blocking I/O model that makes it lightweight and efficient, perfect for data-intensive real-time applications that run across distributed devices.”
- Persistent storage is managed by Riak, a database from Basho:
  - “Riak is an open source, highly scalable, fault-tolerant distributed database.”
  - Riak stores data redundantly with a #-of-stores parameter. Voxer will be storing 3 copies of each data element, which improves throughput as well as storing redundant copies. The base system plan is therefore 3 servers.

### Feature List

Feature	Description of Feature
Single Communication Platform	The applications and communication system integrate live voice, messaged voice, and text in one conversational structure. Instead of users needing to pick different applications depending on whether they want to be live, messaged, text or voice, they choose the conversation they want to have and use whatever modality is right at that moment.
Transition Between Live Voice and Messaged Voice	The system can seamlessly transition between live and messaged voice enabling a significant set of new features. ( <i>* Features enabled by this patent pending technique.</i> )
Live and Messaged Voice in One Conversation	Live voice and messaged voice are managed in the same application in the same conversations with the same application protocol.
Any Network	The system uses any available IP network to transmit all communications (live voice, messaged voice, text.) Applications or users can choose which network to use for whatever optimizations required; depending on the options allowed by the operating system.
On Any Device	Applications can be custom developed to run on any computational device that has IP network support, microphone and speakers, and enough computational power: phones, desktop computers, laptops, mobile internet devices, gaming devices, television systems, or other IP computing devices.
Single Communication Platform	The applications and communication system integrate live voice, messaged voice, text, data, and location in one conversational structure.

## Voxer Architecture and Feature List

Store and Stream	All voice can be streamed out as fast as the network will allow and stored in the cloud as well as on the sending and receiving devices. This enables the system to service a wide range of behaviors (live voice and messaging, offline and on) in one system and to be resilient to network availability (like email) and efficient with network resources.
Optimal Receiver Experience	All parties in a conversation (including multi-party conversations) get to choose how to interact with every received message (live or later; voice or text) without ever missing any part of a conversation (including the live portions.)
Optimal Sender Experience	Every sender/speaker in a conversation gets to send/speak whenever they want with whatever supported media they want from whatever device they want. They can speak instantly.
Multiple Devices	Every conversation can be available and engaged in on any and all of a user's devices.
Persistence of live conversations	Live voice is managed the same way as messaged voice and is therefore stored, viewable, and manageable as persistent messages from specific participants. Users can listen to certain people and/or at certain times.
Conversations are the core	Application context can be centered on conversations, which can be contacts or groups of any size, regardless of the media types used by the participants.
Granularity of live voice conversations	Live voice is stored by speaker and time, so that live conversations are navigable by the user interface in the same way that instant voice and messaging systems are.
Multi-party, multi-media persisted conversations	Users participating in group chats get an optimal experience. They can be live or not, and never miss anything. They can contribute text or audio.
No need for fixed-mobile convergence	Users can participate in any of their conversations from any (and all) of their devices at any time. Everything is delivered everywhere a user desires.
All media types in single conversational thread	Each participant can choose the media of his contributions independent of other users. One person could be texting back and forth with another user who is speaking, for example.
Conversations prioritized by the system	The system will transmit live voice (a voice stream which is being listened to live) optimized for low latency; it will transmit everything else for optimal network capacity. (More sophisticated versions of this implementation will be delivered over time.)
Multiple Conversations	Simultaneous messages will be delivered simultaneously. If the user is listening live, the live message will be prioritized. Users can see information about everything that is arriving and are able to prioritize their engagement as they prefer.

## Voxer Architecture and Feature List

<b>Presence</b>	
Who is Speaking Now	Users can see who is speaking to them right now on any and all conversations.
Push-To-Talk, -Text, -Video, -Listen	Users can speak instantly to one or more other users. Users can listen instantly to live voice coming in or to messages that have been recorded. It works the same for voice or text.
Full duplex, multi-party instant experience	A user can quickly select a group of participants and start a conversation instantly.
Instant Reply	Users experience no system overhead when replying to a message. (No ringing, waiting, prompts.)
<b>Outgoing call management</b>	
Efficiency of guaranteed delivery to the device	A receiver can decline to take a live call knowing that 1) he will not miss the message, 2) there will be little overhead for the speaker, 3) there will be no overhead when he goes to pick up the message.
<b>Incoming Call Management</b>	
Comprehensive Incoming Call Management: Ignore	A user can choose not to participate in a call that the system is receiving live.
Comprehensive Incoming Call Management: Screen*	A user can choose to listen to a message that is coming in live without the speaker knowing that he is listening.
Comprehensive Incoming Call Management: Join *	A user can choose to join someone who is leaving a message and create a full duplex conversation.
Comprehensive Incoming Call Management: Play Faster *	A user can choose to play any message faster than it was recorded from the beginning even if it is coming in live.
Comprehensive Incoming Call Management: Catch Up to Live *	A user can choose to listen to a message (which is being left live) from the beginning. After listening faster the system will automatically transition into monitoring the call live, and then the user can choose to join live without having missed any of the conversation.

## Voxer Architecture and Feature List

Comprehensive Incoming Call Management: Monitor	A user can choose to monitor a conversation (channel or contact) and when anyone speaks on that conversation the voice will automatically be rendered live.
<b>Works Offline</b>	
Generate messages offline	Users can generate new conversations, messages, or replies when they have no network service. The conversations are “synchronized” when the user is back online.
Listen offline	Users can listen to their messages or multi-party conversations when they have no network connectivity.
Manage dropped calls	Users can continue to speak their messages when the network loses the ability to maintain a live connection knowing that the message will go out as fast and as soon as the network can send it.
Voice messages sent when network is poor	User can create messages that will go out even though the network is not good enough for a live call. There will be additional latency, but communication will happen.
<b>Conference Calling</b>	
Instant group calls	A user can instantly select a group or set of contacts and start speaking. There is no overhead. People available to speak live at that moment can join live as their applications notify them; others will have the conversation to listen to whenever they want. Such a call could, of course, be scheduled.
Media recorded	All communications are persisted.
Multiple simultaneous calls	A user can monitor all of his incoming messages on his various conversations and choose which one to participate in at any given moment. The conversation list will show him where he has unheard/unread content, and where someone is speaking to him live.
Pause a live call	A user can be on a conference call with several parties when he notices another message coming in live. He can “pause” the conference call, take the incoming call, and then user can catch up to live to get back into the conference call without missing any of the conversation.
<b>Public Safety Solutions</b>	
Full-duplex PTT	Users can have the same talk-instantly user model as Push-To-Talk without the limitation of a single speaker.
Live and Non-Live	Users will never miss a message because they were on another channel or had another priority. They will always be able to get the message.

## Voxer Architecture and Feature List

Multi-channel management	Users will be able to see who is talking on multiple channels and can prioritize their engagement accordingly, knowing they will never miss anything.
Publicly available IP networks	User communications can go over any publicly available IP network including commercial mobile data networks.