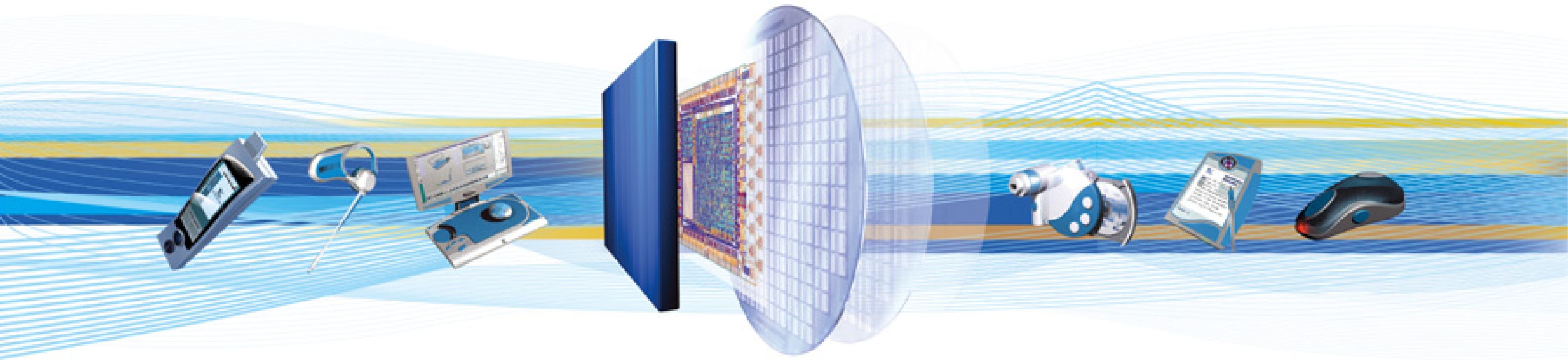
CSR Synergy®

CSR Synergy Bluetooth 18.2.0

SAPS – SIM Access Server Profile

API Description

November 2011



Cambridge Silicon Radio Limited

Churchill House
Cambridge Business Park
Cowley Road
Cambridge   CB4 0WZ
United Kingdom

Registered in England and Wales 3665875

Tel: +44 (0)1223 692000
Fax: +44 (0)1223 692001
www.csr.com

Bluetooth®

Contents

List of Figures

List of Tables

1 Introduction

1.1 Introduction and Scope

This document describes the message interface provided by the SIM Access Profile Server (SAPS). The SAPS conforms to the server side of the SIM Access Profile, ref. [SAPSPEC].

1.2 Assumptions

The following assumptions and preconditions are made in the following:

- There is a secure and reliable transport between the profile, i.e. the SAPS and the application
- There is only one active instance of the SAPS

It is assumed that the user has knowledge of the SIM Access Profile specification.

It is a requirement from [SAPSPEC] that the encryption key size is at least 64 bits. With the current stack it is not possible to ensure that clients connecting to the SAPS have 64 bits encryption.

2 Description

2.1 Introduction

The SIM access profile describes two user scenarios for its use. In the first scenario the client accesses the SIM card in the server, as if the client had direct access. In this scenario the ME-SIM interface is extended over the Bluetooth® link.

In the second scenario the client has access to its own SIM card, but controls an additional SIM card through a Bluetooth® connection to a SAP server.

The SAPS supplies functionality for:

- Establishing a connection with a SIM Access client
- Transfer APDU
- Transfer ATR
- Power SIM Off
- Power SIM On
- Reset SIM
- Card Reader Status
- Set Transfer Protocol
- Status Indication
- Disconnect

The SAPS does not provide functionality for handling security, this should be handled from the application through the Security Controller.

2.2 Reference Model

The SAPS interfaces to the Connection Manager (CM) and to the Service Discovery Server (SDS) through the CM. The application must interface to the SAPS profile and to the Security Controller (SC) in order to handle bonding.

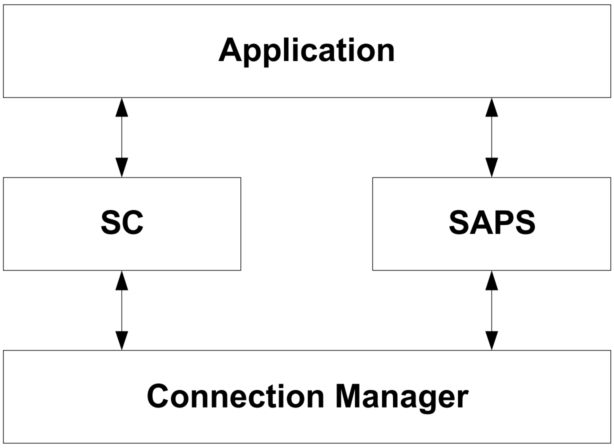


Figure 1: Reference model

2.3 Sequence Overview

When the SAPS is initialised it will be in Idle state. Before any client can connect to the server, the server must be activated by the application. After the server has been activated it is ready to accept a connection from a SIM access client. Once a client has connected, it will access the SIM card in the server through the SIM access messages.

The server can disconnect a client, which means it will return to activated state. It is also possible to deactivate the server in any state, which will bring SAPS back to Idle state.

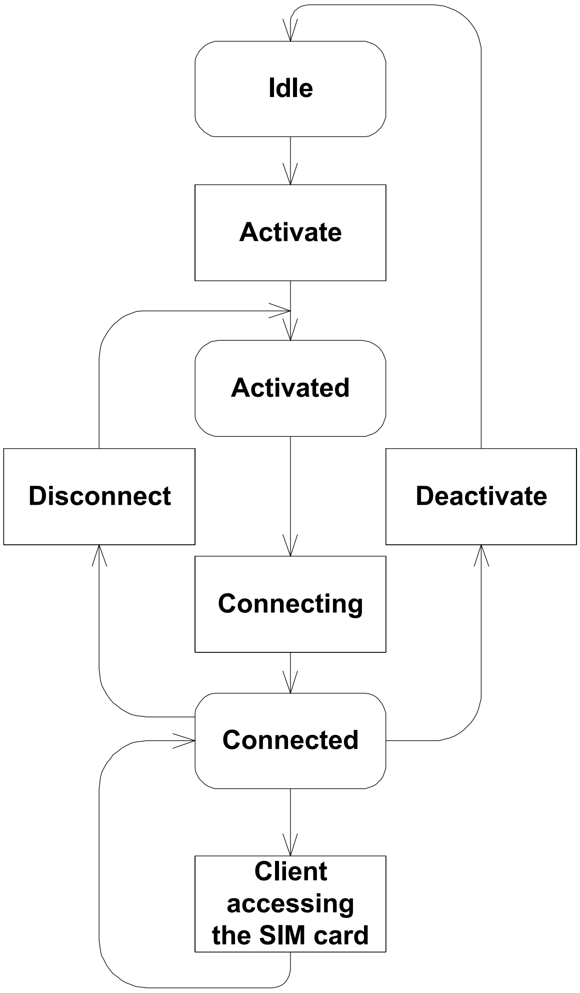


Figure 2: SAPS state diagram

3 Interface Description

In this section a series of MSC will be presented to explain the usage of the available messages in the SAPS. The messages presented in this section will be described further in section 4 where more details of the parameters of each message are available.

3.1 Activation and Deactivation

Activating the SAPS is done with the message CSR_BT_SAPS_ACTIVATE_REQ. This causes the profile to register its service record with SDS and inform the CM that it is ready to accept a connection. When the profile is activated it returns a CSR_BT_SAPS_ACTIVATE_CFM.

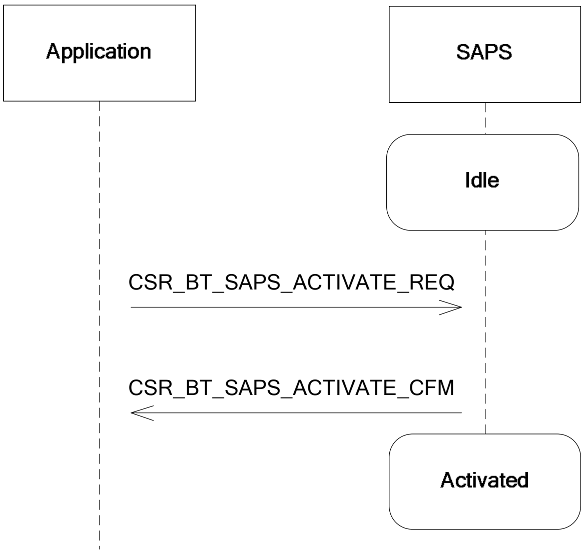


Figure 3: Sequence diagram for activating the SAPS

Deactivating the SAPS is done with the CSR_BT_SAPS_DEACITVATE_REQ message, which is confirmed by the profile with the CSR_BT_SAPS_DEACTIVATE_CFM message. Deactivating the server means the service record is removed from the SDS server.

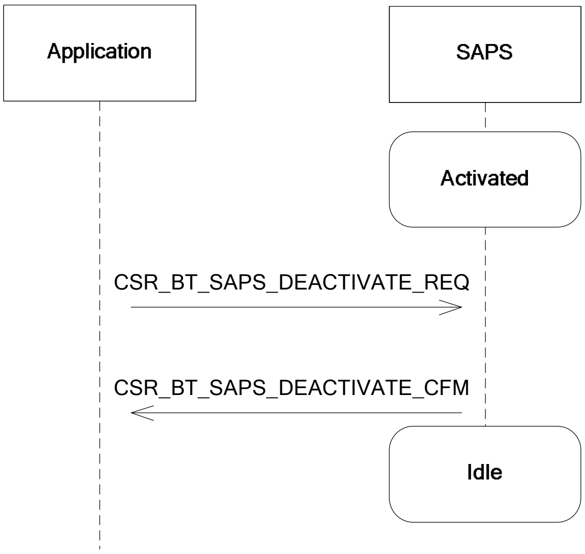


Figure 4: Deactivating the SAPS Connection Establishment

After the SAPS has been activated, the SAP client can connect to the server. The application receives a CSR_BT_SAPS_CONNECT_IND when a client is connecting to the SAPS. The CSR_BT_SAPS_CONNECT_IND message contains the MaxMsgSize parameter, which is the maximum message size that the client can use.

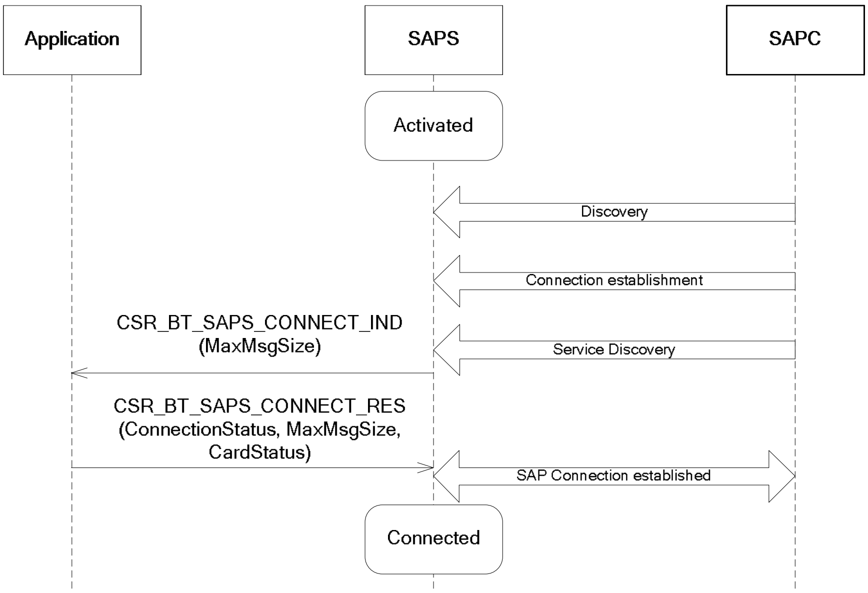


Figure 5: Connection established to SAPS

The application shall respond to the CSR_BT_SAPS_CONNECT_IND, with a CSR_BT_SAPS_CONNECT_RES. If the server accepts the MaxMsgSize parameter received from the client it responds with a ConnectionStatus CSR_BT_OK_CONNECT and returns the MaxMsgSize received from the client. The values used in the ConnectionStatus parameter are defined in csr_bt_sap.h.

If the application regards the MaxMsgSize as too big it should return CSR_BT_MAX_MSG_SIZE_NOT_SUPPORTED in the ConnectionStatus and include a new (and smaller) value in the MaxMsgSize parameter. In this case the client should try to connect again with the new MaxMsgSize parameter.

If the application regards the MaxMsgSize as too small, it sends CSR_BT_MSG_SIZE_TO_SMALL in the ConnectionStatus, which means the connection will not be established.

The CardStatus parameter of the CSR_BT_SAPS_CONNECT_RES message indicates the status of the SIM card on the server. This means the application, after receiving a CSR_BT_SAPS_CONNECT_IND, should get the status of the SIM before responding. After the connection has been established the SAPS will send the CardStatus parameter in a Status indication message to the client as described in ref [SAPSPEC].

3.2 Transfer APDU

When the client wants to transfer command APDU (described in ref [GSM11.11] and ref [GSM11.14]) to the SIM card a CSR_BT_SAPS_TRANSFER_APDU_IND will be received by the application. This message will contain the command APDU, the length of the APDU and a Boolean value that indicates if the APDU is coded according to the ISO 7816-4 specification or if it is coded as in the GSM 11.11.

The application should send the contents of the APDU parameter to the SIM card. On a successful response from the SIM card the application should send the response APDU to the client with the resultCode CSR_BT_RSLT_OK_REQUEST. The values used in the resultCode parameter are defined in csr_bt_sap.h. It is the responsibility of the application to release memory allocated by the profile for the command APDU.

In case of an error while reading the SIM card, only the appropriate resultCode is returned.

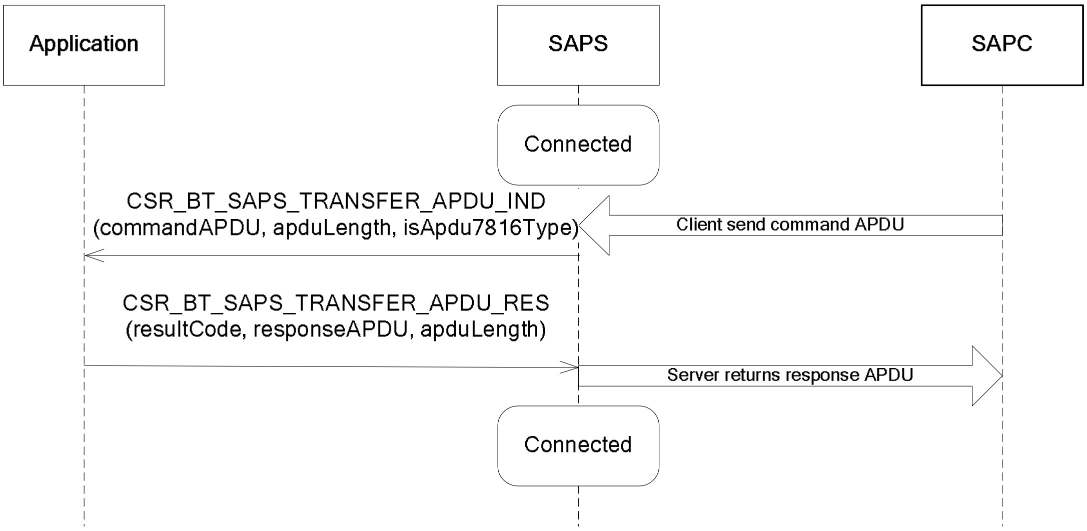

Application
SAPS
SAPC
Connected
CSR_BT_SAPS_TRANSFER_APDU_IND
(commandAPDU, apduLength, isApdu7816Type)
Client send command APDU
CSR_BT_SAPS_TRANSFER_APDU_RES
(resultCode, responseAPDU, apduLength)
Server returns response APDU
Connected

Figure 6: Client sends a command APDU and server responses

3.3 Transfer ATR

If the client requests the ATR from the SIM card, the application will receive a CSR_BT_SAPS_TRANSFER_ATR_IND from the SAPS. The application should request the ATR from the SIM card. In case of a success the ATR is returned in the CSR_BT_SAPS_TRANSFER_ATR_RES message, with CSR_BT_RSLT_OK_REQUEST as the resultCode value.

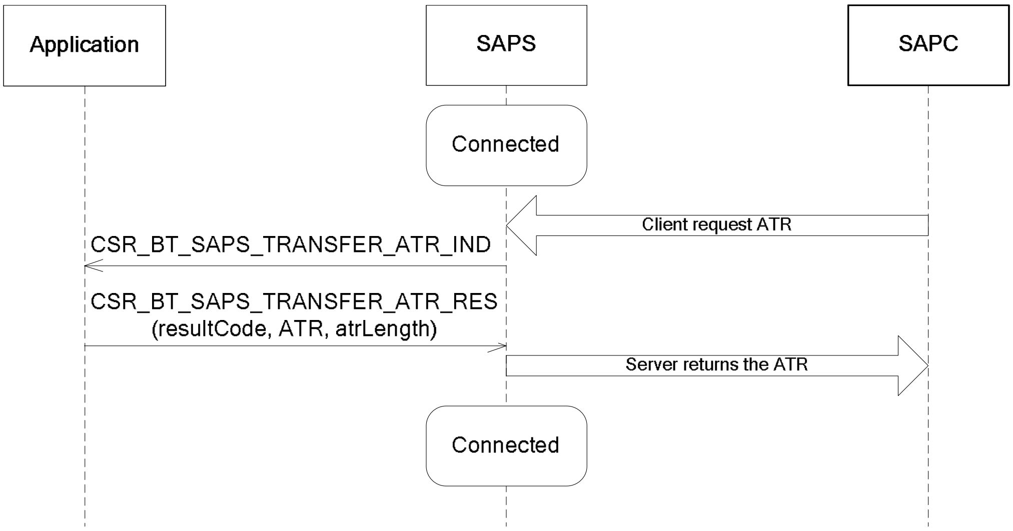

Application
SAPS
SAPC
Connected
Client request ATR
CSR_BT_SAPS_TRANSFER_ATR_IND
CSR_BT_SAPS_TRANSFER_ATR_RES
(resultCode, ATR, atrLength)
Server returns the ATR
Connected

Figure 7: Client requests ATR and server responses

In case an error occurs while obtaining the ATR, only the appropriate resultCode will be returned in the response.

3.4 Power Off Request

The client can request that the SIM access server powers off the SIM card. This is indicated to the application with the CSR_BT_SAPS_POWER_SIM_OFF_IND. The application should power off the SIM card and return CSR_BT_RSLT_OK_REQUEST in the resultCode parameter of the CSR_BT_SAPS_POWER_SIM_OFF_RES.

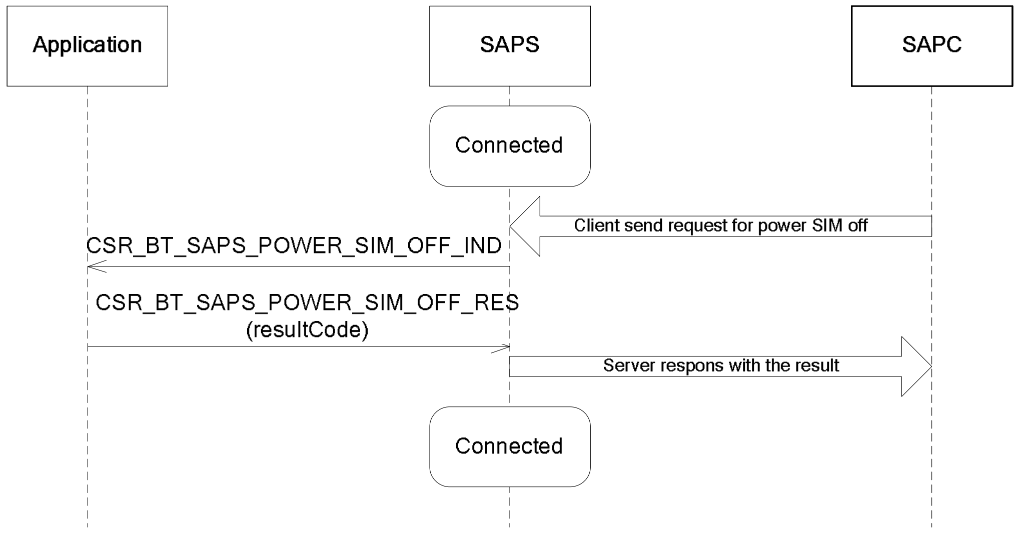

Application
SAPS
SAPC
Connected
Client send request for power SIM off
CSR_BT_SAPS_POWER_SIM_OFF_IND
CSR_BT_SAPS_POWER_SIM_OFF_RES
(resultCode)
Server respons with the result
Connected

Figure 8: Client requests power SIM off

In case an error occurs while the server powers off the SIM card an error code is returned in the resultCode parameter.

3.5 Power On Request

If the client requests that the server powers on the SIM card, the application will receive a CSR_BT_SAPS_POWER_SIM_ON_IND message. The application should power on the SIM and return the resultCode CSR_BT_RSLT_OK_REQUEST in the CSR_BT_SAPS_POWER_SIM_ON_RES message, if it powered the SIM card on successfully.

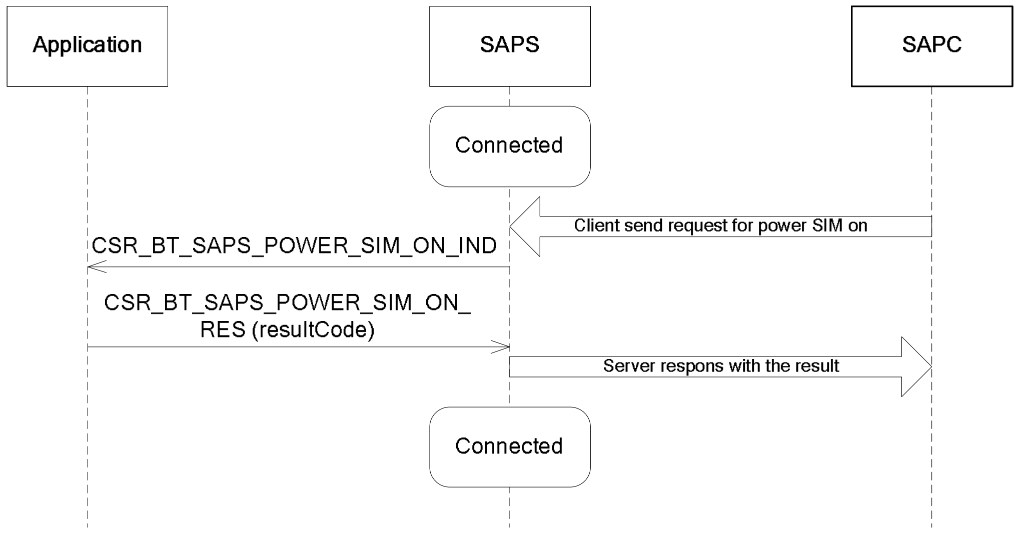

Application
SAPS
SAPC
Connected
Client send request for power SIM on
CSR_BT_SAPS_POWER_SIM_ON_IND
CSR_BT_SAPS_POWER_SIM_ON_
RES (resultCode)
Server respons with the result
Connected

Figure 9: Client requests power SIM on

In case an error occurs while the server powers of the SIM card an error code is returned in the resultCode parameter.

3.6 Reset SIM Request

The application will be notified with a CSR_BT_SAPS_RESET_SIM_IND when the client requests that the server should reset the SIM card. The application should reset the SIM and respond with the CSR_BT_SAPS_RESET_SIM_RES.

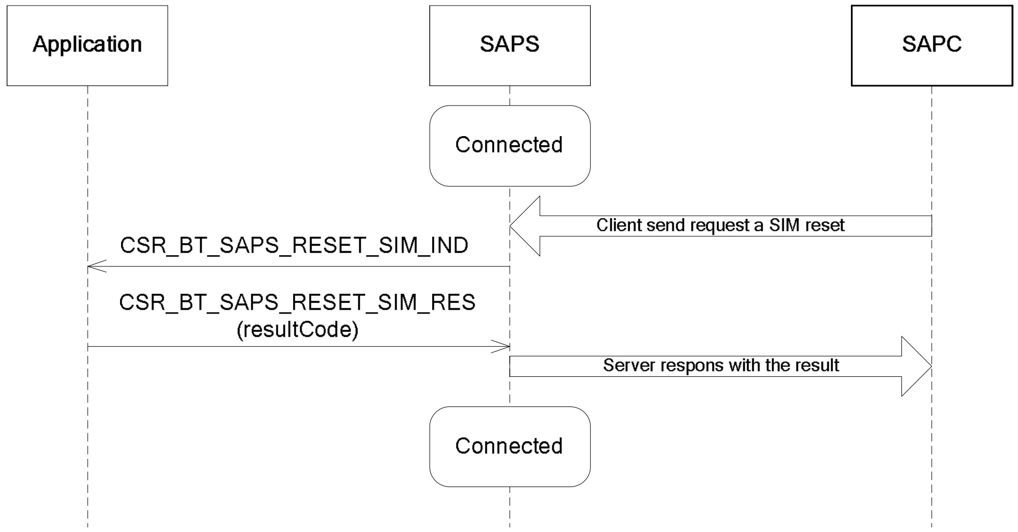

Application
SAPS
SAPC
Connected
Client send request a SIM reset
CSR_BT_SAPS_RESET_SIM_IND
CSR_BT_SAPS_RESET_SIM_RES (resultCode)
Server respons with the result
Connected

Figure 10: Client requests reset SIM card

In case an error occurs while the server powers off the SIM card an error code is returned in the resultCode parameter.

3.7 Transfer Card Reader Status Request

The client can request the status of the SIM card reader. This request from the client is indicated to the application with the CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_IND message.

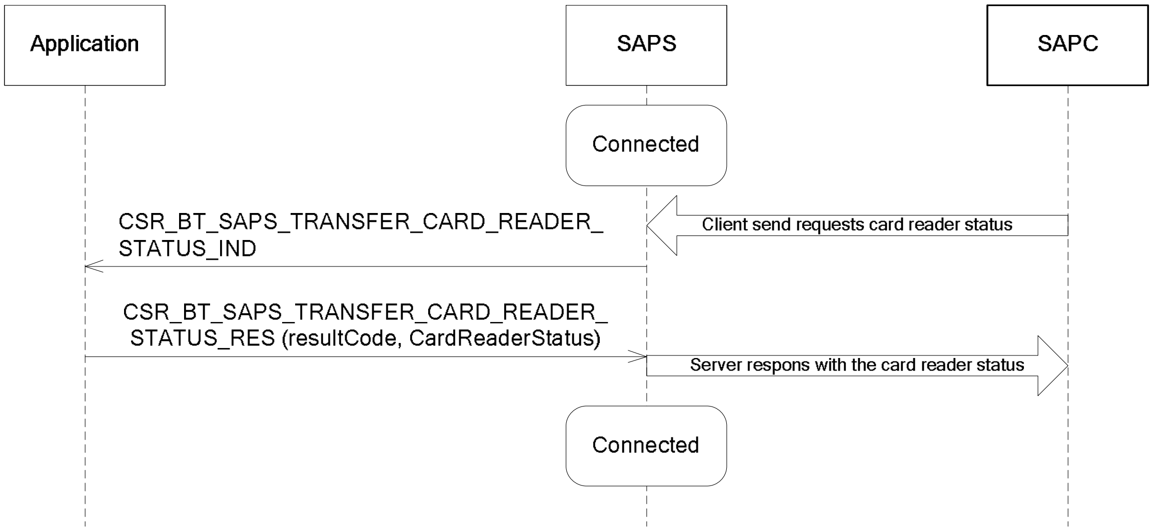

Application
SAPS
SAPC
Connected
Client send requests card reader status
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_IND
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_RES (resultCode, CardReaderStatus)
Server respons with the card reader status
Connected

Figure 11: Client requests the status of the SIM card reader

If the application successfully reads the status of the card reader, it responses with the message CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_RES where the resultCode is CSR_BT_RSLT_OK_REQUEST, and the card reader status is indicated in the CardReaderStatus parameter. The values used in the CardReaderStatus parameter are described in ref [GSM11.14].

3.8 Set Transfer Protocol Request

The CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_IND message indicates to the server application that the connected SAP client wants to change the transfer protocol. If the server supports the requested protocol it shall change to this protocol, reset the SIM card and send the CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES message. After the SIM card has been reset the server shall send a status indication to the client with the result of reset operation.

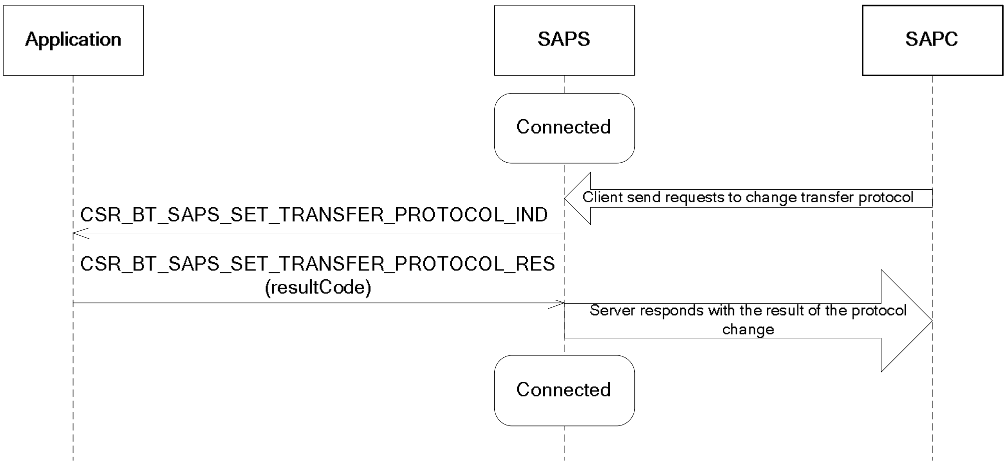

Application
SAPS
SAPC
Connected
Client send requests to change transfer protocol
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_IND
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES
(resultCode)
Server responds with the result of the protocol change
Connected

Figure 12: Client requests to change the transfer protocol.

If the SAP server does not support changing the transfer protocol it shall set the resultCode to CSR_BT_RSLT_ERR_NOT_SUPPORTED in the CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES message.

3.9 Send Status Indication to Client

At any time after a client is connected to the SAPS, the application can send a status indication to the client indicating a change in the availability of the SIM card. The application does this by sending a CSR_BT_SAPS_SEND_STATUS_REQ with StatusChange parameter set to indicate the status of the SIM card. The accepted values of the StatusChange parameter are defined in csr_bt_sap.h.

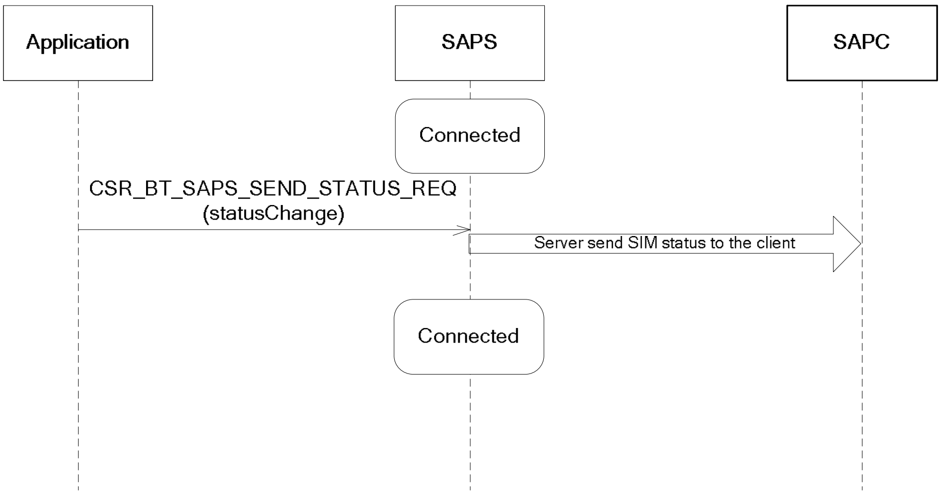

Application
SAPS
SAPC
Connected
CSR_BT_SAPS_SEND_STATUS_REQ
(statusChange)
Server send SIM status to the client
Connected

Figure 13: Application sends a status indication to the client

3.10 Disconnecting

The SIM access server can disconnect the client at any time after a connection has been established. The application disconnects a client by sending the CSR_BT_SAPS_DISCONNECT_REQ to the SAPS. The message for disconnect has the parameter disconnectType that indicates how the server wants to disconnect. The two values of the disconnectType parameter are defined in csr_bt_sap.h.

3.10.1 Disconnect from Server

Graceful Disconnect:

When the application sends a CSR_BT_SAPS_DISCONNECT_REQ with disconnectType CSR_BT_GRACEFUL_DISCONNECT it indicates to the client that the server would like to disconnect. This message does not disconnect the client, but informs the client that it should disconnect. It is left to the client application to handle this request, which means it can continue to transfer APDUs and disconnect later.

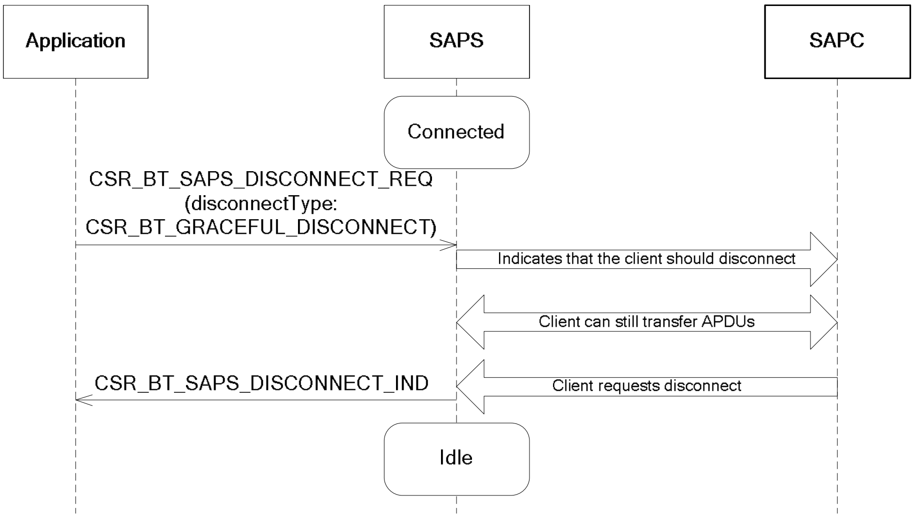

Application
SAPS
SAPC
Connected
CSR_BT_SAPS_DISCONNECT_REQ (disconnectType: CSR_BT_GRACEFUL_DISCONNECT)
Indicates that the client should disconnect
Client can still transfer APDUs
CSR_BT_SAPS_DISCONNECT_IND
Client requests disconnect
Idle

Figure 14: Application sends a Graceful disconnect

Immediate Disconnect:

If the application requests an immediate disconnect it will send a message to the client and disconnect the connection between the client and server.

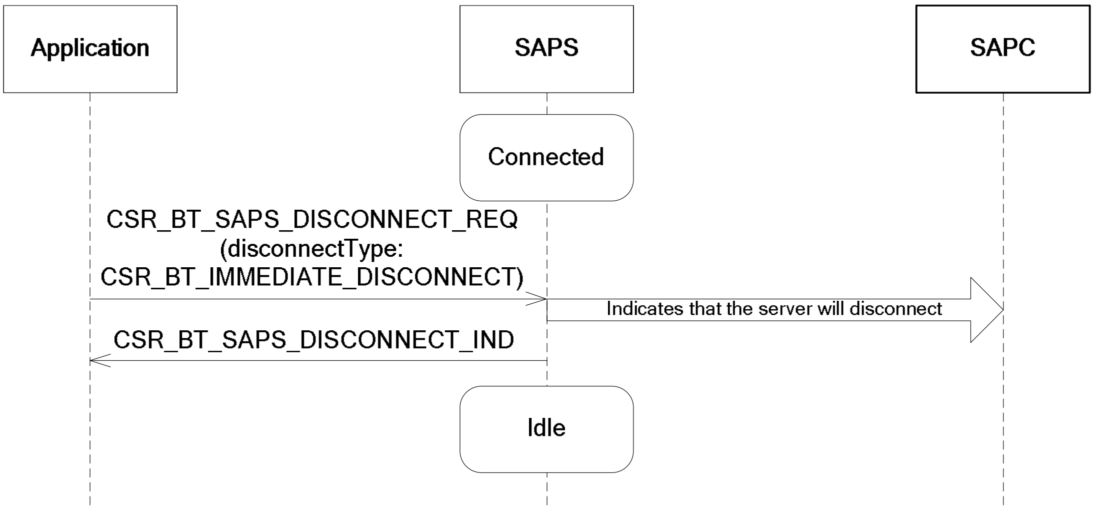

Application
SAPS
SAPC
Connected
CSR_BT_SAPS_DISCONNECT_REQ (disconnectType: CSR_BT_IMMEDIATE_DISCONNECT)
Indicates that the server will disconnect
CSR_BT_SAPS_DISCONNECT_IND
Idle

Figure 15: Application sends an Immediate disconnect

3.10.2 Disconnect from Client

It is also possible that the client application disconnects the connection to the SAPS. This is indicated to the server application with the CSR_BT_SAPS_DISCONNECT_IND message.

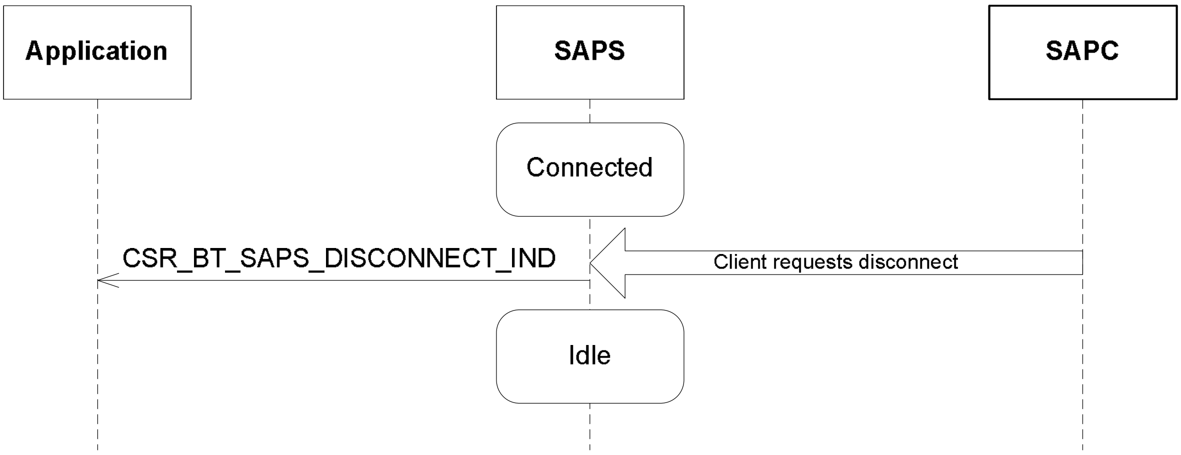


Figure 16: Client disconnects the server

Common for all of the scenarios above is that the SAPS is disconnected from any client when the application receives the CSR_BT_SAPS_DISCONNECT_IND. Once the server is disconnected it will be ready to accept a new connection.

4 SIM Access Profile Server Primitives

This section gives an overview of the primitives and parameters in the interface. Detailed information can be found in the corresponding csr_bt_saps_prim.h file.

4.1 List of All Primitives

Primitives:	Reference:
CSR_BT_SAPS_ACTIVATE_REQ	See section 4.2
CSR_BT_SAPS_ACTIVATE_CFM	See section 4.2
CSR_BT_SAPS_DEACTIVATE_REQ	See section 4.3
CSR_BT_SAPS_DEACTIVATE_CFM	See section 4.3
CSR_BT_SAPS_CONNECT_RES	See section 4.4
CSR_BT_SAPS_CONNECT_IND	See section 4.4
CSR_BT_SAPS_DISCONNECT_REQ	See section 4.5
CSR_BT_SAPS_DISCONNECT_IND	See section 4.5
CSR_BT_SAPS_TRANSFER_ATR_RES	See section 4.6
CSR_BT_SAPS_TRANSFER_ATR_IND	See section 4.6
CSR_BT_SAPS_TRANSFER_APDU_RES	See section 4.7
CSR_BT_SAPS_TRANSFER_APDU_IND	See section 4.7
CSR_BT_SAPS_POWER_SIM_OFF_RES	See section 4.8
CSR_BT_SAPS_POWER_SIM_OFF_IND	See section 4.8
CSR_BT_SAPS_POWER_SIM_ON_RES	See section 4.9
CSR_BT_SAPS_POWER_SIM_ON_IND	See section 4.9
CSR_BT_SAPS_RESET_SIM_RES	See section 4.10
CSR_BT_SAPS_RESET_SIM_IND	See section 4.10
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_RES	See section 4.11
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_IND	See section 4.11
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES	See section 4.12
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_IND	See section 4.12
CSR_BT_SAPS_SEND_STATUS_REQ	See section 4.13
CSR_BT_SAPS_SECURITY_IN_REQ	See section 4.14
CSR_BT_SAPS_SECURITY_IN_CFM	See section 4.14

Table 1: List of all primitives

4.2 CSR_BT_SAPS_ACTIVATE

Parameters / Primitives	type	phandle
CSR_BT_SAPS_ACTIVATE_REQ	✓	✓
CSR_BT_SAPS_ACTIVATE_CFM	✓	

Table 2: CSR_BT_SAPS_ACTIVATE Primitives

Description

This signal activates the SAPS by registering the service record with the SDS, and enables page scan. This will make the SAPS activated until it receives a CSR_BT_SAPS_DEACTIVATE_REQ.

Parameters

type	Signal identity CSR_BT_SAPS_ACTIVATE_REQ/CFM.
phandle	The identity of the calling process. It is possible to initiate the procedure by any higher layer process as the response is returned to phandle.

4.3 CSR_BT_SAPS_DEACTIVATE

Parameters / Primitives	type
CSR_BT_SAPS_DEACTIVATE_REQ	✓
CSR_BT_SAPS_DEACTIVATE_CFM	✓

Table 3: CSR_BT_SAPS_DEACTIVATE Primitives

Description

The CSR_BT_SAPS_DEACTIVATE_REQ signal is used for deactivating the SAPS. This causes the service record to be unregistered from the SDS and page scan will be cancelled. A CSR_BT_SAPS_DEACTIVATE_CFM will be returned to the application when the SAPS is successfully deactivated.

Parameters

type	Signal identity CSR_BT_SAPS_DEACTIVATE_REQ/CFM.

4.4 CSR_BT_SAPS_CONNECT

Parameters / Primitives	type	maxMsgSize	resultCode	cardStatus	deviceAddr	btConnId
CSR_BT_SAPS_CONNECT_RES	✓	✓	✓	✓		
CSR_BT_SAPS_CONNECT_IND	✓	✓			✓	✓

Table 4: CSR_BT_SAPC_CONNECT Primitives

Description

The signal CSR_BT_SAPS_CONNECT_IND indicates to the application that a client wants to connect to the SAPS server. The server application should respond with the CSR_BT_SAPS_CONNECT_RES.

Parameters

type Signal identity CSR_BT_SAPS_CONNECT_RES/IND.

maxMsgSize This is the maximum message size used on the SAP connection. This is sent from the client and the server will respond with the same value if it agrees on the message size.

resultCode The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

cardStatus The parameter holds the card status from the SIM card. After a connection has been established the SAPS will send a status indication, with the contents of this parameter.

deviceAddr Bluetooth address of the remote device that is connecting.

btConnId Identifier used when moving the connection to another AMP controller, i.e. when calling the `CsrBtAmpmMoveReqSend`-function.

4.5 CSR_BT_SAPS_DISCONNECT

Parameters Primitives	type	disconnectType	reasonCode	reasonSupplier
CSR_BT_SAPS_DISCONNECT_REQ	✓	✓		
CSR_BT_SAPS_DISCONNECT_IND	✓		✓	✓

Table 5: CSR_BT_SAPS_DISCONNECT Primitives

Description

The CSR_BT_SAPS_DISCONNECT_REQ signal is used for disconnecting the client from the server. The server application will receive a CSR_BT_SAPS_DISCONNECT_IND once the connection is closed. The CSR_BT_SAPS_DISCONNECT_IND will also be sent to the application if the client disconnects.

Parameters

type — Signal identity CSR_BT_SAPS_DISCONNECT_REQ/IND.

disconnectType — The type of disconnect that the server application wants. Can be either Graceful or Immediate. The possible values for this parameter are defined in csr_bt_sap_common.h.

reasonCode — The reason code of the operation. Possible values depend on the value of reasonSupplier. If e.g. the reasonSupplier == CSR_BT_SUPPLIER_CM then the possible reason codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h files are regarded as reserved and the application should consider them as errors.

reasonSupplier — This parameter specifies the supplier of the reason given in reasonCode. Possible values can be found in csr_bt_result.h

4.6 CSR_BT_SAPS_TRANSFER_ATR

Primitives \ Parameters	type	resultCode	*atrResponse	atrResponseLength
CSR_BT_SAPS_TRANSFER_ATR_RES	✓	✓	✓	✓
CSR_BT_SAPS_TRANSFER_ATR_IND	✓			

Table 6: CSR_BT_SAPS_TRANSFER_ATR Primitives

Description

This message indicates to the server that the client would like to have the ATR from the SIM card.

Parameters

type Signal identity CSR_BT_SAPS_TRANSFER_ATR_RES/IND.

resultCode The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

*atrResponse The ATR from the SIM card.

atrResponseLength The length of the ATR.

4.7 CSR_BT_SAPS_TRANSFER_APDU

Parameters / Primitives	type	resultCode	apduCommand	* apduResponse	apduResponseLength	isApdu7816Type
CSR_BT_SAPS_TRANSFER_APDU_RES	✓	✓		✓	✓	
CSR_BT_SAPS_TRANSFER_APDU_IND	✓		✓		✓	✓

Table 7: CSR_BT_SAPS_TRANSFER_APDU Primitives

Description

The server application will receive a command APDU from the client in the CSR_BT_SAPS_TRANSFER_APDU_IND and responds with the CSR_BT_SAPS_TRANSFER_APDU_RES.

Parameters

type Signal identity CSR_BT_SAPS_TRANSFER_APDU_RES/IND.

resultCode The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

*apduCommand The command APDU from the client. The memory pointed to by this parameter should be released by the application.

*apduResponse The response APDU from the SIM card.

apduResponseLength The length of the APDU.

isApdu7816Type Indicate the type of APDU. (TRUE or FALSE)

4.8 CSR_BT_SAPS_POWER_SIM_OFF

Parameters / Primitives	type	resultCode
CSR_BT_SAPS_POWER_SIM_OFF_RES	✓	✓
CSR_BT_SAPS_POWER_SIM_OFF_IND	✓	

Table 8: CSR_BT_SAPS_POWER_SIM_OFF Primitives

Description

This is the client's request to power off the SIM card.

Parameters

type	Signal identity CSR_BT_SAPS_POWER_SIM_OFF_RES/IND.
resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

4.9 CSR_BT_SAPS_POWER_SIM_ON

Parameters Primitives	type	resultCode
CSR_BT_SAPS_POWER_SIM_ON_RES	✓	✓
CSR_BT_SAPS_POWER_SIM_ON_IND	✓	

Table 9: CSR_BT_SAPS_POWER_SIM_ON Primitives

Description

This is the client's request to power on the SIM card.

Parameters

type	Signal identity CSR_BT_SAPS_POWER_SIM_ON_RES/IND
resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

4.10 CSR_BT_SAPS_RESET_SIM

Parameters **Primitives**	type	resultCode
CSR_BT_SAPS_RESET_SIM_RES	✓	✓
CSR_BT_SAPS_RESET_SIM_IND	✓	

Table 10: CSR_BT_SAPS_RESET_SIM Primitives

Description

This is the client’s request to reset the SIM card.

Parameters

type — Signal identity CSR_BT_SAPS_RESET_SIM_RES/IND.

resultCode — The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.

4.11 CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS

Parameters / Primitives	type	resultCode	cardReaderStatus
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_RES	✓	✓	✓
CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_IND	✓		

Table 11: CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS Primitives

Description

This is the client's request to obtain the status of the card reader

Parameters

type	Signal identity CSR_BT_SAPS_TRANSFER_CARD_READER_STATUS_RES/IND
resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
cardReaderStatus	The status of the card reader. Values for this parameter are defined in ref. [GSM11.14].

4.12 CSR_BT_SAPS_SET_TRANSFER_PROTOCOL

Parameters Primitives	type	resultCode	protocol
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES	✓	✓	
CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_IND	✓		✓

Table 12: CSR_BT_SAPS_SET_TRANSFER_PROTOCOL Primitives

Description

This is the clients request to change the transfer protocol.

Parameters

type	Signal identity CSR_BT_SAPS_SET_TRANSFER_PROTOCOL_RES/IND
resultCode	The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the application should consider them as errors.
protocol	The protocol the client wants to change to. Can be T=0 or T=1.

4.13 CSR_BT_SAPS_SEND_STATUS

Parameters Primitives	type	statusChange
CSR_BT_SAPS_SEND_STATUS_REQ	✓	✓

Table 13: CSR_BT_SAPS_SEND_STATUS Primitives

Description

This signal is send from the SAPS server application to inform the client about the status of the SIM card.

Parameters

type Signal identity CSR_BT_SAPS_SEND_STATUS_REQ

statusChange The status of the SIM card. The possible values for this parameter are defined in csr_bt_sap_common.h.

4.14 CSR_BT_SAPS_SECURITY_IN

Parameters Primitives	type	appHandle	secLevel	resultCode	resultSupplier
CSR_BT_SAPS_SECURITY_IN_REQ	✓	✓	✓		
CSR_BT_SAPS_SECURITY_IN_CFM	✓			✓	✓

Table 14: CSR_BT_SAPS_SECURITY_IN Primitives

Description

Applications that wish to change the enforcement to a specific profile security level, i.e. authentication, encryption and/or authorisation, can use this API to set up the security level for *new* connections. Note that this API is for the local device only and can be used from within any state.

The *CSR_BT_SECURITY_IN_REQ* signal sets up the security level for new incoming connections. Already established or pending connections are not altered.

Note, that any attempts to set security to a less secure level than the mandatory security level will be rejected. See csr_bt_profiles.h for mandatory security settings. The default settings used by CSR Synergy Bluetooth are set to require authentication and encryption.

Note that if MITM protection is requested and the remote device does not have the required IO capabilities, pairing/bonding will fail and connections to the remote device *cannot* be made. See [SC] for further details.

Parameters

type Signal identity CSR_BT_SAPS_SECURITY_IN_REQ/CFM.

appHandle Application handle to which the confirm message is sent.

secLevel The application must specify one of the following values:

- CSR_BT_SEC_DEFAULT : Use default security settings
- CSR_BT_SEC_MANDATORY : Use mandatory security settings
- CSR_BT_SEC_SPECIFY : Specify new security settings

If CSR_BT_SEC_SPECIFY is set the following values can be OR'ed additionally:

- CSR_BT_SEC_AUTHORISATION: Require authorisation
- CSR_BT_SEC_AUTHENTICATION: Require authentication
- CSR_BT_SEC_ SEC_ENCRYPTION: Require encryption (implies authentication)
- CSR_BT_SEC_MITM: Require MITM protection (implies encryption)

resultCode The result code of the operation. Possible values depend on the value of resultSupplier. If e.g. the resultSupplier == CSR_BT_SUPPLIER_CM then the possible result codes can be found in csr_bt_cm_prim.h. All values which are currently not specified in the respective prim.h file are regarded as reserved and the

	application should consider them as errors.
resultSupplier	This parameter specifies the supplier of the result given in resultCode. Possible values can be found in csr_bt_result.h

5 Document References

Document	Reference
SIM Access Profile Version: 1.1 Date: 18-12-2008	[SAPSPEC]
GSM 11.11 Specification of the Subscriber Identity Module – Mobile Equipment (SIM-ME) Interface	[GSM11.11]
GSM 11.14 Specification of the SIM Application Toolkit for the Subscriber Identity Module – Mobile Equipment (SIM – ME) Interface	[GSM11.14]
CSR Profile Layer, Security Controller Interface Description.	[SC]

Terms and Definitions

APDU	Application Protocol Unit
ATR	Answer To Reset
BlueCore®	Group term for CSR’s range of Bluetooth wireless technology chips
Bluetooth®	Set of technologies providing audio and data transfer over short-range radio connections
CSR	Cambridge Silicon Radio
SAPC	SIM Access Profile Client
SAPS	SIM Access Profile Server
SIM	Subscriber Identity Module
UniFi™	Group term for CSR’s range of chips designed to meet IEEE 802.11 standards

Document History

Revision	Date	History
1	26 SEP 11	Ready for release 18.2.0

TradeMarks, Patents and Licences

Unless otherwise stated, words and logos marked with ™ or ® are trademarks registered or owned by CSR plc or its affiliates. Bluetooth® and the Bluetooth logos are trademarks owned by Bluetooth SIG, Inc. and licensed to CSR. Other products, services and names used in this document may have been trademarked by their respective owners.

The publication of this information does not imply that any licence is granted under any patent or other rights owned by CSR plc.

CSR reserves the right to make technical changes to its products as part of its development programme.

While every care has been taken to ensure the accuracy of the contents of this document, CSR cannot accept responsibility for any errors.

Life Support Policy and Use in Safety-critical Compliance

CSR's products are not authorised for use in life-support or safety-critical applications. Use in such applications is done at the sole discretion of the customer. CSR will not warrant the use of its devices in such applications.

Performance and Conformance

Refer to www.csrsupport.com for compliance and conformance to standards information.