Rescue by LogMeIn

RESCUE ARCHITECTURE AND SECURITY

Overview Brochure

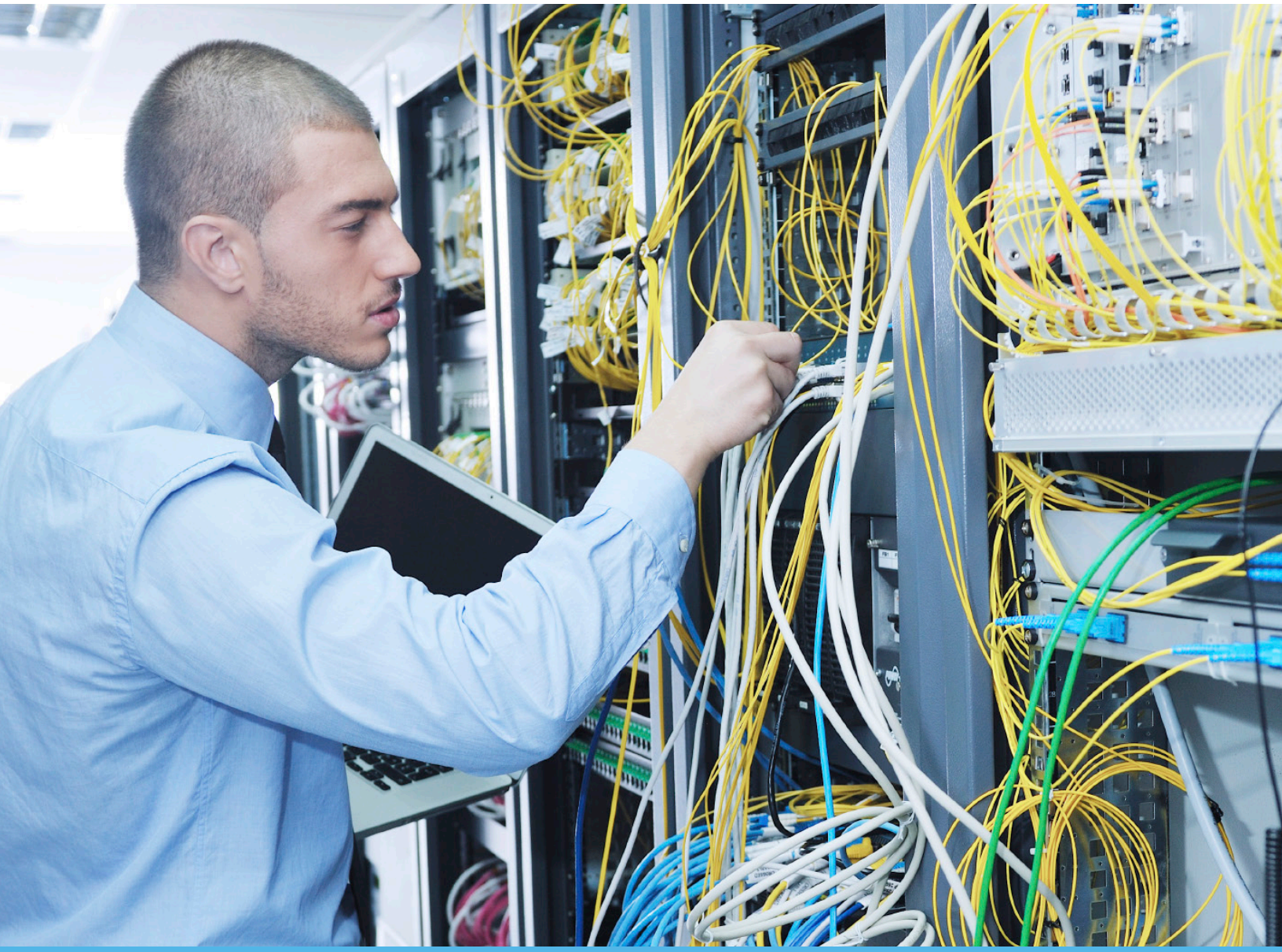


Table of Contents

INTRODUCTION

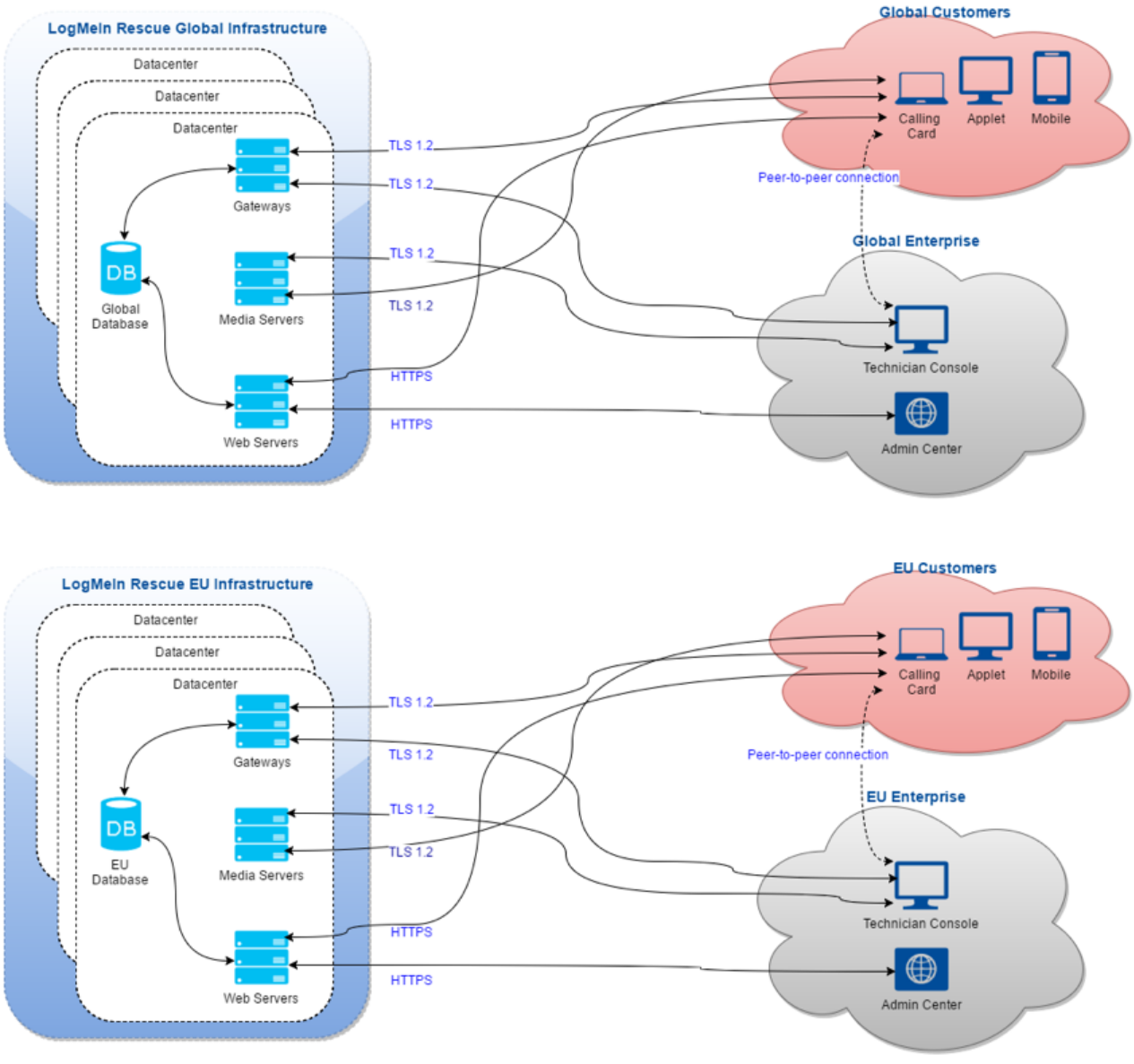
Scalability, security, reliability, ease of use. These four characteristics describe a great remote support solution, but they do not always go hand-in-hand. It's easy to find a remote support solution that provides two or three of these criteria, but a solution that delivers on all four fronts is rare. LogMeIn, Inc. provides just such a solution with LogMeIn Rescue.

Scalability. Whether you have a single technician or a call center with ten thousand employees, Rescue will get the job done.

Security. Support sessions are protected with end-to-end 256-bit AES encryption. Support operations must be permitted by the end user before the technician can perform them. Support session logs are stored in a database in encrypted format and can be queried later. Remote control sessions can be recorded to a video file.

Reliability. Rescue is hosted in six carrier-grade datacenters with a fully redundant infrastructure.

Ease of use. Your technicians will be up and running in a matter of hours. Your supported end users will get help with a few clicks. No software has to be installed by either party.



DATA CONFIDENTIALITY

Often security is equated to data confidentiality, data confidentiality is equated to encryption, and encryption is characterized by the symmetric cipher used and its key length. These misconceptions lead to misnomers such as "256-bit AES secure." Needless to say, this is misleading.

A secure online system should always meet the following objectives:

- Authentication of the communicating parties
- Negotiation of encryption keys without a man-in-the-middle intercepting them
- Confidential exchange of messages
- Ability to detect if a message has been modified in transit

SSL/TLS, short for Secure Sockets Layer and Transport Layer Security, has been designed to provide support for the above steps. Originally created by Netscape Communications Corporation in the mid-90s, it has since become the de-facto standard for secure communications over the Internet, and has been endorsed by Visa, MasterCard and American Express.

The SSL implementation used by LogMeIn Rescue is OpenSSL (http://www.openssl.org). LogMeIn always uses the latest available version. At the time of publication, the version used by Rescue is 1.0.2j.

KEY AGREEMENT

When a support session starts and a connection is established between the supported user and the technician, their computers must agree on an encryption algorithm and a corresponding key to be used for the duration of the session. The importance of this step is often overlooked, and this is somewhat understandable: it seems like a mundane task that should be simple and straightforward.

It is, however, anything but simple: to counter so-called man-in-the-middle attacks (where computer C would position itself between computer A and B and impersonate the other party to both A and B) again, certificates must be employed. Since neither the technician nor the end user have server software and an SSL certificate installed on their computers, they both turn to one of the LogMeIn Rescue servers and perform the initial phase of the key agreement with this computer. Verification of the certificate by both the Technician Console and the end user applet ensures that only a Rescue server can mediate the process.

MESSAGE EXCHANGE

TLS allows for a wide range of cipher suites to be used and the communicating parties can agree on an encryption scheme they both support. This has two primary purposes: first, the protocol can be extended with new cipher suites without breaking backwards compatibility, and second, newer implementations can drop support for suites that are known to contain cryptographical weaknesses.

Since all three components of the LogMeIn Rescue communications system are under LogMeIn's control, the cipher suite used by these components is always the same: AES256-SHA in cipher-block chaining mode with RSA key agreement. This means the following:

- The encryption keys are exchanged using RSA private/public key pairs, as described in the previous section
- AES, short for Advanced Encryption Standard, is used as the encryption/decryption algorithm
- The encryption key is 256 bits long
- SHA-2 is used as the basis of message authentication codes (MACs). A MAC is a short piece of information used to authenticate a message. The MAC value protects both a message's integrity, as well as its authenticity, by allowing the communicating parties to detect any changes to the message.
- Cipher-block chaining (CBC) mode ensures that each ciphertext block is dependent on the plaintext blocks up to that point, and that similar messages cannot be distinguished on the network.

The above ensures that data traveling between the supported end user and the technician are encrypted end-to-end, and only the respective parties have access to the information contained within the message stream.

AUTHENTICATION AND AUTHORIZATION

Authentication and authorization in LogMeIn Rescue serves two distinct purposes.

Authentication ensures that the technician or administrator logging in to the Rescue system is in fact who they claims to be. In Rescue, authentication is handled in a very straightforward manner: Technicians are assigned login IDs (usually matching their email addresses) and corresponding passwords by their administrators. These credentials are entered into the Login form on the LogMeIn Rescue website at the start of a technician workday.

In LogMeIn Rescue, the Rescue system is first authenticated to the technician (or rather, the technician's web browser) with its 2048-bit premium RSA SSL certificate. This ensures that the technician will be entering his username/password into the right website. The technician then logs in to the system with his credentials.

LogMeIn Rescue does not store any passwords, but instead uses SCrypt to create hashes from passwords that are then stored in the Rescue database. The hashes are salted with a 24 character string generated by CSPRNG for each unique password.

LogMeIn Rescue gives administrators a number of options for password policy:

- Administrators can enforce a minimum required password strength and a maximum password age – a built-in meter shows administrators and technicians the strength of the chosen password
- Technicians can be forced to change their Rescue password upon their next login

LogMeIn Rescue also allows Administrators to implement a Single Sign-On (SSO) policy. Security Assertion Markup Language (SAML) is employed, which is an XML standard for exchanging authentication and authorization data between security domains (between an identity provider and a service provider). Technicians then have access only to pre-defined applications and a single SSO ID to log in to those applications. At the flick of a switch, a technician's SSO ID can be disabled.

Authorization, on the other hand, happens very frequently – at least once during every remote support session. The supported end user, after downloading and running the support applet, will be contacted by a technician. The technician can chat with the end user via the applet, but any further action, such as sending a file or viewing the end user's desktop, requires express permission from the user. A "single prompt" can also be implemented. This is intended for lengthy remote support work where the customer might not be present for the entire duration of the session. If this flag is enabled for a Technician Group, then the technicians in that group can request a "global" permission from the customer, and, if granted, will be able to perform actions such as viewing system information or entering a remote control session without being further authorized by the end user.

Administrators can also impose IP address restrictions on their technicians. When selected, the IP addresses available can be restricted to a very narrow list. Technicians assigned to a particular task can then only access Rescue from pre-approved IP addresses for that task.

The administrator of a Technician Group can also disable certain features in the Administration Center. For example, members of a Technician Group can be prevented from receiving files from end users. Here are some of the permissions an Administrator can grant or deny:

- Launch remote control
- Reboot
- Launch Desktop Viewing
- Record sessions
- Send and receive files
- Start private sessions

- Launch File Manager
- Request Windows credentials
- Send URLs
- Allow clipboard synchronization
- View system information
- Deploy scripts
- Use single prompt for all permissions
- Transfer sessions
- Allow screen sharing with customers

The Rescue system is also authenticated to the supported end user. The applet, downloaded and run by the user is signed with LogMeIn's code signing certificate (based on a 2048-bit RSA key), and this information is typically displayed to the user by their web browser when they are about to run the software.

The supported user is not authenticated. It is up to the technician to determine who the user is, either via chat or a telephone conversation. The Rescue system does provide authentication-like mechanisms such as unique PIN codes, but these are used for routing the support session to the correct private or shared queue, and should not be construed as an authentication system.

AUDITING AND LOGGING

Any remote support solution must place strong emphasis on accountability. LogMeIn Rescue provides two distinct auditing features.

First, the so-called "Chat log" is saved in the Rescue database. The Chat log is transmitted to the Rescue servers by the Technician Console in real time, and contains events as well as chat messages that pertain to a particular support session. For example, a log file would display when a remote control session is started or ended, or when a file is sent by the technician to the end user. Accompanying metadata, such as the name and MD5 Hash thumbprint of a transmitted file, is also included in the log when applicable. The Chat log database can be queried from the Administration center. At the time of publication, LogMeIn's data retention policies stipulate that the contents of the logs will be made available online for two years after the end of a remote support session and archived for two years after that. To facilitate integration with CRM systems, LogMeIn Rescue can post session details to a URL. Administrators can choose whether to allow chat text to be excluded from these details. Additionally, all records of chat texts between technicians and clients can automatically be omitted from the session details stored at the Rescue Data Center.

Second, LogMeIn Rescue allows the technicians to record the events that transpire during a desktop viewing or remote control session into a video file. This is a very important feature for accountability and liability reasons. The recording files are stored in a directory specified by the technician. In the case of a large support organization, this location should be on a network server. The disk space taken up by these recordings varies widely and depends entirely on the contents and compressibility of the supported end user's desktop. Based on an analysis of millions of remote control sessions utilizing LogMeIn's technology, the average disk space requirement for one minute of remote control data is between 372 and 1024 Kbytes. The recordings are stored direct to AVI or in an intermediate LogMeIn proprietary format that can be converted to standard AVI files by the "Rescue AVI Converter" application downloadable from help.logmein.com. The LogMeIn proprietary format, called RCREC, can cut recording size by about 10%.

DATA CENTER ARCHITECTURE

LogMeIn Rescue is hosted in state-of-the-art, secure data centers with the following features:

- Multi-layer security control procedures, biometric entry systems, and 24/7 closed-circuit video 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

The LogMeIn Rescue infrastructure itself is highly secure and reliable:

- Redundancy on the server component level: redundant power supplies and fans, RAID-1 mirrored harddisks
- Redundancy on the server level: depending on role, active/passive or active/active clusters
- Redundancy on the datacenter level: Six datacenters (US West Coast, US Central, US South-Central, US East Coast, London UK, Frankfurt Germany) with near-instant failover capabilities
- Dual redundant firewalls with only ports 80 and 443 open
- Active/passive database clusters
- Redundant load balancers including SSL
- Load-balanced and redundant web and application server clusters
- Load-balanced and redundant gateway server clusters

An Overview of the LogMeIn Rescue Gateway Hand-off Process

When the digitally signed Rescue applet is started on a machine:

- It contains a session authentication GUID (Globally Unique Identifier), which has been embedded in the .exe file as a resource by the site when it was downloaded
- It then downloads a list of available gateways from secure.logmeinrescue.com
- It picks a gateway from the list and connects to it using TLS ; the gateway is authenticated by the applet using its SSI Certificate
- The gateway authenticates the applet in the database with the GUID and registers that the user is waiting for a technician

When a session is picked up in the Rescue Technician Console:

- A request is sent to the gateway with the session authentication GUID to relay connections between the Technician Console and the client applet
- The gateway authenticates the connection and starts relaying data at the transport level (it does not decrypt relayed data)

When a connection relay is started, the parties try to establish a peer-to-peer (P2P) connection:

- The applet starts listening for a TCP connection on a port assigned by Windows
- If the TCP connection cannot be established within a time limit (10 seconds), an attempt is made to establish a UDP connection with the help of the gateway
- If either a TCP or a UDP connection is established, the parties authenticate the P2P channel (using the session authentication GUID), and it takes over traffic from the relayed connection
- If a UDP connection has been set up, TCP is emulated on top of the UDP datagrams using XTCP, a LogMeIn-proprietary protocol based on the BSD TCP stack

Every connection is secured with the TLS protocol (using AES256 encryption with SHA256 MAC). The Session Authentication GUID is a 128-bit, cryptographically-random integer value.

DATABASE

- All data containing sensitive information is secured with 256-bit AES encryption (Chat log and Custom Fields).
- The Rescue database is backed up automatically every 24 hours. The backup database is stored in the data center with the same encryption as the original.

- Rescue's Data Residency Option allows you to choose where to store end-user data: either within the European Union (Frankfurt, London) or in the USA. LogMeIn guarantees that those choosing data residency within the EU will only connect to datacenters within the EU and that customer data remains solely within the chosen region. There is no connection between our EU-based and USA-based datacenters.

RESCUE MEDIA ARCHITECTURE

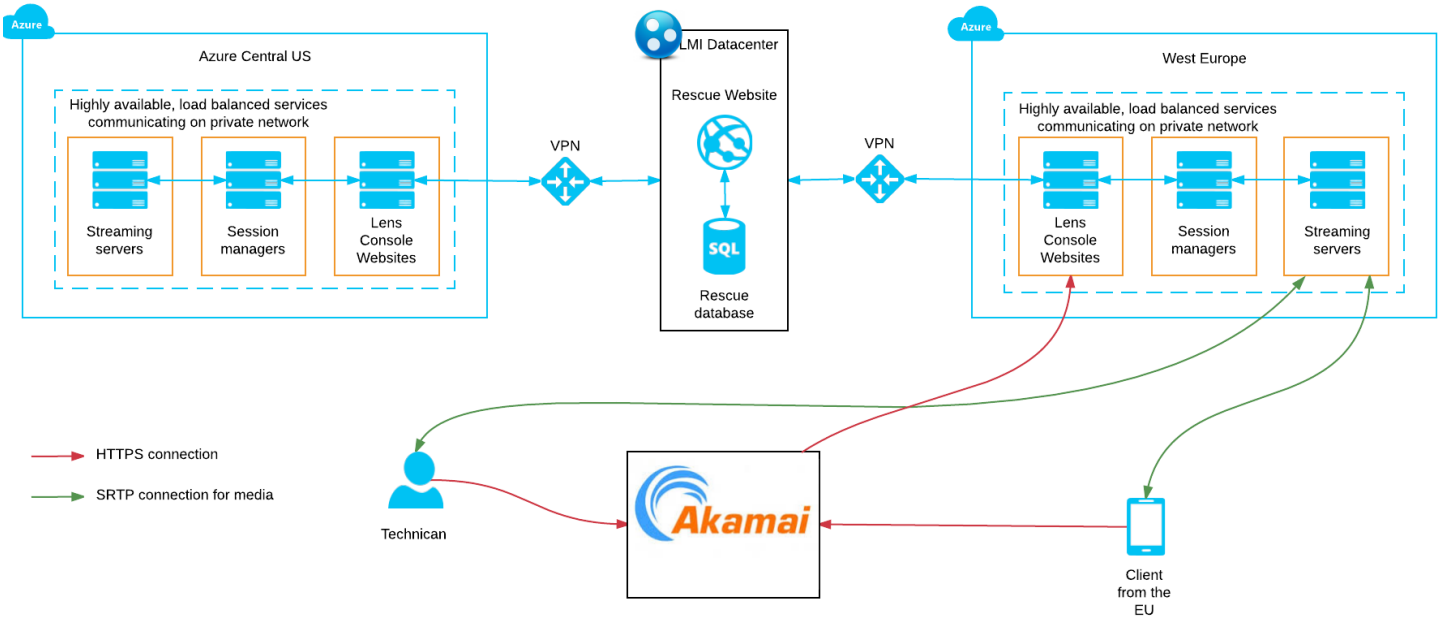
The Rescue Media Service is a WebRTC-based standalone service that powers Rescue Lens video streaming. It manages so-called "conferences" for Rescue session that use the Lens feature. Conference participants (peers) join and leave conferences and clients send video and audio streams for other participants to receive. Lens sends video content in a unidirectional stream from the Lens Applet to the Technician Console.

There are three main components of the media service: the MediaSDK, the Session Manager, and the Streaming Server. These components manage the process of creating/destroying and joining/leaving conferences. These components communicate via the existing secure connections between the Technician Console and the website and between the Lens App and the website.

MediaSDK

The Media Service was built on top of WebRTC with a thin wrapper around the WebRTC code base. This so-called MediaSDK is used in the Technician Console and the mobile Lens apps.

RESCUE LENS WEB CONSOLE - HOSTING OVERVIEW



Session Managers

The Session Manager is a simple load balanced website providing a REST API to manage (create/destroy/join) the conferences. The Session Manager only accepts requests from the website.

Streaming Servers

The Media Service is using the Jitsi open source streaming server solution to handle streams between peers (the Technician Console and the Lens app). Both the Technician Console and the Lens app are connected to the streaming server. The Lens app streams its video content up to the streaming server. The Technician Console streams video content down from the server. Jitsi behaves like a relay server between peers. A Lens session has two streams (one is sent, the other is received).

LENS WEB CONSOLE

Lens Web Console is the second generation of the Rescue Lens live interactive video support tool. With Lens Web Console, technicians can use Google Chrome to support customers without downloading additional software or plugins. From an architectural perspective, Lens Web Console is hosted in Microsoft Azure. It does not require an application gateway since both technician and end user are communicating securely via the Lens Web Console website.

Hosting Overview

At the time of publication, Lens Web Console is hosted in two Microsoft Azure regions: Central USA and Western Europe. Akamai Global Traffic Management guarantees that Lens sessions are always created in the region closest to the supported end-user. The following diagram shows how Lens Web Console's architecture provides for load balancing, fail-over, and high availability.

Data Security

Microsoft Azure regions are connected to the LogMeIn datacenter through a secured VPN operated and supervised by LogMeIn Network Operations Center staff.

The authentication and authorization of the supported end user happens on the Rescue website using OAuth authorization. No user data is stored in the public cloud. Session information is stored in the public cloud, but it is cleaned up after the Lens session has ended. Once the connection has been established between the technician and end user, all traffic goes though the Lens Web Console infrastructure hosted in Microsoft Azure.

For information about Microsoft Azure and compliance requirements (such as HIPAA), see Microsoft Azure Legal Information.

LOGMEIN RESCUE HIPAA CONSIDERATIONS

Although LogMeIn cannot control the content shared by users during a support session, the LogMeIn Rescue service is designed to meet strict security standards and help HIPAA regulated entities comply with relevant regulatory guidelines.

Access Controls

- Define permission-based access on a granular level (such as permitting some technicians to use remote view, but not remote control)
- No data from remote devices are stored on LogMeIn datacenter servers (as discussed above, only session and Chat log are stored). In addition, chat text logs can be removed from session details.
- Permissions can be set to prevent Technicians from transferring files, thus eliminating their ability to take files from remote devices.
- The end user must be present at the remote device to permit remote access

- The end user maintains control and can terminate the session at any time
- Technicians can be prevented from using certain features until the end user has explicitly granted them permission (example: remote control, desktop view, file transfer, system information, reboot & reconnect)
- Access rights are automatically revoked when the session is terminated
- Predetermined time of inactivity forces automatic logoff
- Hosted at redundant carrier-grade data centers with restricted, secured access

Audit Controls

- Option to force session recording, with the ability to store audit files on a secure shared network
- Technician sessions and remote session activity is logged on the host computer to ensure security and maintain quality control (successful logins, unsuccessful logins, remote control started, remote control ended, reboot initiated, logout)
- Person or entity authentication
- The technician's identity is defined by a unique email address, or via an SSO ID, and the technician must be authenticated
- Excessive number of unsuccessful login attempts locks the account
- Technicians can be allowed to log in only from approved IP addresses

Transmission Security

- End-to-end 256-bit AES encryption of all data
- MD5 Hash for enhanced traceability of file transfers

CONCLUSION

Choosing a remote support solution is often a decision based on features and pricing. If you are reading this document, then it is likely that LogMeIn Rescue has met your needs in these categories. With the information set forth above, we believe we were able to prove that the architecture behind Rescue provides the right levels of scalability, security, reliability and ease of use.