HipLink Mobile

Enterprise Notification Solution

Communication & Security Overview

The Enterprise Mobile Notification Solution from HipLink Software provides the most advanced infrastructure for any organization that delivers messages in real-time with capabilities to support business messaging workflows, and deliver rich content. This whitepaper gives an overview of the HipLink Mobile apps communication and security design.

HIPLINK MOBILE ENTERPRISE NOTIFICATION SOLUTION
COMMUNICATION & SECURITY

ABOUT THIS WHITEPAPER

This Whitepaper provides a complete, 360-degree overview of the communication and security design for the HipLink Mobile notification solution for the Enterprise. It provides vital information to IT Managers, Security Officers, Information Officers, Administrators and other decision makers, and helps them understand the communication flows, its requirements for mobile and desktop notification, and how HipLink's solution provides security and other related functions.

INTRODUCTION

HipLink Mobile provides a cutting-edge platform for the enterprise which is unique from other mobile notification solutions. Firstly, the solution frees the user from dependence on carrier SMS text services and adds support for communication over an Internet protocol using either Wi-Fi or the carrier data networks. Along with it, the solution provides complete end-to-end security for messages with support for remote administration and device control. Above all, the solution provides real-time messaging services with full back-up to platform specific push notification services and carrier SMS text reminders in case the User doesn't connect. The solution also enables advanced mobile workflows which are now one of the basic needs of today's enterprise.

CLIENT SERVER MODEL

The solution follows a simple client - server model in which the HipLink application server provides the Service and the Clients are mobile apps on Android or iOS platforms that consume the service to receive messages.

THE PROTOCOL – HNP

The solution's communication is based on a proprietary protocol called HNP (HipLink Notification Protocol). This protocol has been designed to allow interconnection with clients on heterogeneous platforms. The protocol design has special considerations for security, scalability, extensibility, and usability.

COMMUNICATION CHANNEL

The HNP protocol uses TCP/IP as the channel for its communication. This enables the client apps to communicate with the server using any network type whether it is LAN (Ethernet), WLAN (Wi-Fi), carrier data networks (GPRS/EDGE, EVDO, HSPA, LTE, etc.). Also, the communication is session oriented and its packet ordering, sequencing, and retransmission all depends on the TCP layer.

CIA - CONFIDENTIALITY, INTEGRITY & AUTHENTICATION

The HipLink solution uses the TLS (Transport Layer Security) protocol to implement full confidentiality, integrity and authentication (CIA) in the communication path. All communication is fully encrypted, with proper message checksums to ensure message integrity, along with digital signatures to authenticate the server side.

NEW MESSAGE NOTIFICATION

The HipLink solution integrates with specific push notification services for the phone OS platforms. This allows HipLink Mobile to push alerts to users, alerting them to new waiting messages. This communication follows different workflows depending on connectivity, and often goes through an OS-specific notification service gateway. This contacts the mobile device by its own defined means and then routes the push notifications to the device. As a further safeguard, HipLink Mobile has a built-in SMS text mechanism as a reminder if the User fails to timely retrieve the message. Currently, HipLink Mobile supports integration with Apple Push Notification Service (APNS) for iOS clients and with Google Cloud Messaging (GCM) for Android devices.

SOLUTION DESIGN ATTRIBUTES

The HipLink Mobile notification solution has several design attributes, of which the most essential are related to communication and security aspects. These attributes are:

ONE MECHANISM PROTOCOL FOR ALL PLATFORMS

The solution is based on a platform neutral communication mechanism or protocol that enables communication to all types of HipLink mobile and desktop platform clients. There is no functional distinction on the HipLink server-side for the receiver type, whether it is Android, iPhone or a Desktop client. This enables the solution to support environments with heterogeneous platforms.

WI-FI SUPPORT

The solution support of Wi-Fi-enabled devices has never been easier than with the HipLink Smartphone apps. The user can set the Smartphone for Wi-Fi communication to the HipLink server when in the office, and HipLink automatically switches between the carrier's data network and the Wi-Fi network when in range.

The mobile apps are designed to give preference to Wi-Fi and automatically switch from carrier data network to Wi-Fi when the device enters a Wi-Fi domain. Not only does this have a positive impact on the use of carrier data, but its primary purpose is to preserve the battery life of the phone.

PERSISTENT CONNECTION

The solution requires that each client maintain a persistent connection with the HipLink server. The persistent connection allows for real-time messaging. The mobile apps are designed to reconnect automatically when an existing connection breaks or when network coverage changes from Wi-Fi to carrier data network or vice versa.

ALWAYS REAL-TIME

The mechanism provides real-time or close to real-time notification with messaging failures identified in the reports and logs at the earliest possible moment. If the device is offline and a message cannot be delivered, a configurable offline queue will hold the message and attempt redelivery on regular defined intervals.

HIGHLY RELIABLE

The solution provides a deterministic messaging service with clear, on-time delivery and message read reports. The reliability is further enhanced with the inclusion of support for offline messaging, which is configurable. If delivery mechanisms go offline or messaging support is disabled, Users can define backup workflows to continue to attempt message delivery using other protocols. Redundant communication channels are used to ensure that a message is delivered to the device timely and efficiently.

SECURE COMMUNICATION

The solution provides security at the transport layer encasing all communication between the HipLink server and the mobile clients in a secure transmission channel. This channel acts as a secure tunnel between the server and each mobile client and all communication that takes place within it. This process not only makes the messaging secure, but all other communication such as signaling, file transfer, etc. is also secure.

The secure communication layer is TLS based and is established after a proper authentication procedure. This mechanism provides security against replay and other common attacks.

SUPPORTS 2-WAY COMMUNICATION

The 2-way mechanism in HipLink Mobile supports complex messaging workflows which are now required in everyday operations in most organizations. It allows HipLink and the mobile clients to send default or custom response choices with the message and the ability for the User to select a choice with a single tap, and have the response sent immediately back to the HipLink server to be reflected in reports. Depending on the response, it can initiate other actions on the server.

SUPPORT SECURE FILE TRANSFER MECHANISMS

The solution supports messaging with attachments. The attachments are sent separately from the message, but shown in association with the message on the mobile app. The attachments can be in any format such as pictures and video, Microsoft Office files, or other types, such as Adobe Acrobat. It supports any file type that the smartphone supports opening.

END-TO-END SECURITY, CONTENT PROTECTION, & CONTENT CONTROL

The HipLink Mobile solution provides full end-to-end security using the industry accepted cryptographic algorithms such as AES, SHA2, RSA, etc.

End-to-end security means:

1. All communication between the HipLink server and HipLink Mobile is secured in transit, in both directions.
2. All messages and their attachments are saved on the devices in a dedicated, encrypted container within the application.
3. Functions and options for the application can be enforced through General Policy on the server that the user cannot change or override.
4. All the data saved on the devices can be managed from the server with automatic deletion, message expiration, and remote administration.

To ensure message confidentiality, integrity and authenticity, along with security against various attacks, HipLink uses the best possible mechanism Transport Layer Security (TLS), which is recognized as a security standard in the enterprise community. TLS, the successor to SSL, offers a robust security protocol meeting IETF standards for connection security. Using TLS, HipLink Mobile supports a wide variety of bit-rate encryption options that include 128, 196 and 256-bit AES encryption standards configurable by the administrator.

One of the more interesting features HipLink developed using this standard is a “single session” handshake process. By using this method, the TLS encryption key is constantly changing with each communication session between the HipLink server and the mobile device.

This short “time to live” makes cracking the encryption extremely difficult as the key is constantly regenerating with each communication transaction. The security features apply to all phases of message delivery, both messages sent to the phone and responses back.

MEETS REGULATORY COMPLIANCE REQUIREMENTS FOR CONFIDENTIAL MESSAGING

HipLink Mobile is both HIPAA (Health Insurance Portability and Accountability Act) and CJIS (Criminal Justice Information Services) compliant, per published requirements and provides the following safeguards:

- Confidential data (e.g. PHI, warrant data) protection in activity logs and audit trails
- End-to-end secure communication between HipLink servers and mobile apps with AES encryption algorithm using TLS. This covers all data in transit and at rest
- Device-level data protection with the AES encryption algorithm (using 256 or other bit keys) to protect messages and received media content
- Use of server-side enforcement for allowing confidential data to be sent to only authenticated, secure Users and prevent delivery of secure information to insecure devices
- Mobile app access security to prevent access of information (messages and media content) by unauthorized personnel
- Remote administration to manage and control mobile app access
- Ability to remove (all or selected) data on a device from remote admin console
- Ability to disable mobile app from remote admin console

HipLink Mobile supports a unique messaging mode called "Confidential" in the application that triggers a series of defined, secure messaging measures in the messaging workflow. This mode ensures the overall process is compliant with stringent regulatory requirements. All messages sent in “Normal” mode (not confidential) are also secured on device and during transit, but the content is revealed in reports, audit trails, and logs. Moreover, “Confidential” messaging mode enforces messaging to only secured recipients. Messages to insecure devices are failed at the server level.

REALTIME MESSAGING WITH PUSH NOTIFICATION SUPPORT

Real-time delivery of messages in HipLink Mobile is supported so that, when connected, messages can be received almost instantly after dispatch in the application. All messaging workflows are designed to be 100% deterministic, letting the sender know the message delivery status immediately whether it succeeded, failed, or was put on hold

for later delivery. In the case of a message pending delivery, the messages are queued as offline messages and the user is notified. Queued messages are delivered when the recipient comes back online.

PUSH NOTIFICATION

The HipLink solution supports integration with mobile platform-specific, push notification services to dispatch new message alerts to offline Users alerting them to come online. The solution currently supports Apple Push Notification Service (APNS) for the iOS devices and Google Cloud Messaging (GCM) for the Android devices. Full details of the HipLink implementation for each service is described below.

ALWAYS ONLINE

HipLink Mobile implements an "always online" methodology for mobile Users. It is designed to always keep the User online for easy access to new messages in real-time. The Android app has special capability to start automatically upon device reboot and to run in the background. The apps and their processes are designed to have a minimum impact on the device battery. The iPhone app is tested to have an average battery drain of approximately 2% per hour, whereas Android app has approximately 3% per hour with normal operations. This can vary based on individual User phone settings and the overall age and health of the battery.

SECURE INTERNET CALLING

Besides text-based messaging, which comes in the form of an Alert or Chat conversation, HipLink Mobile supports secure Internet calling. The call mechanism supports both audio and video calls. This feature is dependent on a widely used service called TokBox which offers HIPAA compliant P2P Audio and Video calling. Since these types of calls are carried over IP networks, there are no additional cellular voice charges. The call uses either Wi-Fi or cellular data Internet channels, though the preference is always Wi-Fi if available. A corporate TokBox account is required to enable this function in HipLink Android and iOS clients.

APPLE PUSH NOTIFICATION SERVICE (APNS)

INTRODUCTION

The Apple Push Notification Service (APNS) is a service created by Apple that was launched together with iOS 3.0 in 2009. The APNS uses push technology through a constantly open IP connection to forward notifications from the servers of third-party applications to Apple devices. The service has undergone major enhancements over the years since its launch.

Notifications can cover a variety of different forms including badges, sounds, and custom text alerts. Due to limitations in reliability, security, and compliance, HipLink does not send the actual message to the iOS device via APNS. The APNS is only used to let iOS users know that there is a new message that needs to be reviewed in HipLink Mobile and gives them the option to launch the app.

REQUIREMENTS

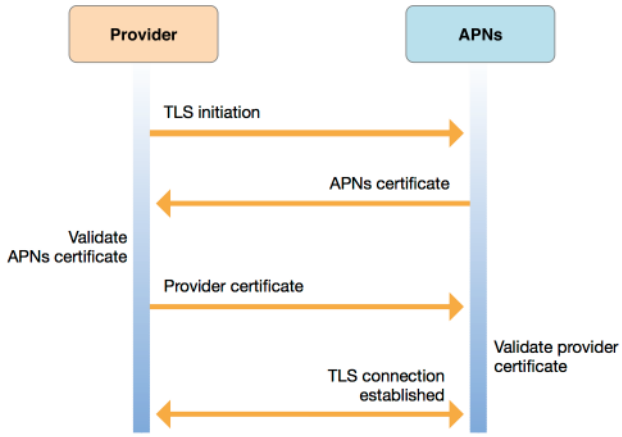
To support the APNS integration and alert functionality, there are specific requirements for certain artifacts in order to enable the HipLink server software to send push notifications to the iOS devices.
These are:

1. Define App ID
2. A SSL Certificate for the server product

3. The private key of the SSL certificate

In addition, a new key pair is required with the private key saved in a separate file and public key as a certificate request (.CSR) file. The certificate request is then sent to Apple to be signed and sealed. App IDs must be registered in Apple's iOS developer portal for the iOS app.

After that, the SSL certificate is available to download from the provisioning profile panel. This SSL certificate and the private key is then used in the HipLink server to configure the integration with the APNS gateway.



Note: The SSL certificate and the associated private key file MUST be in PEM format.

For details, see online documentation at Apple's developer portal:
https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/APNSOverview.html#//apple_ref/doc/uid/TP40008194-CH8-SW3

HOW IT WORKS

The iOS app is created with a provisioning profile enabled with APNS support.

1. The iOS app registers for APNS alerts with the APNS gateway and receives its device-token.
2. The iOS app sends its token to HipLink server.
3. The HipLink server must be configured with APNS SSL Certificate and its associated private key.
4. When it needs to send a notification, the HipLink server connects to APNS gateway using SSL/TLS sockets, and dispatches the alert.
5. The APNS server sends the alert to the iOS device.

NETWORKING & COMMUNICATION REQUIREMENTS

The solution needs access to the Apple's APNS gateway for push notification support. Following are the access requirements:

Server	Port	Purpose
api.push.apple.com	443	HipLink server hands off push notification to APNS gateway for delivery. When SSL certificate is of type production.
17.0.0.0/8 *	5223, 443	iOS device persistent connection to APNS gateway to register, and receive push notifications.

* The IP address range for the push service is subject to change; the expectation is that providers will connect by hostname rather than IP address. The push service uses a load balancing scheme that yields a different IP address for the same hostname. However, the entire 17.0.0.0/8 address block is assigned to Apple, so network administrators can specify the range in their firewall rules.

GOOGLE CLOUD MESSAGING SERVICE (GCM)

INTRODUCTION

The Google Cloud Messaging (GCM) for Android is a service that lets developers send data from their servers to Android applications on Android devices, and upstream messages from the User's device back to the cloud. This can be either a lightweight message telling the Android application that there is new data to be fetched from the server (for instance, a "new email" notification) or it can be a message containing up to 4KB of payload data so apps like instant messaging can consume the message directly. The GCM service handles all aspects of queuing messages and delivery to the target Android application running on the target device.

GCM requires devices to run Android 2.2 or higher and have the Google Play Store application installed or an emulator running Android 2.2 with Google APIs. However, one is not limited to deploying the Android applications through Google Play Store. It uses an existing connection for Google services.

REQUIREMENTS

The GCM support integration requires certain artifacts (or credentials) in order to enable server software to send push notifications to the android devices. These are:

1. Sender ID
2. Application ID
3. Registration ID
4. Google User Account
5. Sender Authentication Token

The Sender ID and Server Authentication Token (#1 and #5) must be configured and known to the server in order for the application to be eligible to send messages to GCM server. The Application ID (#2) is then used by the Android app to get Registration ID (#3) which then needed to be shared with the server to get notifications. A Google User account (#4) is only required if the device OS version is before 4.0.4.

In HipLink, the only requirement is for the admin to setup the Sender ID and the Sender Authentication Token in the HNP configuration panel.

HOW IT WORKS

The Android app is created with GCM support enabled.

1. The Android app registers for GCM alerts with the GCM gateways and receives its Registration ID.
2. The Android app sends its Registration ID to HipLink server.
3. The HipLink server must be configured with Sender ID and its associated Sender Authentication Token.
4. When HipLink needs to send a notification, the server connects to GCM gateway using an HTTP session and dispatches the alert.
5. The GCM server then sends the alert to the Android device.

NETWORKING & COMMUNICATION REQUIREMENTS

The solution needs access to the Google's GCM gateway for push notification support. Following are the access requirements:

Server	Port	Purpose
1) gcm.googleapis.com	5235	HipLink server hands off push notification to GCM gateway for delivery.
2) gcm-staging.googleapis.com	5236	HipLink server hands off push notification to GCM gateway for delivery.
3) play.google.com	5228	Android device persistent connection to Google Play Services gateway to register, receive push notifications, and other Google events.

TOKBOX INTEGRATION FOR SECURE INTERNET CALLING

INTRODUCTION

TokBox® is a development platform for WebRTC applications. WebRTC is a standard for enabling Real-time Communication (RTC) in the browser without the need of any other plugin. It includes components for building high quality audio and video chat applications.

Using TokBox as a middleware for audio and video services, HipLink Mobile offers interactive media features within its app for next generation customer needs. This service is completely secure, scalable and HIPAA compliant.

REQUIREMENTS

Customers who wish to leverage the secure Internet calling features of HipLink Mobile require a subscription to the TokBox service. Please note that TokBox has a separate pricing structure. Once a service subscription is secured, TokBox issues an ID and a Key. The ID and Key are used to access the TokBox API and services from within HipLink.

The Secure Internet Calling feature on iOS and Android apps will only be enabled once ID and Key are configured on the HipLink Server side.

The requirement for a separate subscription for every installation is because every organization has different needs for Audio and Video call volume as well as archival options. The individual organizational needs will affect the subscription fee for each installation.

HOW IT WORKS

1. HipLink Server is configured with TokBox Key and ID.
2. When Android and iOS apps establish a session, they are notified regarding the availability of this feature and the required meta information.
3. Whenever the User wishes to make calls, the Key and ID are used to access TokBox Services.
4. TokBox services facilitate HipLink client apps in establishing a secure call between two HipLink client apps.

NETWORK AND COMMUNICATION REQUIREMENTS

Since TokBox is an Internet-based service, certain network access is required at the client side. Please refer to: https://support.tokbox.com/hc/en-us/articles/115001376024-What-are-the-TokBox-network-connectivity-requirements- for up to date details of network requirements of TokBox.

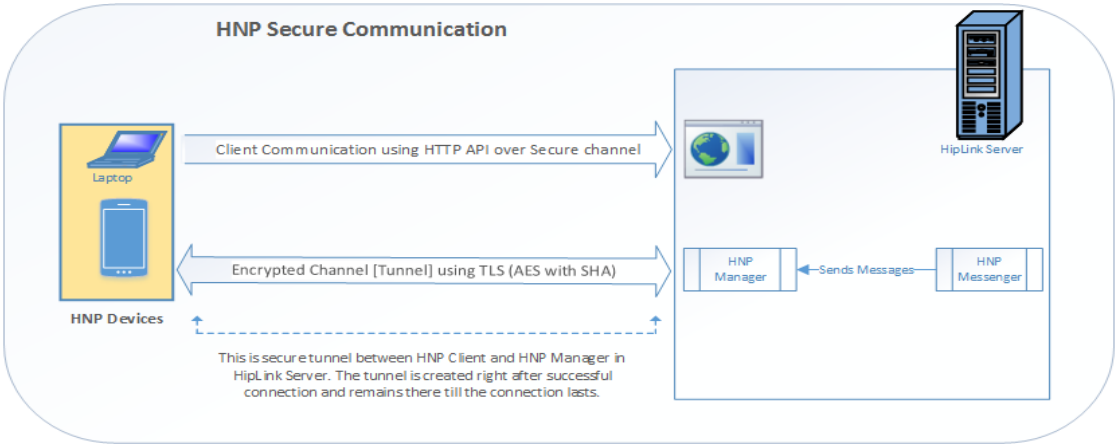
COMMUNICATION OVERVIEW

The HipLink Mobile Notification Solution is based on a simple client - server model. The HipLink Server opens up the HNP server port for real-time communication, whereas the clients - Android, iOS or Windows Desktop, connect to the server port to establish the communication link. Besides the HNP Service port, HipLink also exposes HNP API over HTTPS for clients to consume. If the HipLink server is configured for push notifications, it will make connections to the APNS/GCM gateway on a fixed port to send push notifications. Each mobile device (Android, iOS) will make connections to their cloud gateways to register themselves for the push notification services and to get new notifications.

HNP - THE PRIMARY COMMUNICATION PROTOCOL

HNP is the primary communication protocol and the basic requirement for HipLink Mobile to work. The HipLink server will open up an HNP service server port. All client apps connect to this port. This port is completely configurable, and can be set or changed from within the HNP configuration panel.

This communication link is protected and secured by using Transport Layer Security (TLS) encryption. As per the TLS security requirements, the server must be configured with a server certificate and its associated private key. The TLS handshake will establish a secure tunnel between the server and client. All alerts and notifications are sent to devices using this tunnel. If a device is found offline, Push Services are used to notify the device regarding the pending notification. HNP API over HTTPS is used by the clients for several in app functions that include communication, messaging status, reporting, and miscellaneous other requests.



FILE TRANSFER COMMUNICATION

File Transfer is a secondary communication link. It is required only if the organization is going to be transferring files that are attached to a message being sent to the device or from the device. The server port is defined in HipLink's global settings and can be changed to any value per specific requirements. The default value is set to 19979.

When a client app logs into HipLink server, it receives the IP and port of the file transfer service. The client then uses these when making connections, unless there is a modification to the settings which would be changed when the User logins again. When a client app receives a message with attachment(s), the attached file IDs are provided, along with their decryption keys. Hence, on the separate connection to the file transfer service, the files are queried and downloaded in encrypted form. They are later decrypted using the keys that were already transmitted securely with the message.

Starting in HipLink 6.5, client versions for iOS and Android, the legacy file transfer service was deprecated in favor of the more robust and scalable HTTPS-based file transfer mechanism. Once all clients adopt this new mechanism, this service will be decommissioned and file transfers will function without any additional network requirement.

SENDING APNS PUSH NOTIFICATION

This is only required when the server is configured for APNS support. When a message is sent to an offline iOS device user, the server dispatches a notification through APNS to the iOS device to let it know about the new message waiting in server. This is done by connecting to "api.push.apple.com."

RECEIVING APNS PUSH NOTIFICATION

Each iOS device running HipLink Mobile communicates with the Apple APNS gateway server. Each device maintains a single persistent connection with the APNS gateway. After establishing the connection, the app first registers itself for the APNS notifications and then gets its device token. Secondly, on the same connection, it waits for push notifications. When a push notification is sent from the HipLink server, the APNS gateway forwards the notification object to the device over this connection.

SENDING GOOGLE CLOUD MESSAGING (GCM) PUSH NOTIFICATION

GCM is used when server is configured for push notification support. When a message is sent to an offline Android User, the server dispatches a notification through GCM to the phone to let the user know that a new message is waiting on the server. This is done by connecting to "gcm.googleapis.com:5235."

The connection is secured using SSL/TLS, in which the server uses the configured Sender ID and Authentication token to authenticate itself as a client to the GCM gateway. After establishing the secure connection, the push notification is sent to the GCM gateway in a structured XML format along with the device's registration ID already communicated to the server by the device at login.

RECEIVING GCM PUSH NOTIFICATION

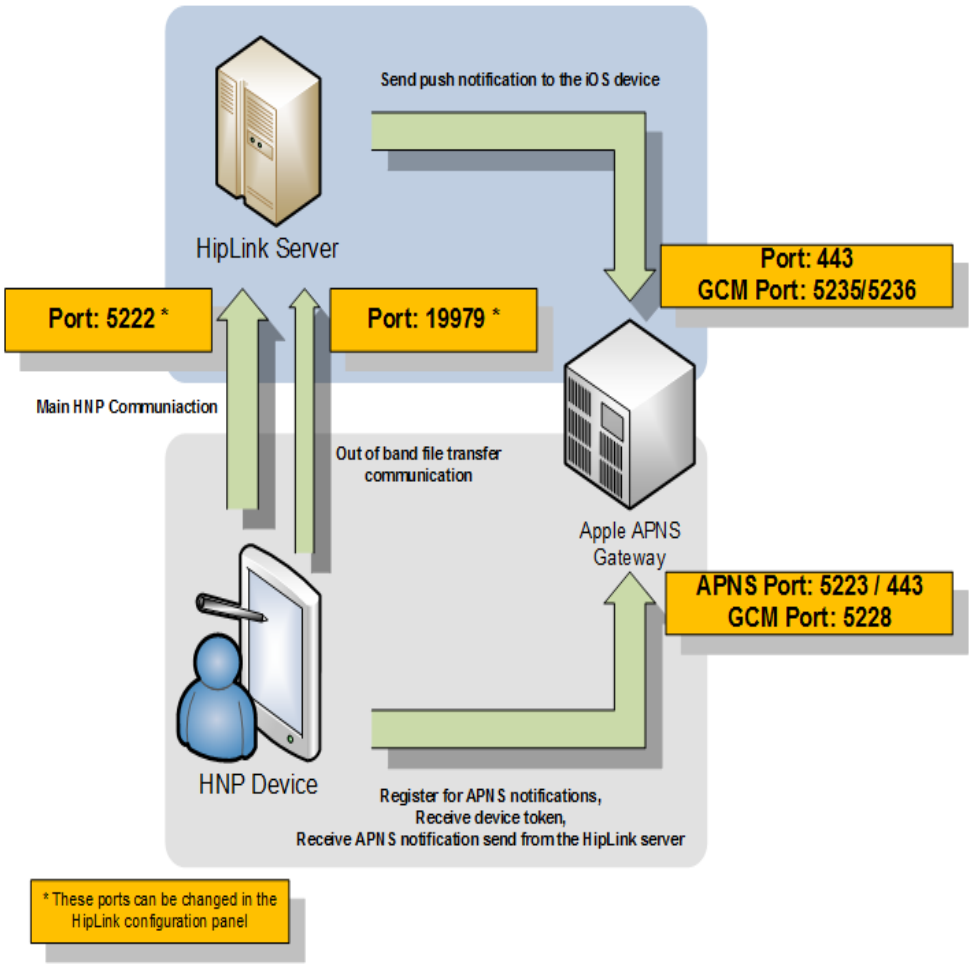
Each Android device running HipLink Mobile communicates with the Google Play Services gateway and the device maintains a single persistent connection. First, the app registers itself for GCM notifications using the application ID and then gets its registration ID. Next, the registration ID is then sent to the HipLink server upon login. The device uses the established connection for various purposes like getting event notifications, syncing data, and receiving and sending push notifications. When a push notification is sent from the HipLink server, the GCM gateway forwards the notification object to the device over this connection. The Google Play client service running in the device that owns the connection, then relays and invokes the target application to handle the push notification object.

SUMMARY OF NETWORK COMMUNICATION PORTS

The following table provides a list of all the network communications that occurs across ports in the HipLink Mobile Notification Solution. This information can be used to define network firewall rules.

Host	Port	Protocol	Type	Remote	Purpose
HipLink Server	5222 (default)	TCP	Inbound <====	HipLink HNP Clients	HNP communication over persistent connection between HipLink server and client apps. The port can be changed.
HipLink Server	5223 (default)	TCP	Inbound <====	HipLink HNP Clients	HipLink clients communicate with HipLink server. This port is configurable and can be changed.
HipLink Server	19979 (default)	TCP	Inbound <====	HipLink HNP Clients	HNP communication over persistent connection between HipLink server and client apps exclusively for file transfer. The port can be changed.
HipLink Server	443	TCP	Outbound ====>	APNS Gateway	Push notification handoff to APNS gateway through HTTP/2 provider API [api.push.apple.com]
HipLink Server	5235	TCP	Outbound ====>	GCM Gateway	Push notification handoff to GCM gateway [gcm.googleapis.com]
HipLink Server	5236	TCP	Outbound ====>	GCM Gateway	Push notification handoff to GCM gateway [gcm-staging.googleapis.com]
HipLink Server	587	TCP	Outbound ====>	HipLink SMTP Relay Server	Push notification fallback to SMS through HipLink SMTP Relay Server [mymail.myoutlookonline.com]
iOS App	5223	TCP	Outbound ====>	APNS Gateway	iOS device persistent connection to APNS gateway to register and receive push notifications
iOS App	443	TCP	Outbound ====>	APNS Gateway	iOS device persistent connection to APNS gateway to register and receive push notifications, if 5223 outbound port is inaccessible
Android App	5228	TCP	Outbound ====>	Google Play Services	Android device persistent connection to Google Play Services gateway to register and receive push notifications

The following illustration shows the overall communication and data flow in the solution. The arrow shows the connection initiation direction, whereas data flow will be bidirectional in each of the communication flows. The communication flow engaging APNS gateway is only applicable for iOS HipLink Mobile clients.



HIPLINK MOBILE HNP COMMUNICATION PROTOCOL IN DETAIL

The HipLink Mobile proprietary communication protocol is HNP and is the primary link between the HipLink server and the HipLink client apps for communication. The client app creates a permanent connection to the server, and then attempts to keep it alive for the life of the app's process, if possible. In the case where platform restrictions or network circumstances keep devices from maintaining an always available connection with server, devices rely on periodic polling using the HNP API over HTTPS.

TLS (TRANSPORT LAYER SECURITY)

The connection is made over TCP and, after a successful connection, TLS is started immediately. First, the server sends its public key in its server certificate and the client uses it to establish the initial secure connection. The TLS protocol then performs a handshake procedure in which cipher capabilities are negotiated, and a shared secret key is established. After that, the newly created key is used to secure the subsequent session.

The shared key has an expiration time and, when it expires, it forces both sides to go through the key creation procedure again. This is done to ensure the session stays protected and if a key is compromised, it is changed to a new one. It also helps to protect against replay attacks, and man-in-the-middle attacks.

HNP AUTHENTICATION & SESSION

The first process is the user authentication. In this process the User ID, password, and activation key are sent to the server. The server authenticates the user, checks its activation, and then responds with user permissions including access & privileges in addition to any optional settings. Permissions further disable or enable app functions, whereas Settings change the usage behavior. After that, the HNP session is initiated.

HNP OPERATIONS

Once a HNP session is started, the client app can send any request to execute an operation on the server through the HNP API. If the server has a message for the device, it pushes the message to the device through a secure tunnel, if available, or it pushes the message to prompt a login.

Besides receiving messages in this session, the client can send requests to:

- query for contacts
- send a message to other contacts
- query a sent message status
- confirm/reject a message
- log out of the session
- run custom actions
- update subscription to groups
- update their profile info
- establish secure internet video or audio call

HNP SESSION LIFECYCLE

The session will remain active until:

- user does not log off explicitly or
- the session expiry configured on server is reached

WHO WE ARE?

The first implementation of HipLink was in 1995 in a major project with Nextel. Since then, HipLink Software has grown to serve hundreds of customer organizations while benefiting millions of Users. Successfully deployed across multiple verticals, HipLink is meeting the needs for IT alerting, alarm management, emergency response, mass notification and business continuity.

With corporate headquarters in the heart of Silicon Valley California, HipLink is a stable, woman-owned business, that continues to demonstrate a high commitment to its customers, while introducing numerous technological innovations. HipLink has been the premier provider of software for wireless text and voice communication to global organizations of all sizes for over twenty years.

The HipLink platform provides critical communication software serving Healthcare, Public Safety and enterprise organizations. With unparalleled leadership, patented technology, a proven track record and extensive customer base, HipLink delivers state-of-the-art solutions customized to meet the unique needs of its customers. Customers and partners include Allina Healthcare, Province of Alberta, Blue Cross Blue Shield, Dignity Health, Providence, Ameriprise, Motorola, Spillman, and AT&T to mention a few. For more information, please visit www.hiplink.com

HipLink Software
800 524-7503 Toll Free
HLsales@hiplink.com
www.hiplink.com