Oracle® Communications
Evolved Communications Application Server

Concepts

Release 7.1

E66294-01

May 2016

Oracle Communications Evolved Communications Application Server Concepts, Release 7.1

E66294-01

Contents

4 About Change Management

5 About Managing and Using Subscriber Data

6 About the OCECAS Architecture and Environment

7 About Network Function Virtualization

8 About Session Control

Glossary

Preface

This book presents basic Oracle Communications Evolved Communications Application Server (OCECAS) concepts.

Audience

This document is intended for anyone who uses Oracle Communications Evolved Communications Application Server.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Accessing Oracle Communications Documentation

OCECAS documentation is available from the Oracle Documentation Web site: `http://docs.oracle.com`.

Related Documents

For more information, see the following OCECAS documentation:

- *Oracle Communications Evolved Communications Application Server Release Notes*
- *Oracle Communications Evolved Communications Application Server Installation Guide*
- *Oracle Communications Evolved Communications Application Server System Administrator's Guide*
- *Oracle Communications Evolved Communications Application Server Operator's Guide*
- *Oracle Communications Evolved Communications Application Server Security Guide*
- *Oracle Communications Evolved Communications Application Server Compliance Guide*

- *Oracle Communications Evolved Communications Application Server RESTful API Reference*
- *Oracle Fusion Middleware 12c Documentation Library*
- *Oracle Database Installation Guide 12c Release 1 (12.1) for Linux*
- *Oracle Database Administrator's Guide 12c Release 1 (12.1)*

1

About Oracle Communications Evolved Communications Application Server

This chapter provides an overview of the Oracle Communications Evolved Communications Application Server (OCECAS) product.

See "Glossary" if you are unfamiliar with the telecommunications terms or acronyms used in this document.

About OCECAS

You use OCECAS to provide SIP multimedia services to your IMS subscribers. It contains a Voice over Long Term Evolution (VoLTE) and WiFi-based VOIP (VoWiFi) application that you can modify and expand to define the services that you offer to your subscribers.

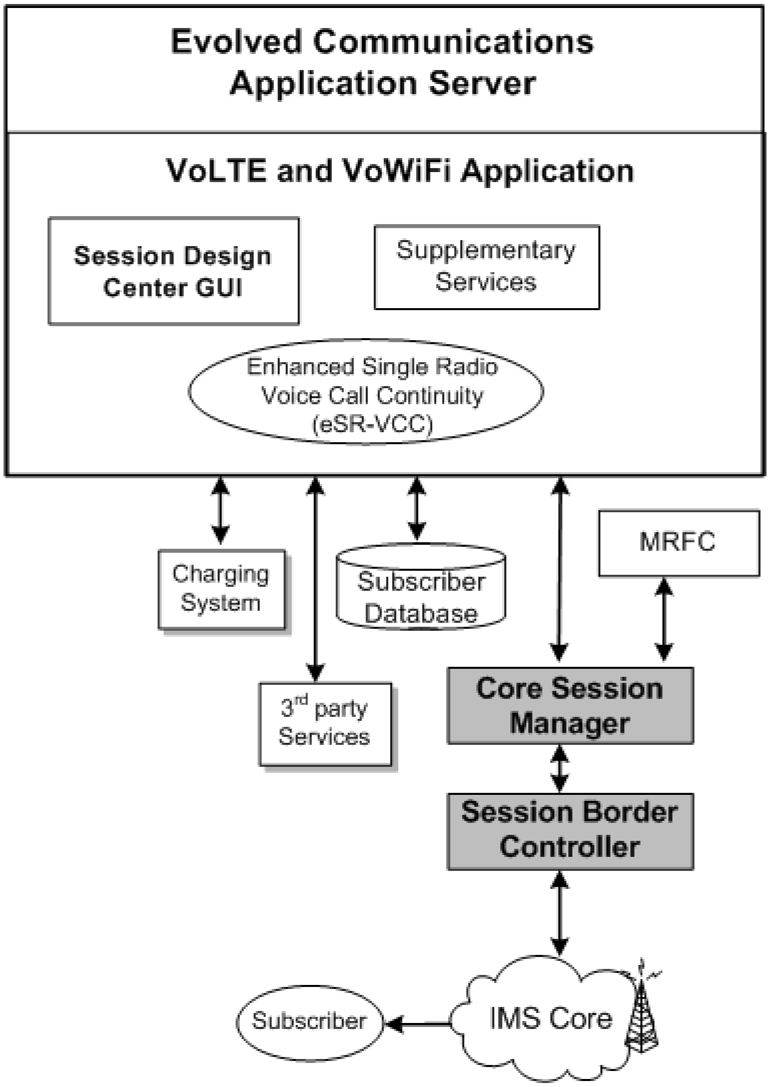
The integrated suite of features, includes:

- Voice call continuity. This feature ensures that a multimedia phone call can continue when a subscriber moves between LTE and 3G/2G coverage. OCECAS offers single radio voice call continuity (SRVCC) functionality.
- Service centralization and continuity. This feature ensures that a multimedia phone call transitions smoothly from packet-switched equipment to circuit-switched equipment. OCECAS acts as a service centralization and continuity server (SCC-AS) that facilitates single radio voice call continuity (SRVCC).
- Supplementary services. These features provide out-of-the-box, standards-compliant, integrated, multimedia application services for subscribers to use in multimedia calls. They include capabilities such as Communication Forwarding, Barring, Hold, Identification, and Ad-hoc Conferencing.
- An easy-to-use GUI interface named *Session Design Center*. You use Session Design Center to modify SIP call behavior and application service (session control) logic. Session Design Center provides you with graphical (drag and drop) control over the workflow from creation through to testing, staging, and ultimately production.
- Powerful change management. OCECAS uses a powerful, scalable, flexible, and secure architecture pre configured to package multiple service changes into a single unit that is easy to manage. This functionality includes the *production pipeline*, a fine-grained, flexible architecture that you use to develop, test, and finally deploy multimedia services to your subscribers.
- A back-to-back user agent (B2BUA) capability. This feature enables OCECAS to control a SIP call from both the originating and terminating ends.

- Fine-grained authorization of user actions through Oracle Platform Security Services, which provides security to Oracle Fusion Middleware including WebLogic Server.

Figure 1–1 shows an overview of the OCECAS architecture and how it fits into your network. OCECAS includes the VoLTE and VoWiFi Application, the Session Design Center GUI, Supplementary services, and provides enhanced single radio voice call continuity (eSRVCC) functionality. OCECAS connects to your charging system, 3rd party services, your subscriber database, and Oracle Communications Core Session Manager/Session Border Controller. Session Border Controller interfaces with the rest of your IMS core.

Figure 1–1 OCECAS Network Architecture


Evolved Communications Application Server
VoLTE and VoWiFi Application
Session Design Center GUI
Supplementary Services
Enhanced Single Radio Voice Call Continuity (eSR-VCC)
Charging System
Subscriber Database
MRFC
3rd party Services
Core Session Manager
Session Border Controller
Subscriber
IMS Core

For additional information about OCECAS services, see "About OCECAS Services".

About Service Control Continuity and Voice Call Continuity

This feature enables a subscriber call to continue uninterrupted when it moves from a part of your network serviced by VoLTE packet-switched equipment to a part of your network serviced by legacy circuit-switched equipment with no call degradation.

OCECAS functions as an SCC-AS to provide SRVCC capability. The SRVCC capability is a VoLTE/VoWiFi functionality that allows IMS networks to transition a voice call from the VoIP/IMS packet-switched domain to a circuit-switched domain. OCECAS uses the Session Border Controller and Core Session Manager software products to provide the signaling controller (Access Transfer Control Function or ATCF) and media anchor point (Access Transfer Gateway, or ATGW) capabilities required for SRVCC capability.

About the VoLTE and VoWiFi Functionality

OCECAS implements the *IR.92 IMS Profile for Voice and SMS* specification for multimedia calls. This specification lists the set of 3GPP multimedia features that a wireless device and network must implement. Implementing this specification guarantees that your customers can use an interoperable, high-quality IMS-based telephony service over Long Term Evolution (LTE) radio access. The IR.92 specification defines the required features as *supplementary service*s.

About XCAP and the Supported Supplementary Services

OCECAS includes supplementary services that allow you to provide multimedia capabilities that your subscribers expect. OCECAS includes the supplementary services listed in "About Using VoLTE Supplementary Services".

OCECAS supports configuration of supplementary services from the subscriber's device (or user equipment (UE)), using the XML configuration access protocol (XCAP). The UE generates the configuration query or request, which is sent over HTTP/1.1 to OCECAS, which processes the request. The URI indicates the document or part of the document to be accessed. The HTTP requests trigger control flows to process requests, allowing designers to take alternate actions based on exchanged data. See "About Applications" for a description of the HTTP control flows.

For a description of the XCAP communication flow, see Figure 6–1 in "About the OCECAS Environment" and "Understanding How OCECAS Communicates".

About the Session Design Center UI

OCECAS includes the VoLTE and VoWiFi multimedia application that you modify to provide services for your subscribers. This application uses the session initiation protocol (SIP) signaling protocol to determine multimedia call behavior.

You use the Session Design Center UI to create the SIP session logic for calls, and to add functionality in the form of subscriber services. This UI is designed so that service designers no longer need the complex coding knowledge to create and change SIP applications. Staff familiar with your IMS network can rapidly and easily create new call flows with decision forks based on subscriber profiles. You set up the session logic based on subscriber information and input. For example, you could set up a different session logic (call experience) based on:

- The current account state.
- UDR information, such as the account location.
- SIP Session information, such as information from a SIP message header.
- External factors, such as time-of-day or date.
- A response from an external system (such as the results of a web service query).

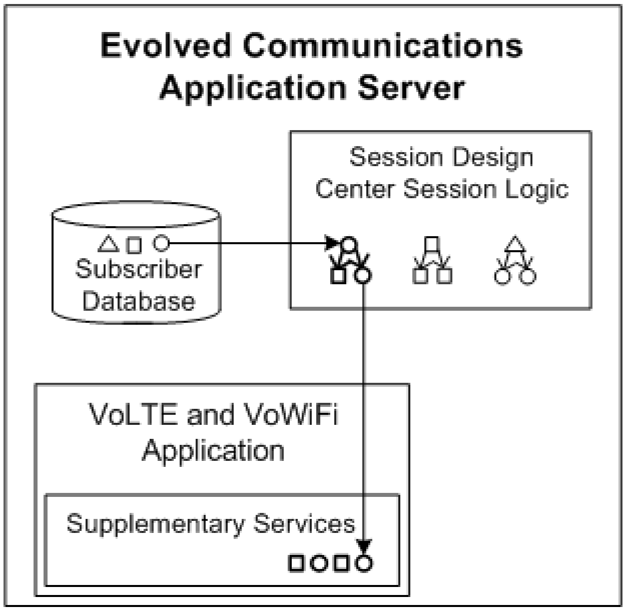
Session Design Center is a service assembly environment (SAE). The sessions take the form of a flowchart of decisions and activities.

The Session Design Center offers these features:

- The Control Flow Editor, an intuitive and powerful GUI-oriented drag-and-drop environment for controlling sessions. You can build flows to manage your sessions from start to finish. You use this tool to identify key resources, create triggering criteria to use in session logic, and create hierarchical component services.
- Powerful change management features that allow you to deploy new services and features to your production pipeline. You can deploy changes to the testing, staging, and production environments with a simple drag-and-drop action. If the changes are no longer required, you remove them just as easily.
- Resource management. You can reference announcements, create white and black lists of resources, and create templates for notifications and web services.
- A Deployment view that you use to manage changes across a pipeline of multiple environments, such as Testing, Staging, and Production.

Figure 1–2 shows how Session Design Center works with your subscriber database to apply supplementary services to a multimedia call. You first use the Session Design Center to create session logic. Then when a call requires a supplementary service, it automatically selects the session logic based on the subscriber characteristics you specify. The session logic then specifies the supplementary service to use.

Figure 1–2 OCECAS Session Logic and Multimedia Features


Evolved Communications Application Server
Session Design Center Session Logic
Subscriber Database
VoLTE and VoWiFi Application
Supplementary Services

About Change Management and the Production Pipeline

OCECAS provides you with a low-risk, solid, predictable service and change deployment structure to control changes to your implementation. The production pipeline itself uses these stages:

- A testing environment that you use to develop and test services.
- A pre-release (staging) environment that you use to validate services.
- A production environment that your subscribers use to access your services.

The change management platform enables you to combine features into a single entity called a *change set*. Change sets can include changes to control flows, announcements, number sets, configuration data, and so on. Change sets do not contain any data external to OCECAS, and do not contain any subscriber data.

After you create a change set, you deploy, roll back, or delete all of the changes it contains with a single action. A change set could define a subscriber service, configuration changes, or other modifications that you manage as a single unit. Using this system, you can:

- Design new services (or modify existing services) as a collection of related changes to different entities.
- Test a group of changes as a single entity.
- Quickly and easily migrate the service, configuration changes, or other changes from your development system to your staging system, and then to your production system.
- Roll back the set of related changes with a single action.
- See a complete history of a set of related changes. The history is often useful for auditing.

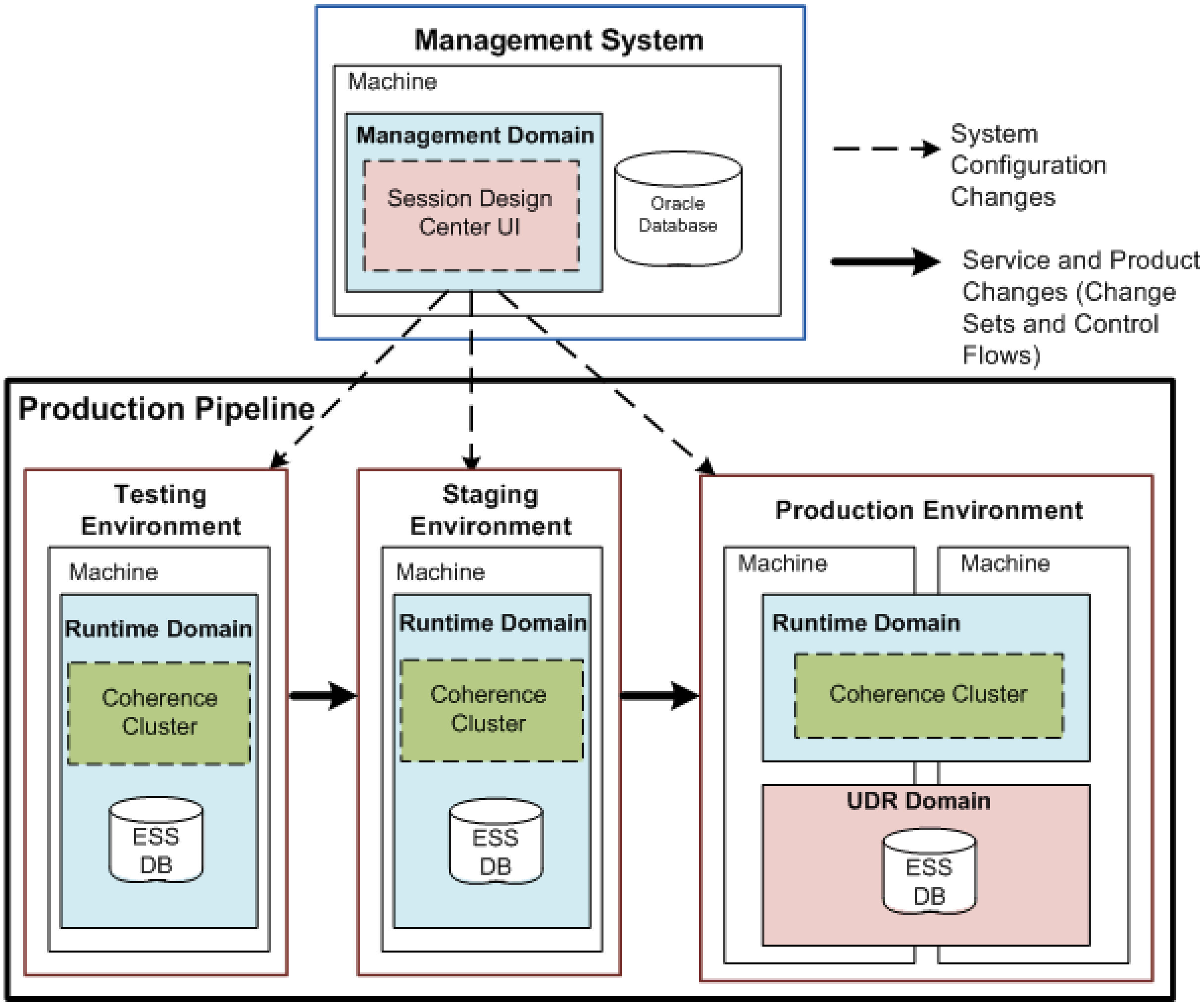
Figure 1–3 shows an overview of the OCECAS production pipeline that you use for change management. The OCECAS Management System at the top contains a system with a special Oracle Fusion Middleware WebLogic domain. This is a special domain that you use to manage the other OCECAS domains. You push configuration changes from the Management System to the systems in your production pipeline.

The lower part of the diagram shows the production pipeline with its series of three OCECAS systems. Each of the systems runs on its own WebLogic machine. The machines could be real or virtual. The system on the left is a testing system for creating new services. The middle system is for staging (verification) services. And finally, the system on the right is the production system where your subscribers can access your services. The arrows show change sets moving from the testing system to the staging system, and finally to the production system. The arrows represent new session behavior or products that you test, then stage, and then finally offer to your subscribers.

All three systems include WebLogic domains, coherence clusters, OCECAS session information, and internal user data repository (UDR) databases. The UDRs in this diagram are internal databases required for OCECAS data. You can locate this domain inside runtime domains (as shown in Testing and Staging), or use a dedicated UDR domain to store it (as shown in Production). This internal UDR is different from the home subscriber server (HSS) that you use to store subscriber information. However, connections to the internal UDR and to your external HSS are defined in the same file. The connection to your HSS is not shown in this diagram.

Each The production system is shown using two clustered machines to take advantage of the OCECAS high availability features.

Figure 1–3 Evolved Communications Application Server Change Management Architecture



Note: A user with access to the Session Design Center UI also has permission to make pipeline changes. So only give the Session Design Center credentials to trusted personnel.

About SIP Call Control (Back-to-Back User Agent)

The OCECAS server includes a SIP back-to-back user agent (B2BUA), which means it sets up and controls a SIP call from both the originating and terminating ends. OCECAS controls the entire call in two different portions, and allows you to add value-added services to one or both of the call portions. For example, you can:

- Add new legs to a conference call
- Redirect the call to a different endpoint
- Charge for the call, using on-going session authorization (time-based) or by individual service (service-based)
- Add any interaction involving media servers, such as announcements
- Perform call hunting
- Perform or change call failure handling

- Perform basic location and mobile state session control

See "About Back-to-Back User Agent" for details about the default back-to-back user agent capabilities.

About SIP Chaining

SIP chaining is a standards-based mechanism that enables OCECAS to co-exist with other SIP servlets, application servers, and applications, including applications that reside on different nodes.These other application servers and applications can often perform specialized tasks.

About SIP Chaining with Other Application Servers

When OCECAS is deployed with other application servers, the Serving Call Session Control Function (S-CSCF) is responsible for routing traffic between them. In the OCECAS environment, Oracle's Control Session Manager (CSM) is the S-CSCF.

IMS application servers receive SIP traffic by way of the S-CSCF based on Initial Filter Criteria (IFC) that is defined in the Application Server Subscription Information for the user being served. When the S-CSCF receives a SIP message, it requests the relevant IFCs from the HSS, examines them and, when a match is found, forwards the message to the specified application server. Application servers such as OCECAS, therefore, can be chained together through IFCs under the control of the S-CSCF. For information on the criteria that determines whether an application server can be invoked, refer to the 3GPP technical specification 23.218, version 12.3.0 specification, available from the "3GPP Specification detail" web site.

You must define IFCs to ensure that OCECAS receives SIP signalling from user endpoints (UEs) and third-party REGISTER requests for registration, re-registration, and de-registration. OCECAS must be informed of all such requests to maintain the registration expiry timestamp information that is held in the subscriber data store.

For information on specifying Initial Filter Criteria for OCECAS to co-exist with other application servers, see "About Configuring SIP Chaining" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About SIP Chaining with Other Applications

When other applications are deployed on the OCECAS engine nodes, the OCCAS Default Application Router is responsible for routing the SIP traffic between them.

When OCCAS receives an initial SIP request from an external entity, it invokes the application selection process as specified by the SIP Servlet 2.0 specification (JSR-000359) to select the first application to service the request. If the first application proxies the request or acts as a B2BUA and sends a new initial request, OCCAS again invokes the application selection process to invoke the next application. As applications proxy or send requests, they create a chain of applications that ends when an application acts as a user agent server (UAS) or when the request is sent to an external SIP entity.

OCCAS provides a Default Application Router (DAR) that acts as the SIP Application Router to implement the application selection process. The DAR is not configured in the default OCECAS installation. Instead, the OCECAS Session Control Framework SIP servlet is specified as the default SIP servlet to cause all traffic to be directed to OCECAS. To enable OCECAS to interface with other applications, you must modify the OCCAS SIP server configuration, after installation, to provide a DAR configuration that invokes OCECAS and any other applications, or a custom application router.

For information on specifying DAR configuration for OCECAS to co-exist with other applications, see “About Configuring SIP Chaining” in *Oracle Communications Evolved Communications Application Server System Administrator’s Guide*.

About Oracle Platform Security Services

OCECAS supports fine-grained authorization of user actions through the assignment of roles, which have associated permissions. OCECAS uses the underlying Oracle Platform Security Services (OPSS) framework to implement the authorization of user actions. OPSS provides security to Oracle Fusion Middleware, including WebLogic Server.

You initialize the OPSS repository during installation of OCECAS. For information on OCECAS application roles, see "Implementing Authorization and Permissions" in *Oracle Communications Evolved Communications Application Server Security Guide*.

2

About OCECAS Services

This chapter explains the functionality of VoLTE services that you create and manage on an Oracle Communications Evolved Communications Application Server (OCECAS) implementation.

About Using the Session Design Center GUI to Create Multimedia Services

You use the Session Design Center graphical user interface (GUI) to create the control flows that define the multimedia services that you sell to subscribes, and the change sets that you use to manage changes to your OCECAS implementation.

See "About the Session Design Center" for a thorough overview of the Session Design Center.

About Using SRVCC Functionality to Handover Multimedia Calls

Enhanced service centralization and continuity application server (SCC-AS) is an LTE functionality that allows you to transition a voice call from a packet-switched domain to a legacy circuit-switched domain. This functionality allows you to offer VoLTE capabilities while using networks that also includes circuit-switched coverage. OCECAS supports the SRVCC procedure for handover from packet-switched to circuit-switched networks. The handover requires access and transfer control function (ATCF) and access and transfer gateway (ATGW) functionality. This is specified in the 3GPP Release 10 specification for SRVCC, and most parts of the 3GPP Release 12 specification.

OCECAS provides this functionality in conjunction with Oracle Communications Session Border Controller (Session Border Controller). Session Border Controller acts as the ATCF and ATGW required for SRVCC. Session Border Controller is a different Oracle product that you purchase separately.

OCECAS itself acts as the service centralization and continuity application server (SCC-AS). It also provides home network anchoring, and terminating access domain selection (T-ADS).

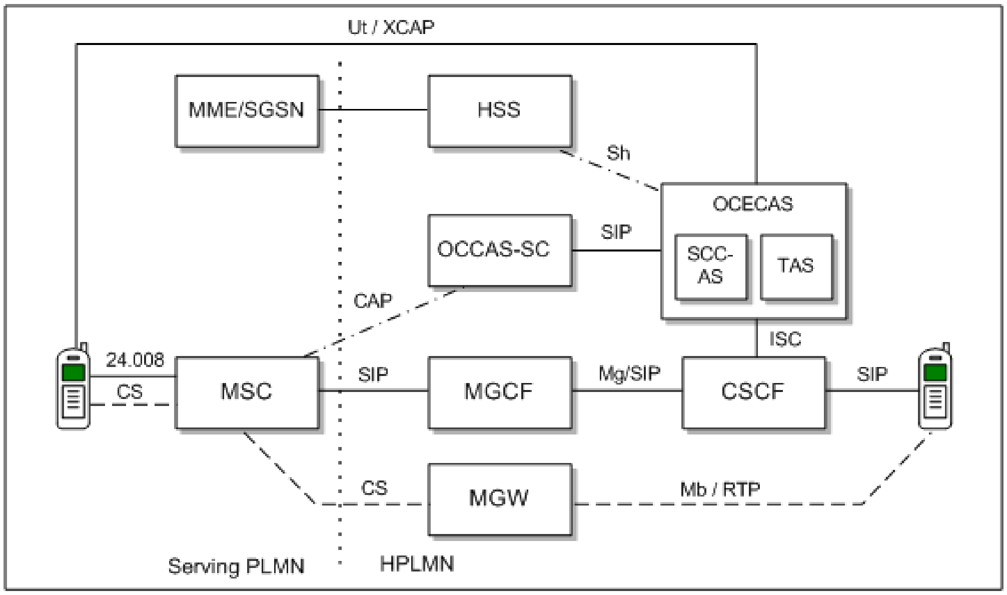
About Service Centralization

OCECAS acts as a Service Continuity and Centralization Application Server (SCC-AS) to deliver services, such as supplementary services, from the IMS network across different network technologies and user entities. By acting as a single service point, it is able to provide service centralization. In so doing, it assumes that the MSC is enhanced to support the use of Session Initiation Protocol (SIP) towards the IMS because this supports service centralization before, during, and after SRVCC hand-over. However, your MSC might not be enhanced for IMS centralized services (ICS) and roaming networks also are not necessarily enhanced for ICS. OCECAS, therefore, also uses a network-based solution that uses ISUP and CAMEL capabilities to support integration with an MSC that is not enhanced for ICS. OCECAS uses Oracle Communications Converged Application Server Server Controller (OCCAS-SC) to integrate with an MSC that is not enhanced for ICS and provide the following capabilities:

- Support for SIP messages that are mapped from CAMEL
- Access from within the control flow editor to all SIP messages and header information mapped from CAMEL
- Inherent IP Multimedia Routing Number (IMRN) facility
- Required service logic for inherent integration with OCCAS service controller for service centralization

Figure 2–1 illustrates how OCECAS woks together with OCCAS to support an MSC that is not enhanced for ICS.

Figure 2–1 Service Centralization Support for MSC Not Enhanced for ICS



For information about configuring OCECAS for an MSC not enhanced for ICS, as well related configuration options, see "Configuring Transfer" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About HSS-HLR Synchronization

OCECAS supports synchronisation of supplementary service (SS) data changes to both the Home Subscriber Server (HSS) and the Home Location Register (HLR). On receipt of a supplementary service change from an IMS user equipment (UE), OCECAS updates the subscriber profile in the HSS and updates the HLR. Similarly, on receipt of a supplementary service change notification from the HLR, OCECAS updates the subscriber profile in the HSS.

OCECAS integrates with OCCAS-SC to communicate with the HLR. Specifically, OCCAS-SC forwards MAP NoteSubscriberDataModified messages from HLR to

OCECAS, and OCECAS sends MAP AnyTimeModification messages to HLR by way of OCCAS-SC. MAP messages are encoded through XML Encoding Rules (XER-encoded) and passed in the body of SIP SUBSCRIBE or NOTIFY messages between OCECAS and OCCAS-SC.

For information on configuring HSS-HLR synchronization, see configuration settings under "Configuring Transfer" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About the Business Logic that Your Calls Can Use

OCECAS provides these tools in a drag-and-drop environment that you use to create VoLTE services:

- Make control flow decisions based on:
  - Date and time
  - Geographic location
  - Data in data stores
  - SIP message headers
  - Session context or call state
- Interact with other systems, including:
  - Media servers (JSR 309)
  - Charging systems (using Diameter Ro and Gy)
  - Messaging gateways (using SOAP or JSON)
  - Enterprise and Middleware (using SOAP, JSON, JMX, and SNMP traps)
- Perform Actions, including:
  - Manipulating the session state (end session or participant, hold participant, forward session, or add participant)
  - Early media and pre-call, in-call, or post-call announcements
  - Call hunting, call forwarding, and call barring
  - Using location services
  - Using cross-session context look-up
  - Using a charging trigger function
  - Using custom operations, such as raising alarms and branching on statistics
  - Execute other session logic
- Add responses to actions:
  - Manage call failure
  - Wait for and act on an event
  - Store information for later use

You create control flows in the Session Design Center UI to use these business logic features.

About Using VoLTE Supplementary Services

The supplementary services listed in the *IR.92 IMS Profile for voice and SMS* specification define the service capabilities required to continue a multimedia call that originated in a VoLTE network. Essentially, OCECAS becomes the SIP Back-to-back User Agent (B2BUA) for the call.

OCECAS supports these supplementary services:

- Anonymous Communication Rejection (ACR) (3GPP 24.611)
- Originating Identification Presentation (OIP) (3GPP TS 24.607)
- Terminating Identification Presentation (TIP) (3GPP TS 24.608)
- Originating Identification Restriction (OIR) (3GPP TS 24.607)
- Terminating Identification Restriction (TIR) (3GPP TS 24.608)
- Communication Forwarding Unconditional (CFU) (3GPP TS 24.604)
- Communication Forwarding on not Logged in (CFNL) (3GPP TS 24.604)
- Communication Forwarding on Busy (CFB) (3GPP TS 24.604)
- Communication Forwarding on not Reachable (CFNLc) (3GPP TS 24.604)
- Communication Forwarding on no Reply (3GPP TS 24.604)
- Incoming Call Barring of (ICB) (3GPP TS 24.611)
- Outgoing Call Barring (OCB) (3GPP TS 24.611)
- Barring of Outgoing International Calls (BOIC) (3GPP TS 24.661)
- Barring of International Calls - ex Home Country (BOIC-HC) (3GPP TS 24.611)
- Barring of Incoming Calls - When Roaming (BIC-Roam) (3GPP TS 24.611)
- Communication Hold (CH) (3GPP TS 24.610)
- Ad-hoc Conferencing (3GPP TS 24.605)

3

About the Session Design Center

This chapter describes the Session Design Center, which is the user interface to Oracle Communications Evolved Communications Application Server (OCECAS).

The Session Design Center contains an interface for creating and managing control flows. Control flows include the service logic and resources that you use to control a subscriber’s voice and video sessions.

You create and manage control flows through change sets, which are collections of configuration and control flow changes. Each change set is based on a *baseline* change set. You deploy change sets into your production environment through a production pipeline that consists of Testing, Staging, and Production environments.

The Session Design Center is divided into the following three areas of operation:

- **Application**

  Contains applications, all control flows and all resources

- **Projects**

  Contains all change sets

- **Environments**

  Contains the deployment pipeline, consisting of the Testing, Staging, and Production environments into which you can sequentially deploy change sets. Each environment also provides access to its service configuration.

From within these areas, you can return to the home page by clicking the home icon 🏠 in the upper left corner of the screen, below the product name.

About Application Configuration

From the Session Design Center home page, enter the Application area by clicking the **View Application Configuration** link. The Application Configuration contains the following three tabs:

- Application

  Lists the available applications and for each application that you select, lists the control flows that make up that application.

- All Control Flows

  Lists all control flows, including the control flows that have been delivered with OCECAS and any that have been added since.

- All Resources

Lists resource data and media resources that are available for a control flow to use.

About Applications

The VoLTE and VoWiFi application controls and manages voice calls and multimedia sessions over Long Term Evolution (LTE) and voice calls over Wi-Fi. You make changes to the VoLTE and VoWiFi application by creating or modifying control flows to define new services and behaviors. The VoLTE and VoWiFi application serves as the initial baseline change set.

The VoLTE and VoWiFi application consists of eight message control flows and five application control flows.

Message control flows are triggered when the application server receives one of the following HTTP or SIP messages: HTTP DELETE, HTTP GET, HTTP POST, HTTP PUT, SIP INVITE, SIP OPTIONS, SIP REGISTER, or SIP SUBSCRIBE.

The HTTP messages represent queries or change requests from the subscriber's endpoint for the configuration of supplementary services. The SIP messages are call session messages.

Receipt of any of these HTTP or SIP messages triggers a control flow of the same name. The HTTP control flows perform the query or request for the specified supplementary service. Table 3–1 describes the SIP message control flows.

Table 3–1 VoLTE and VoWiFi SIP Message Control Flows

Control Flow	Description
SIP INVITE	Analyzes the SIP INVITE headers and determines the originating or terminating criteria, retrieves the subscriber record and MVNO details, and invokes the Session Origination or Session Termination service.
SIP OPTIONS	Triggered when a SIP OPTIONS message is received during the transfer and negotiation of video sessions in the event of a hand over from a packet-switched to circuit-switched network.
SIP REGISTER	Analyzes the SIP REGISTER headers and stores both the registration expiry and contact header of the endpoint in the subscriber record. Manages the lifecycle of registration and supports Access Transfer, in the event of a handover from a packet-switched to a circuit-switched network
SIP SUBSCRIBE	Analyzes the SIP SUBSCRIBE headers and determines which service flow to run to start and manage the subscription session.

Table 3–2 describes the five SIP application control flows, which manage the following stages of a call session:

Table 3–2 VoLTE and VoWiFi Application Control Flows

Control Flow	Description
Session Deregistration	Manages the process of deregistration.
Session Origination	Manages the life cycle of the originating session, invoking the VoLTE Supplementary Services for originating sessions.
Session Registration	Manages the life cycle of registration.
Session Termination	Manages the life cycle of the terminating session, invoking VoLTE Supplementary Services for terminating sessions.

Table 3–2 (Cont.) VoLTE and VoWiFi Application Control Flows

Control Flow	Description
Terminating Access Domain Selection	Selects the circuit-switched access or packet-switched access network(s) to deliver a terminating voice session to the user endpoint.

For the Session Origination and Session Termination control flows, you can click **Services & Control Flows** to list services invoked and the flows executed.

About Control Flows

A control flow is a logical, flowchart-like structure that consists of linked Activities and decisions that specify how Evolved Communications Application Server processes a voice call or multimedia session.

You base control flow logic primarily on a subscriber's profile and preferences and on the circumstances of the call itself. Factors affecting a session can include the subscriber's calling plan, the originating and terminating locales, time of day, day of week, subscriber's available minutes, and so on. Control flows also send notifications, play announcements, forward calls, and much more.

The **All Control Flows** tab contains the set of all control flows that are delivered with OCECAS, as well as any that have been created since it was installed.

About Activities

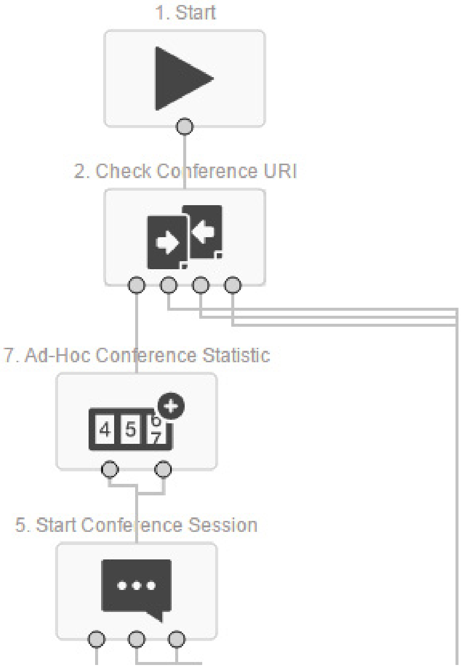
An Activity is an action block that is embodied by an icon that you can drag and drop in a control flow. Each Activity performs a particular task in a control flow and has one input and a number of outputs, except for the Start and End Activities, which have one output and one input respectively.

Activities provide individual service functions such as playing announcements, sending notifications, and branching based on session data. For a voice call, a Day Of Week Activity allows you to branch based on the day of the week on which the call is made. You might, for example, need to account for special holiday rates.

For a list of the Activities available to use in a control flow, see "Activities Reference" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

Figure 3–1 illustrates the Activities at the beginning of the Ad-Hoc Conferencing control flow.

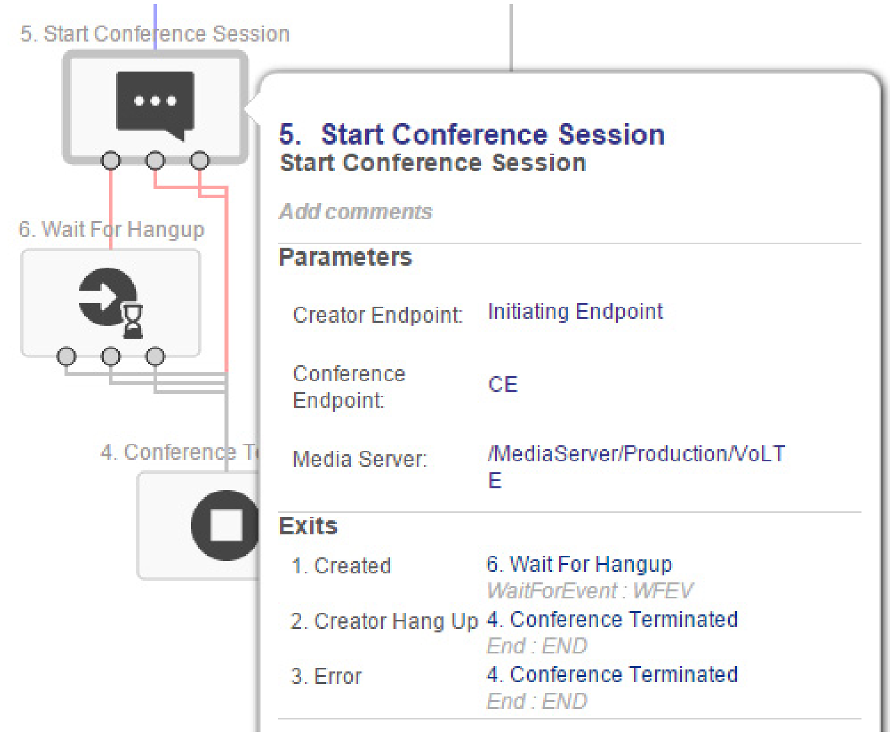
Figure 3–1 Activities in the Ad-Hoc Conferencing Control Flow



Each Activity in a control flow, except for the End Activity, has one or more exits. The Start Conference Session Activity in Figure 3–1, for example, has three exits. If you move the cursor over an exit point, a label identifies the type of exit, such as Success, Caller Hangs Up, or Error.

If you double click an Activity, an informational window displays any parameters that have been set for the Activity, as well as the reason for each exit and the next destination in each exit path. Figure 3–2 illustrates the informational window associated with the Start Conference Session Activity.

Figure 3–2 Informational Window for Start Conference Session Activity



5. Start Conference Session
Start Conference Session

Add comments

Parameters

Creator Endpoint:	Initiating Endpoint
Conference Endpoint:	CE
Media Server:	/MediaServer/Production/VoLTE

Exits

1. Created	6. Wait For Hangup *WaitForEvent : WFEV*
2. Creator Hang Up	4. Conference Terminated *End : END*
3. Error	4. Conference Terminated *End : END*

When creating a new control flow, or editing one that has not been deployed, you can assign parameter values to an Activity or edit existing values. Once you have deployed the control flow, it becomes read-only. If you subsequently wish to modify the control flow, you must create a new change set from the appropriate baseline and modify the control flow within that change set.

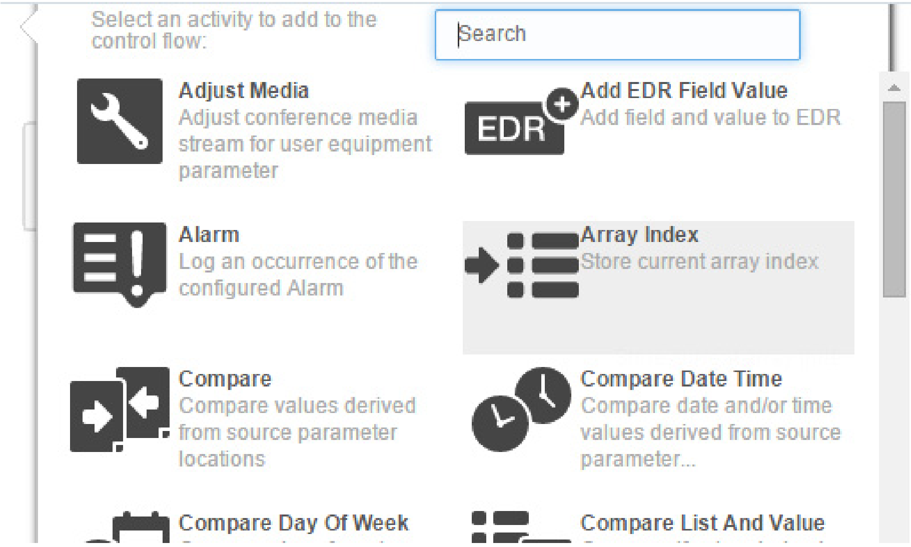
When you are adding Activities to a control flow, you can specify a different name for the Activity and add informative comments. The name you assign appears above the Activity in the control flow. The Activity also retains its original name, however, for purposes internal to OCECAS.

About Creating a Control Flow

To create a new control flow, open a change set, select the **All Control Flows** tab and then click **New**. Enter a name and description when prompted and then click **Create**.

When you create a new control flow, you open the control flow editor and the canvas is populated with the Start Activity. If you drag your cursor from the exit for the Start Activity, you create a connector line. When you stop and release the cursor, the window shown in Figure 3–3 opens to display the set of available Activities.

Figure 3–3 Available Activities


Select an activity to add to the control flow:
Search
Adjust Media
Adjust conference media stream for user equipment parameter
EDR
Add EDR Field Value
Add field and value to EDR
Alarm
Log an occurrence of the configured Alarm
Array Index
Store current array index
Compare
Compare values derived from source parameter locations
Compare Date Time
Compare date and/or time values derived from source parameter...
Compare Day Of Week
Compare List And Value

You can either scroll down through the list to select an Activity or do a search and select one from the result set. Each exit for the new Activity must be connected to another Activity. Continue this process until all branches terminate with an End Activity and the control flow is complete.

For more information on creating control flows and working with Activities, see "Working with Control Flows" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

About Viewing and Editing Control Flows

To edit a control flow you must open it within a change set. If you open a control flow outside of a change set, it opens in read-only mode and you must click the **Edit** button, which prompts you to either create a new change set or open an existing one. You can open control flows in the following ways

- Clicking a control flow in the **Application** tab
- Clicking a control flow in the **All Control Flows** tab
- Opening a change set in the **Open Projects** area and then opening a control flow from within the change set.

About Connecting Control Flows

You can invoke another control flow from within a control flow by using the Run Control Flow Activity, which requires the target control flow name as a parameter.

Nesting control flows helps to keep a master control flow to a manageable size. It also allows you to put common service logic in a single place and reuse it multiple times.

About SNMP Events

You can issue an SNMP event from a control flows by adding the SNMP Event Activity at the point at which the event occurs. For example, you might want to raise an SNMP event for an Activity error branch in a control flow.

The Activity parameters allow you to specify an object identifier (OID) postfix, which is appended to the fixed OCECAS OID prefix of 1.3.6.1.4.1.111.10. The OID uniquely identifies the source of the SNMP message. The SNMP Event Activity also allows you to specify alarm text and an alarm resolution.

OCECAS converts the SNMP event to an SNMP trap and sends it to the Network Management System (NMS) to be logged. The NMS determines whether the event should be turned into an alarm and also decides what level of severity to assign the event. The NMS can also clear an alarm based on receipt of a specific event. For example, the NMS can be configured to raise an alarm when it receives a linkDown event and to clear the alarm when it receives a linkUp event.

For more information on creating an SNMP event in a control flow, see “Creating an SNMP Event” in the *Oracle Communications Evolved Communications Application Server Operator’s Guide*. For information on configuring SNMP events, see “Managing SNMP Events” in *Oracle Communications Evolved Communications Application Server System Administrator’s Guide*.

About Validating and Compiling Control Flows

An editable control flow is automatically saved, validated and compiled whenever you make a change to the flow in the control flow editor. For a control flow to compile successfully, all the Activities in the control flow must contain valid configuration data and all exits must be connected.

Only successfully compiled control flows can be assigned to a service. A control flow is successfully compiled when it has no errors shown.

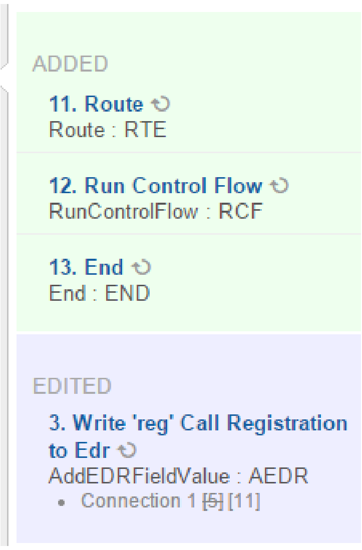
About Using Control Flow Tools

The left border of the control flow canvas contains a palette with tools that enable you to take actions such as search, undo, redo, save, cut, copy, and paste. Other tools allow you to examine issues in the control flow and view differences between the current change set and the baseline. For more information about the tools palette, see "About the Control Flow Editor" in *Oracle Communications Evolved Communications Application Server Operator’s Guide*.

About Viewing Control Flow Differences

You can view the differences between a control flow in the current change set and the control flow in the baseline change set by clicking the Pencil icon in the Tools palette. Figure 3–4 illustrates an example instance of control flow differences:

Figure 3–4 Example of Control Flow Differences



ADDED

11. Route
Route : RTE

12. Run Control Flow
RunControlFlow : RCF

13. End
End : END

EDITED

3. Write 'reg' Call Registration to Edr
AddEDRFieldValue : AEDR

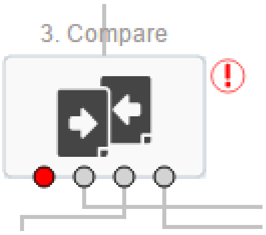
- Connection 1 ~~[5]~~ [11]

About Control Flow Errors

If a control flow has errors, a warning icon appears next to the **Control Flows** entry in the Changes panel. (See "About Opening and Viewing Change Sets" for information about the Changes panel.) Clicking **Control Flows** in the Changes panel lists any control flows in error with a list of the errors found for each one.

As Figure 3–5 illustrates, a warning icon also displays on the editing canvas next to any Activity that has errors, for example unspecified parameters or unconnected exit points. Moving the cursor over the warning icon displays a summary of the errors found.

Figure 3–5 Activity with an Error



You can also double click the Activity to see a list of errors in the Activity's informational panel.

Clicking the warning icon in the tools palette also displays a list of Activities in error with a list of the errors for each one.

About Resources

The **All Resources** tab opens a menu of data and media resources that are available for a control flow to use. The resources are grouped by the following resource types:

- **General**

  General resources includes a table of locales. A locale is a combination of a language and a country, which serves to identify the location of an originating or terminating endpoint in a call. The country is identified by a country code and the language by a language code. For example, the locale Arabic (Sudan) has a language code of **ar** and a country code of **SD**.

- **Media**

  Media resources consist of Resources and Servers. Media resources are announcements that a control flow can play. Media resources reside on a controlling media server that manages them. For example, a media resource could be a **Hold-Music** announcement and the server could be **/MediaServer/Production/MyServer**.

- **Numbers**

  The Numbers resource consists of a prefix tree, called CountryCodePrefixTree, that maps country codes and global prefixes. The prefix tree entry for Chile, for example, is CHL:56. If a country has multiple prefixes, sub-trees are embedded to contain the global prefixes. For example, the entry for northern Alberta, Canada is CAN->Alberta->Northern:1780.

- **Templates**

  The Templates resource consists of the following types of templates:

  – Notifications

    Each notification definition has parameters that a control flow can use to locate a locale-specific template. The parameters are placeholders that consist of a name and a data type and an optional description. The control flow editor allows you to specify a value for each parameter when you select the notification definition within a control flow.

    Each locale is configured with a template that a control flow can use to generate the contents of an end-user notification at run-time. For example, assume: 1) that a subscriber has suspended service and 2) a notification called Suspended exists and 3) the notification has an associated template called Japanese Suspended and 4)the caller's profile has a language of Japanese. In this case, the call to that subscriber fails and the Japanese notification is sent to the caller.

  – Web Services

    The Web Services resource enables you to manually create a template for accessing a web service.

  – Import WSDL

    The Import WSDL resource enables you to create web service templates by importing WSDL files to the server.

For more information about creating and using resources, see "Working with Resources" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

About Projects

The Projects area of the Session Design Center contains all change sets.

About Change Sets

As previously mentioned, a change set is a grouping of related configuration and control flow changes. Change sets facilitate the change management process, making it easier to organize and deploy changes. For information about deploying change sets, see "About Environments".

Each change set is identified by the tag CS-*n*, where *n* is an automatically assigned sequence number, and by a name that you assign.

About Creating a Change Set

You can create a new change set in either the **Application** area or the **Projects** area. From the **Application** area, click **Edit** and then **New Change Set**. In the **Projects** area click **New**.

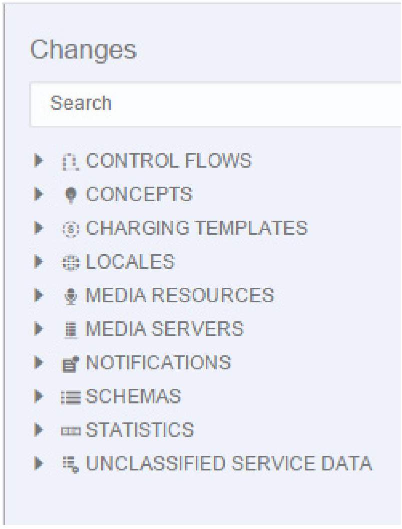
Enter a name and description for the change set and select a baseline change set to be the parent of the new change set. The changes you add to the new change set represent modifications to the baseline change set.

See "About Creating a Control Flow" for an overview of adding content to a change set. See "Working with Change Sets" in *Oracle Communications Evolved Communications Application Server Operator's Guide* for information on creating and working with change sets.

About Opening and Viewing Change Sets

If there are open change sets, the name **Projects** on the home page changes to **Open Projects** and the open projects are listed below. You can click these to open them for editing. The change set opens in the same view as described in "About Application Configuration", with tabs for **Application**, **All Control Flows**, and **Resources** and an additional **Changes** panel to the right, shown in Figure 3–6.

Figure 3–6 Change Set Changes Panel



You can expand each category in the Changes panel to see the specific changes made in that category. Entering a value in the Search box reduces the items shown in each category to those that contain a matching value.

Click **View All Projects** to view all change sets in a list. You can filter the list to select either **All**, **Open**, or **Deployed** change sets. An open change set is followed by a pencil icon that indicates it is available for editing. A deployed change set is followed by an eye icon that indicates it is read-only.

You can search the list of change sets by entering a string that occurs in the name or description of the change set. The list is reduced to those change sets in which a match occurs.

Open a change set by clicking either its name or the pencil or eye icon that appears next to it. The change set opens in the indicated mode.

About Change Set Details

Clicking the change-set list item expands it to display the change-set creation date and the baseline change set from which the change set was created. See "About Creating a Change Set" for additional information about the baseline change set.

If the change set is editable, you can click Edit Details to open a dialog that allows you to edit the name, description, and baseline change-set.

About Environments

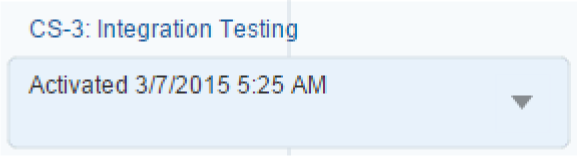
The **Environments** area provides a change management workflow for deploying change sets. The **Environments** area allows you to move a change set from open-project status into the deployment pipeline, which consists of Testing, Staging, and Production environments. Each environment in the pipeline represents a SCF runtime domain in OCECAS.

A change set must have compiled successfully to be eligible for deployment. If the change set contains errors, it is flagged with the warning icon in the **Open Projects** panel and you will not be able to deploy it. You also cannot deploy a change set whose baseline change set, known as its parent, has not yet been deployed.

You can only deploy a change set to the next environment in the pipeline. You cannot skip from the Testing environment to the Production environment, for example.

To move an open project into the pipeline, drag it from the Open Projects panel into the Testing environment. Figure 3–7 illustrates how a change set is represented in the pipeline.

Figure 3–7 A Change Set Object in the Deployment Pipeline



CS-3: Integration Testing

Activated 3/7/2015 5:25 AM

You can take the following change management actions on a change set. These options are available from the pull-down menu on the change-set object.

- Deploy to Staging or Deploy to Production

  Deploys the change set to the indicated stage in the pipeline.

- Retract

  Retracts a change set from its current environment. This prevents a change set from being used in new sessions, but allows existing sessions to continue.

- Redeploy

  Redeploys a retracted change set to the current environment; can also redeploy an edited change set to a testing environment.

- Remove (Testing environment only)

Removes a change set from the Testing Environment. This menu item is inactive in the Staging and Production environments. Change sets that have been deployed to the Staging and Production environments cannot be removed, only retracted.

See "About Change Management" for a complete discussion of change management concepts. See "Working with Change Sets" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

4 About Change Management

This chapter explains the Oracle Communications Evolved Communications Application Server (OCECAS) change management features.

About Modifying the VoLTE and VoWiFI Application

You collect the control flows that define your multimedia services and SIP call behaviors into groupings called *change sets*. You can include any number of control flows into a change set and manage them as a unit. You use change sets to ensure that the integrity of your changes is maintained from inception through production deployment. You manage change sets using the Session Design Center UI. This tool has a one-click interface for deploying, retracting, redeploying, or removing change sets.

See "About Control Flows" for details on control flows.

About Change Sets

Essentially, change sets add version control to your multimedia services. They gather software changes into an immutable set so you can manage them as a unit. Most organizations only have their code files under version control. However, Evolved Communications Application Server allows you to manage complete components (control flows, product types) and their control logic using change sets. You manage control flows and service data with change sets.

You create change sets by copying an existing change set or branching from an existing change set. All change sets have a single parent change set, and different change sets can share the same parent. By default, every OCECAS implementation starts with these default change sets:

- Installation - Which provides the OCECAS software binaries. This change set cannot be used as a parent.
- VoLTE, VoWiFi, and eSRVCC - Which contains the control flows that define the default services for the VoLTE, VoWiFi, multimedia application.

Each change set can contain *message control flows* and *application control flows*. Message control flows define application behavior for a specific SIP message. The default message controls define behavior for the INVITE, OPTIONS, REGISTER, and SUBSCRIBE SIP messages. You can define your own message control flows to define the call control behaviors of the other SIP message types.

Application control flows define all other multimedia services that you define. The default application control flows provided with OCECAS are listed here. The subheadings under Session Origination and Session Termination represent supplementary services. See "About Using VoLTE Supplementary Services" for details on the supplementary services.

- Session Deregistration
- Session Origination
  - Ad-hoc Conferencing
  - Communication Hold Charging
  - Communication Hold
  - Originating Identity Restriction
  - Outgoing Communication Barring
  - Terminating Identity Presentation
- Session Registration
- Session Termination
  - Anonymous Communication Restriction
  - Communication Diversion
  - Communication Hold
  - Early Communication Diversion
  - Incoming Communication Barring
  - Originating Identity Presentation
  - Terminating Identity Restriction
- Terminating Access Domain Selection

About Creating Change Sets

This section explains some best practices for creating and using change sets:

- You can think of a change set as a single project. Group related changes (with a common purpose) into a single change set.
- Err on the side of versioning components. If you can add a component to a change set, then definitely do so.

You can change the parent that a change set uses if the change set has not been deployed. However, you must manage your changes carefully. If you specify an incompatible parent. you make the change set undeployable, for example:

- Assume change set 1 uses a “Hello World” announcement.
- You then create change set 2, and delete the “Hello World” announcement.
- You then create change set 3 using change set 1 as the parent, and it uses the “Hello World” announcement.
- If you modify change set 3 to use change set 2 as the parent, change set 3 fails to compile because it relies on the announcement that is no longer available.

About Deploying Change Sets

The production pipeline includes testing, staging, and production environments. You deploy the change sets from one environment to another using the Session Design Center UI.

Change sets are immutable once they are deployed. At that point they cannot be changed, only retracted.

Change sets are compiled automatically by the Session Design Center UI, and cannot be deployed with errors.

You deploy change sets in order (from testing to staging, and then to production) with a one-click action in the Session Design Center UI. You can retract deployments as needed, the same way. You first deploy (or retract) the change set parent, then deploy or retract the child.

You set a “schedule for activation” date when deploying a change set that specifies when it becomes visible.

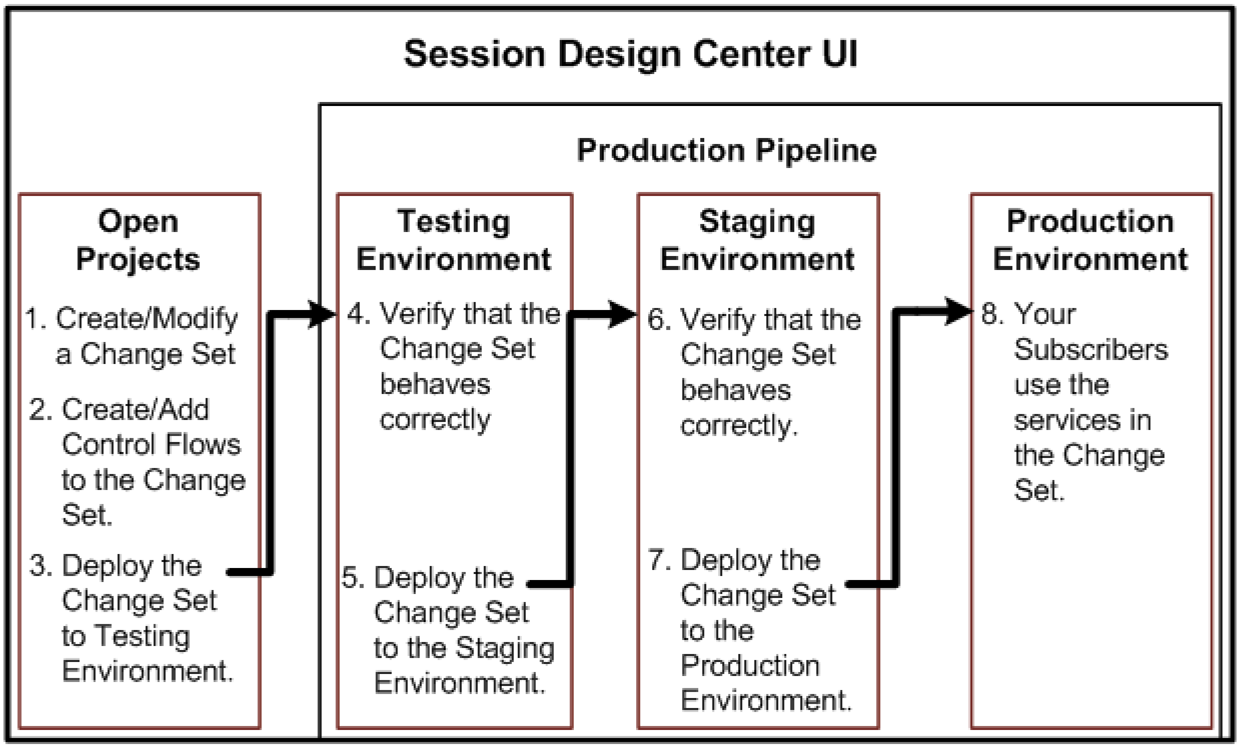
See “Deploying a Change Set” in *Oracle Communications Evolved Communications Application Server Operator’s Guide* for details on how to deploy change sets to your production pipeline.

Change sets that have not yet been deployed to an environment are displayed in the **Open Projects** area of the Session Design Center UI.

Figure 4–1 illustrates the process of deploying changes to your production pipeline. The pipeline shown in this figure has Testing, Staging, and Production environments. To the left of the production pipeline is a box marked Open Projects. You first create a change set as a vehicle for the changes. Once you create a change set, it is displayed in the Open Projects area of the Session Design Center. The new change set inherits all of the control flows and resources from its parent. You then any other control flows that your implementation requires.

Once a change set contains all the control flows required, you deploy it to the first environment in your production pipeline, the Testing Environment. After you verify that it behaves as you intend, deploy it to the Staging Environment, and then verify that it works as you intend there. Then you deploy the change set to the Production Environment so your subscribers can use it.

Figure 4–1 Deploying Changes to the Production Pipeline


Session Design Center UI
Production Pipeline
Open Projects
1. Create/Modify a Change Set
2. Create/Add Control Flows to the Change Set.
3. Deploy the Change Set to Testing Environment.
Testing Environment
4. Verify that the Change Set behaves correctly
5. Deploy the Change Set to the Staging Environment.
Staging Environment
6. Verify that the Change Set behaves correctly.
7. Deploy the Change Set to the Production Environment.
Production Environment
8. Your Subscribers use the services in the Change Set.

About Retracting Change Sets

You retract a change set by changing the state of the change set. Retracted change sets are still immutable, but you can create a new change set from a change set in an immutable state.

5

About Managing and Using Subscriber Data

This chapter describes how Oracle Communications Evolved Communications Application Server (OCECAS) manages subscriber data.

About Subscriber Data Stores

OCECAS uses three possible types of subscriber data stores to store and access subscriber data: a home subscriber server (HSS) accessed through the Diameter Sh interface, the OCECAS default subscriber server (ESS) NoSQL database, which is optional, or a third-party data store from another provider. The subscriber data stores are also known as user data repositories (UDRs).

> **Note:** The ESS NoSQL database is only for OCECAS to use for storing subscriber data. Do not store anything else in this database.

If you are using the default ESS NoSQL database, every runtime environment in an OCECAS pipeline must have access it. However, more than one runtime environment can share an ESS, which you can place in the following locations:

- In a runtime domain.
- In a dedicated UDR domain.
- In another location that you specify.

However, the data store configuration is flexible. You can use an external database as the ESS instead of the default NoSQL database. See "Replacing the ESS NoSQL Database" in *Oracle Communications Evolved Communications Application Server Operator's Guide* for information.

You can populate the ESS with subscriber data using a REST API. For more information, see "Provisioning the ESS with Subscriber Data" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

OCECAS includes data *views* that specify connection details to these data stores and invoke *federation scripts* to merge subscriber data into the format that the application expects. An application requests subscriber data from a specific view. Views are defined in a *domain_home*/**config/custom/csp.xml** file for each runtime domain.

The Groovy-based federation scripts create views insulate applications from the HSS and ESS data sources and their data formats. The federation scripts do the following:

- Federate the data as necessary from the data stores for use by the calling applications.
- If necessary, translate the data into a format that the calling application can use.

Consequently, you need to update the federation scripts if your subscriber data or data stores change.

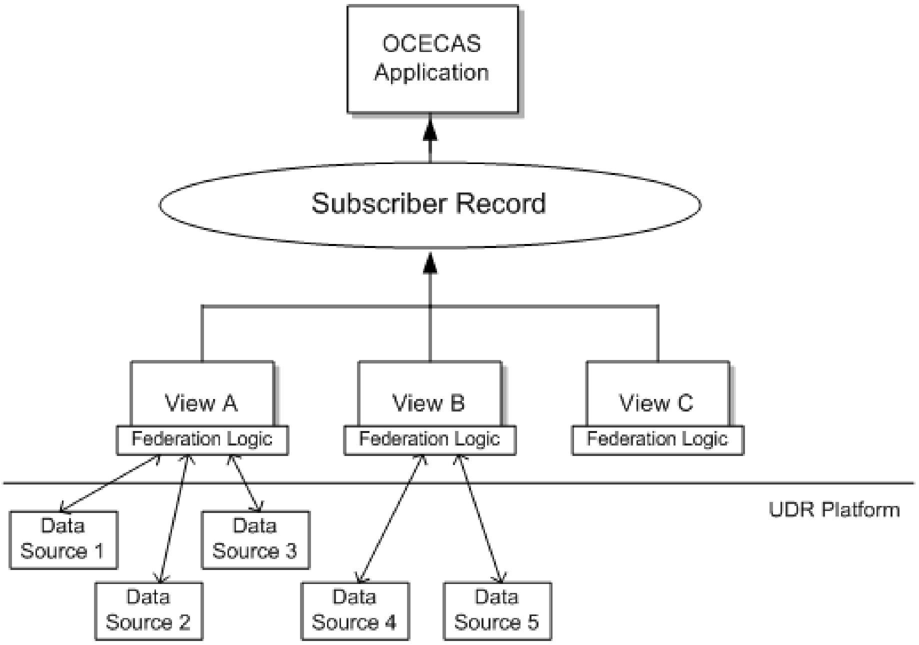
You can make service decisions such as selecting features, or changing charging levels based on subscriber information. For example, you could obtain subscriber charging information from an HSS, and federate it with details about the services the subscriber is using from a roaming location.

About Views

The **csp.xml** file defines views that specify the data stores to use, for example, an HSS, the default NoSQL ESS or a data store from a third-party provider. Views contain connection details for the data stores and specify how to merge data from different sources and translate it from the data-store format into the format the application expects.

Figure 5–1 illustrates the construction of a subscriber record from multiple views and data sources.

Figure 5–1 Subscriber Record Constructed from Multiple Views and Data Sources



If necessary, you can make changes to the data federation logic at runtime by changing the **csp.py** Python script using the WebLogic Scripting Tool (WLST). For more information, see "Updating Federation Script Behavior During Runtime" in the *Oracle Communications Evolved Communications Application Server Operator's Guide*.

You need to ensure that the **csp.xml** file is protected. See "About Developing Secure Applications for OCECAS" in *Evolved Communications Application Server Security Guide* for information.

See "Working with Subscriber Data" in *Evolved Communications Application Server Operator's Guide* for details on the default views and how to change them for your implementation.

6

About the OCECAS Architecture and Environment

This section explains the Oracle Communications Evolved Communications Application Server (OCECAS) architecture and how its components work together.

About the OCECAS Architecture

OCECAS is an application server for VoLTE services. More specifically, it is an IMS application server responsible for call control. It provides VoLTE-centric single radio voice call continuity (SRVCC) telephony voice and video communications to packet-switched IMS networks. It comes pre configured to work with border controllers such as the Oracle Communications Session Border Controller.

OCECAS is based on and uses features from both the Oracle Communications Converged Application Server (Converged Application Server) and the Oracle Fusion Middleware WebLogic application server (WebLogic server). These products are included in OCECAS.

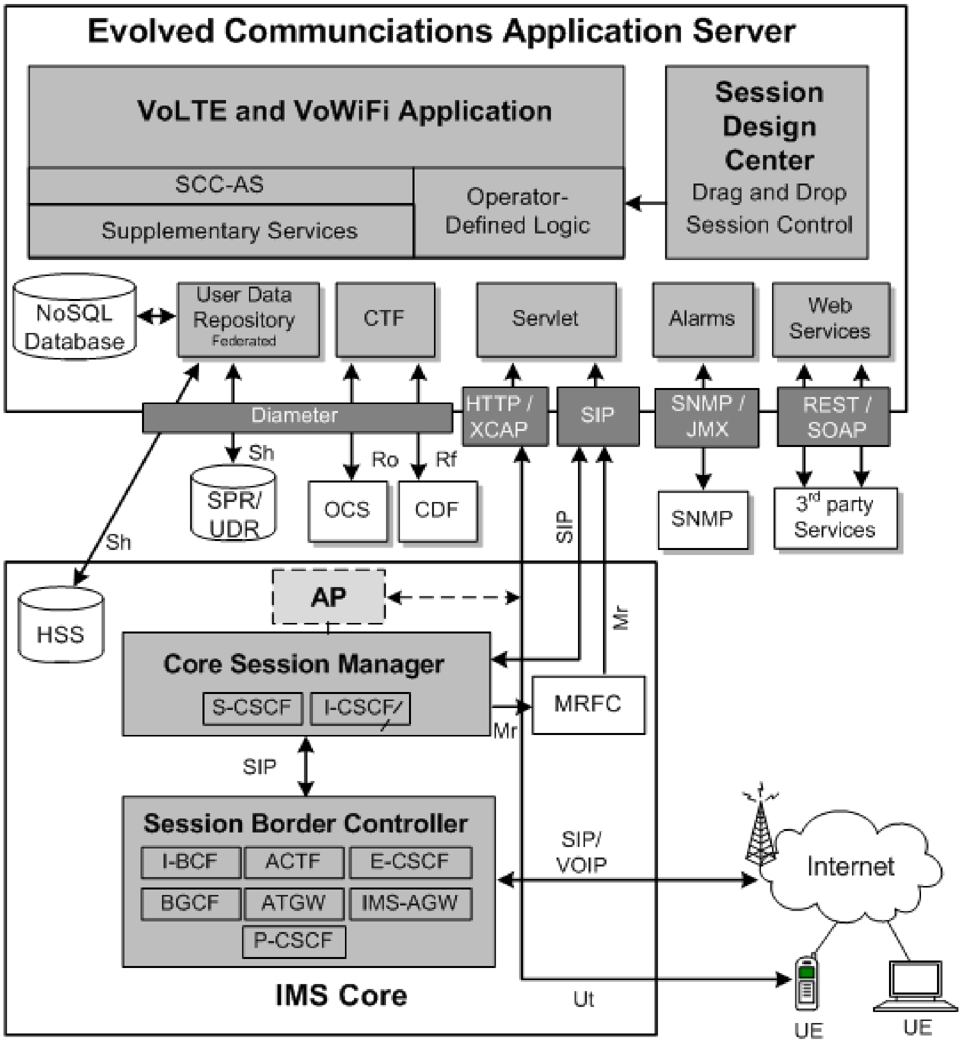
OCECAS includes the Session Design Center GUI tool that allows an operator to customize subscriber communication sessions. The Session Design Center tool uses a simple and efficient drag-and-drop interface to create and manage session control flows.

About the OCECAS Environment

OCECAS sits on the edge of your IMS telecommunications network and serves as a multimedia (VoLTE voice and video) application server. It enables you to serve VoLTE applications on hybrid 2G/3G/IMS networks.

Figure 6–1 illustrates the OCECAS components, their relationships, and their intercommunication.

Figure 6–1 ECAS Components and Their Intercommunication



The top level includes the main components of the Evolved Communication Application Server: the VoLTE and VoWifi application, the Session Design Center, the default NoSQL user data repository (UDR) or ECAS Subscriber Server (ESS), the UDR views, the Charging Trigger Function (CTF), the HTTP and SIP servlet, alarms, and web services.

OCECAS also connects to the following external elements: a home subscriber server (HSS), optional third-party UDRs, an online charging system (OCS), a charging data function (CDF), a device managed by the Simple Network Management Protocol, third-party services, an optional authentication proxy (AP), and the Core Session Manager (CSM).

OCECAS connects to your IMS network through Core Session Manager using Session Border Controller (SBC). User endpoint devices (UEs) connect to ECAS through the SBC and the CSM for voice and video sessions. They connect directly to ECAS using XCAP / HTTP 1.1 to configure supported supplementary services.

Understanding the OCECAS Components

Together, the OCECAS components and technologies provide a pure IMS solution for serving applications as a Service Centralization and Continuity Application Server

(SCC-AS), which provides SRVCC voice and video communication over high-bandwidth packet-switched networks.

By providing supplementary services, OCECAS offers video call support as defined in the GSMA *IR.92 IMS Profile for Voice and SMS* specification. ECAS also supports the User Entity (UE) registration of video as defined in the IR.92 IMS Profile for Video and SMS.

Understanding Converged Application Server

The Oracle Communications Converged Application Server (OCCAS) serves multimedia applications that use the SIP protocol. As the name implies, one of its strengths is serving converged applications that rely on various protocols and interfaces. OCCAS supports 3rd-party application development and hosting, and allows multiple applications to execute on the same incoming or outgoing request. Each application can execute separately and use its own set of rules. Each SIP servlet stores application data in a container-managed session object, so implementations persist and replicate this data to create a highly available implementation. Converged Application Server supports the *JSR 359 SIP Servlet Specification, version 2.0*:

`https://jcp.org/en/jsr/detail?id=359`

Understanding WebLogic Server

Oracle WebLogic Server, which OCECAS incorporates, is a scalable, enterprise-ready Java Platform, Enterprise Edition (Java EE) application server. It is an ideal foundation for building applications based on service-oriented architecture (SOA).

The WebLogic Server implementation of the Java EE 6.0 specification provides a standard set of APIs for creating distributed Java applications. These applications can access a wide variety of services, such as databases, messaging services, and connections to external enterprise systems. WebLogic server also provides server support, such as diagnostic tools to tune performance, and the ability to capture business intelligence.

Understanding Session Border Controller

OCECAS requires a border controller and it comes pre-integrated to use Oracle Communications Session Border Controller (Session Border Controller) and Oracle Communications Core Session Manager (Core Session Manager) for this purpose. However, you can use any session border controller that your implementation requires.

Session Border Controller is a component required by OCECAS that you purchase separately. It enables service providers to deliver trusted, first-class real-time communications services across Internet Protocol (IP) network borders, including any services across IMS borders. The functions offered by Session Border Controller satisfy critical service provider requirements in five major areas: security, interoperability, reliability and quality, regulatory compliance, and revenue/cost optimization.

Understanding Core Session Manager

Core Session Manager is a product that you need to purchase separately. However most implementations require it. It is an agile session core for supporting Voice over LTE (VoLTE) services (VS-CSCF). It directs signaling flows through the network and between elements. Network Function Virtualization (NFV) allows service providers to share hardware resources efficiently across different services and customers. Core

Session Manager is designed to support core session management capabilities in virtualized environments. It provides a complete set of session core functions including IP Multi-media Subsystem (IMS) Call/Session Control Functions (CSCF) and their associated 3GPP interfaces.

About Integrating OCECAS into Your Network

When you install OCECAS, you automatically get Converged Application Server and WebLogic Server technologies included in the installation.

In addition, obtain and install these Oracle Communications products:

- Core Session Manager
- Session Border Controller

> **Note:** Instead of purchasing Core Session Manager and Session Border Controller separately, you could purchase a copy of Oracle Communications Unified Session Manager (Unified Session Manager). Unified Session Manager contains both Core Session Manager and Session Border Controller pre-configured to work on their own hardware system as a I/S-CSCF.

Once installed, configure OCECAS to connect to them.

Understanding How OCECAS Communicates

The OCECAS servers use the SIP protocol to build up and tear down calls for voice and video communications.

OCECAS uses the Diameter Sh specification to communicate with the home subscriber server (HSS) and external subscriber data repository (SPR/UDR).

The charging control function (CTF) communicates with the online charging server (OCS) using the Diameter Ro online charging specification. The CTF communicates with the charging data function (CDF) using the Diameter Rf offline charging specification.

The Core Session Manager communicates using the *JSR 309 Media Control Server API Standard* to communicate with media resource servers. The JSR 309 specification is available from the Java Community Process web site:

`https://jcp.org/en/jsr/detail?id=309`

OCECAS uses the REST and SOAP protocols to interact with 3rd-party applications and to provision subscribers.

OCECAS uses XCAP / HTTP 1.1 to communicate with a user handset to support supplementary services.

About Overload Protection

OCECAS uses the Converged Application server capabilities to provide failover protection. As shown in Figure 1–3, the OCECAS production environment uses the Converged Application Server engine and Coherence cluster (in this case, to store the Services and Change Set data). The engine controls the SIP sessions, but keep the session state and data inside the Coherence cluster. The Coherence cluster is distributed and replicated to enable failover so calls are not lost.

OCECAS is self-throttling to prevent call degradation in overload situations. Any existing sessions continue, but the platform only lets new sessions form if there is enough bandwidth to run them efficiently.

7

About Network Function Virtualization

This chapter describes the concepts and implementation of network function virtualization for Oracle Communications Evolved Communications Application Server (OCECAS), which is pre-integrated with Oracle Communications Application Orchestrator (OCAO) to centrally manage and automate your virtual network deployment.

For information on configuring and administering network function virtualization, see "Managing Network Function Virtualization" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*. For information on using Oracle Communications Application Orchestrator, see *Oracle Communications Application Orchestrator User Guide*.

Overview

Network function virtualization (NFV) virtualizes network node functions into building blocks that you can connect to create communication services. A virtualized network function (VNF) consists of one or more virtual machines running software and processes on industry-standard, high-volume servers, switches, storage, or even cloud computing infrastructure, instead of using custom hardware appliances for each network function.

NFV requires running OCECAS as a series, or cluster, of virtual machines. A VNF manager controls the automatic deployment and shutdown of VMs, based on various load conditions. Oracle Communications Application Orchestrator is a VNF manager.

OCAO supports Oracle OpenStack for Oracle Linux 1.0 and vCloud Director 5.5.x (VMware) and as Virtual Interface Managers (VIMs)/Network Function Virtualization Infrastructure (NFVI) interface. OCAO supports KVM, OVM_PV and OVM_HVM hypervisors under Oracle OpenStack and ESXi hypervisor under vCloud Director. For more information on which environments are supported by your version of OCAO, please refer to the product documentation.

Figure 7–1 illustrates the OCECAS NFV architecture.

Figure 7–1 Figure 7-1 OCECAS NFV Architecture

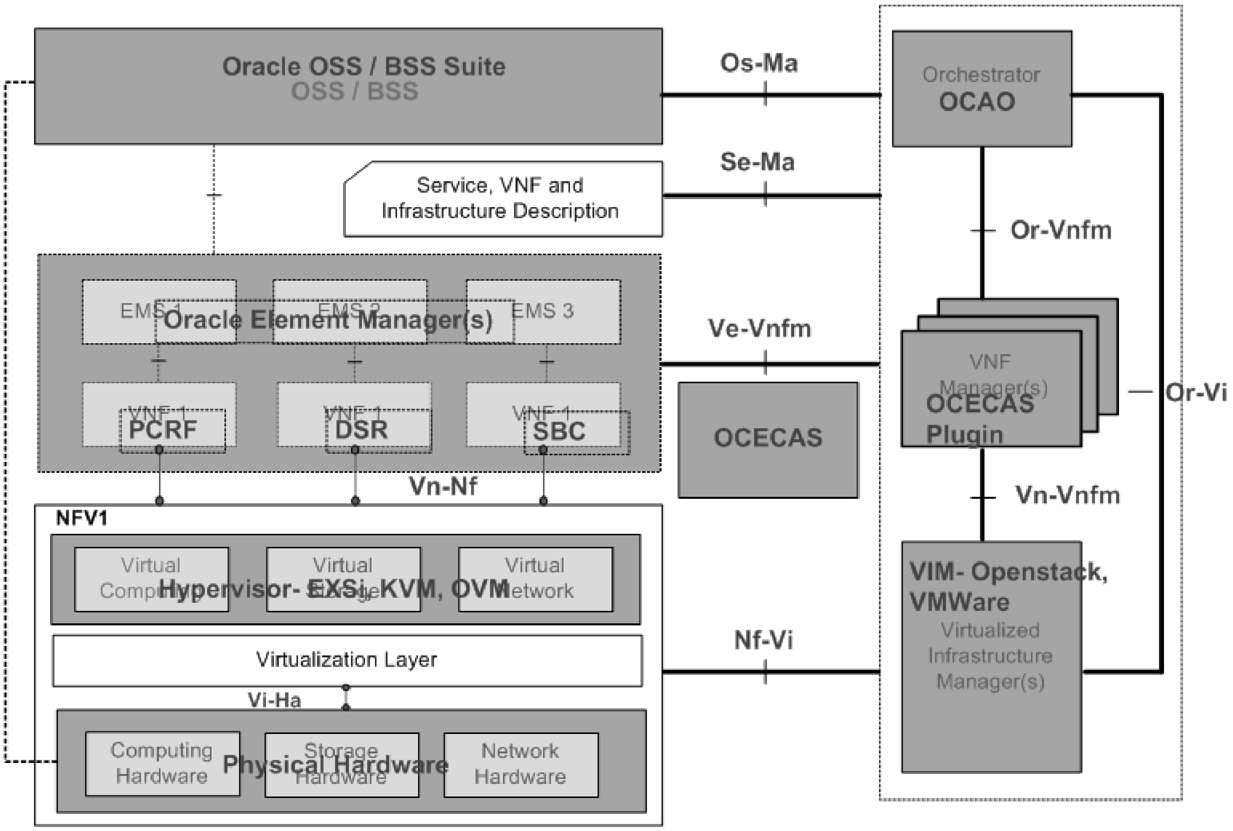


Table 7–1 describes the major NFV components:

Table 7–1 NFV Components

Component	Description	Example
VNFx	The running network application	OCECAS
CNF	A composite network function, which is a network function that can be physical or virtual, or both. This is the main configuration element for an AO plugin. This is the closest equivalent to the ETSI VNF definition. For OCECAS, there is one CNF for each domain.	Managed Server
EMS	The management function of the VNF.	Oracle Communications Session Element Manager. Also known as Net-Net Central or OC Session Delivery Manager
NFVI	The physical resources and the VM software	VMWare, OpenStack, or Oracle Virtual Machine

Table 7–1 (Cont.) NFV Components

Component	Description	Example
VIM	The orchestration manager, which controls the NFVI resources and the interaction of the VMs with them.	Oracle Enterprise Manager Orchestration Engine or OpenStack
VNF Manager	Gathers statistics from the VNFs and instructs VIM to alter VMs. Automates deployment and shutdown of VMs.	Oracle Communications Application Orchestrator

In a traditional implementation, an element manager collects application statistics from the VMs and reports them as key performance indicators (KPIs) to OCAO. OCAO collects statistics from the VNFs (via the EMS) and applies configured logic to decide if more VNFs should be added or removed.

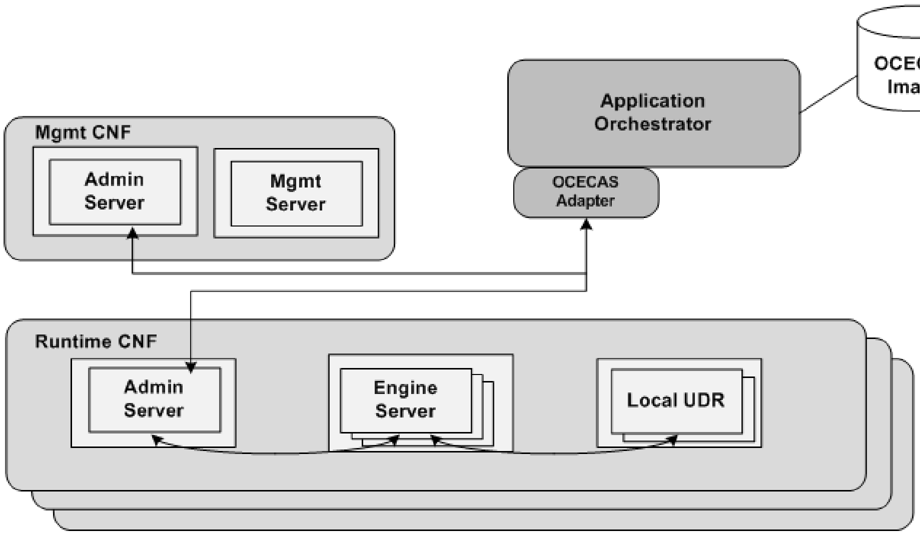
For OCECAS there is no physical element manager. A combination of the administration or managed server and the OCAO adapter performs the role of element management.

About the OCECAS NFV Topology

OCECAS is divided into multiple CNFs based on various factors like KPI rules, policies, deployment environments and logical functionality. These CNFs consist of a single management CNF and multiple runtime CNFs, one for each testing, staging and production environment.

Figure 7–2 illustrates the OCECAS NFV topology.

Figure 7–2 OCECAS NFV Topology



About the Management CNF

By design, the management platform is not a good candidate for horizontal scaling based on KPIs. The management platform is used primarily by service designers and does not expand beyond a predictable limit. Scaling for the future needs of database storage and throughput can be accomplished within a single VM.

The management CNF does not need runtime scaling but does require a separate VM. For more information, see "Configuring the Domain and CNF and "Example CNFD Parameters for a Management Domain" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About the Runtime CNF

The runtime CNF consists of two network function groups, the engine group and the UDR group. Each group can scale independently.

For information about the CNF description parameters, see "Configuring the Domain and CNF" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About the Engine Group

The engine forms the runtime environment for OCECAS. The runtime environment is where live calls execute the control flows that are designed and stored on the management platform. Data from the management platform is replicated to a Coherence cache that the engines read.

Any updates to a Coherence cache are replicated across all of the caches so scaling does not adversely affect the Coherence cache.

About the UDR Group

The UDR is a NoSQL database that holds subscriber and service information that you provision through the RESTful API. NoSQL provides for scaling of nodes and supports the scaling philosophy. For more information about the NoSQL database, see "About Managing and Using Subscriber Data" in this guide and "Working with Subscriber Data" in *Oracle Communications Evolved Communications Application Server Operator's Guide*.

For information on administering NoSQL, see the *Oracle NoSQL Database Administrator's Guide*.

About the OVA File

The OCECAS AO adapter requires an Open Virtualization Archive (OVA) image of a virtual device to apply VM configuration and deploy instances. The OCECAS AO adapter uploads a separate OVA file into OCAO for each NF group. The file contains application-specific configuration information for OCECAS, such as Java Message Service (JMS), data source, and so on. For the procedure to create the OVA file, see "Creating the OVA Image File", in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

Table 7–2 describes the components that OVA files must have installed and configured for each type of NF group.

Table 7–2 Installed Components for Each NF Group

NF Group (Domain)	Components
Management	OCECAS Management platform installed Managed server, machines and node manager created and configured **Note**: Oracle database is not installed on management image. The database has to be installed separately before activating management administration because of OPSS. When management administration is activating it needs to create the schema for OPSS.
Runtime	Runtime needs an image with SCF template installed AO will add and configure the managed servers, machines and node manager
UDR	Oracle NoSQL Database, NoSQL should not be configured as it will be done by AO Adaptor OCECAS UDR from UDR template AO will add and configure the managed servers, machines and node manager

For more information about the OVA file, see *Oracle Communication Application Orchestrator User Guide*.

About Deployment

Table 7–3 describes the deployment policy that OCECAS enforces for each type of deployment unit.

Table 7–3 OCECAS Deployment Policy

Deployment Unit	Policy
Management	Standalone
Runtime	Standalone. High Availability active-active.
UDR	Standalone. High Availability active-active.

About Bootstrapping

The OCECAS AO adapter facilitates the specific configuration that must be made on a device at the time of deployment for each deployment unit. For more information see "Specifying Boot Parameters" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

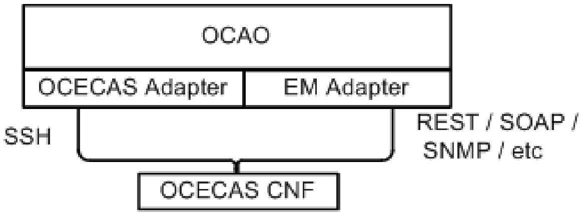
About the AO Adapter

OCECAS provides a customized adapter that plugs into the OCAO framework to interface with OCECAS. The adapter is contained in a **.zip** file that is deployed into the OCAO plugins directory. For OCECAS, this file is **ocecasaoAdaptor.zip**.

The adapter uses the OCCAS-layer interface for communication with the administration server to gather KPI information. This OCCAS layer uses ssh commands to execute WebLogic Scripting Tool scripts that are delivered as part of OCCAS, which is embedded in OCECAS.

Figure 7–3 illustrates the interface between the OCAO adapters and the CNF.

Figure 7–3 Interface Between OCAO and CNF



About Key Performance Indicators

OCECAS makes the following VM-hardware key performance indicators (KPIs) available:

- CPU
- Memory
- Threads
- TCP connections or ports used
- Disk utilization (network I/O)
- JVM statistics

By default, the OCECAS installation configures KPIs for CPU, memory and disk usage. OCECAS configures the engine and UDR VMs to scale on these three statistics. OCECAS does not configure the administration servers or the management VMs to scale.

For information on monitoring and managing overload conditions, see "Monitoring and Managing Overload Conditions" in *Oracle Communications Evolved Communications Application Server System Administrator's Guide*.

About Capacity and Scaling

The OCECAS AO adapter determines what CNF resources are needed for any given input criteria, allowing it to deploy a CNF based on that criteria. Using available key performance indicators, the adaptor determines if capacity is over a threshold criteria and decides whether scaling is required, including whether horizontal or vertical scaling is needed.

8

About Session Control

This chapter describes how Oracle Communications Evolved Communications Application Server (OCECAS) implements session control through several example message flows that illustrate call establishment, SRVCC, and Supplementary Services.

About Session Control

The Session Initiation Protocol (SIP) specifies messages that control communication sessions for voice and video calls over Internet Protocol (IP) networks. These messages establish, terminate, and comprise other required elements of a communication session between two or more endpoints. You can use SIP for communication sessions consisting of one or more media streams. SIP applications can include voice calls, instant messaging, online games, video conferencing, streaming, and file transfer, as well as others.

SIP is a text-based protocol that is standardized in RFC 3261 under the auspices of the Internet Engineering Task Force (IETF).

About Back-to-Back User Agent

OCECAS is an Application Server (AS) in the IMS network. As an application server, the SIP signalling for call establishment can be sent to the AS for originating and terminating sessions (configured as trigger points in the S-CSCF). The OCECAS control flow can react to these call establishment messages by acting as either a Redirect Server or a back-to-back user agent (B2BUA).

As a Redirect Server the control flow makes a decision and allows the call to continue to the original destination or a new destination, or drops the call. In either case, the AS drops out of the call signalling.

By default, the VoLTE control flows operate as a B2BUA. The back-to-back user agent splits the communication channel into two legs and moderates all signaling between the end points, from call setup to termination. For a full description of application server modes, refer to 3GPP TS 24.229 section 5.7.

Normally, OCECAS acts as a back-to-back user agent for all established sessions.

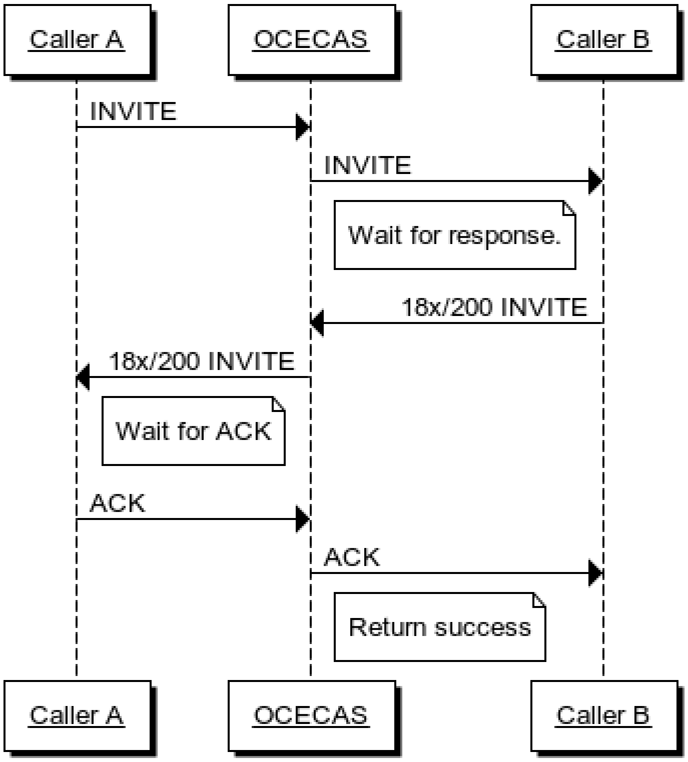
About Session Establishment

The signal flow diagrams in this section illustrate the role of OCECAS in session establishment.

Basic Call

Figure 8–1 shows a basic call between two parties, in which all requests and responses are received and the call setup completes with OCECAS returning a success outcome to fully establish the call.

Figure 8–1 Basic Call



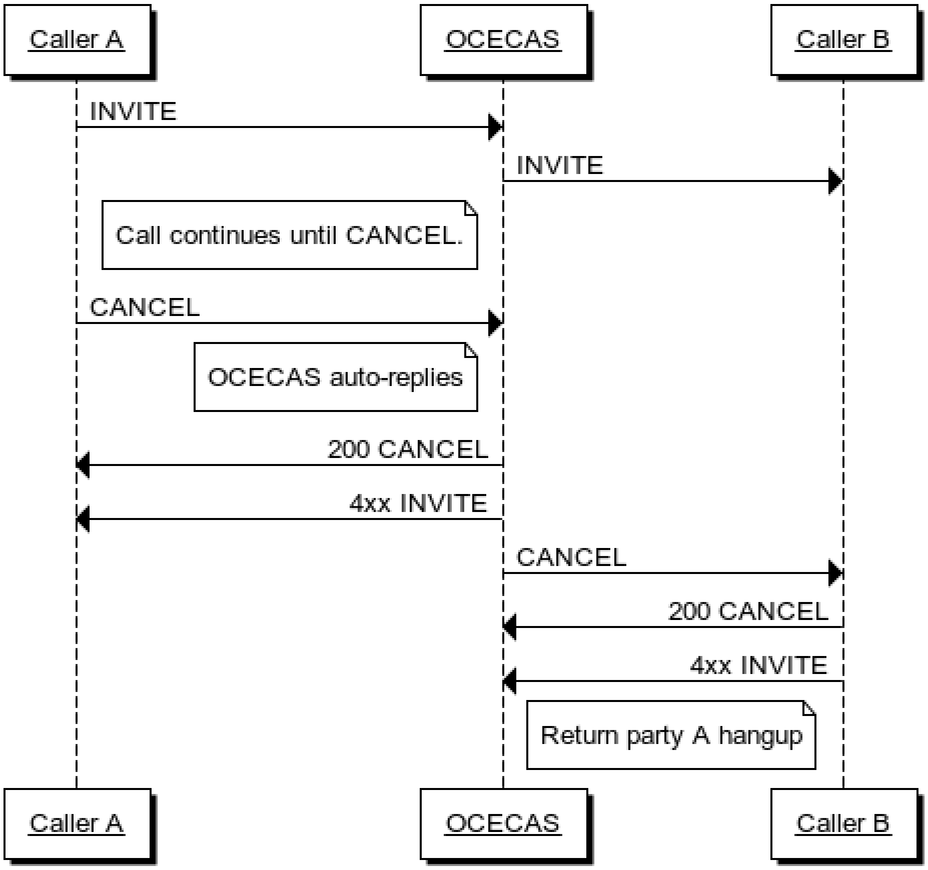
In this sequence, OCECAS takes the following actions:

1. Sends an INVITE message to the destination address.
2. Waits for SIP responses that match the call ID, and the To and From fields of the new call leg.
3. On receipt of 18x response, sends the same response back to the calling party, copying any body content sent by the called party.
4. Waits for ACK from the calling party that matches the call ID and the To and From fields of the original call leg.
5. On receipt of an ACK from the calling party, sends an ACK (success) to the called party. Raises an `Answered` event for the control flow.

Caller Hangs Up

As seen in Figure 8–2, when the caller abandons the call before any significant setup occurs, OCECAS sends the CANCEL message to the called party before a reply to the INVITE occurs and the called party responds accordingly. The exit illustrates that the caller hung up.

Figure 8–2 Caller Abandons Call (Basic)



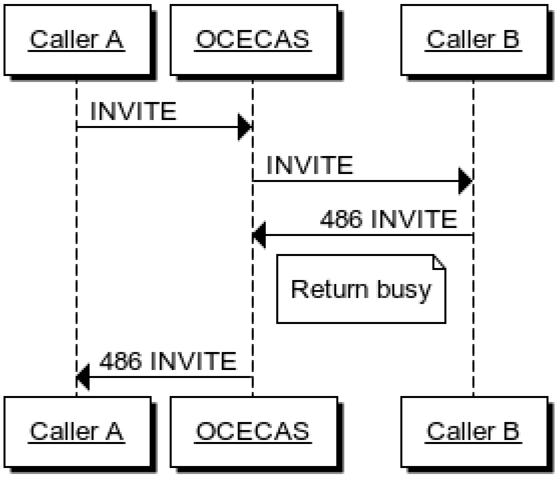
OCECAS takes the following actions when the caller hangs up:

1. Waits for SIP requests that match the call ID, and the To and From fields of the original call leg.
2. On receipt of a CANCEL message from the caller, generates a CANCEL message on the new call leg.
3. On receipt of a CANCEL 200 response from the called party, generates the same response back to the calling party.
4. On receipt of an INVITE 4xx response from the called party, the sequence is complete.

Called Party Busy

When the called party is busy, OCECAS simply returns a status of busy upon receipt of a 486 INVITE message from the called party. Figure 8–3 illustrates this signalling flow.

Figure 8–3 Called Party Busy



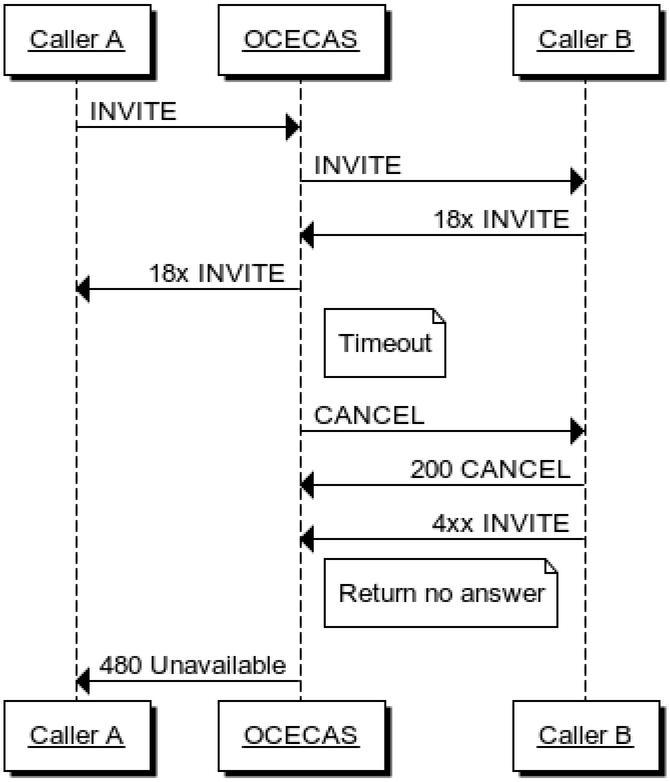
If OCECAS receives a 486 response from the called party, by default it returns a 486 (Busy) to the calling party.

The original SIP session is still active to allow the control flow to perform any further tasks before fully dropping the session. For example the control flow might wish to try to connect to a different party, effectively invoking a back-to-back session with a different destination address.

No Answer

Figure 8–4 illustrates the signalling flow that occurs when the called party does not answer. When the called party does not answer within a specified amount of time, OCECAS sends a CANCEL message. The called party responds and the sequence exits on no answer.

Figure 8–4 No Answer



When OCECAS receives the first 18x response from the called party, it sets a session timer to the number of seconds specified by the Timeout parameter. If the timer expires before it receives the 200 response, it takes the following actions:

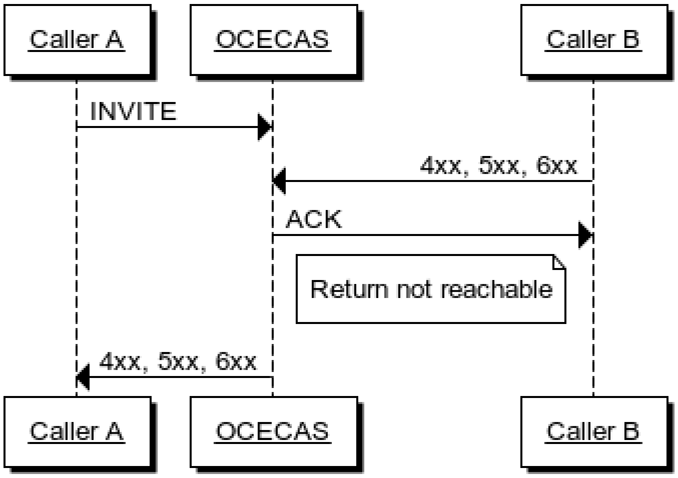
1. Generates a CANCEL on the new call leg.
2. Waits for a CANCEL 200 response from the called party.
3. Raises a No Answer event for the control flow.
4. Returns a 480 (Temporarily Unavailable) to the calling party.

The original SIP session remains active to allow the control flow to perform any additional tasks before dropping the session. For example the control flow might wish to connect to a different party, effectively invoking a back-to-back session with a different destination address.

Called Party Not Reachable

Figure 8–5 illustrates the signalling flow that occurs when the called party is not reachable. When the called party is not reachable, OCECAS handles any 4xx, 5xx or 6xx response (barring a 486) from the called party. OCECAS returns not reachable.

Figure 8–5 Called Party Not Reachable



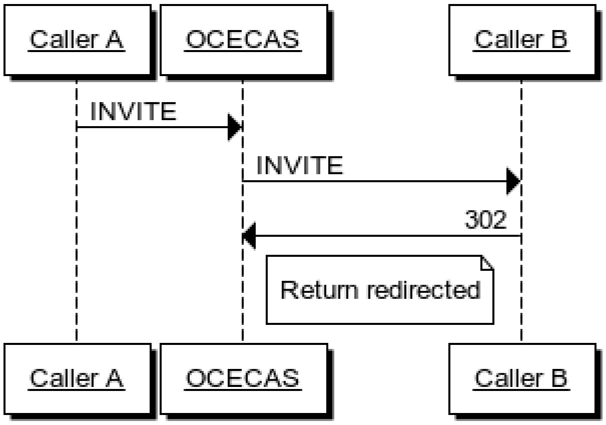
This sequence is similar to the Busy scenario, except that OCECAS receives a fail status (4xx, 5xx, or 6xx) other than 486. In this case, OCECAS returns not-reachable back to the control flow, which exits on the Not Reachable branch. OCECAS sends a fail status (4xx, 5xx, or 6xx) to the calling party.

The original SIP session remains active to allow the control flow to perform any further tasks before fully dropping the session. For example the control flow might wish to connect to a different party, effectively invoking a back-to-back session activity with a different destination address.

Called Party Deflection

Figure 8–6 illustrates a called-party deflection. In this sequence, OCECAS sends a redirect response.

Figure 8–6 Called Party Deflection



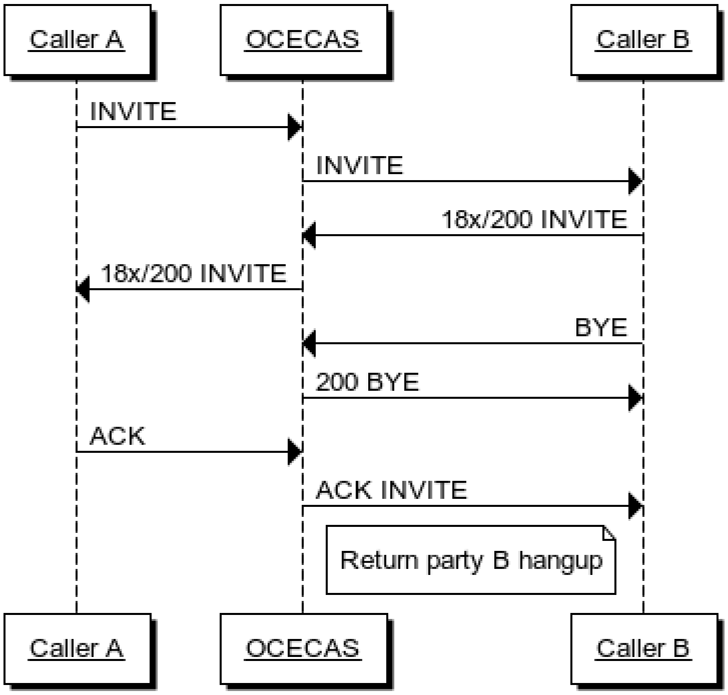
If OCECAS receives a 302 response from the called party, it returns Redirected back to the control flow. It does not send a 302 response to the calling party.

The original SIP session remains active to allow the control flow to act upon the redirection. If it wants to honour the redirection request, it can invoke another back-to-back session, taking the destination address from the Contact SIP header in the 302 response. (In the context map that would be /Incoming/Sip/Response/Header/Contact/Uri). Because the new destination address is different than the original destination address, the control flow first sends a 181 response to the caller to inform the caller about the redirection.

Called Party Ends Call Early

When OCECAS receives an early BYE request from the called party, it does not directly forward it onto the calling party. It responds to the request and the control flow returns on a called party hangup event. Figure 8–7 illustrates the signalling flow.

Figure 8–7 Called Party Ends Call Early



When the called party hangs up before the call is fully established, causing a BYE request before the final ACK is sent from the originator, OCECAS takes the following actions:

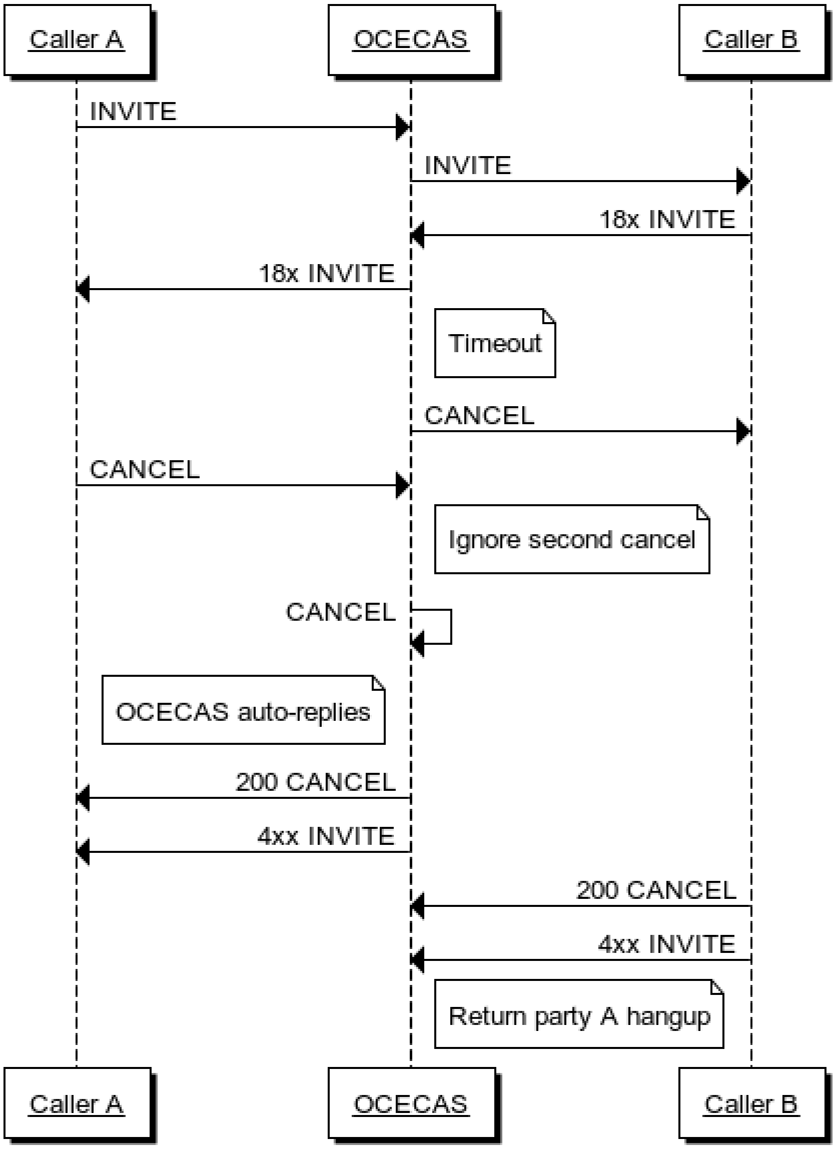
1. Waits for SIP requests matching the call ID and the To and From fields of the new call leg.
2. On receipt of the BYE from the called party, generates a BYE 200 response and returns it on the new call leg.
3. Sets the call state to indicate a BYE has been received from the called party. Raises a "B hungup" event for the control flow.
4. Continues to wait for and send on the ACK from caller to called party.

The original SIP session remains active to allow the control flow to perform any further tasks before fully dropping the session. For example, the control flow could play an announcement to the calling party.

Called Party Does Not Answer or Abandons Call

Figure 8–8 illustrates the scenario in which the called party does not answer or abandons the call. In this scenario, OCECAS responds to a timeout, sending a CANCEL message, and the calling party hanging up. OCECAS ignores the second CANCEL, from the Called party, and waits for a response from its message. OCECAS handles the reply to the Calling Party, sending the 200 CANCEL and 4xx INVITE messages. OCECAS exits showing that the Called Party hung-up.

Figure 8–8 Called Party Does Not Answer or Abandons Call (Advanced)



When OCECAS receives the first 18x response from the called party, it starts a session timer that expires after a specified number of seconds. If the timer expires before the 200 response is received, OCECAS takes these actions:

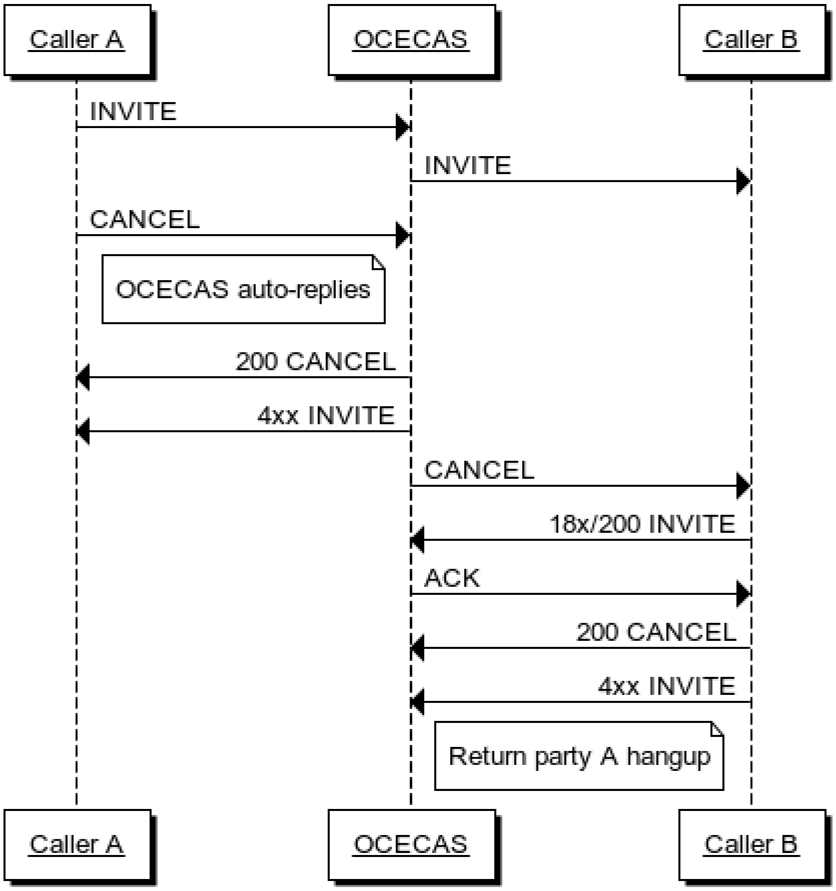
1. Generates a CANCEL message on the new call leg.
2. Waits for a CANCEL 200 response from the called party.
3. Waits for an INVITE 4xx response from called party.
4. Raises a No Answer event for the control flow.

The original SIP session remains active to allow the control flow to perform any further tasks before fully dropping the session. For example the control flow may wish to connect to a different party, effectively invoking another back-to-back activity with a different destination address.

Caller Abandons Call

When the calling party hangs up early but OCECAS subsequently receives a response to the original INVITE, OCECAS handles the replies to the CANCEL for the calling party and sends a CANCEL to the called party. Once it receives the 200 CANCEL and 4xx INVITE, OCECAS exits with a calling party hangup event. Figure 8–9 illustrates this scenario.

Figure 8–9 Caller Abandons Call (Advanced)



When the caller hangs up before the call is established, OCECAS takes these actions:

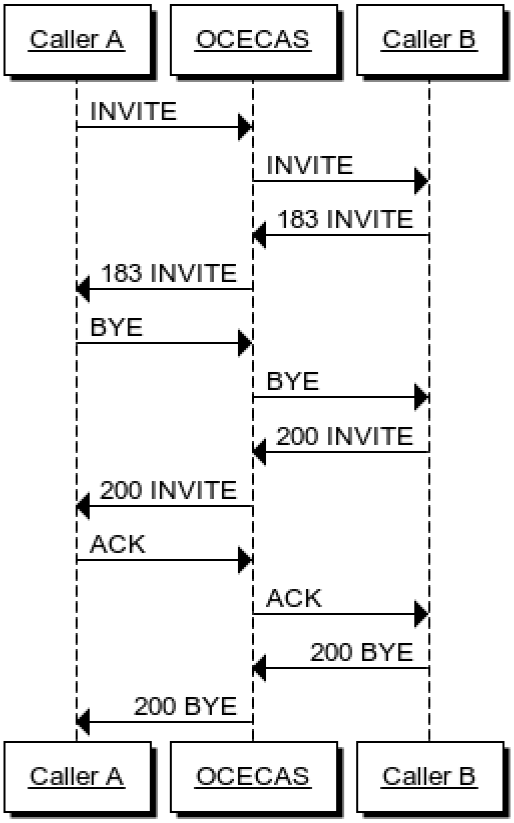
1. Waits for SIP requests matching the call ID and the To and From fields of the original call leg.
2. On receipt of a CANCEL message from the caller, generates a CANCEL message on the new call leg.
3. On receipt of a CANCEL 200 response from the called party, generates the same response back to the calling party.
4. On receipt of INVITE 4xx response from the called party, action is complete.
5. Returns a calling party hungup event to the control flow.

The original SIP session remains active to allow the control flow to perform any further tasks before fully dropping the session. However, because the caller has effectively hung up OCECAS is not able to perform any further interaction with the caller. It is up to the control flow to end the original SIP session with a suitable release cause, such as 487 Request Terminated, when it is ready to do so.

Caller Abandons Call (Early BYE)

Figure 8–10 illustrates an end-to-end BYE call flow, which occurs when a caller finishes before OCECAS receives a 200 message.

Figure 8–10 Caller Abandons Call (Early BYE, Advanced)



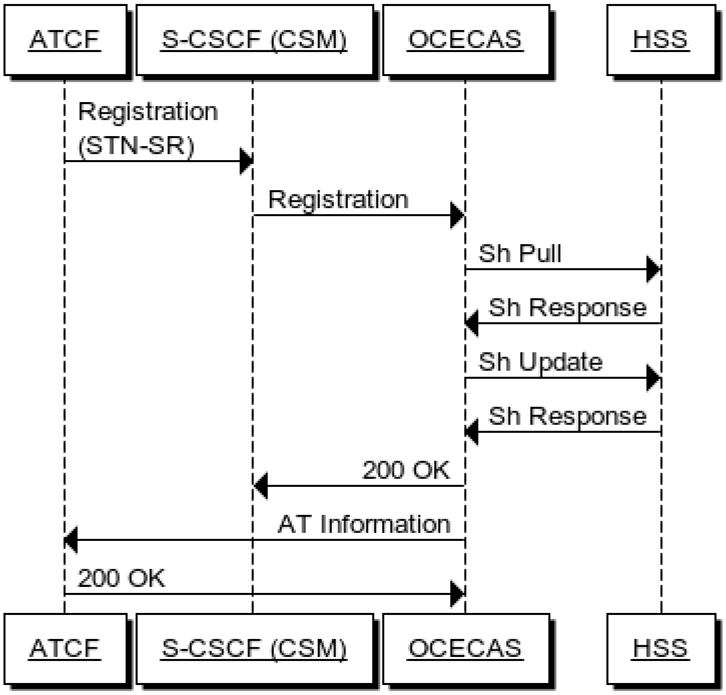
About SRVCC

The following session control flow diagrams illustrate key aspects of the messages exchanged between an Access Transfer Control Function (ATCF) and the Service Centralization and Continuity Application Server (SCC AS).

Registration

The Access Transfer Control Function (ATCF) sends a REGISTER message to S-CSCF with the STN-SR and C-MSISDN. The S-CSCF then sends the SCC AS a third-party REGISTER message. The SCC AS checks this information with the HSS, and possibly updates it, and then sends a MESSAGE directly to the ATCF with the C-MSISDN and ATU-STI. Figure 8–11 illustrates this scenario.

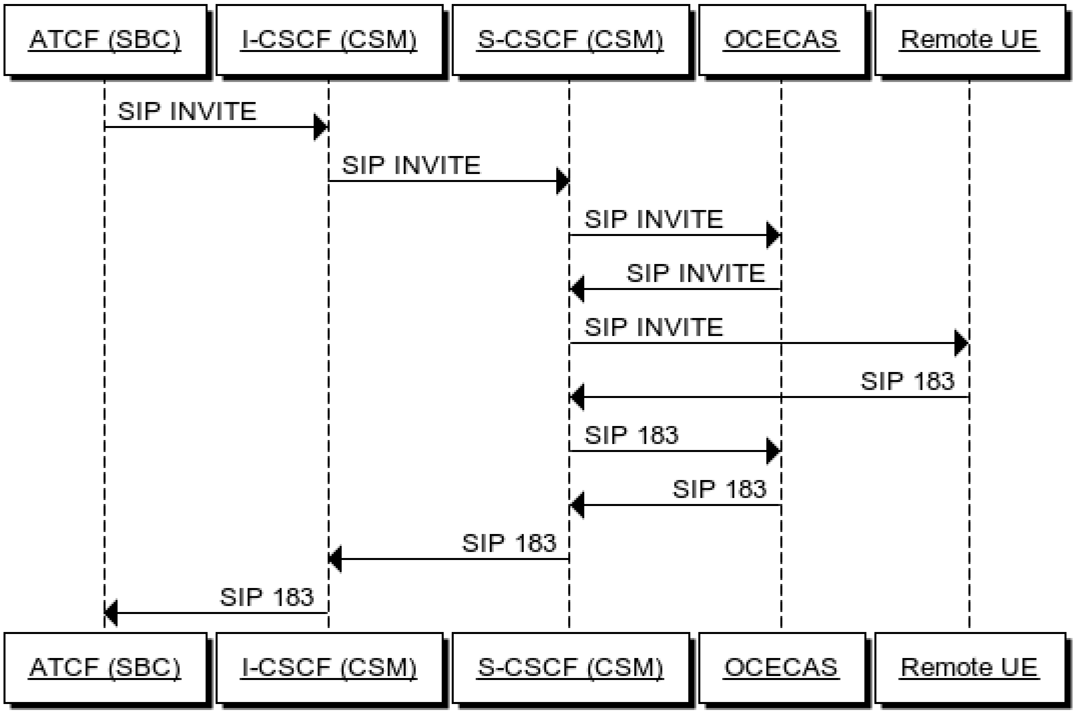
Figure 8–11 Registration



Origination

The SRVCC impact to originate a call is minimal. Figure 8–12 illustrates that as part of normal back-to-back user agent behavior, OCECAS passes the INVITE and Session Progress response through the SCC AS.

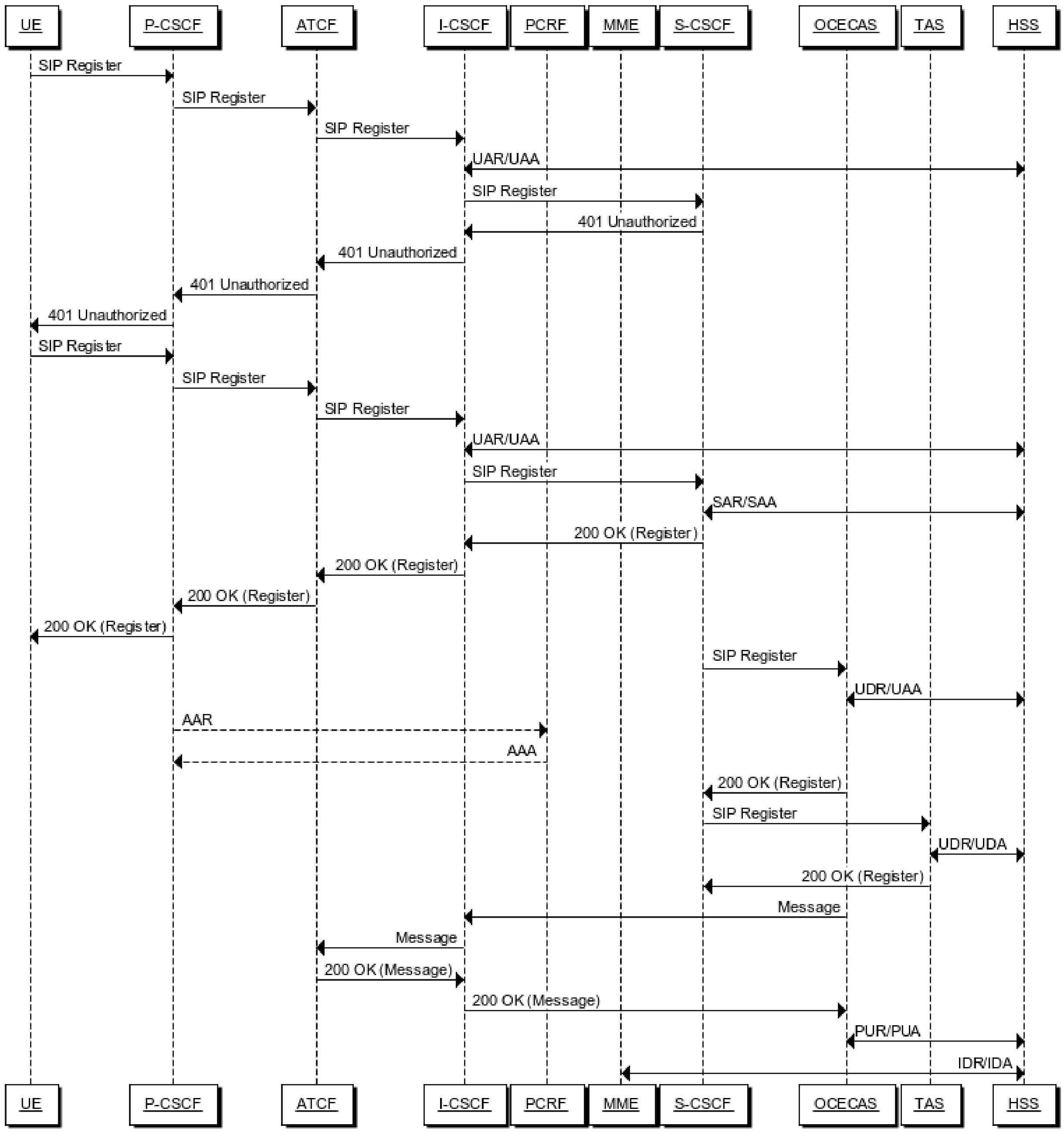
Figure 8–12 Origination



Attachment

Figure 8–13 describes the signalling message flow for attachment.

Figure 8–13 Attachment



Call Establishment - Calling Party

Figure 8–14 and Figure 8–15 describe call establishment for the calling party.

Figure 8–14 Call Establishment - Calling Party

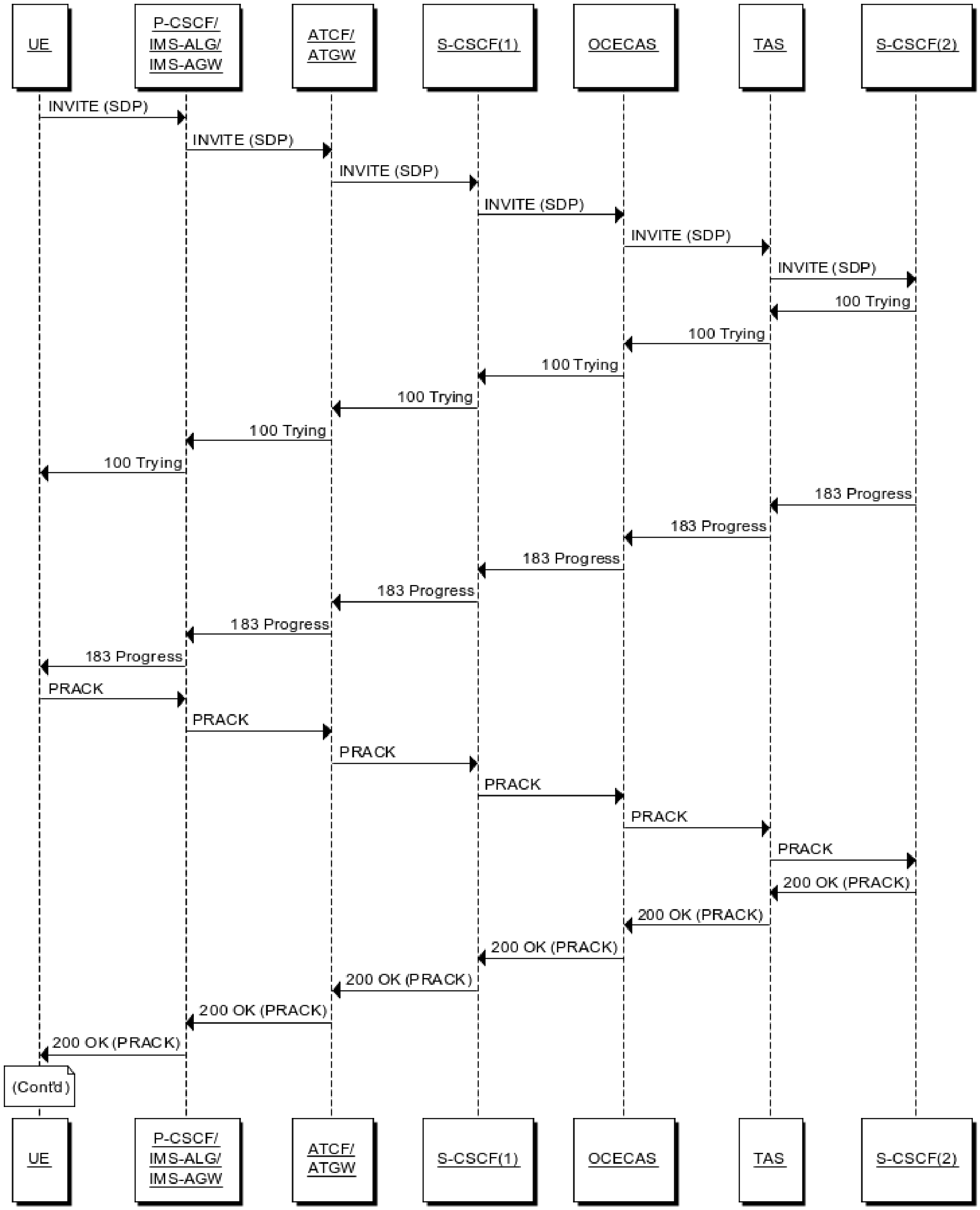
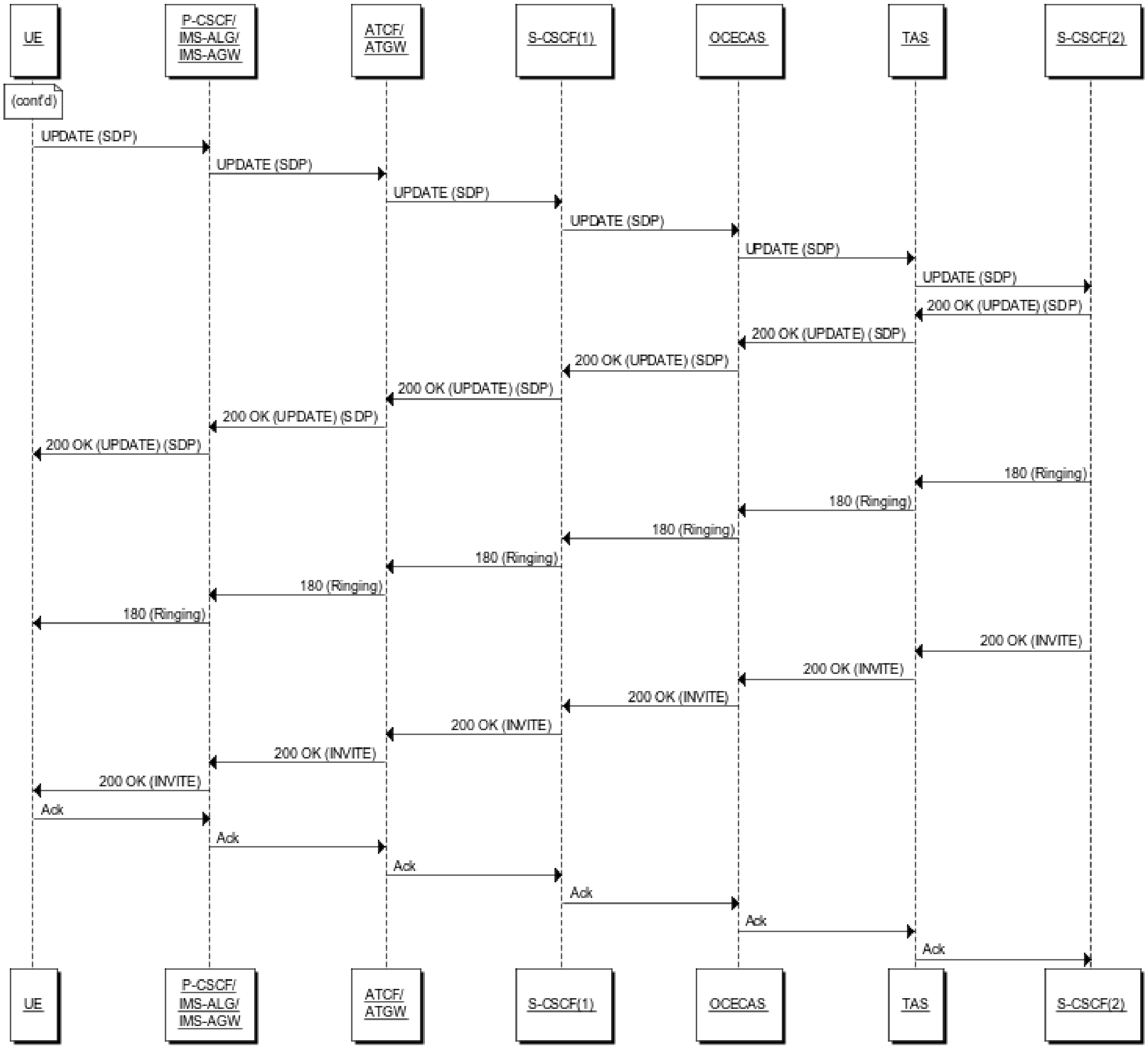


Figure 8–15 Cont'd: Call Establishment - Calling Party



Call Establishment - Called Party

Figure 8–16 and Figure 8–17 describe the signalling message flow for call establishment for the called party.

Figure 8–16 Call Establishment - Called Party

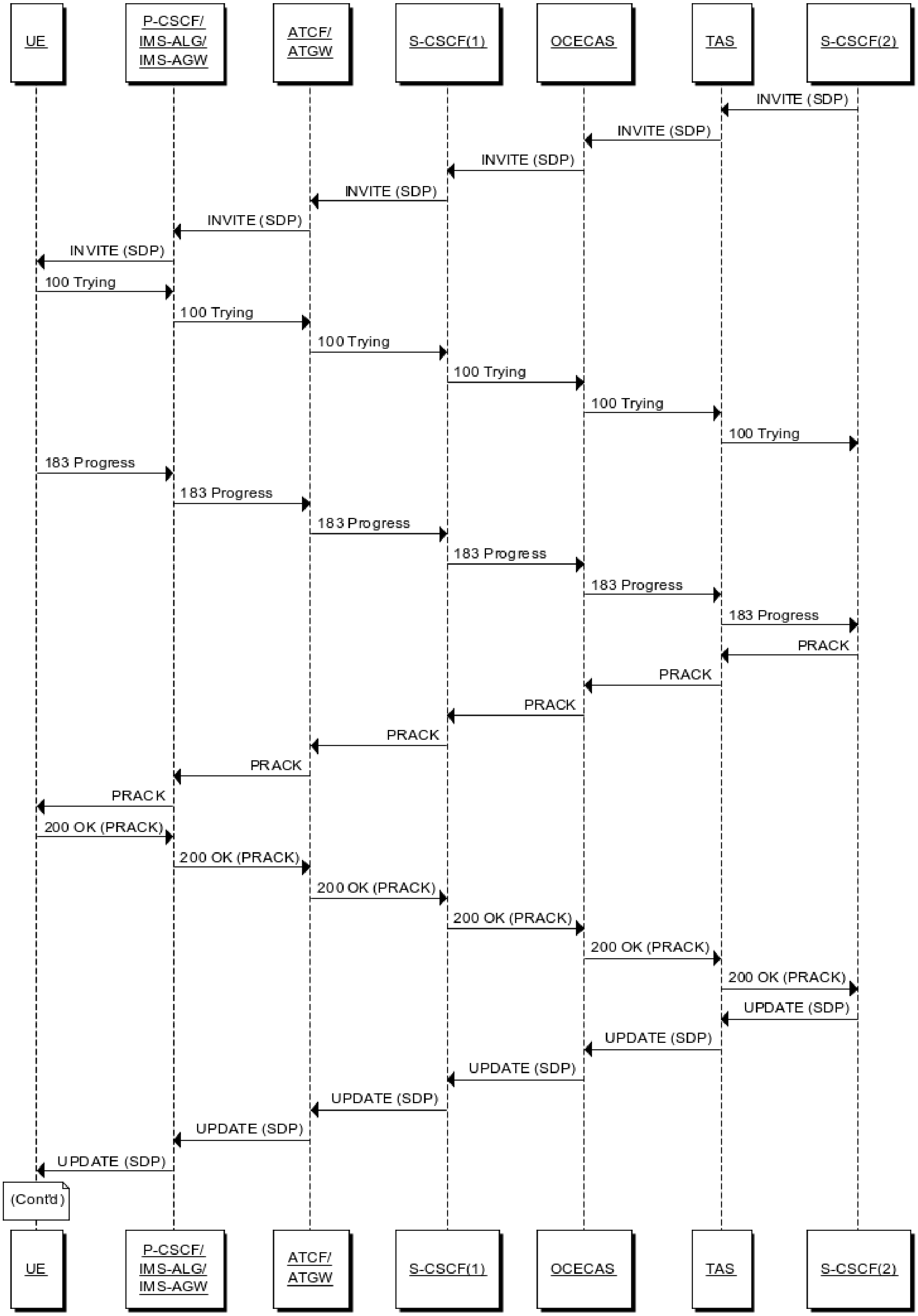
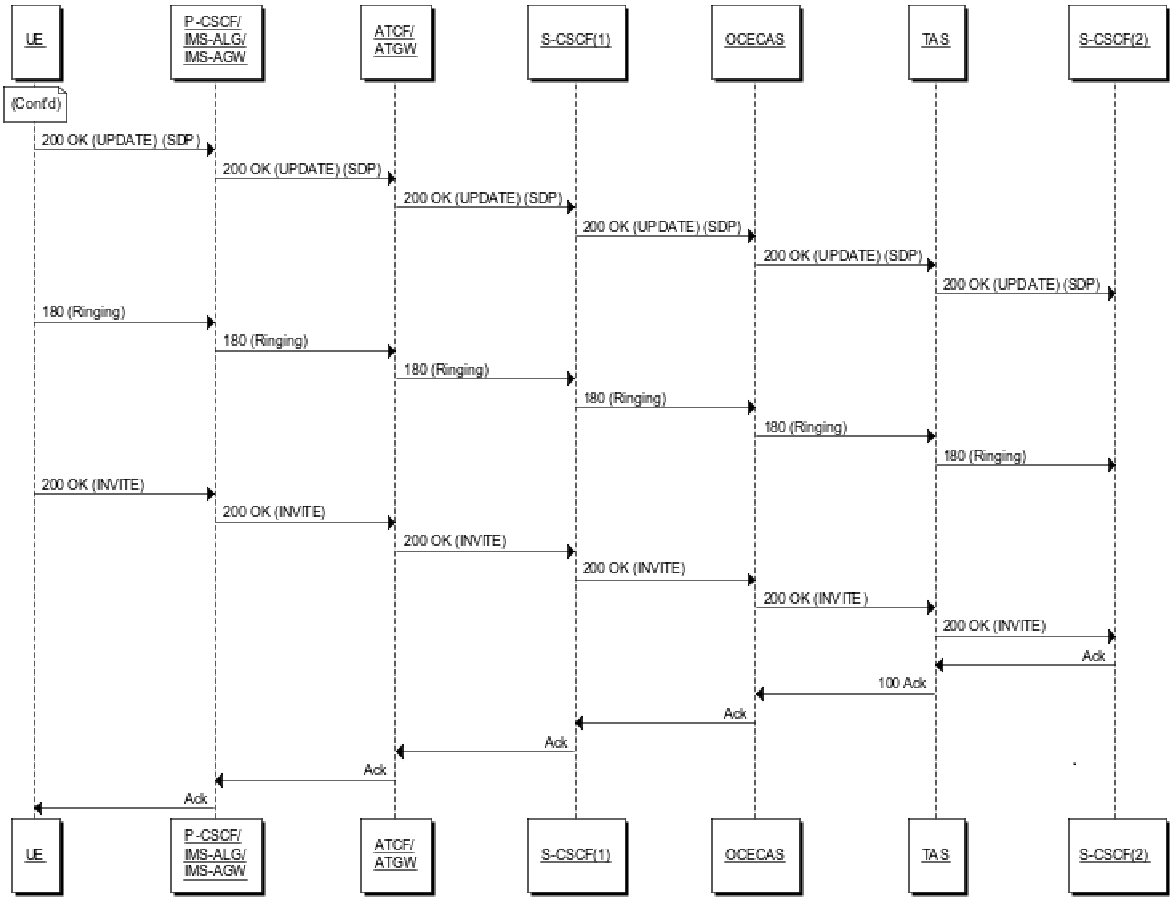


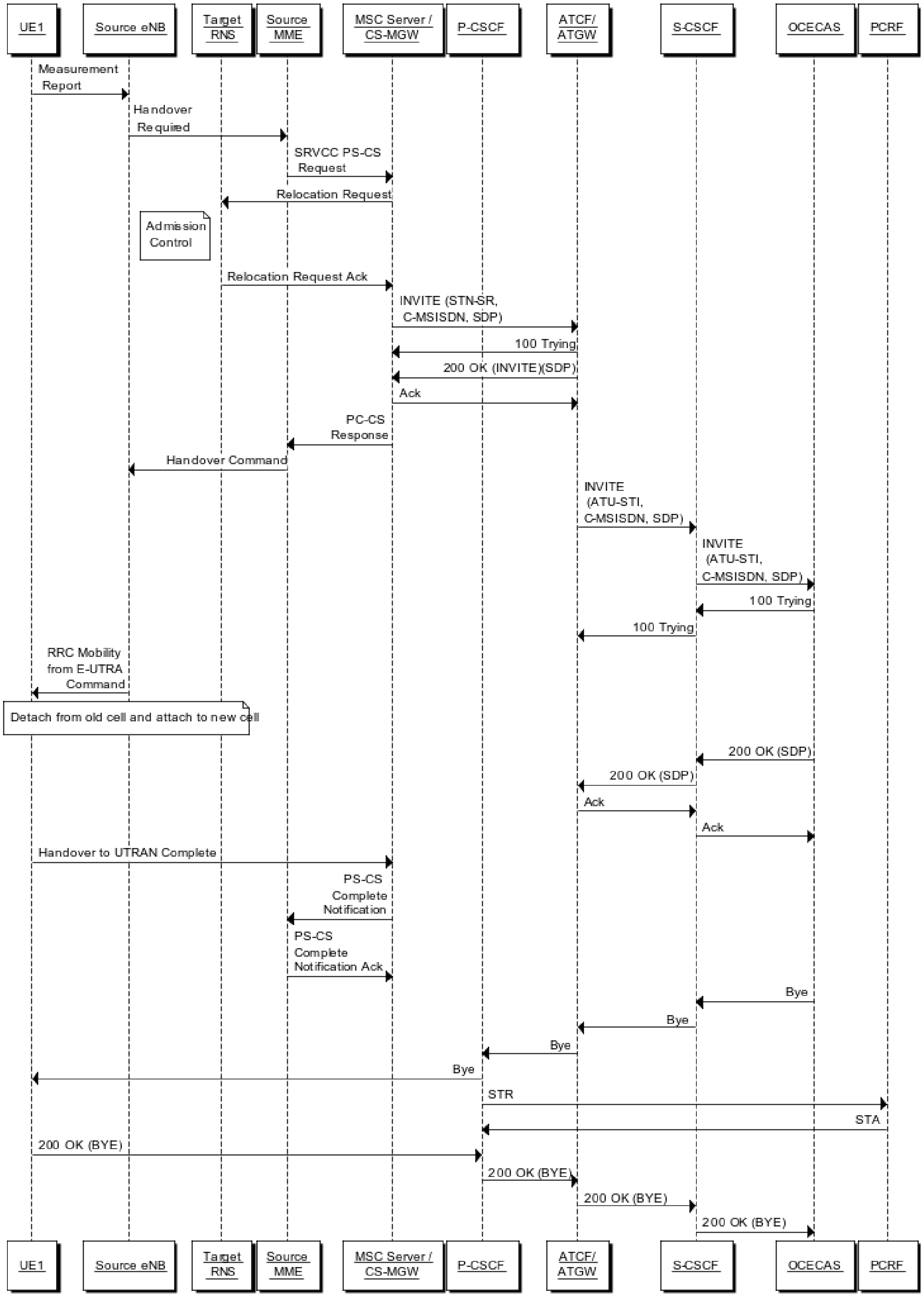
Figure 8–17 Cont'd: Call Establishment - Called Party



Packet-Switched to Circuit-Switched

Figure 8–18 describes the signalling message flow for packet-switched to circuit-switched handoff.

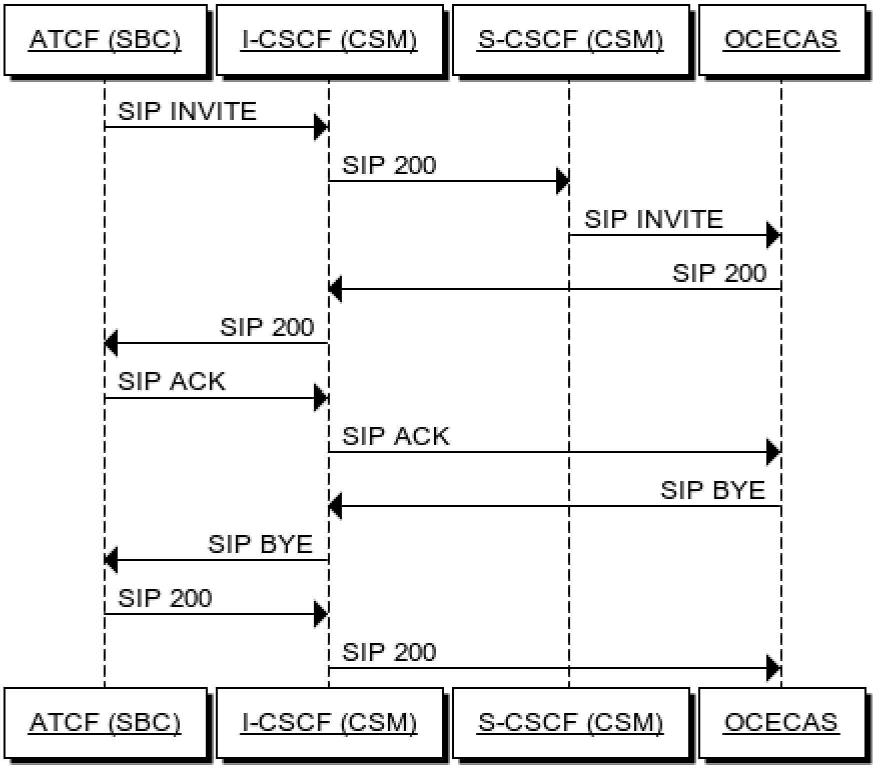
Figure 8–18 Packet-Switched to Circuit-Switched



Packet-Switched to Circuit-Switched with ATGW Anchoring

Figure 8–19 illustrates the signalling flow that occurs when a packet-switched to circuit-switched handoff occurs with ATGW anchoring. In the packet-switched to circuit-switched handoff with ATGW anchoring, the original INVITE that arrives at the SCC AS contains a Target-Dialog header.

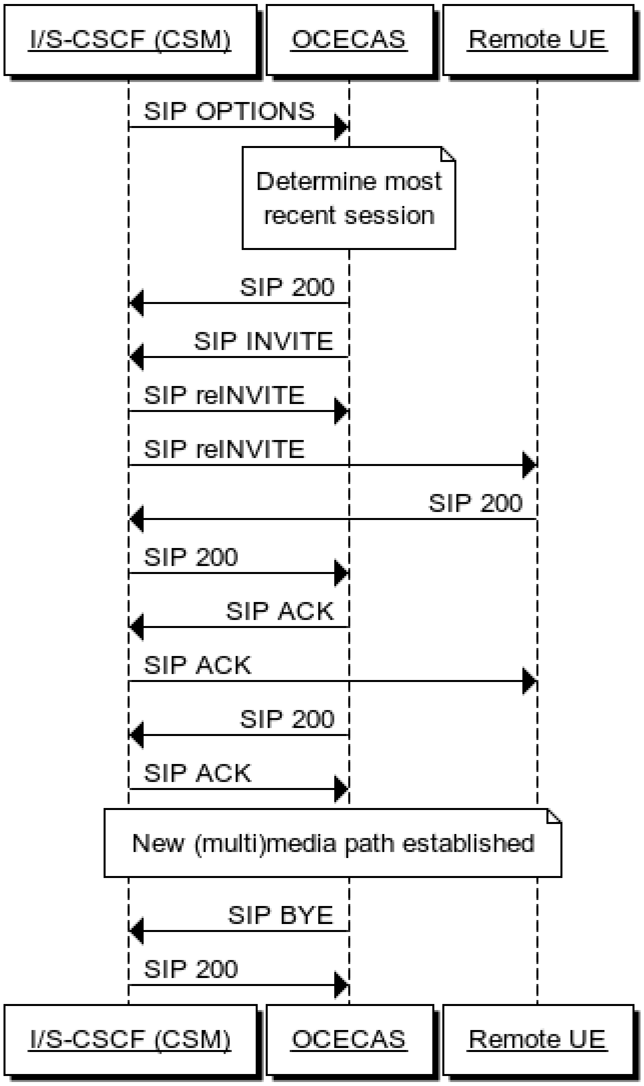
Figure 8–19 Packet-Switched to Circuit-Switched with ATGW Anchoring



Packet-Switched to Circuit-Switched for Speech and Video

VRSCC for video is the same as a standard transfer, except that the MSC server requests information about the latest active session using a SIP OPTIONS message. The SCC AS uses the C-MSISDN to find the latest active session. The OK response contains the SDP information for that session. Figure 8–20 illustrates this scenario.

Figure 8–20 Packet-Switched to Circuit-Switched for Speech and Video



About Supplementary Services

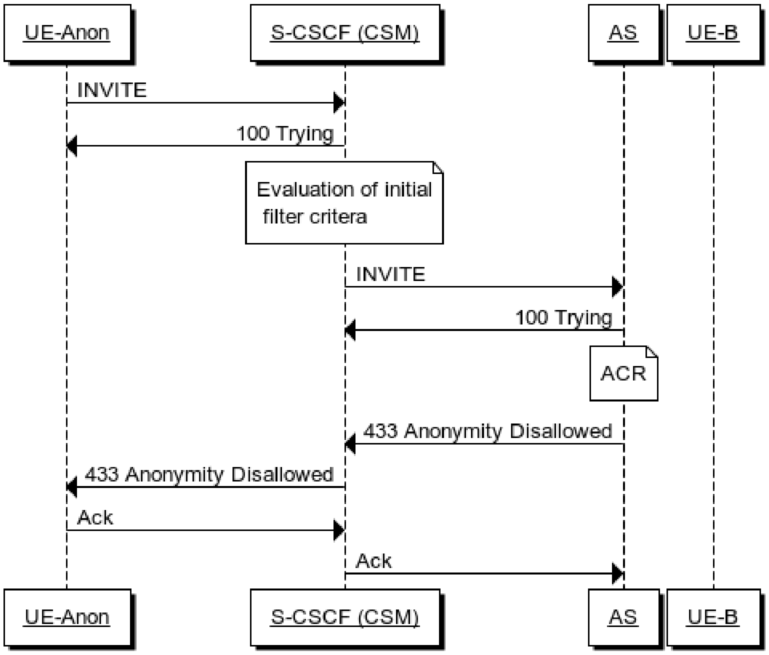
OCECAS provides VoLTE supplementary services as required by the GSMA document "IR.92 IMS Profile for Voice and SMS". These services help to deliver the characteristics of a traditional telephony service for IP transport access.

The following message flows illustrate services for barring of an incoming call, unconditional communication forwarding, and a communication hold.

Barring of Incoming Call

Figure 8–21 shows a signalling flow that illustrates the barring of an anonymous call (UE-Anon) for user endpoint B (UE-B) using the anonymous call rejection (ACR) service:

Figure 8–21 Signal Flow for Barring of Anonymous Call



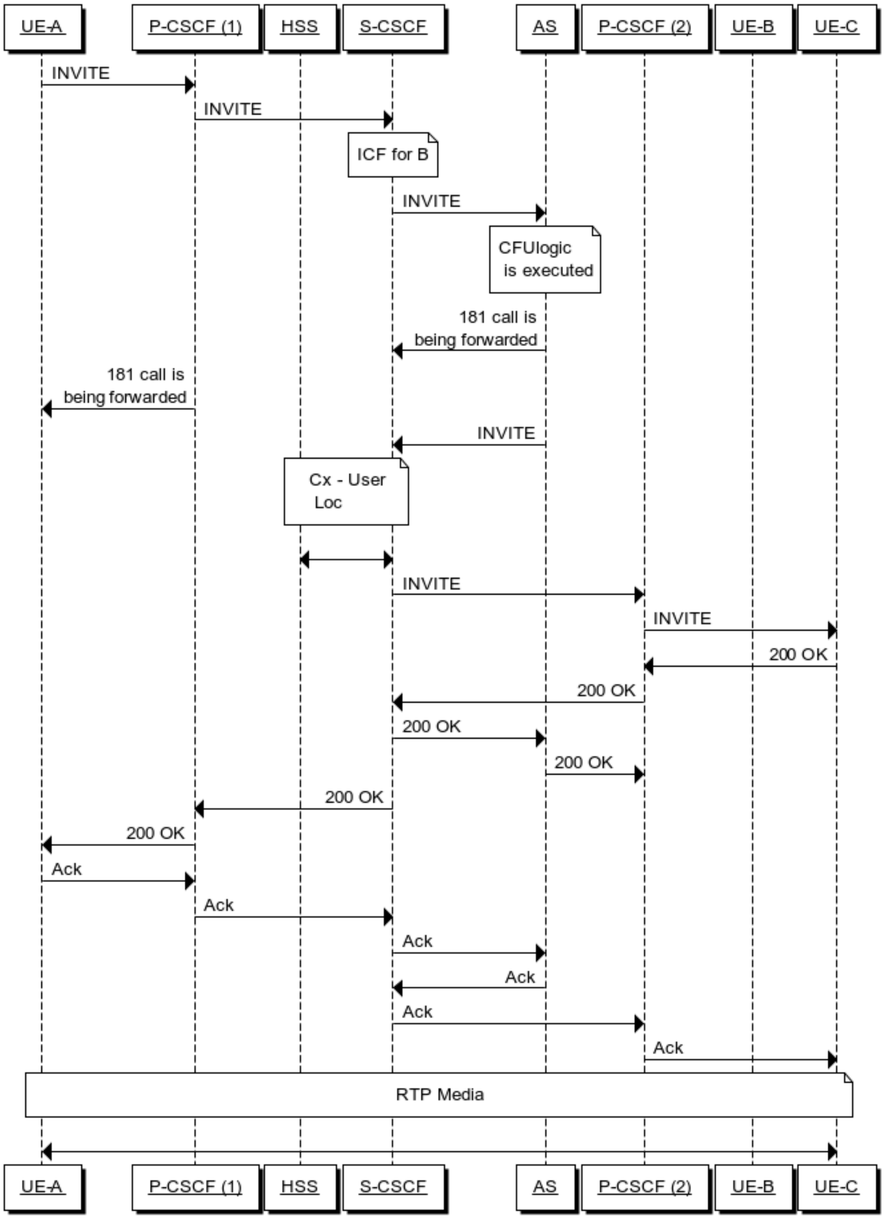
This example of the barring of an incoming anonymous call includes the following sequence of messages and events:

- The incoming INVITE request is sent to the S-CSCF serving user endpoint B.
- The called user subscribes to the Anonymous Call Rejection (ACR) service so the INVITE request is forwarded to the AR AS.
- AS identifies the call as anonymous and sends a 433 (Anonymity Disallowed) response.
- The caller acknowledges (Ack) the final response.

Communication Forwarding

Figure 8–22 shows a sample signalling flow for unconditional communication forwarding with AS providing the forwarding.

Figure 8–22 Unconditional Communication Forwarding



This example of unconditional communication forwarding includes the following sequence of messages and events:

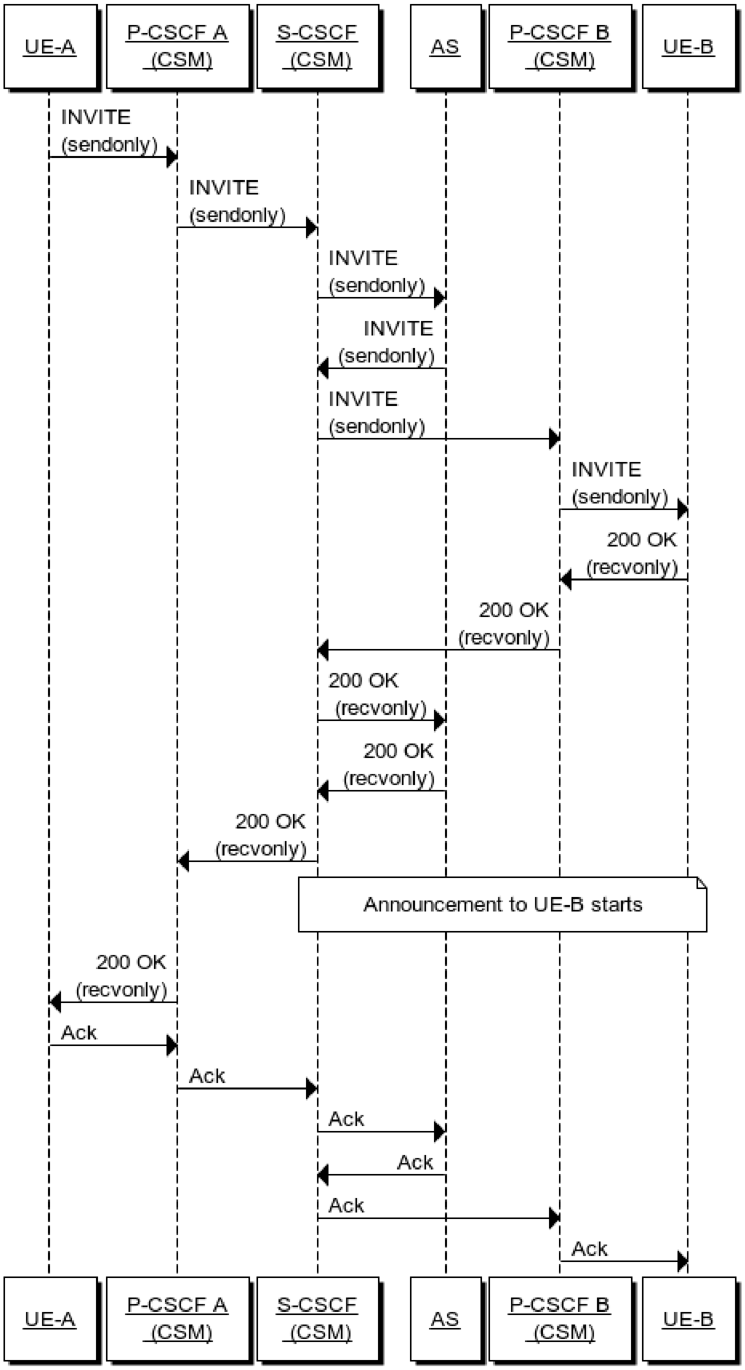
- INVITE request is sent towards user B (UE-B) who has activated.
- The INVITE is forwarded to the AS using the Initial Filter Criteria (IFC).
- Procedures for communication forwarding unconditional (CFU) are executed.
- The caller is notified that communication has been forwarded or deflected; a 181 response (Call is Being Forwarded) is sent to user A.
- An INVITE request for user C is sent back to S-CSCF.
- S-CSCF queries HSS to lookup user C (UE-C) and identify the location.
- Communication is routed to user C.

- The 200 (OK) response is sent to user A (UE-A).
- User A sends an Ack to user B.
- Real-time transport protocol (RTP) media is established.

Communication Hold

Figure 8–23 shows a sample signalling flow for a communication hold.

Figure 8–23 Communication Put on Hold



This example shows a communication session put on hold using a re-INVITE request with AS playing an announcement to the held party.

Glossary

OCECAS refers to Oracle Communications Evolved Communications Application Server in the following entries.

3GPP

The 3rd Generation Partnership Project is a group of telecommunications organizations that cooperates on standards development to provide a reliable environment in which to define their technologies. These technologies include network technologies for cellular telecommunications and encompass radio access, the core transport network, and service capabilities.

activity

An action block that is embodied by an icon that can be dragged and dropped in a control flow. Each activity performs a particular task in a control flow and has one input and a number of outputs, except for the Start and End activities, which have one output and one input respectively.

announcement

A stored message that is played to one of the parties in a voice call session. A sample announcement might be something such as: "The number you have called is no longer in service."

back to back user agent (B2BUA)

A back-to-back user agent (B2BUA) acts both as a user-agent client and a user-agent server in SIP call processing. It maintains the complete call state and participates in all call requests that are sent on dialogs that it has established. OCECAS acts as a back-to-back user agent when it needs to detect changes, or monitor the progress of an on-going call.

change management

The process of managing and deploying change sets across testing, staging, and production environments in OCECAS.

change set

A set of changes to a session or service that is grouped as a single deployable bundle in the OCECAS change management system. A change set can consist of both control flow and configuration changes.

charging data function (CDF)

The function that generates event detail records (EDRs) for offline charging.

charging triggering function

The application server feature that triggers a charging event to either the charging data function for offline charging, or the online charging system.

circuit switched network

A network that employs legacy Signalling System 7 (SS7) protocols such as CAMEL, which are used by 3G networks.

control flow

A logical, flowchart-like structure that consists of linked activities and decisions that specify the management and processing of a voice call or multimedia session. A control flow is created through the OCECAS Session Design Center.

Core Signalling Manager

The software-based signalling components of the IP Multimedia System (IMS) core.

Diameter

The Diameter protocol evolved from the RADIUS protocol and is considered the next generation authentication, authorization, and accounting protocol.

domain

A set of related WebLogic server resources that you manage as a unit. A domain includes an administration server and can include additional managed servers. You use the administration server to configure and manage all resources in the domain. You can define multiple domains based on application requirements, geographic locations and so on. Or you can use a single domain to centralize all WebLogic Server administration activities.

event detail record (EDR)

A written record of actions taken by OCEAS; for example, when an activity is processed in a control flow. The format and content of each event detail record is based on the particular action that generated it and any associated data.

Home Subscriber Server (HSS)

The central store for subscriber data in the IMS network.

IMS and IMS Core

IP multimedia subsystem (IMS) is a standardized framework, originally designed by the 3rd Generation Partnership Project for delivering multimedia content (voice, video, etc) over IP packet switched networks.

IPv6

Internet Protocol version 6.

JSON

JavaScript Objejct Notation (JSON) is a text-based data-interchange format that is programming-language independent and is easy for humans to read and write.

locale

The combination of country and language that identifies the originating or terminating location in a voice call or multimedia session.

Network Functions Virtualization (NFV)

A network architecture that virtualizes network node functions to create communication services from discrete building blocks. A virtualized network function allows you to perform a network function in place of a custom hardware appliance.

NoSQL

Stands for *not only* SQL. Its principal characteristic is its non-conformance to the relational database model. It is a database management and design model that is intended to scale and perform better for very large data storage applications.

Oracle Universal installer (OUI)

Oracle Universal Installer is a Java-based installer application that enables you to install Oracle applications and components using a graphical user interface.

packet switched network

A network that delivers Internet Protocol (IP) based communications.

REST

An acronym for representational state transfer, it is a style of network architecture that imposes a set of constraints on components, connections and data in the provision of services in a distributed hypermedia network system. The World Wide Web, for example, is said to represent the REST architectural style. For more information, see the *W3C REST* web page.

RESTful API

An API used in web services that conform to the REST architectural constraints. Typically, RESTful APIs use JSON or XML for data interchange and standard HTTP methods such as GET, PUT, POST, and DELETE.

Service Centralization and Continuity Application Server (SCCAS)

Provides anchoring of calls in the IMS home network as part of SRVCC.

service provider

An organization that uses OCECAS to provide voice call and multimedia session-processing services to its customers.

session border controller

A device that interacts with OCECAS in Voice over Internet Protocol (VoIP) networks in setting up, processing, and tearing down secure, SIP-based interactive communication sessions.

Short Message Service (SMS)

A service in a telecommunications system that sends short messages of up to 160 characters in length, commonly known as text messages. Messages are typically exchanged between mobile devices.

Single Radio Voice Call Continuity (SRVCC)

Provides for the transfer of a call from a circuit switched network to a packet switched network.

SIP

The Session Initiation Protocol (SIP) works with other protocols in the creation and management of real-time multimedia communication sessions between Internet endpoints such as mobile devices. For a complete description, see *SIP: Session Initiation Protocol* by the Internet Engineering Task Force (ietf).

Supplementary Services

A set of services that support a VoLTE network by delivering the telephony services that subscribers expect.

UDR

A user database repository (UDR) consists of subscriber data, including information about services to which the subscriber has subscribed. The data can reside in multiple data sources that are accessed and presented through a defined view.

Unified Session Manager (USM)

Provides IMS core and session border controller.

value added service (VAS)

A service built and delivered to add additional value to an end subscribers experience.

VoIP

Voice over Internet Protocol (VoIP) provides voice and multimedia sessions over IP networks rather than the public switched telephone network (PSTN).

VoLTE

Voice over Long Term Evolution provides voice and video calls over an LTE system, which was originally regarded as IP system for carrying data. VoLTE uses IMS to be part of a rich media solution. VoLTE is defined by the GSM Association (GSMA). For more information, see *VoLTE Service Description and Implementation Guidelines* by GSMA.

VoWIFI

VoWiFi provides Wi-Fi based voice over IP service for devices such as personal digital assistants (PDAs) and laptops.

WebRTC

Browser-based Web Real-Time Communication technology.