Business Talk & BTIP Configuration Guidelines with Oracle Enterprise Customer eSBC

version addressed in this guide: Oracle E-SBC 9.3.0

Version of 16/01/2026

Information included in this document is dedicated to customer equipment (IPBX, TOIP ecosystems) connection to Business Talk & BTIP service: it shall not be used for other goals or in another context.

Table of contents

1 General

1.1 Scope of the document

The aim of this document is to provide configuration guidelines to ensure the interoperability between Oracle E-SBC with Business Talk (BTalk) or Business Talk IP (BTIP) service from Orange Business, hereafter so-called “service”.

1.2 References documents

Title	Link
Ex: Enterprise Session Border Controller S-Cz9.3.0	https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/9.3.0/index.html
[1] ACLI Reference Guide	https://docs.oracle.com/en/industries/communications/session-border-controller/9.3.0/aclireference/acli-reference-guide.pdf
[2] E-SBC Configuration Guide	https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/9.3.0/configuration/E-SBC-configuration-guide.pdf
[3] Security Guide	https://docs.oracle.com/en/industries/communications/session-border-controller/9.3.0/security/security-guide.pdf
[4] Release Notes	https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/9.3.0/releasenotes/esbc-release-notes.pdf
[5] Admin Security Guide	https://docs.oracle.com/en/industries/communications/session-border-controller/9.3.0/adminsecurity/admin-security-guide.pdf
[6] HMR Guide	https://docs.oracle.com/en/industries/communications/session-border-controller/9.3.0/hmr/hmr-guide.pdf
[7] Preparation & Instalation Guide	https://docs.oracle.com/en/industries/communications/session-border-controller/9.3.0/installation/platform-preparation-and-installation-guide.pdf
[8] Licensing Guide	https://docs.oracle.com/communications/scz930/esbc_scz930_license.pdf
[9] Solution Documentation – Application notes	http://www.oracle.com/technetwork/indexes/documentation/acme-packet-2228107.html

1.3 Prerequisites

1.3.1 Certificates

In case of encrypted SIP trunk architecture, TLS configuration is mandatory to exchange a certificate with Orange BT/BTIP A-SBC. Orange's TLS implementation operates in "Mutual Authentication" mode (also known as "two-way" authentication).

The customer must generate on the Oracle E-SBC a Certificate Signing Request (CSR) and submit it to a trusted public Certificate Authority (CA) to obtain a publicly signed certificate. After that, the Root CA and any intermediate CA certificates (all in PEM format) must be transmitted to Orange BT/BTIP team.

In turn, the Orange BT/BTIP team will provide you with our public Root and intermediate Certificate Authority (CA) certificates. These are the certificates that signed our Orange BT/BTIP A-SBC's certificate and must be imported onto your Oracle E-SBC to ensure proper trust and communication.

1.3.2 Public DNS configuration

For encrypted SIP trunk architecture, public DNS servers must be used for outgoing calls (e.g., from the customer's SIP endpoints to BTalk over InternetI/BTIP over Internet). To meet this requirement, you can configure, in E-SBC configuration, either the IP addresses of two private DNS servers that relay queries to the Internet, or the IPs of two accessible public DNS servers, such as Orange's public DNS (80.10.246.2 and 80.10.246.129).

1.3.3 NTP

The configuration of NTP on the Oracle E-SBC is not detailed in this document; however, it is recommended to implement an NTP server to ensure the E-SBC maintains accurate date and time, which is essential for validating remote party certificates. The configuration details are provided in the annexes.

1.3.4 Firewall flows for BTIP over Internet and BTalk over Internet

Firewalls in the way of traffic between Oracle eSBC and Orange infrastructure have to be updated in order to open required ports for BT over Internet or BTIP over Internet vary concerning the UDP Media ports range.

For BTIP over Internet, please note the Orange infrastructure Media public IP termination is different from Orange infrastructure SIP Signaling public FQDN/Public IP termination.

Refer to the 'BTalk over Internet & BTIP pre-requisites' and "BTalk/BTIP STAS" documents provided by your sales/project manager team for more details about firewall rules needed to be open.

1.4 Orange BT/BTIP specifications

The information in this chapter is the SIP trunk specifications required to interconnect Orange BTalk/BTIP network. The Enterprise SBC must be compliant with those specifications. Those information's were used to define the configuration described in this document.

✓ ***Supported RFC's:***

- RFC 2046: MIME part2: media types
- RFC 2396: Uniform Resource Identifiers (URI): Generic Syntax
- RFC 2976: The SIP INFO method
- RFC 3204: MIME media types for SUP and QSIG Objects
- RFC 3261: Session Initiation Protocol (SIP)
- RFC 3264: An offer/answer Model with the Session Description Protocol
- RFC 3311: The Session Initiation Protocol (SIP) UPDATE Method
- RFC 3323: A privacy Mechanism for the session Initiation Protocol (SIP)
- RFC 3325: Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks
- RFC 3326: The Reason header field
- RFC 3362: Real-Time Facsimile (T.38) image/t38 MIME
- RFC 3455: Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3GPP
- RFC 3725: Best Current Practices for Third Party Call Control (3pcc) in the Session Initiation Protocol (SIP)
- RFC 3960: Early Media and Ringing Tone generation in the Session Initiation Protocol
- RFC 3966: The tel URI for Telephone Numbers
- RFC 4566: SDP: Session Description Protocol
- RFC 4733: RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals
- RFC 5009: Private Header Extension to the Session Initiation Protocol for Authorization of early media
- RFC 5621: Message Body Handling in the Session Initiation Protocol (SIP)
- RFC 5806: Diversion Indication in SIP
- RFC 7434: Interworking ISDN Call Control User Information with SIP
- RFC 8119: SIP "cause" URI Parameter for Service Number Translation
- RFC 8147: Next-Generation Pan-European eCall

Note: RFC's not listed above are not supported in this context

✓ ***SIP methods supported:***

- INVITE
- ACK
- CANCEL
- UPDATE (only in confirmed dialog)
- BYE
- OPTIONS
- INFO

Note: SIP methods not listed are not supported in this context

✓ ***SIP message size specifications are as follows:***

- SIP message limited to 4096 Bytes on BT and 1500 Bytes on BTIP
- SDP Body limited to 1024 Bytes

✓ ***SIP signaling specifications are as follows:***

- For unencrypted architecture we need to configure UDP port 5060
- For encrypted architecture (TLS) we need to configuration TCP port 5061

✓ ***Media specifications are by default listed below and should be adapted to your customer service offer:***

- For unencrypted architecture we need to configure RTP port 6 000 to 20 000
- For encrypted architecture (TLS) we need to configuration SRTP port 6 000 to 20 000 for Business Talk over Internet or SRTP port 6 000 to 38 000 for Business talk IP over Internet.

✓ ***Customer equipment identification:***

- For Audit purpose E-SBC must include a “User Agent” header in INVITE messages and a "Server" header in all 18x messages sent to BT/BTIP infrastructure. The required format for these two headers is: "<IPBX/UC Vendor v.X.Y / SBC vendor v.X.Y>"

✓ ***SIP Signaling encryption specifications are as follows:***

- TLS version: 1.3 (Recommended)
  - Cipher suites:
    - TLS_AES_256_GCM_SHA384
    - TLS_AES_128_GCM_SHA256
    - TLS_CHACHA20_POLY1305_SHA256
- TLS version: 1.2 (only if TLS 1.3 is not supported)

- Cipher suites:
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

✓ ***Media encryption specifications are as follows:***

- SDES key exchange protocol (MIKEY not supported)
- Crypto suite: AES_CM_128_HMAC_SHA1_80
- Both RTP and RTCP are encrypted

✓ ***Codec/Packet rate specifications are as follows:***

- List of supported audio codecs and frame size:
  - G.722 20 ms
  - G.711 A law 20 ms, G.711 µ law 20 ms
  - G.729 20 ms, annexb = no

- BTalk (BVPN) – international
  - Either G.711A or µ, G.729 (most preferred codecs list)
  - Or G.711A or µ
  - Or G.729

- BTOI (Internet access) – international
  - Only G711A or µ is supported.

- BTIP (BVPN) – France
  - Either G.722, G.711A, G.729 (most preferred codecs list)
  - Or G.722, G.711A
  - Or G.711A, G.729
  - Or G.711A
  - Or G.729

- BTIPoI (Internet access) – France

- Only G711A is supported.

✓ ***Voice Activity Detection (VAD) is not supported***

✓ ***DTMF:***

- For Human to Machine, the "telephone-event" [RFC 4733] MUST be used for DTMF transport.
- Only events 0 through 15 are supported.
- Payload type value SHALL be configurable (recommended value is 101).

✓ ***SIP probing:***

- BT/BTIP SIP trunk relies on SIP OPTIONS method to “probe” the E-SBC, both within-dialog and out-of-dialog.
- The following answers are expected:
  - Out of dialog: 200 OK (or any error responses) if the UE is up, no response if the UE is down.
  - Within dialog: 200 OK if the call is active and 481 if the call is no more active.
- The Customer SBC may periodically send OPTIONS messages, each 300s, with Max-Forwards = 0 to probe the BT/BTIP SIP trunk. In this case, the BT/BTIP infrastructure will respond with a 483.
- Session Timer [RFC 4028] is not supported

✓ ***FAX support:***

T.38 parameters	Expected value	Parameters' value importance
T.38 Fax over UDP	UDPTL over UDP	Mandatory
Use of NSF/NSC requests	Optional	Optional
NSF value	0	Recommended
		NSF value matching to an existing NSF vendor value is forbidden
		Expected NSF value is 000000 or FFFFFF
Use of NTE ([RFC 4733]) or NSE (Cisco)	No	Mandatory
Fax rate management method	Transferred TCF	Mandatory
UDP redundancy method	T38UDPredundancy	Mandatory
Coding method (fillbitRemoval, JBIG, MMR)	No (MH only)	Mandatory
T.38 version parameter	0	Mandatory
T.30 data	V.21	Mandatory
Data signaling rate	V.17, V.29, V.27ter	At least one of those modulations is mandatory
		Those three modulations are highly recommended
		Any other modulation (like V.34) is forbidden
Error Correction Method	Enabled	Highly recommended
V.8 parameter	Disabled	Mandatory
Polling mode	Disabled	Mandatory
Fax rate	14400 bps	Recommended
		Any fax rate greater than 14,4kbps is forbidden
Low speed T.38 redundancy	4	LS redundancy is mandatory
		Level 4 is recommended
High speed T.38 redundancy	1	HS redundancy is mandatory
		Level higher than 1 is forbidden
SG3-G3 fallback method	Either ANSam removal	Mandatory
	or CM removal	
T.38 payload size	40 ms	Highly recommended
		Any payload size different from 40 and 20 ms is forbidden
Switching from voice mode to fax mode	T.38 Re-INVITE sent as callee AND as caller (BTalk and BTIP)	Mandatory

Note: For T.38 the Oracle E-SBC will be transparent. No adaptation will be done at the SBC level; DSP resources would be required in certain conditions.

✓ ***Packet marking:***

- Both SIP signaling and audio must be marked with DSCP 46 (Expedited Forwarding).

✓ ***Call initiation:***

- E-SBC shall provide an SDP within his initial INVITE, delay offer (INVITE without SDP) is not supported.

✓ ***Media session modification:***

- Modification of media (IP, codec, attributes ...) in reception/transmission based on UPDATE (With SDP) in Early Dialog and Re-INVITE in confirmed Dialog (with or without SDP)
- Attributes “a=” must be equal to “send only, recvonly, inactive, sendrecv”.
- Same Methods/Attributes/headers may be sent from BTalk/BTIP to Customer SBC.
- Call Transfer
  - For supervised call transfer, sends RE-INVITE in confirmed dialog
  - For blind call transfer:
    - send RE-INVITE in confirmed dialog
- Call Forward
  - In case of Call Forward, the diversion header must be provided by the Customer SBC to maintain the original caller information.
- Call on Hold
  - Send SDP with a=inactive; Setting connection to 0.0.0.0 FORBIDDEN.
- Music on Hold
  - Initiate a new INVITE to the media server
  - Use Re-INVITE to stop, closing the second dialog.

✓ ***3-Way Conference***

- Use a media mixer with existing dialogs
- "Join" header [RFC 3911] NOT SUPPORTED by Orange.

✓ ***Ring back tone and early media:***

- ***Incoming calls***
  - Use the "P-Early-Media" header in 18x responses to signal early media transmission. Nevertheless, the service does not guarantee to relay this early media (depending on specific agreement)
  - If SIP endpoint sends a 18x response with SDP without "P-Early-Media", it SHOULD send a ring back tone. However, this tone may not be heard by the remote party.
- ***Outgoing calls***
  - Presence of "P-Early-Media" header with "sendrecv" or "sendonly" values in 18x responses indicates that early media will be sent. SIP endpoint SHALL inhibit local tones generation and wait for incoming audio.
  - If "P-Early-Media" header with "inactive" or "recvonly" values is set in 18x responses, SIP endpoint SHALL generate local tones.
  - If "P-Early-Media" header is not set in 18x responses, SIP endpoint SHOULD generate local tones unless it can detect early media sent by the remote party.

✓ ***Anonymous calls:***

- If anonymization is requested, the Customer SBC should:
  - Set the Privacy header to at least "user" and ensure the From header contains the calling party's identity.

    Or
  - Set the Privacy header to at least "id". Ensure the From header contains an anonymous URI (such as "Anonymous" sip:anonymous@anonymous.invalid), and the P-Asserted-Identity header contains the calling party's identity.

✓ ***Number format specifications are:***

- Called party identities must be sent to the Orange network in E.164 format (i.e. +CCNSN).
- Calling party identities must be sent to the Orange network in E.164 format (i.e. +CCNSN).

✓ ***Rerouting scenario:***

- On reception of an error response, the customer SIP endpoint must try a second route towards the backup BT/BTIP A-SBC if response code is either 408 or 5xx.
- When a customer has multiple components (e.g., active/backup servers), upon receiving an error response from a SIP endpoint, the BT/BTIP core network will reroute the call to a backup SIP endpoint if the response code is 408 or 5xx.

✓ ***Call defection:***

- Sends only RE-INVITE in confirmed dialog.
- INVITE sent to the deflection destination SHOULD include Diversion header with Deflection reason.
- 3xx Sip messages are not supported by BTalk/BTIP services. Those messages will be converted into SIP error messages.
- "Retry-After" ignored by Orange.

✓ ***RTCP***

- Customer SBC will receive reports every 5 seconds from Orange backbone, and it is recommended that SIP endpoint generates RTCP reports

✓ ***STIR SHAKEN***

- If caller identity authentication is requested, SIP endpoint MUST accept to receive the following information:
  - Identity (up to 650 bytes), P-Attestation-Indicator, P-Origination-ID headers
  - Verstat parameter in user-part of From, P-Asserted-Identity and Diversion headers.

2 Certified Architecture

2.1 Introduction to architecture components and features

This document provides configuration guidelines for the Oracle E-SBC north (carrier) interface used by the Orange Business (OB) within the VISIT Program.

It outlines the configuration requirements necessary to ensure interoperability between the Oracle E-SBC and the Business Talk (BTalk) and Business Talk IP (BTIP) SIP infrastructure, including the A-SBC, Application Server, and interconnections with the PSTN or SIP carriers.

These guidelines apply specifically to the north (carrier) side of the Oracle E-SBC, which interfaces with BTalk and BTIP services:

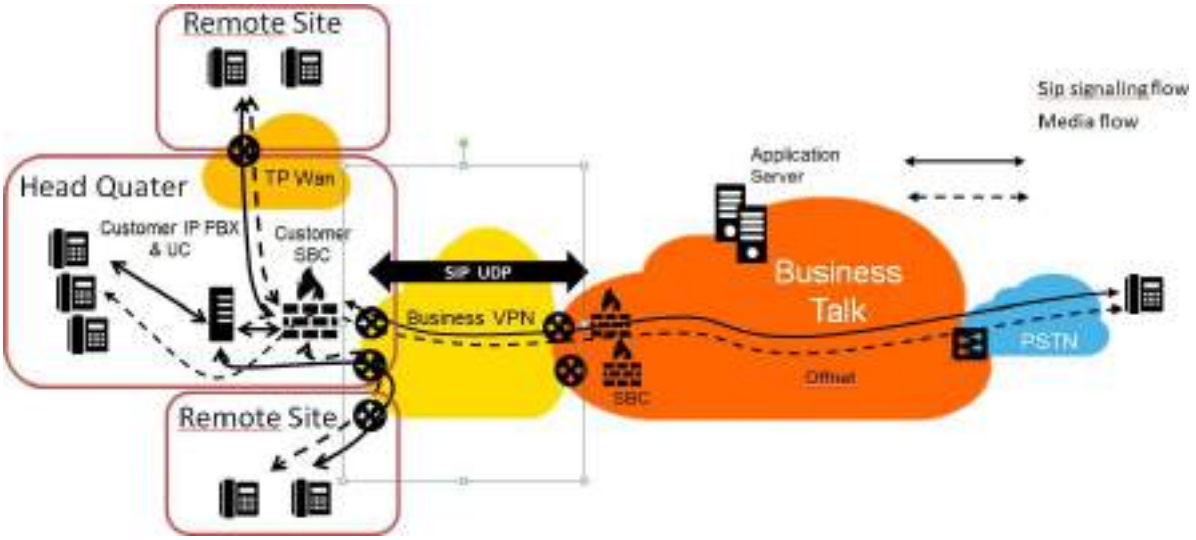
- The configuration will only consider the Carrier aspect of the Oracle E-SBC (north side), which faces BTalk/BTIP offers.
- The E-SBC's North-side SIP termination will act as the demarcation point for Orange Business.
- The south side of the Oracle E-SBC falls outside of OB's control and responsibility.

The primary objective of these guidelines is to ensure that the Oracle E-SBC configuration complies with the requirements (SIP/T.38 profile) of BTalk and BTIP offers. Any complexities introduced by diverse UC/IPBX environments must be managed on the south side and fall outside of OB’s responsibility.

Note: Fax communications via Business Talk are currently allowed but not officially supported

2.2 Architecture with Oracle “customer” E-SBC

2.2.1 Unencrypted SIP Trunk through BVPN

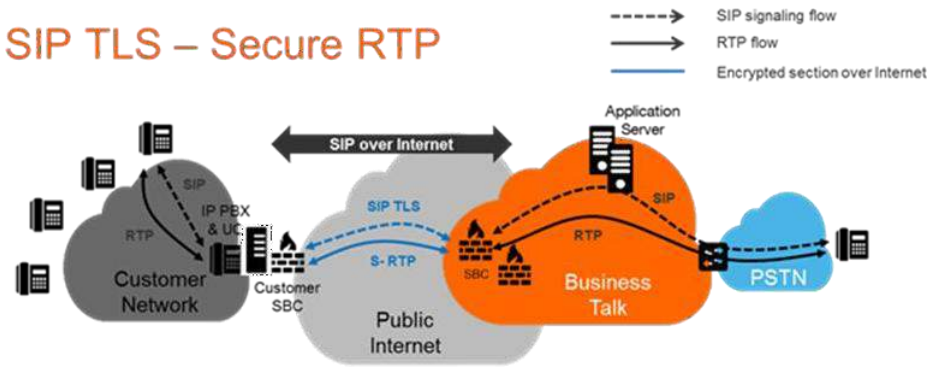


In this architecture:

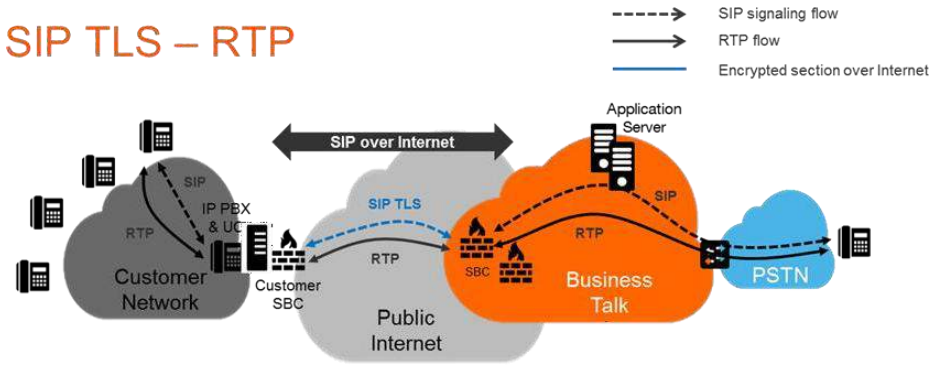
- Both SIP SIG and RTP media flows between endpoints and the BTalk/BTIP are anchored by the customer SBC.
- For Headquarter, SIP SIG flows are routed through the Customer SBC and RTP media flows are direct to private “South” interface of the E-SBC through the main BVPN connection.
- For remote sites interconnected through BVPN, SIG flows are routed through the Customer SBC and RTP media flows are direct to private “South” interface of the E-SBC and the main BVPN connection.
- For remote sites interconnected through 3rd Party Wan, both SIP SIG & RTP media flows are routed through the Customer SBC direct to private “South” interface of the E-SBC through the main BVPN connection.

2.2.2 Encrypted SIP Trunk Over Internet

- SIP over TLS + Secured RTP: all SIP messages and media packets are encrypted on the public internet between Orange and the customer's SIP endpoints. This is the level of encryption recommended by default by Orange to ensure security and privacy.



- SIP over TLS + (unencrypted) RTP: all SIP messages are encrypted on the public internet between Orange and the customer's SIP endpoints. RTP flows are shared without encryption between the customer media endpoints and Orange's backbone. This solution is not recommended by Orange but is allowed as an alternative when customers face encryption/decryption limitations.



2.2.4 Parameters to be provided by customers to access the service

2.2.4.1 Unencrypted SIP Trunk through BVPN

Depending on the architecture scenario selected by the Customer, several IPv4 addresses must be provided. The table below summarizes the required IP addresses (highlighted in red) for each scenario.

Applicable to all Session Border Controller with BTalk or BTIP over BVPN

Customer SBC – architecture with E-SBC	Level of Service	@IP used by service	
1 Single Customer SBC	No redundancy	E-SBC @IP	
2 Customer SBC Nominal / Backup mode	- **Local redundancy**: both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 @IP	E-SBC2 @IP
2 Customer SBC in Load Sharing	- **Local redundancy**: both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 @IP E-SBC2 @IP	
2 Customer SBC in HA mode (Cluster)	- **Local redundancy:** both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites **warning:** Link level 2 between SBC with max delay 50ms required for geo-redundancy	E-SBC VIP @IP	

2.2.4.2 Encrypted SIP Trunk through Internet

Applicable to Customer SBC with BTalk over Internet only (International)

Customer SBC – architecture with E-SBC	Level of Service	@IP used by service	
1 Single Customer SBC	No redundancy	E-SBC1 @IP or Public FQDN	
2 Customer SBC Nominal / Backup mode	- **Local redundancy**: both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 @IP or Public FQDN	E-SBC2 @IP or Public FQDN
2 Customer SBC in Load Sharing	- **Local redundancy**: both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 @IP or Public FQDN E-SBC2 @IP or Public FQDN	

2 Customer SBC in HA mode (Cluster)	- **Local redundancy:** both SBC are hosted on the same site *OR* - **Geographical redundancy** both SBC are hosted on 2 different sites **warning:** Link level 2 between SBC with max delay 50ms required for geo-redundancy	E-SBC VIP @IP

Applicable to Customer SBC with BTIP over internet only (France)

Customer SBC – architecture with E-SBC	**Level of Service**	**@IP used by service**	
1 Single Customer SBC	No redundancy	E-SBC1 FQDN Type A	
2 Customer SBC Nominal / Backup mode (**DNS Resiliency model**)	- **Local redundancy**: both SBC are hosted on the same site OR - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC public FQDN DNS Type **SRV**	
2 Customer SBC Nominal / Backup mode (**SIP Resiliency model**)	- **Local redundancy**: both SBC are hosted on the same site OR - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 FQDN Type A *	E-SBC2 FQDN Type A*
2 Customer SBC in Load Sharing (**SIP Resiliency model**)	- **Local redundancy**: both SBC are hosted on the same site OR - **Geographical redundancy** both SBC are hosted on 2 different sites	E-SBC1 FQDN Type A* E-SBC2 FQDN Type A*	
2 Customer SBC in HA mode (Cluster) **(IP Resiliency model)**	- **Local redundancy:** both SBC are hosted on the same site OR - **Geographical redundancy** both SBC are hosted on 2 different sites **warning:** Link level 2 between SBC with max delay 50ms required for geo-redundancy	E-SBC VIP FQDN type A*	

* Only E-SBC public FQDN's SIP termination will be supported, E-SBC public IP's termination will not.

2.2.4.3 Information and syntax

The naming of the different objects created (Network interface, Rules names, …) must be respected in order to guaranty the coherence of the configuration and easy to check by Orange in case of issue.

Few parameters highlighted in "Green" color (IP Address, FQDN, capacity, …) in this document are given as example and must be replaced by the real values specifically for each interconnection.

Additional objects names highlighted in "Yellow" color (in-manipulationid …) must not be replaced by other value as they are call in other objects afterward.

Several tables in the following Chapters, will contain lines in "Grey" color. Those lines are indicated as example and reminder of the existing configuration of the "south" side (IPPBX side) inside the eSBC. If the eSBC used is a new one without existing configuration, you must replace those "Grey" lines according to the specifications of your IPBX/UC environment you want to interconnect to BTalk/BTIP network through the eSBC.

Examples

Description	Host/domain	Server Lookup	Port number	Protocol
Orange_BTalk	<BT-NOMINAL> <BT-BACKUP>	<IP>	<5060>	<UDP >
Orange_BTalk_TLS	<BT_Public IP_Nominal> <BT-Public_IP_Backup Or BT_Public FQDN_Nominal> <BT- Public_FQDN_Backup>	<Public_IP>	<5061>	<TCP>
Orange_BTIP	<BTIP-NOMINAL> <BTIP-BACKUP>	<IP>	<5060>	<UDP >
Orange_BTIP_TLS	BTIP_Public FQDN_Nominal> <BTIP- Public_FQDN_Backup> Or <BTIP- Public_SRV_record>	<Public_IP>	<5061>	<TLS 5061>
Oracle SBC	<C-SBC-FQDN>	<Public_IP>	<5061>	<TLS 5061>

The <DefaultSiteDID> call later is the one provided by Orange and corresponding to the BT/ BTIP default logical site number provisioned in E.164 format.

2.3 BTalk & BTIP Oracle Enterprise SBC certified versions

Oracle E-SBC – software versions			
Hardware or Virtual	Product reference	Software version	Certification
Hardware	1100	9.3p7 and later patchs	✓
	3900		
	3950		
	4600		
	4900		
	6350		
	6400		
Software	Private Virtual Infrastructures		

Notes :

- More information's on HW & Software supported following this link : https://docs.oracle.com/en/industries/communications/enterprise-session-border-controller/9.3.0/releasenotes/rn-platform-support-930.html
- Oracle recommends its customers to use most up-to-date software release. This ensures customers are taking advantages of the latest fixes (security fix included).
- At the moment of writing, 9.3 releases were used. This means that newer releases would also contain the features needed to support OB's services.
- In case of doubts or if you seek guidance, please contact your partner or Oracle directly.

2.4 Oracle Global configuration

Oracle offers different shapes of SBC, several appliances of NFV formats are available. But Oracle strives to provide a similar ACLI or ways to edit SBC's configurations. This means that if either you are using an appliance SBC or a virtual SBC, ultimately you should be able to use the configuration elements depicted in this chapter.

In these guidelines, it is assumed that customer or his partner is configuring one unique offer from Orange Business. In case, several offers need to be collocated on the same setup, modifications would be required and are not covered in this document.

2.4.1 Entitlements

Before starting to configure SBC for BTalk or BTIP trunking services, it is assumed that the management interface of the SBC is accessible, product is defined as an "Enterprise Session Border Controller" and entitlements are defined.

Configuration

```
CSBC# conf t
CSBC# setup product

----------------------------------------------------------------------------------------------------------
WARNING:
Alteration of product alone or in conjunction with entitlement
changes will not be complete until system reboot

Last Modified 2025-03-26 11:00:14
----------------------------------------------------------------------------------------------------------
 1 : Product       : Enterprise Session Border Controller

Enter 1 to modify, d' to display, 's' to save, 'q' to exit. [s]:
save SUCCESS
CSBC# setup entitlements

----------------------------------------------------------------------------------------------------------
Entitlements for Enterprise Session Border Controller
Last Modified: 2025-06-16 14:54:02
----------------------------------------------------------------------------------------------------------
 1 : Session Capacity                          : 100
 2 :   Advanced                                : enabled
 3 :   STIR/SHAKEN Client                      :
 4 : Admin Security                            :
 5 : Data Integrity (FIPS 140-3)               :
 6 : MSRP B2BUA Sessions                       : 0
 7 : Transcode Codec AMR                       :
 8 : Transcode Codec AMR Capacity              : 0
 9 : Transcode Codec AMRWB                     :
 10: Transcode Codec AMRWB Capacity            : 0
 11: Transcode Codec EVS                       :
 12: Transcode Codec EVS Capacity              : 0
 13: Transcode Codec OPUS                      :
 14: Transcode Codec OPUS Capacity             : 0
 15: Transcode Codec SILK                      :
 16: Transcode Codec SILK Capacity             : 0
```

Entitlements define the number of base licenses (ie. concurrent sessions) acquired with optional additional features. Customer must only enable items acquired. For example, concurrent sessions are defined by editing item "1: Session Capacity".

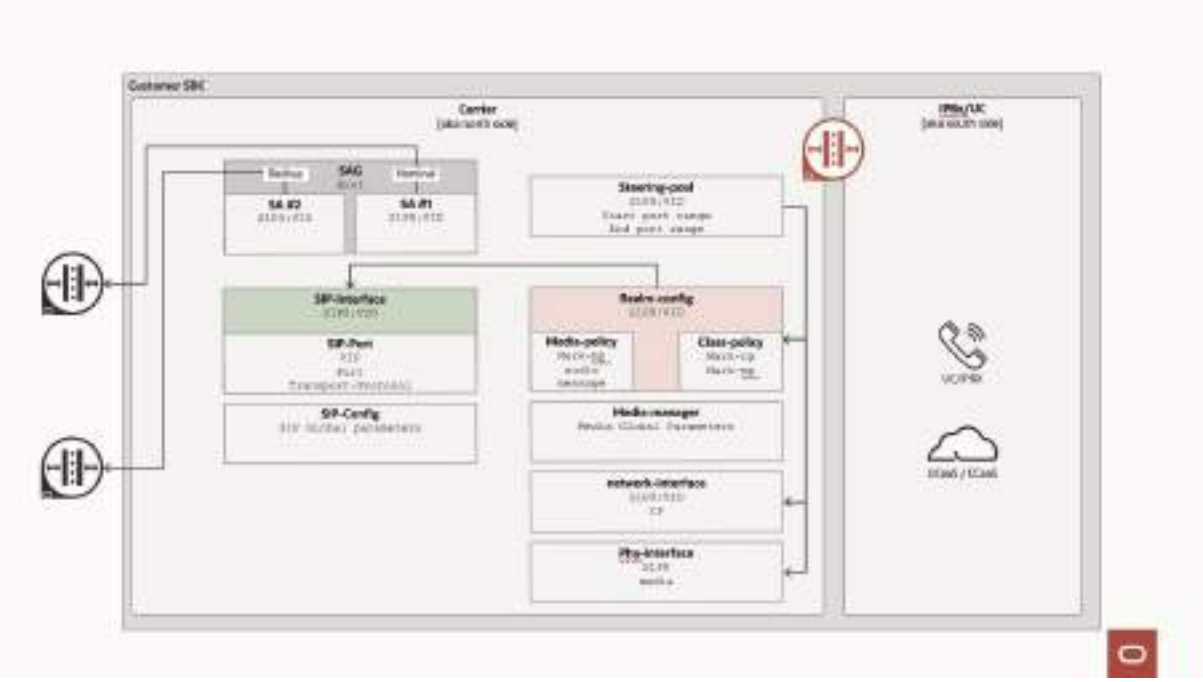
In case TLS is used as a transport protocol for SIP, an activation key is required for virtual SBCs only (ie. not needed for HW appliances). Same applies for SRTP. For more details or assistance please reach out to Oracle's partner or Oracle directly.

For more information, please refer to Oracle 9.3 licensing guide and to Oracle 9.3 Configuration Guide.

2.4.2 Physical interface configuration

At least, two phy-interfaces need to be configured, one for north (carrier) and one for south (UC/IPBx). It could work with only one phy-interface, but it is not the recommended option.

Note: In this guide, for example purpose, S0P0 interface will be used as the south side interface and S1P0 as the north side. Your setup might differ, so please be careful and update your configuration accordingly. On virtual-appliance, when activating a new interface on the hypervisor and on the virtual SBC, interface-mapping might need to be used to check or to edit the interface mapping. For more information, please refer to this chapter from Oracle Preparation and Installation Guide.



Configuration

```
CSBC# conf t
CSBC(configure)# system phy-interface
CSBC(phy-interface)# name S0P0
CSBC(phy-interface)# slot 0
CSBC(phy-interface)# port 0
CSBC(phy-interface)# operation-type media
CSBC(phy-interface)# done
<Output omitted // SBC displaying configuration of phy-interface S0P0!>
CSBC(phy-interface)# name S1P0
CSBC(phy-interface)# slot 1
CSBC(phy-interface)# port 0
```

```
CSBC(phy-interface)# operation-type media
CSBC(phy-interface)# done
<Output omitted // SBC displaying configuration of phy-interface S1P0!>
CSBC(phy-interface)# quit
CSBC# save
CSBC# activate
```

2.4.2.1 HA Cluster & Base MAC

If you are running a cluster of SBC, an additional step is needed to properly set the virtual mac (requirement for HA).

For more information, please refer to dedicated chapter in Oracle E-SBC Configuration Guide.

The below paragraph applies only for Hardware Appliance.

To identify your virtual-macs, you need to find your base-mac.

Configuration

```
CSBC# show prom-info mainboard
Mainboard Info
--------------

Contents of Main Board IDPROM
        Assy, NetNet6350
        <Output omitted>
        Starting MAC Address:           00 08 25 a2 45 b0

Note: show interfaces could work, but it means phy-interface are already defined.
```

Identify “Starting MAC Address” from each SBC in your cluster which reads: `00:08:25:XX:YY:ZN`.

`00:08:25` refers to Acme Packet, `XX:YY:Z` refers to the specific SBC. `N` is a 0-f hexadecimal value available for Oracle SBC.

To create a virtual MAC address, replace the “N” value with unused hexadecimal values for Oracle SBC: e or f.

Example:

- `Primary SBC = 00:08:25:a2:45:b0`
- `Secondary SBC = 00:08:25:a2:45:c0`
- `Virtual MAC for slot 0: 00:08:25:a2:45:be or 00:08:25:a2:45:bf`
- `Virtual MAC for slot 1: 00:08:25:a2:45:ce or 00:08:25:a2:45:cf`

The below paragraph applies only for VNF SBC.

To support HA, configure virtual MAC addresses based on the Burned In Addresses (BIA) of the media interfaces. To determine what the virtual MAC addresses should be, you first document a BIA and then calculate what the virtual MACs should be. Use the following steps to define the virtual addresses you need to configure for each interface:

1. Document the BASE-MAC of `eth0/wancom0` physical interface
2. Set the bottom nibble of the first byte to 2. Note that this defines the address as locally administered
3. Set the top nibble of the first byte to 0 and increment for each interface

For example, assuming the base-MAC for eth0 is `00:50:56:C0:00:08`, you would assign the virtual addresses as follows:

- 1st media interface virtual MAC = 02:50:56:C0:00:08
- 2nd media interface virtual MAC = 12:50:56:C0:00:08
- 3rd media interface virtual MAC = 22:50:56:C0:00:08
- 4th media interface virtual MAC = 32:50:56:C0:00:08

Configuration

```
CSBC# conf t
CSBC(configure)# system phy-interface
CSBC(phy-interface)# name S0P0
CSBC(phy-interface)# slot 0
CSBC(phy-interface)# port 0
CSBC(phy-interface)# operation-type media
CSBC(phy-interface)# virtual-mac <00:08:25:MM:AA:CE>
CSBC(phy-interface)# done
<Output omitted // SBC displaying configuration of phy-interface S0P0!>
CSBC(phy-interface)# name S1P0
CSBC(phy-interface)# slot 1
CSBC(phy-interface)# port 0
CSBC(phy-interface)# operation-type media
CSBC(phy-interface)# virtual-mac <00:08:25:MM:AA:CF>
CSBC(phy-interface)# done
<Output omitted // SBC displaying configuration of phy-interface S1P0!>
CSBC(phy-interface)# quit
CSBC# save
CSBC# activate
```

2.4.3 Network interface configuration

2.4.3.1 BT or BTIP over BVPN

Here we assumed wancom0 and other media-interfaces for IPBx/UC have been configured during initial SBC deployment. Configuration is focused on north side (carrier).

Configuration

```
CSBC# conf t
CSBC(configure)#
CSBC(configure)# system network-interface
CSBC(network-interface)# name S0P0
CSBC(network-interface)# sub-port-id <0 or VLAN_ID>
CSBC(network-interface)# ip-address <SBC Private IP>
CSBC(network-interface)# netmask <Netmask>
CSBC(network-interface)# gateway <GW IP>
<Optional>
CSBC(network-interface)# gw-heartbeat
CSBC(gateway-heartbeat)# select
CSBC(gateway-heartbeat)# state enabled
CSBC(gateway-heartbeat)# heartbeat 1
CSBC(gateway-heartbeat)# retry-count 2
CSBC(gateway-heartbeat)# retry-timeout 1
CSBC(gateway-heartbeat)# health-score 31
CSBC(gateway-heartbeat)# done
<Output omitted // SBC displaying gw-heartbeat configuration!>
CSBC(gateway-heartbeat)# exit
</Optional>
CSBC(network-interface)# done
<Output omitted // SBC displaying network-interface configuration!>
CSBC(network-interface)# quit
```

2.4.3.2 BT or BTIP over Internet

Configuration

```
CSBC# conf t
CSBC(configure)#
CSBC(configure)# system network-interface
CSBC(network-interface)# name S1P0
CSBC(network-interface)# hostname <SBC FQDN>
CSBC(network-interface)# sub-port-id <0 or VLAN_ID>
CSBC(network-interface)# ip-address <SBC Public IP>
CSBC(network-interface)# netmask <Netmask>
CSBC(network-interface)# gateway <GW IP>
CSBC(network-interface)# dns-ip-primary <Primary Public DNS>
CSBC(network-interface)# dns-ip-backup1 <Backup Public DNS>
CSBC(network-interface)# dns-domain <DNS Domain>
<Optional>
CSBC(network-interface)# gw-heartbeat
CSBC(gateway-heartbeat)# select
CSBC(gateway-heartbeat)# state enabled
CSBC(gateway-heartbeat)# heartbeat 1
CSBC(gateway-heartbeat)# retry-count 2
CSBC(gateway-heartbeat)# retry-timeout 1
CSBC(gateway-heartbeat)# health-score 31
CSBC(gateway-heartbeat)# done
```

```
<Output omitted // SBC displaying gw-heartbeat configuration!>
CSBC(gateway-heartbeat)# exit
</Optional>
CSBC(network-interface)# done
<Output omitted // SBC displaying network-interface configuration!>
CSBC(network-interface)# quit
```

Note: if HA is required, please define primary-utility-address and secondary-utility-address as it is a requirement for SBC cluster. Additionally, it is recommended to enable gateway-heartbeat mechanism for the SBC to track the availability of the default gateway of given network-interface (even if the physical interface is up, in some case, default-gateway could be unreachable).

Configuration

```
CSBC(network-interface)# pri-utility-addr <SBC-PRI-IP Address>
CSBC(network-interface)# sec-utility-addr <SBC-SEC-IP Address>
CSBC(network-interface)# gw-heartbeat
CSBC(gateway-heartbeat)# select
CSBC(gateway-heartbeat)# state enabled
CSBC(gateway-heartbeat)# heartbeat 1
CSBC(gateway-heartbeat)# retry-count 2
CSBC(gateway-heartbeat)# retry-timeout 1
CSBC(gateway-heartbeat)# health-score 31
CSBC(gateway-heartbeat)# done
<Output omitted // SBC displaying gw-heartbeat configuration!>
CSBC(gateway-heartbeat)# exit
CSBC(network-interface)# done
<Output omitted // SBC displaying network-interface configuration!>
CSBC(network-interface)# quit
```

2.4.4 Packet marking

To implement packet marking, a media-policy and a class-policy objects must be created. These objects are referenced in the realm-configuration.

This implementation marks DSCP=46 (Expedite Forwarding) both SIP signaling and audio.

2.4.4.1 media-policy

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager media-policy
CSBC(media-policy)# name mark-mp
CSBC(media-policy)# tos-settings
CSBC(tos-settings)# media-type audio
CSBC(tos-settings)# tos-value 0xB8
CSBC(tos-settings)# done
CSBC(tos-settings)# media-type message
CSBC(tos-settings)# media-sub-type SIP
CSBC(tos-settings)# tos-value 0xB8
CSBC(tos-settings)# done
CSBC(tos-settings)# exit
CSBC(media-policy)# done
CSBC(media-policy)# quit
CSBC#
```

2.4.4.2 class-policy

Configuration

```
CSBC# conf t
CSBC(configure)# session-router class-profile
CSBC(class-profile)# policy
CSBC(class-policy)# profile-name mark-cp
CSBC(class-policy)# to-address *
CSBC(class-policy)# media-policy mark-mp
CSBC(class-policy)# done
CSBC(class-policy)# quit
CSBC#
```

2.4.5 Media configuration

2.4.5.1 Media manager configuration

Here are some basic recommendations for media-manager configuration. This element can have additional parameters and options configured depending on your setup.

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# media-manager
CSBC(media-manager-config)# select
CSBC(media-manager-config)# state enabled
CSBC(media-manager-config)# initial-guard-timer 30
CSBC(media-manager-config)# max-untrusted-signaling 1
CSBC(media-manager-config)# min-untrusted-signaling 1
CSBC(media-manager-config)# options +dont-terminate-assoc-legs
CSBC(media-manager-config)# done
<Output omitted // SBC displaying media-manager configuration!>
CSBC(media-manager)# quit
```

Explanations:

- **`initial-guard-timer`**
  - Lowering the value, allows the SBC to tear down a call with no media quicker, as default value is 300 seconds. In other words, if calls is teared down by SBC after 30 seconds, there is no RTP reaching the Customer SBC.
- **`max-untrusted-signaling & min-untrusted-signaling`**
  - Customer SBC is a Peering model. Thus, we recommend lowering the untrusted portion of SBC signaling bandwidth to expand the trusted portion. If your setup is quite complex, and you are hosting several models or use-cases, please contact Oracle for guidance.
- **`options +dont-terminate-assoc-legs`**
  - In specific case (harpinned calls), SBC can delete wrongly both legs of an ongoing session (example: mid-dialog signaling timeout on one leg during a transfer).

2.4.5.2 Codec policy

Codec-policy defines allowed-codecs and their ordering.

It can also activate transcoding (add-codec-on-egress) or transrating (force-ptime). Those cases require DSP or, for virtual SBC, dedicated vCPU assigned as “X” core(s) (aka. transcoding cores for software-based). For DSP sizing exercise, please contact Oracle or your partner.

2.4.5.2.1 BTalk over BVPN

BTalk supports following codecs:

- T.38
- PCMA (G711aLaw)
- PCMU (G711uLaw)
- G729 (Annex b=no)
- telephone-event

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# codec-policy
CSBC(codec-policy)# name CP_BT
CSBC(codec-policy)# allow-codecs "T.38 PCMA PCMU G729 telephone-event
application:no video:no"
CSBC(codec-policy)# order-codecs "PCMA PCMU G729 *"
<Optional>
<!-- Require DSP -->
CSBC(codec-policy)# force-ptime enabled
</Optional>
CSBC(codec-policy)# done
CSBC(codec-policy)# quit
```

Note: Enabling force-ptime parameter will result in DSP consumption to perform transrating.

2.4.5.2.2 BTIP over BVPN

BTIP supports following codecs:

- T.38
- G722
- PCMA (G711aLaw)
- PCMU (G711uLaw)
- G729 (Annex b=no)
- telephone-event

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# codec-policy
CSBC(codec-policy)# name CP_BTIP
CSBC(codec-policy)# allow-codecs "G722 T.38 PCMA PCMU G729 telephone-event
appplication:no video:no"
CSBC(codec-policy)# order-codecs "G722 PCMA PCMU G729 *"
<Optional>
<!-- Require DSP -->
CSBC(codec-policy)# force-ptime enabled
</Optional>
CSBC(codec-policy)# done
CSBC(codec-policy)# quit
```

Note: Enabling force-ptime parameter will result in DSP consumption to perform transrating.

2.4.5.2.3 BTalk over Internet

BTalk over Internet supports following codecs:

- T.38
- PCMA (G711aLaw)
- PCMU (G711uLaw)
- telephone-event

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# codec-policy
CSBC(media-manager)# name CP_BToI
CSBC(codec-policy)# allow-codecs "T.38 PCMA PCMU telephone-event application:no
video:no"
<Optional>
<!-- Require DSP -->
CSBC(codec-policy)# force-ptime enabled
</Optional>
CSBC(codec-policy)# done
CSBC(codec-policy)# quit
```

Note: Enabling force-ptime parameter will result in DSP consumption to perform transrating.

2.4.5.2.4 BTIP over Internet

BTIP over Internet supports following codecs:

- T.38
- PCMA (G711aLaw)
- telephone-event

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# codec-policy
CSBC(media-manager)# name CP_BTIPoI
CSBC(codec-policy)# allow-codecs "T.38 PCMA telephone-event application:no
video:no"
<Optional>
<!-- Require DSP -->
CSBC(codec-policy)# force-ptime enabled
</Optional>
CSBC(codec-policy)# done
CSBC(codec-policy)# quit
```

Note: Enabling force-ptime parameter will result in DSP consumption to perform transrating.

2.4.5.3 RTCP Policy

Orange highly recommends activating RTCP report generation to monitor and assess the quality of the network path between the Customer SBC and Orange BT/ BTIP infrastructure.

Note: This configuration requires DSP. As `rtcp-generate` is set to all-calls, all calls will require a DSP channel even if the media isn't transcoded.

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager rtcp-policy
CSBC(rtcp-policy)# name rtcp
CSBC(rtcp-policy)# rtcp-generate all-calls
CSBC(rtcp-policy)# done
CSBC(rtcp-policy)# quit
```

2.4.6 Global Sip Configuration

2.4.6.1 Sip-config

SIP-Config is a mandatory element for SBC to handle SIP Traffic; it contains global options that will define SBC's behavior.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-config
CSBC(sip-config)# select
CSBC(sip-config)# dialog-transparency disabled
CSBC(sip-config)# trans-expire 12
CSBC(sip-config)# options +max-udp-length=0
CSBC(sip-config)# options +multiple-dialogs-enhancement
CSBC(sip-config)# options +reinvite-trying=yes
CSBC(sip-config)# options +sag-target-uri=ip
CSBC(sip-config)# options +set-inv-exp-at-100-resp
CSBC(sip-config)# add-reason-header enabled
CSBC(sip-config)# extra-method-stats enabled
<Optional>
<!-- Recommended for BT or BTIP over Internet – offers additional stats -->
CSBC(sip-config)# sa-routes-stats enabled
</Optional>
CSBC(sip-config)# done
CSBC(sip-config)# quit
```

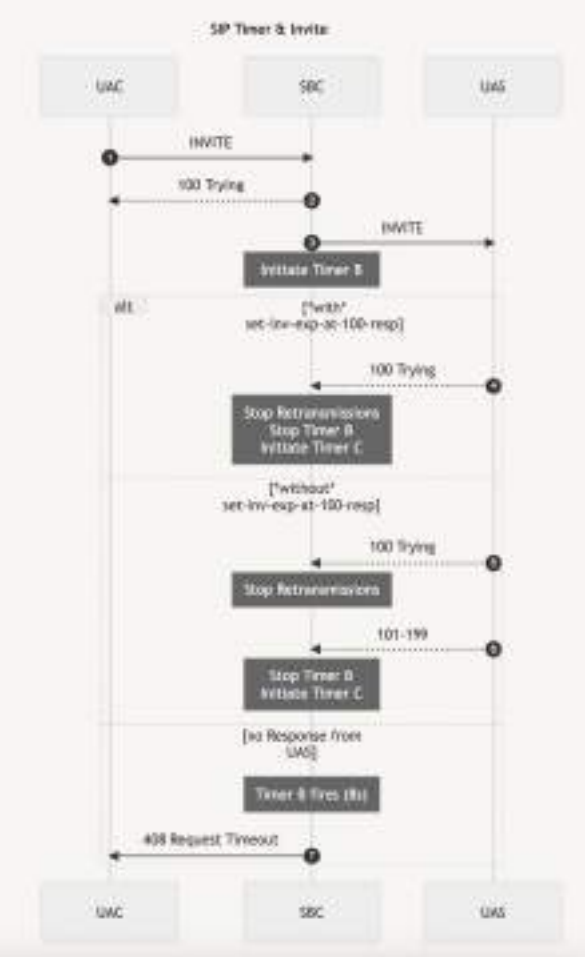
Explanations:

- **`dialog-transparency disabled`**
  - SBC rewrites Call-ID, this is best practices.
- **`trans-expire 12`**
  - This parameter, by default, defines Timer B & Timer D. Reducing its value, enable faster rerouting.
- **`options +max-udp-length=0`**
  - Enables UDP fragmentation based on MTU defined on network-interface or system-config)
  - Without this option, SBC could reject incoming Request with SIP Response Code & Phrase: "513 Message too large".
- **`options +multiple-dialogs-enhancement`**
  - Enables additional capabilities to properly handle Multiple Early Dialogs call flows and is a prerequisite for merging early dialogs (see Realm configuration).

Note: Handling multiple early dialogs is a specific case (for a given INVITE from A, it would result in several 18x with different to-tag from the B parties). Merging those might require DSP to transcode media flows to

properly merge early dialogs. This parameter is highly recommended to match Orange's requirements. But it is required only if your setup have or may have this specific case of multiple early dialogs.

- **`options +reinvite-trying=yes`**
  - Forces SBC to send 100 Trying on reINVITE to avoid retransmissions.
- **`options +sag-target-uri=ip`**
  - Forces SBC to rewrite the host part of the Request-URI to have the remote IP-Address instead of the SAG name.
- **`options +set-inv-exp-at-100-resp`**
  - With the option configured, when receiving a 100 Trying response to the INVITE sent, timer C is started.



- **`add-reason-header`**
  - Enables the system to add the reason header into response messages.
- **`extra-method-stats`**

- Enable or disable the expansion SIP Method tracking feature. For example, it enables the system to track transaction messages for specific SIP session agents, SIP realms, and SIP interfaces.

- **`sa-routes-stats`**
  - Enables collecting session agent statistics for DNS-resolved session agents.

2.4.6.2 SIP enforcement profile

To reject unwanted methods with SIP error response "405 Method Not Allowed" we configure later on an HMR to reject non-supported SIP Methods.

2.4.6.3 SIP features

SIP Feature enables the SBC to edit Supported & Required header to remove unwanted SIP Options tag. The configuration provided as part of these guidelines will remove unsupported SIP Options tags to prevent sending those to Orange Business infrastructure.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-feature
CSBC(sip-feature)# name 100rel
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done

CSBC(sip-feature)# name histinfo
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done

CSBC(sip-feature)# name join
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done

CSBC(sip-feature)# name precondition
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done
```

```
CSBC(sip-feature)# name record-aware
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done

CSBC(sip-feature)# name replaces
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done

CSBC(sip-feature)# name timer
CSBC(sip-feature)# realm OB-<BT_or_BTIP_or_BToI_or_BTIPoI>-Peering
CSBC(sip-feature)# support-mode-inbound Strip
CSBC(sip-feature)# support-mode-outbound Strip
CSBC(sip-feature)# proxy-require-mode-inbound Reject
CSBC(sip-feature)# proxy-require-mode-outbound Reject
CSBC(sip-feature)# done
CSBC(sip-feature)# exit
```

2.4.6.4 Response maps

Response-maps are useful to edit sip-reason-code and sip-reason-phrase to trigger a specific behavior on the hop that is going to receive the SIP error. For example, forcing a 503 error code could enable recursion on Orange Business infrastructure, so the session is tried via a redundant trunk between the customer and Orange Business infrastructure.

Please check § "Rerouting scenario" here.

At this stage, it was not seen as needed in these guidelines.

2.4.6.5 Saving configuration

Before continuing, save the current configuration.

Configuration

```
**CSBC# save-config
```

2.5 Orange BTalk & BTIP Carrier North unencrypted SIP configuration for Oracle E-SBC (UDP)

Through this chapter, regarding BTIP/BTalk over BVPN architecture mentioned in § 2.2.1, describe step by step configuration to interconnect to.

2.5.1 North realm configuration

A north realm (realm-identifier = "OB-<BT_or_BTIP>-Peering") is created once to represent the OB carrier side of the E-SBC and enables media port sharing. This realm is associated with a SIP interface that is shared by all SIP endpoints on the south side of E-SBC.

`mm-in-realm` is required as customer need to anchor media related to his calls, even if calls are no longer controlled by their IPBx/UC.

`Media-policy` is required to mark ToS/DSCP of packets egressing SBC.

`Access-control-trust-level` is set to high, to be sure that signaling from this realm are using trusted queues.

`Codec-policy` is applied to Realm because it is a prerequisite to enable **`merge-early-dialogs`**, to only show one early-dialog to carrier (Orange).

Note: Handling multiple early dialogs is a specific case (for a given INVITE from A, it would result in several 18x with different to-tag from the B parties). Merging those might require DSP to transcode media flows to properly merge early dialogs. This parameter is highly recommended to match Orange's requirements. But it is required only if your setup have or may have this specific case of multiple early dialogs.

`Rtcp-policy` is activated to generate RTCP reports from Customer SBC to carrier, so the network link can be measured properly.

To ensure a proper RTP stream towards the carrier, **`hide-egress-media-update`** is enabled to correct unexpected changes to session continuity information (SSRC / Sequence number).

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager realm-config
CSBC(realm-config)# identifier OB-<BT_or_BTIP>-Peering
CSBC(realm-config)#  network-interfaces <network-interface:vlan-id>     ex.S1P0:0.4
CSBC(realm-config)# mm-in-realm enabled
CSBC(realm-config)# media-policy mark-mp
CSBC(realm-config)# access-control-trust-level high
CSBC(realm-config)# codec-policy CP_<BTIP_or_BT>
CSBC(realm-config)# rtcp-policy rtcp
CSBC(realm-config)# hide-egress-media-update enabled
CSBC(realm-config)# merge-early-dialogs enabled
<Optional>
<!-- Required for BT or BTIP over Internet or if south side is using SRTP -->
CSBC(realm-config)# media-sec-policy RTP
</Optional>
```

2.5.2 North sip-interface configuration

A sip-interface must be associated to the north realm previously defined with the port 5060. Only one SIP interface is used on north side to represent all SIP endpoints from south side (customer's SIP endpoints).

The parameter "allow-anonymous agents-only" enables only a provisioned session-agent to send requests to the E-SBC: messages received from unknown sources will be rejected with "403 Forbidden".

The option "strip-route-headers" removes any "Route" header from received requests (which would be honored by the E-SBC as described in RFC 3261).

The parameter **`Diversion-info-mapping-mode`** set to **`hist2div`** enables the SBC to convert any valid history-info it may receive to Diversion header.

The parameter **`add-sdp-invite`** set to **`both`** enforces SDP insertion, to avoid sending offerless INVITE or reINVITE (aka. Delayed offer) to the carrier.

For reINVITE, SBC will use previous SDP seen, but for out-of-dialog INVITE, SBC will rely on the media-profiles listed under **`add-sdp-profiles`**.

Note: As a reminder BTIP offer only supports PCMA, while BT supports PCMA and PCMU.

The option **`rfc2833-allow-asymmetric-pt`** enables the SBC to support asymmetric RFC 2833 where each participants uses a different payload for DTMF. SBC will convert payload type to interwork and close the gap.

The option **`strip-route-headers`** enables the SBC to remove unwanted Route headers (Security recommendations).

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-interface
CSBC(sip-interface)# state enabled
CSBC(sip-interface)# realm-id OB-<BT_or_BTIP>-Peering
CSBC(sip-interface)# sip-port
CSBC(sip-port)# address <SBC Private IP>
CSBC(sip-port)# port 5060
CSBC(sip-port)# transport-protocol UDP
CSBC(sip-port)# allow-anonymous agents-only
CSBC(sip-port)# done
<Output omitted // SBC displaying configuration of sip-port!>
CSBC(sip-port)# exit
CSBC(sip-interface)# diversion-info-mapping-mode hist2div
CSBC(sip-interface)# add-sdp-invite both
<Info>
```

```
<!-- Required for BT offer only-->
CSBC(sip-interface)# add-sdp-profiles "PCMA PCMU G729 telephone-event"
</Info>
<Info>
<!-- Required for BTIP offer only-->
CSBC(sip-interface)# add-sdp-profiles "PCMA G729 telephone-event"
</Info>
CSBC(sip-interface)# options +rfc2833-allow-asymmetric-pt
CSBC(sip-interface)# options +strip-route-headers
CSBC(sip-interface)# done
CSBC(sip-interface)# quit
```

2.5.3 Steering-pool configuration

A single steering-pool must be provided for north realm and shared by all SIP endpoints, connected on south side, to exchange media on the core network.

The IP address is the same as the one used by north sip-interface.

A maximum of 14,000 ports can be configured, allowing up to 7,000 simultaneous calls, depending on your BT/BTIP voice channel order. Here it is assumed that each call setup an audio session that consumes 2 ports (one for RTP and one for RTCP).

Note: Please adapt the number of ports configured on your steering pool accordingly with the appliance model used or the shape of your virtual SBC. Not all setups can handle 7,000 concurrent calls.

For more details about steering-pool and port allocations, please refer to dedicated chapter on Oracle ESBC configuration guide.

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager steering-pool
CSBC(steering-pool)# ip-address <SBC Private IP>
CSBC(steering-pool)# start-port 6000
CSBC(steering-pool)# end-port 20000
CSBC(steering-pool)# realm-id OB-<BT_or_BTIP>-Peering
CSBC(steering-pool)# done
CSBC(steering-pool)# quit
```

2.5.4 Media-sec-policy

A media-security-policy "RTP" is configured and applied to the north realm. It specifies that media is not encrypted on the north side of the E-SBC. This configuration is necessary when at least one realm is set up to support SRTP.

Configuration

```
CSBC# conf t
CSBC(configure)# security media-security media-sec-policy
CSBC(media-sec-policy)# name RTP
CSBC(media-sec-policy)# done
CSBC(media-sec-policy)# quit
```

2.5.5 BT/BTIP Session objects

2.5.5.1 Session-agents

A session-agent must be configured to represent each address by which BT/BTIP infrastructure can be targeted. The availability of any address is monitored through the periodic OPTIONS mechanism.

The session-agent is put "out-of-service" in case it doesn't answer a ping-transaction (OPTIONS must sent every 300 sec minimum), or it doesn't answer two subsequent non-ping transactions (option **`trans-timeouts`**) and will be put back "in-service" as soon as it starts sending SIP traffic or it answers a ping-transaction.

The parameter **`ping-response`** enables SBC to answer to SIP Options probes, without the need of HMRs.

The **`manipulation-string`** is later referenced in HMR, in case no subscriber numbers is sent in case of an anonymous call. Thus customer needs to define the default site DID (E.164 format) to provide proper location. Customer needs to make sure this number is properly routed in case of callback.

For more info you can refer to this chapter.

- Nominal BT session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BT_or_BTIP>-NOMINAL-SA     ex: BT or BTIP NOMINAL SA
CSBC(session-agent)# ip-address <BT_or_BTIP_NOMINAL_SA_IP>       ex: 82.82.24.71
CSBC(session-agent)# port 5060
CSBC(session-agent)# description "OB Nominal"
CSBC(session-agent)# transport-method UDP
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-<BT_or_BTIP>-Peering
CSBC(session-agent)# ping-method OPTIONS;hops=0
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# out-manipulationid OUT_TO_<BT_or_BTIP>
CSBC(session-agent)# manipulation-string <DefaultSiteDID>
CSBC(session-agent)# done
CSBC(session-agent)# quit
```

- Backup BT session-agent

The same applies to the backup BT/BTIP SA: SIP termination must be configured. Please follow the instructions below.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BT_or_BTIP>-BACKUP-SA      ex: BT_or_BTIP_BACKUP_SA
CSBC(session-agent)# ip-address <BT_or_BTIP_BACKUP_SA_IP>        ex: 82.82.24.72
CSBC(session-agent)# port 5060
CSBC(session-agent)# description "OB Backup"
```

```
CSBC(session-agent)# transport-method UDP
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-<BT_or_BTIP>-Peering
CSBC(session-agent)# ping-method OPTIONS;hops=0
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# out-manipulationid OUT_TO_<BT_or_BTIP>
CSBC(session-agent)# manipulation-string <DefaultSiteDID>
CSBC(session-agent)# done
CSBC(session-agent)# quit
```

2.5.5.2 Session-agent groups

One SAG is configured with both nominal and backup session agents operating in hunt mode. Round-robin might be permitted upon explicit request to Orange.

When a call is to be routed to BT/BTIP, the nominal BT/BTIP SA is selected as the primary destination. In case of no response, or if a SIP 408 or 5xx error is received, the call is rerouted to the backup BT/BTIP SA.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-group
CSBC(session-agent-group)# group-name OB-<BT_or_BTIP>
CSBC(session-agent-group)# dest(<BT_or_BTIP-NOMINAL_SA_IP> <BT_or_BTIP-BACKUP-SA_IP>
CSBC(session-agent-group)# strategy <hunt>
<Optional>
CSBC(session-agent-group)# strategy <roundRobin>
</Optional>
CSBC(session-agent-group)# sag-recursion enabled
CSBC(session-agent-group)# stop-sag-recurse 400-407,409-499
CSBC(session-agent-group)# app-protocol SIP
CSBC(session-agent-group)# done
CSBC(session-agent-group)# quit
CSBC#
```

2.5.5.3 Access control

For each configured session-agent, an access-control entry is created, specifying the session-agent's IP address as the source-address, and the IP address of the SIP interface associated with the E-SBC as the destination-address. Any signaling packet whose source or destination address does not match a configured access-control entry will be discarded at the IP level.

- Access control for nominal BT session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router access-control
CSBC(access-control)# source-address <BT_or_BTIP_NOMINAL_SA_IP>    ex: 82.82.24.71
CSBC(access-control)# destination-address <C-SBC_NOMINAL_IP>    ex: 138.132.169.2
CSBC(access-control)# realm-id OB-<BT_or_BTIP>-Peering
CSBC(access-control)# description "OB Nominal"
CSBC(access-control)# application-protocol SIP
CSBC(access-control)# access permit
CSBC(access-control)# trust-level high
CSBC(access-control)# transport-protocol UDP
CSBC(access-control)# done
```

```
CSBC(access-control)# quit
```

- Access control for backup BT session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router access-control
CSBC(access-control)# source-address <BT_or_BTIP_BACKUP_SA_IP>     ex: 82.82.24.72
CSBC(access-control)# destination-address <C-SBC_NOMINAL_IP>    ex: 138.132.169.2
CSBC(access-control)# realm-id OB-<BT_or_BTIP>-Peering
CSBC(access-control)# description "OB Secours"
CSBC(access-control)# application-protocol SIP
CSBC(access-control)# access permit
CSBC(access-control)# trust-level high
CSBC(access-control)# transport-protocol UDP
CSBC(access-control)# done
CSBC(access-control)# quit
```

2.5.6 Provisioning BTalk/BTIP on a backup E-SBC

Configure the backup E-SBC following the same procedure from § 2.4 used for the nominal E-SBC.

2.5.7 Configuration changes to be made on the south side

The aim of this paragraph is to describe the adjustments that may be necessary on the south side of the E-SBC to fully implement certain Oracle features required to comply with Orange Business requirements.

Note: At this stage, Oracle did not apply specific changes on south side to fulfill Orange's requirements. Depending on customer setup, some additional manipulations or configuration changes might be needed. In case assistance is needed, please refer to your partner or Oracle directly.

Please note that Oracle Applications Notes are available *here*.

2.5.8 SIP header manipulation

For unencrypted BTalk/BTIP SIP trunk architecture, message manipulations are required on outgoing messages toward Orange BTalk/BTIP, and possibly on incoming messages as well.

These manipulations rules are detailed in the chapter

Note: By default, SBC doesn't prompt the unsaved (**) or unapplied (*) changed. To enable it, use CLI command: prompt-enabled enabled

Configuration

```
CSBC# prompt-enabled enabled
```

2.5.8.1 SIP Message Manipulation

For encrypted BTalk/BTIP SIP Trunk architecture, it is required to implement some Message Manipulation for the outgoing message toward Orange BTalk/BTIP. Those Manipulations Rules are detailed on the chapter 2.7 SIP rules and manipulations. Please jump to this Chapter directly.

SIP rules and manipulations". Please refer directly to that chapter.

2.5.9 Local policy

2.5.9.1 From north to south

A local policy must be created to route incoming calls from the BTalk/BTIP infrastructure to the appropriate customer SIP endpoint on the south side.

The local policy from north to south includes two next hops: the nominal group and the backup group of the Customer's UC/IPBX. The E-SBC will first attempt to route calls to the nominal group; only if this fails (or all nominal elements are out of service) will a second attempt be made to route calls to the backup group. For clarity, a cost of 1 is assigned to the backup group route (compared to 0 for the nominal group).

The **`action replace-uri`** is to force SBC to rewrite the Request-URI with the next hop details.

This local policy should be adapted to the Customer's specific environment. The configuration provided here is for example purposes only.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router local-policy
CSBC(local-policy)# from-address *
CSBC(local-policy)# to-address *
CSBC(local-policy)# source-realm OB_<BT_or_BTIP>-Peering
CSBC(local-policy)# policy-attribute
CSBC(local-policy-attributes)# next-hop SAG:<SOUTH_SIDE_NOMINAL_SAG>
CSBC(local-policy-attributes)# realm <SOUTH_SIDE_REALM>
CSBC(local-policy-attributes)# app-protocol SIP
CSBC(local-policy-attributes)# action replace-uri
CSBC(local-policy-attributes)# done
CSBC(local-policy-attributes)# next-hop SAG:<SOUTH_SIDE_BACKUP_SAG>
CSBC(local-policy-attributes)# realm <SOUTH_SIDE_REALM>
CSBC(local-policy-attributes)# cost 1
CSBC(local-policy-attributes)# app-protocol SIP
CSBC(local-policy-attributes)# action replace-uri
CSBC(local-policy-attributes)# done
CSBC(local-policy-attributes)# exit
CSBC(local-policy)# done
CSBC(local-policy)# quit
```

2.5.9.2 From south to north

A local policy must be created to route outgoing calls from the Customer's SIP endpoints to the BTalk/BTIP infrastructure.

The local policy from south to north uses a single next-hop, which is the group containing both the nominal and backup BTalk/BTIP A-SBCs.

The next hop for BTalk/BTIP SIP infrastructure is OB-SBC (i.e. Session-agent group for BTalk/BTIP A-SBCs).

The **`action replace-uri`** is to force SBC to rewrite the Request-URI with the next hop details.

Note: This local policy should be adapted to the Customer's specific environment. The configuration provided here is for example purposes only.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router local-policy
CSBC(local-policy)# from-address *
CSBC(local-policy)# to-address *
CSBC(local-policy)# source-realm <SOUTH_SIDE_REALM>
CSBC(local-policy)# policy-attribute
CSBC(local-policy-attributes)# next-hop SAG:OB-<BT_or_BTIP>
CSBC(local-policy-attributes)# realm OB_<BT_or_BTIP>-Peering
CSBC(local-policy-attributes)# app-protocol SIP
CSBC(local-policy-attributes)# action replace-uri
CSBC(local-policy-attributes)# done
CSBC(local-policy-attributes)# exit
CSBC(local-policy)# done
CSBC(local-policy)# quit
```

2.5.9.3 Saving and activating configuration

Before continuing, save and activate the current configuration.

Configuration

```
**CSBC# save-config
*CSBC# activate-config
CSBC#
```

Note: By default, SBC doesn't prompt the unsaved (**) or unapplied (*) changed. To enable it, use CLI command: `prompt-enabled enabled`

Configuration

```
CSBC# prompt-enabled enabled
```

2.5.9.4 SIP Message Manipulation

For unencrypted BTalk/BTIP SIP Trunk architecture, it is required to implement some Message Manipulation for the outgoing message toward Orange BTalk/BTIP. Those Manipulations Rules are detailed on the chapter 2.7 SIP rules and manipulations. Please jump to this Chapter directly.

2.6 Orange BTalk over Internet & BTIP over Internet Carrier North encrypted SIP configuration for Oracle SBC (TLS)

Through this chapter, regarding BTIP /BTalk over Internet architecture mentioned in § 2.2.2, describe step by step configuration to interconnect to.

In encrypted SIP Trunking deployments, the northbound SIP interface of the E-SBC is typically accessible from the public internet via the DMZ.

As such, it is imperative to implement robust security measures to protect this interface from potential malicious traffic and signaling-based attacks.

To harden your configuration, valuable information can be found in Oracle E-SBC Security Guide, Oracle E-SBC admin Security Guide and from Oracle Application notes (Best Current Practices).

2.6.1 E-SBC certificate

The E-SBC certificate is associated with the SIP interface of the north realm. The certificate is stored in a certificate-record object, which must be created with the following details:

- Country, State, Locality: These fields should reflect the geographical location of the E-SBC.
- Organization: Name of the company.
- Common Name (CN): Only applicable if a FQDN is required.
- Extended Key Usage: serverAuth
- Key Size: Set at least the key length to 2048 bits.

2.6.1.1 Certificate-record

Configuration

```
CSBC# conf t
CSBC(configure)# security certificate-record
CSBC(certificate-record)# name <CUST_CERT_REC>
CSBC(certificate-record)# country <CUST_COUNTRY>
CSBC(certificate-record)# state <CUST_STATE>
CSBC(certificate-record)# locality <CUST_LOCALITY>
CSBC(certificate-record)# organization <CUST_ORGANIZATION>
CSBC(certificate-record)# common-name <CUST_E-SBC_FQDN>
CSBC(certificate-record)# extended-key-usage-list serverAuth
CSBC(certificate-record)# key-size 2048
CSBC(certificate-record)# done
<Output omitted // SBC displaying configuration of certificate-record!>
CSBC(certificate-record)# quit
CSBC#
```

At this point, the configuration must be saved and activated. This is required in order to perform the next step: generating the CSR.

Configuration

```
CSBC# save-config
CSBC# activate-config
```

2.6.1.2 CSR generation

Configuration

```
CSBC# generate-certificate-request <CUST_CERT_REC>
Generating Certificate Signing Request. This can take several minutes....

-----BEGIN CERTIFICATE REQUEST-----
MIIC+jCCAeICAQAweDELMAkGA1UEBhMCRlIxDDAKBgNVBAgTA04vQTEYMBYGA1UE
BxQPQ2Vzc2lvbl9TZXZpZ25lMQ8wDQYDVQQKEwZPcmFuZ2UxITAfBgNVBAsTGE9y
YW5nZSBidXNpbmVzcyBTZXJ2aWNlczENMAsGA1UEAxMEQ1NCQzCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBALK3Jqz99IYjLQa6MeD4IPGl3OhLQ5lkyIaB
NRqQceh31nxpBMp6033n1RnG1Xc4DASK7DIiGWny55A3CvwKHWreC492my6PUT7D
Zsl3w7jIYvos4KHBTZd+Z2RKdRzL1wwJvHnKWtdX+dq6ibVw9WimtQvIi3Qa3bS0
efQTzfSgx+9oTbe5RKatpW8UD9pEEqOxjU6kLH36D01IgSerPaR0EE0dfqtKBZIf
AkessCbUePb+TgpNqpJ2JlstytvmZx2eS1w0NkdTkU872ntgFEj5UQh79/J5efLk
s9KNHVYNGchwDOFnM7PIglSu720PFFths2nL5YEmruSyM96yysCAwEAAaA9MDsG
CSqGSIb3DQEJDjEuMCwwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMB
BggrBgEFBQcDAjANBgkqhkiG9w0BAQsFAAOCAQEAYXnA7djcvExFkKreAKdTRNnq
hJdtIJn8SjkLfmewiNOhFL/nau8NZs3aer75sBNt/KtfIbU3Onl9CoGh+bLajlxd
c9fELKl5i4xQoNtRusBL5MoL30aVijfGRFcCaH48lrDynJ8iB4RL9gFyERwwGlPO
NRdTL8ujtr9Hb6DlaeDP0G61+nePKvEp75ubhHIRImdciwTxXXL3cGxcSsxdR68C
emG+iwAs7Q/rdJ6+RcqhK8bhV8LtekOeG+LVzmDWyoGadjSdVP77eqxTogzf+i1T
pLNrSYt91nrMOOdTpVTbqdp3dDVOjF1itOSADKUgZg81ADi+y7v5ra5enW8cwg==
-----END CERTIFICATE REQUEST-----

WARNING: Configuration changed, run "save-config" command.
```

The CSR generation includes also the key pair creation (public and private keys).

Save and activate the configuration:

Configuration

```
CSBC# save-config
CSBC# activate-config
```

Copy and paste the CSR (including the 2 lines with "BEGIN CERTIFICATE REQUEST" and "END CERTIFICATE REQUEST") into a text file named <CUSTOMER_NAME>_<SBC_HOSTNAME.csr> (PKCS10 PEM format).

2.6.1.3 Submitting the CSR to a trusted public Certificate Authority (CA)

Submit the CSR to a trusted public Certificate Authority (CA) that meets your preferred security level, as Orange imposes no restrictions on the client's choice of CA.

Once signed by your CA, download the signed certificate (PKCS7 or X509v3 format) in PEM format.

Open it in a text editor, it must look like this example:

```
-----BEGIN CERTIFICATE-----
MIIDQjCCAiqgAwIBAgIGAU9AQSPHMA0GCSqGSIb3DQEBBQUAMEwxCzAJBgNVBAYT
AkZSMRgwFgYDVQQKDA9KTENPcmdhbmlzYXRpb24xFTATBgNVBAsMDEpMQ0F1dGhv
cml0eTEMMAoGA1UEAwwDSkxDMB4XDTE1MDgxODEwMDA1NFoXDTE2MDgxNzEwMDA1
NlowgYAxCzAJBgNVBAYTAkZyMQwwCgYDVQQIEwNOL0ExFzAVBgNVBAcTDkNlc3Nv
bi1TZXZpZ25lMQ8wDQYDVQQKEwZPcmFuZ2UxITAfBgNVBAsTGE9yYW5nZSBCdXNp
bmVzcyBTZXJ2aWNlczEWMBQGA1UEAxMNMTcyLjIyLjIzMi40OTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwffkCgYEAxDjw/Of3a7NK0KrL2J5e4ke9c0A4UdcW1LeudbiP
Fd9LKWj1OBgG9nmjMMM/I3CGGGRWS1AemaHQocPZTmRVvTSV812bh/AIX7rFEmHT
VqwQwvUro6pJ5BkhmYVOfVavheT95BaQjInDOYWzniHaURt1ZF6iiFLzUYwvyHzg
i9kCAwEAAaN5MHcwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjAfBgNVHSMEGDAWgBR7V5fff9VlSUpp+/DSQYMCG1yjuTAdBgNVHQ4E
FgQUPqevs3dI/4/1bvDRBB0A+g4kWMowCQYDVR0TBAIwADANBgkqhkiG9w0BAQUF
AAOCAQEAaRTsnnCkT3PyL9tkVYZqTwcihB+P3qLl2CReguTfL1WrPGT0XeDtPX8b
HyM7tY7ShJr65YFHUIpWVvhla8wMYHWfod5cUrjiXh7xkz/VEBIIh95U1O8FcUzW
RwSjiGAu/V0XrjUNq6SdedcjMnpPynETbjjHSQtJnDTwff8lmIAsLbTo5fBdgl1e
pVOKQ49xGRFCvxvIwgQUniqnolHowtt3CTtliRTI/fqNKFwHB06+VlAhGEkIn2rH
/9iCW+N42MYSUE3Pwyh+o55R/VwPgVfAoohNqD0JEKVEenQfLqQ8w8dv+Jd71BQl
y5C+L5mm5HGce5OjBfZJu6+djI5ttA==
-----END CERTIFICATE-----
```

2.6.1.4 Importing the signed certificate

The signed certificate (<CUST_CERT.pem>) must now be imported using the import-certificate command.

- With the text editor, copy the entire certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines.
- Run the import-certificate command, paste the whole certificate content, and finish by entering ";" to indicate the end of the certificate:

Configuration

```
CSBC# import-certificate try-all <CUST_CERT_REC>
IMPORTANT:
Please enter the certificate in the PEM format.
Terminate the certificate with ";" to exit.......
-----BEGIN CERTIFICATE-----
MIIDQjCCAiqgAwIBAgIGAU9AQSPHMA0GCSqGSIb3DQEBBQUAMEwxCzAJBgNVBAYT
AkZSMRgwFgYDVQQKDA9KTENPcmdhbmlzYXRpb24xFTATBgNVBAsMDEpMQ0F1dGhv
cml0eTEMMAoGA1UEAwwDSkxDMB4XDTE1MDgxODEwMDA1NFoXDTE2MDgxNzEwMDA1
NlowgYAxCzAJBgNVBAYTAkZyMQwwCgYDVQQIEwNOL0ExFzAVBgNVBAcTDkNlc3Nv
bi1TZXZpZ25lMQ8wDQYDVQQKEwZPcmFuZ2UxITAfBgNVBAsTGE9yYW5nZSBCdXNp
bmVzcyBTZXJ2aWNlczEWMBQGA1UEAxMNMTcyLjIyLjIzMi40OTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAxDjw/Of3a7NK0KrL2J5e4ke9c0A4UdcW1LeudbiP
Fd9LKWj1OBgG9nmjMMM/I3CGGGRWS1AemaHQocPZTmRVvTSV812bh/AIX7rFEmHT
VqwQwvUro6pJ5BkhmYVOfVavheT95BaQjInDOYWzniHaURt1ZF6iiFLzUYwvyHzg
i9kCAwEAAaN5MHcwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjAfBgNVHSMEGDAWgBR7V57Zf9VlSUpp+/DSQYMCG1yjuTAdBgNVHQ4E
FgQUPqevs3dI/4/1bvDRBB0A+g4kWMowCQYDVR0TBAIwADANBgkqhkiG9w0BAQUF
AAOCAQEAaRTsnnCkT3PyL9tkVYZqTwcihB+P3qLl2CReguTfL1WrPGT0XeDtPX8b
```

```
HyM7tY7ShJr65YFHUIpWVvhla8wMYHWfod5cUrjiXh7xkz/VEBIIh95U1O8FcUzW
RwSjiGAu/V0XrjUNq6SdedcjMnpPynETbjjHSQtJnDTwyo8lmIAsLbTo5fBdgl1e
pVOKQ49xGRFCvxvIwgQUniqnolHowtt3CTtliRTI/fqNKFwHB06+VlAhGEkIn2rH
/9iCW+N42MYSUE3Pwyh+o55R/VwPgVfAoohNqD0JEKVEenQfLqQ8w8dv+Jd71BQl
y5C+L5mm5HGce5OjBfZJu6+djI5ttA==
-----END CERTIFICATE-----
;
Certificate imported successfully....
WARNING: Configuration changed, run "save-config" command.
CSBC# save-config
CSBC# activate-config
```

The details of the signed certificate can now be displayed using the following command:

Configuration

```
CSBC# show security certificates detail <CUST_CERT_REC>
Certificate:
    Data:
        Version: 1 (0x0)
        Serial Number: 3f:58:13:93:5d:36:f9:27:28:68:72
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:
            C=FR
            ST=Bretagne
            L=Cesson-Sevigne
            O=Orange
            OU=Orange Business
            CN=ca.orange-business.com
        Validity
            Not Before: May 18 07:18:37 2021 GMT
            Not After: May 17 07:18:37 2026 GMT
        Subject:
            C=FR
            ST=Bretagne
            L=Cesson-Sevigne
            O=Orange
            OU=Orange Business
            CN=csbc.btip.orange-business.com
```

2.6.2 Customer CA certificate(s)

The procedure described in this chapter must be followed for each certificates authority used to sign the customer certificate (<CUST_CERT.pem>), including both Root and Intermediate CA certificates.

2.6.2.1 Certificate-records

The certificate records that will store the CA certificates must be configured first:

Configuration

```
CSBC# conf t
CSBC(configure)# security certificate-record
CSBC(certificate-record)# name CACERT_<CUST_CA_NAME>
CSBC(certificate-record)# done
Warning: Required field "common-name" is empty
Do you still want to save configuration [y/n]?: y
certificate-record
        name                            CACERT_<CUST_CA_NAME>
        country                         US
        state                           MA
        locality                        Burlington
        organization                    Engineering
        unit
        common-name
        key-size                        2048
        alternate-name
        trusted                         enabled
        key-usage-list                  digitalSignature
                                        keyEncipherment
        extended-key-usage-list         serverAuth
        key-algor                       rsa
        digest-algor                    sha256
        ecdsa-key-size                  p256
        cert-status-profile-list
        options
        last-modified-by                admin@112.22.217.86
        last-modified-date              2025-06-20 17:16:19
CSBC(certificate-record)# quit
CSBC#
```

2.6.2.2 Importing Customer CA Certificates

Next, customer CA certificates (Root & Intermediate) (<CUST_CACERT.pem>) must be imported using the import-certificate command.

- With the text editor, copy the entire certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines.
- Run the import-certificate command, paste the whole certificate content, and finish by entering ";" to indicate the end of the certificate:

Configuration

```
CSBC# import-certificate try-all CACERT_<CUST_CA_NAME>
IMPORTANT:
Please enter the certificate in the PEM format.
Terminate the certificate with ";" to exit.......
-----BEGIN CERTIFICATE-----
MIIEuTCCA6GgAwIBAgIQAp7/0k8LfSOZNZubH9bz3jANBgkqhkiG9w0BAQUFADBs
```

```
AkZSMRgwFgYDVQQKDA9KTENPcmdhbmlzYXRpb24xFTATBgNVBAsMDEpMQ0F1dGhv
cml0eTEMMAoGA1UEAwwDSkxDMB4XDTE1MDgxODEwMDA1NFoXDTE2MDgxNzEwMDA1
NlowgYAxCzAJBgNVBAYTAkZyMQwwCgYDVQQIEwNOL0ExFzAVBgNVBAcTDkNlc3Nv
bi1TZXZpZ25lMQ8wDQYDVQQKEwZPcmFuZ2UxITAfBgNVBAsTGE9yYW5nZSBCdXNp
bmVzcyBTZXJ2aWNlczEWMBQGA1UEAxMNMTcyLjIyLjIzMi40OTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAxDjw/Of3a7NK0KrL2J5e4ke9c0A4UdcW1LeudbiP
Fd9LKWj1OBgG9nmjMMM/I3CGGGRWS1AemaHQocPZTmRVvTSV812bh/AIX7rFEmHT
VqwQwvUro6pJ5BkhmYVOfVavheT95BaQjInDOYWzniHaURt1ZF6iiFLzUYwvyHzg
i9kCAwEAAaN5MHcwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjAfBgNVHSMEGDAWgBR7V57Zf9VlSUpp+/DSQYMCG1yjuTAdBgNVHQ4E
FgQUPqevs3dI/4/1bvDRBB0A+g4kWMowCQYDVR0TBAIwADANBgkqhkiG9w0BAQUF
AAOCAQEAaRTsnnCkT3PyL9tkVYZqTwcihB+P3qLl2CReguTfL1WrPGT0XeDtPX8b
HyM7tY7ShJr65YFHUIpWVvhla8wMYHWfod5cUrjiXh7xkz/VEBIIh95U1O8FcUzW
RwSjiGAu/V0XrjUNq6SdedcjMnpPynETbjjHSQtJnDTwyo8lmIAsLbTo5fBdgl1e
pVOKQ49xGRFCvxvIwgQUniqnolHowtt3CTtliRTI/fqNKFwHB06+VlAhGEkIn2rH
viAoT4LnGWrGK71LcJl1UdZD7GuPp4hGmrf0ReXUiqXzLD4Yt88LR95Xecx/K8Wt
2mgPG/cieHl9M/NZ/g==
-----END CERTIFICATE-----
;
Certificate imported successfully....
WARNING: Configuration changed, run "save-config" command.
CSBC# save-config
CSBC# activate-config
```

The details of the signed certificate can now be displayed using the following command:

Configuration

```
CSBC# show security certificates detail CACERT_<CUST_CA_NAME>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6b:39:16:72:ef:6d:FF:f8:4b:9e:c9:58:ee:be:91:0f:1d:93:03:99
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:
            C=FR
            ST=Ile de France
            L=Paris
            O=SecureTrust Authority
            OU=Certification Services
            CN=SecureTrust CA
        Validity
            Not Before: Mar 5 08:58:06 2024 GMT
            Not After: Mar  13 08:58:06 2034 GMT
        Subject:
            C=FR
            ST=Ile de France
            L=Paris
            O=SecureTrust Authority

            OU=Certification Services
            CN=SecureTrust CA
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                CA:20:CF:F5:9F:FF:54:66:EE:53:7F:75:80:6D:2E:C3:40:16:AF:99
            X509v3 Authority Key Identifier:
                keyid:CA:20:CF:F5:9F:BB:54:66:EE:53:7F:FF:80:6D:2E:C3:40:16:AF:99

            X509v3 Basic Constraints: critical
                CA:TRUE
```

2.6.3 Orange CA certificate(s)

The procedure described in this chapter must be followed for each certificate's authority provided by the Orange BT/BTIP team (Root and intermediate CA certificates).

2.6.3.1 Certificate-records

The certificate records that will store the CA certificate must be configured first:

Configuration

```
CSBC# conf t
CSBC(configure)# security certificate-record
CSBC(certificate-record)# name CACERT_<ORANGE_CA_NAME>
CSBC(certificate-record)# done
Warning: Required field "common-name" is empty
Do you still want to save configuration [y/n]?: y
certificate-record
        name                            CACERT_<ORANGE_CA_NAME>
        country                         US
        state                           MA
        locality                        Burlington
        organization                    Engineering
        unit
        common-name
        key-size                        2048
        alternate-name
        trusted                         enabled
        key-usage-list                  digitalSignature
                                        keyEncipherment
        extended-key-usage-list         serverAuth
        key-algor                       rsa
        digest-algor                    sha256
        ecdsa-key-size                  p256
        cert-status-profile-list
        options
        last-modified-by                admin@112.22.217.86
        last-modified-date              2025-06-20 14:16:19
CSBC(certificate-record)# quit
CSBC#
```

2.6.3.2 Importing Orange CA Certificate

Next, the Orange CA certificates (<ORANGE_CACERT.pem>) provided by the Orange BT/BTIP team must be imported using the import-certificate command.

- With the text editor, copy the entire certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines.
- Run the import-certificate command, paste the whole certificate content, and finish by entering ";" to indicate the end of the certificate:

Configuration

```
CSBC# import-certificate try-all CACERT_<ORANGE_CA_NAME>
IMPORTANT:
Please enter the certificate in the PEM format.
Terminate the certificate with ";" to exit.......
```

```
-----BEGIN CERTIFICATE-----
AwIBAQjCCAiqgAwIBAgIGAU9AQSPHMA0GCSqGSIb3DQEBBQUAMEwxCzAJBgNVBAYT
AkZSMRgwFgYDVQQKDA9KTENPcmdhbmlzYXRpb24xFTATBgNVBAsMDEpMQ0F1dGhv
cml0eTEMMAoGA1UEAwwDSkxDMB4XDTE1MDgxODEwMDA1NFoXDTE2MDgxNzEwMDA1
NlowgYAxCzAJBgNVBAYTAkZyMQwwCgYDVQQIEwNOL0ExFzAVBgNVBAcTDkNlc3Nv
bi1TZXZpZ25lMQ8wDQYDVQQKEwZPcmFuZ2UxITAfBgNVBAsTGE9yYW5nZSBCdXNp
bmVzcyBTZXJ2aWNlczEWMBQGA1UEAxMNMTcyLjIyLjIzMi40OTCBnzANBgkqhkiG
9w0BAQEFAAOBjQAwgYkCgYEAxDjw/Of3a7NK0KrL2J5e4ke9c0A4UdcW1LeudbiP
Fd9LKWj1OBgG9nmjMMM/I3CGGGRWS1AemaHQocPZTmRVvTSV812bh/AIX7rFEmHT
VqwQwvUro6pJ5BkhmYVOfVavheT95BaQjInDOYWzniHaURt1ZF6iiFLzUYwvyHzg
i9kCAwEAAaN5MHcwCwYDVR0PBAQDAgWgMB0GA1UdJQQWMBQGCCsGAQUFBwMBBggr
BgEFBQcDAjAfBgNVHSMEGDAWgBR7V57Zf9VlSUpp+/DSQYMCG1yjuTAdBgNVHQ4E
FgQUPqevs3dI/4/1bvDRBB0A+g4kWMowCQYDVR0TBAIwADANBgkqhkiG9w0BAQUF
AAOCAQEAaRTsnnCkT3PyL9tkVYZqTwcihB+P3qLl2CReguTfL1WrPGT0XeDtPX8b
HyM7tY7ShJr65YFHUIpWVvhla8wMYHWfod5cUrjiXh7xkz/VEBIIh95U1O8FcUzW
RwSjiGAu/V0XrjUNq6SdedcjMnpPynETbjjHSQtJnDTwyo8lmIAsLbTo5fBdgl1e
pVOKQ49xGRFCvxvIwgQUniqnolHowtt3CTtliRTI/fqNKFwHB06+VlAhGEkIn2rH
/9iCW+N42MYSUE3Pwyh+o55R/VwPgVfAoohNqD0JEKVEenQfLqQ8w8dv+Jd71BQl
y5C+L5mm5HGce5OjBfZJu6+djAwIBA==
-----END CERTIFICATE-----
;
Certificate imported successfully....
WARNING: Configuration changed, run "save-config" command.
CSBC# save-config
CSBC# activate-config
```

The details of the signed certificate can now be displayed using the following command:

Configuration

```
CSBC# show security certificates detail CACERT_<ORANGE_CA_NAME>
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            6b:39:16:72:ef:6d:FF:f8:4b:9e:c9:58:ee:be:91:0f:1d:93:03:07
        Signature Algorithm: sha256WithRSAEncryption
        Issuer:
            C=FR
            ST=Bretagne
            L=Cesson-Sevigne
            O=Orange
            OU=Orange Business
            CN=ca.orange-business.com
        Validity
            Not Before: Mar 11 08:58:06 2024 GMT
            Not After: Mar  6 08:58:06 2044 GMT
        Subject:
            C=FR
            ST=Bretagne
            L=Cesson-Sevigne
            O=Orange
            OU=Orange Business
            CN=ca.orange-business.com
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                CA:20:CF:F5:9F:FF:54:66:EE:53:7F:75:80:6D:2E:C3:40:16:AF:01
            X509v3 Authority Key Identifier:
                keyid:CA:20:CF:F5:9F:BB:54:66:EE:53:7F:FF:80:6D:2E:C3:40:16:AF:01

            X509v3 Basic Constraints: critical
```

```
CA:TRUE
```

2.6.4 TLS profile

Configure a TLS profile by specifying the SBC certificate, the mutual authentication method, and the supported TLS version 1.3 only.

Configuration

```
CSBC# conf t
CSBC(configure)# security tls-profile
CSBC(tls-profile)# name TLS_ORANGE
CSBC(tls-profile)# end-entity-certificate <CUST_CERT_REC>
CSBC(tls-profile)# trusted-ca-certificates CACERT_<ORANGE_CA_NAME>
CSBC(tls-profile)# mutual-authenticate enabled
CSBC(tls-profile)# tls-version tlsv13
CSBC(tls-profile)# cipher-list (TLS_AES_256_GCM_SHA384 TLS_AES_128_GCM_SHA256
TLS_CHACHA20_POLY1305_SHA256)
CSBC(tls-profile)# done
CSBC(tls-profile)# quit
CSBC#
```

2.6.5 SRTP configuration

2.6.5.1 SDES profile

SDES is the key exchange protocol supported by the E-SBC for SRTP. The E-SBC is configured in single-ended SRTP termination mode (meaning the SBC terminate SRTP on north side and use RTP on south side).

We define here a profile with AES/128 bit key for encryption and HMAC/SHA-1 80-bit digest for authentication, which is the default profile. Both RTP and RTCP streams are encrypted.

Configuration

```
CSBC# conf t
CSBC(configure)# security media-security sdes-profile
CSBC(sdes-profile)# name SDES-ORANGE
CSBC(sdes-profile)# crypto-list "AES_CM_128_HMAC_SHA1_80"
CSBC(sdes-profile)# done
```

2.6.6 Media-sec-policy

A media-security-policy "SRTP" is configured and applied to the north realm. It specifies that media is encrypted on the north side of the E-SBC. It calls the SDES profile defined in the previous paragraph.

Configuration

```
CSBC# conf t
CSBC(configure)# security media-security media-sec-policy
CSBC(media-sec-policy)# name SRTP
CSBC(media-sec-policy)# inbound
CSBC(media-sec-inbound)# profile SDES-ORANGE
CSBC(media-sec-inbound)# mode srtp
CSBC(media-sec-inbound)# protocol sdes
CSBC(media-sec-inbound)# hide-egress-media-update disabled
CSBC(media-sec-inbound)# done
CSBC(media-sec-inbound)# exit
CSBC(media-sec-policy)# outbound
```

```
CSBC(media-sec-outbound)# profile SDES-ORANGE
CSBC(media-sec-outbound)# mode srtp
CSBC(media-sec-outbound)# protocol sdes
CSBC(media-sec-outbound)# done
CSBC(media-sec-outbound)# exit
CSBC(media-sec-policy)# done
CSBC(media-sec-policy)# quit
```

2.6.7 North realm configuration

A north realm (realm-identifier = " OB-<BToI_or_BTIPoI>-Peering ") is created once to represent the OB carrier part of the E-SBC and provides media-ports sharing. This realm is associated with a SIP-interface that is common for all SIP endpoints of south side (customer's SIP endpoints).

When the customer uses SRTP, the media-sec-policy is configured with the value "SRTP".

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager
CSBC(media-manager)# realm-config
CSBC(realm-config)# identifier OB-<BToI_or_BTIPoI>-Peering
CSBC(realm-config)#  network-interfaces  <network-interface:vlan-id>     ex.S1P0:0.4
CSBC(realm-config)# mm-in-realm enabled
CSBC(realm-config)# media-policy mark-mp
CSBC(realm-config)# access-control-trust-level high
CSBC(realm-config)# codec-policy CP_<BTIPoI_or_BToI>
CSBC(realm-config)# rtcp-policy rtcp
CSBC(realm-config)# hide-egress-media-update enabled
CSBC(realm-config)# merge-early-dialogs enabled
<!-- Be sure to set a media-sec-policy (RTP or SRTP) on all realm -->
CSBC(realm-config)# media-sec-policy SRTP
CSBC(realm-config)# done
CSBC(realm-config)# quit
```

2.6.8 North sip-interface configuration

A sip-interface must be associated to the north realm previously defined with the port 5061. Only one SIP interface is used on north side to represent all customer's SIP endpoints in front of BT/BTIP.

The parameter "allow-anonymous agents-only" enables only a provisioned session-agent to send requests to the E-SBC: messages received from unknown sources will be rejected with "403 Forbidden".

The option "strip-route-headers" removes any "Route" header from received requests (which would be honored by the E-SBC as described in RFC 3261).

Configuration

```
CSBC# conf t
CSBC(configure)# session-router
CSBC(session-router)# sip-interface
CSBC(sip-interface)# state enabled
CSBC(sip-interface)# realm-id OB-<BToI_or_BTIPoI>-Peering
CSBC(sip-interface)# sip-port
CSBC(sip-port)# address <SBC_Public_IP>
CSBC(sip-port)# port <5061 or agreed port>
CSBC(sip-port)# transport-protocol TLS
```

```
CSBC(sip-port)# tls-profile TLS_ORANGE
CSBC(sip-port)# allow-anonymous agents-only
CSBC(sip-port)# done
<Output omitted // SBC displaying configuration of sip-port!>
CSBC(sip-port)# exit
CSBC(sip-interface)# diversion-info-mapping-mode hist2div
CSBC(sip-interface)# add-sdp-invite both
<Info>
<!-- Required for BToI offer only-->
CSBC(sip-interface)# add-sdp-profiles "PCMA PCMU telephone-event"
</Info>
<Info>
<!-- Required for BTIPoI offer only-->
CSBC(sip-interface)# add-sdp-profiles "PCMA telephone-event"
</Info>
CSBC(sip-interface)# options +rfc2833-allow-asymmetric-pt
CSBC(sip-interface)# options +strip-route-headers
CSBC(sip-interface)# done
CSBC(sip-interface)# quit
```

2.6.9 Steering-pool configuration

A single steering-pool must be provided for north realm and shared by all SIP endpoints, connected on south side, to exchange media on the core network.

The IP address is the same as the one used by north sip-interface.

A maximum of 14,000 ports can be configured, allowing up to 7,000 simultaneous calls, depending on your BT/BTIP voice channel order. Here it is assumed that each call setup an audio session that consumes 2 ports (one for RTP and one for RTCP).

Note: Please adapt the number of ports configured on your steering pool accordingly with the appliance model used or the shape of your virtual SBC. Not all setups can handle 7,000 concurrent calls.

For more details about steering-pool and port allocations, please refer to dedicated chapter on Oracle ESBC configuration guide.

Configuration

```
CSBC# conf t
CSBC(configure)# media-manager steering-pool
CSBC(steering-pool)# ip-address <SBC_Public_IP>
CSBC(steering-pool)# start-port 6000
CSBC(steering-pool)# end-port 20000
CSBC(steering-pool)# realm-id OB-<BToI_or_BTIPoI>-Peering
CSBC(steering-pool)# done
CSBC(steering-pool)# quit
```

2.6.10 BToI/BTIPoI Session objects

A session-agent must be configured to represent each address by which BToI/BTIPoI infrastructure can be targeted. The availability of any address is monitored through the periodic OPTIONS mechanism.

The session-agent is put "out-of-service" in case it doesn't answer a ping-transaction (OPTIONS sent every 300 sec), or it doesn't answer two subsequent non-ping transactions and will be put back "in-service" as soon as it starts sending SIP traffic or it answers a ping-transaction.

The table below summarizes the different address resolution methods that can be used to access Orange Business core network, depending on the offer:

Offer / Address Resolution Type	Public IP address	DNS A record	DNS SRV record
BToI (International)			+
BTIPoI (France)	+		

When using DNS resolution, two configuration scenarios are considered, based on the public DNS query method. These scenarios differ in concept:

- Setup using a DNS A record

With DNS Type A records, the two remote FQDNs (Nominal and Backup) must be declared as remote peers in dedicated Session Agents (each DNS Type A record resolves to a unique IP address; a nominal or a backup node).

- Setup using a DNS SRV record

With a DNS SRV record, only one remote FQDN needs to be declared as a remote peer in a dedicated Session Agent (a DNS SRV record resolves to two IP addresses: a nominal and a backup node).

Note: DNS configuration is usually done directly on the network-interface facing internet using public dns-servers or Orange ones directly. But in some cases, one may prefer to use an internal network-interface to lookup DNS records. This can be done using the **`dns-realm`** parameter. For more information, please refer to Oracle Configuration Guide.

2.6.10.1 Orange BToI (SIP/TLS, SRTP) with Public IP addresses

2.6.10.1.1 Session-agents

- Nominal session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BToI_NOMINAL_SA>                ex: BToI_NOMINAL_SA
CSBC(session-agent)# ip-address <BToI_NOMINAL_SA_IP>           ex: 82.82.24.71
CSBC(session-agent)# port 5061
CSBC(session-agent)# transport-method StaticTLS
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-BToI-Peering
CSBC(session-agent)# ping-method OPTIONS
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# done
CSBC(session-agent)# quit
CSBC#
```

- Backup session-agent

The same applies to the backup BToI SA: SIP termination must be configured. Please follow the instructions below.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BToI_BACKUP_SA>                 ex: BToI_BACKUP_SA
CSBC(session-agent)# ip-address <BToI_BACKUP_SA_IP>            ex: 82.82.24.72
CSBC(session-agent)# port 5061
CSBC(session-agent)# transport-method StaticTLS
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-BToI-Peering
CSBC(session-agent)# ping-method OPTIONS
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# done
CSBC(session-agent)# quit
CSBC#
```

2.6.10.1.2 Session-agent groups

One SAG is configured with both nominal and backup session agents operating in hunt mode. Round-robin might be permitted upon explicit request to Orange.

When a call is to be routed to BToI, the nominal BToI SA is selected as the primary destination. In case of no response, or if a SIP 408 or 5xx error is received, the call is rerouted to the backup BToI SA.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-group
CSBC(session-agent-group)# group-name OB-SBC
```

```
CSBC(session-agent-group)# dest (<BToI_NOMINAL_SA> <BToI_BACKUP_SA>)
CSBC(session-agent-group)# strategy <hunt>
<Optional>
CSBC(session-agent-group)# strategy <roundRobin>
</Optional>
CSBC(session-agent-group)# sag-recursion enabled
CSBC(session-agent-group)# stop-sag-recurse 400-407,409-499
CSBC(session-agent-group)# app-protocol SIP
CSBC(session-agent-group)# done
CSBC(session-agent-group)# quit
CSBC#
```

2.6.10.1.3 Access control

For each configured session-agent, an access-control is created specifying as source address the IP address of the session-agent, as destination-address the IP address of the sip-interface associated to the customer. A signaling packet whose source/destination don't match one of the configured access-controls will be discarded at IP level.

- Access control for nominal BToI session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router access-control
CSBC(access-control)# source-address <BToI_NOMINAL_SA_IP>        ex: 82.82.24.71
CSBC(access-control)# destination-address <E-SBC_NOMINAL>        ex: 138.132.169.2
CSBC(access-control)# realm-id OB-BToI-Peering
CSBC(access-control)# application-protocol SIP
CSBC(access-control)# access permit
CSBC(access-control)# trust-level high
CSBC(access-control)# transport-protocol TCP
CSBC(access-control)# done
CSBC(access-control)# quit
CSBC#
```

- Access control for backup BToI session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router access-control
CSBC(access-control)# source-address <BToI_BACKUP_SA_IP>        ex: 82.82.24.72
CSBC(access-control)# destination-address <E-SBC_NOMINAL>       ex: 138.132.169.2
CSBC(access-control)# realm-id OB-BToI-Peering
CSBC(access-control)# application-protocol SIP
CSBC(access-control)# access permit
CSBC(access-control)# trust-level high
CSBC(access-control)# transport-protocol TCP
CSBC(access-control)# done
CSBC(access-control)# quit
CSBC#
```

2.6.10.2 Orange BToI/BTIPoI (SIP/TLS, SRTP) with DNS Type A

2.6.10.2.1 Session-agents

- Nominal session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BToI_NOMINAL_SA_FQDN>    ex: BToI or BTIPoI FQDN
(A)
CSBC(session-agent)# ip-address 0.0.0.0
CSBC(session-agent)# port 5061
CSBC(session-agent)# transport-method StaticTLS
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-<BToI_or_BTIPoI>-Peering
CSBC(session-agent)# ping-method OPTIONS;hops=0
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# out-manipulationid OUT_TO_<BToI_or_BTIPoI>
CSBC(session-agent)# manipulation-string <DefaultSiteDID>
CSBC(session-agent)# done
CSBC(session-agent)# quit
CSBC#
```

- Backup session-agent

The same applies to the backup BToI/BTIPoI SA: SIP termination must be configured. Please follow the instructions below.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BToI_BACKUP_SA_FQDN>    ex: BToI or BTIPoI FQDN
(A)
CSBC(session-agent)# ip-address 0.0.0.0
CSBC(session-agent)# port 5061
CSBC(session-agent)# transport-method StaticTLS
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB-<BToI_or_BTIPoI>-Peering
CSBC(session-agent)# ping-method OPTIONS;hops=0
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# out-manipulationid OUT_TO_<BToI_or_BTIPoI>
CSBC(session-agent)# manipulation-string <DefaultSiteDID>
CSBC(session-agent)# done
CSBC(session-agent)# quit
CSBC#
```

2.6.10.2.2 Session-agent groups

One SAG is configured with both nominal and backup session agents operating in hunt mode. Round-robin might be permitted upon explicit request to Orange.

When a call is to be routed to BToI/BTIPoI, the nominal BToI/BTIPoI SA is selected as the primary destination. In case of no response, or if a SIP 408 or 5xx error is received, the call is rerouted to the backup BToI/BTIPoI SA.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-group
CSBC(session-agent-group)# group-name OB-SBC
CSBC(session-agent-group)# dest (<BToI_NOMINAL_SA_FQDN> <BToI_BACKUP_SA_FQDN>)
CSBC(session-agent-group)# strategy <hunt>
<Optional>
CSBC(session-agent-group)# strategy <roundRobin>
</Optional>
CSBC(session-agent-group)# sag-recursion enabled
CSBC(session-agent-group)# stop-sag-recurse 400-407,409-499
CSBC(session-agent-group)# app-protocol SIP
CSBC(session-agent-group)# done
CSBC(session-agent-group)# quit
CSBC#
```

2.6.10.2.3 Access control

Please refer to § 2.6.10.1.3.

2.6.10.3 Orange BTIPoI (SIP/TLS, SRTP) with DNS Type SRV

2.6.10.3.1 Session-agents

- Nominal session-agent

Configuration

```
CSBC# conf t
CSBC(configure)# session-router session-agent
CSBC(session-agent)# hostname <BTIPoI_NOMINAL_SA_FQDN>      ex: BTIPoI FQDN (SRV)
CSBC(session-agent)# ip-address 0.0.0.0
CSBC(session-agent)# port 0
CSBC(session-agent)# transport-method StaticTLS
CSBC(session-agent)# trust-me enabled
CSBC(session-agent)# realm OB_BTIPoI-Peering
CSBC(session-agent)# ping-method OPTIONS;hops=0
CSBC(session-agent)# ping-interval 300
CSBC(session-agent)# ping-response enabled
CSBC(session-agent)# ping-all-addresses enabled
CSBC(session-agent)# load-balance-dns-query <hunt>
<Optional>
CSBC(session-agent)# load-balance-dns-query <roundRobin>
</Optional>
CSBC(session-agent)# options +trans-timeouts=2
CSBC(session-agent)# out-manipulationid OUT_TO_<BToI_or_BTIPoI>
CSBC(session-agent)# manipulation-string <DefaultSiteDID>
CSBC(session-agent)# done
CSBC(session-agent)# quit
CSBC#
```

2.6.10.3.2 Session-agent groups

No SAG is created when session agent DNS-SRV load balancing is configured.

Instead, the north SIP interface must be updated with the "stop-recurse" parameter to specify for which SIP error codes recursion should be performed within the resolved IP addresses.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-interface
CSBC(sip-interface)# select
<realm-id>: OB_BTIPoI-Peering
1: OB-Peering X.X.X.X:5061

selection: 1
CSBC(sip-interface)# stop-recurse 400-407,409-499
CSBC(sip-interface)# done
CSBC(sip-interface)# quit
CSBC#
```

When a call is routed to BTIPoI, the nominal BTIPoI session agent (configured with a DNS SRV FQDN) provides two IP addresses via SRV resolution. The call is first sent to the primary IP address. If this IP does not respond or returns a SIP 408 or 5xx error, the call is then retried on the secondary IP address according to the SRV record's priority and weight.

2.6.10.3.3 Access control

Please refer to § 2.6.10.1.3.

Note: For each IP address resolved via SRV resolution, an access-control is created specifying as source address the IP address of the session-agent, as destination-address the IP address of the sip-interface associated to the customer.

2.6.11 Provisioning BToI/BTIPoI on a backup E-SBC

Configure the backup E-SBC following the same procedure used for the nominal E-SBC.

2.6.12 Configuration changes to be made on the south side

The aim of this paragraph is to advise adjustments that may be necessary on the south side of the E-SBC to fully implement certain Oracle features required to comply with Orange Business requirements.

For example, pay attention that BTIP/BTalk over Internet offers use SRTP, SBC configuration shall be updated to specific where RTP shall be used by specifying the correct media-sec-policy for UC/IPBX SIP legs.

Note: Depending on customer setup, some additional manipulations or configuration changes might be needed which must be handle under Customer responsibilities.

In case assistance is needed, please refer to your partner or Oracle directly.

2.6.13 SIP header manipulation

For the encrypted BTalk/BTIP SIP trunk architecture, message manipulations are required on outgoing messages toward Orange BTalk/BTIP, and possibly on incoming messages as well.

These manipulations rules are detailed in the chapter

Note: By default, SBC doesn't prompt the unsaved (**) or unapplied (*) changed. To enable it, use CLI command: `prompt-enabled enabled`

Configuration

```
CSBC# prompt-enabled enabled
```

2.6.13.1 SIP Message Manipulation

For encrypted BTalk/BTIP SIP Trunk architecture, it is required to implement some Message Manipulation for the outgoing message toward Orange BTalk/BTIP. Those Manipulations Rules are detailed on the chapter 2.7 SIP rules and manipulations. Please jump to this Chapter directly.

SIP rules and manipulations". Please refer directly to that chapter.

2.6.14 Local policy

2.6.14.1 From north to south

Please refer to § 2.5.9.1 and adapt “source-realm” with “OB_<BToI_or_BTIPoI>-Peering”

2.6.14.2 From south to north

Please refer to § 2.5.9.2 and adapt next-hop “realm” with “OB_<BToI_or_BTIPoI>-Peering”

Note: For configurations utilizing a DNS SRV record, the Service Agent <BTIPoI_NOMINAL_SA_FQDN> must be specified as the next hop in the local policy, replacing the SAG OB-SBC.

2.6.14.3 Saving and activating configuration

Before continuing, save and activate the current configuration.

Configuration

```
**CSBC# save-config
*CSBC# activate-config
CSBC#
```

Note: By default, SBC doesn’t prompt the unsaved (**) or unapplied (*) changed. To enable it, use CLI command: `prompt-enabled enabled`

Configuration

```
CSBC# prompt-enabled enabled
```

2.6.14.4 SIP Message Manipulation

For encrypted BTalk/BTIP SIP Trunk architecture, it is required to implement some Message Manipulation for the outgoing message toward Orange BTalk/BTIP. Those Manipulations Rules are detailed on the chapter 2.7 SIP rules and manipulations. Please jump to this Chapter directly.

2.7 SIP rules and manipulations

2.7.1 Preamble : Manipulation principles on Oracle E-SBC

The following figure summarizes the Call Routing configuration for the BTalk SIP Trunk on the Oracle E-SBC, focusing on the main internal elements that implement SIP manipulations.

2.7.2 SIP message manipulations

Several SIP manipulations (also known as “HMR”) are required to modify SIP headers and the SDP body, control message content, and ensure interoperability with BTalk/BTIP/BToI/BTIPoI services.

These manipulations can be configured on the E-SBC using one of two methods: either manually via the CLI or by importing predefined ".gz" files provided as a complement to this guide. Due to the complexity of SIP manipulation rules and the deep understanding of system syntax they require, we recommend the second option, which is safer and less error-prone. However, both methods remain available.

The HMR files can be downloaded here: To be update later on when available

List of HMRs	Overview
OUT_TO_BT (Parent procedure)	This manipulation is the main procedure for messages sent to the Orange Business BTalk infrastructure on the north side of E-SBC.
OUT_TO_BTIP (Parent procedure)	This manipulation is the main procedure for messages sent to the Orange Business BTIP infrastructure on the north side of E-SBC.
OUT_TO_BToI (Parent procedure)	This manipulation is the main procedure for messages sent to the Orange Business BToI infrastructure on the north side of E-SBC.
OUT_TO_BTIPoI (Parent procedure)	This manipulation is the main procedure for messages sent to the Orange Business BTIPoI infrastructure on the north side of E-SBC.
SIP_Out-CLIR (child procedure)	This manipulation is enforcing Orange’s requirements for anonymous call (CLIR)
SIP_Out-CleanSDP (child procedure)	This manipulation is enforcing Orange’s requirements for SDP
SIP_Out-DelStirShakenHeaders (child procedure)	This manipulation is enforcing Orange’s requirements by removing unwanted Stir Shaken headers
SIP_Out-EnforceAllow (child procedure)	This manipulation is enforcing Orange’s requirements by editing Allow SIP header and by rejecting unsupported Methods with a 405.

SIP_Out-FixHeaders (child procedure)	This manipulation is enforcing Orange's requirements by editing several SIP headers to perfect topology hiding
SIP_Out-FixHeaders-TLS (child procedure)	This manipulation is enforcing Orange's requirements by editing several SIP headers to perfect topology hiding (TLS use-cases, where FQDN is inserted in SIP uri-host).
SIP_Out-ForceAnnexB (child procedure)	This manipulation is enforcing Orange's requirements for G729 codec.
SIP_Out-ModDiversion (child procedure)	This manipulation is enforcing Orange's requirements for Diversion header
SIP_Out-ModTEPayload101 (child procedure)	This manipulation is enforcing Orange's requirements for telephone-event if present in initial SDP.
SIP_Out-Server (child procedure)	This manipulation is enforcing Orange's requirements by editing or adding Server SIP header
SIP_Out-UserAgent (child procedure)	This manipulation is enforcing Orange's requirements by editing or adding User-Agent SIP header
SIP_Out-addPemSupported (child procedure)	This manipulation forces the inclusion of a "P-Early-Media: supported" header in every initial INVITE.
SIP_Out-addTEPayload101 (child procedure)	This manipulation is enforcing Orange's requirements for telephone-event in initial SDP by adding it.
SIP_Out-stripTimers (child procedure)	This manipulation is enforcing Orange's requirements by removing unwanted Session Timers related headers
SIP_Out_Add_ptime_replies (child procedure)	This manipulation is enforcing Orange's requirements by handling ptime in SDP for replies.

2.7.2.1 OUT_TO_BT (Parent)

- Context / Usage:

This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. An ordered sequence of rules allows SIP messages to be modified to comply with Orange's requirements.

- Input:

Almost all SIP Messages if SIP header matches configured header-name.

- Actions performed:

Several actions are performed serially. The action **`sip-manip`** enables the SBC to call the specified SIP Manipulations in `new-value` element, if other conditions match. The order of the different header-rules matters.

- Output / Results:

The outcome is several modifications at different levels within SIP messages. To better understand the results, please refer to each child sip-manipulations invoked.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name OUT_TO_BT
CSBC(sip-manipulation)# description "Parent HMR Outbound for BT"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addPemSupported
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addPemSupported
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ForceAnnexB
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ForceAnnexB
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModDiversion
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModDiversion
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_FixHeaders
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-FixHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_UserAgent
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-UserAgent
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_Server
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out-Server
```

```
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_stripTimers
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-stripTimers
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_EnforceAllow
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-EnforceALLOW
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addptime
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out_Add_ptime_replies
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModCLIR
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CLIR
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_CleanSDP
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CleanSDP
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_DelStirShakenHeaders
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-DelStirShakenHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.2 OUT_TO_BTIP (Parent)

- Context / Usage:

This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. An ordered sequence of rules allows SIP messages to be modified to comply with Orange's requirements.

- Input:

Almost all SIP Messages if SIP header matches configured header-name.

- Actions performed:

Several actions are performed serially. The action **sip-manip** enables the SBC to call the specified SIP Manipulations in `new-value` element, if other conditions match. The order of the different header-rules matters.

- Output / Results:

The outcome is several modifications at different levels within SIP messages. To better understand the results, please refer to each child sip-manipulations invoked.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name OUT_TO_BTIP
CSBC(sip-manipulation)# description "Parent HMR Outbound for BTIP"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addPemSupported
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addPemSupported
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ForceAnnexB
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ForceAnnexB
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModDiversion
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModDiversion
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_FixHeaders
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-FixHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
```

```
CSBC(header-rule)# name Call_UserAgent
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-UserAgent
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_Server
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out-Server
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_stripTimers
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-stripTimers
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_EnforceAllow
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-EnforceALLOW
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addptime
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out_Add_ptime_replies
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModCLIR
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CLIR
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_CleanSDP
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CleanSDP
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_DelStirShakenHeaders
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-DelStirShakenHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.3 OUT_TO_BToI (Parent)

- Context / Usage:

This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. An ordered sequence of rules allows SIP messages to be modified to comply with Orange's requirements.

- Input:

Almost all SIP Messages if SIP header matches configured header-name.

- Actions performed:

Several actions are performed serially. The action **`sip-manip`** enables the SBC to call the specified SIP Manipulations in `new-value` element, if other conditions match. The order of the different header-rules matters.

- Output / Results:

The outcome is several modifications at different levels within SIP messages. To better understand the results, please refer to each child sip-manipulations invoked.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name OUT_TO_BToI
CSBC(sip-manipulation)# description "Parent HMR Outbound for BToI"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addPemSupported
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addPemSupported
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ForceAnnexB
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ForceAnnexB
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModDiversion
CSBC(header-rule)# header-name Diversion
```

```
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModDiversion
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_FixHeaders
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-FixHeaders-TLS
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_UserAgent
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-UserAgent
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_Server
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out-Server
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_stripTimers
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-stripTimers
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_EnforceAllow
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-EnforceALLOW
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addptime
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out_Add_ptime_replies
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModCLIR
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CLIR
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_CleanSDP
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CleanSDP
CSBC(header-rule)# done
```

```
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_DelStirShakenHeaders
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-DelStirShakenHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.4 OUT_TO_BTIPoI (Parent)

- Context / Usage:

This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. An ordered sequence of rules allows SIP messages to be modified to comply with Orange's requirements.

- Input:

Almost all SIP Messages if SIP header matches configured header-name.

- Actions performed:

Several actions are performed serially. The action **`sip-manip`** enables the SBC to call the specified SIP Manipulations in `new-value` element, if other conditions match. The order of the different header-rules matters.

- Output / Results:

The outcome is several modifications at different levels within SIP messages. To better understand the results, please refer to each child sip-manipulations invoked.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name OUT_TO_BTIPoI
CSBC(sip-manipulation)# description "Parent HMR Outbound for BTIPoI"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addPemSupported
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addPemSupported
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-addTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModTEPayload101
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModTEPayload101
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ForceAnnexB
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ForceAnnexB
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModDiversion
CSBC(header-rule)# header-name Diversion
```

```
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-ModDiversion
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_FixHeaders
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-FixHeaders-TLS
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_UserAgent
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-UserAgent
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_Server
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out-Server
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_stripTimers
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-stripTimers
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_EnforceAllow
CSBC(header-rule)# header-name CSeq
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-EnforceALLOW
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_addptime
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# new-value SIP_Out_Add_ptime_replies
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_ModCLIR
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CLIR
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_CleanSDP
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-CleanSDP
CSBC(header-rule)# done
```

```
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Call_DelStirShakenHeaders
CSBC(header-rule)# header-name From
CSBC(header-rule)# action sip-manip
CSBC(header-rule)# new-value SIP_Out-DelStirShakenHeaders
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.5 SIP_Out-CLIR (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation enforces a compliant formatting for anonymous calls. If no CLI is available, SBC will add the default DID.
- Output / Results: Initial anonymous INVITE modified (RFC 3323).

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-CLIR
CSBC(sip-manipulation)# description "Enforce format for Anonymous Call - CLIR"
CSBC(header-rule)# header-rule
CSBC(header-rule)# name HR_CheckPrivacy
CSBC(header-rule)# header-name Privacy
CSBC(header-rule)# action delete
CSBC(header-rule)# comparison-type pattern-rule
CSBC(header-rule)# msg-type request
CSBC(header-rule)# match-value "^.*$"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_CheckFrom
CSBC(header-rule)# header-name From
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_CheckFrom
CSBC(element-rule)# type uri-user
CSBC(element-rule)# action store
CSBC(element-rule)# comparison-type pattern-rule
CSBC(element-rule)# match-value "[Aa]nonymous"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_CheckFromHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action store
CSBC(element-rule)# comparison-type pattern-rule
CSBC(element-rule)# match-value "^[Aa]nonymous\.invalid"
CSBC(element-rule)# done
```

```
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_AddPAI
CSBC(header-rule)# header-name P-Asserted-Identity
CSBC(header-rule)# action add
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# match-value "!$PAI_USER&$HR_CheckFrom.$ER_CheckFrom.$0"
CSBC(header-rule)# new-value "<sip:"+$MANIP_STRING+@+"<C-SBC-FQDN>"+">"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_AddPrivacy
CSBC(header-rule)# header-name Privacy
CSBC(header-rule)# action add
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type request
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# match-value "$PAI_USER&$HR_CheckFrom.$ER_CheckFrom.$0"
CSBC(header-rule)# new-value "id;user"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_AddPrivacyFromOnly
CSBC(header-rule)# header-name Privacy
CSBC(header-rule)# action add
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type request
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# match-value "
$HR_CheckPrivacy.$0&!$PAI_USER&!$HR_CheckFrom.$ER_CheckFrom.$0"
CSBC(header-rule)# new-value "user"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_EnforceFrom
CSBC(header-rule)# header-name From
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_EnforceFromHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# comparison-type boolean
CSBC(element-rule)# match-value
"$HR_CheckFrom.$ER_CheckFrom.$0&!$HR_CheckFrom.$ER_CheckFromHost.$0"
CSBC(element-rule)# new-value "anonymous.invalid"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(header-rule)# done
CSBC(sip-manipulation)# exit
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.6 SIP_Out-CleanSDP (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: All messages with SDP
- Actions performed: This manipulation enforces a compliant formatting for SDP by removing unwanted attributes.
- Output / Results: SDP is cleaned by SBC.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-CleanSDP
CSBC(sip-manipulation)# description "Clean SDP by removing unnecessary
attributes"
CSBC(…)# mime-sdp-rule
CSBC(…)# name MSR_CleanSDP
CSBC(…)# action manipulate
CSBC(…)# sdp-media-rule
CSBC(…)# name SMR_CleanSDP
CSBC(…)# media-type media
CSBC(…)# action manipulate
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_CleanSDP
CSBC(…)# type a
CSBC(…)# action delete
CSBC(…)# comparison-type boolean
CSBC(…)# match-value
!$REGEX("^(crypto|fmtp|ptime|maxptime|rtpmap|sendrecv|sendonly|recvonly|inactive|
T38|sqn|cdsc)(.*)")
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveB
CSBC(…)# type b
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveI
CSBC(…)# type i
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveK
CSBC(…)# type k
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-session-rule
CSBC(…)# name SSR_CleanSDP
CSBC(…)# action manipulate
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveB
CSBC(…)# type b
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveI
CSBC(…)# type i
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# sdp-line-rule
CSBC(…)# name SLR_RemoveK
```

```
CSBC(…)# type k
CSBC(…)# action delete
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# done
CSBC(…)# exit
CSBC(…)# exit
```

2.7.2.7 SIP_Out-DelStirShakenHeaders (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation removes specific headers related to Stir Shaken mechanism.
- Output / Results: Initial INVITE without specified headers in this HMR.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-DelStirShakenHeaders
CSBC(sip-manipulation)# description "Delete unwanted headers APNF Related"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_DelIdentity
CSBC(header-rule)# header-name Identity
CSBC(header-rule)# action delete
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_DelOrigID
CSBC(header-rule)# header-name Origination-ID
CSBC(header-rule)# action delete
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_DelAttestationInfo
CSBC(header-rule)# header-name Attestation-Info
CSBC(header-rule)# action delete
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.8 SIP_Out-EnforceALLOW (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Any Request messages containing an Allow header
- Actions performed: Removes unwanted methods from the Allow header.
- Output / Results: Allow header matches Orange's requirements

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-EnforceALLOW
CSBC(sip-manipulation)# description "Remove and Reject unwanted Methods"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_filterAllow
CSBC(header-rule)# header-name Allow
CSBC(header-rule)# action manipulate
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_filterAllow
CSBC(element-rule)# type header-value
CSBC(element-rule)# action find-replace-all
CSBC(element-rule)# match-val-type any
CSBC(element-rule)# comparison-type pattern-rule
CSBC(element-rule)# match-value
(,\s*[RSPMNrspmn][euproEUPRO][^,]*|[RSPMNrspmn][euproEUPRO][^,]*\s*,\?\s\?)
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(header-rule)# name HR_Reject405
CSBC(header-rule)# header-name Cseq
CSBC(header-rule)# action reject
CSBC(header-rule)# methods (MESSAGE,NOTIFY,PUBLISH,REGISTER,SUBSCRIBE)
CSBC(header-rule)# new-value "405:Method Not Allowed"
CSBC(header-rule)# done
CSBC(header-rule)# exit

CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.9 SIP_Out-FixHeaders (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT or BTIP parent manipulations (all except TLS case).
- Input: Initial INVITE message or replies from C-SBC to INVITE
- Actions performed: This manipulation rewrites several SIP headers (From, To, P-Asserted-Identity, Contact, To, Diversion) to ensure topology is properly hidden (no information about south-side nodes). Also, the max-forward header is resetted to its maximum possible value.
- Output / Results: Topology is hidden in SIP messages from C-SBC to Orange.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-FixHeaders
CSBC(sip-manipulation)# description "Topology hiding on To, From PAI, RPI,
Contact"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixPAI
CSBC(header-rule)# header-name P-Asserted-Identity
CSBC(header-rule)# action manipulate
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixPAIHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_IP
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixRPI
CSBC(header-rule)# header-name Remote-Party-ID
CSBC(header-rule)# action manipulate
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixRPIHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_IP
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixFrom
CSBC(header-rule)# header-name From
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixFromHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_IP
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixFromPort
CSBC(element-rule)# type uri-port
CSBC(element-rule)# action delete-element
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixContact
CSBC(header-rule)# header-name Contact
CSBC(header-rule)# action manipulate
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixContactHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_IP
```

```
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixTo
CSBC(header-rule)# header-name To
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixToHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $REMOTE_IP
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixToPort
CSBC(element-rule)# type uri-port
CSBC(element-rule)# action delete-element
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_ResetMaxForwards
CSBC(header-rule)# header-name Max-Forwards
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# new-value 70
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixDiversion
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# action manipulate
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixDiversionHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_IP
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixDiversionPort
CSBC(element-rule)# type uri-port
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_PORT
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.10 SIP_Out-FixHeaders-TLS (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BToI or BTIPoI parent manipulations (TLS case).
- Input: Initial INVITE message or replies from C-SBC to INVITE
- Actions performed: This manipulation rewrites several SIP headers (From, To, P-Asserted-Identity, Contact, To, Diversion) to ensure topology is properly hidden (no information about south-side nodes). Also, the max-forward header is resetted to its maximum possible value.
- Output / Results: Topology is hidden in SIP messages from C-SBC to Orange.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-FixHeaders-TLS
CSBC(sip-manipulation)# description "Topology hiding on To, From PAI, RPI,
Contact"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixPAI
CSBC(header-rule)# header-name P-Asserted-Identity
CSBC(header-rule)# action manipulate
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixPAIHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<C-SBC-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixRPI
CSBC(header-rule)# header-name Remote-Party-ID
CSBC(header-rule)# action manipulate
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixRPIHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<C-SBC-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixFrom
CSBC(header-rule)# header-name From
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixFromHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<C-SBC-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(element-rule)# element-rule
CSBC(element-rule)# name ER_fixFromPort
```

```
CSBC(element-rule)# type uri-port
CSBC(element-rule)# action delete-element
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixContact
CSBC(header-rule)# header-name Contact
CSBC(header-rule)# action manipulate
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixContactHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<C-SBC-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixTo
CSBC(header-rule)# header-name To
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixToHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<OB_BT*-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixToPort
CSBC(element-rule)# type uri-port
CSBC(element-rule)# action delete-element
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_ResetMaxForwards
CSBC(header-rule)# header-name Max-Forwards
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type request
CSBC(header-rule)# new-value 70
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_fixDiversion
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# action manipulate
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixDiversionHost
CSBC(element-rule)# type uri-host
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value "<C-SBC-FQDN>"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_fixDiversionPort
CSBC(element-rule)# type uri-port
```

```
CSBC(element-rule)# action replace
CSBC(element-rule)# new-value $LOCAL_PORT
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.11 SIP_Out-ForceAnnexB (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message and replies
- Actions performed: This manipulation removes unwanted g729 codec, as the annexb variant enables VAD (Voice Activation Detection) which is not supported.
- Output / Results: If SDP contains g729, annexb will always be set to no.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-ForceAnnexB
CSBC(sip-manipulation)# description "Handle AnnexB if missing"
CSBC(sip-manipulation)# mime-sdp-rule
CSBC(...)# name MSR_AddAnnexBNo
CSBC(...)# msg-type any
CSBC(...)# methods ACK,INVITE,UPDATE
CSBC(...)# action manipulate
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_TestG729
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_MatchG729
CSBC(...)# type m
CSBC(...)# action store
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "\s18\b"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_Delete
CSBC(...)# type a
CSBC(...)# action delete
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^.*annexb.*$"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_AddAnnexB
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "$MSR_AddAnnexBNo.$SMR_TestG729.$SLR_MatchG729[0]"
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_AddAnnexbNo
CSBC(...)# type a
CSBC(...)# action add
CSBC(...)# new-value "fmtp:18 annexb=no"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
```

```
CSBC(...)# done
CSBC(...)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.12 SIP_Out-ModDiversion (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation forces the formatting of Diversion header.
- Output / Results: Initial INVITE containing a well formatted Diversion header.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-ModDiversion
CSBC(sip-manipulation)# description "Enforce Diversion format"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_CheckDiversion
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# action manipulate
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# element-rule
CSBC(header-rule)# name ER_CheckReason
CSBC(element-rule)# parameter-name reason
CSBC(element-rule)# type header-param
CSBC(element-rule)# action delete-element
CSBC(element-rule)# comparison-type boolean
CSBC(element-rule)# match-value !$REGEX("^(unknown|user-busy|no-
answer|unavailable|unconditional|time-of-day|do-not-disturb|deflection|follow-
me|out-of-service|away)")
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_CheckNoReason
CSBC(element-rule)# parameter-name reason
CSBC(element-rule)# type header-param
CSBC(element-rule)# action store
CSBC(element-rule)# comparison-type pattern-rule
CSBC(element-rule)# match-value "^$"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_AddDefaultReason
CSBC(element-rule)# parameter-name reason
CSBC(element-rule)# type header-param
CSBC(element-rule)# action add
CSBC(element-rule)# comparison-type boolean
CSBC(element-rule)# match-value "$HR_CheckDiversion.$ER_CheckNoReason"
CSBC(element-rule)# new-value unknown
CSBC(element-rule)# done
CSBC(element-rule)# exit
```

```
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_CheckCounter
CSBC(element-rule)# parameter-name counter
CSBC(element-rule)# type header-param
CSBC(element-rule)# action store
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# element-rule
CSBC(element-rule)# name ER_AddCounter
CSBC(element-rule)# parameter-name counter
CSBC(element-rule)# type header-param
CSBC(element-rule)# action add
CSBC(element-rule)# comparison-type boolean
CSBC(element-rule)# match-value "!$HR_CheckDiversion.$ER_CheckCounter"
CSBC(element-rule)# new-value 1
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_convertDiversionTeltoSIP
CSBC(header-rule)# header-name Diversion
CSBC(header-rule)# comparison-type pattern-rule
CSBC(header-rule)# match-value ([^<]*)<[\s]*tel:([^>]+)>(.*)
CSBC(header-rule)# new-value $1+"sip:"+$2+"@"+$REMOTE_IP+";user=phone"+$3
CSBC(header-rule)# action manipulate
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.13 SIP_Out-ModTEPayload101 (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message and replies
- Actions performed: This manipulation forces the inclusion of telephone-event attributes within SDP to match Orange's requirements (Payload: 101 and symbols: 0-15).
- Output / Results: SDP with expected values and attributes for DTMF.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-ModTEPayload101
CSBC(sip-manipulation)# description "Enforce Telephone-Event Payload"
CSBC(sip-manipulation)# mime-sdp-rule
CSBC(...)# name MSR_getpayload
CSBC(...)# msg-type any
CSBC(...)# methods ACK,INVITE,UPDATE
CSBC(...)# action manipulate
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_getpayload
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# sdp-line-rule
```

```
CSBC(...)# name SLR_geta
CSBC(...)# type a
CSBC(...)# action replace
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^rtpmap:([0-9]{2,3}) telephone-event/8000$"
CSBC(...)# new-value "rtpmap:101 telephone-event/8000"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_geta1
CSBC(...)# type a
CSBC(...)# action store
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^fmtp:([0-9]{2,3}).*0-([0-9]{1,2})$"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_chgmline
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "$MSR_getpayload.$SMR_getpayload.$SLR_geta[~]"
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_chgfmtp
CSBC(...)# type a
CSBC(...)# action find-replace-all
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value
"^fmtp:({$MSR_getpayload.$SMR_getpayload.$SLR_geta[~].$1}).*[[:1:]]"
CSBC(...)# new-value 101
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_replacemline
CSBC(...)# type m
CSBC(...)# action find-replace-all
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^(audio [0-8]+ RTP.*)(
{$MSR_getpayload.$SMR_getpayload.$SLR_geta[~].$1})(.*$)[[:2:]]"
CSBC(...)# new-value " 101"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_mod_fmtp
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "$MSR_getpayload.$SMR_getpayload.$SLR_geta[~]"
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_addfmtp
CSBC(...)# type a
CSBC(...)# action add
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "!$MSR_getpayload.$SMR_getpayload.$SLR_geta1[~]"
CSBC(...)# new-value fmtp:101+" "+"0-15"
CSBC(...)# done
CSBC(...)# exit
```

```
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_modfmtp
CSBC(...)# type a
CSBC(...)# action replace
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^(fmtp:101 0-)(.*)$"
CSBC(...)# new-value $1+15
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.14 SIP_Out-Server (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Replies messages
- Actions performed: This manipulation edits or enforces value of Server header.
- Output / Results: Replies containing a detailed Server header.

Note: Customer needs to reflect SBC's version in this sip-manipulation (specially in case of upgrades).

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-Server
CSBC(sip-manipulation)# description "Add SBC Version to Server"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_CheckServer
CSBC(header-rule)# header-name Server
CSBC(header-rule)# action store
CSBC(header-rule)# comparison-type pattern-rule
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# match-value "^.+$"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_ModServer
CSBC(header-rule)# header-name Server
CSBC(header-rule)# action manipulate
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# match-value "$HR_CheckServer"
CSBC(header-rule)# new-value $ORIGINAL+" Oracle SBC 9.3.p7"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_AddServer
CSBC(header-rule)# header-name Server
CSBC(header-rule)# action add
CSBC(header-rule)# comparison-type boolean
```

```
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# match-value "!$HR_CheckServer"
CSBC(header-rule)# new-value "Oracle SBC 9.3.p7"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.15 SIP_Out-UserAgent (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation add SIP header User-Agent or edit it to add SBC version and default user-agent value from previous hop.
- Output / Results: Initial INVITE containing a detailed User-Agent header.

Note: Customer needs to reflect SBC's version in this sip-manipulation (specially in case of upgrades).

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-UserAgent
CSBC(sip-manipulation)# description "Add SBC Version to User-Agent"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_CheckUserAgent
CSBC(header-rule)# header-name User-Agent
CSBC(header-rule)# action store
CSBC(header-rule)# comparison-type pattern-rule
CSBC(header-rule)# msg-type request
CSBC(header-rule)# match-value "^.+$"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_ModUserAgent
CSBC(header-rule)# header-name User-Agent
CSBC(header-rule)# action manipulate
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type request
CSBC(header-rule)# match-value "$HR_CheckUserAgent"
CSBC(header-rule)# new-value $ORIGINAL+" Oracle SBC 9.3.p7"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_AddUserAgent
CSBC(header-rule)# header-name User-Agent
CSBC(header-rule)# action add
CSBC(header-rule)# comparison-type boolean
CSBC(header-rule)# msg-type request
CSBC(header-rule)# match-value "!$HR_CheckUserAgent"
CSBC(header-rule)# new-value "Oracle SBC 9.3.p7"
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.16 SIP_Out-addPemSupported (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message

- Actions performed: This manipulation forces the inclusion of a "P-Early-Media" header with value "Supported" in every initial INVITE.
- Output / Results: Initial INVITE containing a "P-Early-Media: Supported" header.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-addPemSupported
CSBC(sip-manipulation)# description "Enforce PEM header for Early-Media"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_addPemSupported
CSBC(header-rule)# header-name P-Early-Media
CSBC(header-rule)# action add
CSBC(header-rule)# msg-type out-of-dialog
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# new-value supported
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.17 SIP_Out-addTEPayload101 (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation forces the inclusion of SDP attributes related to telephone-event if missing, payload is set to 101 and symbols to 0-15.
- Output / Results: Initial INVITE containing SDP with telephone-event

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-addTEPayload101
CSBC(sip-manipulation)# description "Add Telephone-Event if missing"
CSBC(sip-manipulation)# mime-sdp-rule
CSBC(...)# name MSR_getDTMFpayload
CSBC(...)# msg-type any
CSBC(...)# action store
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_isDtmf
CSBC(...)# media-type audio
CSBC(...)# action store
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_geta
CSBC(...)# type a
CSBC(...)# action store
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "rtpmap:[0-9]+ telephone-event/8000"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
```

```
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# mime-sdp-rule
CSBC(...)# name MSR_setDtmf
CSBC(...)# msg-type any
CSBC(...)# action manipulate
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "!$MSR_getDTMFpayload.$SMR_isDtmf.$SLR_geta[~]"
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_setDtmf_alines
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_addrtpmap
CSBC(...)# type a[^]
CSBC(...)# action add
CSBC(...)# new-value "rtpmap:101 telephone-event/8000"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_addfmtp
CSBC(...)# type a[^]
CSBC(...)# action add
CSBC(...)# new-value "fmtp:101 0-15"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-media-rule
CSBC(...)# name SMR_chgmline
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# sdp-line-rule
CSBC(...)# name SLR_add101
CSBC(...)# type m
CSBC(...)# action replace
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^(audio [0-9]+ RTP.*)$"
CSBC(...)# new-value $ORIGINAL+" 101"
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.18 SIP_Out-stripTimers (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Initial INVITE message
- Actions performed: This manipulation removes headers related to Session Timer.
- Output / Results: Initial INVITE edited to remove Min-SE and Session-Expires.

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out-stripTimers
CSBC(sip-manipulation)# description "Remove Min-SE and Session-Expires"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_delMinSE
CSBC(header-rule)# header-name Min-SE
CSBC(header-rule)# action delete
CSBC(header-rule)# msg-type any
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name HR_delExp
CSBC(header-rule)# header-name Session-Expires
CSBC(header-rule)# action delete
CSBC(header-rule)# msg-type any
CSBC(header-rule)# done
CSBC(header-rule)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.2.19 SIP_Out_Add_ptime_replies (Child)

- Context / Usage: This manipulation is applied to all calls originating from the client's ecosystem and destined for the Orange Business core network. It is invoked in all OUT_TO_BT* parent manipulations.
- Input: Replies to INVITE message
- Actions performed: This manipulation forces the inclusion and the value "20" of the SDP parameter "ptime".
- Output / Results: SDP from replies edited to always have ptime:20

Configuration

```
CSBC# conf t
CSBC(configure)# session-router sip-manipulation
CSBC(sip-manipulation)# name SIP_Out_Add_ptime_replies
CSBC(sip-manipulation)# description "Add ptime if missing or fix existing"
CSBC(sip-manipulation)# header-rule
CSBC(header-rule)# name Find_Ptime
CSBC(header-rule)# header-name Content-Type
CSBC(header-rule)# action store
CSBC(header-rule)# msg-type reply
CSBC(header-rule)# methods (INVITE)
CSBC(header-rule)# element-rule
CSBC(element-rule)# name Find_ptime_SDP
CSBC(element-rule)# parameter-name application/sdp
CSBC(element-rule)# type mime
CSBC(element-rule)# action store
CSBC(element-rule)# comparison-type pattern-rule
CSBC(element-rule)# match-value ".*ptime.*"
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(element-rule)# done
CSBC(element-rule)# exit
CSBC(sip-manipulation)# mime-sdp-rule
```

```
CSBC(...)# name FIX_ptime
CSBC(...)# msg-type reply
CSBC(...)# methods INVITE
CSBC(...)# action manipulate
CSBC(...)# sdp-media-rule
CSBC(...)# name FIX_ptime_media
CSBC(...)# media-type audio
CSBC(...)# action manipulate
CSBC(...)# sdp-line-rule
CSBC(...)# name Fix_ptime_media_A_line
CSBC(...)# type a
CSBC(...)# action replace
CSBC(...)# comparison-type pattern-rule
CSBC(...)# match-value "^ptime:[0-9]{2}$"
CSBC(...)# new-value ptime:20
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# sdp-line-rule
CSBC(...)# name Add_ptime_media_A_line
CSBC(...)# type a
CSBC(...)# action add
CSBC(...)# comparison-type boolean
CSBC(...)# match-value "!$Find_Ptime.$Find_ptime_SDP"
CSBC(...)# new-value ptime:20
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(...)# done
CSBC(...)# exit
CSBC(sip-manipulation)# done
CSBC(session-router)# exit
CSBC(conf)# exit
```

2.7.3 Number manipulations

The purpose of this chapter is to describe all number manipulations required for normalizing calling and called party identities to comply with the ITU-T E.164 format (+CCNSN) before routing the call to Orange Business core network.

This means that all calling and called party numbers presented with an international prefix (e.g., 00CCNSN) or a national prefix (e.g., 0NSN) must be converted to the E.164 format.

One way of manipulating numbers could be using Session-Translation, for more information on this feature, please see here.

Another way would be to use HMR (Header Manipulation Rules) to perform required changes. For more information about HMR and sip-manipulation, please see here and check the provided example below.

Note: The +CC prefix represents the country code of the country where the E-SBC is deployed. It is the Customer's responsibility to provide the correct country code (e.g., +33 for France). If the IPBX uses a local dial plan (private numbering plan), the number manipulation must be adjusted accordingly by the Customer.

2.7.4 Outbound manipulations

At the egress, SIP messages already processed by the SBC can be modified to meet the SIP requirements of the upstream device.

Note: At this stage, Oracle did not apply specific changes on south side to fulfill Orange's requirements. Depending on customer setup, some additional manipulations or configuration changes might be needed. In case assistance is needed, please refer to your partner or Oracle directly.

2.7.5 Inbound manipulations (if required)

At the ingress, inbound SIP messages can be modified to permit proper handling by the SBC's routing function.

Note: At this stage, Oracle did not apply specific changes on south side to fulfill Orange's requirements. Depending on customer setup, some additional manipulations or configuration changes might be needed. In case assistance is needed, please refer to your partner or Oracle directly.

Here an example of an HMR that enforces +E.164 format. Here it is set as ingress, assuming the configuration of routing on the SBC is done using +E164 format, but this HMR could be activated on egress as well. Keep in mind that this is an example, there are many ways of achieving this via HMR.

This HMR would later need to be "called" using action "sip-manip" on north-side or south-side.

Example

```
sip-manipulation
        name                                    SIP_In_SetToE164
        description                             Provided the destination number
starts with 0, replace with +33
        split-headers
        join-headers
        header-rule
                name                                    HR_MatchReqUri00
```

```
                header-name                             request-uri
                action                                  manipulate
                comparison-type                         case-sensitive
                msg-type                                any
                methods
                match-value
                new-value
                element-rule
                        name                                    ER_ManipNumber00
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^00(.*$)
                        new-value                               $ORIGINAL-
^"00"+^"+"
                element-rule
                        name                                    ER_ManipNumber0
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^0[^0](.*$)
                        new-value                               $ORIGINAL-
^"0"+^"+33"
        header-rule
                name                                    HR_MatchTo00
                header-name                             to
                action                                  manipulate
                comparison-type                         case-sensitive
                msg-type                                any
                methods
                match-value
                new-value
                element-rule
                        name                                    ER_ManipNumber00
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^00(.*$)
                        new-value                               $ORIGINAL-
^"00"+^"+"
                element-rule
                        name                                    ER_ManipNumber0
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^0[^0](.*$)
                        new-value                               $ORIGINAL-
^"0"+^"+33"
        header-rule
```

```
                name                                    HR_MatchFrom00
                header-name                             From
                action                                  manipulate
                comparison-type                         case-sensitive
                msg-type                                any
                methods
                match-value
                new-value
                element-rule
                        name                                    ER_ManipNumber00
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^00(.*$)
                        new-value                               $ORIGINAL-
^"00"+^"+"
                element-rule
                        name                                    ER_ManipNumber0
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^0[^0](.*$)
                        new-value                               $ORIGINAL-
^"0"+^"+33"
        header-rule
                name                                    HR_MatchPAI00
                header-name                             P-Asserted-Identity
                action                                  manipulate
                comparison-type                         case-sensitive
                msg-type                                any
                methods
                match-value
                new-value
                element-rule
                        name                                    ER_ManipNumber00
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^00(.*$)
                        new-value                               $ORIGINAL-
^"00"+^"+"
                element-rule
                        name                                    ER_ManipNumber0
                        parameter-name
                        type                                    uri-phone-number-
only
                        action                                  replace
                        match-val-type                          any
                        comparison-type                         pattern-rule
                        match-value                             ^0[^0](.*$)
                        new-value                               $ORIGINAL-
^"0"+^"+33"
```

3 Glossary

BTalk: Business Talk

BTIP: Business Talk IP

BToI: Business Talk over Internet

BTIPoI: Business Talk IP over Internet

C-SBC: Customer Session Border Controller = E-SBC

DTMF: Dual Tone Multi Frequency

E-SBC: Enterprise Session Border Controller

FQDN: Fully Qualified Domain Name

IP: Internet Protocol

LAN: Local Area Network

LLDP: Link Layer Discovery Protocol

NET: Network Equipment Technologies

PBX: Private Branch eXchange

PSTN: Public Switched Telephone Network

RS: Remote Site

SBC: Session Border Controller

SIP: Session Initiation Protocol

TCP: Transmission Control Protocol

TLS: Transport Layer Security

UDP: User Datagram Protocol

WAN: Wide Area Network

4 Annexes

4.1 Import HMR's via CLI

To be updated as soon as available.

4.2 Example of SIP INVITE message

4.2.1 From Customer IPBX to Orange BTalk

To be updated as soon as available.

4.2.2 From Orange BTalk to Customer IPBX

To be updated as soon as available.

4.3 NTP server configuration

For NTP Server configuration, Oracle E-SBC supports NTP servers defined via an IP address or via FQDN. Second option requires DNS resolution.

For more details, on how to configure NTP Server, please refer to the dedicated chapter in Oracle E-SBC Configuration Guide.

4.4 Example of TLS Handshakes

To be updated as soon as available.