WHITEPAPER

JOIN.ME ARCHITECTURE

A technical overview of **join.me**'s secure and reliable architecture.

WHITEPAPER

join.me Architecture

join.me

Introduction

join.me is a simple but powerful screen sharing tool providing the following popular features:

- Private meetings
- Screen sharing
- Audio/video conferencing
- Recording
- Chat (text messaging)
- Ability to send files
- Ability to allow others to present
- Annotations
- Shared mouse control (remote control)
- Scheduled meetings
- Mobile Whiteboard

join.me components are compatible with most major operating systems:

- Windows
- Mac OS X
- iOS (iPhone, iPad)
- Apple Watch™
- Android

This document provides technical insight into how **join.me** provides these services robustly and securely.

Design principles

Simple and fast screen sharing tool
Usable within seconds of visiting the **join.me** website

High availability
Meets or exceeds 99.99% uptime

High security
Applies security at all layers using the latest security standards

High performance and scalability
Indefinitely scalable architecture

Redundancy
No single point of failure, geographical redundancy

Future trends
Built using the latest software development methods and technologies

Definitions

Presenter
The organizer of the meeting

Presenter Software
The **join.me** software instance running on the presenter's device

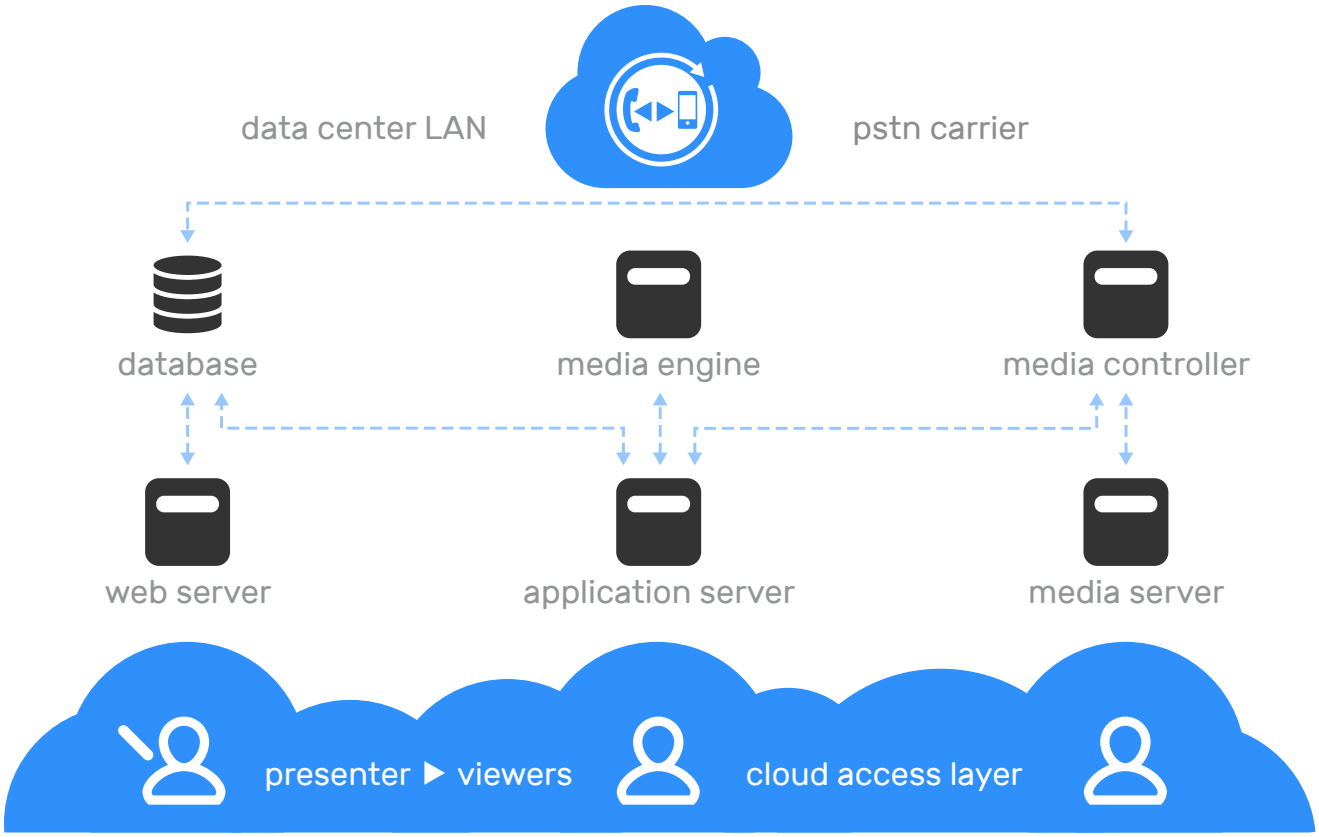
Viewer (one or more)
Participant of the meeting

Viewer Software
The **join.me** software instance running on the viewer's device

Session code
Nine-digit unique identifier of the meeting

Architecture Overview

A typical **join.me** session involves at least the following components:



Web server – User registration, account and meeting settings, meeting launch

Application server – Maintains meetings, distributes data among appropriate viewers

Media server – Distributes media streams among appropriate viewers

Database – Stores user profiles and meeting settings

Media controller – Controls media sessions and PSTN connections

Media engine – Post-processes media elements in order to provide recorded meeting video

Architect Overview (Continued)

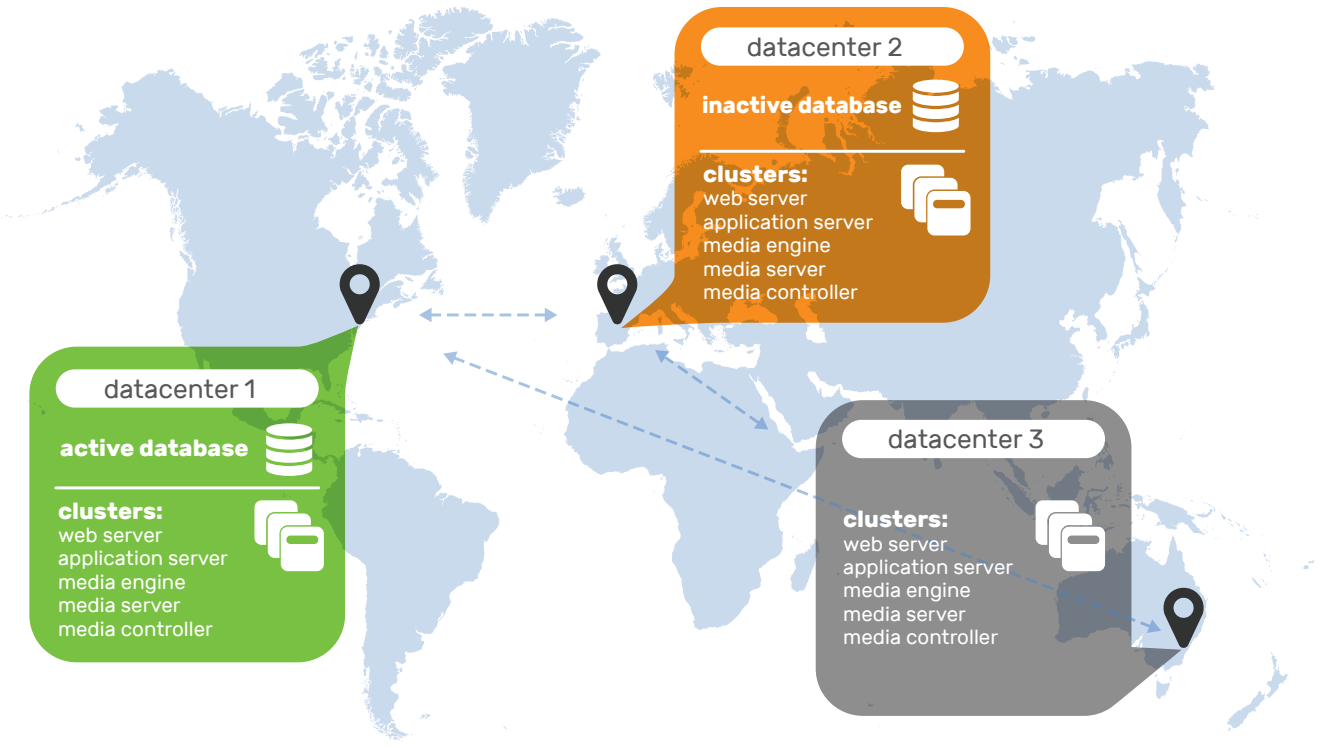
High availability

At the time of publication, **join.me** servers are hosted in **eight geographically distributed datacenters**.

Architect Overview (Continued)

Redundancy

A look at the server-side architecture helps us understand the highest level redundancy layers (illustrated below). In essence, after a session has been established, the Presenter Software communicates with one or more viewers through a specific Application Server. The system provides multiple layers of redundancy.



Datacenters are connected with a VPN mesh, meaning, for example, that if Datacenter #1 were to lose direct connection to Datacenter #2 it would communicate instead through Datacenter #3. Inter-datacenter communication is essential because only one datacenter hosts an active **join.me** database. In the illustration above this is Datacenter #1.

One of the datacenters (shown here as Datacenter #2) hosts a so-called passive database, which continuously receives database transaction logs from its active counterpart. In the event of a prolonged outage at Datacenter #1, the passive database in Datacenter #2 can quickly be brought online to resume operations.

Servers in the same facility communicate through the LAN; otherwise through the VPN mesh.

Presenters connect to an Application Server using geographic load balancing. A datacenter is allocated based on availability, proximity, and load, and then a particular Application Server within that datacenter is allocated based on availability and load.

Architect Overview / Redundancy (Continued)

Application-level logic ensures that participants always connect to the same Application Server as the presenter. This is important because the Application Server maintains session state in memory (for example, the Presenter's current screen).

Should an Application Server or datacenter go offline or become unreachable to the Presenter, the session is migrated to a different Application Server within seconds. Users only experience a brief pause. Webservers within a datacenter are also interchangeable as they share session state with each other through a large memory cache cluster.

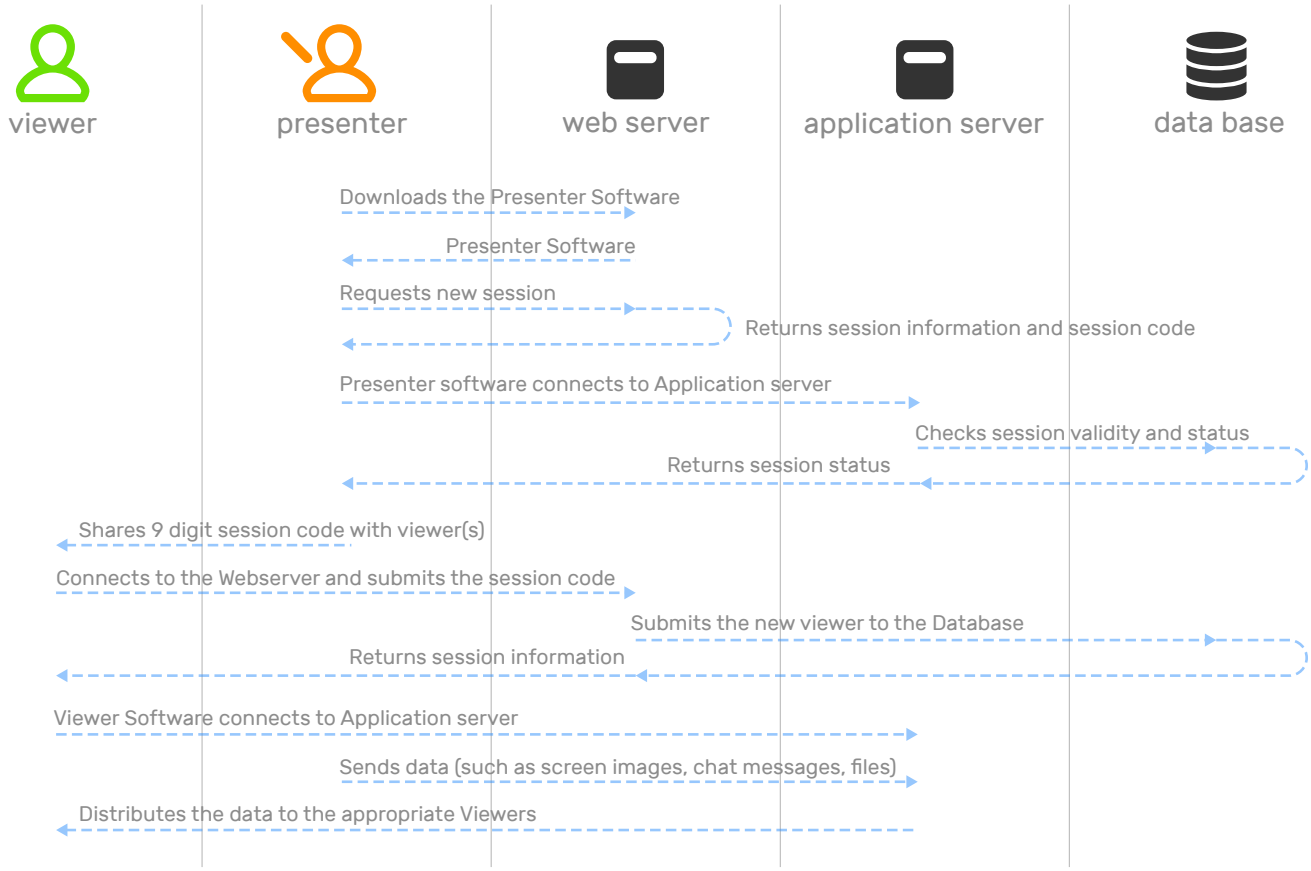
All components have several instances within a datacenter with automated fail-over that is nearly imperceptible to users. We support inter-DC fail-over in case of more serious problems.

How it works

Screen sharing, file transfer, chat

Session flow

A typical session flow looks like this:



All information travelling between components travels on an encrypted and secure channel. During file transfer and chat, data is relayed by the Application Server between peers: **join.me** stores no data during this process. See more in the "Data Security" section.

Bandwidth considerations

join.me's screen sharing aggressively optimizes for latency, using whatever bandwidth is available. The **join.me** codec is particularly efficient at handling bandwidth differences between meeting participants.

During screen sharing, **join.me** monitors the contents of the screen. For each screen update, it determines the smallest region that covers the updated parts. **join.me** then hands these regions over to the codec, which then compresses the captured bitmap and schedules the result for upload. If screen updates come in more quickly than the network can handle, the codec skips regions that overlap with later updates. This may happen because of a temporary network congestion, a permanently low-bandwidth uplink, or simply because screen contents animate/change too frequently for any reasonable network connection. Screen regions are uploaded to our proprietary meeting relay, which then uses a similar scheduler to distribute the regions to meeting participants.

The **join.me** screen sharing solution works without drastic, discontinuous quality level changes across the entire spectrum of possible network conditions. In extreme low bandwidth conditions, transmitted screen regions (or received regions when the weak link is on a viewer's side) will lag behind the original screen and screen quality will become progressively worse as updates become less frequent. In this case, the viewer may see inconsistent screen contents since some parts of the presenter's display may have been captured at different points in time. Since our relays use the same algorithm to distribute regions to viewers, we are capable of optimizing the screen share experience for each individual participant.

> **Tip:** Actual bandwidth requirements for high-quality screen sharing varies with screen resolution and the content being shared. As a general rule, we recommend at least 1Mbps of bandwidth on both the viewer and presenter side for sustained full-screen animation and shared control (when a participant is remotely controlling the presenter's screen). Content that changes infrequently (such as a PowerPoint presentation) can be usefully shared on links as weak as a GPRS connection.

We use a standard VoIP stack for our audio conference solution with an advanced, high-quality/low-bandwidth codec. To eliminate quality issues, VoIP needs at least 100kbps of sustained bandwidth available, preferably on a stable, low-latency connection. Network congestion may result in progressively larger audio dropouts until the link is restored.

How it Works / Screen sharing file transfer, chat (Continued)

The table below provides an estimate of bandwidth consumption in various scenarios.

What's being shared	Example	Details	Bandwidth
Small Video	YouTube	360p movie clip, very few screen block fragmentation	~800kbps
Fullscreen Video		1080p movie clip, few screen block fragmentation	~2500kbps
Website with various content	Facebook	Heavy scrolling	~800kbps
Website with mostly text	News Site	Mild scrolling	~400kbps
Presentation	PowerPoint	Many images, colors, animations	~800kbps
Presentation	PowerPoint	Mainly bullet points and solid background	~400kbps

Tip: See also Video bandwidth considerations.

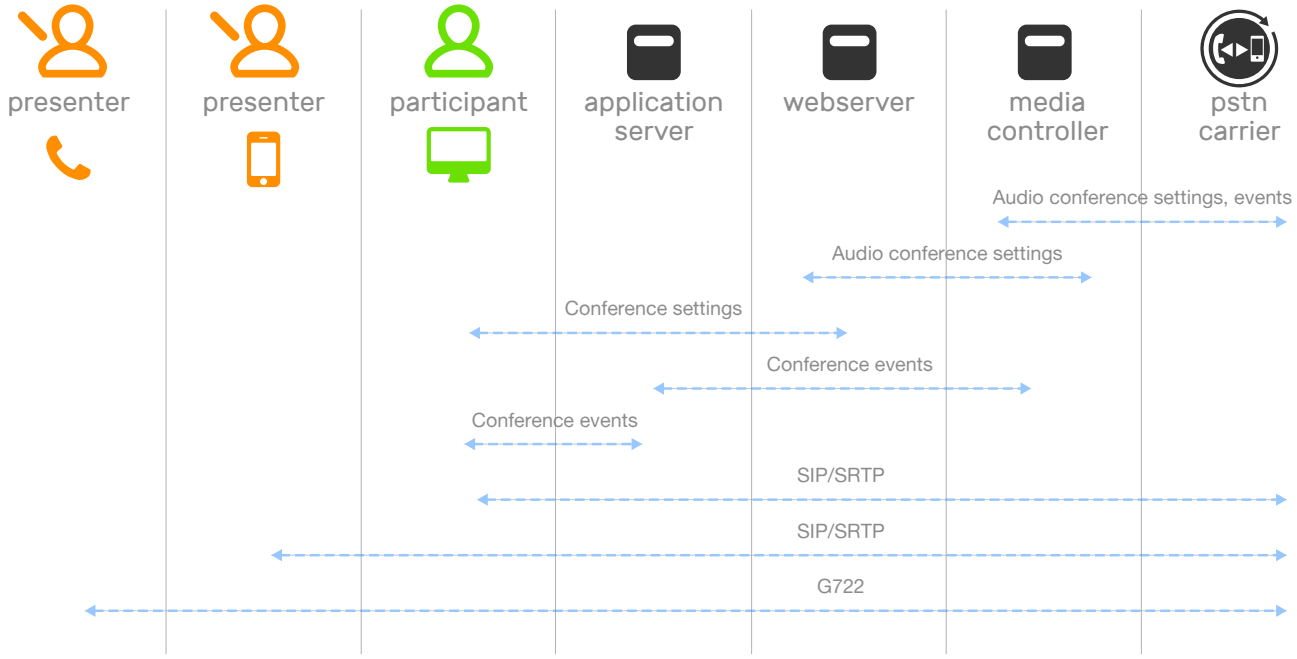
Audio conference

During an audio conference, **join.me** provides several ways for users to join:

- Traditional landline phones and mobile phones via dial-in numbers in over 50 countries
- SIP compatible devices
- Downloadable **join.me** native apps on Windows, Mac OSX, iOS, Android platforms
- **join.me** HTML5 application in Google Chrome browser (no download required)

How it Works / Audio conference (Continued)

To provide these services, **join.me** partners with a third-party telecommunication company. They provide the software- and hardware-based audio conference infrastructure that allows audio participants to connect to each other regardless of how they join. The presence of multiple endpoints, devices, and technologies forces **join.me** to handle several events and protocols. All of these protocols are safe and secure. The following schematic shows a typical audio conference:



join.me audio is based on the Voice over IP (VoIP) using the secure SIPS standard to facilitate communication between the computer and the audio infrastructure. In some network environments, this may require the following ports and IP ranges to be opened in the firewall:

Port ranges that should be opened:

- 5060-5100 TCP Outbound (SIP Signaling ports)
- 7800-32000 UDP Outbound (Voice Traffic ports)

The above ports should be opened to the following IP address ranges:

- 64.95.96.146
- 66.151.98.0/26
- 189.8.82.112/28
- 199.195.235.64/28
- 209.197.28.0/25
- 209.234.245.224/28
- 216.133.231.0/26
- 117.120.4.96/28
- 115.187.137.232/29

This information is subjet to change. Visit **help.join.me** for an up-to-date list of ranges.

join.me supports up to 250 concurrent audio participants per applicable meeting. **join.me** supports mix modes, so callers are provided the highest possible quality depending on their connections to the audio infrastructure. Audio conferencing features can be triggered by any type of callers if they have the organizer role in the conference. From **join.me** native and HTML5 apps, these features can be executed using the **join.me** user interface, from landline callers, mobile callers, and SIP devices, these audio features can be triggered using the numerical dial-pad (see the **join.me** online knowledge base for a full reference).

Video conferencing

join.me uses WebRTC technology to deliver video conferencing on the following platforms:

- Windows with the **join.me** desktop app
- Mac OS X with the **join.me** desktop app
- HTML5 client running in Google Chrome
- iOS and Android coming soon

join.me supports up to 10 video feeds and 250 video viewers per applicable meeting. During the video conference all video stream are relayed by a central service (Media Server). Video streams are encrypted by DTLS - SRTP. Video streams are only relayed on the LogMeIn infrastructure: No data is stored on our servers.

To use video conferencing, the following port ranges should be opened:

- 1853 outbound on UDP
- 443 outbound on TCP

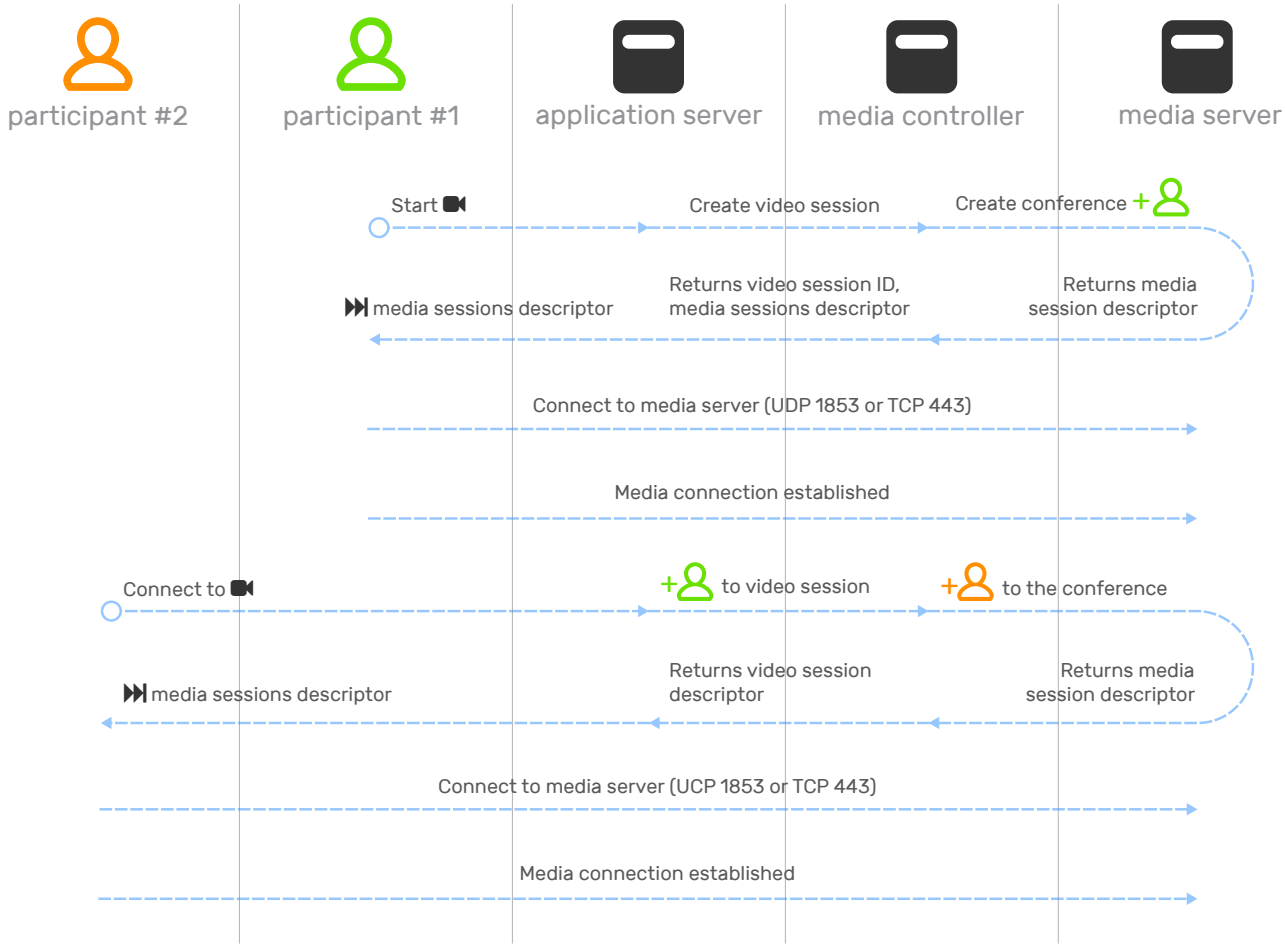
The following IP Ranges are used by **join.me** video:

- 64.94.18.0/24
- 64.74.103.0/24
- 63.251.34.0/24
- 216.52.233.0/24
- 212.118.234.0/24
- 111.221.57.0/24
- 117.20.45.0/24
- 190.210.65.0/24

This information is subjet to change. Visit **help.join.me** for an up-to-date list of ranges.

How it Works / Video conference (Continued)

The following schematic illustrates a session involving video conferencing:



Video bandwidth considerations

The WebRTC uses the VP8 codec, which requires minimum 100kbits/s bandwidth. The table below provides an estimate of bandwidth consumption in various scenarios.

	Resolution	Bandwidth (at 30 FPS)
Screen shared	320 x 240	100 - 500 Kbps
Screen paused (2 participants)	1280 x 720	1.0 - 2.0 Mbps
Screen paused (>2 participants)	640 x 480	0.5 - 1.0 Mbps

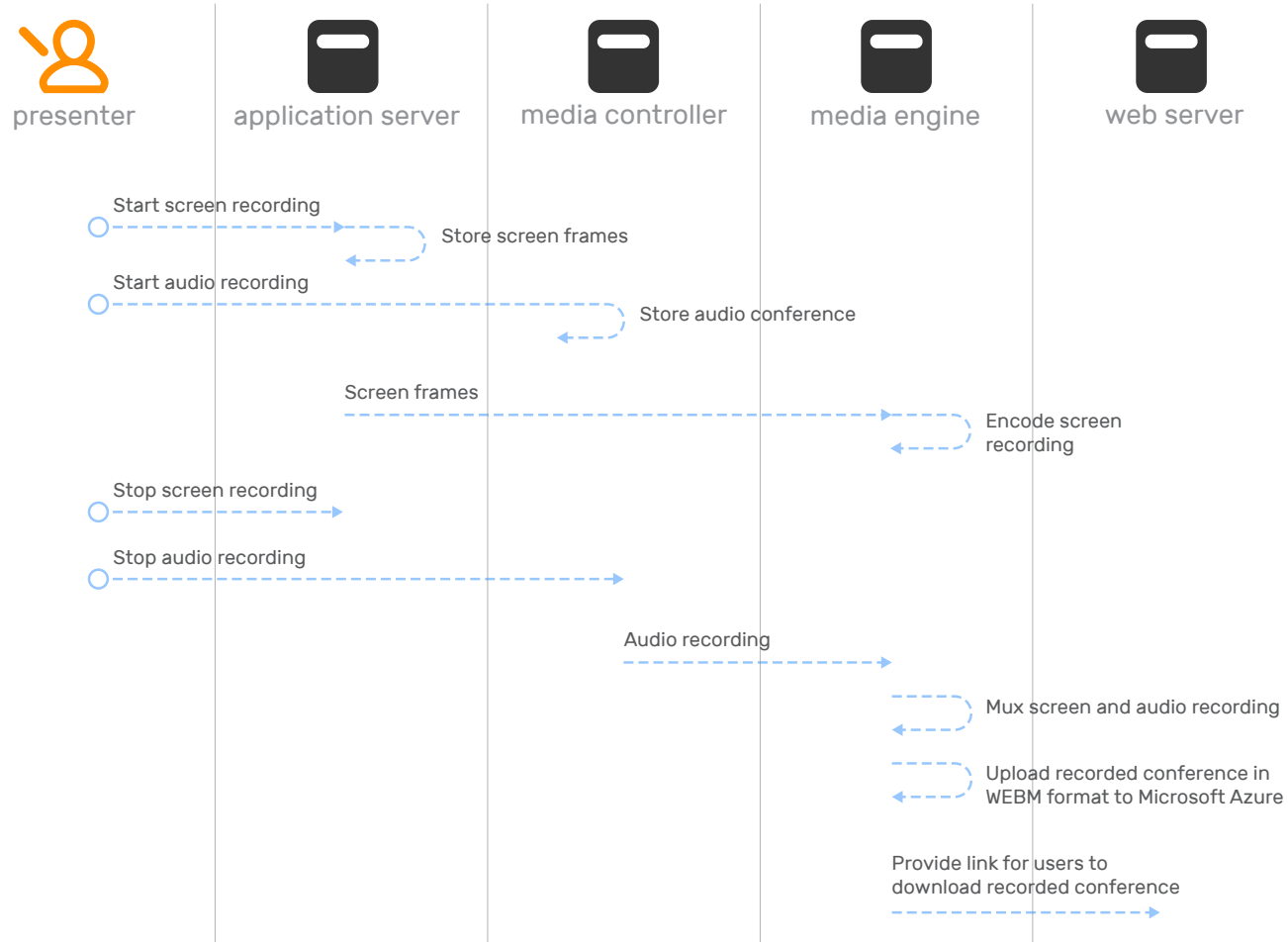
How it Works (Continued)

Recording

The process of recording a **join.me** meeting occurs on the LogMeIn infrastructure, thus protecting the meeting organizer from the computationally intensive process of recording, encoding, and storing a high resolution video. Since the video file is encoded as the meeting occurs, users can usually download and share their recording within a few minutes of their meeting.

Recording are saved in the WebM video format, which allows for playback directly in a web browser using HTML 5 without plugins. These WebM videos are stored in Microsoft Azure Storage. By default, they are stored both in the Azure storage region closest to the presenter's location at the time of recording and in a secondary region. This geo-redundancy provides high durability and availability. Upon request, customers can set their account to store files exclusively in a single region. Contact your **join.me** representative for assistance. There is a CDN over every storage account, so users will experience the optimal response times when accessing recorded content.

The following diagram describes this complex flow:



Data Security

join.me meets the following requirements for a secure communications system:

- Authentication of the communicating parties
- Confidential exchange of messages
- Detection of compromised messages

At the protocol level, **join.me** uses TLS for communications security. The key exchange protocol is ECDHE, while data encryption is AES (preferably AES256-SHA384). All modern browsers support AES256-SHA384, including current versions of internet Explorer, Firefox, and Chrome.

Every session is secured using the Application Server's TLS certificate. The Application Server terminates the SSL connections that are established by the viewer and the presenter. The need for this is obvious: While a single viewer/presenter pair could potentially employ end-to-end encryption and use the Application Server as a simple networking relay, this becomes unfeasible when multiple viewers are present. As designed, the system supports multiple viewers without placing bandwidth constraints on the presenter.

All **join.me** communications are secured using TLS, including access to the website itself.

No session data, such as screens, video, or chat logs, are stored on our servers unless you choose to record the session using the **join.me** recording feature. Recordings are saved to Microsoft Azure Storage cloud storage service. Microsoft Azure Storage meets several compliance requirements,including HIPAA. You can remove your recordings at any time.

Session & website security

Sessions are identified and secured by session codes. The ephemeral session codes are nine digits and are recycled after a session ends. At the time of publication, based on current usage trends, LogMeIn has determined that nine digits offer a de minimis chance of a collision. As **join.me** grows in popularity the length of ephemeral session codes may be extended beyond nine digits.

Static session codes (also known as "personal links") are available to paying customers. This is a user-defined alphanumeric string of up to 127 characters. A well-chosen "personal link" can be impossible to guess, but their static nature restricts their use to trusted parties. When a static code is in use, sessions start as locked and the presenter must approve individual viewers.

Data Security / Session & website security (Continued)

Viewers are authenticated to the system using the session code. The authentication to the presenter himself is typically done implicitly ("if the viewer has the code, it must have come from the presenter"), but viewers can also enter a display name, which is helpful when there are multiple participants. Presenters can use the website for ad-hoc sessions without logging in; in this case they are anonymous and are only authenticated by the system-generated session code. If they choose to log in (to access scheduled sessions or make use of a static code) they are authenticated with an email/password combination. **join.me** requires a valid email address (click-verified) and a password at least six characters in length. Upon registration, a simple password-strength indicator encourages complexity.

Presenters can choose to have the website remember them, which results in an already-logged-in state when they next visit the **join.me** website using the same browser/device combination. This "remember me" feature uses cryptographically secure random strings. No user ID is stored in cookies and no AES or other encryption is used. The auto-login cookie contains the key to a record in SQL server with the User ID. "Remember me" cannot be used to perform high-risk functions, such as changing account information. The user must always enter a valid password to perform high-risk functions.

Presenters can also download and install software to their computer to perform screen sharing sessions without further visits to the website. This software (commonly referred to as the **join.me** desktop app) can be attached to a presenter's account, thus providing access to scheduled sessions and static codes (personal link). When the presenter software is attached to an account, it receives a 32-character token generated by a cryptographically random algorithm using a 62-character alphabet (upper and lower case alphanumeric characters). This token is stored on the presenter's computer permanently and is used by the software to authenticate towards the system.

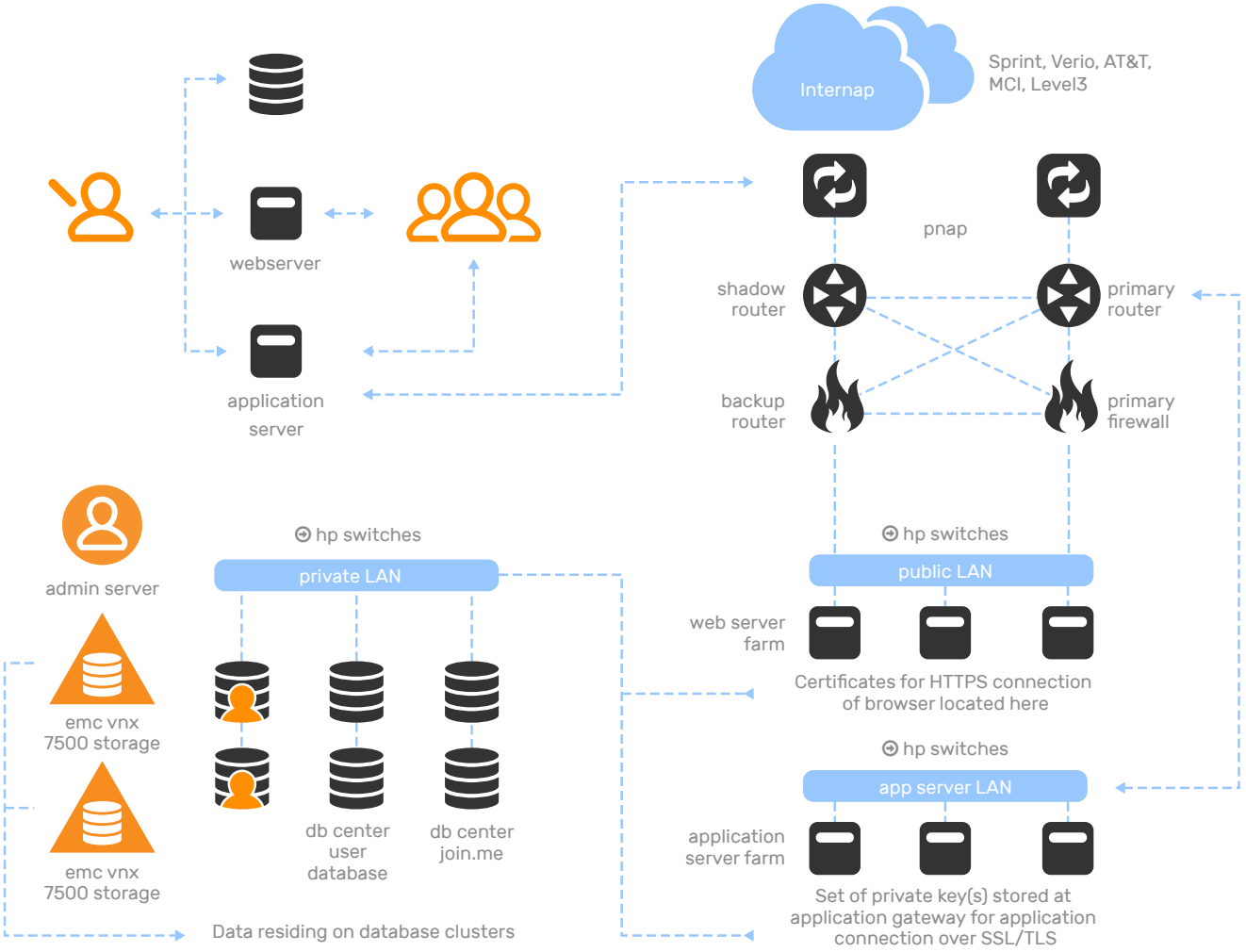
join.me offers two-factor authentication (2FA) for logging in to the website or native applications. The second factor of authentication can happen by using LogMeIn Authenticator (available on Android and iOS), TOTP (Google Authenticator and others), SMS (text-based) or e-mail.

Single sign-on (SSO) is also available; for more details, contact your **join.me** account representative.

Hosting Overview

The **join.me** service shares hosting facilities with all other LogMeIn services. These are Tier-1 Colocation Facilities with the following features:

- Multi-layer security control procedures, biometric entry systems, 24/7 CCTV and alarm monitoring
- Uninterruptible redundant AC and DC power, onsite backup power generators
- HVAC redundant design with air distribution under raised flooring for maximum temperature control
- Smoke detection system above and below raised floor; double-interlock, pre-action, dry-pipe fire suppression



Hosting Overview (Continued)

The LogMeIn servers (including all **join.me** servers) are in dedicated cages in all datacenters. These cages are protected by two-factor (biometric & PIN) electronic locks.

The LogMeIn infrastructure in each datacenter connects to the Internet with multiple 10Gb uplinks via multiple tier-1 NSPs. The edge routers are clustered in an active/passive configuration, and are connected to an active/passive firewall cluster in a fully meshed manner.

These in turn connect in a fully meshed manner to an active/passive IP-level load balancer cluster.

Behind the firewalls is a DMZ that contains the Webservers, the Application Servers, the Audio and the Video infrastructure. The Database Servers communicate with the servers in the DMZ using a private, non-routable LAN. An internal IDS system has been set up to warn of and help prevent malicious access.

Physical access to the cages is very strict and limited to a team of network engineers. Logical access for remote administration and software deployment is through LogMeIn's own services (LogMeIn Pro) and an out-of-band SSH gateway. Remote access for administrators is protected with two-factor authentication (username/password combination and a hardware or software token).

All servers undergo monthly self-audits as well as quarterly and annual third-party audits. These audits include network penetration testing and the review of server configurations.

Conclusion

While the **join.me** service appears deceptively simple, it is supported by a world-class architecture that can fulfill the needs of even the most concerned user.

Further information is potentially available upon execution of an NDA.

Want to learn more about how join.me can help grow your business? **Request a demo** or **call us at 1-877-251-8373**.