Audience & Purpose

The primary audience for these diagrams consists of network and BigFix administrators in an organization charged with securing their networks for the operation of BigFix. For example, these diagrams can provide input into how firewalls should be configured. Some of the diagrams might be used by BigFix Administrators to debug BigFix behavior across the network. These diagrams use default deployment ports, e.g., 443, 52311, 52315, you may have installed with a different set of deployment ports.

There are three types of diagrams in this document:

- Component Architecture – identifies a subset of BigFix components without any communication information. Only components and sub-components with non-localhost communication are shown.

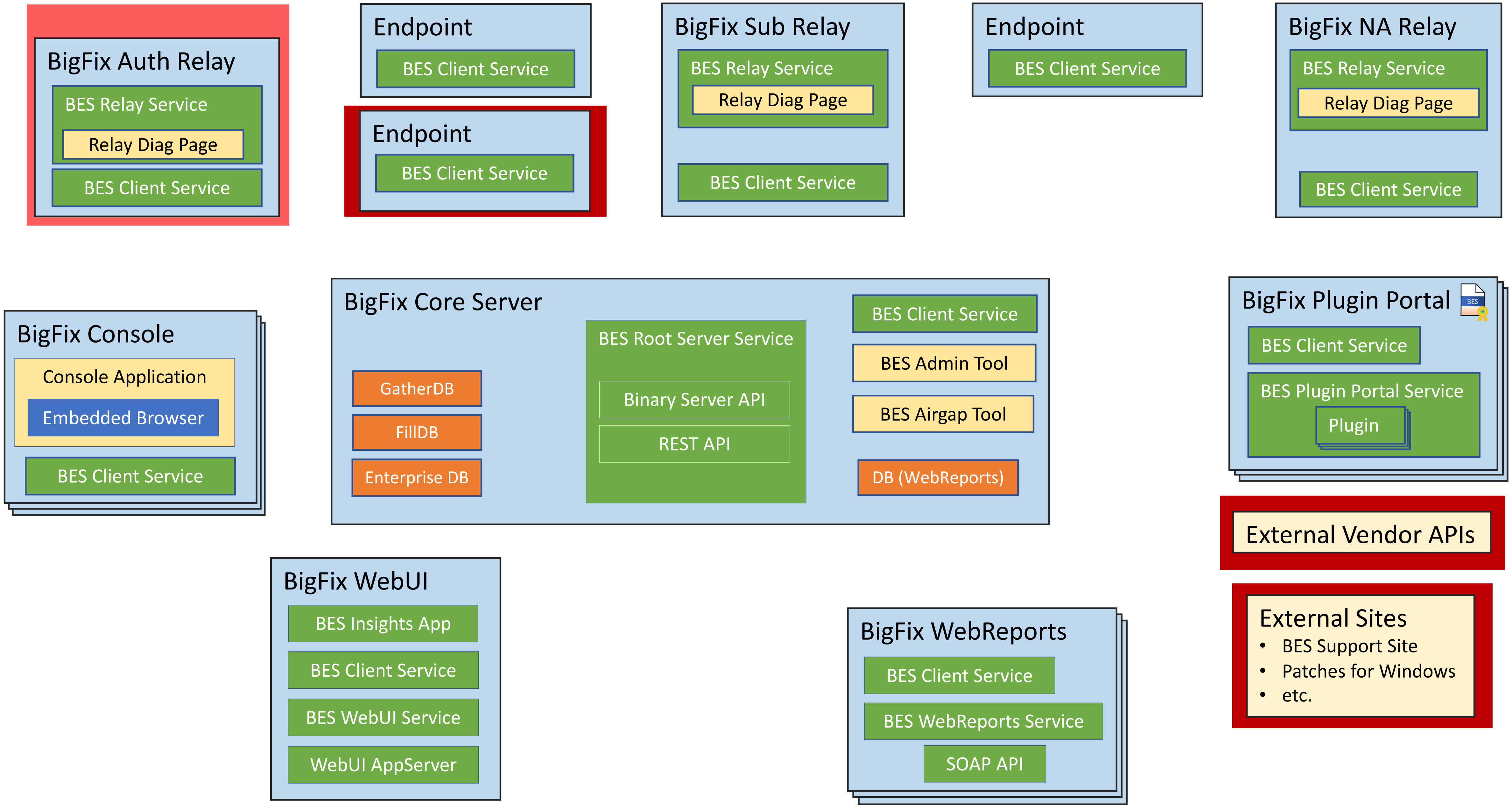
- Communication Architecture (Logical) – documents the communication flow between components. Includes:
  - Communication initiation directions
  - Default ports used
  - Protocols used

- Communication Architecture & Flow (CA&F) – focuses on a specific BigFix configuration and use case. The diagram on these slides is accompanied by the documentation of a “Happy Path” flow.

Important: These diagrams are not a substitute for the diagrams or guidance provided by HCL’s official product documentation. A future version of these diagrams may be distributed as part of HCL’s official product documentation.

Platform

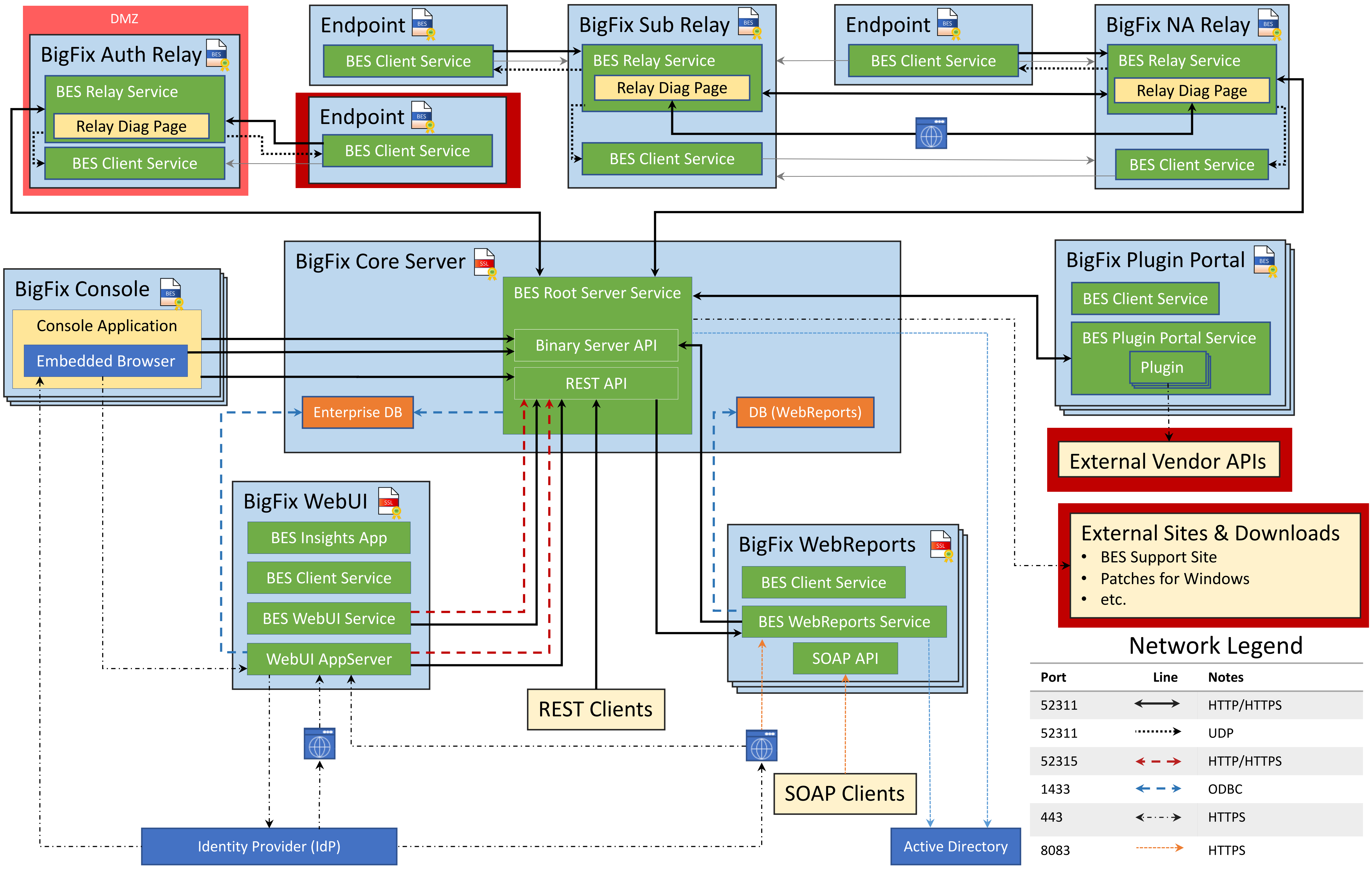
Platform: Component Architecture



Platform: Communication Architecture

Customer supplied SSL Cert

BigFix Generated Trusted SSL Cert



DMZ

BigFix Auth Relay
- BES Relay Service
- Relay Diag Page
- BES Client Service

Endpoint
- BES Client Service

Endpoint
- BES Client Service

BigFix Sub Relay
- BES Relay Service
- Relay Diag Page
- BES Client Service

Endpoint
- BES Client Service

BigFix NA Relay
- BES Relay Service
- Relay Diag Page
- BES Client Service

BigFix Console
- Console Application
- Embedded Browser

BigFix Core Server
- BES Root Server Service
- Binary Server API
- REST API
- Enterprise DB
- DB (WebReports)

BigFix Plugin Portal
- BES Client Service
- BES Plugin Portal Service
- Plugin

External Vendor APIs

External Sites & Downloads
- BES Support Site
- Patches for Windows
- etc.

BigFix WebUI
- BES Insights App
- BES Client Service
- BES WebUI Service
- WebUI AppServer

REST Clients

BigFix WebReports
- BES Client Service
- BES WebReports Service
- SOAP API

SOAP Clients

Identity Provider (IdP)

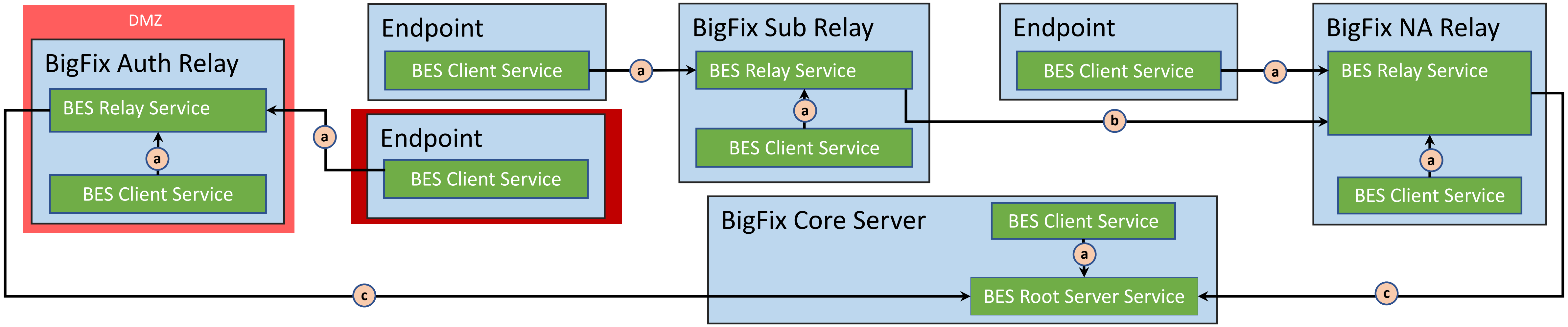
Active Directory

Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
52311	·······→	UDP
52315	← – – →	HTTP/HTTPS
1433	← – – →	ODBC
443	←·–·→	HTTPS
8083	-------→	HTTPS
389/636/3268	-------→	LDAP
N/A	→	ICMP

Version 1.03

CA&F: BES Client Certificate Request



BES Client Certificate Request Flow

This flow is initiated when the Endpoint starts the BES Client Service for the very first time or after a prior Certificate Request failure.

Key Exchange steps:

1. The Client generates a local key for a Certificate Signing Request (CSR) and sends it to the Server via a Relay or path of Relays.
2. The Client/Relay(s)/Server attempt to open and hold a concurrent connection across all required communication paths (ⓐ ⓑ ⓒ) from the Client to the Server.
    - Each component (client, relay and server) will timeout the connection based on the following Relay setting: _HTTPRequestSender_Connect_TimeoutSeconds
    - Each Relay adds its own IP Address to the request forming a chain back to the Client.
    - If the Relay fails to gain a connection to the Root Server Service, due to timeout, the certificate request attempt is failed.
3. The Server, acting as a certificate authority (CA), generates an SSL Certificate using the provided CSR.
4. The Server sends the SSL Certificate back to the requesting Client via the (ⓒ ⓑ ⓐ) path.
5. The Client stores the SSL Certificate.

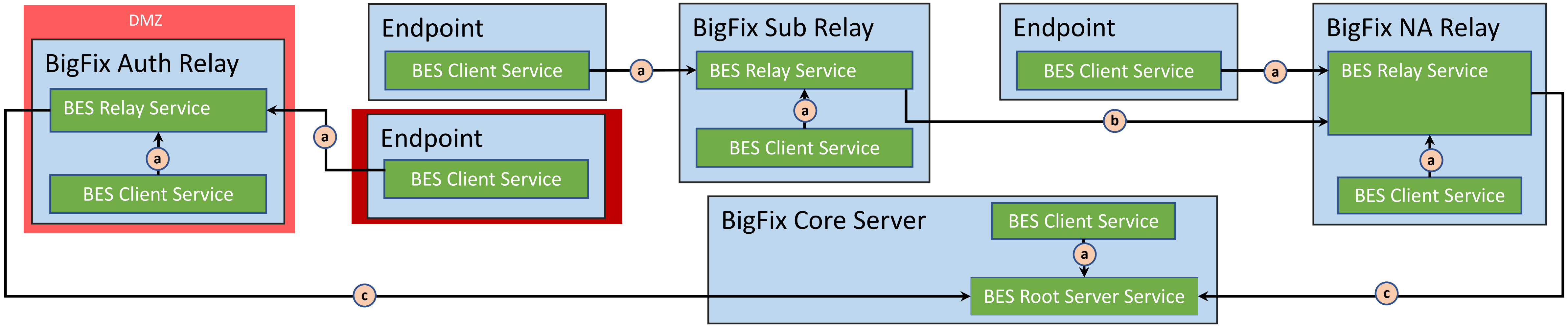
Network Legend

Port	Line	Notes
52311	⟷	HTTPS

CA&F: Platform Registration

Network Legend

Port	Line	Notes
52311	⟷	HTTPS



BES Client Registration Flow

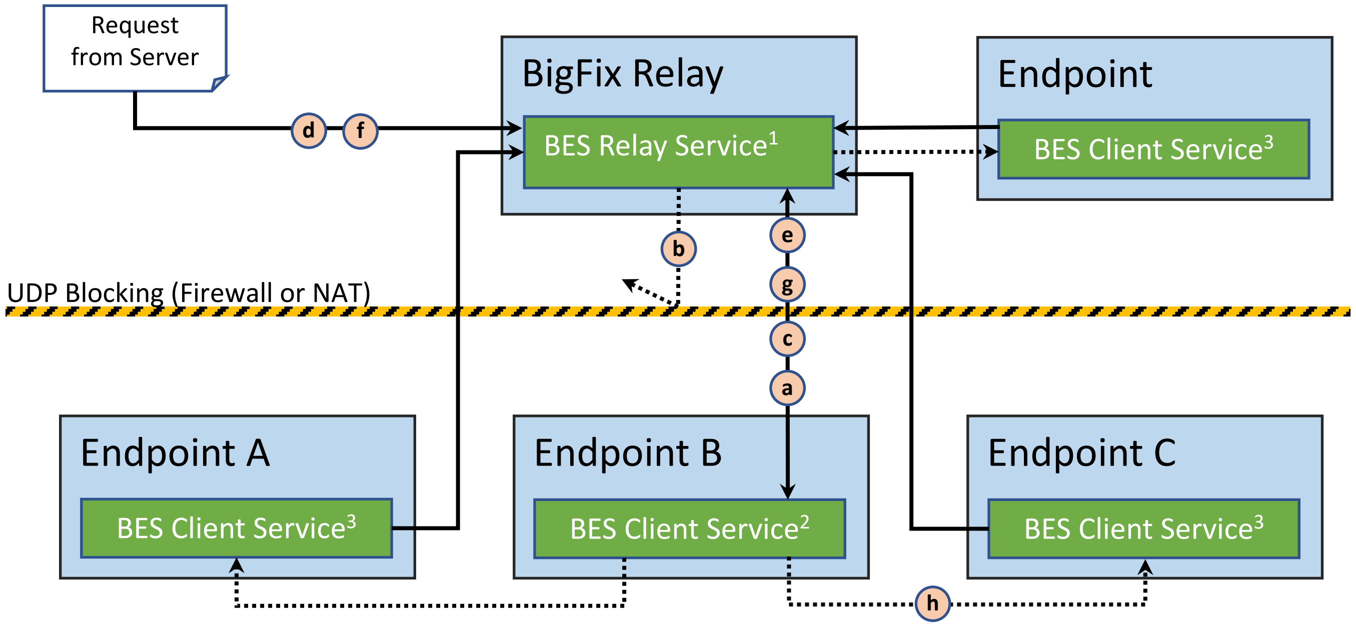
Registrations are initiated:

- When the computer starts the BES Client Service, e.g., post-install, reboot or after the BES Client service is killed.
- When the register interval (_BESClient_Register_IntervalSeconds) is met.
- When a Relay responds to a Client Service request by telling it to re-register due to the Relay clearing its registration list. Controlled by setting: _Enterprise Server_ClientRegister_ClientRegistrationExpirationPeriod.
- On any Registration failure.
- Every time the BES Client Service chooses a new Relay (not a re-selection of the existing one).
  - If a user sets a new Relay
  - If the threshold is met (_BESClient_RelaySelect_IntervalSeconds) to force the BES Client Service to select a potentially new Relay.
  - If the IP address for the computer running the BES Client Service is changed and setting (_BESClient_RelaySelect_AlwaysOnIPListChange) is set to 1.

Registration steps:

1. The Client generates a keypair and calls the registration service with the public key.
2. The Client/Relay(s)/Server attempt to open and hold a concurrent connection across all required communication paths (ⓐ ⓑ ⓒ) from the Client to the Server.
   - Each component (client, relay and server) will timeout the connection based on the following Relay setting: _HTTPRequestSender_Connect_TimeoutSeconds
   - Each Relay adds its own IP Address to the request forming a chain back to the Client.
   - If the Relay fails to gain a connection to the Root Server Service, due to timeout, the registration attempt is failed.
3. The Server generates an ID and generates a certificate with the ID and the public key, signing with the Client CA key/certificate.
4. Server sends the client the ID and Cert back to the requesting Client via the (ⓒ ⓑ ⓐ) path.
5. The Client stores the certificate and ID.

CA&F: Client Persistent Connections



[1] _BESRelay_PersistentConnection_Enabled = 1
[2] _BESClient_PersistentConnection_Enabled = 1
[3] _BESClient_PersistentConnection_Enabled = 0

Persistent Connection Registration Flow

Note: After being enabled, a persistent TCP connection between a client and its parent relay is normally established at the next registration of the client:

a) The Endpoint B Client Service requests a registration from Relay Service and receives a test notification.

- The relay checks whether the client is eligible to open a persistent connection, based on the overall number of persistent connections that the relay is already handling, and their partition by subnet. If the client is ineligible, the registration fails, otherwise a test notification is returned.
- The Client listens for a UDP test notification for up to 60 seconds, if it is received the registrations fails.

b) Relay sends UDP test notification to Client.

c) After 60 seconds without receipt of test notification, the Client establishes a persistent TCP connection with the Relay.

Persistent Connection Communication Flow - Direct

d) Relay receives a UDP notification for Endpoint B

e) Relay sends notification to Client Service on Endpoint B

Persistent Connection Communication Flow - Indirect

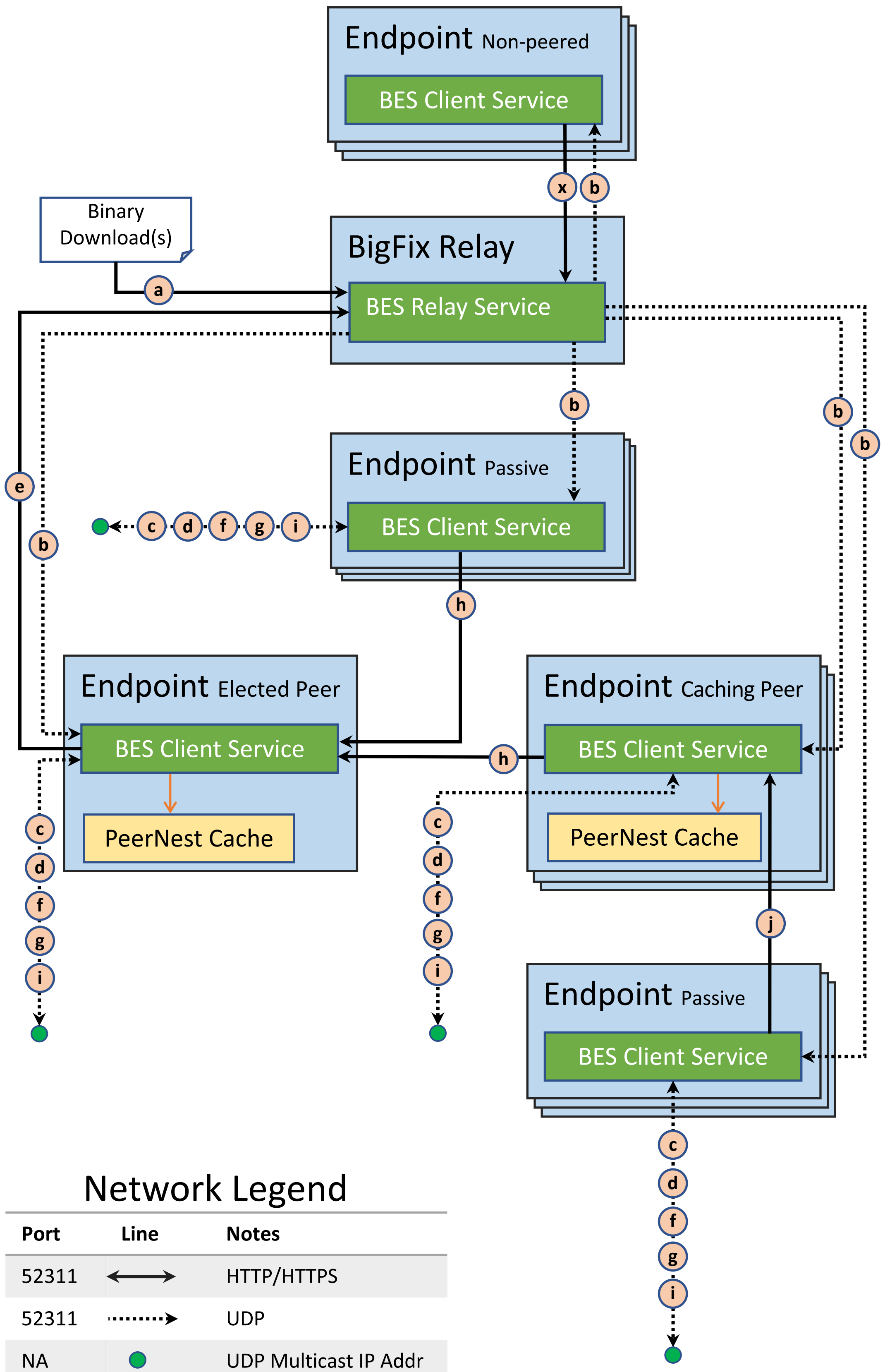
f) Relay receives a UDP notification for Endpoint C

g) Relay sends the notification, including destination target, to the Client Service on Endpoint B.

- Client Service looks up the target address

h) The Client Service sends the notification to the target via UDP.

CA&F: PeerNest (IPV4 & IPV6)



Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
52311	⋯→	UDP
NA	●	UDP Multicast IP Addr

PeerNest Flow

Important: PeerNest behavior and the use of multicast addresses is based on the definition of multicast standard itself, the communications are expected to stay in the local subnet. BigFix does not perform any enforcement to ensure the communication stay in the local subnets. References to the standards documentation:

- https://www.rfc-editor.org/rfc/rfc1112.html
- https://www.rfc-editor.org/rfc/rfc5771.html
- https://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml

a) Relay receives binary download(s) for one or more Endpoints.
- The relay stores the binary file(s).

b) Relay sends a binary download notification to targeted Endpoints.
- All non-peered Endpoints continue by downloading the file(s) directly from the Relay using default behavior x.

c) Each peered Endpoint, notified of a download, asks its peers if they have the file(s) already downloaded.
- If the file(s) is/are available from a peer, the flow continues at h.

d) Each peered Endpoint, notified of a download, elect a Peer to download the file(s) from the Relay.

e) The elected-peer Endpoint begins to download the file(s) from the Relay.

f) The elected-peer Endpoint periodically notifies its peers of download progress.
- When the elected-peer Endpoint finishes a download, it stores the file in its PeerNest Cache.

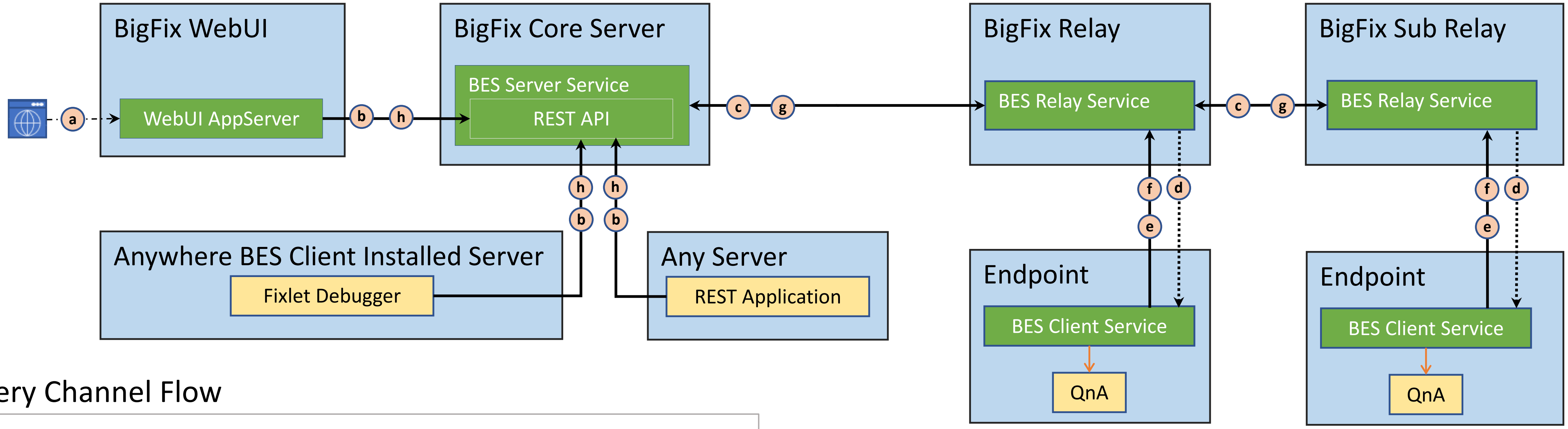
g) The elected-peer Endpoint notifies its peers when download(s) are complete.

h) Each peer notified and interested in the file(s) initiates file download(s) from the elected-peer.
- A Caching Peer will store the downloaded file(s) in its PeerNest Cache.
- A Passive Peer will download the file(s) from the elected-peer or a Caching Peer.

i) Each Caching Peer will notify its peers that it is severing the file(s) for download.
- Each peer that has not downloaded the file(s) can choose between the elected-peer and all Caching Peers as a source for file(s) download.

j) The notified peer downloads the file(s) from either the elected-peer or any Caching Peer that sent a notification.

CA&F: WebUI Query Channel



Query Channel Flow

a) User creates a query in browser via WebUI, submits it, and waits to view results.

b) WebUI AppServer sends the Query to Core Server REST API.

- (POST /api/clientquery) [Fixlet Debugger or any other REST API user can also call]
- Receives the query id in response.

c) Root server service sends the query through the relay chains to the parent relays of the endpoints

- Each Relay queues the query in its memory.

d) Relay sends a Query Notification (QN) to the BES Client Service on its Endpoints.

e) BES Client Service requests and receives the query from the Relay.

- Query is immediately evaluated, one of two ways: run in the client context or the query is passed to QnA for processing.
- Either way, the Query Result (QR) consisting of status and results back are realized.
- The QR is encrypted and signed.

f) BES Client Service sends the QR to the Relay.

- The Relay queues each received QR in a memory queue, not the Report queue.

g) Relay sends queued QRs to Server (directly or through relay chain).

- Root Server Service upon receipt of any QR stores it in memory

h) WebUI AppServer periodically requests results from Core Server.

- (POST /api/clientqueryresults/{id}...
- Returned results are displayed in displayed in browser

Notes:

1. The ⓒ communication paths may be HTTP if the **minimumsupportedrelay** setting, held in the Masthead and set via BESAdmin Tool, is set to **0.0.0**. A setting of **9.5.6** enforces HTTPS.
2. The same Query Channel communication can be initiated ⓑ and completed ⓗ by both the Fixlet Debugger Application and a REST API call initiated by a user or an application.
3. A Relay in the DMZ typically has UDP communication ⓓ blocked and would require the configuration of a persistent connection of at least one endpoint per subnet serviced by the Relay.
4. The query evaluation method used in the aftermath of the ⓔ communication can give different results and affect performance of the agent in the process.

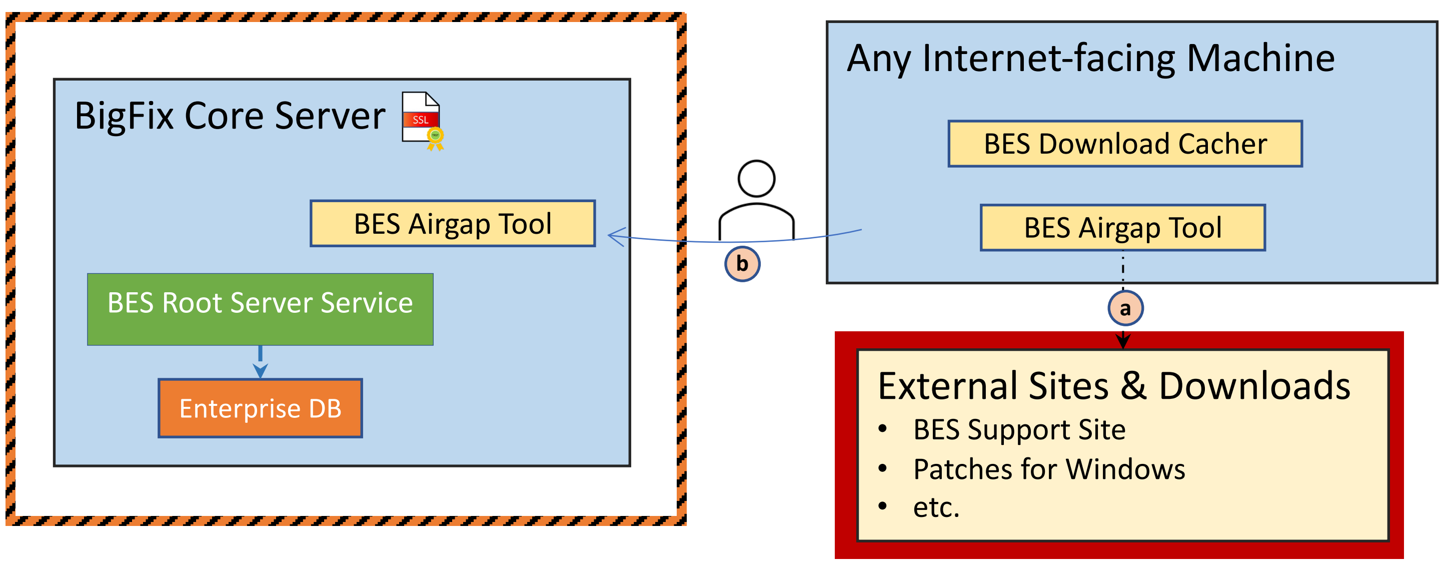
Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
52311	·········>	UDP
443	<-·-·->	HTTPS

CA&F: Airgap Tool

Customer supplied SSL Cert

BigFix Generated Trusted SSL Cert



Airgap Tool Flow

a) User downloads content from the internet to any internet-facing machine outside the air-gapped environment using the BES Airgap Tool.
   - The user uses the Download Cacher to place the downloaded content into a package on transferrable media.

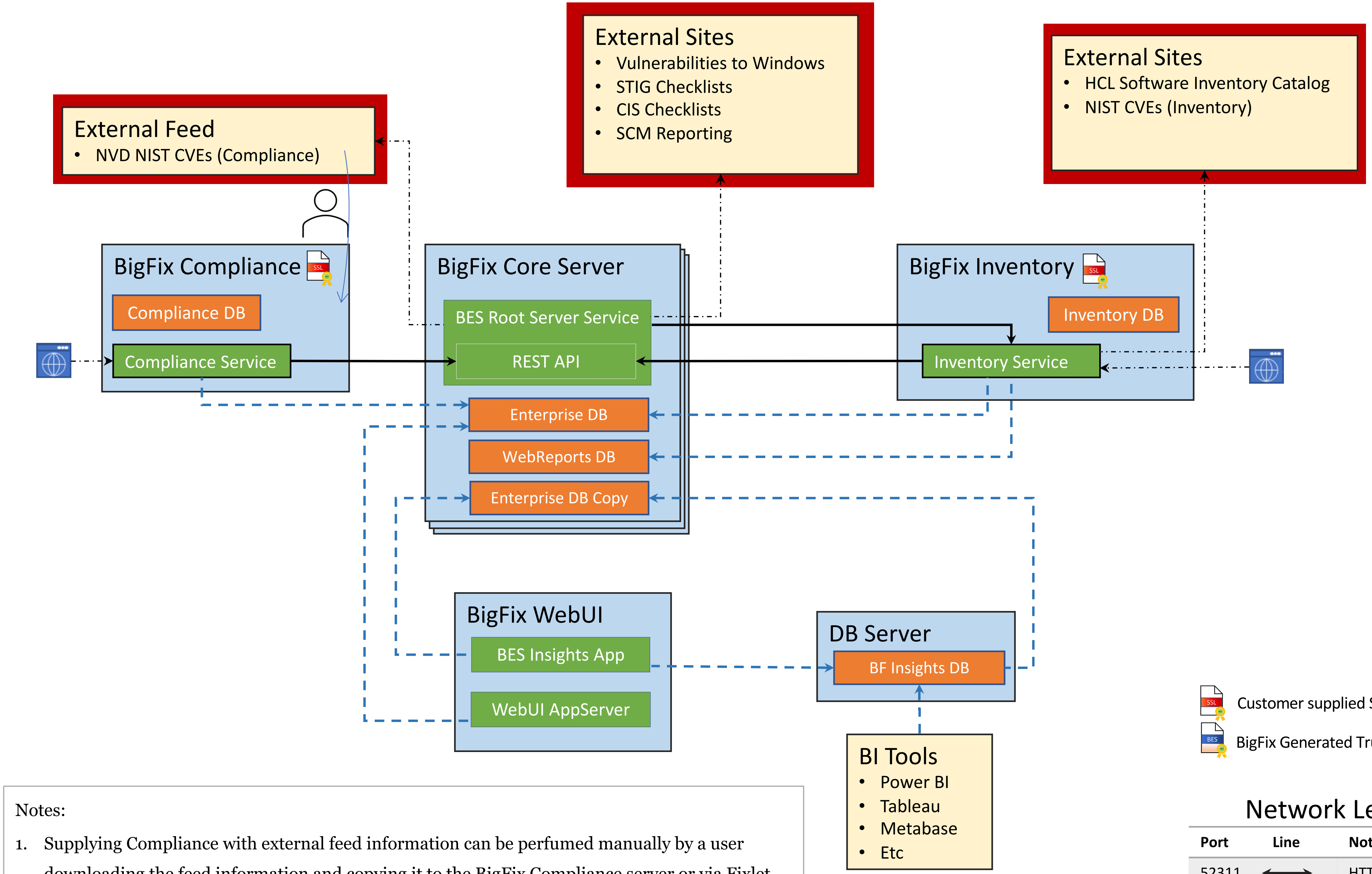
b) User takes media to air-grapped environment.
   - User runs the Airgap Tool in the air-gapped environment which places the site files/downloads in the filesystem in a place where the BigFix Server services will grab them and mirror them in the filesystem, then place them in the database (Root Server service performs this work as the 'gatherdb' component)

Network Legend

Port	Line	Notes
443	← – →	ODBC/JDBC
NA	⤻	Human file copy

Modules

Platform: Communication with Modules



Notes:

1. Supplying Compliance with external feed information can be perfumed manually by a user downloading the feed information and copying it to the BigFix Compliance server or via Fixlet through the Root Server Service.
2. The Enterprise DB Copy is created manually by an administrator, it can exist on any server.

SSL Customer supplied SSL Cert

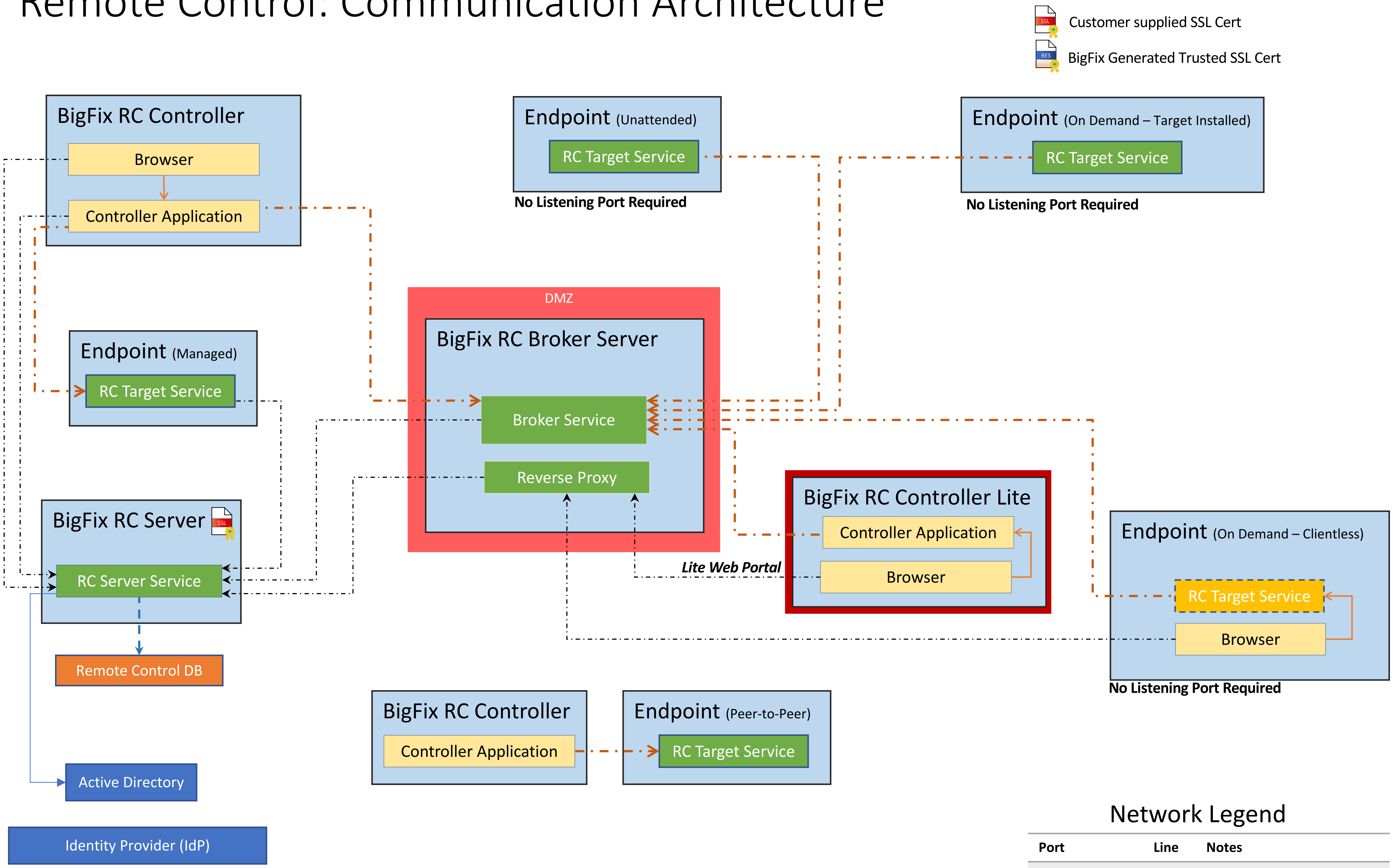
BES BigFix Generated Trusted SSL Cert

Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
1433	← - - →	ODBC
443	←-·-·→	HTTPS
NA	⤻	Human file copy

Remote Control

Remote Control: Communication Architecture

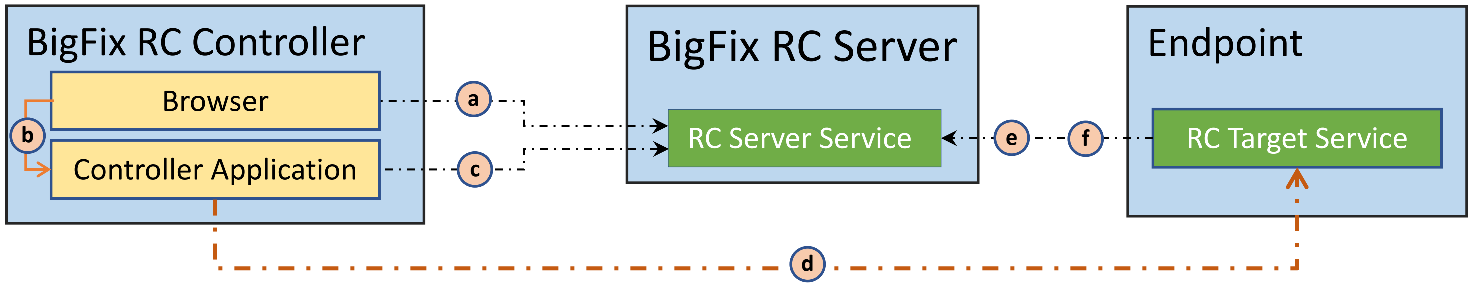


Network Legend

Port	Line	Notes
888	<- · ->	TLS 1.2 over TCP
443	<·-·->	HTTPS
DB Port	<- - ->	JDBC
389/636/3268	——>	LDAP

CA&F: Managed Session Remote Control

Managed Session: Communication Architecture & Flow



Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
888	←·→	TLS 1.2 over TCP – proprietary protocol

Notes:

1. The d path is protected via TLS 1.2 over TCP but using a proprietary protocol. All ports in BigFix Remote Control can be chosen by the user, defaults are shown.
2. Managed sessions require the Controller to have network visibility to the Endpoint.
3. The recordings that might be captured by the target in a remotely-controlled session can also be saved on the controller or on the Endpoint; choices are made in policy settings.
4. Depending on how the RC Session Policies are configured, there may be a risk of exposing BII/PII and allowing unauthorized application use on the Endpoint when using Managed Sessions.

Managed Session Remote Control Flow

User opens browser on the RC Controller and navigates to the RC Server URL.

a) The user logs into the RC Server , selects an endpoint to remote to and the Server responds with a TRCJWS file.

b) RC Controller Browser launches the Controller Application and passes it the TRCJWS file.

c) The Controller Application requests and receives authorization to access the Endpoint from the Server.

d) The Controller Application initiates a remotes session request the Endpoint's RC Target Service.

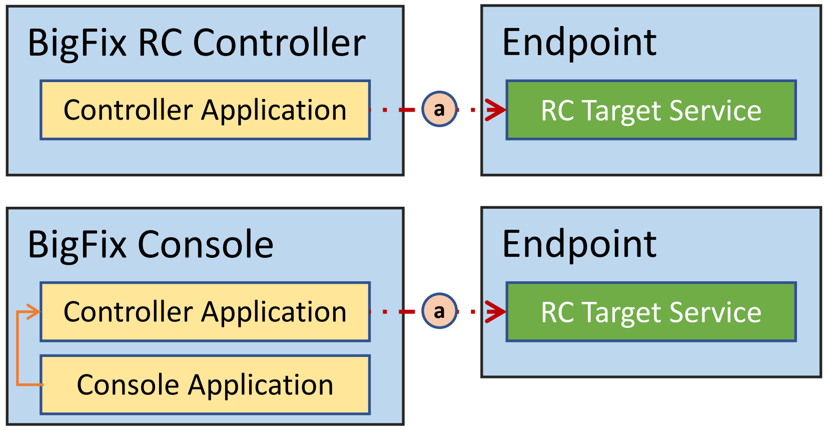
e) The Target Service requests and receives authorization to allow the remote-controlled session from the Server .

- The Target Service starts the remotely controlled session with the Controller.

f) During the session it sends auditing events to the server. At the end of the session, if recording has been set, the recordings will be sent to the server for storage.

CA&F: Peer-to-Peer Session Remote Control

Peer-to-Peer Session : Communication Architecture & Flow



Peer-to-Peer Remote Control Flow

[Top Diagram]User opens the Controller Application on the RC Controller and inputs the address and port of the Endpoint.

[Bottom Diagram] User right clicks on an Endpoint in the Computer Panel of the Console and chooses remote connection. Requires a Controller installed on the Console machine and the integration configured.

a) Controller requests a remote-control session with the identified Endpoint from the Endpoint's Target Service.
   - Target Service checks the policies in its registry to determine if the session can proceed
   - Target Service and starts the session
   - If configured, audits will be saved on target.
   - If enabled, recordings will be saved on the controller.

Network Legend

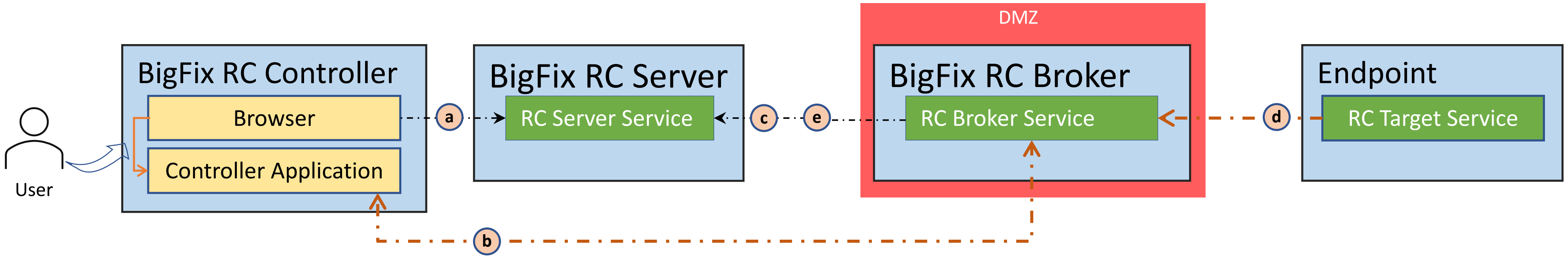
Port	Line	Notes
443	<-·-·->	HTTPS
888	<- · ->	TLS 1.2 over TCP, proprietary protocol

Notes:

1. The ⓐ path is protected via TLS 1.2 but uses a proprietary protocol.
2. All ports in BigFix Remote Control can be chosen by the user, defaults are shown.
3. Peer-to-peer may require holes to be opened in the firewall to reach some endpoints.
4. Proxies can be used to allow the Controller to reach an Endpoint.
5. Depending on how the RC Target is configured, there may be a risk of exposing BII/PII and allowing unauthorized application use on the Endpoint when using Peer-to-peer Sessions.

CA&F: Unattended Session (target installed) Remote Control

Unattended (Managed over Internet) Target Service Installed: Communication Architecture & Flow



On Demand Remote Control Flow w/target service installed

Note: When an Endpoint has been configured for Unattended Remote Control, it will poll the RC Broker every 2 minutes (configurable) to see if a Remote Control Request is pending for the Endpoint.

Note: Depending on how the RC Session Policies are configured, there may be a risk of exposing BII/PII and allowing unauthorized application use on the Endpoint when using Unattended Sessions.

User opens the browser on the Controller and navigates to the Server:

a) Via the Browser user logs into the RC Server , selects Unattended session and the Server responds with a TRCJWS file.
- The Browser launches the controller and passes the TRCJWS file to it.
- The Controller displays a list of remote-control targets that are configured to operate in Unattended Mode.
- The User selects one of these targets.

b) Controller requests authorization to be contacted by Endpoint via the RC Broker Service.

c) The Broker Service requests and receives authorization to allow the Controller Remotely Control the Endpoint
- The Broker Service queues a connection request for the Endpoint.

d) The Endpoint's Target Service, during it periodic polling of the Broker, receives the connection request including session configuration.
- If the session configuration requires user acceptance of the session, the user screen will ask for permission to continue, otherwise the session will start and communication between Controller and Endpoint will continue through the Broker.

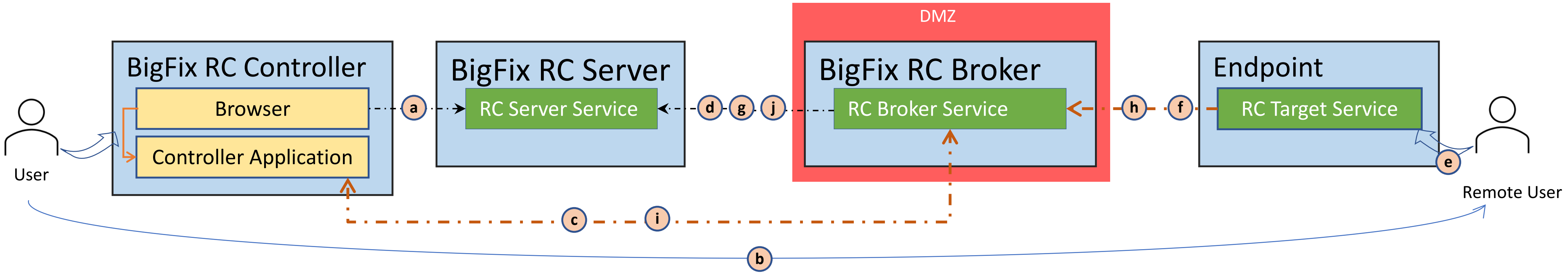
e) Broker will cache the auditing and recordings and send to the server on session termination.

Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
888	←·→	TLS 1.2 – proprietary protocol

CA&F: On Demand Session (target installed) Remote Control

On Demand with Target Service Installed: Communication Architecture & Flow



On Demand Remote Control Flow w/target service installed

User opens the browser on the Controller and navigates to the Server:

a) Via the Browser user logs into the RC Server , selects a Broker-based session and the Server responds with a TRCJWS file.

- The Browser launches the controller and passes the TRCJWS file to it.
- The Controller displays a remote target connection code.

b) User communicates the remote target connection code the Remote User.

c) Controller requests authorization to be contacted by Endpoint via the RC Broker Service.

d) The Broker Service requests and receives authorization to allow the Controller Remotely Control the Endpoint

- The Broker Service listens for a connection acceptance from the Endpoint.

e) The Remote User enters the code into the Endpoint's Target Service.

f) Target Service requests and receives validation of the session code and the policies to be used from the Broker.

g) Broker Server requests and receives validation of the session code from the Server. It also receives the policies to be used for the session.

h) The Target Service sends a connection acceptance to the Broker Service.

i) The Broker completes the connection with the Controller Application and remote session is started. It will communicate via the broker and reverse proxy over the ⓕ and ⓘ paths.

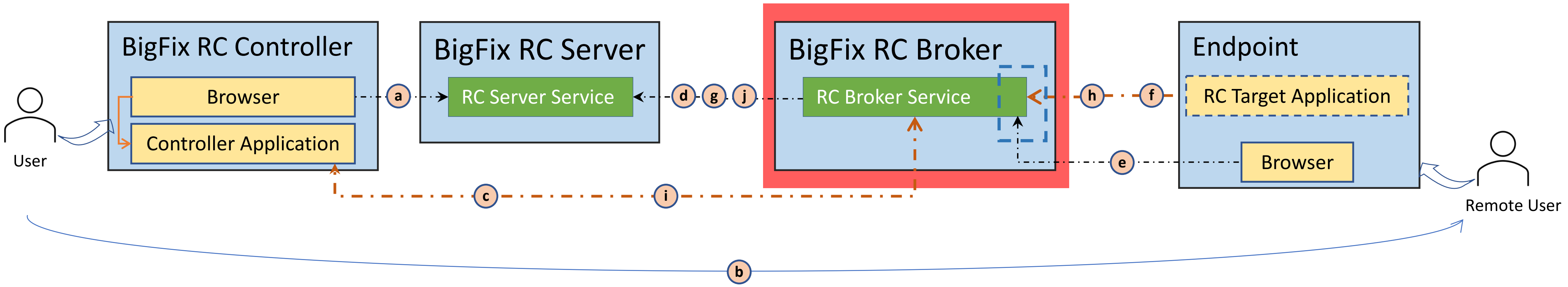
j) Broker will cache the auditing and recordings and send to the server on session termination.

Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
888	⇠·⇢	TLS 1.2 over TCP – proprietary protocol
N/A	↗	Human to human communication

CA&F: On Demand Session (clientless)Remote Control

On Demand with no Target Service Installed (Clientless) : Communication Architecture & Flow



On Demand Remote Control Flow w/o target service pre-installed

User opens the browser on the Controller machine and navigates to the Server:

a) Via the Browser user logs into the RC Server , selects a Broker-based session and the Server responds with a TRCJWS file.

- The Browser launches the controller and passes the TRCJWS file to it.
- The Controller displays a remote target connection URL.

b) User communicates the URL to the Remote User.

c) Controller requests authorization to be contacted by Endpoint via the RC Broker Service.

d) The Broker Service requests and receives authorization to allow the Controller Remotely Control the Endpoint

- The Broker Service listens for a connection acceptance from the Endpoint.

e) The Remote User navigates to the provided URL in a browser on the Endpoint and receives a RC Target Executable from the RC Broker (through a reverse proxy – either the default provided by BigFix or a customer provided reverse proxy that could exist on another server).

- The user installs the downloaded RC Target Application and executes it.

f) Target Executable requests and receives validation of the session code (from the URL) and the policies to be used from the RC Broker (through the reverse proxy).

g) Broker Server requests and receives validation of the session code from the Server. It also receives the policies to be used for the session.

h) The Target Application sends a connection acceptance to the Broker Service (through the reverse proxy).

i) The Broker completes the connection with the Controller Application and remote session is started. It will communicate via the broker and reverse proxy over the ⓕ and ⓘ paths.

j) Broker will cache the auditing and recordings and send to the Server on session termination.

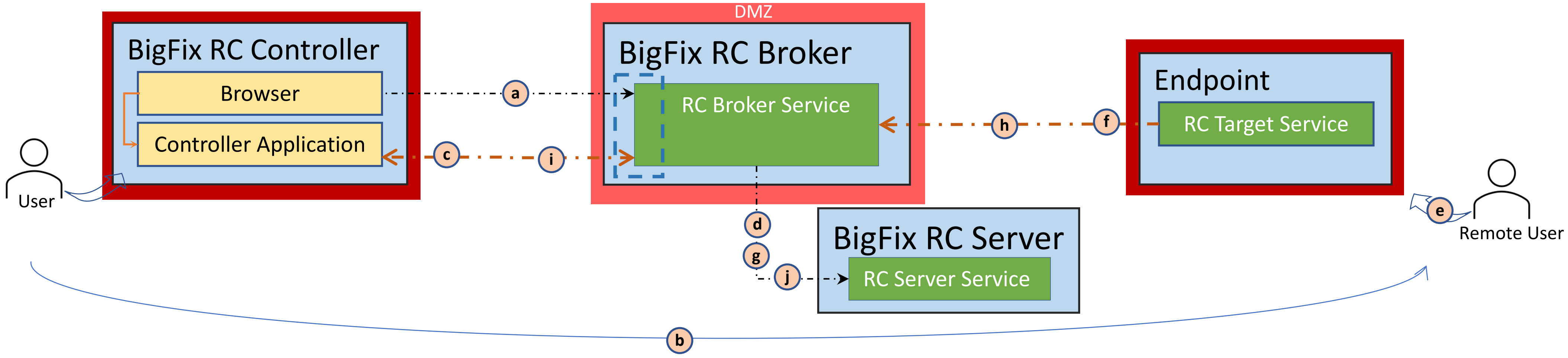
- The RC Target Application is removed from the Endpoint.

Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
888	⇠·⇢	TLS 1.2 over TCP, proprietary protocol
N/A	⤳	Human to human communication
N/A	⬚	Reverse Proxy

CA&F: On Demand & Unattended Session (Lite Web Portal) Remote Control

On Demand Lite Web Portal: Communication Architecture & Flow



On Demand Session (from/to Internet) Flow

User opens the browser on the Controller and navigates to the Server (through a reverse proxy provided by BigFix or the organization):

a) Via the Browser user logs into the RC Server , selects a Broker-based session and the Server responds with a TRCJWS file.

- The Browser launches the controller and passes the TRCJWS file to it.
- The Controller displays a remote target connection code.

b) User communicates the remote target connection code the Remote User.

c) Controller requests authorization to be contacted by Endpoint via the RC Broker Service.

d) The Broker Service requests and receives authorization to allow the Controller Remotely Control the Endpoint

- The Broker Service listens for a connection acceptance from the Endpoint.

e) The Remote User enters the code into the Endpoint's Target Service.

f) Target Service requests and receives validation of the session code and the policies to be used from the Broker.

g) Broker Server requests and receives validation of the session code from the Server. It also receives the policies to be used for the session.

h) The Target Service sends a connection acceptance to the Broker Service.

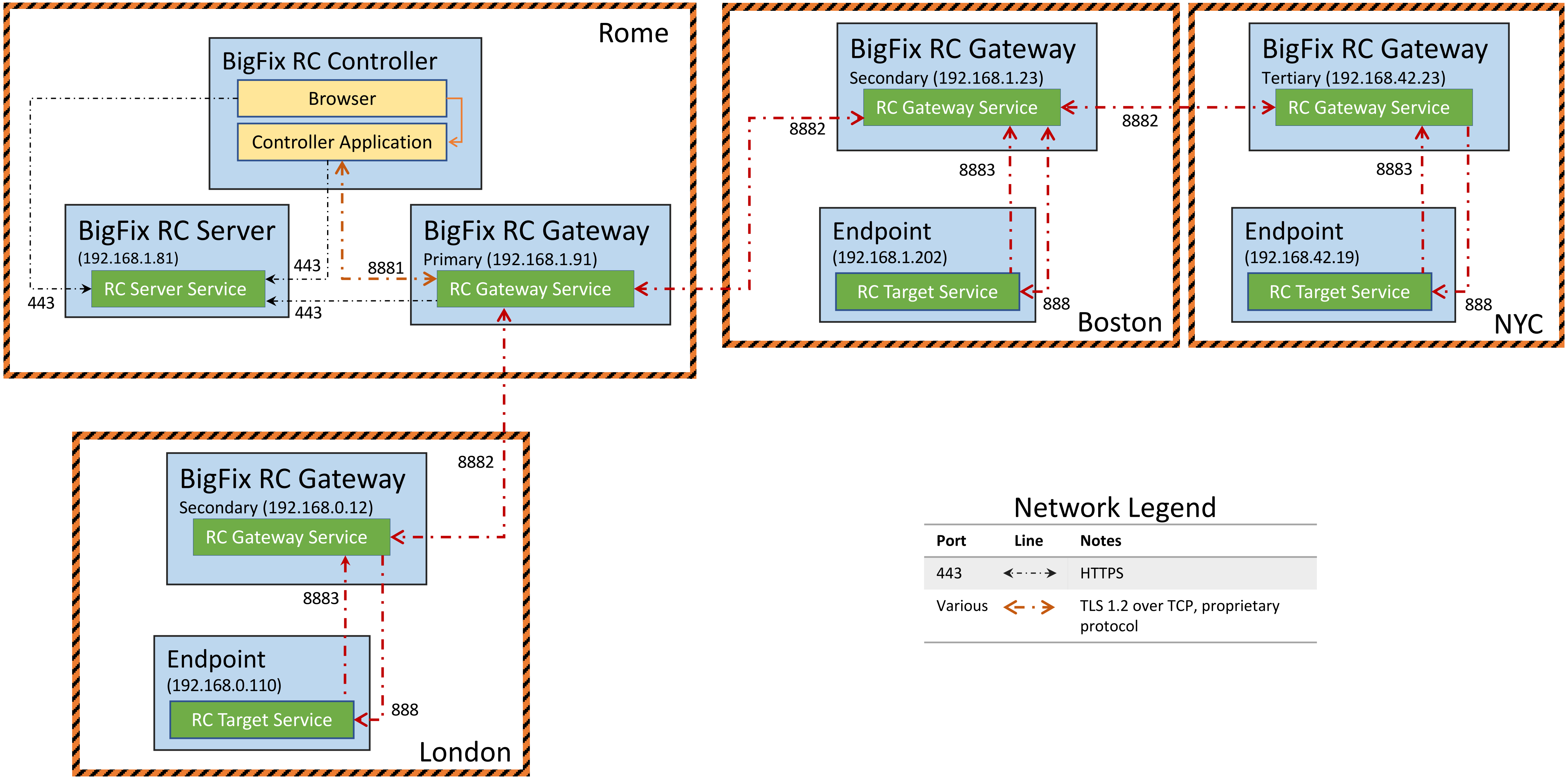
i) The Broker completes the connection with the Controller Application and remote session is started. It will communicate via the broker and reverse proxy over the h and i paths.

j) Broker will cache the auditing and recordings and send to the server on session termination.

Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
8887	← · →	TLS 1.2 – proprietary protocol
N/A	⤻	Human to human communication
N/A	⬚	Reverse Proxy

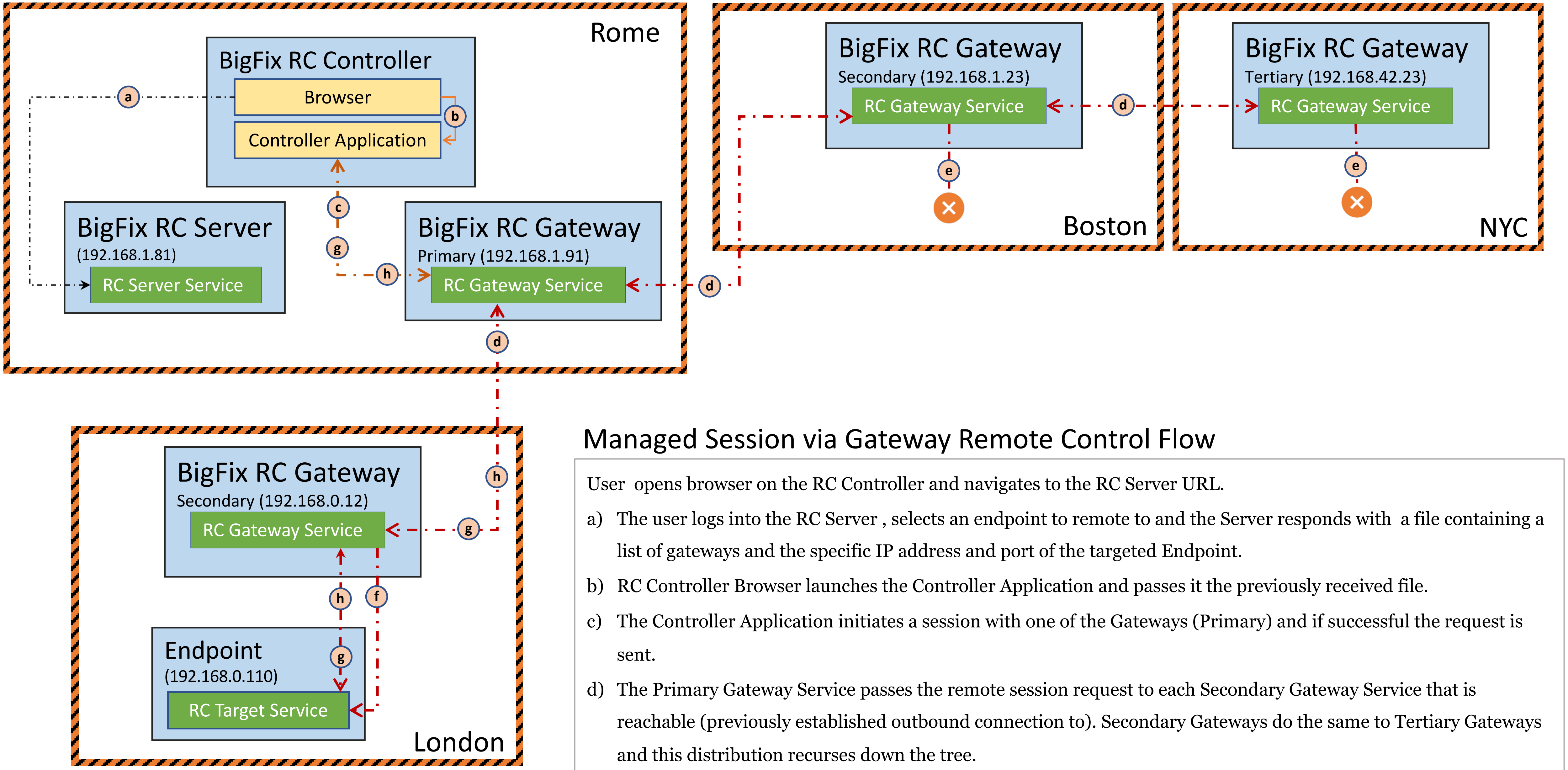
Remote Control: Gateway Communication Architecture



Network Legend

Port	Line	Notes
443	<-·-·->	HTTPS
Various	<-·->	TLS 1.2 over TCP, proprietary protocol

CA&F Managed Session via Remote Control Gateway



Managed Session via Gateway Remote Control Flow

User opens browser on the RC Controller and navigates to the RC Server URL.

a) The user logs into the RC Server , selects an endpoint to remote to and the Server responds with a file containing a list of gateways and the specific IP address and port of the targeted Endpoint.

b) RC Controller Browser launches the Controller Application and passes it the previously received file.

c) The Controller Application initiates a session with one of the Gateways (Primary) and if successful the request is sent.

d) The Primary Gateway Service passes the remote session request to each Secondary Gateway Service that is reachable (previously established outbound connection to). Secondary Gateways do the same to Tertiary Gateways and this distribution recurses down the tree.

e) If the subnet containing the Target is not reachable, no further action is taken down that path.

f) The Target Service receives the requests from a Gateway and confirms it is the correct Target.

g) The Gateway path from the Target to the Controller is used to inform that the Target has been reached.

h) The remote control session is started over the established path through one or more Gateways. Target Service starts the remotely controlled session with the Controller.

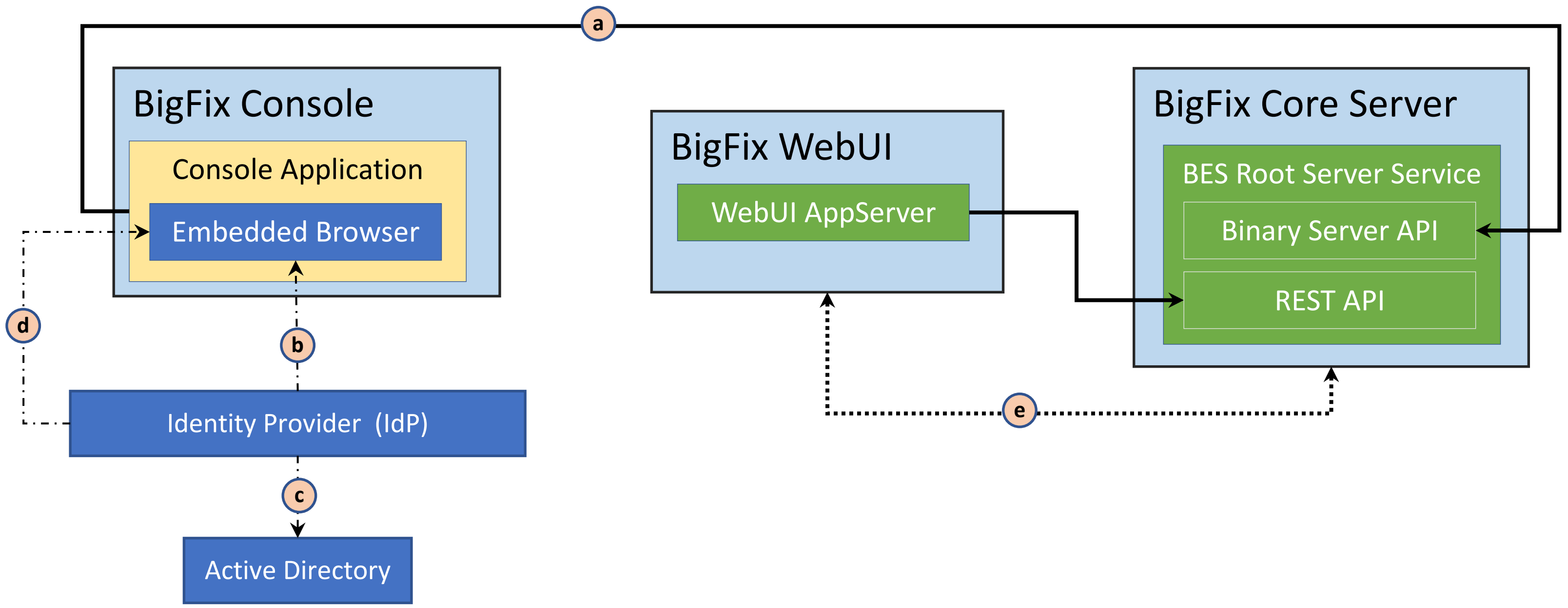
- During the session it sends auditing events to the server. At the end of the session, if recording has been set, the recordings will be sent to the server for storage via the Gateways.

Network Legend

Port	Line	Notes
443	←·-·→	HTTPS
Various	←·→	TLS 1.2 – proprietary protocol

SAML Authentication

CA&F: SAML Authentication from Console



BigFix Console

Console Application

Embedded Browser

BigFix WebUI

WebUI AppServer

BigFix Core Server

BES Root Server Service

Binary Server API

REST API

Identity Provider (IdP)

Active Directory

Network Legend

Port	Line	Notes
52311	⟵⟶	HTTP/HTTPS
443	←-·-·→	HTTPS
5000	←······→	HTTPS

SAML Authentication Flow

User opens Console Application:

a) The Console Application requests and receives the location of the WebUI AppServer.

- The Embedded Browser redirects to IdP.

b) IdP requests authentication credentials from the user via the Embedded Browser and receives the entered credentials.

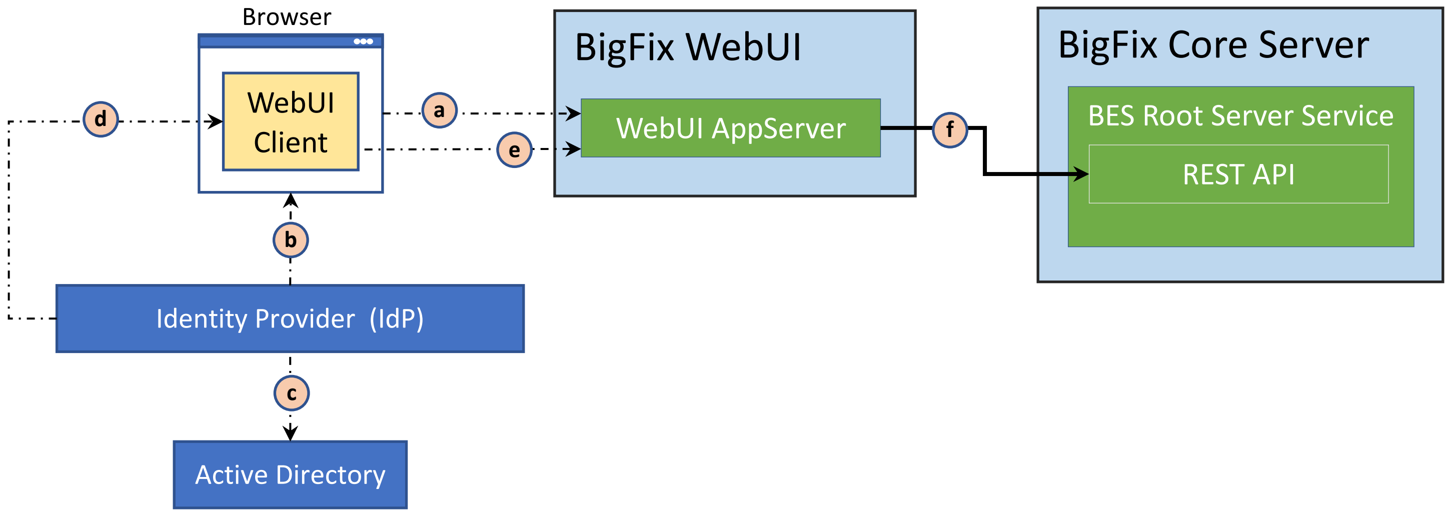
c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

- IdP determines if the user has authenticated successfully

d) IdP sends the SAML Assertion to the Embedded Browser.

e) The Root Server Service sends the SAML Assertion and receives User Authorization (Groups) information from the WebUI AppServer.

CA&F: SAML Authentication from WebUI



Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
443	←·-·→	HTTPS

SAML Authentication Flow

a) User navigates to the WebUI URL and the WebUI Client is downloaded and started in the browser.

- WebUI Client redirects the browser to IdP, e.g., Active Directory Federation Service (ADFS).

b) IdP requests authentication credentials from the user via the browser and receives the entered credentials.

c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

- IdP determines if the user has authenticated successfully

d) IdP sends the SAML Assertion to the WebUI Client.

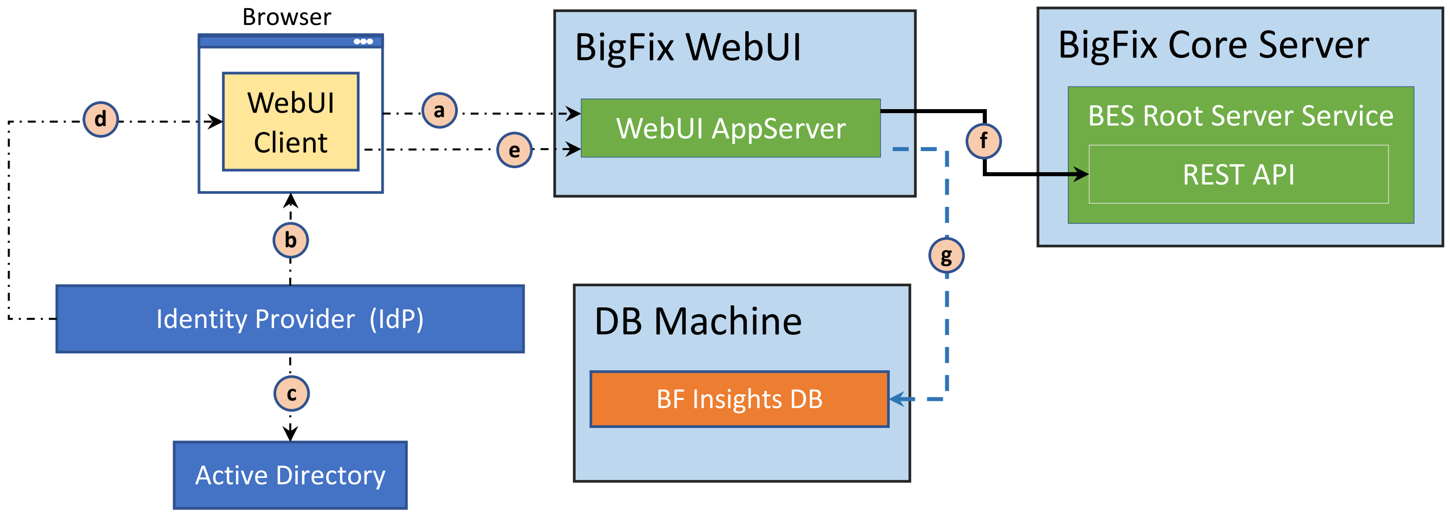
e) WebUI Client sends SAML Assertion to WebUI AppServer.

- This connection (secure and authenticated) is used for the balance of communication between WebUI Client and Server.

f) WebUI Client requests and receives User Authorization data from the Core Server REST API.

- Root Server Service retrieves User Authorization (Groups) information from the Core Server's local Active Directory.

CA&F: SAML Authentication for Insights



Network Legend

Port	Line	Notes
52311	←——→	HTTP/HTTPS
443	←·-·-→	HTTPS
1433	← – – →	ODBC

SAML Authentication Flow

a) User navigates to the WebUI URL and the WebUI Client is downloaded and started in the browser.

- WebUI Client redirects the browser to IdP, e.g., Active Directory Federation Service (ADFS).

b) IdP requests authentication credentials from the user via the browser and receives the entered credentials.

c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

- IdP determines if the user has authenticated successfully

d) IdP sends the SAML Assertion to the WebUI Client.

e) WebUI Client sends SAML Assertion to WebUI AppServer.

- This connection (secure and authenticated) is used for the balance of communication between WebUI Client and Server.

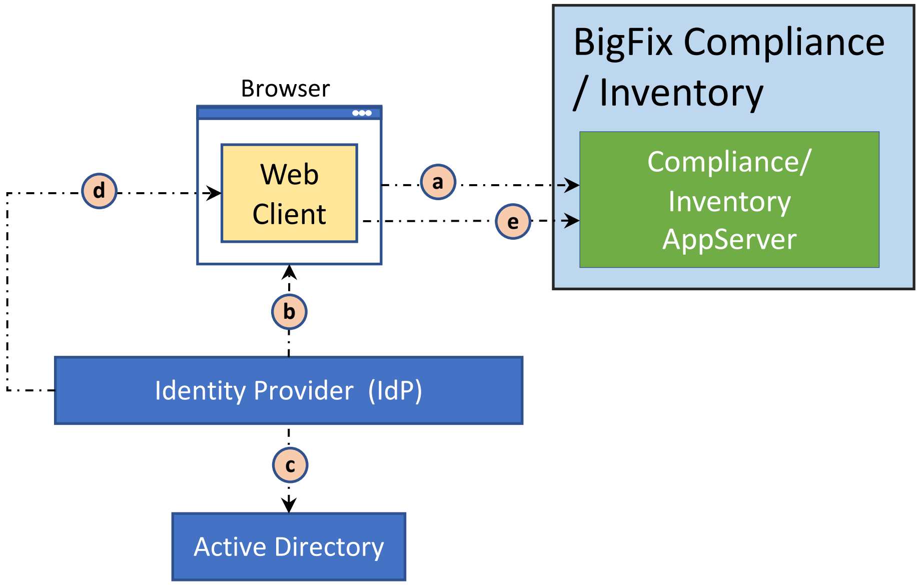
f) WebUI Client requests and receives User Authorization data from the Core Server REST API.

- Is the user a member of the AD Group (Master Operator)
- Root Server Service retrieves User Authorization (Groups) information from the Core Server's local Active Directory.

g) WebUI AppServer requests and receives confirmation that the SAML authenticated user is authorized to run Insights

- Uses SQL Authentication to verify authority

CA&F: SAML Authentication from Compliance/Inventory



Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
443	⇠⇢	HTTPS

SAML Authentication Flow

a) User navigates to the Compliance URL and the Compliance / Inventory Web Client is downloaded and started in the browser.

- Compliance / Inventory Web Client redirects the browser to IdP, e.g., Active Directory Federation Service (ADFS).

b) IdP requests authentication credentials from the user via the browser and receives the entered credentials.

c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

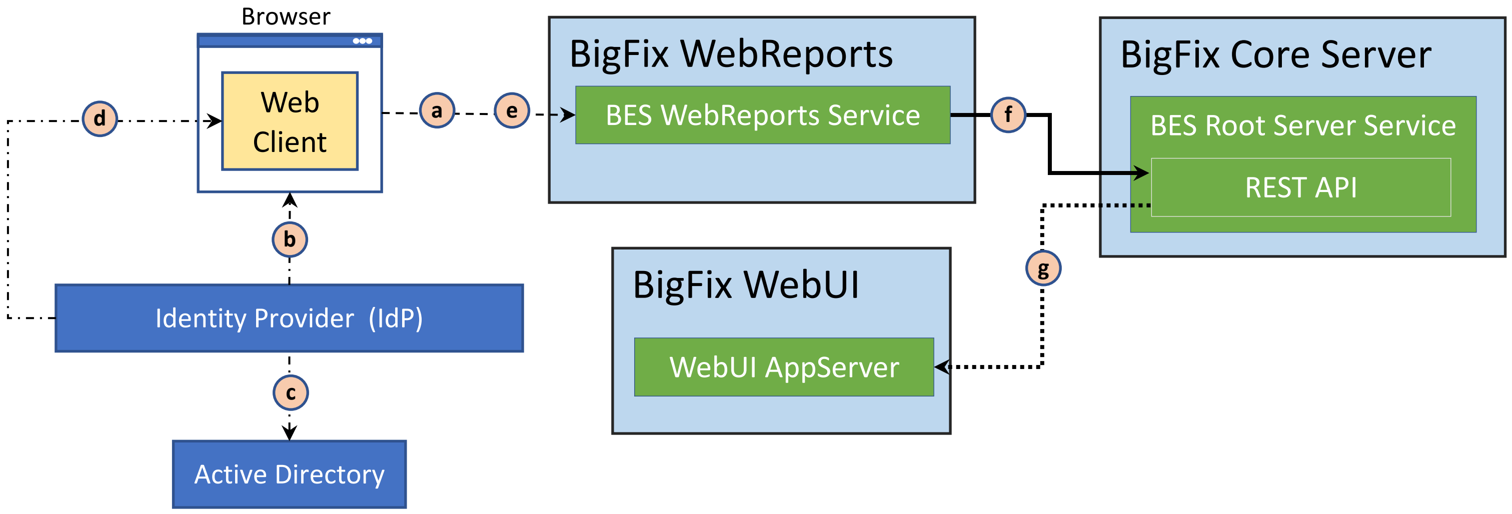
- IdP determines if the user has authenticated successfully

d) IdP sends the SAML Assertion to the Compliance / Inventory Web Client.

e) Compliance / Inventory Web Client sends SAML Assertion to Compliance / Inventory AppServer

- This connection (secure and authenticated) is used for the balance of communication between Compliance / Inventory Web Client and Compliance / Inventory AppServer.

CA&F: SAML Authentication from WebReports



Network Legend

Port	Line	Notes
52311	←——→	HTTP/HTTPS
443	←-·-·→	HTTPS
8083	← – →	HTTPS
5000	←······→	HTTPS

SAML Authentication Flow

User navigates browser to WebReports:

a) The Browser requests and receives the WebReports Web Client.

- The Web Client redirects the Browser to IdP.

b) IdP requests authentication credentials from the user via the Browser and receives the entered credentials.

c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

- IdP determines if the user has authenticated successfully

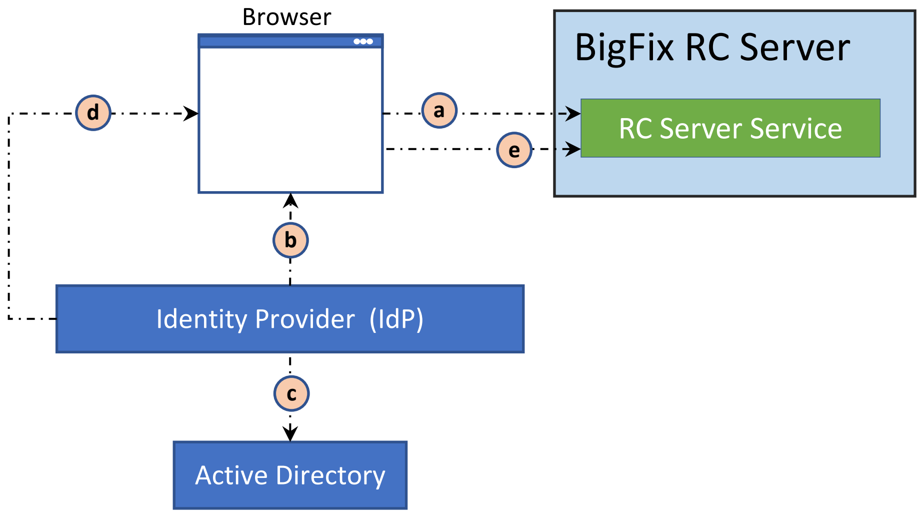
d) IdP sends the SAML Assertion to the Web Client.

e) The Web Client sends SAML Assertion to WebReports Service.

f) The WebReports Service sends the SAML Assertion and receives User Authorization data from the Core Server REST API.

g) The Root Server Service sends the SAML Assertion and receives User Authorization (Groups) information from the WebUI AppServer.

CA&F: SAML Authentication from Remote Control



Network Legend

Port	Line	Notes
52311	⟷	HTTP/HTTPS
443	⇠⇢	HTTPS

SAML Authentication Flow

a) User navigates to the RC Server Service URL and is redirected to the Identity Provider.

- RC Server Service redirects the browser to IdP, e.g., Active Directory Federation Service (ADFS).

b) IdP requests authentication credentials from the user via the browser and receives the entered credentials.

c) IdP requests and receives stored user account information to complete the authorization from Active Directory.

- IdP determines if the user has authenticated successfully

d) IdP sends the SAML Assertion to the Browser

e) Browser sends SAML Assertion to RC Server Service.

- This connection (secure and authenticated) is used for the balance of communication between Browser/Controller and RC Server.

End of Deck