



# Cisco Finesse AMA

Architecture, Deployment and Troubleshooting.

Ritesh Desai

Technical Consulting Engineer, Global Contact Center

29-June-2022 – 13-July-2022.

# AGENDA

- 1 About Finesse
- 2 Architecture.
- 3 Serviceability.
- 4 Design.
- 5 Features.
- 6 Logging
- 7 Troubleshooting
- 8 Best practices
- 9 Important URL's

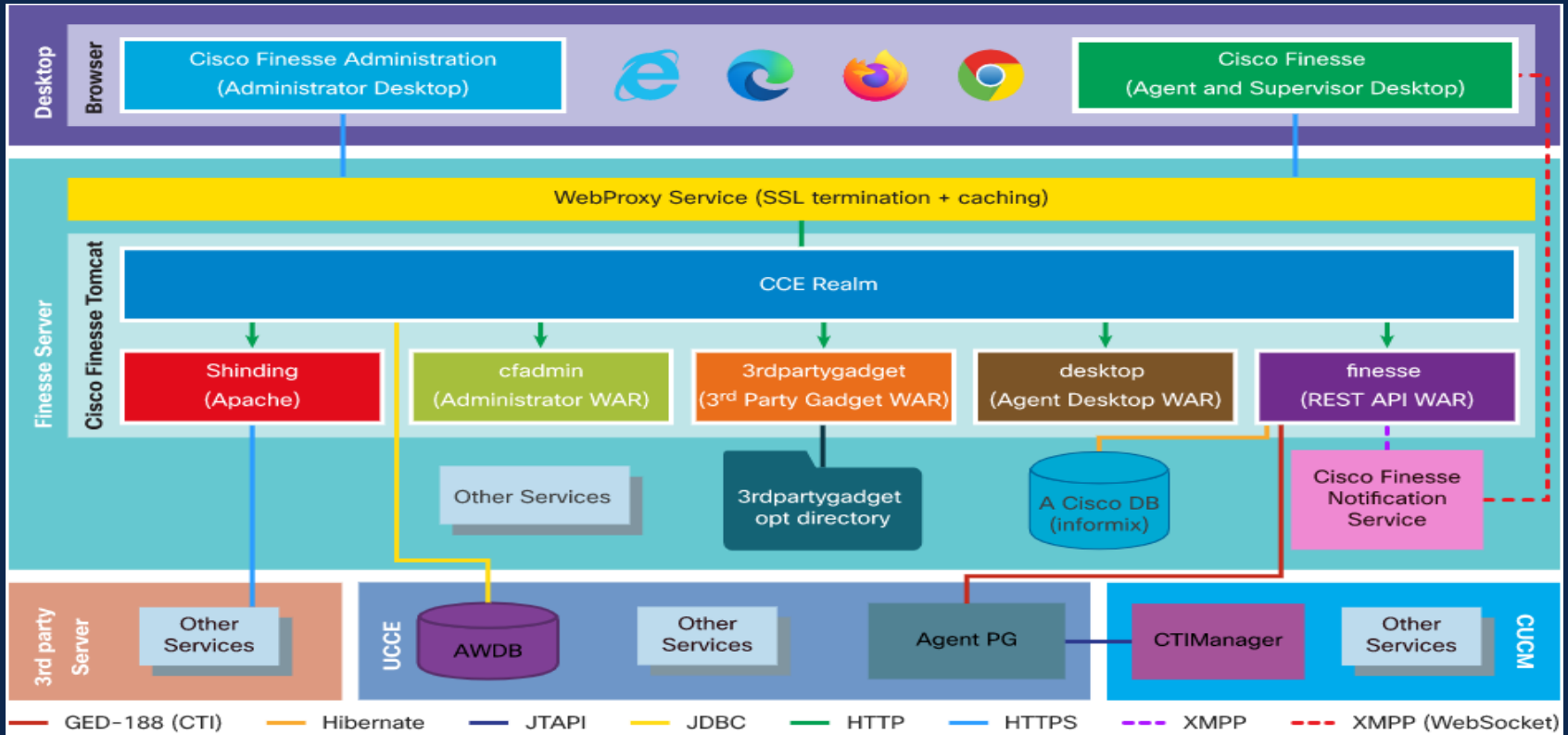
# TOPIC's COVERED

- About Finesse
- ARCHITECTURE - Finesse
- ARCHITECTURE - Deployment
- ARCHITECTURE – Websockets, XMPP and BOSH
- ARCHITECTURE – Websockets v/s BOSH protocol flow.
- ARCHITECTURE – WebSocket's events.
- ARCHITECTURE – Finesse Out of Box.
- ARCHITECTURE – Finesse in CCE solution.
- DESIGN – Network considerations.
- SERVICEABILITY – Finesse services.
- FEATURES – v12.6 Newly introduced.
- LOGGING – v12.6 Newly introduced.
- POSSIBILITIES when Finesse is Out of Service (OOS)
- BEST PRACTICES while raising SR with TAC
- IMPORTANT URL's

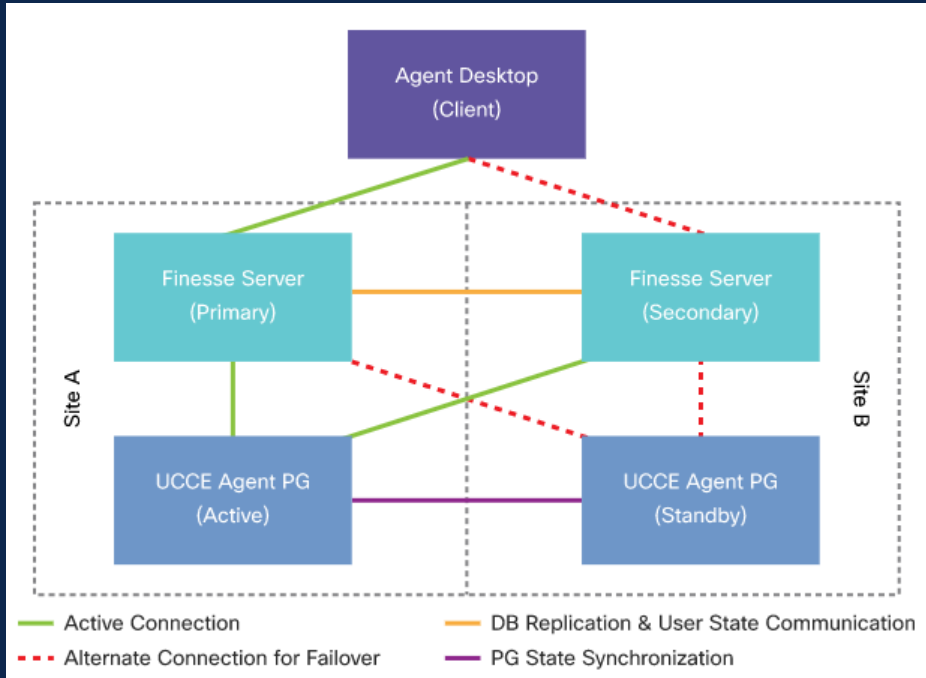
# About Finesse

1. Cisco Finesse is a next-generation agent and supervisor desktop designed to provide a collaborative experience for the various communities that interact with your customer service organization
2. Cisco Finesse offers transparent integration with the Cisco Collaboration portfolio. It is standards compliant and offers low cost customization of the agent and supervisor desktops.
3. Cisco Finesse provides;
  1. An agent and supervisor desktop that integrates traditional contact center functions into a thin-client desktop.
  2. A 100% browser-based desktop implemented through a web 2.0 interface; no client-side installations required.
  3. Gives customer care providers quick and easy access to multiple assets and information sources through single customizable interface.
  4. Open web 2.0 APIs that simplify the development and integration of value-added applications and minimize the need for detailed desktop development expertise.
4. With help of gadget container multiple applications accessible to agents on single UI framework at the same time. Gadgets can communicate with one another as well as backend servers, allowing for a seamless user experience throughout the call.
5. REST API - robust and easier the API is to use. Lower the cost of developing applications.
6. Finesse is a web application - browser based agent desktop. No browser plugins, JRE or any installation required on client machines to run the finesse application (administration and agent desktop).
7. CCX and CCE features different architecture.

# ARCHITECTURE - Finesse

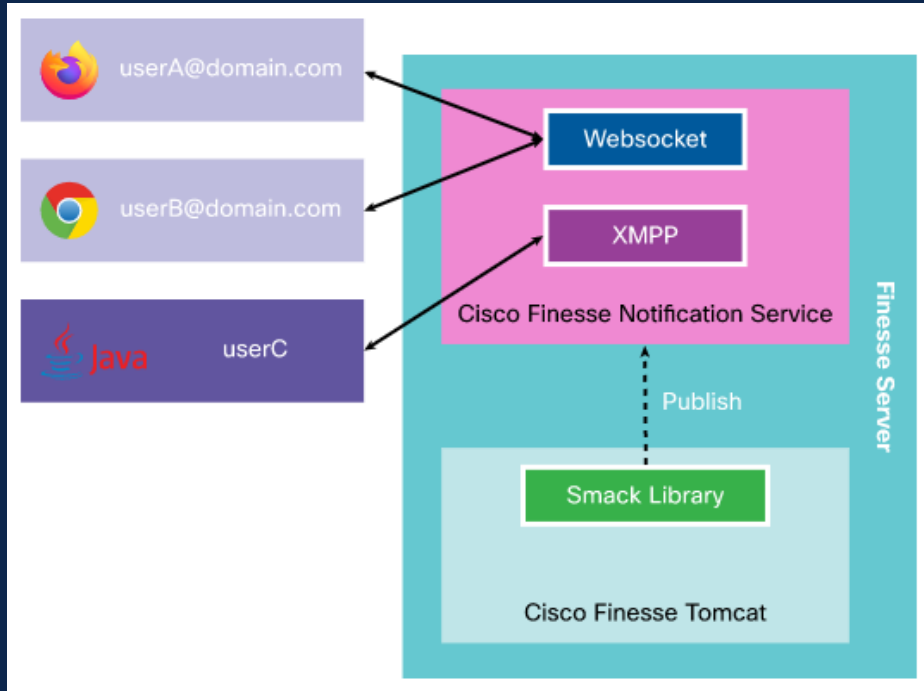


# ARCHITECTURE - Deployment



- **Standard deployment** of Finesse servers in Enterprise deployments.
- **Active-Active** mode in Enterprise deployments.
- **At a time**, primary and subscriber connects to **1 PG node**.
- Clients can **dynamically initiate failover** to the other server and recover their state if connectivity failures.
- Internal user and dialog state in sync via the events received via the CTI interface.

# ARCHITECTURE – Websockets, XMPP and BOSH



- **BOSH – Deprecated** from v12.6
- WebSocket / XMPP is used for communication.
- WebSocket: TCP 8445
- Unsecure XMPP: TCP 5222
- Secure XMPP: TCP 5223.

# ARCHITECTURE – Websockets, XMPP and BOSH

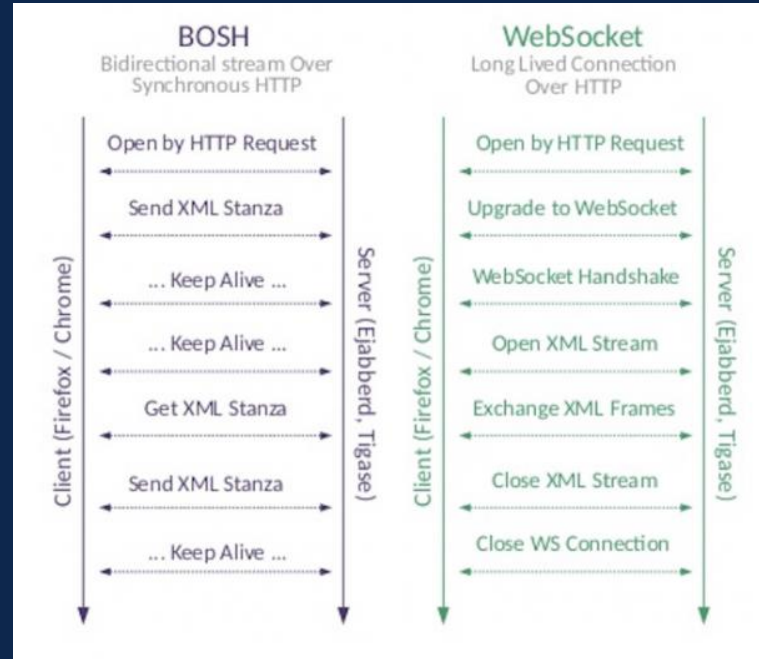
- Cisco Finesse Notification service is an instance of **OpenFire server**, running as a separate process on the platform, that uses **XMPP** protocol as the data communication channel and encoding mechanism for dispatching user and dialog events to the client via a seemingly asynchronously.
- For web applications, in order to send the events to the browser, which communicates only via the HTTP protocol, the **XMPP payload needs to be encapsulated within the HTTP protocol** either using **WebSocket** or **BOSH** (deprecated starting Finesse 12.6(1)).
- **WebSockets** is an advanced technology that makes it possible to open an interactive communication session between the user's browser and a server. With this API, you can send messages to a server and receive event-driven responses, **without having to poll the server for a reply**.
- WebSocket connections are established like regular HTTP connections but using a **ws:// or a wss:// URI**, the **server handshake upgrades the connection to WebSocket protocol**, and it is treated like a regular TCP connection after this initial handshake.



# ARCHITECTURE – Websockets, XMPP and BOSH

- **Bidirectional-streams Over Synchronous HTTP (BOSH)** allows real-time communication between a browser and a web server.
- The browser connects to the server and **will keep the connection open as long as it has no data to send**.
- HTTP is a request-response protocol, the BOSH client in the browser works by sending a HTTP request to the BOSH server (OpenFire) **asking for any outstanding events**. If there are no events to send, the request is kept pending for a certain period of time instead of returning immediately with an error or an empty payload. If events arrive at OpenFire from Finesse during this time, it is sent as a HTTP response to the pending HTTP request. The BOSH client receives these events and immediately fires the next HTTP request to OpenFire for the next set of events. If, on the other hand, the timer expires before any events are available to be dispatched, a HTTP response with a heartbeat message is sent back to indicate the liveness of the OpenFire server to the client. This explains why the event dispatch from the Finesse server to the browser appears to happen in a **"seemingly asynchronous manner"**.

# ARCHITECTURE – Websockets v/s BOSH protocol flow.



# ARCHITECTURE – WebSocket's Tunnel created

The screenshot displays a network log on the left and a detailed view of a request on the right. A red circle highlights the request to `https://finpri.raducce.com/tunnel/` in the log, with a red arrow pointing to the corresponding request details on the right. In the details pane, the `Request URL` and `path` fields are also circled in red.

Time	Method	URL	Status	Size	Time
03:00:46.652	200	GET https://finpri.raducce.com/desktop/scripts/js/cd.layout.js	83 ms		
03:00:46.658	200	GET https://finsec.raducce.com/finesse/api/SystemInfo	6 ms		
03:00:46.785	200	OPTIONS https://finsec.raducce.com/finesse/api/SystemInfo	186 ms		
03:00:46.789	200	GET https://finsec.raducce.com/finesse/api/SystemInfo	9 ms		
03:00:47.008	200	OPTIONS https://finsec.raducce.com/finesse/api/SystemInfo	234 ms		
03:00:47.259	200	GET https://finpri.raducce.com/finesse/api/SystemInfo	11 ms		
03:00:47.291	200	GET https://finpri.raducce.com/finesse/api/User/114909665	49 ms		
03:00:47.350	200	GET https://finpri.raducce.com/tunnel/	10 ms		
03:00:47.414	101	716 GET https://finpri.raducce.com/tunnel/openftr.js	21189 ms		
03:00:47.478	200	GET wss://finpri.raducce.com/ws/	38 ms		
03:00:47.539	202	GET https://finpri.raducce.com/finesse/api/User/114909665/Dialogs	20 ms		
03:00:47.622	200	PUT https://finpri.raducce.com/finesse/api/User/114909665	49 ms		
03:00:47.683	200	GET https://finpri.raducce.com/finesse/api/User/114909665/Workflows	24 ms		
03:00:47.709	200	GET https://finpri.raducce.com/finesse/api/TeamResource/5000/ReasonCodes			

**Request Details:**

- Request URL:** https://finpri.raducce.com/tunnel/
- HTTP Version:** http/2.0
- Request method:** GET
- Remote Address:** 192.168.1.14
- Headers:**
  - `authority` finpri.raducce.com
  - `method` GET
  - `path` /tunnel/
  - `scheme` https
  - `accept` text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9
  - `accept-encoding` gzip, deflate, br
  - `accept-language` en-US,en;q=0.9
  - `cache-control` no-cache
  - `cookie` timeBeforeFailover=1652803756324; timeBeforeAttemptingLoginiframe=1652803818819; seqNumberGenerated=1; attemptsMade=18; finesse\_ag\_extension=400001
  - `pragma` no-cache
  - `referrer` https://finpri.raducce.com/desktop/container/?locale=en\_US&fromlogout=true
  - `sec-ch-ua` "Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
  - `sec-ch-ua-mobile` ?0
  - `sec-ch-ua-platform` "Windows"
  - `sec-fetch-dest` iframe
  - `sec-fetch-mode` navigate
  - `sec-fetch-site` same-origin
  - `upgrade-insecure-requests` 1
  - `user-agent` Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.0.0 Safari/537.36

**Query arguments:**

A WebSocket detects the presence of a proxy server and automatically sets up a tunnel to pass through the proxy.

# ARCHITECTURE – WebSocket's Tunnel created

The screenshot displays the network tab of a web browser's developer tools. The left pane shows a list of network requests, with the 200 OK response for `https://finpri.raducce.com/tunnel/` highlighted in yellow. A green arrow points from this entry to the right pane, which shows the details of the response. The response status is `200 OK`, and the server is identified as `nginx`. A yellow box at the bottom of the response details indicates that the cache data is missing.

Time	Method	URL	Status	Size	Timing
03:00:46.335	GET	https://finpri.raducce.com/desktop/api/app-config	200	11 ms	
03:00:46.471	GET	https://finpri.raducce.com/desktop/scripts/fonts/CiscoSansTTRegular.woff	200	110 ms	
03:00:46.539	GET	https://finpri.raducce.com/finesse/api/SystemInfo	200	89 ms	
03:00:46.541	GET	https://finpri.raducce.com/desktop/scripts/js/cd.layout.vendor.js	200	87 ms	
03:00:46.652	GET	https://finpri.raducce.com/desktop/scripts/js/cd.layout.js	200	83 ms	
03:00:46.658	GET	https://finsec.raducce.com/finesse/api/SystemInfo	200	6 ms	
03:00:46.785	OPTIONS	https://finsec.raducce.com/finesse/api/SystemInfo	200	186 ms	
03:00:46.789	GET	https://finsec.raducce.com/finesse/api/SystemInfo	200	9 ms	
03:00:46.789	OPTIONS	https://finsec.raducce.com/finesse/api/SystemInfo	200	234 ms	
03:00:47.008	GET	https://finpri.raducce.com/finesse/api/SystemInfo	200	11 ms	
03:00:47.259	GET	https://finpri.raducce.com/finesse/api/User/114909665	200	9 ms	
03:00:47.291	GET	https://finpri.raducce.com/finesse/api/TeamResource/5000/ReasonCodes	200	49 ms	
03:00:47.350	GET	https://finpri.raducce.com/tunnel/	200	10 ms	
03:00:47.414	GET	https://finpri.raducce.com/tunnel/openfire.js	101	716	21189 ms
03:00:47.478	GET	wss://finpri.raducce.com/ws/			

**Response Details:**

- Server's response:** 200 OK
- Full response:** http/2.0 More info about HTTP status codes in [Wikipedia](#).
- Headers:**
  - accept-ranges:** bytes
  - content-length:** 941
  - content-security-policy:** default-src \* unsafe-inline; frame-ancestors https://finpri https://finpri.raducce.com https://finpri.raducce.com:8445 https://finpri:8445 https://finsec https://finsec.raducce.com https://finsec.raducce.com:8445 https://finsec:8445;
  - content-type:** text/html
  - date:** Mon, 04 Jul 2022 02:58:15 GMT
  - last-modified:** Thu, 29 Apr 2021 11:25:28 GMT
  - server:** nginx
  - strict-transport-security:** max-age=31536000; includeSubdomains
  - x-cache-status:** HIT
  - x-content-type-options:** nosniff
  - x-xss-protection:** 1; mode=block

**Cache:** Cache data missing  
The cache information is missing from the entry

**Size:** Headers: (computed) 607 bytes

The tunnel is established by issuing an HTTP CONNECT statement to the proxy server, which requests that the proxy server open a TCP/IP connection to a specific host and port. Once the tunnel is set up, communications can flow unimpeded through the proxy. Since HTTP/S works in a similar fashion, secure WebSockets can leverage the same HTTP CONNECT technique over SSL.

# ARCHITECTURE – WebSocket's client request.

The screenshot displays a network log on the left and a detailed view of a request on the right. A blue circle highlights the request for `ws://finpri.raducce.com/ws/` in the network log, and another blue circle highlights the same request URL in the 'Request' tab. A blue arrow points from the highlighted request in the network log to the detailed view on the right.

**Network Log (Left):**

Time	Method	URL	Status	Size	Time
03:00:46.652	GET	https://finpri.raducce.com/desktop/scripts/js/cd.layout.js	200	83 ms	
03:00:46.658	GET	https://finsec.raducce.com/finesse/api/SystemInfo	200	6 ms	
03:00:46.785	OPTIONS	https://finsec.raducce.com/finesse/api/SystemInfo	200	186 ms	
03:00:46.789	GET	https://finsec.raducce.com/finesse/api/SystemInfo	200	9 ms	
03:00:47.008	GET	https://finpri.raducce.com/finesse/api/SystemInfo	200	234 ms	
03:00:47.259	GET	https://finpri.raducce.com/finesse/api/User/114909665	200	11 ms	
03:00:47.291	GET	https://finpri.raducce.com/finesse/api/TeamResource/5000/ReasonCodes	200	9 ms	
03:00:47.350	GET	https://finpri.raducce.com/tunnel/	200	49 ms	
03:00:47.414	GET	https://finpri.raducce.com/tunnel/openfire.js	200	10 ms	
03:00:47.478	GET	wss://finpri.raducce.com/ws/	101	716	21189 ms
03:00:47.539	GET	https://finpri.raducce.com/tunnel/TimedCallbackWorker.js	200	38 ms	
03:00:47.622	PUT	https://finpri.raducce.com/finesse/api/User/114909665	202	20 ms	
03:00:47.683	GET	https://finpri.raducce.com/finesse/api/User/114909665/Dialogs	200	49 ms	
03:00:47.709	GET	https://finpri.raducce.com/finesse/api/TeamResource/5000/Workflows	200	24 ms	

**Request Details (Right):**

Request on 2022-07-04T03:00:47.478Z

**General:**

- Request URL: wss://finpri.raducce.com/ws/
- HTTP Version: HTTP/1.1
- Request method: GET

**Headers:**

- Accept-Encoding: gzip, deflate, br
- Accept-Language: en-US,en;q=0.9
- Cache-Control: no-cache
- Connection Upgrade
- Cookie: timeBeforeFailover=1652803756324; timeBeforeAttemptingLoginInframe=1652803818819; seqNumberGenerated=1; attemptsMade=18; finesse\_ag\_extension=400001
- Host: finpri.raducce.com
- Origin: https://finpri.raducce.com
- Pragma: no-cache
- Sec-WebSocket-Extensions: permessage-deflate; client\_max\_window\_bits
- Sec-WebSocket-Key: uKifQsvpuMuLNf10USyVRA==
- Sec-WebSocket-Protocol: xmpp
- Sec-WebSocket-Version: 13
- Upgrade: websocket
- User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/101.0.0.0 Safari/537.36

**Headers:** 716 bytes  
**Body:** 0 bytes  
**Total:** 716 bytes

WebSocket connections are established like regular HTTP connections but using a `ws://` or a `wss://` URI. The server handshake upgrades the connection to WebSocket protocol, and it is treated like a regular TCP connection after this initial handshake. The browser sends a request to the server, indicating that it wants to switch protocols from HTTP to WebSocket. The client expresses its desire through the Upgrade header:

# ARCHITECTURE – WebSocket's server response.

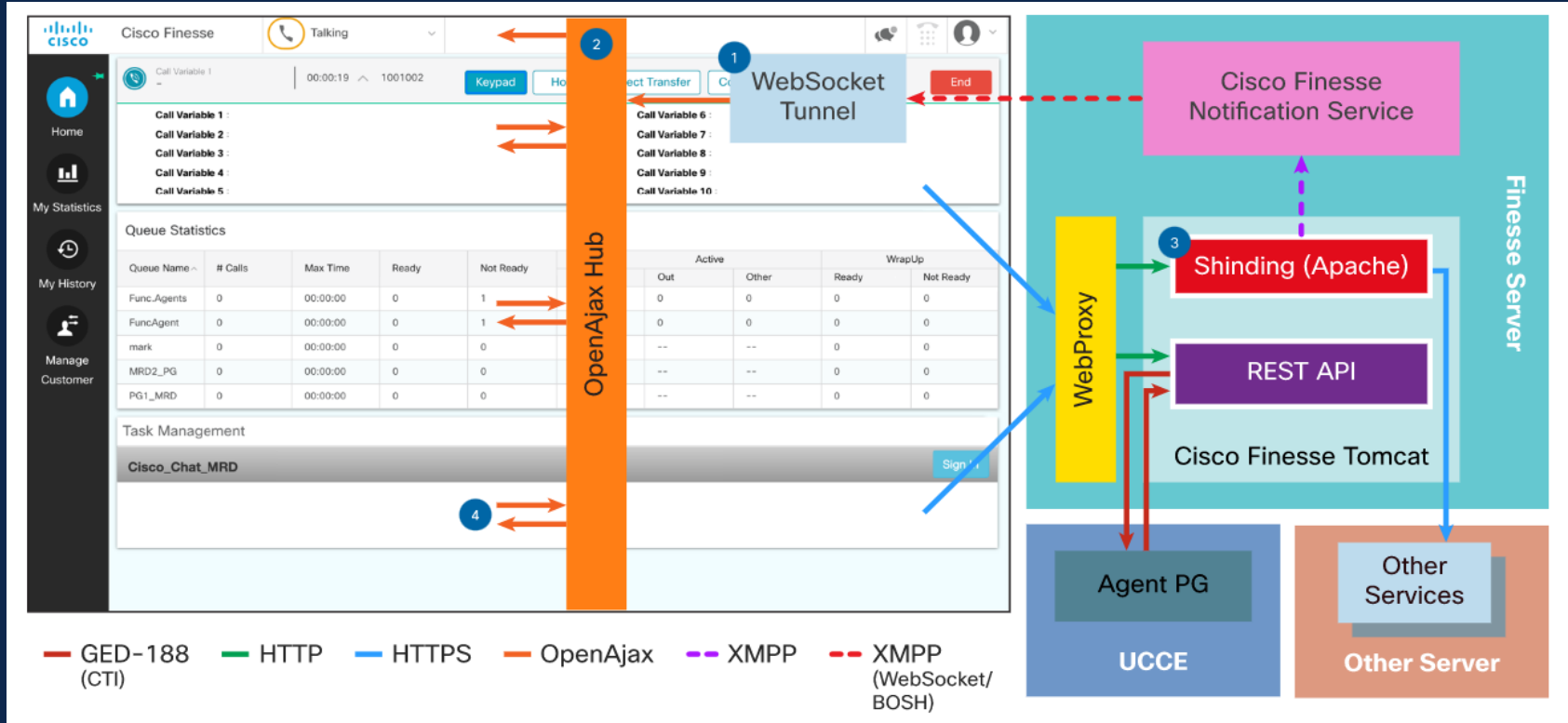
The screenshot displays a network log on the left and a detailed view of a response on the right. The network log entry at 03:00:47.478 shows a 101 status code for the request `GET wss://finpri.raducce.com/ws/`. The details pane on the right shows the response headers:

- **Connection** upgrade
- **Date** Mon, 04 Jul 2022 02:58:15 GMT
- **Sec-WebSocket-Accept** qzBntuIBOHTVqMIRJTjaI0iY6E=
- **Sec-WebSocket-Extensions** permessage-deflate
- **Sec-WebSocket-Protocol** xmpmp
- **Server** Webproxy
- **Upgrade** WebSocket

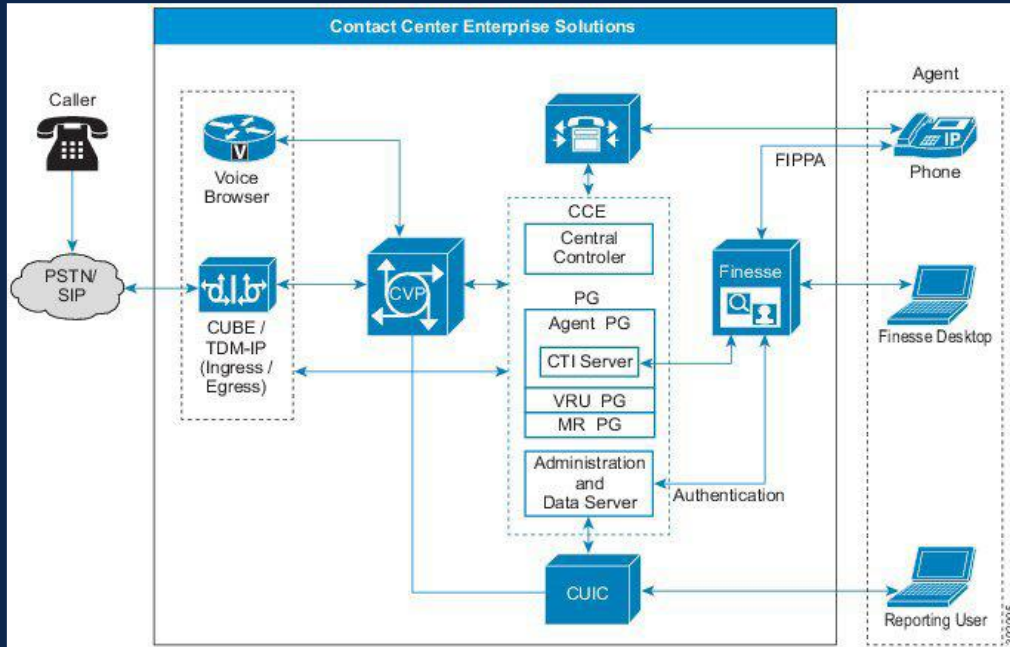
The response body is highlighted as "Cache data missing".

At this point, the server understands the WebSocket protocol, it agrees to the protocol switch through the Upgrade header. The HTTP connection breaks down and is replaced by the WebSocket connection over the same underlying TCP/IP connection. By default, the WebSocket connection uses the same ports – HTTP (80) and HTTPS (443). This allows WebSocket to work reliably even in the presence of strict firewall restrictions. If web access is guaranteed to the client, WebSockets should work – even in the presence of web proxies.

# ARCHITECTURE – Finesse Out of Box.



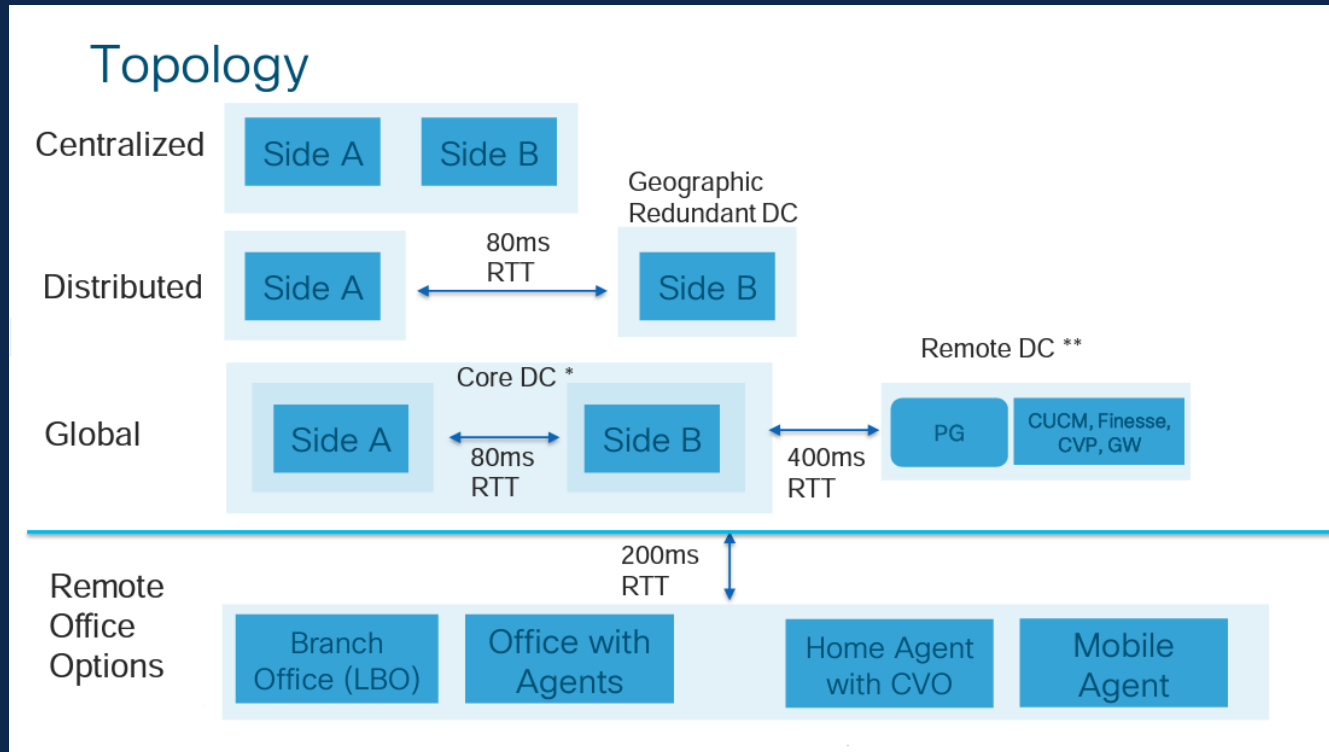
# ARCHITECTURE – Finesse in CCE solution.



- Cisco Finesse connects to the Administration and Data server for **authentication** so that agents can sign in.
- Finesse server must show as **registered** in CTI Manager under procmon command line utility. Command: *clients*.
- **FIPPA** – Finesse server communicates with Cisco IP Phone configured with service URL.



# DESIGN – Network considerations.



# SERVICEABILITY – Finesse services.

- **Cisco Finesse Notification service** - service is used for messaging and events. The Cisco Finesse desktop uses this service to view call events, agent state changes, and statistics.
- **Cisco Finesse Tomcat service** - service contains all deployed Cisco Finesse applications as below;
  - Cisco Finesse desktop application – Agent – Supervisor desktop (browser) interface.
  - Cisco Finesse IP Phone Agent (FIPPA) application - Agent – Supervisor Cisco IP Phone login.
  - Cisco Finesse REST API application - provides a programming interface, client applications to access supported server features, can use HTTPS to transport application data.
  - Cisco Finesse administration application - administrative operations for Cisco Finesse.
  - Cisco Finesse Diagnostic Portal application - provides performance-related information for Cisco Finesse.

# FEATURES – v12.6 Newly introduced.

1. New **Desktop Interface API's** Desktop Configuration, Languages List and Verify Desktop and Third-Party URLs
2. **Agent Device Selection** - When agents and supervisors need to use different devices that are configured with the same extension
3. Edge Chromium Browser Support (**Microsoft Edge**).
4. **Agent PG Maintenance Mode** - peripheral gateway (PG) maintenance mode, which allows the Cisco Finesse server to reconnect to the alternate PG without interrupting the current operations. No interruption during sign in, state operations, or call operations.
5. **Finesse Maintenance Mode** - transitioning live Cisco Finesse nodes into a maintenance mode for performing administrative tasks, without causing any disruption to contact center activities.
6. **Multi-Tab Gadgets** - Finesse desktop supports accessing multiple gadgets through tabs within a single gadget called Multi-Tab gadget.
7. **Agent Answers** - Agent Answers feature provides relevant suggestions and recommendations in real time for the agent to consider.
8. **Call Transcript** – Real time transcript made available to Cisco Finesse Desktop.
9. **Change IP Address and Hostname**
10. **Supported Content Security Policy Directives** - release allows the administrator to use the CLI commands to view, add, or delete the frame-access sources in the response header of Cisco Finesse. This ensures that only the configured sources can embed the Cisco Finesse in an iFrame within their HTML pages.
11. **Drop Participant**: Ability for Agent/Supervisor to Drop Participant from a conference call.

# FEATURES – v12.6 Newly introduced.

## 12. New CLI

- **utils finesse locked\_out\_users** list has been added to view the list of locked out users. (Finesse v12.6(1) ES02).
- **utils finesse log configuration {add|update|delete|list}** - administrator can use the following CLIs to add, delete, update, or view the logger configuration in the system for Finesse.
- **utils finesse set\_property webservices enableExternalNotificationPortAccess** to *true* to enable the external access to the Cisco Finesse Notification Service XMPP port (5223).
- **utils finesse notification external\_port\_access {add|delete|list}** - Restricting Access to the External XMPP Notification Port 5223.
- **utils finesse layout updateCuicGadgetUrl 12.5.1** - specific CLIs to change the .jsp references of Cisco Unified Intelligence Center (CUIC) gadgets to .xml in the Finesse desktop layout.
- **utils finesse layout updateCuicGadgetUrl 12.6.1+** - specific CLIs to change the .jsp references of Cisco Unified Intelligence Center (CUIC) gadgets to .xml in the Finesse desktop layout.
- **utils finesse show\_connected\_users summary** - administrators can view the list of connected users in the current Finesse server.
- **utils finesse show\_connected\_users detail** - administrators can view the list of connected users in the current Finesse server.

13. **REST APIs** - Get List of Devices for Extension, Finesse MaintenanceMode—Get, Finesse MaintenanceMode—Update, ConnectedUsersInfo—Summary, ConnectedUserInfo—Get Connected Users Information.

14. **jQuery** - jQuery version hosted by Finesse has been upgraded from 3.4.1 to 3.5.1.

15. **API Authentication changes for VPN-Less Deployment (ES02 Update)** - authentication changes made for VPN-Less deployment, primarily impacts third-party desktops and external API access.

16. **SystemInfo API (ES02 Update)** - SystemInfo API is now authenticated when accessed via VPN-Less proxy.

17. **Connected Agents Gadget** - the gadget is for administrators that lists all the agents currently signed into Cisco Finesse.

# LOGGING – v12.6 Newly introduced.

Additional debugging level introduced on top of default logging that is enabled already.

## *utils finesse log configuration add*

Creates a custom log configuration in the Finesse system. The logs record information about the encountered issues of different severity levels for a specific Finesse module.

Command Syntax:

*utils finesse log configuration add [module] [name] [level]*

**[module]** Unique name of Finesse module for which log configuration has to be added.

The module name is case sensitive. The following are the valid Finesse modules.

***admin, audit, desktop, diagnostics, fippa, realm, shindig, valve, webservice***

**[name]** Unique name of the log configuration. The log configuration name is case sensitive.

**[level]** Defines the different severity level associated with the log configuration. The following are the valid log configuration levels.

***OFF, ERROR, WARN, INFO, DEBUG, TRACE, ALL***

# LOGGING – v12.6 Newly introduced.

## *utils finesse log configuration list*

Lists all log configurations in the Finesse system

Command Syntax:

*utils finesse log configuration list*

Example:

The following is the sample output for all the log configuration in the Finesse system

```
admin:utils finesse log configuration list
Requesting log configurations, please wait...
Below is the list of log configurations in Finesse.
```

No.	Module	Name	Level
1.	admin	ROOT	INFO
2.	audit	ROOT	INFO
3.	desktop	ROOT	INFO
4.	diagnostics	ROOT	INFO
5.	fippa	ROOT	INFO
6.	realm	ROOT	INFO
7.	shindig	ROOT	INFO
8.	valve	ROOT	INFO
9.	webservices	ROOT	INFO
10.	fippa	org.jivesoftware	WARN
11.	webservices	org.hibernate	INFO
12.	webservices	com.cisco.cc.common.subsystem	TRACE

# LOGGING – v12.6 Newly introduced.

## *utils finesse log configuration delete*

Deletes an existing custom log configuration in the Finesse system.

Command Syntax:

*utils finesse log configuration delete [module] [name]*

Example:

The following is the sample output for deleting the log configuration named **com.cisco.cc.common.subsystem** under the Finesse **webservices** module's

### ***Important Notes***

- ROOT log configuration cannot be deleted.

```
admin:utils finesse log configuration delete webservices com.cisco.cc.common.subsystem
Deleting log configuration, please wait...
Successfully deleted the log configuration. Changes might take approximately 30 seconds to
take effect.
```

# LOGGING – v12.6 Newly introduced.

## *Important Notes*

- Earlier trace CLIs used to enable/disable/check trace logging in Finesse systems has been **removed**  
*utils finesse trace enable*  
*utils finesse trace disable*  
*utils finesse trace status*
- Earlier notification logging CLIs used to enable/disable trace logging in Cisco Finesse Notification Service has been **removed**.  
*utils finesse notification logging enable*  
*utils finesse notification logging disable*
- Setting the log configuration level to DEBUG or TRACE impacts system performance. It is advised to set custom log configuration only for debug purpose and need to be deleted after the debug.
- Understand the impact of the custom logging before enabling on the production systems. Please consult CiscoTAC if required.



# LOGGING – v12.6 Newly introduced.

- New directories such as *shindig*, *valve* and are added under finesse logging directory which specifically contains the logs of those components.
- Name format of the logs has been changed. For example,  
*Desktop-webservices.2021-01-15T00-03-45.208.log*  
has been changed to  
*webservices. 2021-01-15T00-03-45.208.log*
- Name conventions used earlier to indicate start-up of Cisco Finesse Tomcat Service has been removed. i.e. earlier logs containing log name suffix *startup* is no longer present. Instead, *shutdown* suffix has been added to log name to indicate Cisco Finesse Tomcat Service shutdown.  
Example:  
*Desktop-webservices.2020-08-19T04-24-04.303.startup.log*  
*webservices.2021-02-03T17-27-13.737.log.shutdown.zip*

# FINESSE SYSTEM URL's

Finesse Product version	<a href="https://&lt;FQDN&gt;/finesse-dp/rest/DiagnosticPortal/GetProductVersion">https://&lt;FQDN&gt;/finesse-dp/rest/DiagnosticPortal/GetProductVersion</a>
Finesse SystemInfo	<a href="https://&lt;FQDN&gt;/finesse/api/SystemInfo">https://&lt;FQDN&gt;/finesse/api/SystemInfo</a>
Finesse logs	<a href="https://&lt;FQDN&gt;/finesse/logs">https://&lt;FQDN&gt;/finesse/logs</a>
Get Performance Info	<a href="https://&lt;FQDN&gt;/finesse-dp/rest/DiagnosticPortal/GetPerformanceInformation">https://&lt;FQDN&gt;/finesse-dp/rest/DiagnosticPortal/GetPerformanceInformation</a>
Connected Users info	<a href="https://&lt;FQDN&gt;/finesse/api/ConnectedUsersInfo">https://&lt;FQDN&gt;/finesse/api/ConnectedUsersInfo</a>
Users Info	<a href="https://&lt;FQDN&gt;/finesse/api/User/&lt;agentid&gt;">https://&lt;FQDN&gt;/finesse/api/User/&lt;agentid&gt;</a>
MaintenanceMode status	<a href="https://&lt;FQDN&gt;/finesse/api/MaintenanceMode">https://&lt;FQDN&gt;/finesse/api/MaintenanceMode</a>
Runtime Config Info	<a href="https://&lt;FQDN&gt;/finesse/api/RuntimeConfigInfo">https://&lt;FQDN&gt;/finesse/api/RuntimeConfigInfo</a>
Locked Out Users	<a href="https://&lt;FQDN&gt;/finesse/api/LockedOutUsers">https://&lt;FQDN&gt;/finesse/api/LockedOutUsers</a>

- Finesse server must be UP for the REST API to function and respond.
- For REST-API authentication, the app-admin user created during installation credentials required.

# Possibilities when Finesse is Out of Service (OOS)

Common scenario's when Finesse reports **Finesse server is out of Service**.  
**Please contact system administrator.**

- **Public / Private Link failure** between SideA and SideB.
- ICM **rtr** service down / bounce.
- ICM rtr **mds** down / bounce.
- ICM rtr **ccag** service down / bounce.
- ICM PG **pgag**
- ICM CG **ctisvr** service down / bounce.
- ICM PG **jgw1** service down / bounce.
- ICM Agent PG **pim** service down / bounce / CONFIGURING.
- Core process **crash**.
- **pguser password mismatch** (CG installation / CUCM APP user)
- Agent phone **extension length mismatch**.
- Port communication failure.

# BEST PRACTICES while raising SR with TAC

Publish the following details for the TAC engineer to understand about the architecture.

- Define PCCE or UCCE with agent deployment.
- Mention if any ES installed.
- Is Finesse cluster geographically distributed over the WAN or is on LAN (Same DC).
- How many nodes of Finesse in architecture?
- Any troubleshooting been carried and if yes, what is identified or results.
- Any recent changes in infrastructure or network that have caused any issues?
- Was this working before?
- Is the issue re-producible or its intermittent?
- How did the reported issue occur. What led to this issue?
- Duration of the outage?
- If call drops, the call information (ANI, DNIS, timestamp the call was made) to trace in logs.

# IMPORTANT URL's

1. **CCE Compatibility Matrix:** <https://www.cisco.com/c/en/us/support/customer-collaboration/unified-contact-center-enterprise/products-device-support-tables-list.html>
2. **PCCE Virtualization guide:** [https://www.cisco.com/c/dam/en/us/td/docs/voice\\_ip\\_comm/uc\\_system/virtualization/virtualization-packaged-contact-center-enterprise.html](https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-packaged-contact-center-enterprise.html)
3. **UCCE Virtualization guide:** [https://www.cisco.com/c/dam/en/us/td/docs/voice\\_ip\\_comm/uc\\_system/virtualization/virtualization-unified-contact-center-enterprise.html](https://www.cisco.com/c/dam/en/us/td/docs/voice_ip_comm/uc_system/virtualization/virtualization-unified-contact-center-enterprise.html)
4. **Cisco Devnet:** <https://developer.cisco.com/docs/finesse/#!finesse-overview/finesse-out-of-the-box-desktop>
5. **Finesse 12.6 release notes:**  
[https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cust\\_contact/contact\\_center/icm\\_enterprise/icm\\_enterprise\\_12\\_6\\_1/release/guide/rcct\\_b\\_cce-solution-rns-12-6/cfin\\_m\\_1261\\_finesse-release-notes.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/icm_enterprise/icm_enterprise_12_6_1/release/guide/rcct_b_cce-solution-rns-12-6/cfin_m_1261_finesse-release-notes.html)
6. **Solution Design Reference Guide:**  
[https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cust\\_contact/contact\\_center/icm\\_enterprise/icm\\_enterprise\\_12\\_6\\_1/design/guide/ucce\\_b\\_ucce\\_soldg-for-unified-cce-1261.html](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/icm_enterprise/icm_enterprise_12_6_1/design/guide/ucce_b_ucce_soldg-for-unified-cce-1261.html)
7. **Finesse Administration Guide:**  
[https://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cust\\_contact/contact\\_center/finesse/finesse\\_1261/admin/guide/cfin\\_b\\_1261-cisco-finesse-administration-guide/cfin\\_m\\_1261-cisco-finesse-failover-mechanisms.html#Cisco\\_Concept.dita\\_665eecd-f2d2-44b8-8336-278570d622ed](https://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cust_contact/contact_center/finesse/finesse_1261/admin/guide/cfin_b_1261-cisco-finesse-administration-guide/cfin_m_1261-cisco-finesse-failover-mechanisms.html#Cisco_Concept.dita_665eecd-f2d2-44b8-8336-278570d622ed)
8. **Cisco Troubleshooting technotes:** <https://www.cisco.com/c/en/us/support/customer-collaboration/finesse/series.html#~tab-documents>
9. **Cisco Support Community:** <https://community.cisco.com/t5/contact-center/bd-p/5926-discussions-contact-center>

**THANK YOU**

For your time and attention

