unify

Harmonize
your enterprise

CSTA Overview

OpenScape Voice (OSV)
July 2010

Presentation Overview

- CSTA protocol and standards overview
- OpenScape Voice architectural overview for CSTA

CSTA Protocol Standards

What is CSTA?

- CSTA = Computer Supported Telecommunications Applications
- Basic call model standardized in 1992 – continually refined and enhanced based upon significant industry implementation experiences, new protocols, etc.
- CSTA specifies an Applications Interface for monitoring and controlling calls, and devices in a communications network
- CSTA creates an abstraction layer for telecomm applications:
  - CSTA is independent of any underlying signaling protocols
    - E.g. H.323, SIP, Analog, T1, ISDN, etc.
  - CSTA is independent of the devices chosen by users
    - E.g. intelligent endpoints, low-function/stimulus devices, SIP Signaling models - 3PCC vs. Peer/Peer

What is CSTA?

- Equally well suited for different modes of operation:
  - 3rd-party call control
  - 1st-party call control
- Design goal of CSTA is to enhance portability across various CSTA implementations:
  - Specifies normalized call model and behavior
  - Complete functional definition of Services, Events
  - Specific conformance criteria

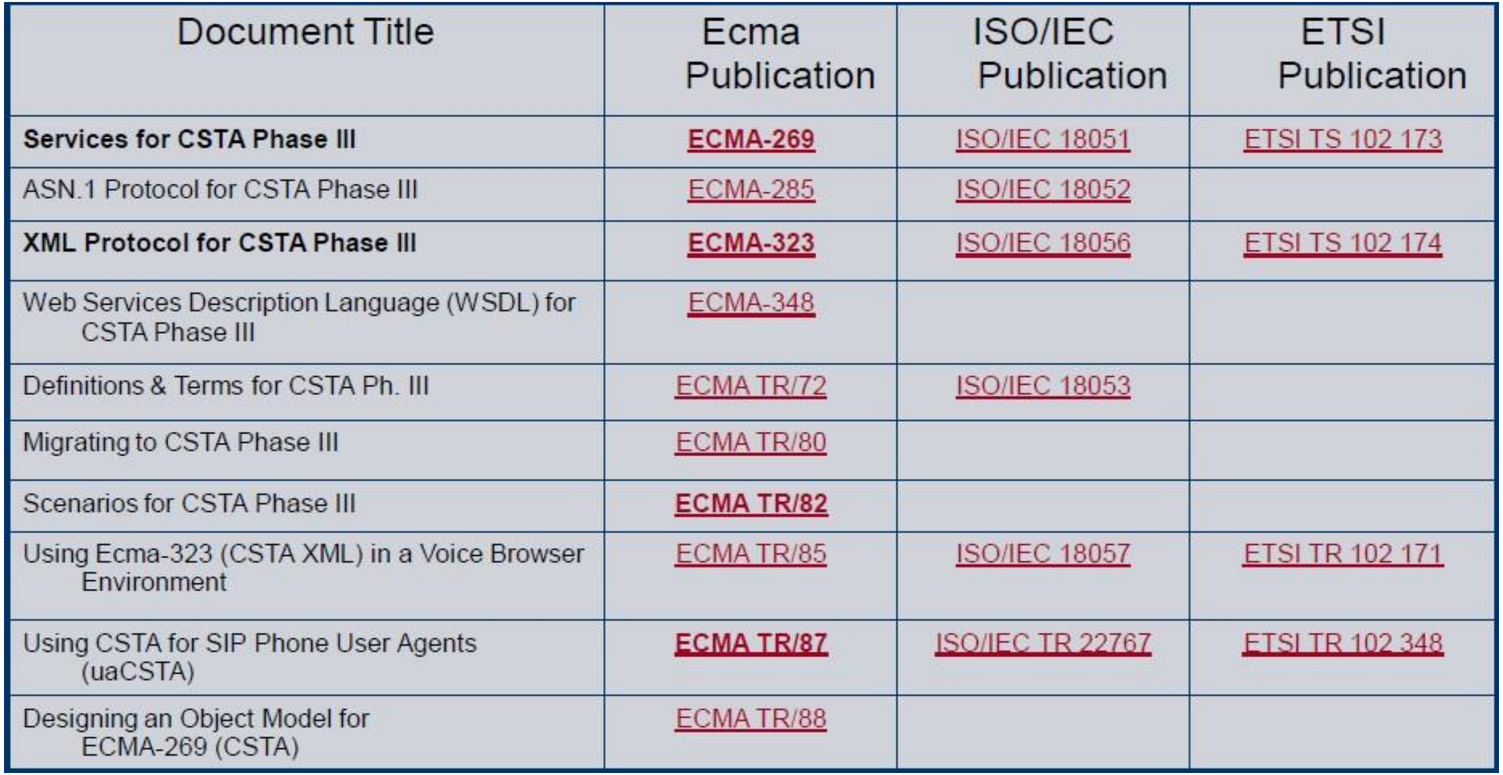
CSTA Standards - Evolution

- Phase I (published June ’92)
  - 40 features, 66 pages (Services Specification)
  - focus on call control
- Phase II (published Dec. ’94)
  - 77 features, 145 pages (Services Specification)
  - I/O & Voice Unit Services, more call control services
- Phase III - CSTA Phase II Features & versit CTI Technology
  - Published Dec. ‘98
  - 136 features, 650 pages (Services Specification)
  - Published as an ISO Standard July 2000
  - Published CSTA XML (ECMA-323) June 2004
  - Published “Using CSTA with Voice Browsers” (TR/85) Dec. 02
  - Published CSTA WSDL (ECMA-348) June 2004

CSTA Standards - Evolution

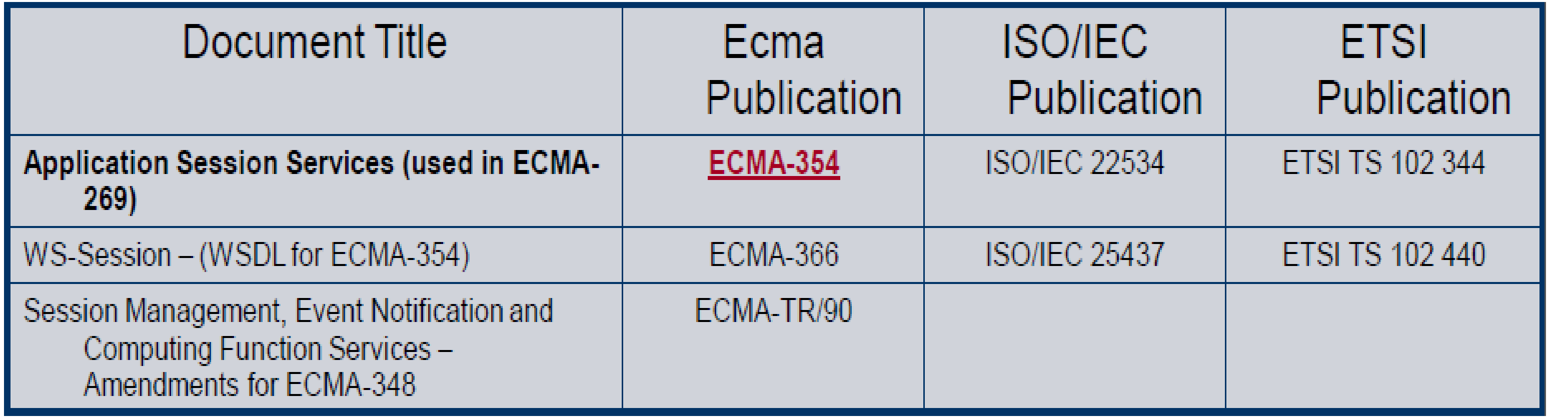
- June 2004: Published Object Model TR/88
- June 2004: Published “Using CSTA for SIP Phone User Agents (uaCSTA)” TR/87
- June 2004: Published “Application Session Services” (ECMA-354)
- June 2005: Published “WS-Session: WSDL for ECMA-354” (ECMA-366)
- December 2005 : Published “Management Notification and Computing Function Services”
- December 2005 : Session Management, Event Notification, Amendements for ECMA-348” (TR/90)
- December 2006 : Published new editions of ECMA-269, ECMA-323, ECMA-348

CSTA Standards Documents



Document Title	Ecma Publication	ISO/IEC Publication	ETSI Publication
Services for CSTA Phase III	**ECMA-269**	ISO/IEC 18051	ETSI TS 102 173
ASN.1 Protocol for CSTA Phase III	ECMA-285	ISO/IEC 18052	
XML Protocol for CSTA Phase III	**ECMA-323**	ISO/IEC 18056	ETSI TS 102 174
Web Services Description Language (WSDL) for CSTA Phase III	ECMA-348		
Definitions & Terms for CSTA Ph. III	ECMA TR/72	ISO/IEC 18053	
Migrating to CSTA Phase III	ECMA TR/80		
Scenarios for CSTA Phase III	**ECMA TR/82**		
Using Ecma-323 (CSTA XML) in a Voice Browser Environment	ECMA TR/85	ISO/IEC 18057	ETSI TR 102 171
Using CSTA for SIP Phone User Agents (uaCSTA)	**ECMA TR/87**	ISO/IEC TR 22767	ETSI TR 102 348
Designing an Object Model for ECMA-269 (CSTA)	ECMA TR/88		

CSTA Standards Documents



Document Title	Ecma Publication	ISO/IEC Publication	ETSI Publication
Application Session Services (used in ECMA-269)	**ECMA-354**	ISO/IEC 22534	ETSI TS 102 344
WS-Session – (WSDL for ECMA-354)	ECMA-366	ISO/IEC 25437	ETSI TS 102 440
Session Management, Event Notification and Computing Function Services – Amendments for ECMA-348	ECMA-TR/90		

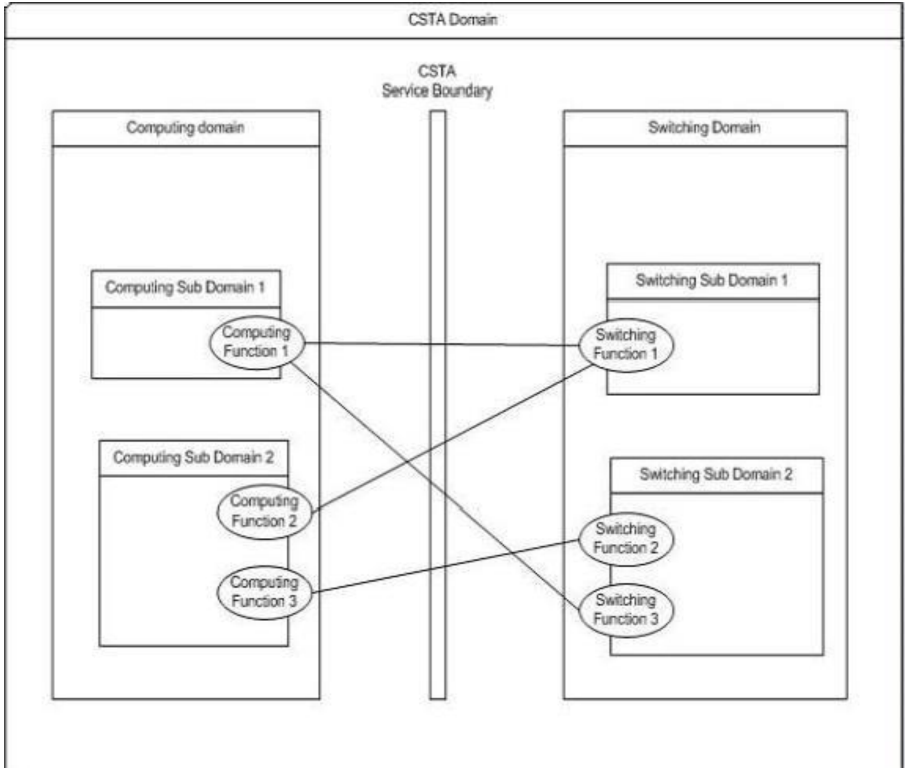
CSTA Standards: Extensibility

- *Additional features can be added to the standard via new editions of the standard*
- *New Parameters, new values to existing parameters can be added via new editions of the standard*
- *Backward compatibility must be maintained in any future editions*
- *Implementations can add proprietary features using built-in CSTA extension mechanisms (e.g. ONS – One Number Service)*

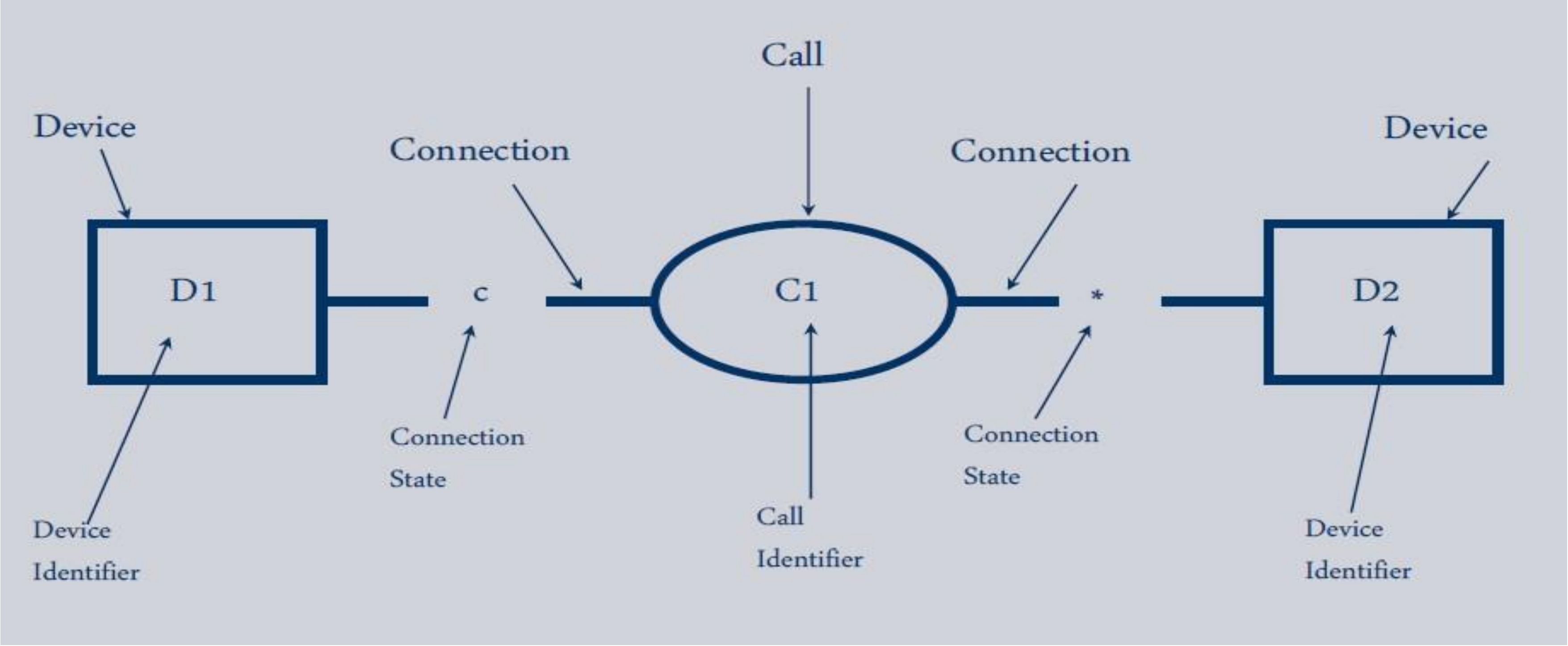
CSTA Operational Model

- CSTA operation model consists of computing and switching domains / sub-domains. The CSTA standard defines the interface between the 2 domain types.
- CSTA Standard specifies the messages (Services and Event Reports), and their associated behavior.
- **Computing Domain** – is the environment/host for CSTA applications that interact with and controls the switching domains.
- **Switching Domain** – The CSTA Model provides an abstract view of a part of a switching sub-domain that an application can control and/or observe. The Switching Domain includes objects such as CSTA calls, devices and connections.

CSTA Operational Model
CSTA Domain



CSTA Operational Model:
Calls, Devices and Connections



Call Control Model: Connection States

- **Alerting** – Indicates an incoming call at an endpoint. Typically the connection may be ringing or it may be in a pre-alerting (e.g. offered) condition.
- **Connected** – Indicates that a connection is actively participating in a call. This connection state can be the result of an incoming or outgoing call.
- **Failed** – Indicates that call progression has stalled. Typically this could represent that an outgoing call attempt that encountered a busy endpoint.
- **Held** – Indicates that an endpoint is no longer actively participating in a call. For implementations that support multiple calls per endpoint (i.e. line), a connection could be Held while the line is used to place another call (consultation transfer on an analog line, for example).
- **Initiated** – A transient state, usually indicating that the endpoint is initiating a service (e.g. dial tone) or the device is being prompted to go off hook.
- **Null** – There is no relationship between the call and the endpoint.

Standardized CSTA Features
ECMA-269, ECMA-323, ECMA-348

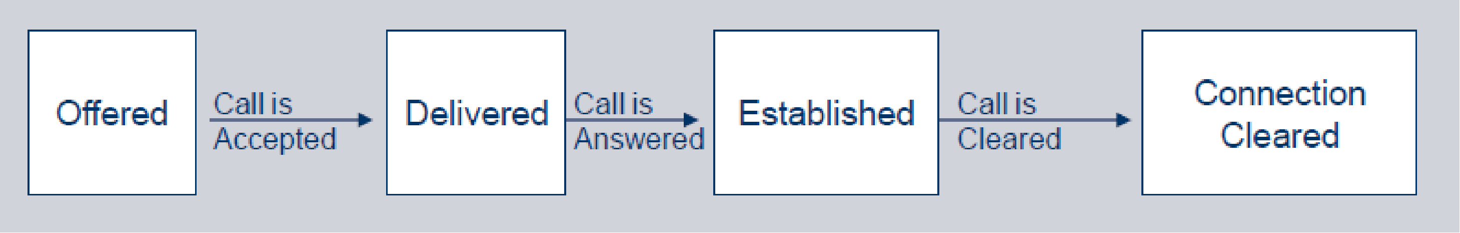
- 26 Call Control features (making call, answering call, etc.)
- 6 Call Associated features (sending user data, etc.)
- 19 Logical Device features (do not disturb, forwarding, etc.)
- 23 Physical Device features (writing to device display, etc.)
- 5 Capability Exchange features (feature discovery, etc.)
- 4 Snapshot features (query existing calls at a device, etc.)
- 3 Monitor features (subscribing to event reports, etc.)
- 17 Voice Services (for Listener, DTMF, Prompt and message resources)
- Other: Routing services, Media Attachment services, Maintenance services, Data Collection services, Accounting services, etc.
- *Any given CSTA implementation does not need to support all of these features to be CSTA compliant! (Profiles)*

Features: Call Control Overview

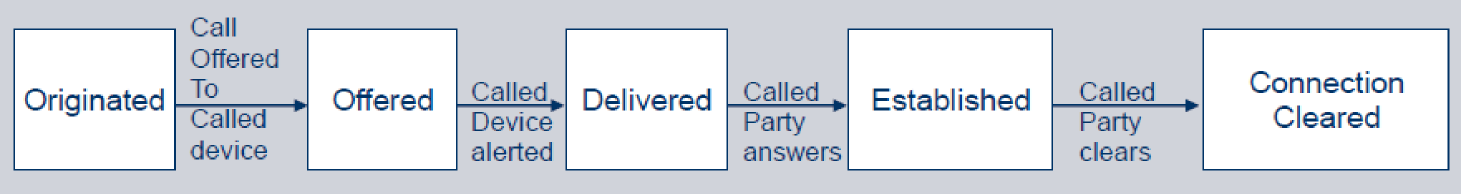
- Exhaustive set of standardized call control features:
- **Services**: Accept, Alternate, Answer, Call Back, Call Back Message, Camp On, Clear Call, Clear Connection, Conference, Consultation, Deflect, Dial Digits, Directed Pick up, Group Pick Up, Hold, Intrude, Join, Make Call, Park, Retrieve, Send Message, Single Step Conference, Single Step Transfer, Transfer.
- **Events**: Bridged, Call Cleared, Conferenced, Connection Cleared, Delivered, Digits Dialed, Diverted, Established, Failed, Held, Network Capabilities Changed, Network Reached, Offered, Originated, Queued, Retrieved, Service Initiated, Transferred.
- Each feature includes a complete operational model on how the feature works, from the applications perspective.
  - Initial and final conditions
  - State transitions
  - Event flows
  - Textual behavior description

Features: Event Sequences

- Typical call events for an incoming call



- Typical call events for an outgoing call (Make Call)



CSTA Application Examples

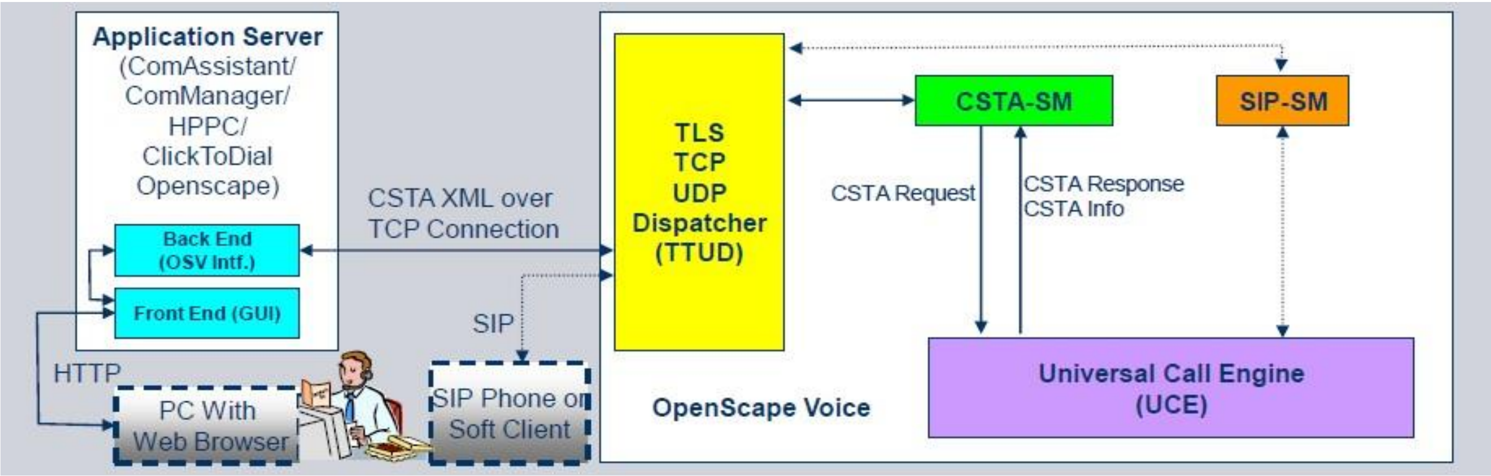
- Personal telephone support, dialers, phone control, soft-phone, etc.
- Multi-media messaging (IM, SMS, MMS, Email, voice mail)
- Inbound & outbound telemarketing, call centers
- Conferencing & collaboration
- User/agent statistics/reporting
- Personal assistant, rules based routing
- Auto-attendant
- Voice browsers, advanced speech, DTMF
- Presence, availability, device context
- Call logging, accounting, billing
- Phone as UI for apps (button press, displays) for hospitality
- Routing/distributing calls (ACD)
- And more…

OpenScape Voice
CSTA Architectural Overview

Topics

- **Architectural Overview and CSTA Features on OpenScape Voice**
- **CSTA Signaling Manager**
- **CSTA Service**
- **uaCSTA**

OSV architectural overview for CSTA: Client (CSTA Application)



- Different Application Servers may connect and monitor/control OpenScape Voice devices via a CSTA connection. XML is used for message encoding, and TCP is used as transport.
  - Up to 16 TCP connections supported (default 4)
- The Application Server is required/responsible to establish and maintain the connection with OSV
  - If the connection is lost the Application Server must reestablish it
  - If the connection is lost all Monitors on OpenScape Voice devices will be lost unless the application supports Application Session Layer (ECMA-354).

CSTA Service
Supported CallP Features (V3.1)

- Make Call Service
- Accept Call Service
- Answer Call Service
- Clear Connection Service
- Consultation Call Service
- Hold/Retrieve Call Service
- Alternate Calls Service
- Reconnect Service
- Deflect incoming call
- Transfer Call (Blind, Consultation, SingleStep, Transfer to VoiceMail)

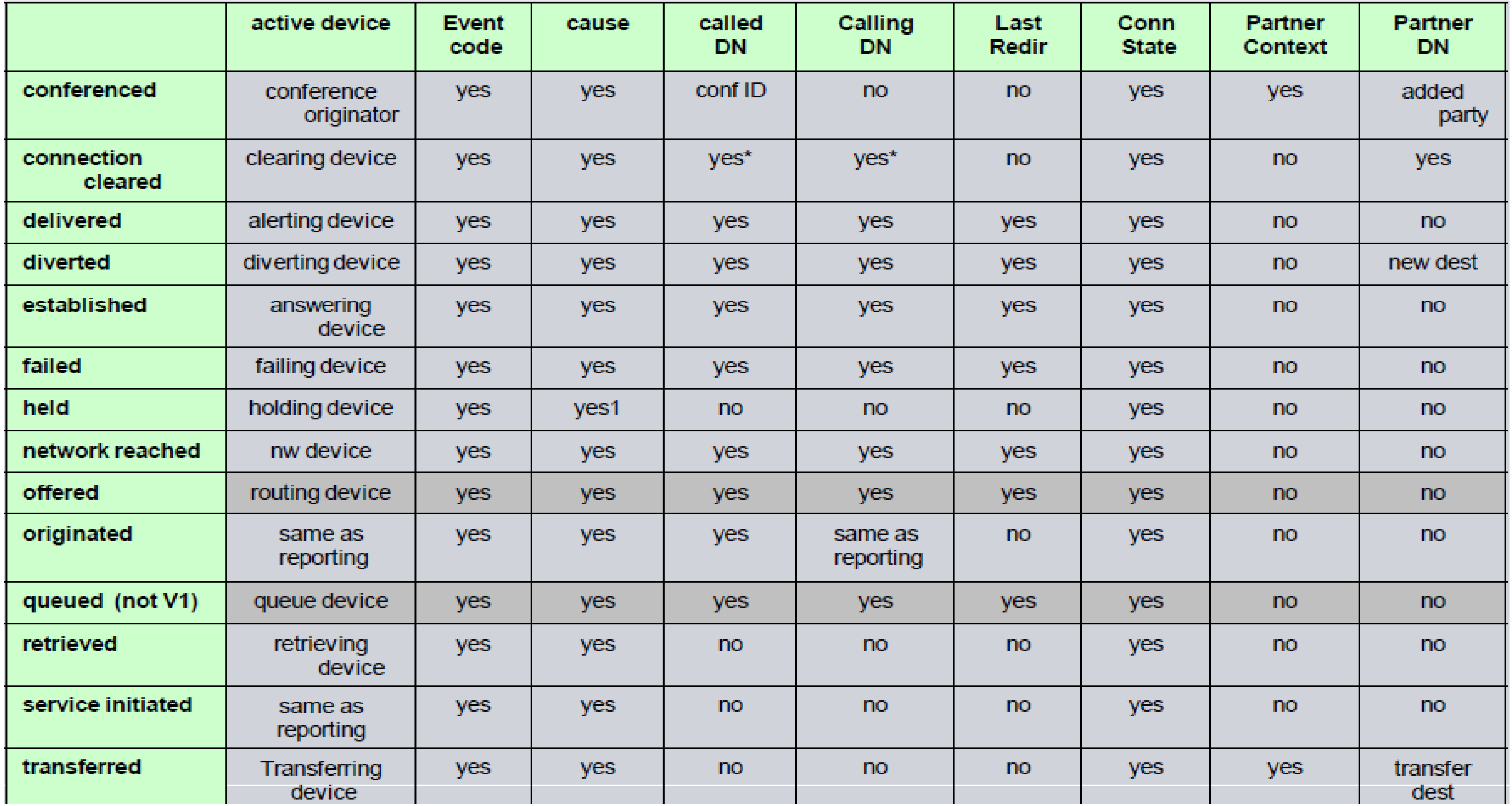
- Change Connection Information
- Generate Digits
- Conference, Add to Conference, Remove From Conference
- Clear Connection After Call Back
- Group Pickup Call,
- Directed Pickup Call
- One Number Service Inbound
- One Number Service Outbound

Features: Call Control Events

OpenScape Voice currently supports the following events (V3.1):

- **Events**: Bridged, Call Cleared, Conference, Connection Cleared, Delivered, Digits Dialed, Diverted, Established, Failed, Held, Network Reached, Offered, Originated, Queued, Retrieved, Service Initiated, Transferred.

CSTA Events/Attributes Matrix



	active device	Event code	cause	called DN	Calling DN	Last Redir	Conn State	Partner Context	Partner DN
conferenced	conference originator	yes	yes	conf ID	no	no	yes	yes	added party
connection cleared	clearing device	yes	yes	yes*	yes*	no	yes	no	yes
delivered	alerting device	yes	yes	yes	yes	yes	yes	no	no
diverted	diverting device	yes	yes	yes	yes	yes	yes	no	new dest
established	answering device	yes	yes	yes	yes	yes	yes	no	no
failed	failing device	yes	yes	yes	yes	yes	yes	no	no
held	holding device	yes	yes1	no	no	no	yes	no	no
network reached	nw device	yes	yes	yes	yes	yes	yes	no	no
offered	routing device	yes	yes	yes	yes	yes	yes	no	no
originated	same as reporting	yes	yes	yes	same as reporting	no	yes	no	no
queued (not V1)	queue device	yes	yes	yes	yes	yes	yes	no	no
retrieved	retrieving device	yes	yes	no	no	no	yes	no	no
service initiated	same as reporting	yes	yes	no	no	no	yes	no	no
transferred	Transferring device	yes	yes	no	no	no	yes	yes	transfer dest

Features
Capabilities Exchange Services

OpenScape Voice currently supports the following services (V3.1):

- **Get CSTA Features:** The Get CSTA Features service obtains the list of CSTA services and events supported by the switching function (OSV).
- **Get Logical Device Information :** The Get Logical Device Information service is used to obtain the current set of characteristics/capabilities associated with the logical element of a given device.
- **Get Switching Function Capabilities :** The Get Switching Function Capabilities service is used by the computing function to obtain the current set of capabilities for the entire switching function
- **Get Switching Function Devices :** The Get Switching Function Devices service is used by the computing function to obtain the current set of devices in the application working domain along with their associated device categories and associated device names.
- **Switching Function Devices :** The Switching Function Devices service is used by the switching function to provide a list of devices in the application working domain. This service is generated as a result of the Get Switching Function Devices service.

Features
Physical Device Services

OpenScape Voice currently supports the following services (V3.1):

- **Get Message Waiting Indicator :** The Get Message Waiting Indicator service provides the message waiting feature status at a specified device. The message waiting feature is typically used to notify a user (typically via a dedicated lamp on a phone device) when messages are available.
- **Get Microphone Mute* :** The Get Microphone Mute service provides the microphone mute feature status of a microphone associated with an auditory apparatus at a specified device.
- **Get Speaker Volume* :** The Get Speaker Volume service provides the speaker volume setting for the speakers associated with one or more auditory apparatuses at a specified device.
- **Set Message Waiting Indicator :** The Set Message Waiting Indicator service allows a computing function to control the status of the message waiting feature at a specified device. The message waiting feature is typically used to notify a user (typically via a dedicated lamp on a phone device) when messages are available.
- **Set Microphone Mute* :** The Set Microphone Mute service allows the computing function to control the microphone mute status of the microphone associated with one auditory apparatus at a specified device.
- **Set Speaker Volume* :** The Set Speaker Volume service allows the computing function to control the speaker volume of the speaker associated with one auditory apparatus at a specified device.
- * This service requires a Siemens UACSTA-capable physical device such as the OpenStage phone.

Features
Logical Device Services

OpenScape Voice currently supports the following services (V3.1):

- **Get Forwarding:** The Get Forwarding service provides the forwarding feature status at a specified device. The status returned may consist of one or more forwarding types that are active at the specified device based on user defined conditions.
- **Get Agent State:** The Get Agent State service provides the agent state at a specified device.
- **Get Do Not Disturb:** The Get Do Not Disturb service provides the do not disturb feature status at a specified device.
- **Set Forwarding:** The Set Forwarding service allows the computing function to control the forwarding feature at a specified device based on user defined conditions. The forwarding feature is used to redirect calls that arrive at a specified device to an alternate destination.
- **Set Agent State:** The Set Agent State service requests a new agent state at a specified device. In the case where an ACD agent is involved with an ACD call, the transition to the requested state may or may not occur until the current connection transitions to the null state.
- **Set Do Not Disturb:** The Set Do Not Disturb service allows the computing function to control the do not disturb feature at a specified device. The do not disturb feature is typically used to prevent a specified device from being alerted.

Features: Snapshot Services

OpenScape Voice currently supports the following services:

- **Get Forwarding:** The Get Forwarding service provides the forwarding feature status at a specified device. The status returned may consist of one or more forwarding types that are active at the specified device based on user defined conditions.
- **Snapshot Call:** The Snapshot Call service provides information about the devices participating in a specified call. The information provided includes device identifiers, their connections in the call, and local connection states of the devices in the call as well as call related information.
- **Snapshot Device:** The Snapshot Device service provides information about calls associated with a given device. The information provided identifies each call the device is participating in and the local connection state of the device in that call.
- **Snapshot DeviceData:** This service is generated as a result of the Snapshot Device service. It is used when the switching function is providing snapshot device response information in multiple messages (otherwise the switching function provides the snapshot device response in the Snapshot Device positive acknowledgement).
  - This includes information about calls associated with a given device. The information provided identifies each call the device is participating in and the local connection state of the device in that call.
  - The switching function may generate a sequence of Snapshot DeviceData services, individually referred to as segments, in response to a single Snapshot Device service request.

Switching Function Representation International Num Format, GNF and Displays

- Switching Function Representation (SFR Format)
  - **Generic Format:** N<DN!SA&CA/EXT%AID>NM (*in this order) e.g.*

    <deviceIdentifier>N<+15615051005>Boca-1005;displayNumber=1005</deviceIdentifier>
- International Number Format and GNF
  - The CSTA International Number Format is defined in ECMA 269 and applies to both the generic Dialable Digits (DD)

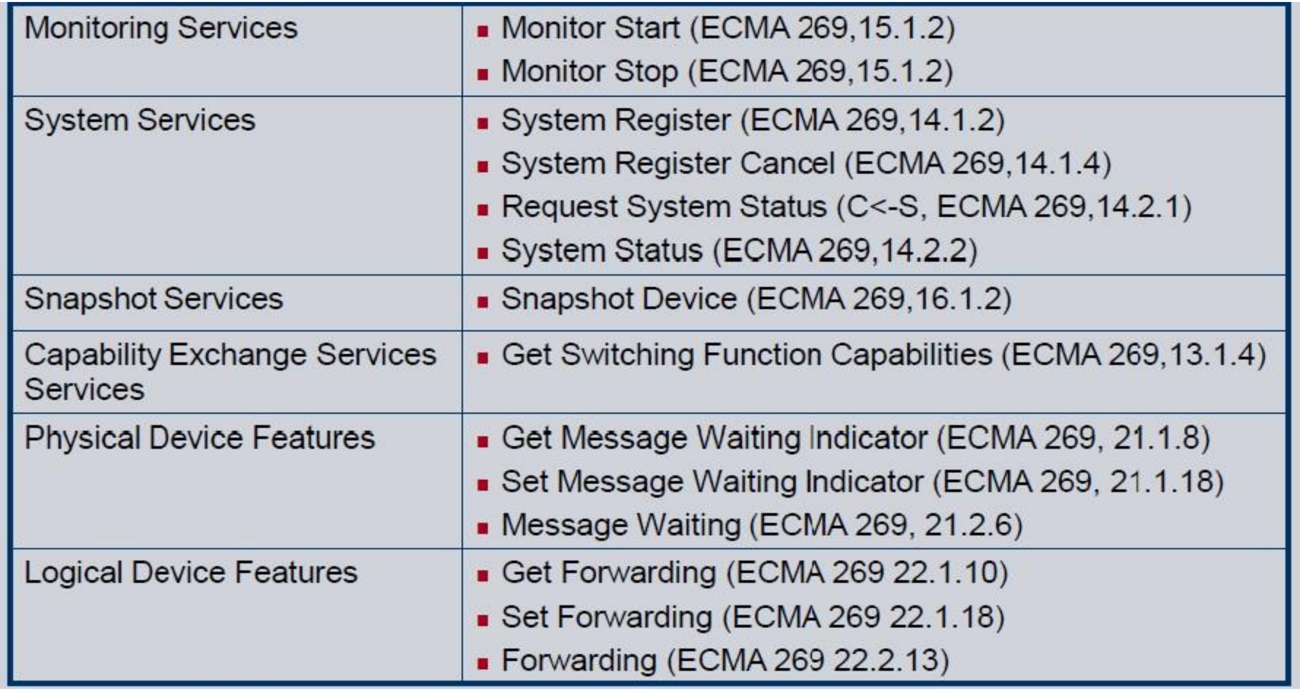
    (e.g. <deviceIdentifier>+15615051009</deviceIdentifier>)

    and Switching Function Representation (SFR) format
- CSTA International Number Format is also referred as Global Number Format (GNF).
- The plus sign indicates that this number is an international number format.
  - GNF is defined as: an international number prefixed with "+"

    e.g. +15619231388 ; +498972231668

CSTA Signaling Manager (SM) Services

Some CSTA Services are handled entirely within CSTASM:



Monitoring Services	▪ Monitor Start (ECMA 269,15.1.2) ▪ Monitor Stop (ECMA 269,15.1.2)
System Services	▪ System Register (ECMA 269,14.1.2) ▪ System Register Cancel (ECMA 269,14.1.4) ▪ Request System Status (C<-S, ECMA 269,14.2.1) ▪ System Status (ECMA 269,14.2.2)
Snapshot Services	▪ Snapshot Device (ECMA 269,16.1.2)
Capability Exchange Services Services	▪ Get Switching Function Capabilities (ECMA 269,13.1.4)
Physical Device Features	▪ Get Message Waiting Indicator (ECMA 269, 21.1.8) ▪ Set Message Waiting Indicator (ECMA 269, 21.1.18) ▪ Message Waiting (ECMA 269, 21.2.6)
Logical Device Features	▪ Get Forwarding (ECMA 269 22.1.10) ▪ Set Forwarding (ECMA 269 22.1.18) ▪ Forwarding (ECMA 269 22.2.13)

CSTA Signaling Manager (CSTA-SM) Services

Monitor Start /Monitor Stop

With a “Monitor Start” the CSTA applications requests that a device be monitored. If it has CSTA service subscribed, CSTASM sets a flag in SDAL, creates a monitor record in shared memory using a Monitor Cross Reference ID (XRefID), and sends a Monitor Start Response with the XrefID. The CSTA application must save the XRefID and send it inside a Monitor Stop request if it wants to stop receiving events for the device: CSTASM will search the XRefID for a “Monitor Stop” request in shared memory, delete the appropriate record, clear the SDAL flag, and send a response to the application.

System Register /System Register Cancel

A CSTA application registers with CSTASM in order to receive or request System Status (register services).

CSTA-SM Services

Request System Status /System Status

The System Status services and events are used to query or report the system status of the peer function (the computing function or switching function)

Snapshot Device

The Snapshot Device service is used to provide information about calls (connection identifier, local state) for a given device included in the request.

Get Switching Function Capabilities

The application queries the current set of capabilities for the entire switching function.

Get /Set Message Waiting Indicator / Message Waiting

The application can query, and set the message waiting feature (typically used via dedicated lamp when messages are waiting for the user)

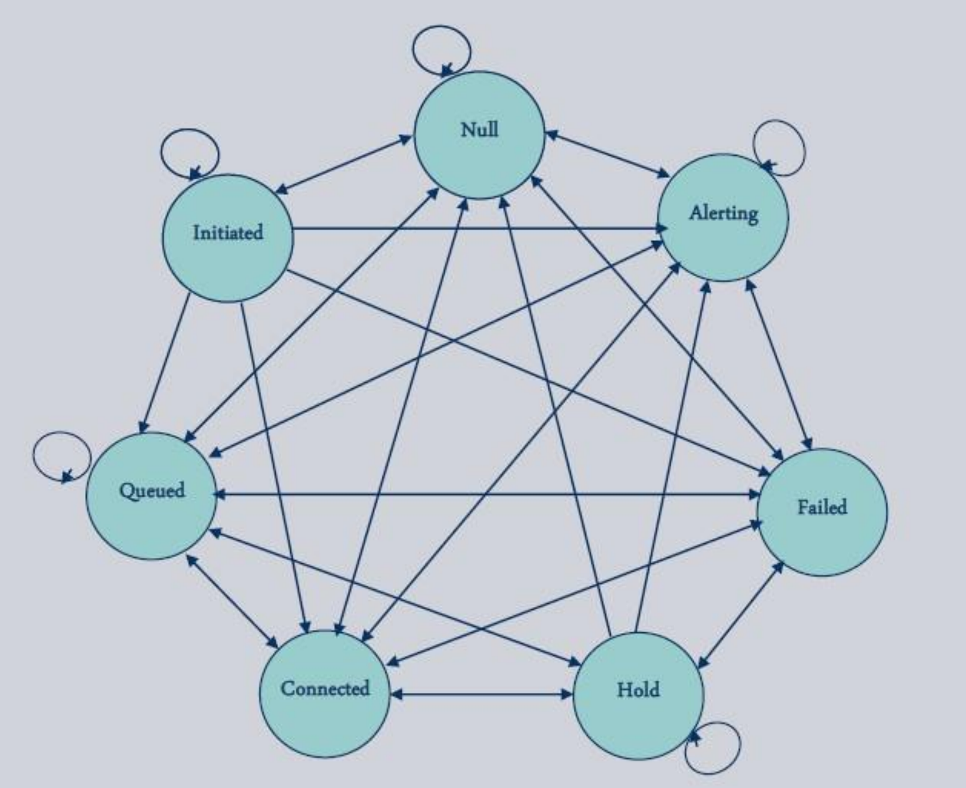
Get Forwarding / Set Forwarding / Forwarding

The application can query and set switch-based “immediate-always” forwarding for a device.

CSTA Service Basics

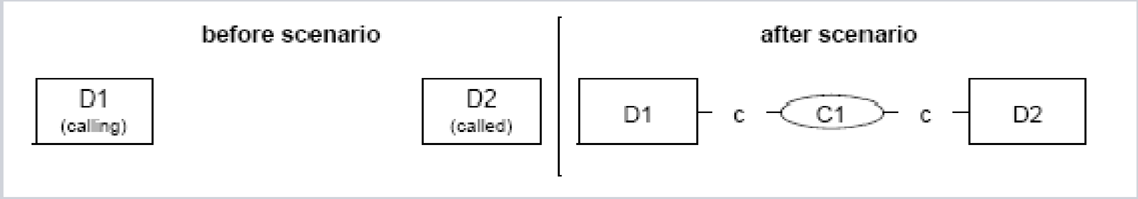
- UCE framework starts the CSTA service, if provisioned, for the subscriber once the UCE detects a call to that subscriber.
- An instance of the service is started for each leg of the call. The master service – CSTA, terminates after it spawns the child services; CSTA-A and CSTA-B (together called the CSTA Service)
- CSTA Service has two functions: monitor calls and invoke features/services on behalf of applications
- CSTA Service communicates the call status to CSTA-SM for every call (incoming or outgoing) for the subscriber provisioned with CSTA
- CSTA Service terminates at the end of each half call OR when explicitly told to terminate.

CSTA Connection Model



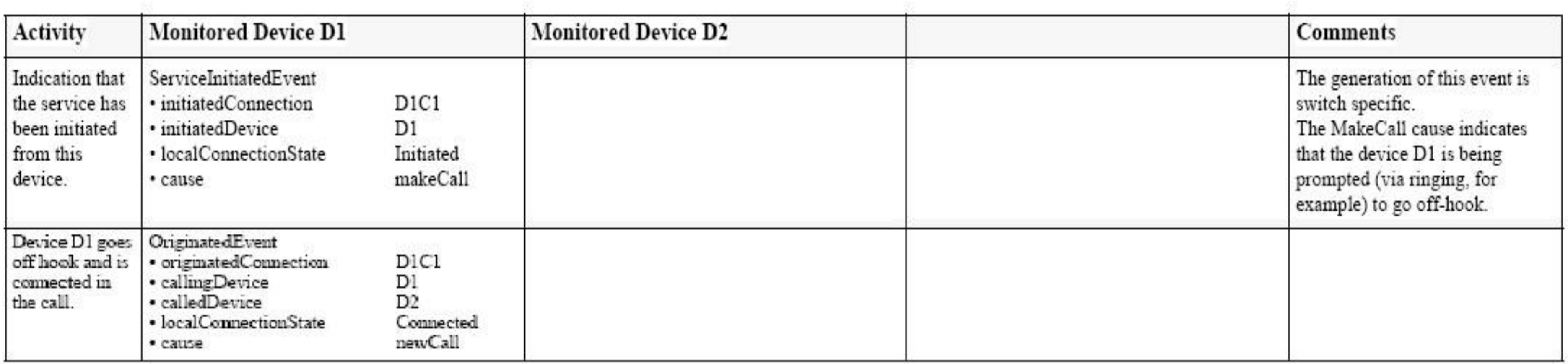
CSTA Feature Example - MakeCall

Call Origination Scenario MakeCall (Part 1 of 3)



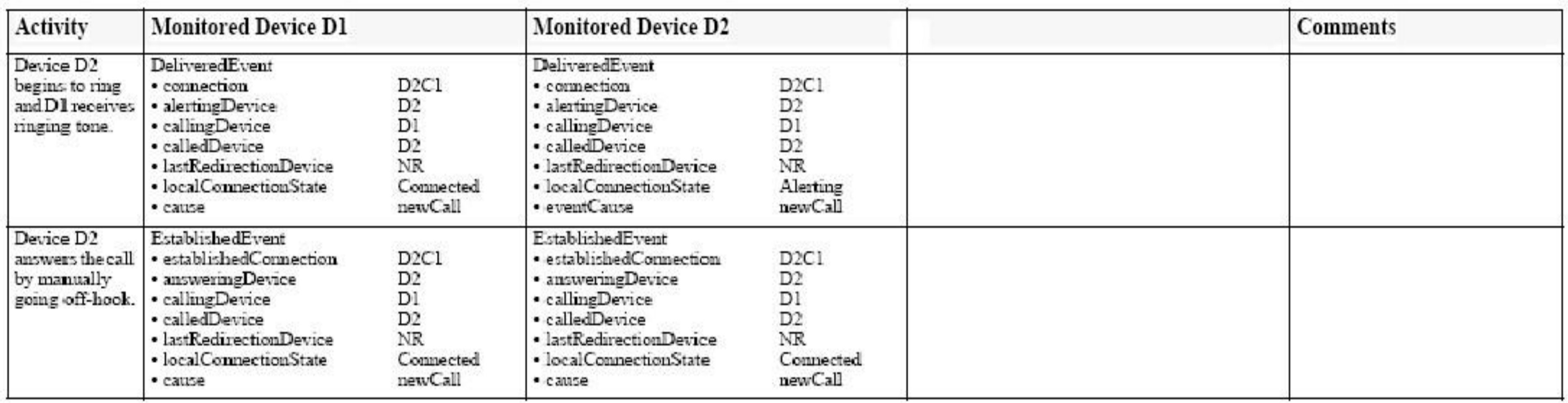
Activity	Monitored Device D1	Monitored Device D2		Comments
A Make Call service to a valid device is invoked on behalf of device D1.	MakeCallRequest • callingDevice D1 • calledDirectoryNumber D2 • autoOriginate Prompt			The Make Call service specifies that device D1 should be prompted to go off-hook.
Acknowl-edgement.	MakeCallResult • initiatedCall D1C1			

Call Origination Scenario MakeCall (Part 2 of 3)



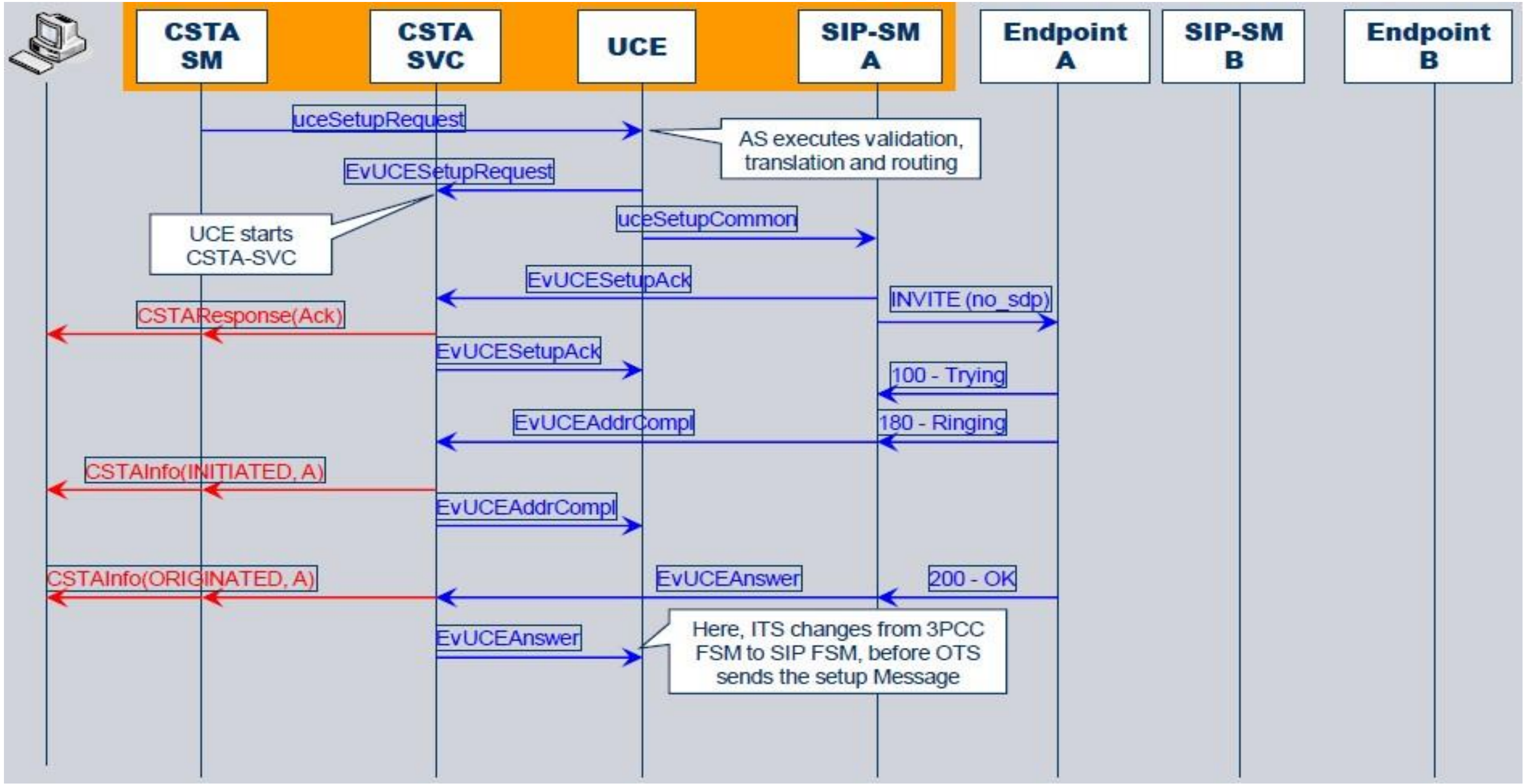
Activity	Monitored Device D1		Monitored Device D2		Comments
Indication that the service has been initiated from this device.	ServiceInitiatedEvent • initiatedConnection • initiatedDevice • localConnectionState • cause	 D1C1 D1 Initiated makeCall			The generation of this event is switch specific. The MakeCall cause indicates that the device D1 is being prompted (via ringing, for example) to go off-hook.
Device D1 goes off hook and is connected in the call.	OriginatedEvent • originatedConnection • callingDevice • calledDevice • localConnectionState • cause	 D1C1 D1 D2 Connected newCall			

Call Origination Scenario MakeCall (Part 3 of 3)

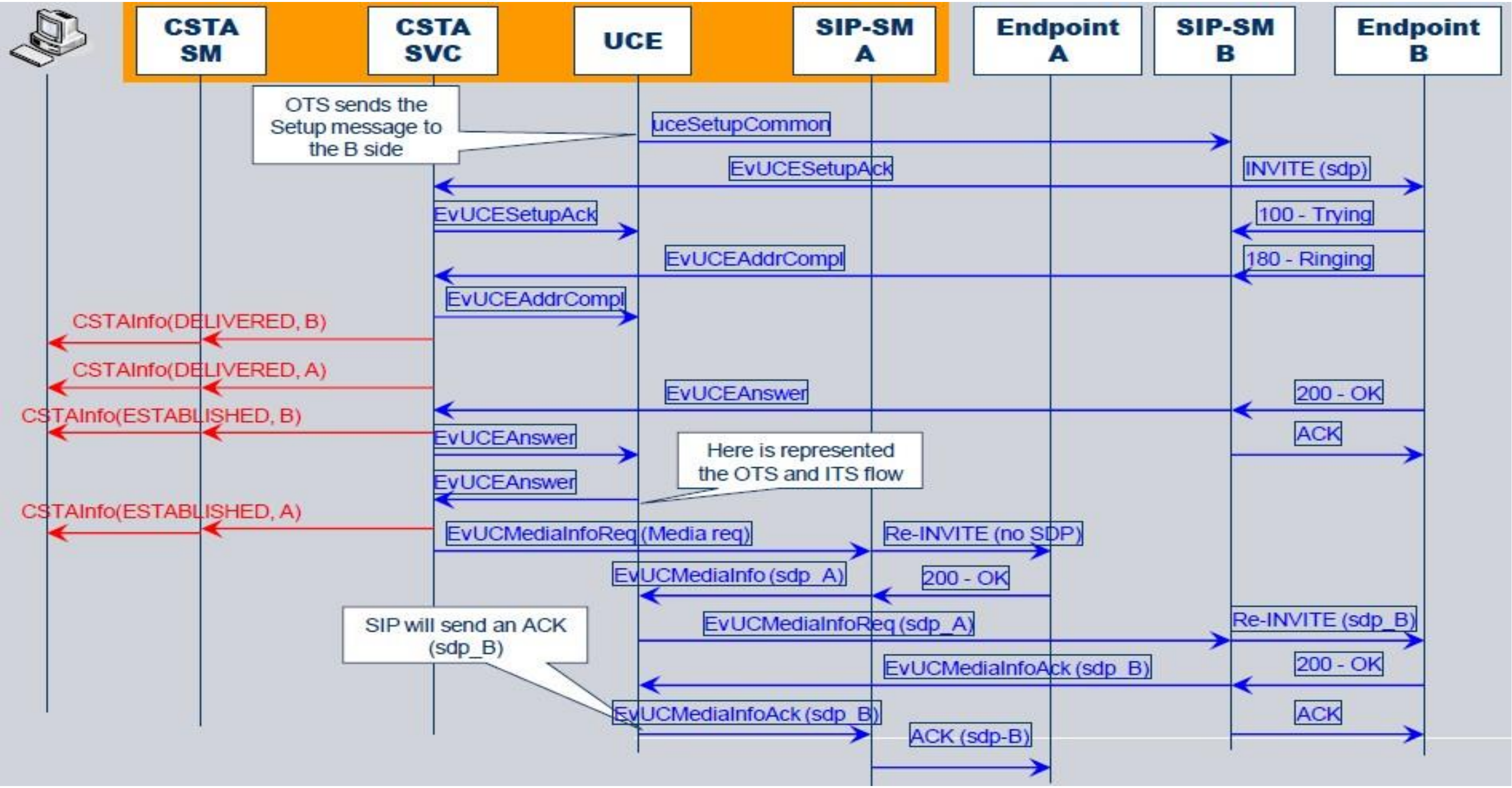


Activity	Monitored Device D1	Monitored Device D2		Comments
Device D2 begins to ring and D1 receives ringing tone.	DeliveredEvent • connection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause newCall	DeliveredEvent • connection D2C1 • alertingDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Alerting • eventCause newCall		
Device D2 answers the call by manually going off-hook.	EstablishedEvent • establishedConnection D2C1 • answeringDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause newCall	EstablishedEvent • establishedConnection D2C1 • answeringDevice D2 • callingDevice D1 • calledDevice D2 • lastRedirectionDevice NR • localConnectionState Connected • cause newCall		

CSTA Services and Events MakeCall Flow (Part 1 of 2)



CSTA Services and Events MakeCall Flow (Part 2 of 2)



OTS sends the Setup message to the B side

Here is represented the OTS and ITS flow

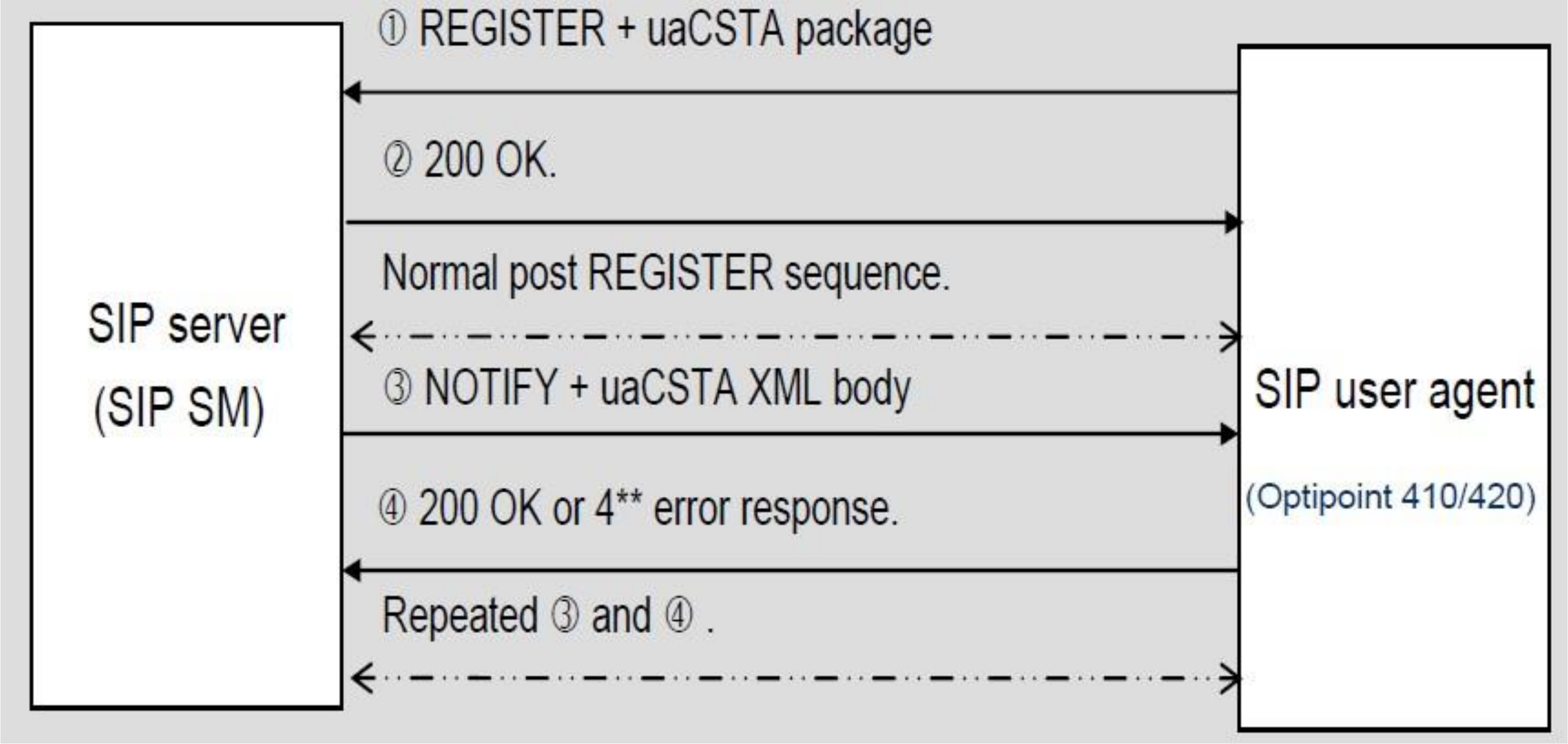
SIP will send an ACK (sdp_B)

User Agent CSTA (uaCSTA)

uaCSTA

- To provide better feature activation
  - 1st party feature activation instead of 3rd party feature activation by directly sending the request to the phone and phone initiating the request. Eg: MakeCall, ClearConnection, etc.
- Performance enhancement
  - By reducing the message interaction. 1st party feature activation results in fewer messages than 3rd party.
- Not a standard conformant (ECMA TR/87)
  - Standard requires phone session specifically established through INVITE and requests send as INFO messages. Currently we use unsolicited NOTIFY messages to send requests to SIP user agents.
  - Fewer SIP SM impacts and easy Phone implementation.

uaCSTA Flows – Device Registration


① REGISTER + uaCSTA package
② 200 OK.
Normal post REGISTER sequence.
③ NOTIFY + uaCSTA XML body
④ 200 OK or 4** error response.
Repeated ③ and ④ .
SIP server
(SIP SM)
SIP user agent
(Optipoint 410/420)

uaCSTA supported CSTA Services

Only Request to SIP-UA

- Call Control services
  - MakeCall
  - ClearConnection
  - HoldCall
  - RetrieveCall
  - GenerateDigits
- Physical Device services
  - GetMicrophoneMute
  - SetMicroPhoneMute
  - GetSpeakerVolume
  - SetSpeakerVolume

SIP user endpoint can not have multiple contacts (multiple registration bindings)

uaCSTA supported CSTA Services

Supported requests from SIP-UA (and responses back to SIP-UA)

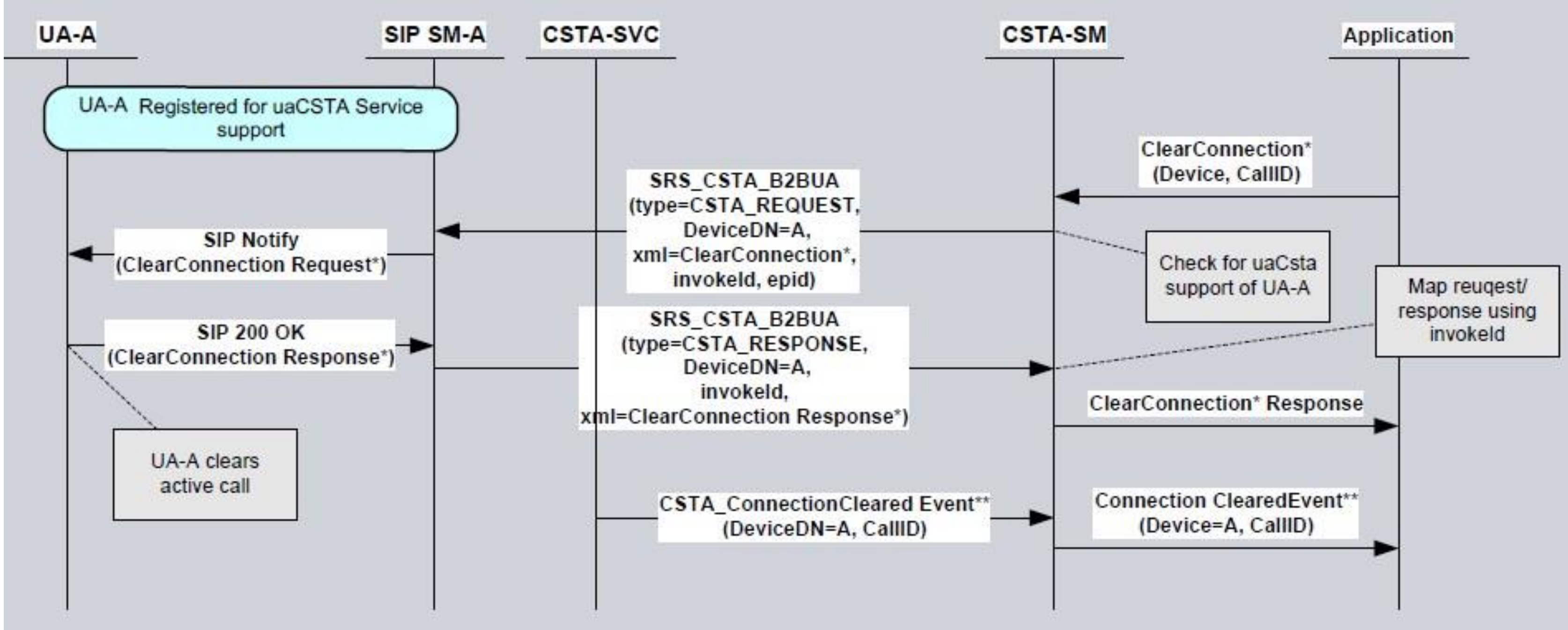
- GetForwarding
- SetForwarding
- GetDoNotDisturb
- SetDoNotDisturb

Supported events to SIP-UA

- DIVERTED EVENT (call forwarding display)

SIP user profile must be provisioning with option CSTA_TYPE_CSTA_OVER_SIP

uaCSTA call control flows ClearConnection


UA-A
SIP SM-A
CSTA-SVC
CSTA-SM
Application
UA-A Registered for uaCSTA Service support
ClearConnection* (Device, CallID)
SRS_CSTA_B2BUA (type=CSTA_REQUEST, DeviceDN=A, xml=ClearConnection*, invokeId, epid)
SIP Notify (ClearConnection Request*)
Check for uaCsta support of UA-A
Map reuqest/ response using invokeId
SIP 200 OK (ClearConnection Response*)
SRS_CSTA_B2BUA (type=CSTA_RESPONSE, DeviceDN=A, invokeId, xml=ClearConnection Response*)
ClearConnection* Response
UA-A clears active call
CSTA_ConnectionCleared Event** (DeviceDN=A, CallID)
Connection ClearedEvent** (Device=A, CallID)

* request would be one of AnswerCall, ClearConnection, HoldCall, RetrieveCall and GenerateDigits request

** events generated by CSTA-SVC would depend on the request invoked

CSTA – References

CSTA Standards: ECMA

http://www.ecma-international.org/activities/Communications/TG11/cstaIII.htm

Other CSTA related documentation at ESY-Wiki

http://usbrws01.bocb.siemens.com/mwiki/index.php/DevCallProcessingCSTA

CSTA Browser (ESY-Wiki)

http://usbrws01.bocb.siemens.com/mwiki/index.php/DevToolsCSTABrowserTool