

了解UCCX Finesse架构深入探讨

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[50,000英尺视图](#)

[Finesse Tomcat](#)

[HTTP\(S\)](#)

[XMPP](#)

[PUBSUB](#)

[BOSH -通过同步HTTP的双向流](#)

[CTI](#)

[JTAPI](#)

[30000 ft视图](#)

[休眠](#)

[AXL](#)

[SOAP](#)

[20000 ft视图](#)

[APACHE SHINDIG](#)

[WAR文件](#)

[10000 ft视图](#)

[AJAX - Finesse之美](#)

[使用AJAX的优点](#)

[AJAX的运行](#)

[使用AJAX向服务器发送请求](#)

[桌面体系结构](#)

[小工具架构](#)

[参考链接](#)

简介

本文档全面介绍了Finesse架构，以便在对Finesse问题进行故障排除时，使底层进程变得合理。

先决条件

要求

思科建议了解以下工具和功能：

JTAPI - Java电话API

API -应用编程接口

UCCX -统一联系中心快捷版

CUCM -思科统一通信管理器

CTI -计算机电话集成

使用的组件

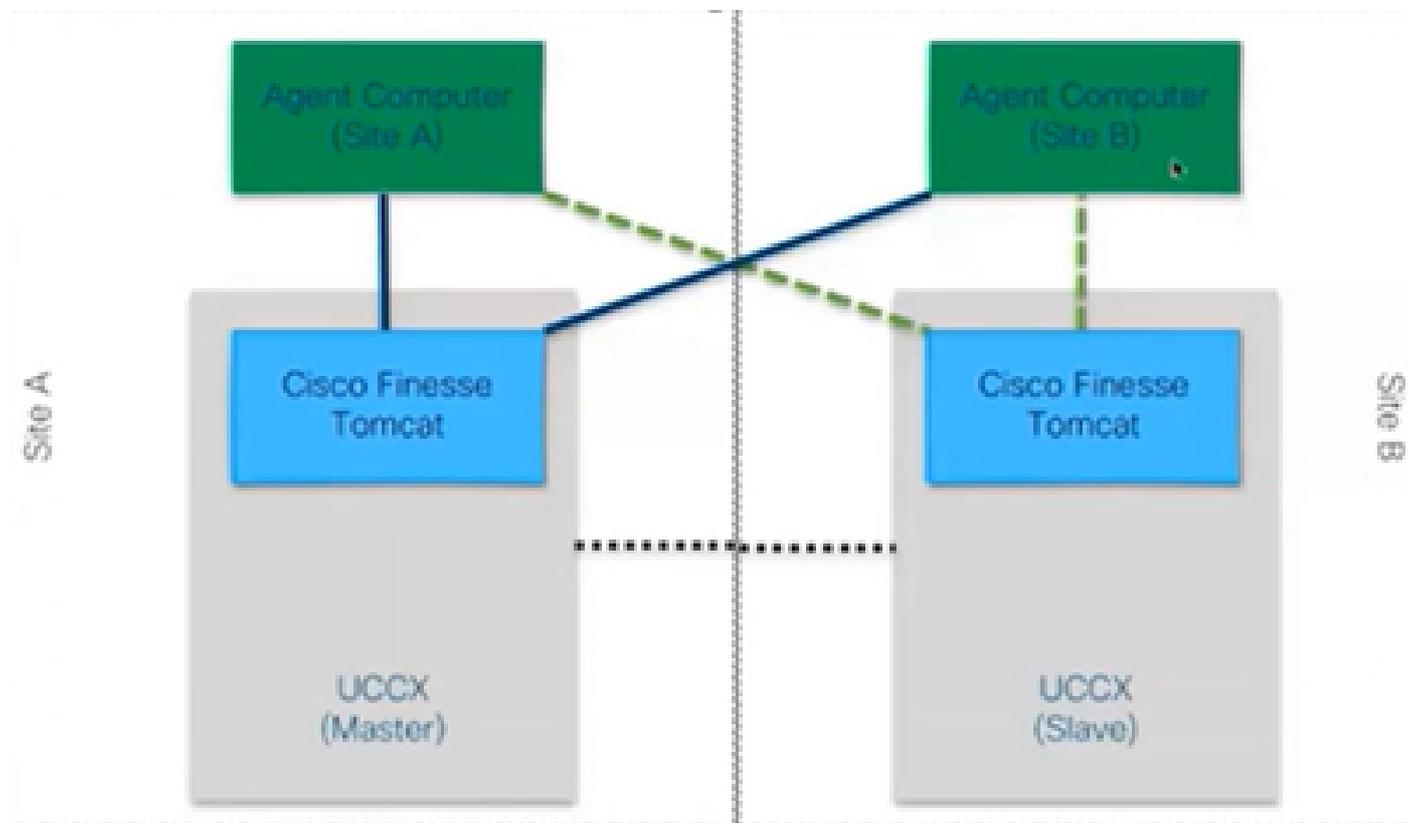
- 思科统一联系中心快捷版(UCCX)

本文档中的信息都是基于特定实验室环境中的设备编写的。本文档中使用的所有设备最初均采用原始（默认）配置。如果您的网络处于活动状态，请确保您了解所有命令的潜在影响。

背景信息

本文档从高级概述开始介绍Finesse架构，然后介绍深层信号流以及示例和图表。

50,000英尺视图



50000视图

Finesse Tomcat

Finesse Tomcat与CUCM中的Cisco Tomcat类似，因为加载网页的功能相同，但适用于不同的服务，称为Finesse。Tomcat是为Finesse设计的，因为它是一个单独的Web应用。根据11.5之前的版本，您只能使用CCX主节点登录finesse。从11.6开始，您可以登录到任何节点（但不推荐），仅在故障切换期间进行。因此，如果您考虑WAN集群，其中两个代理（站点A和站点B）都连接到主节点上的Finesse，则任何时候都会存在从Cisco Finesse到引擎的另一个节点的非活动连接(这是故障切换所需的CTI openconf请求)。

定期发送Openconf请求以检查节点是活动节点还是备用节点。如果发生故障切换，通知服务将使用称为systeminfo API的API，告知客户端此节点已关闭并且需要故障切换。然后运行一个文件，将浏览器重定向到另一个节点，然后发送openconf rejavascripresponse。仅当接受证书时，此故障切换才会运行。（建立到服务器的安全连接）。

提供了3个证书，它们是：

- 该节点的通知服务证书。
- 远程节点的通知服务证书。
- 远程节点的Finesse服务证书。



注意：需要接受远程节点证书，故障切换才能正常工作。

HTTP(S)

它是用于发送超媒体文档（例如HTML）的应用层协议。它是为Web浏览器和Web服务器之间的通信而设计的。它是一个无状态协议，表示服务器在两个请求之间不保留任何数据。这是客户端服务器协议。对于UCCX，Finesse客户端在代理浏览器(PC)上运行。它使用HTTP向服务器发出请求。以下是一些常见的HTTP方法：

GET -从服务器获取信息。

POST -向服务器发送信息。

PUT -

用于替换服务器上的任何内容。

DELETE -从服务器中删除信息。

Finesse使用http请求中的systeminfo api请求。例如，如果您想更改代理的状态，浏览器会发送PUT而不是POST，因为如果发送POST，服务器会混淆，因为它有两个状态，哪一个要选择？因此，使用PUT可以替换当前状态。

XMPP

可扩展消息传送和在线状态协议

它是一组用于即时消息和在线状态的协议。所有XMPP实体均使用其Jabber ID或JID进行识别。此XMPP协议用于小工具的扩展协议之一称为PUBSUB。

PUBSUB

它不是您能想到的发布者订户。它仍然发布和订阅，但与数据库无关。XMPP使用一种称为节点的机制。节点主要监控您关心的实体。任何对您来说重要并希望监控的内容，您都可以在其中添加一个节点。然后PUBSUB会执行什么操作，即订阅该节点上的更新或通知。可以为每种类型的实体（如座席、对话框等）设置节点。如果为代理创建节点，则会订阅该代理上的节点，然后无论代理执行什么操作，都会收到有关它的通知。

此规范的目的在于允许XMPP服务器（通知服务）获取发布到XMPP节点（主题）的信息，然后将XMPP事件发送到订阅该节点的实体。

对于Finesse，计算机电话集成(CTI)服务器将CTI消息发送到Finesse Web服务，以告知Finesse有关配置更新的信息，例如（但不限于）创建座席或联系服务队列(CSQ)或有关呼叫的信息。然后，此信息将转换为Finesse Web服务发布到Finesse通知服务的XMPP消息。然后，Finesse通知服务通过BOSH将XMPP消息发送到订用到某些XMPP节点的代理。

BOSH -通过同步HTTP的双向流

BOSH是长期HTTP连接，在该连接中，服务器将保留请求较长时间，直到收到响应，否则发送空响应。这适用于XMPP客户端和XMPP服务器，但也可用于非XMPP应用。



注意：XMPP是有状态的，而HTTP是无状态的（它不存储有关最后一个请求的信息）

如果Web应用需要与XMPP配合使用，则会出现多个问题。

问题1：浏览器本地不支持基于传输控制协议(TCP)的XMPP。

解决方案1：Web服务器和浏览器通过超文本传输协议(HTTP)消息进行通信，因此Finesse和其他Web应用将XMPP消息包装在HTTP消息中。

问题2：HTTP是无状态协议。

解决方案2：您可为此使用cookie/post数据。

问题3：第三个问题是HTTP的单向行为，这意味着只有客户端发送请求，服务器只能响应。服务器无法推送数据，导致无法通过HTTP实施XMPP。

解决方案3：要解决此问题，您需要在HTTP和XMPP之间建立网桥。

推荐的解决方案包括：

- 轮询（传统协议）：重复的HTTP请求，请求定义的新数据：HTTP轮询
- 长轮询也称为BOSH：传输协议，它通过高效使用多个同步HTTP请求/响应对来模拟两个实体之间长期、双向TCP连接的语义，而无需使用频繁轮询。使用BOSH的原因是为了掩盖服务器不必在请求发出后立即作出响应这一事实。响应会延迟到指定时间，直到服务器有客户端的数据，然后作为响应发送。客户端收到响应后，立即发出新请求，依此类推。

Finesse桌面客户端（Web应用）每30秒通过TCP端口7443建立过时的BOSH连接。30秒后，如果没有来自Finesse通知服务的更新，通知服务将发送一个带有200 OK和（几乎）空响应正文的HTTP应答。例如，如果通知服务更新了座席或对话（呼叫）事件的存在，则数据会立即发送到Finesse Web客户端。

综述：

Finesse Web客户端通过TCP端口7443设置到Finesse服务器的过时HTTP连接(http-bind)。这称为BOSH长轮询。Finesse通知服务是发布有关座席、呼叫等状态的更新的在线状态服务。如果通知服务有更新，它会使用状态更新作为HTTP响应正文中的XMPP消息来响应http-bind请求。如果在收到http-bind请求后的30秒内没有状态更新，通知服务会回复而没有任何状态更新，以允许Finesse Web客户端发送另一个http-bind请求。这样，通知服务就可以知道Finesse Web客户端仍然能够连接到通知服务，并且代理未关闭其浏览器或将其计算机置于休眠状态，等等。

CTI

您可以使用计算机电话集成(CTI)在拨打、接听和管理电话时充分利用计算机处理功能。CTI应用程序允许您执行诸如使用呼叫方ID从数据库检索用户信息之类的任务，或者使用交互式语音应答(IVR)系统收集的信息将来自用户的呼叫与其信息一起路由到相应的服务代表。CUCM上的CTI管理器响应来自UCCX的JTAPI请求。CTI服务器TCP端口为12018。这是Finesse服务器和引擎（CTI服务器）相互通信的方式。

以下是通过CTI交换的一些信息：

- 当前系统配置和未来更新。
- 座席及其状态。
- 呼叫及其状态。
- 实时座席、呼叫和队列统计信息。

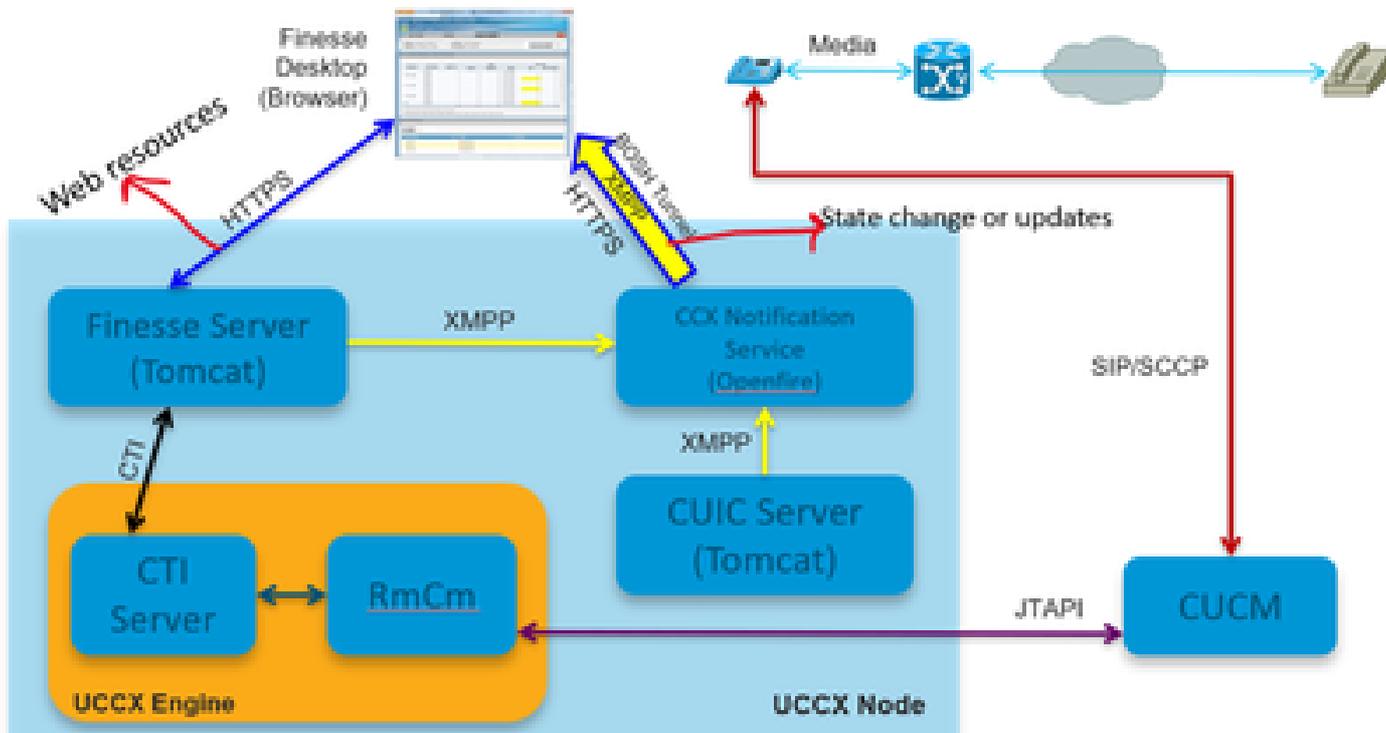
JTAPI

Cisco Unified JTAPI用作由Sun Microsystems开发的编程接口标准，用于基于Java的计算机电话应用。Cisco JTAPI通过其他Cisco扩展实施Sun JTAPI 1.2规范。UCCX和CUCM之间的任何通信都驻留在JTAPI上。这是CUCM和引擎（电话子系统）相互通信的方式。

JTAPI用于控制和监控CUCM电话、使用CTI端口和路由点路由呼叫、开始和停止在CUCM上的录音以及任何呼叫路由功能

30000 ft视图

下图说明UCCX引擎、Finesse、CUCM和浏览器如何相互通信。

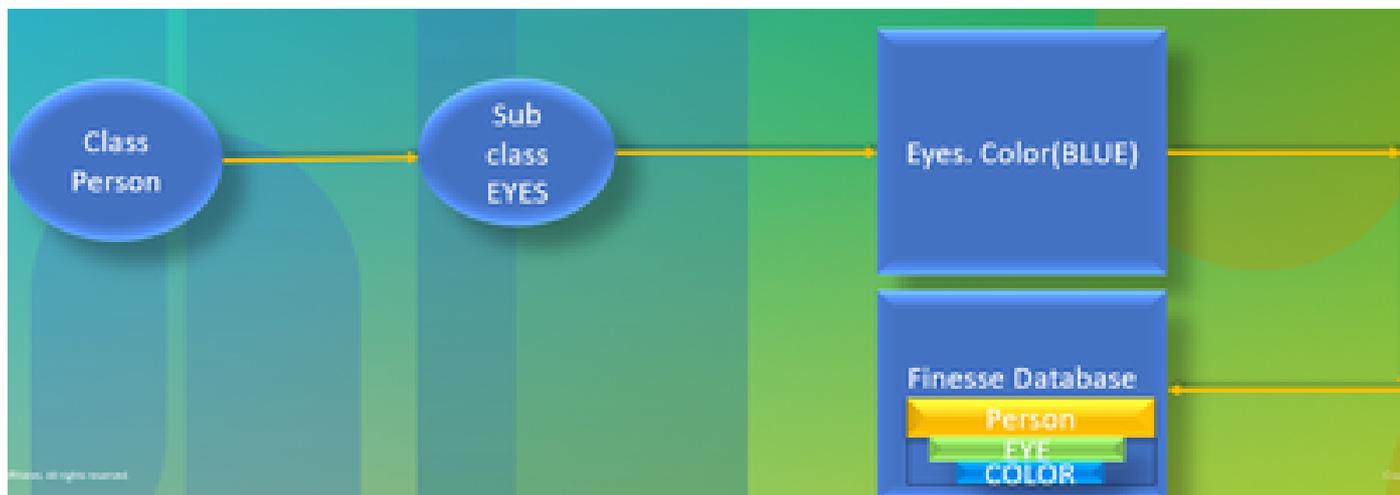


30000视图

让我们考虑一下，该呼叫是与座席建立的。现在，通过JTAPI监视代理扩展的RmCm向CTI服务器告知代理正在通话的状态更改。此信息通过CTI从CTI服务器 (CCX引擎内部) 发送到Finesse服务器(Tomcat)。Finesse服务器使用XMPP将此信息发送到CCX通知服务有关状态更改的信息。通知服务(Openfire)会打开一个到代理浏览器的BOSH隧道，并更新有关状态更改的信息，即您看到代理进入“保留”状态的方式。使用WAR文件、小工具等HTTPS向Finesse服务器请求任何类型的Web资源 (如果不在缓存中)。

休眠

下图说明了Hibernate服务。



休眠

HIBERNATE称为高性能对象/关系持久性和查询服务。简单地说，它将JAVA类映射到数据库表。例如，您有一个名为Team的JAVA对象，并且finesse数据库中有一个名为Team的数据库表。JAVA类控制表内的信息，而HIBERNATE则促成这一过程。它不使用SQL查询，而是使用java类来更新信息。

AXL

管理XML。

XML代表可扩展标记语言，是一种标记语言，用于定义一些相对简单的编码数据规则。它的主要设计是采用两个系统都能理解的明确定义格式传输和接收结构化数据。在最基本的格式中，XML定义了包含在尖括号(<>)中的标记，这些标记包围了该标记所描述的数据。标签可以与其他标签内的标签形成层次结构。例如，要定义基本电话设备，您可以说电话设备需要三个参数：名称、说明和电话号码。

它是基于SOAP的API，可实现CUCM上的远程调配。它用于添加、更新、删除或检索CUCM数据库中的信息。检索功能包括检查用户身份验证和运行SQL查询。AXL API为您提供对整个CUCM数据库的访问权限。AXL API纯粹用于调配，不提供对运行时或性能数据的访问。

AXL API采用HTTPS基本身份验证。任何CUCM用户（应用或最终用户）如果是标准CCM超级用户访问控制组或具有分配的标准AXL API访问角色的任何组的成员，则可通过AXL进行读/写访问。这意味着所有超级用户帐户已隐式地拥有AXL API的访问权限。要创建专用于AXL API使用的帐户，您必须首先创建一个访问控制组并向其分配标准AXL API访问角色，然后将应用程序用户与新创建的组关联。要提供只读AXL API访问，您可以创建一个单独的访问控制组，并仅向其分配标准AXL只读API访问角色。

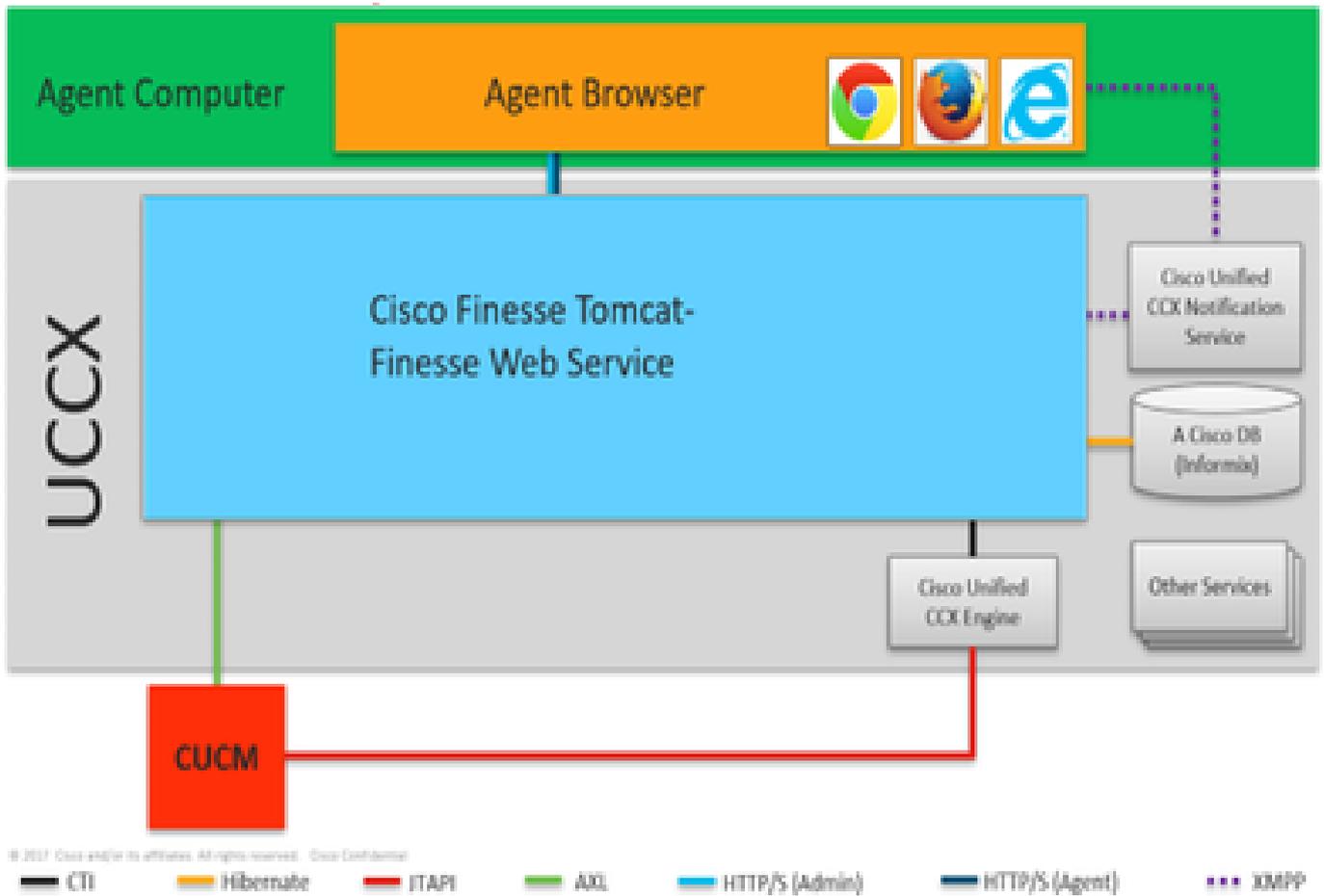
SOAP

简单对象访问协议(SOAP)是一种以XML格式在应用程序之间传递信息的方式。SOAP消息通常通过HTTP会话从发送应用程序传输到接收应用程序。实际的SOAP消息由Envelope元素组成，该元素包含一个Body元素和一个可选的Header元素。

- Envelope - 此强制元素是SOAP消息的根，表示传输的XML是SOAP数据包。信封包含正文部分和可选信头部分。
- Header - 此可选元素提供指示消息处理信息的扩展机制。例如，如果使用邮件的操作需要安全凭据，则这些凭据必须包含在信封信头中。
- Body - 此元素包含消息负载，即在发送和接收应用程序之间传输的原始数据。正文本身可由多个子元素组成，XML架构通常定义此数据的结构。

20000 ft视图

下图详细解释了Finesse架构中涉及的协议。

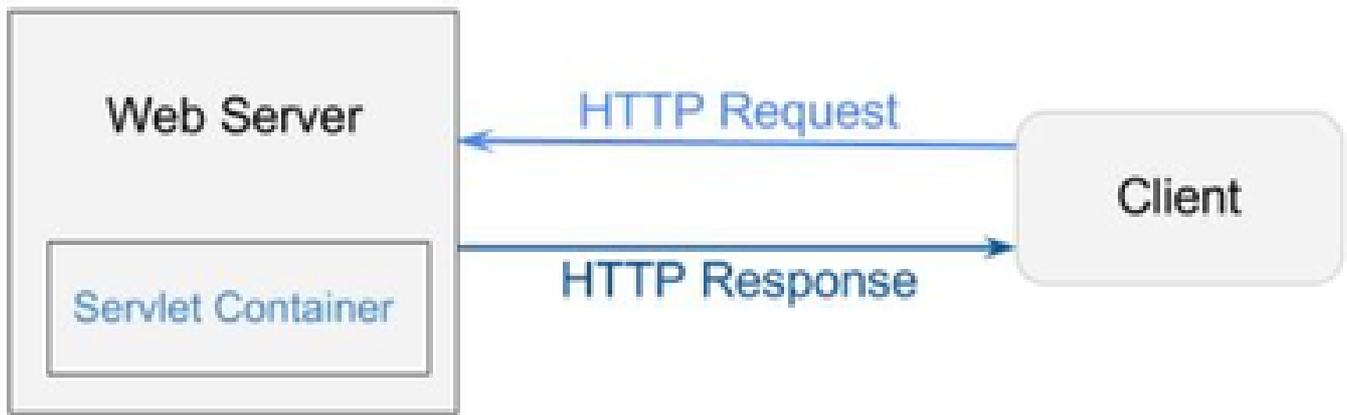


20000视图

这些协议负责不同UCCX组件之间的通信。

- UCCX引擎和Finesse服务器通过CTI协议进行通信。
- UCCX引擎和CUCM JTAPI通话协议。
- Finesse Tomcat和CUCM通过AXL进行通信。
- Finesse通知服务和座席浏览器在XMPP/BOSH上通话。
- Finesse Tomcat和数据库通过Hibernate进行通信。
- Finesse Tomcat和Finesse通知通过XMPP进行通话。

APACHE SHINDIG



Shindig

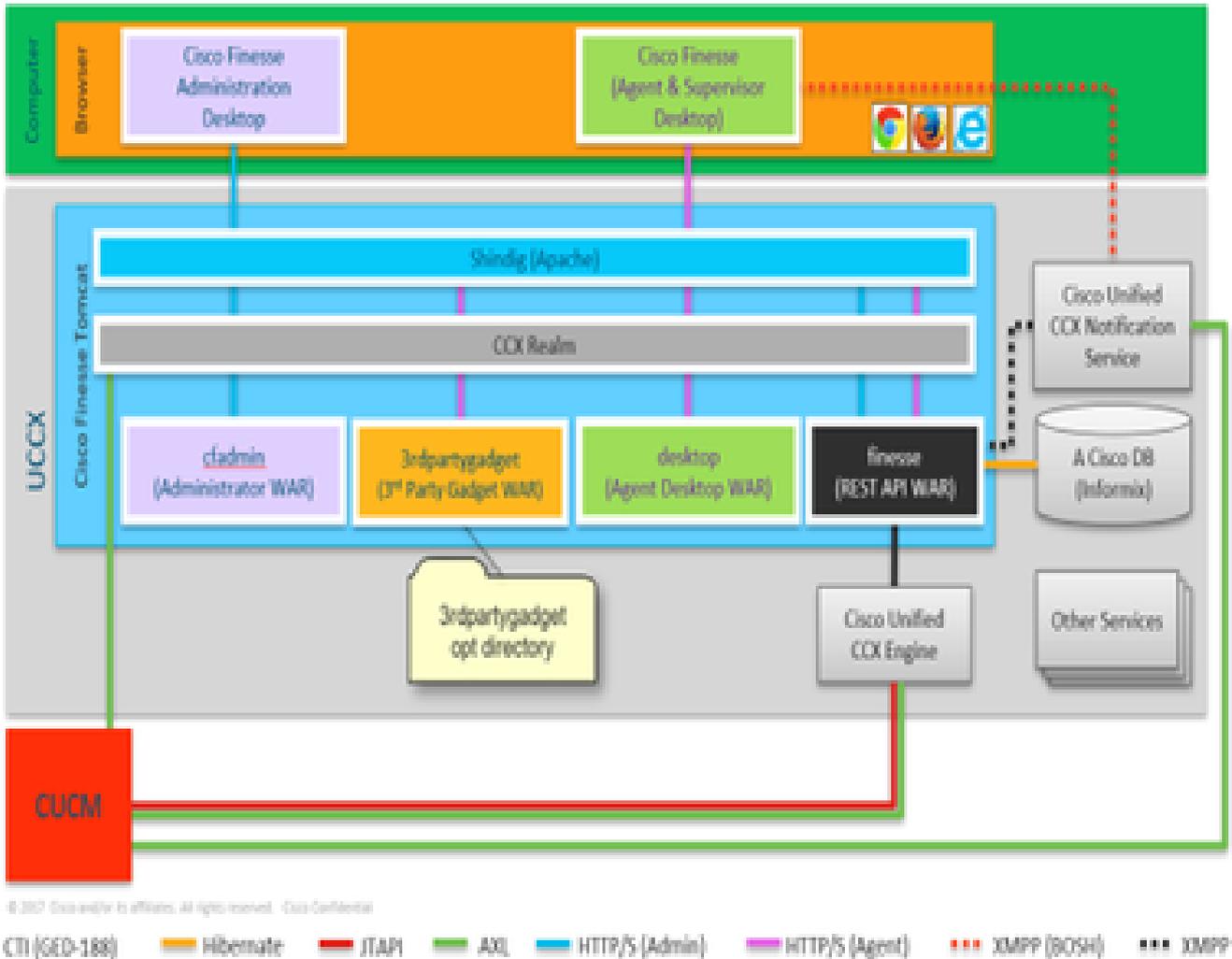
Apache Shindig是一个OpenSocial容器，通过提供用于呈现小工具、代理请求以及处理REST和RPC请求的代码，帮助您开始快速托管OpenSocial应用。OpenSocial是一组API，用于构建在Web上运行的社交应用。(Web/Servlet)容器由Web服务器用于动态生成Web页。

WAR文件

WAR代表Web Archive。它包含Web项目的文件。它可以有servlet、XML、JSP、image、HTML、CSS、JS等。Catalina日志包含有关正在部署的WAR的信息。

10000 ft视图

下图详细说明了身份验证流程在UCCX和Finesse组件中的工作方式。



10000视图

根据您登录的方式，需要使用WAR文件来显示和创建页面。浏览器要求Shindig需要渲染小工具，然后与CUCM对话以渲染小工具。CCX领域用于使用AXL对CUCM进行身份验证。通知服务还使用AXL通过CUCM进行身份验证。

Finesse Rest API WAR是实际与通知服务、CCX引擎和数据库通信的主存储库。Shindig仅与Finesse Rest API (WebServices)对话，因为cfadmin和桌面WAR只是用于显示页面。任何与Finesse Rest API WAR相关的内容，您都可以在Finesse WebServices日志中看到，这些日志是finesse的最重要日志。您可以在Shindig和Finesse Web服务之间使用HTTP通信(Rest API WAR)。Finesse Web服务(Rest API WAR)和引擎通过CTI相互通信。

AJAX - Finesse之美

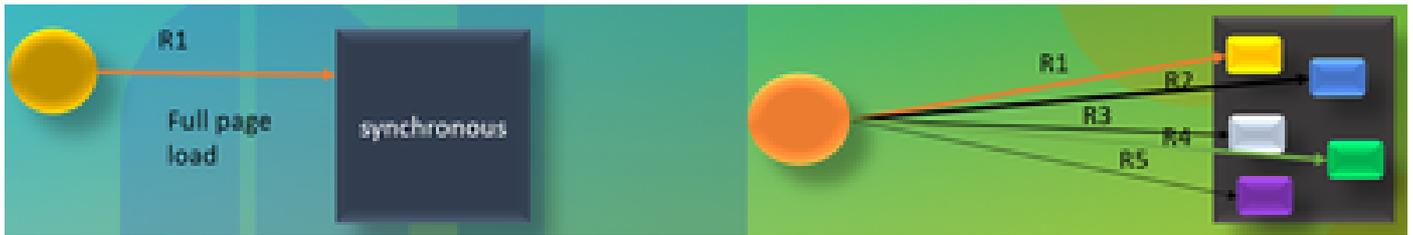
AJAX表示异步Javascript和XML。它不是一种编程语言，而是一种从网页访问Web服务器的方法。AJAX是一种进行部分页面更新的机制。它使您能够使用来自服务器的数据来更新页面的各个部分，而无需刷新整个页面。

例如，如果您谈论Facebook Messenger，收到新消息时，您无需刷新整个页面即可收到该消息，相反，该页面的消息部分会自动刷新，并实时获取新消息，而无需刷新整个页面。

每个浏览器都有一个称为XMLHTTPREQUEST(也称为XHR)的内置对象。对服务器中AJAX的每个请求都会经过此XML请求。其中包含需要更新的内容的详细信息。

使用AJAX的优点

下图说明了异步请求和同步请求之间的区别。



AJAX

如果是同步请求，必须等待第一个请求得到处理，然后可以发送第二个请求。例如，页面刷新是必需的，在页面刷新之前，您不能执行任何操作。另一方面，如果是异步请求，则无需等待第一个请求完成即可发送第二个请求。您可以同时发送多个请求。例如，网站上的天气应用小工具。您可以刷新页面的天气部分，同时也可以同时使用网站的其他部分，而无需刷新整个页面。这是异步请求的主要优点。

AJAX的运行

AJAX是用于从Web服务器发送和接收更新的XMLHTTPREQUEST (XHR)与用于显示或使用数据的Javascript和HTML的组合。

使用AJAX向服务器发送请求

下面将介绍一个3步流程：

1. 创建变量并将XHR对象存储到变量中。

```
Var 请求=新XMLHttpRequest();
```

2. 访问在XHR对象中具有负载的请求变量。

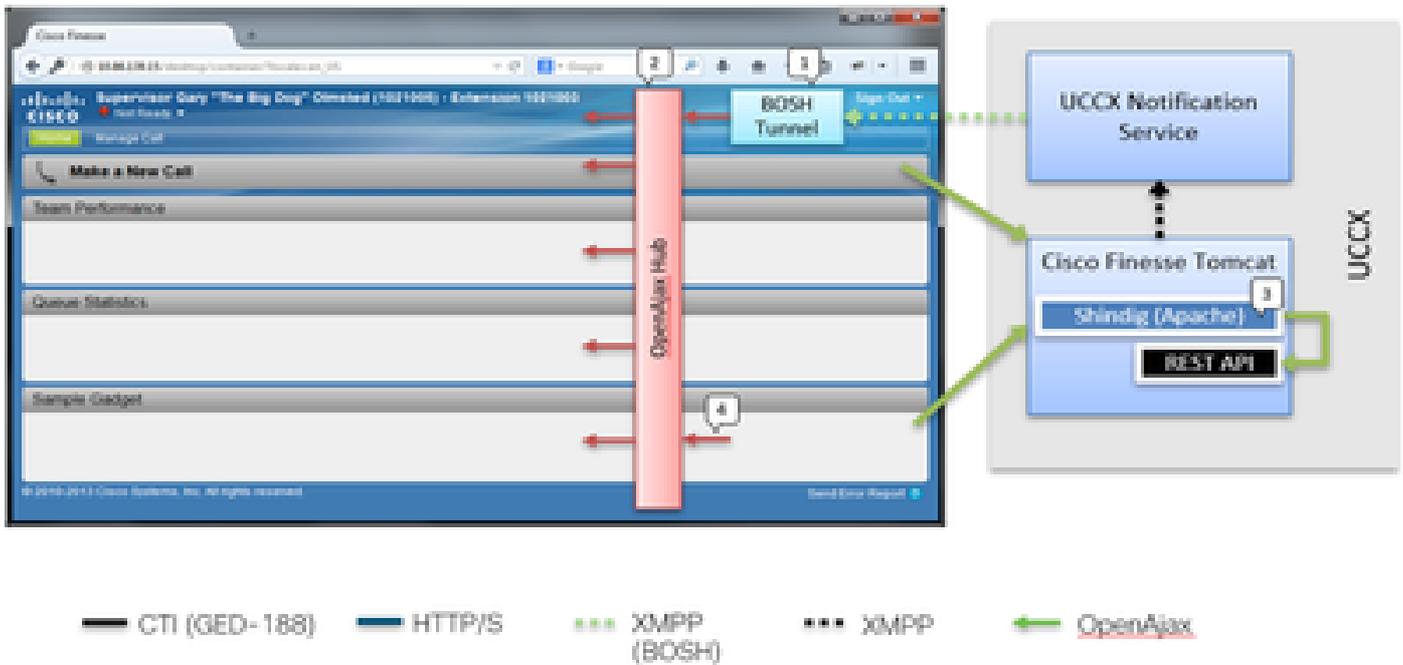
```
请求.open(GET, URL);
```

3. 发送请求

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

桌面体系结构

下图说明了小工具在网页上呈现时AJAX信号的流程。

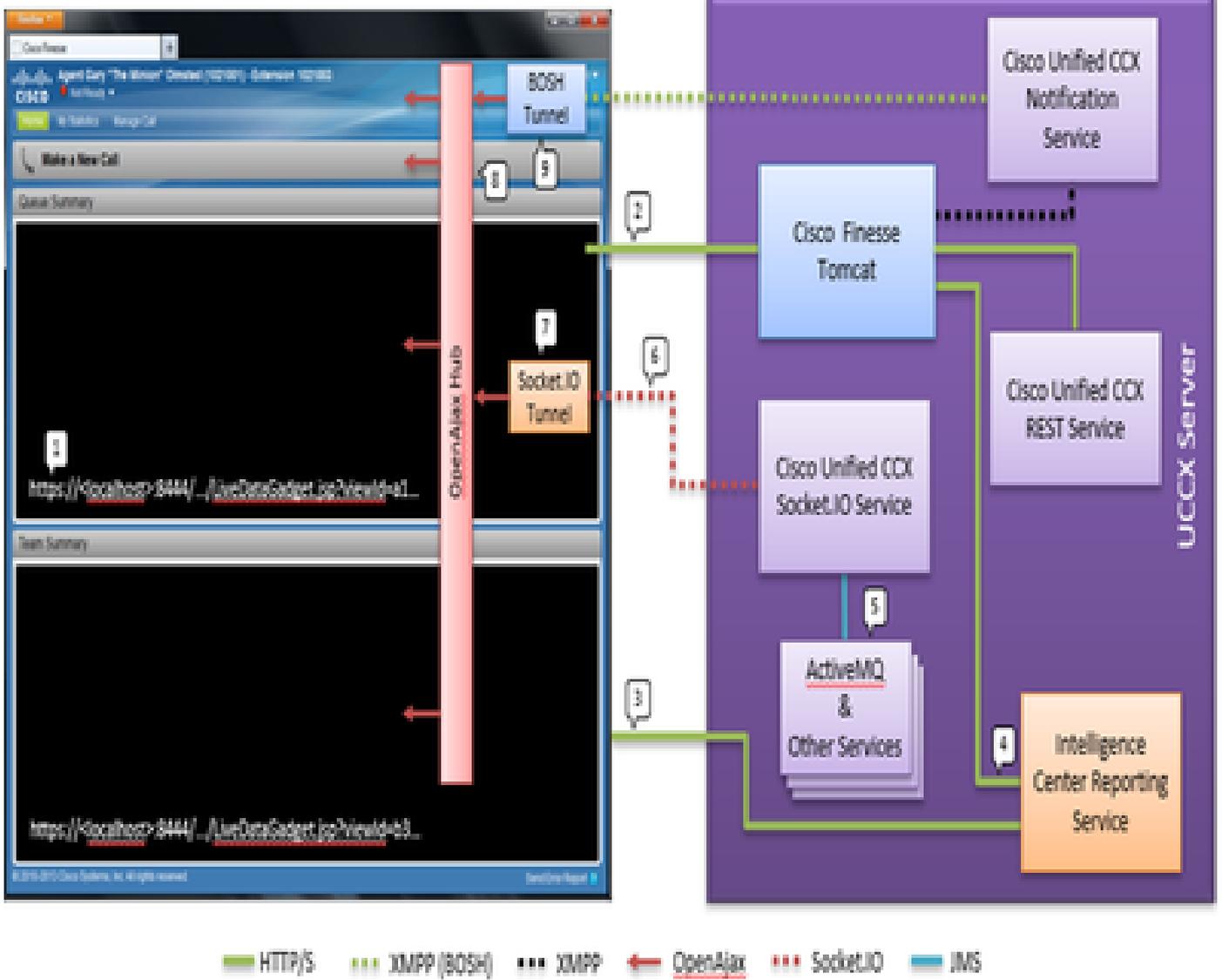


桌面体系结构

IFrame驻留在托管BOSH隧道的容器中。提供OPENAJAX集线器以跨小工具发布消息（使用pubsub方法），REST请求也通过Shindig代理到其他服务器。小工具可以在AJAX中心上发布其自己的消息。

小工具架构

下图详细解释了Finesse小工具架构。



小工具体系结构

与典型的小工具不同，CUIC小工具还接收来自OpenFire的实时XMPP源。在UCCX中，CUIC和Finesse与UCCX共存，则有一个共享OpenFire实例。大多数小工具内容和所有REST API都是通过Finesse服务器中的Shindig代理的。这适用于Finesse小工具和REST API以及CUIC小工具实例和REST API。CUIC小工具使用D-Grid来呈现其报告。必须执行引导过程，此操作直接与CUIC一起完成。因此，CUIC小工具在加载过程中最初直接与CUIC服务器通信。因此，用户浏览器中必须接受CUIC证书（除Socket.IO隧道外）。小工具内容和REST API在Finesse和CUIC之间代理到客户端。Rest API调用将同时发送到Intelligence Center Reporting Service和CCX Web Service。CCX实时数据Socket.IO服务通过JMS从ActiveMQ获取实时数据的消息。CCX实时数据Socket.IO服务通过来自客户端的Socket.IO连接发布实时报告JSON。与Finesse Desktop具有用于维护与思科Finesse通知服务的BOSH连接的BOSH隧道iFrame类似，主实时数据小工具具有用于维护与CCX实时数据Socket.IO服务的Socket.IO (websocket)连接的Socket.IO隧道iFrame。

OpenAjax中心将所有事件分发到订阅的侦听程序。这既可以是小工具，也可以是Finesse容器本身的一部分。Finesse桌面有一个BOSH隧道iFrame，用于维护与Cisco Unified CCX通知服务的BOSH连接。这会将事件发布到OpenAjax中心。

参考链接

- [Finesse Web服务开发人员指南](#)

关于此翻译

思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。