

# UCCX Finesseアーキテクチャの詳細について

## 内容

---

[はじめに](#)

[前提条件](#)

[要件](#)

[使用するコンポーネント](#)

[背景説明](#)

[50,000フィートのビュー](#)

[FinesseのTomcat](#)

[HTTP\(S\)](#)

[XMPP](#)

[PUBSUB](#)

[BOSH:Bi-Directional Streams Over Synchronous HTTP \(同期HTTPによる双方向ストリーム\)](#)

[CTI](#)

[JTAPI](#)

[30000 ftビュー](#)

[休止](#)

[AXL\[AXL\]](#)

[SOAP](#)

[20000 ftビュー](#)

[アバッチ・シンディグ](#)

[WARファイル](#)

[10000 ftビュー](#)

[AJAX - Finesseの美しさ](#)

[AJAXを使用する利点](#)

[AJAXの動作](#)

[AJAXを使用してサーバーに要求を送信中](#)

[デスクトップアーキテクチャ](#)

[ガジェットのアーキテクチャ](#)

[参照リンク](#)

---

## はじめに

このドキュメントでは、Finesseの問題をトラブルシューティングする際に基盤となるプロセスが意味を持つように、Finesseのアーキテクチャについて詳しく説明します。

## 前提条件

要件

次のツールと機能に関する知識があることが推奨されます。

JTAPI:Java Telephony API ( ジャバテレフォニーAPI )

API : アプリケーションプログラミングインターフェイス

UCCX:Unified Contact Center Express

CUCM:Cisco Unified Communications Manager

CTI:Computer Telephony Integration ( コンピュータテレフォニーインテグレーション )

使用するコンポーネント

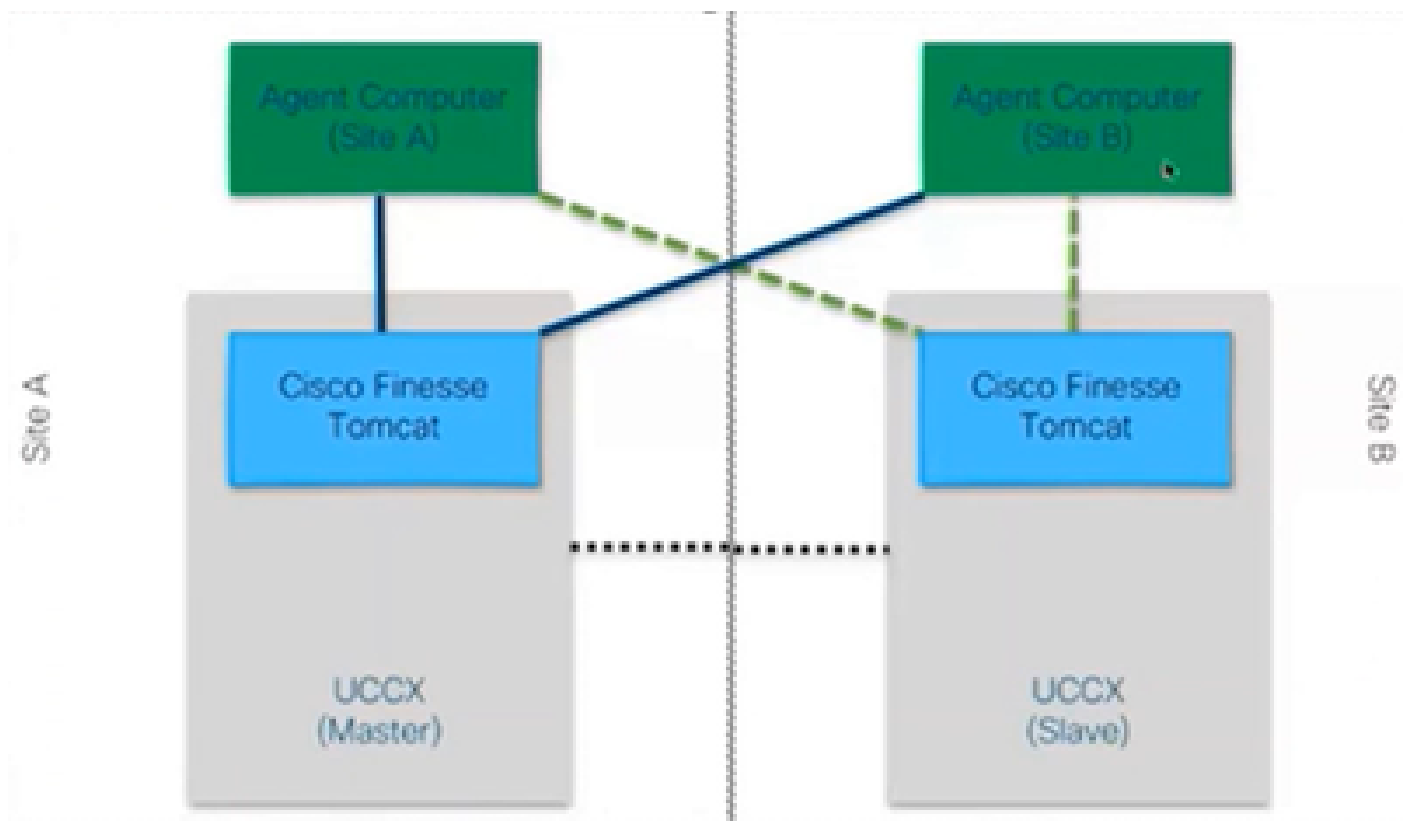
- Cisco Unified Contact Center Express ( UCCX )

このドキュメントの情報は、特定のラボ環境にあるデバイスに基づいて作成されました。このドキュメントで使用するすべてのデバイスは、クリアな ( デフォルト ) 設定で作業を開始しています。本稼働中のネットワークでは、各コマンドによって起こる可能性がある影響を十分確認してください。

背景説明

このドキュメントでは、Finesseアーキテクチャについて概要から説明し、次に信号フローの詳細と例および図を示します。

50,000フィートのビュー



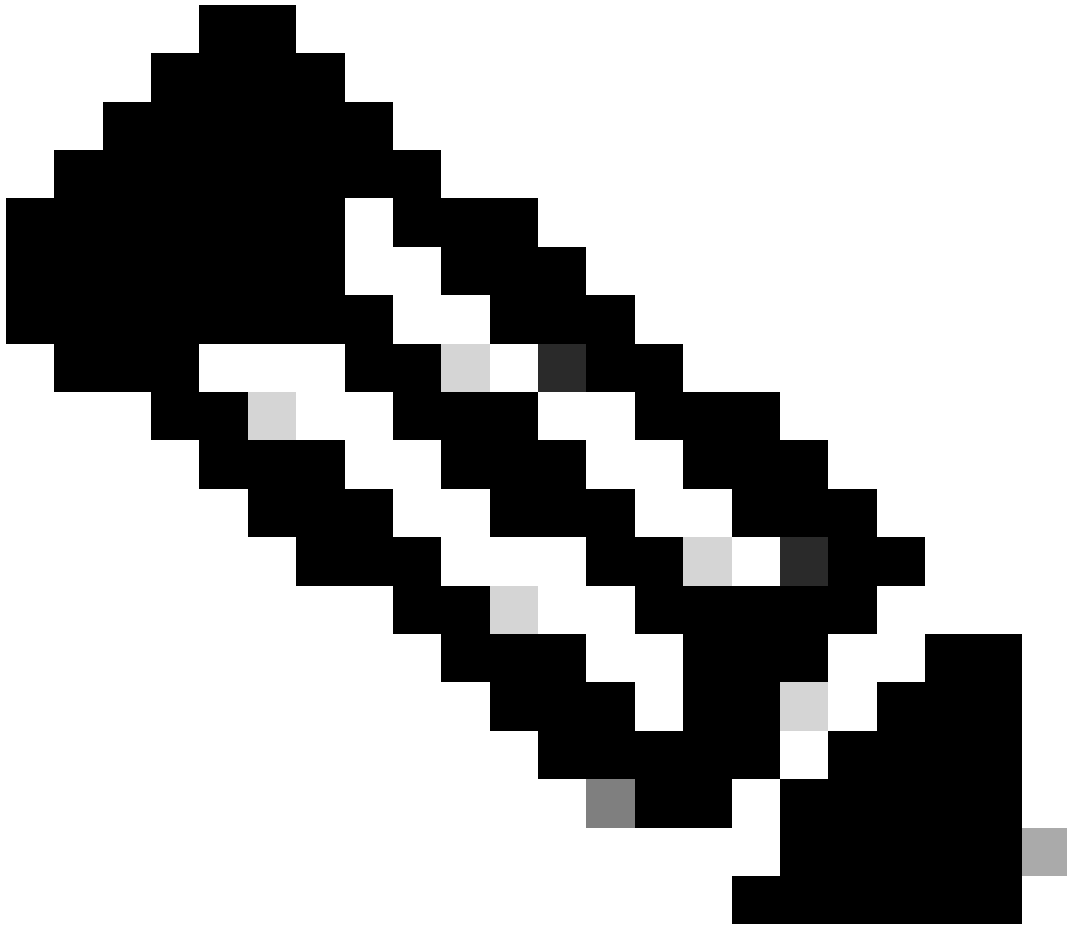
## FinesseのTomcat

Finesse Tomcatの機能は、Webページのロードと同じですが、Finesseという別のサービス向けであるため、CUCMのCisco Tomcatに似ています。Tomcatは独立したWebアプリケーションであるため、Finesse用に設計されています。11.5より前のバージョンに従って、CCXマスターノードを使用してのみFinesseにログインできます。11.6以降では、フェールオーバー中に限り、任意のノードにログインできます(ただし推奨されません)。したがって、両方のエージェント(サイトAとサイトB)がマスターノードのFinesseに接続されているWANクラスタを考えた場合、Cisco Finesseからエンジン(ISE)への他方のノードは常に非アクティブな接続になります。これは、フェールオーバーに必要なCTI openconf要求です。

Openconf要求は、ノードがアクティブかスタンバイかを確認するために定期的送信されます。フェールオーバーが発生した場合、通知サービスはsysteminfo APIと呼ばれるAPIを使用します。このAPIは、このノードがダウンしており、フェールオーバーが必要であることをクライアントに通知します。次に、ブラウザを他のノードにリダイレクトするファイルが実行され、openconf rejavascriptsponseが送信されます。このフェールオーバーは、証明書が受け入れられる場合にのみ機能します。(サーバへのセキュアな接続を確立するため)。

次の3つの証明書が提示されます。

- そのノードの通知サービス証明書。
- リモートノードの通知サービス証明書。
- リモートノードのFinesseサービス証明書。



注：フェールオーバーが機能するには、リモートノード証明書が受け入れられる必要があります。

---

## HTTP(S)

HTMLなどのハイパーメディアドキュメントを送信するためのアプリケーション層プロトコルです。これは、WebブラウザとWebサーバ間の通信を目的として設計されており、ステートレスプロトコルであるため、サーバは2つの要求間でデータを保持しません。これはクライアントサーバプロトコルです。UCCXでは、Finesseクライアントはエージェントブラウザ(PC)で実行されます。HTTPを使用してサーバに要求を行います。一般的なHTTPメソッドの一部を次に示します。

GET：サーバから情報を取得します。

POST

: サーバに情報を送信します。

PUT – サーバ上の何かを置き換えます。

DELETE – サーバから情報を削除します。

Finesseはhttp要求内でsysteminfo api要求を使用します。たとえば、エージェントの状態を変更する場合、ブラウザはPOSTの代わりにPUTを送信します。これは、POSTが送信されると、サーバが2つの状態を保持しているために混乱することがあるためです。どちらを選択しますか。PUTを使用すると、現在の状態が置き換えられます。

## XMPP

eXtensible Messaging and Presence Protocol(eXtensible Messaging and Presence Protocol)

インスタントメッセージングとプレゼンスに使用されるプロトコルセットです。すべてのXMPPエンティティは、それぞれのJabber IDまたはJIDを使用して識別されます。ガジェットに使用されるこのXMPPプロトコルの拡張機能の1つは、PUBSUBと呼ばれます。

## PUBSUB

これは、考えられるパブリッシャサブスクリバではありません。まだパブリッシュと登録は行われますが、データベースとは無関係です。XMPPはノードと呼ばれるメカニズムを使用します。ノードは基本的に、対象となるエンティティを監視します。あなたに重要で、それを監視したいものはすべて、それにノードを追加します。次に、PUBSUBは、そのノード上で更新または通知を購読します。エージェントやダイアログなど、エンティティのタイプごとにノードを設定できます。エージェント用のノードを作成すると、そのエージェントのノードにサブスクリバされ、その後エージェントが何を行ったとしても、そのノードに関する通知を受け取ります。

この仕様の目的は、XMPPサーバ(通知サービス)がXMPPノード(トピック)に公開された情報を取得し、そのノードに加入しているエンティティにXMPPイベントを送信できるようにすることです。

Finesseの場合、コンピュータテレフォニーインテグレーション(CTI)サーバはCTIメッセージをFinesse Webサービスに送信して、エージェントやコンタクトサービスキュー(CSQ)の作成などの設定の更新やコールに関する情報についてFinesseに通知します。この情報は、Finesse WebサービスがFinesse Notificationサービスに公開するXMPPメッセージに変換されます。次に、Finesse Notificationサービスは、特定のXMPPノードに加入しているエージェントにBOSHメッセージを介してXMPPを送信します。

## BOSH:Bi-Directional Streams Over Synchronous HTTP (同期HTTPによる双方向ストリーム)

BOSHは長寿命のHTTP接続で、サーバは応答があるまで要求を保留し、応答がない場合は空の応答を送信します。これはXMPPクライアントとXMPPサーバで動作しますが、非XMPPアプリケーションでも使用できます。



注:XMPPはステートフルであるのに対し、HTTPはステートレスです (最後の要求に関する情報は保存しません)。

---

WebアプリケーションがXMPPと連携する必要がある場合、複数の問題が発生します。

問題1:Transmission Control Protocol (TCP ; 伝送制御プロトコル) を介したXMPPは、ブラウザでネイティブにサポートされていません。

解決策1:Webサーバとブラウザはハイパーテキスト転送プロトコル(HTTP)メッセージを介して通信するため、Finesseおよびその他のWebアプリケーションはXMPPメッセージをHTTPメッセージ内にラップします。

問題2:HTTPはステートレスプロトコルです。

解決策2：このためにクッキー/投稿データを使用できます。

問題3:3番目の問題は、HTTPの単方向動作です。これは、クライアントだけが要求を送信し、サーバだけが応答できることを意味します。サーバがデータをプッシュできないため、HTTP経由でXMPPを実装するのは不自然です。

解決策3：この問題を解決するには、HTTPとXMPPの間にブリッジが必要です。

提案するソリューションは次のとおりです。

- ポーリング (レガシープロトコル)：新しいデータの要求を繰り返すHTTP要求の定義：HTTPポーリング
- ロングポーリングは、BOSHとも呼ばれます。これは、頻繁なポーリングを使用せずに複数の同期HTTP要求/応答ペアを効率的に使用することによって、2つのエンティティ間の長期間にわたる双方向TCP接続のセマンティクスをエミュレートするトランスポートプロトコルです。BOSHを使用する理由は、要求があるとすぐにサーバが応答する必要がないという事実を隠すことです。応答は、サーバがクライアントのデータを取得するまで指定された時間まで遅延され、応答として送信されます。クライアントは、応答を受け取るとすぐに新しい要求を行います。

Finesseデスクトップクライアント (Webアプリケーション) は、30秒ごとにTCPポート7443経由で古いBOSH接続を確立します。30秒後、Finesse通知サービスからアップデートがない場合、通知サービスは200 OKと (ほぼ) 空の応答本文を含むHTTP応答を送信します。エージェントまたはダイアログ (コール) イベントの存在に関する更新が通知サービスに含まれている場合、データは即座にFinesse Webクライアントに送信されます。

まとめ

Finesse Webクライアントに、TCPポート7443経由でFinesseサーバへの古いHTTP接続(http-bind)が設定されています。これはBOSHロングポーリングと呼ばれます。Finesse通知サービスは、エージェント、コールなどの状態に関する更新をポストするプレゼンスサービスです。通知サービスに更新が含まれている場合、通知サービスはHTTP応答本文内のXMPPメッセージとして状態更新を含むhttp-bind要求に応答します。http-bind要求を受信した30秒後に状態更新がない場合、通知サービスは状態更新なしで応答し、Finesse Webクライアントが別のhttp-bind要求を送信できるようにします。これは、Finesse Webクライアントが引き続き通知サービスに接続でき、エージェントがブラウザを閉じたりコンピュータをスリープさせたりしなかったことを通知サービスが認識するための手段となります。

## CTI

コンピュータテレフォニーインテグレーション(CTI)を使用すると、コンピュータ処理機能を利用しながら、通話の発信、受信、および管理を行うことができます。CTIアプリケーションでは、発信者IDを使用してデータベースからユーザ情報を取得するなどのタスクを実行したり、自動音声応答(IVR)システムによって収集された情報を使用して、ユーザからのコールをその情報とともに適切なサービス担当者にルーティングしたりできます。CUCMのCTI ManagerがUCCXからのJTAPI要求に応答します。CTIサーバのTCPポートは12018です。これは、Finesseサーバ (CTIサーバ) とエンジン (CTIサーバ) が相互に通信する方法です。

CTIを介して交換される情報の一部を次に示します。

- 現在のシステム構成と将来のアップデート
- エージェントとその状態。
- コールとその状態。

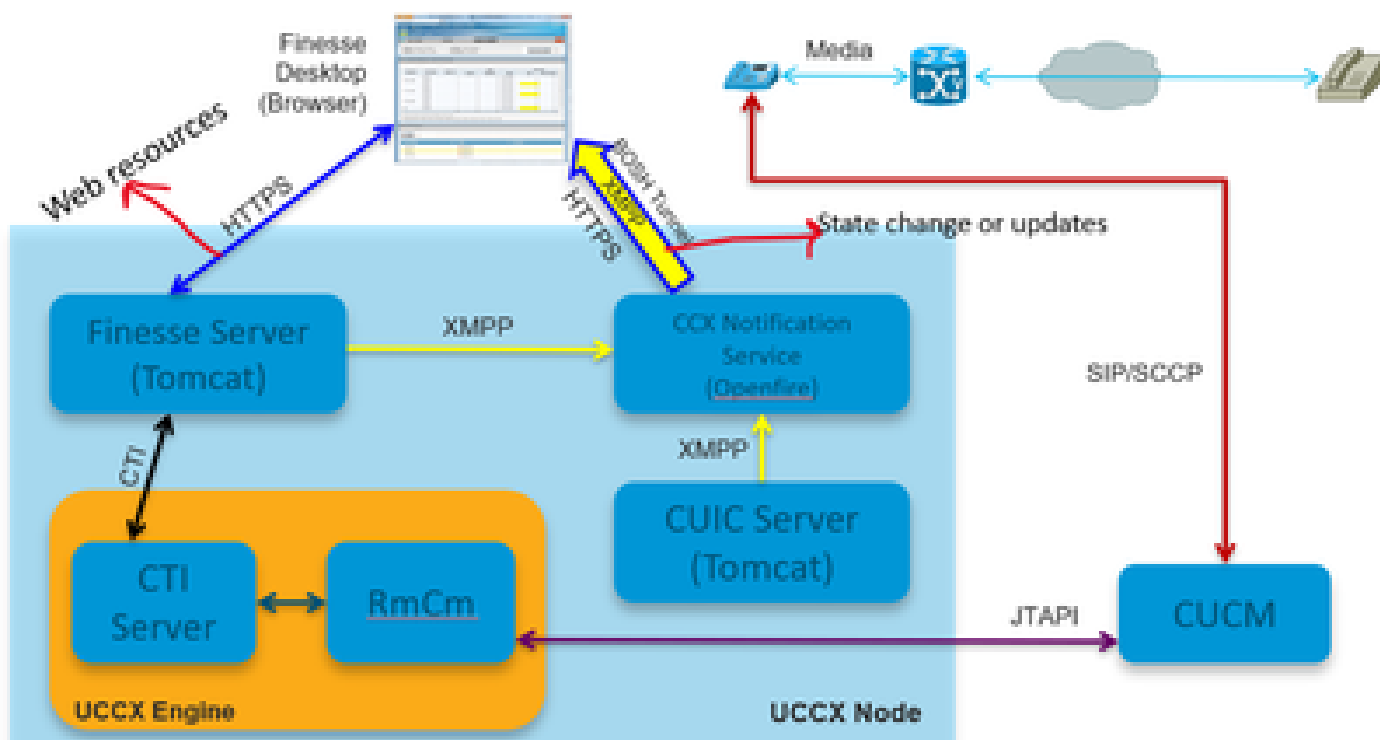
- エージェント、コール、およびキューの統計情報をリアルタイムで表示。

## JTAPI

Cisco Unified JTAPIは、Javaベースのコンピュータテレフォニーアプリケーション用にSun Microsystemsが開発したプログラミングインターフェイス規格として機能します。Cisco JTAPIは、Sun JTAPI 1.2仕様を追加のシスコ拡張機能とともに実装します。UCCXとCUCM間の通信はすべてJTAPI上で行われます。これは、CUCMとエンジン（テレフォニーサブシステム）が互いに通信する方法です。JTAPIは、CUCM電話機の制御とモニタリング、CTIポートとルートポイントを使用したコールのルーティング、CUCMでの録音の開始と停止、およびコールルーティング機能に使用されます。

## 30000 ftビュー

次の図は、UCCXエンジン、Finesse、CUCM、およびブラウザが相互に通信する方法を示しています。



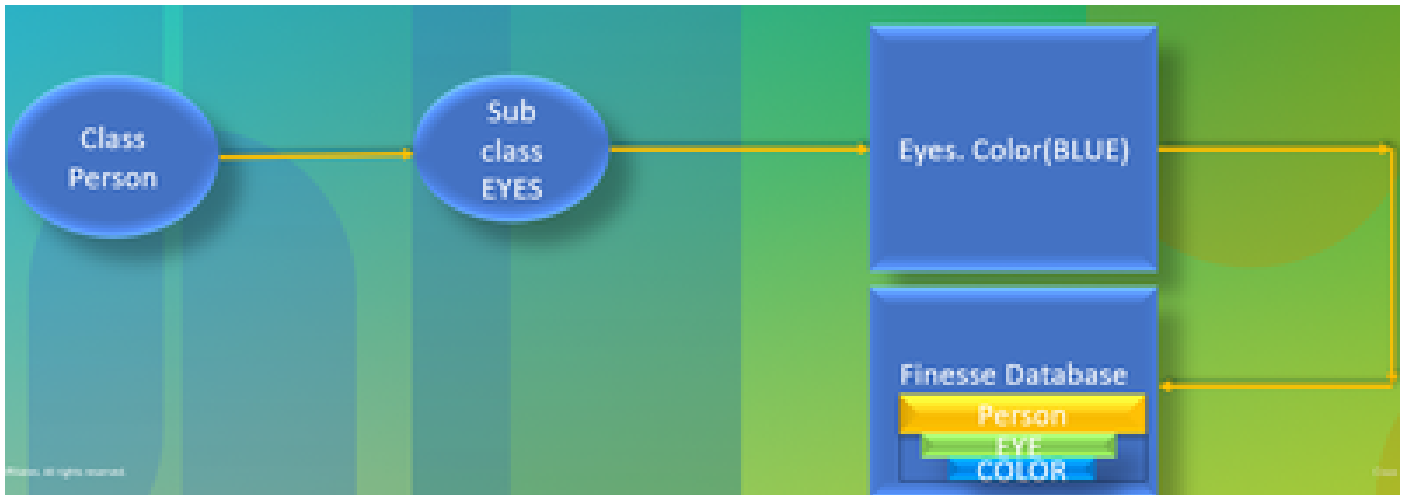
## 30000ビュー

ここでは、コールがエージェントと確立されていると見なします。ここで、JTAPIを介してエージェントの内線番号を監視しているRmCmは、エージェントが通話している状態の変更についてCTIサーバに通知します。この情報は、CTIを使用してCTIサーバ（CCXエンジン内）からFinesseサーバ(Tomcat)に送信されます。Finesseサーバは、状態変更に関するXMPPを使用してこの情報をCCX通知サービスに送信します。通知サービス(Openfire)は、エージェントブラウザへのBOSHトンネルを開き、状態変更に関する情報を更新します。これにより、エージェントがRESERVED状態に移行します。WARファイルやガジェットなどのHTTPSを使用して、あらゆるタイプのWebリソースがFinesseサーバに要求されます（まだキャッシュにない場合）。

## 休止

次の図は、Hibernateサービスについて説明しています。





休止

HIBERNATEは、高性能オブジェクト/リレーショナル持続性およびクエリーサービスと呼ばれます。簡単に言うと、JAVAクラスをデータベーステーブルにマッピングします。たとえば、Teamという名前のJAVAオブジェクトがあり、Teamという名前のFinesseデータベースにデータベーステーブルがあるとします。JAVAクラスはテーブル内の情報を制御し、HIBERNATEはその動作を制御します。SQLクエリーを使用する代わりに、Javaクラスを使用して情報を更新します。

## AXL[AXL]

管理用XML。

XMLはeXtensible Markup Language(eXtensible Markup Language)の略で、データを符号化するための比較的単純なルールを定義するマークアップ言語です。主に、構造化されたデータを、両方のシステムが理解できる明確に定義された形式で送受信するように設計されています。最も基本的な形式では、XMLは山カッコ(<>)で囲まれたタグを定義し、これらのタグはタグで記述されるデータを囲みます。タグは、他のタグ内にタグを持つ階層を形成できます。たとえば、基本的な電話デバイスを定義するには、電話デバイスに3つのパラメータ(名前、説明、電話番号)が必要であると言うことができます。

これは、CUCMでのリモートプロビジョニングを有効にするSOAPベースのAPIです。CUCMデータベースの情報を追加、更新、削除、または取得するために使用されます。取得機能には、ユーザ認証のチェックやSQLクエリの実行などがあります。AXL APIを使用すると、CUCMデータベース全体にアクセスできます。AXL APIは純粋にプロビジョニングのためのものであり、ランタイムデータやパフォーマンスデータへのアクセスは提供されません。

AXL APIはHTTPS基本認証を使用します。Standard CCM Super Usersアクセスコントロールグループのメンバー、またはStandard AXL API Accessロールが割り当てられているグループのメンバーであるCUCMユーザ(アプリケーションまたはエンドユーザ)は、AXLを介して読み取り/書き込みアクセスを持ちます。これは、すべてのスーパーユーザアカウントがすでに暗黙的にAXL APIにアクセスしていることを意味します。AXL API専用のアカウントを作成するには、まずアクセスコントロールグループを作成して、Standard AXL API Accessロールを割り当て、アプリケーションユーザを新しく作成したグループに関連付ける必要があります。読み取り専用のAXL APIアクセスを提供するには、個別のアクセスコントロールグループを作成し、標準のAXL読み取り専用APIアクセス権限のみを割り当てます。

## SOAP

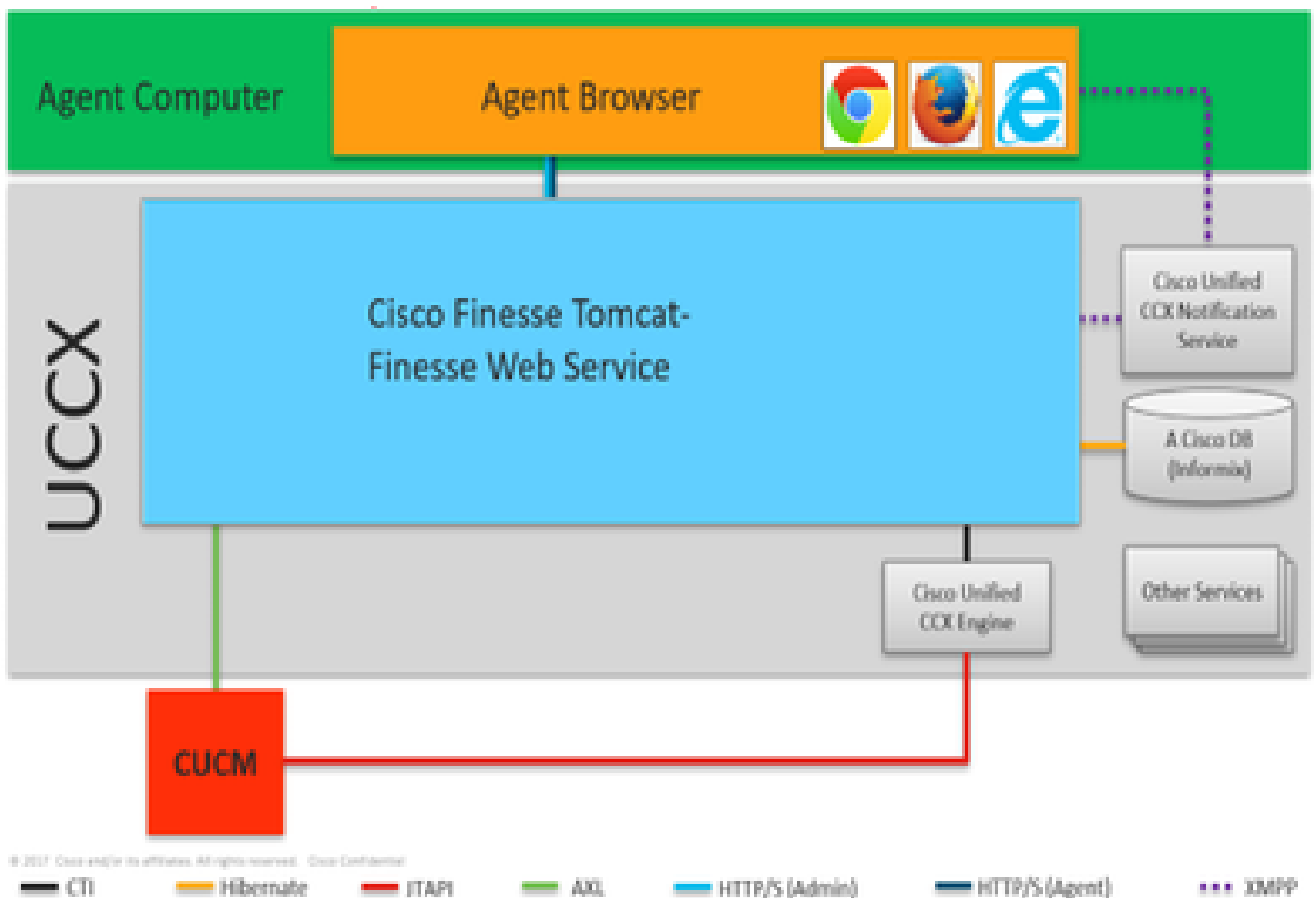
Simple Object Access Protocol(SOAP)は、XML形式でアプリケーション間で情報を渡す方法です。SOAPメッセージは、送信側アプリケーションから受信側アプリケーションに、通常はHTTPセッションを介して送信されます。実際のSOAPメッセージは、

Body要素とオプションのHeader要素を含むEnvelope要素で構成されます。

- エンベロープ:この必須要素はSOAPメッセージのルートで、送信されたXMLをSOAPパケットとして識別します。エンベロープには、本文セクションとオプションのヘッダーセクションが含まれています。
- ヘッダー:このオプションの要素は、メッセージの処理情報を示す拡張メカニズムを提供します。たとえば、メッセージを使用する操作でセキュリティ資格情報が必要な場合、それらの資格情報はエンベロープヘッダーの一部である必要があります。
- Body:この要素には、送信アプリケーションと受信アプリケーション間で送信される未加工データであるメッセージペイロードが含まれます。本体自体は複数の子要素で構成され、通常はこのデータの構造を定義するXMLスキーマを持ちます。

## 20000 ftビュー

次の図は、Finesseアーキテクチャに関連するプロトコルについて、より詳細に説明したものです。



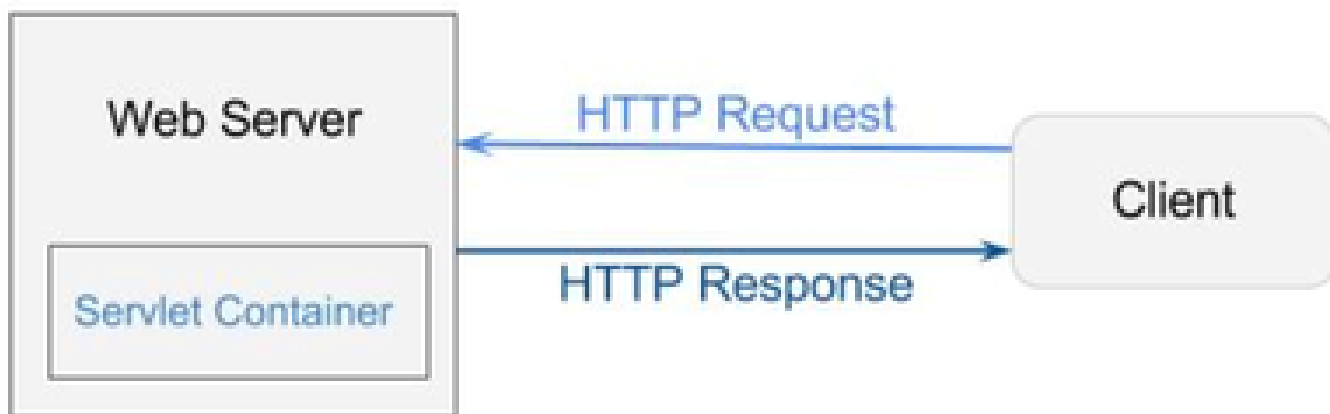
## 20000ビュー

これらは、異なるUCCXコンポーネント間の通信を行うプロトコルです。

- UCCXエンジンとFinesseサーバはCTIプロトコルで通信します。

- UCCXエンジンとCUCMはJTAPIプロトコルを介して通信します。
- Finesse TomcatとCUCMはAXLを介して通信します。
- Finesse通知サービスとエージェントブラウザはXMPP/BOSHで通信します。
- Finesse TomcatとデータベースはHibernateを介して対話します。
- Finesse TomcatとFinesse通知はXMPPを介して通信します。

## アパッチ・シンディグ



## シンディグ

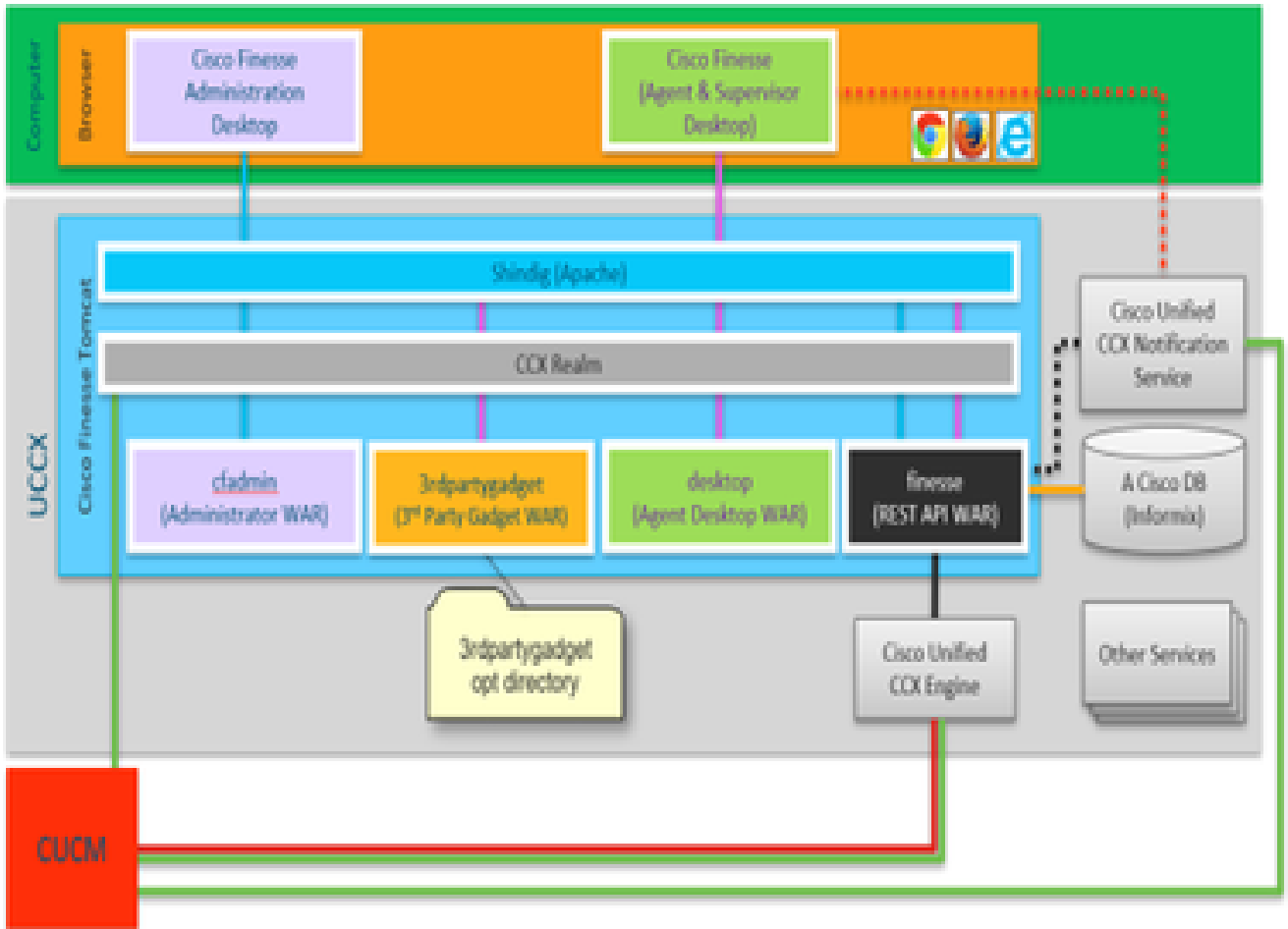
Apache ShindigはOpenSocialコンテナで、ガジェット、プロキシ要求、およびRESTとRPC要求を処理するためのコードを提供することで、OpenSocialアプリケーションのホスティングをすばやく開始できます。OpenSocialは、Web上で動作するソーシャルアプリケーションを構築するためのAPIのセットです。( Web/サーブレット ) コンテナは、Webサーバが動的にWebページを生成するために使用します。

## WARファイル

WARはWebアーカイブを表します。Webプロジェクトのファイルが含まれています。これには、サーブレット、XML、JSP、イメージ、HTML、CSS、JSなどが含まれます。Catalinaのログには、WARの配備に関する情報が含まれています。

## 10000 ftビュー

次の図は、UCCXとFinesseのコンポーネント内で認証フローがどのように機能するかを詳しく説明したものです。



© 2017 Cisco and/or its affiliates. All rights reserved. Cisco Confidential

■ CTI (SED-188) ■ Hibernate ■ JTAPI ■ AXL ■ HTTP/S (Admin) ■ HTTP/S (Agent) ■■■ XMPP (BOSH) ■■■ XMPP

10000ビュー

WARファイルは、ログイン方法に応じてページを表示および作成するために必要です。ブラウザはガジェットをレンダリングする必要があることをShindigに尋ね、shindigはガジェットをレンダリングするためにCUICと通信します。CCXレルムは、AXLを使用したCUCMでの認証に使用されます。通知サービスも、AXLを使用してCUCMで認証します。

Finesse Rest API WARは、通知サービス、CCXエンジン、およびDBと実際に通信するメインリポジトリです。cfadminとデスクトップWARはページを表示するだけなので、ShindigはFinesse Rest API ( Webサービス ) とだけ通信します。Finesse Rest API WARに関する情報は、Finesseの最も重要なログであるFinesse WebServicesログで確認できます。ShindigとFinesse Webサービス(Rest API WAR)の間でHTTPを使用します。Finesse Webサービス(Rest API WAR)とエンジンはCTIを介して相互に通信します。

### AJAX - Finesseの美しさ

AJAXは非同期JavascriptおよびXMLを表します。これはプログラミング言語ではなく、WebページからWebサーバにアクセスする方法です。AJAXは、部分的なページ更新を行うメカニズムです。ページ全体を更新しなくても、サーバーから取得したデータを使用してページのセクションを更新できます。

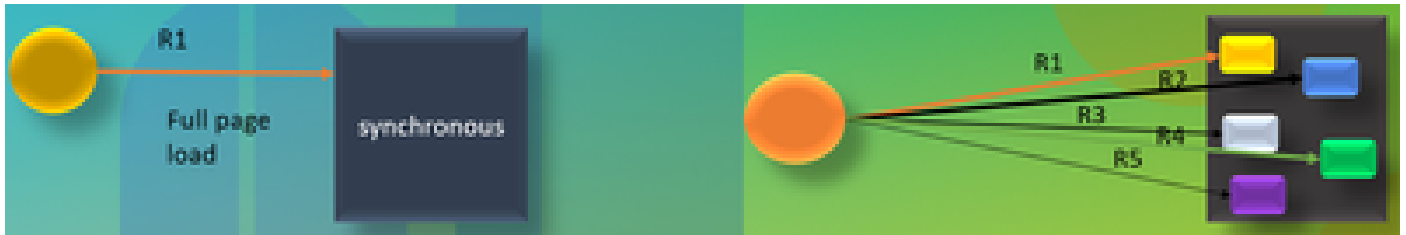
たとえば、Facebook Messengerについて話している場合、新しいメッセージが着信したときに、メッセージを取得するためにページ全体を更新する必要はありません。代わりに、ページのメッセージセクション自体が更新され、ページ全体を更新する必要なく、リアルタイムで新しいメッセージを取得します。

すべてのブラウザには、XMLHttpRequest(XHRとも呼ばれる)という組み込みオブジェクトがあります。サーバ内のAJAXへの

要求は、すべてこのXML要求を通過します。これには、更新が必要な項目の詳細が含まれています。

## AJAXを使用する利点

次の図は、非同期要求と同期要求の違いを示しています。



## アイアス

同期要求の場合は、最初の要求が処理されるのを待ってから、2番目の要求を送信できます。たとえば、ページの更新は必須であり、ページが更新されるまで何もできません。一方、非同期要求の場合は、最初の要求が完了するのを待たずに2番目の要求を送信できます。複数の要求を同時に送信できます。たとえば、Webサイト上の天気アプリケーションガジェット。ページ全体を更新することなく、ページの天気セクションを更新し、同時にWebサイトの他のセクションでも作業できます。これが非同期要求の主な利点です。

## AJAXの動作

AJAXは、Webサーバとの間で更新を送受信するために使用するXMLHTTPREQUEST(XHR)と、データの表示または使用に使用するJavascriptおよびHTMLを組み合わせたものです。

## AJAXを使用してサーバーに要求を送信中

これは、次に説明する3ステップのプロセスです。

1. 変数を作成し、その変数にXHRオブジェクトを格納する。

```
Var request = new XMLHttpRequest();
```

2. XHRオブジェクト内にペイロードを持つ要求変数にアクセスします。

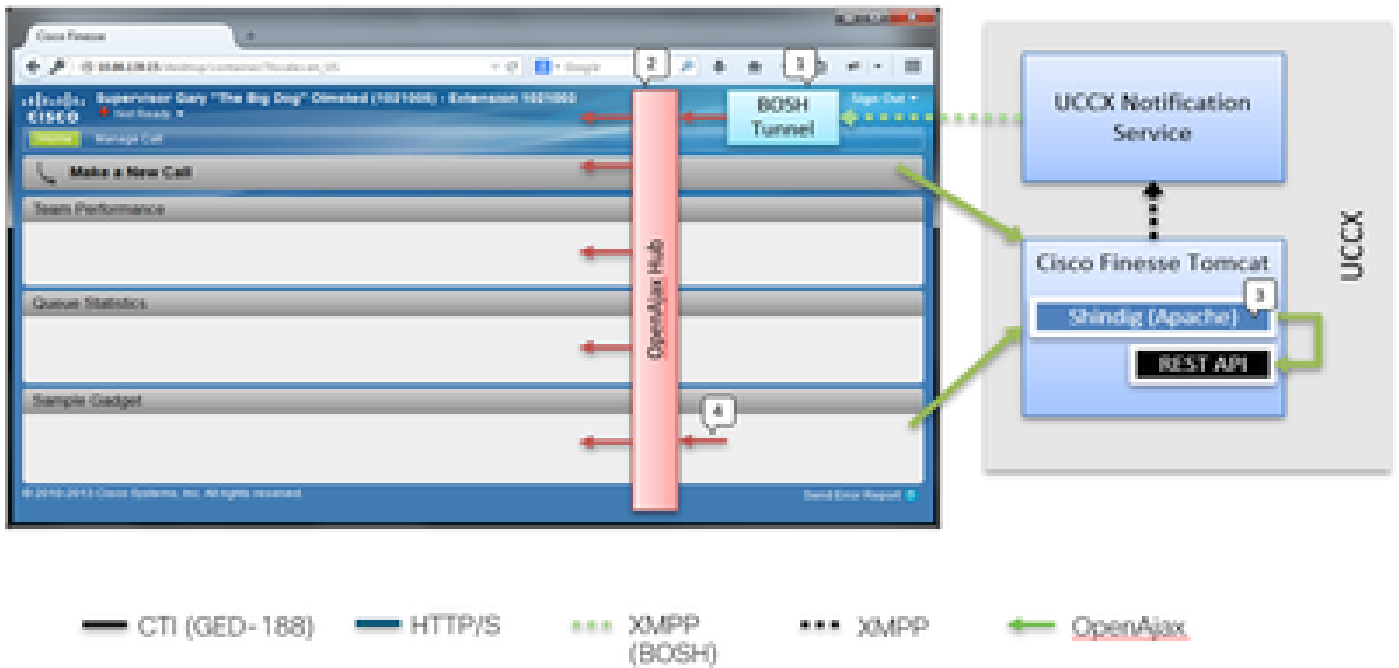
```
request.open(GET, URL)
```

3. 要求の送信

```
Request.send();
```

## デスクトップアーキテクチャ

次の図は、ガジェットがWebページにレンダリングされたときのAJAX信号のフローを示しています。

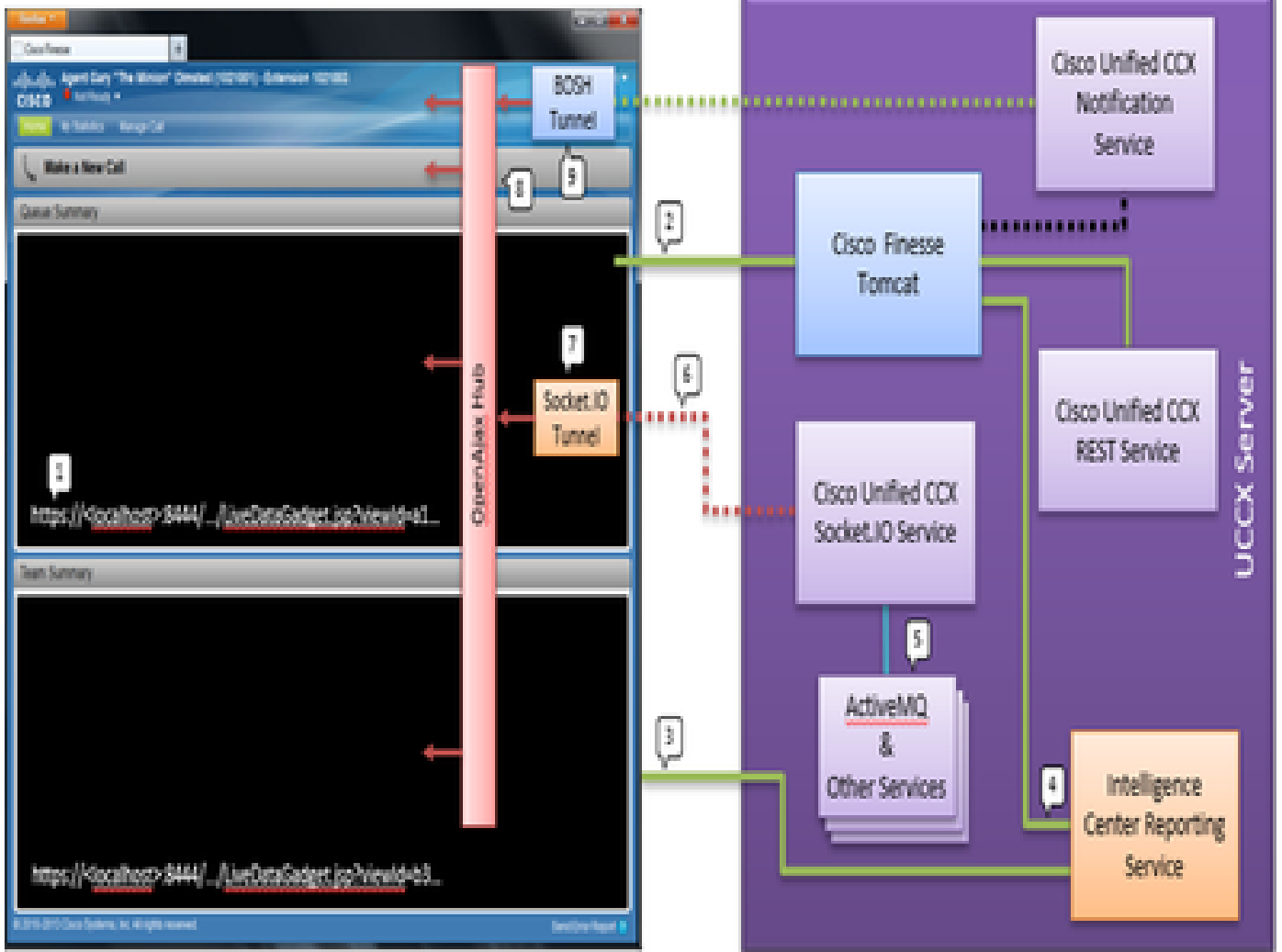


### デスクトップアーキテクチャ

IFrameは、BOSHトンネルをホストするコンテナに常駐します。OPENAJAXハブは、ガジェット全体にメッセージを発行するために提供されます ( pubsubメソッドを使用 )。REST要求は、Shindigを介して他のサーバにもプロキシされます。ガジェットは、AJAXハブ上に独自のメッセージを発行できます。

### ガジェットのアーキテクチャ

次の図は、Finesseガジェットのアーキテクチャの詳細を示しています。



— HTTP/S   
 - - - XMPP (BOSH)   
 - - - XMPP   
 ← OpenAjax   
 - - - Socket.IO   
 — JMS

### ガジェットのアーキテクチャ

一般的なガジェットとは異なり、CUICガジェットもOpenFireからリアルタイムXMPPフィードを受信します。CUICとFinesseがUCCXと共存するUCCXの場合、共有OpenFireインスタンスがあります。ガジェットのコンテンツとすべてのREST APIのほとんどは、FinesseサーバのShindigを介してプロキシされます。これは、FinesseガジェットとREST API、およびCUICガジェットインスタンスとREST APIに適用されます。CUICガジェットは、レポートの表示にDグリッドを使用します。ブートストラッププロセスが必要であり、これはCUICと直接連携して実行されます。このため、CUICガジェットは、最初はロード処理中にCUICサーバと直接対話します。このため、CUIC証明書は ( Socket.IOトンネルに加えて ) ユーザブラウザで受け入れられる必要があります。ガジェットのコンテンツとREST APIは、FinesseとCUICの間でクライアントにプロキシされます。Rest APIコールは、Intelligence Center Reporting ServiceとCCX Web Serviceの両方に対して行われます。CCX Live Data Socket.IOサービスは、ActiveMQからJMSを介してライブデータからメッセージを取得します。CCX Live Data Socket.IOサービスは、クライアントからのSocket.IO接続を介してReal-Time Reporting JSONを公開します。FinesseデスクトップにCisco Finesse通知サービスとのBOSH接続を維持するBOSHトンネルiFrameがあるのと同様に、マスターライブデータガジェットには、CCX Live Data Socket.IOサービスとのSocket.IO(WebSocket)接続を維持するSocket.IOトンネルiFrameがあります。

OpenAjax Hubは、登録されているリスナーにすべてのイベントを配信します。これは、ガジェットとFinesseコンテナ自体の一部の両方になります。Finesseデスクトップには、Cisco Unified CCX Notification ServiceとのBOSH接続を維持するBOSHトンネルiFrameがあります。これにより、イベントがOpenAjax Hubに発行されます。

## 参照リンク

- [Finesse Webサービス開発者ガイド](#)



## 翻訳について

シスコは世界中のユーザにそれぞれの言語でサポート コンテンツを提供するために、機械と人による翻訳を組み合わせて、本ドキュメントを翻訳しています。ただし、最高度の機械翻訳であっても、専門家による翻訳のような正確性は確保されません。シスコは、これら翻訳の正確性について法的責任を負いません。原典である英語版（リンクからアクセス可能）もあわせて参照することを推奨します。