Detailed Specification

SNDCP

Department:	Aalborg Wireless Center
Creation Date:	4 September, 2002
Last Modified:	25 October, 2002 by ABR
ID and Version:	8441.706.99.003
Status:	Accepted

0 Document Control

0.1 Document History

ID	Author	Date	Status
8441.706.99.001	ABR	4 September, 2002	Planned
8441.706.99.002	ABR	23 September, 2002	Being Processed
8441.706.99.003	ABR	25 October, 2002	Accepted

0.2 References, Abbreviations, Terms

[TI 7010.801] 7010.801, References and Vocabulary, Texas Instruments

Table of Contents

1 Introduction

The document is based on existing MSC document for SNDCP and existing SDL documentation for SNDCP.

This document describes the introduction of DTI 2 (Data Transmission Interface 2) for SNDCP. Currently SNDCP supports DTI 1, but must be upgraded to DTI 2.

The Purpose of this DTS (Detailed Specification) is to describe the affected states in SNDCP, include description of new states in SNDCP if needed, and include MSC diagrams for the new DTI 2 signalling.

The document will not give any specific description of SNDCP other than the parts affected by the DTI 2.

2 Architecture

The context of SNDCP is presented in chapter 2.1 below. The architecture of SNDCP shall not be described in deeper detail in this document.

2.1 Context

The context diagram for SNDCP is shown in Figure 1 below.

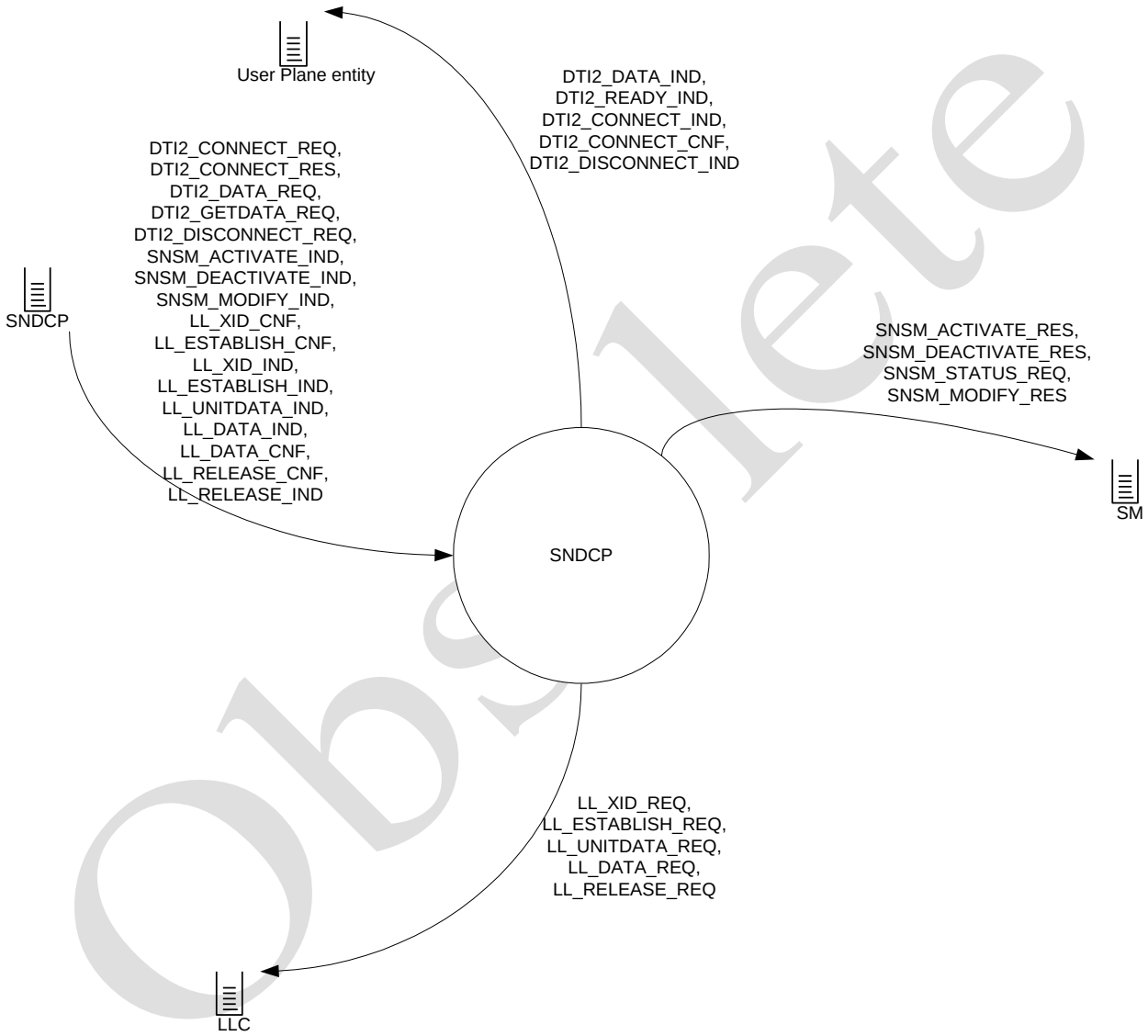

Figure 1 Context diagram for SNDCP

2.1.1 Interface between SNDCP and control plane entities

The interface to SM is used for context activation and deactivation. If modification of the context is needed it is the SM interface that is used. In case any errors are detected then the control plane is informed via the SM interface.

The LLC interface handles XID negotiation, acknowledged establishment, release, uplink-segmented data and downlink-segmented data, which may need reassembly.

2.1.2 Interface between SNDCP and user plane entities

The interface between SNDCP and the user plane is based on DTI 2. The DTI 2 interface is new to SNDCP and is introduced in this document. The new interface will not affect the current state machines in SNDCP, but only introduce new mails (primitives) and logic in some of SNDCPs modules. The introduction of DTI 2 has not caused the removal of existing user plane interface. The existing user plane interface is not described here.

3 Procedures and state machines

3.1 Impacted modules, their states and flag variables in SNDCP

3.1.1 General modules

3.1.1.1 Manager (MG) module

The state machine for the MG module is static, meaning there is only one state.

MG_DEFAULT

The state is static. There is no signalling which has any influence on this state machine.

Instead of a dedicated state machine the manager module uses two global flag registers to keep track of what the NSAPI and SAPI modules are doing. To handle the multiple number of NSAPIs and SAPIs the flag registers are stored in arrays.

The flag themselves are common for both flag registers. However the register for the NSAPI only uses some of the flags and the register for the SAPI another part of the flags.

The flag registers are:

nsapi_state_ra[nsapi], where nsapi is a value between 0-15.

sapi_state_ra[sapi_index], where sapi_index is a value between 0-3.

The flags (states) are described below:

MG_IDLE (0x0000)

No comment needed. The flag applies for both sapi_state_ra and nsapi_state_ra.

MG_EST (0x0001)

At this point an establish request is pending on a SAPI. The given SAPI is waiting for LL_ESTABLISH_CNF. This flag is used for sapi_state_ra only.

MG_XID (0x0002)

At this point an XID request is pending on a SAPI. The given SAPI is waiting for LL_XID_CNF. This flag is used for sapi_state_ra only.

MG_ACT (0x0004)

When this flag is upon reception of for example ll_xid_cnf then an SNSM_ACTIVATE_RES must be sent. This flag applies to nsapi_state_ra only.

MG_REL (0x0008)

This flag indicates that a release request has been sent for LLC and that SNDCP on a given SAPI is waiting for LL_RELEASE_CNF. This flag applies for sapi_state_ra.

MG_DEACT (0x0010)

When this flag is set then a deactivation has been in progress and a SNSM_DEACTIVATE_RES must be sent when internal clean up has been performed or when proper release procedure has been completed.

MG_SEQ (0x0020)

Flag indicating that SNSM_SEQUENCE_IND primitive is awaited. The flag only applies for nsapi_state_ra.

MG_REL_NEC_LOC (0x0040)

A local LL_RELEASE_REQ must be sent after the end of the running sub procedure. The flag is only valid

for sapi_state_ra.

MG_XID_NEC (0x0080)

An LL_XID_REQ must be sent after the end of the running sub procedure. The flag is only valid for sapi_state_ra.

MG_EST_NEC (0x0100)

An LL_ESTABLISH_REQ must be sent after the end of the running sub procedure. The flag is currently not used.

MG_REL_NEC_PEER (0x0200)

A non-local LL_RELEASE_REQ must be sent after the end of the running sub procedure. The flag is only valid for sapi_state_ra

3.1.2 Uplink modules

3.1.2.1 NSAPI Uplink (NU) module

The NU_ are real states for the NSAPI Uplink module. The state machine is used to reflect the connection state to the SAPI module that the NSAPI module is connected to.

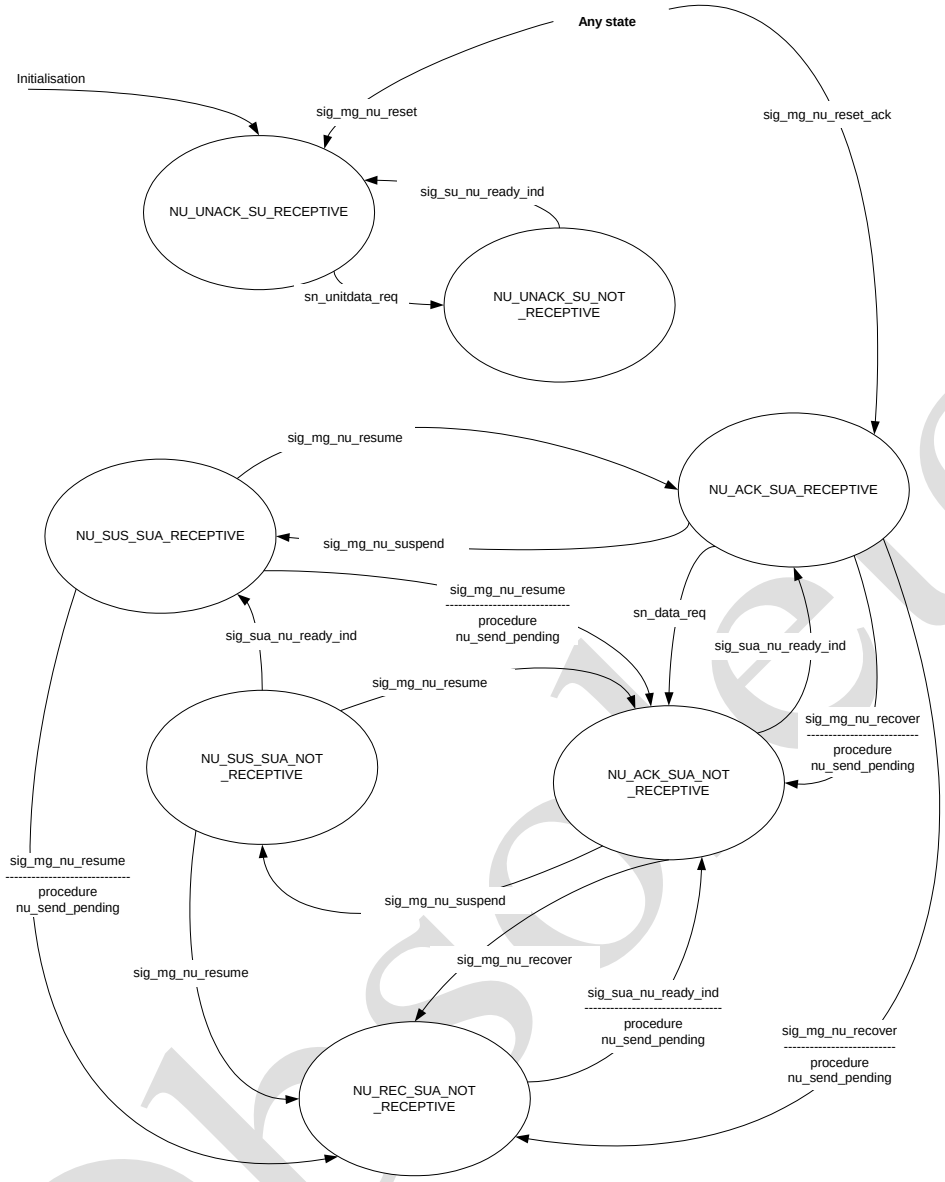

Figure 2 NU module FSM

Below the states are described for the NU-module:

NU_UNACK_SU_RECEPTIVE

This state is for unacknowledged mode of the NSAPI. The NSAPI is connected to a SAPI Uplink module. The SU module is receptive which here means that the SU module can receive an uplink data primitive from the NU module.

The state can be entered at initialization of the NU-module, when *sig_mg_nu_reset* or *sig_su_nu_ready_ind* is received in the NU-module.

NU_UNACK_SU_NOT_RECEPTIVE

This state is for unacknowledged mode of the NSAPI. The NSAPI is connected to a SAPI Uplink module. The SU module is NOT RECEPTIVE, which in this case means that no uplink data can be sent for the

module.

The state can be entered when receiving primitive *sn_unitdata_req* in the NU module.

NU_ACK_SUA_RECEPTIVE

This state is for acknowledged mode of the NSAPI. The NSAPI is connected to one of the SAPI Uplink Acknowledged (SUA) modules. The receptive part means that the SUA module is ready for reception of acknowledged data primitives from the NU module.

The state can be entered when *sig_mg_nu_resume*, *sig_mg_nu_reset_ack* or *sig_sua_nu_ready_ind* is received.

NU_ACK_SUA_NOT_RECEPTIVE

This state is for acknowledged mode of the NSAPI. The NSAPI is connected to one of the SAPI Uplink Acknowledged (SUA) modules. The SUA module has not yet indicated that it is ready for reception of new data primitives thus the NOT RECEPTIVE part of the state.

The state can be entered when receiving primitive *sn_data_req* or signal *sig_mg_nu_resume* or when resending of data primitive is required.

NU_SUS_SUA_RECEPTIVE

This state is for acknowledged mode of the NSAPI. The NSAPI is connected to one of the SAPI Uplink Acknowledged (SUA) module. The SUA module is suspended when the NU module is in this state and cannot process any data primitives. However if any data primitives need to be sent for SUA, then it is allowed to send a data primitive in this state.

The state can be entered when signals *sig_mg_nu_suspend* or *sig_sua_nu_ready_ind* is received.

NU_SUS_SUA_NOT_RECEPTIVE

This state is for acknowledged mode of the NSAPI. The NSAPI is connected to one of the SAPI Uplink Acknowledged (SUA) modules. The SUA module is suspended when the NU module is in this state and cannot process any data primitives. Furthermore when in this state the NU is not allowed to send any data primitives to the SUA module.

The state can be entered when signal *sig_mg_nu_suspend* is received.

NU_REC_SUA_NOT_RECEPTIVE

This state is for acknowledged mode of the NSAPI. The NSAPI is connected to one of the SAPI Uplink Acknowledged (SUA) modules. The SUA module is in recovery mode and cannot receive any data primitives from NU module.

The state can be entered when signal *sig_mg_nu_recover* or *sig_mg_nu_resume*. It can also be entered when resending acknowledged data primitives.

There are furthermore a number of states in the NU module, but they are used for testing purposes and shall not be described in this document. A listing of the states can be seen below.

NU_DISCARD_UNACK_SU_RECEPTIVE

NU_DISCARD_UNACK_SU_NOT_RECEPTIVE

NU_DISCARD_ACK_SUA_RECEPTIVE

NU_DISCARD_ACK_SUA_NOT_RECEPTIVE

NU_DISCARD_SUS_SUA_RECEPTIVE

NU_DISCARD_SUS_SUA_NOT_RECEPTIVE

NU_DISCARD_REC_SUA_NOT_RECEPTIVE

NU_SEND_UNACK_SU_RECEPTIVE

NU_SEND_UNACK_SU_NOT_RECEPTIVE

NU_SEND_ACK_SUA_RECEPTIVE

NU_SEND_ACK_SUA_NOT_RECEPTIVE

NU_SEND_SUS_SUA_RECEPTIVE

NU_SEND_SUS_SUA_NOT_RECEPTIVE

NU_SEND_REC_SUA_NOT_RECEPTIVE

The NU module also contains a flag indicating the current connection state of the DTI link to the upper layer seen from SNDCP point of view.

The flag is:

connection_is_opened

- When TRUE the DTI connection is opened.
- When FALSE the DTI connection is not opened.

This flag combined with the MG module flags will make SNDCP able to determine, for a given nsapi, when a response for a DTI connect request/indication is expected.

3.1.2.2 SAPI Uplink unacknowledged (SU) module

This state machine reflects the connection state between the SU module and the connected LLC SAPI for unacknowledged mode of operation.

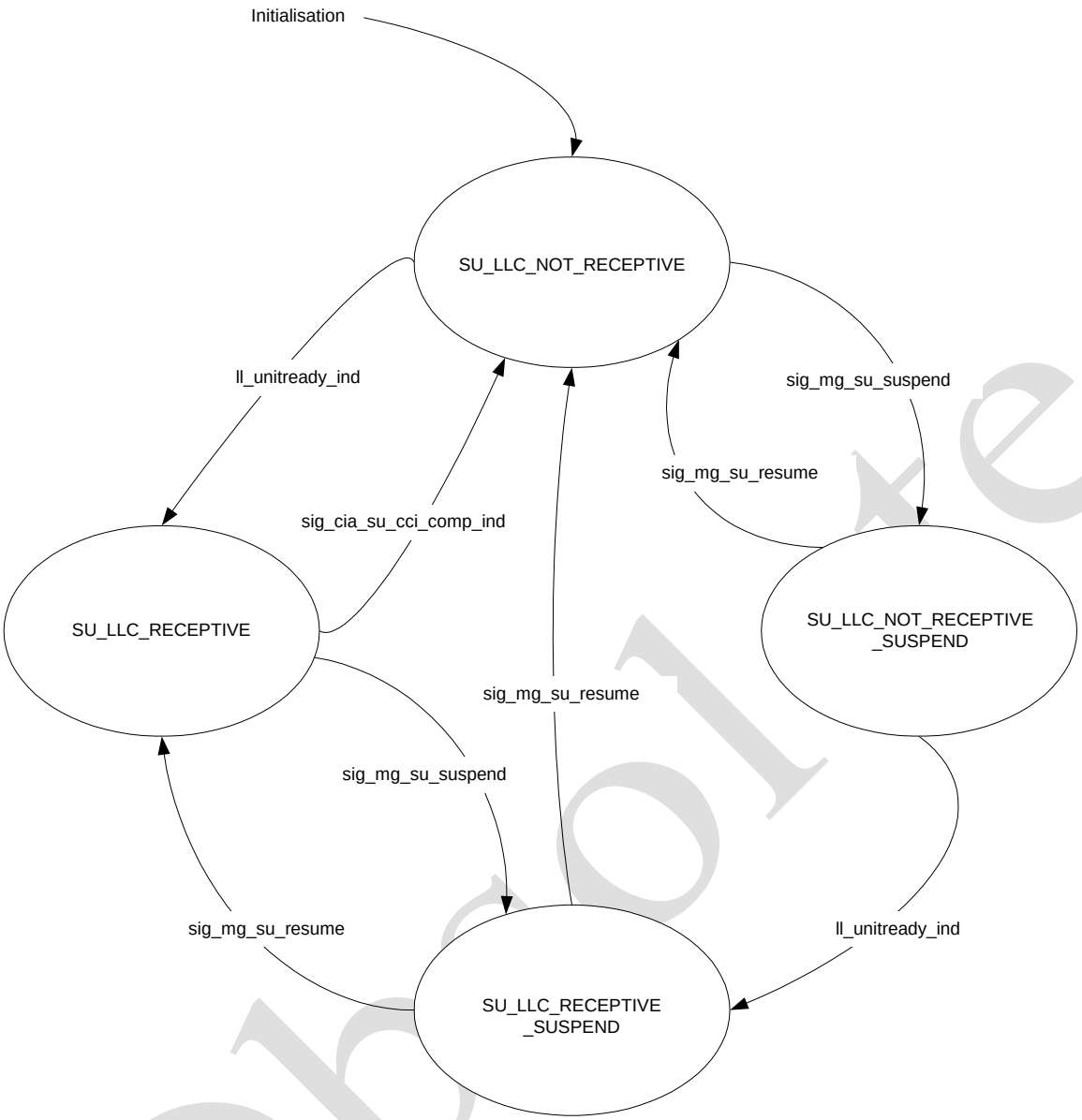

Figure 3 SU module FSM

Below the states are described for the SU-module:

SU_LLC_NOT_RECEPTIVE

This state shows LLC has not indicated readiness for receiving any data primitives from the SU module.

The state is the initial state of the SU module. The state can also be reached when signal *sig_cia_su_cci_comp_ind* or *sig_mg_su_resume* is received.

SU_LLC_RECEPTIVE

This state shows that LLC has indicated that it can receive data primitives.

The state can be entered when primitive *ll_unitready_ind* is received or when signal *sig_mg_su_resume* is received.

SU_LLC_RECEPTIVE_SUSPEND

In this state the SAPI is suspended. The MG module initiates the suspension by sending signal

sig_mg_su_suspend to the SU module. If the case is that primitive *ll_unitready_ind* is received from LLC then the state can also be reached provided that the SU module already is suspended and that LLC cannot receive any data primitives.

SU_LLC_NOT_RECEPTIVE_SUSPEND

In this state the SAPI is suspended. LLC is also not able to receive any data primitives.

The state can be reached when signal *sig_mg_su_suspend* is received from the MG module.

3.1.2.3 SAPI Uplink Acknowledged (SUA) module

This state machine reflects the connection state between the SUA module and the connected LLC SAPI for acknowledged mode of operation.

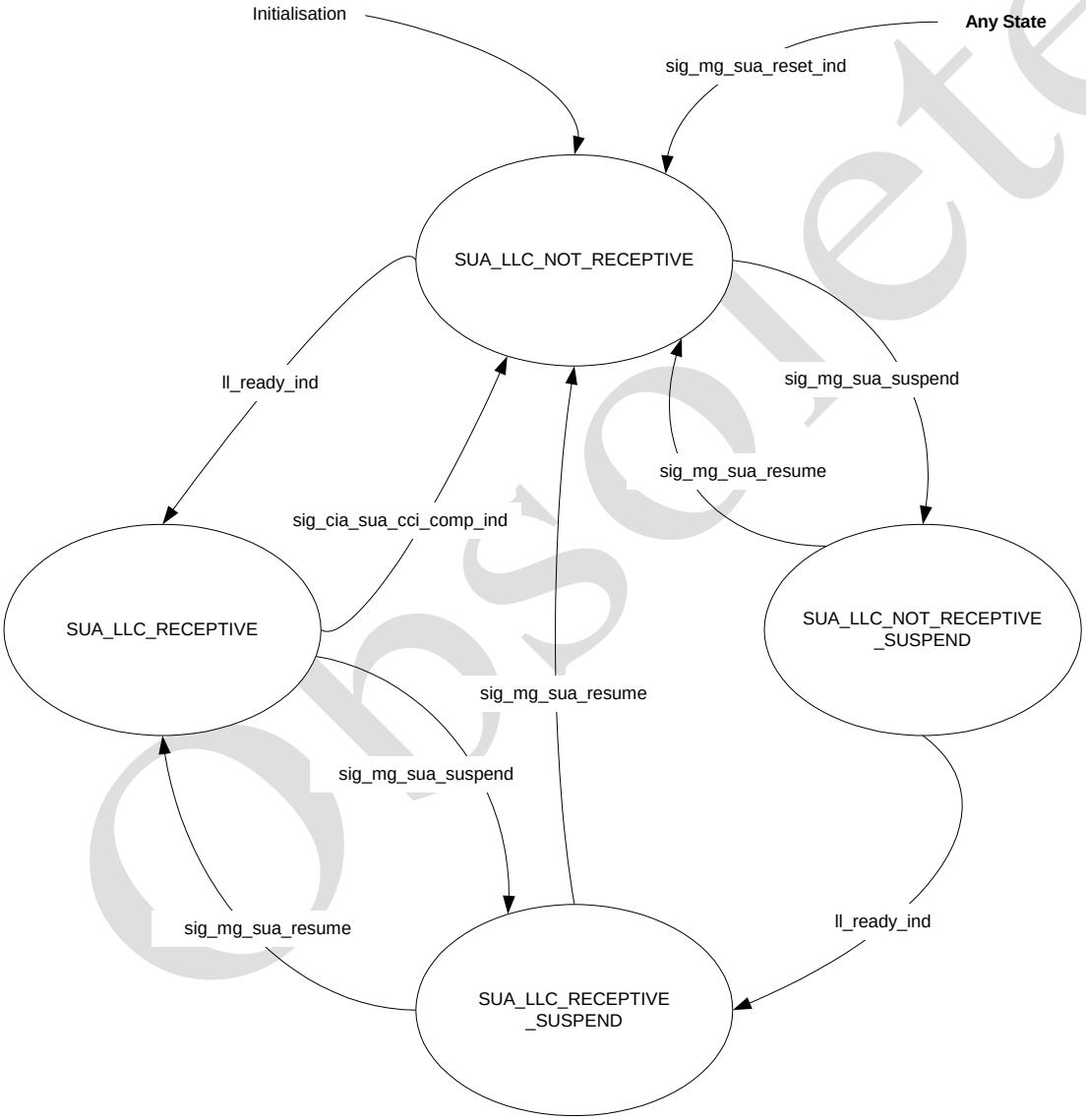

Figure 4 SUA module FSM

Below the states are described for the SUA-module:

SUA_LLC_NOT_RECEPTIVE

This state indicates that LLC presently not has indicated any ability to receive data primitives.

This is the initial state for the SUA module. The state can also be reached at reception of signal *sig_cia_sua_cci_comp_ind*, *sig_mg_sua_reset_ind* or *sig_mg_sua_resume*.

SUA_LLC_RECEPTIVE

This state tells that LLC has indicated ability to receive a data primitive from SUA module.

The state can be reached upon reception of *ll_ready_ind* primitive from LLC, when signal *sig_mg_sua_resume* is received from the MG module.

SUA_LLC_RECEPTIVE_SUSPEND

This state indicates that the SAPI has been suspended.

The state can be reached when primitive *ll_ready_ind* is received from LLC. It can be reached upon reception of signal *sig_mg_sua_suspend* from the MG module within SNDCP.

SUA_LLC_NOT_RECEPTIVE_SUSPEND

This state tells that the SUA module is suspended and that LLC cannot receive any data primitives from SUA module.

The state can be reached when signal *sig_mg_sua_suspend* is received from the MG module.

3.1.3 Downlink modules

3.1.3.1 NSAPI Downlink (ND) module

The ND_ are real states for the NSAPI Downlink module.

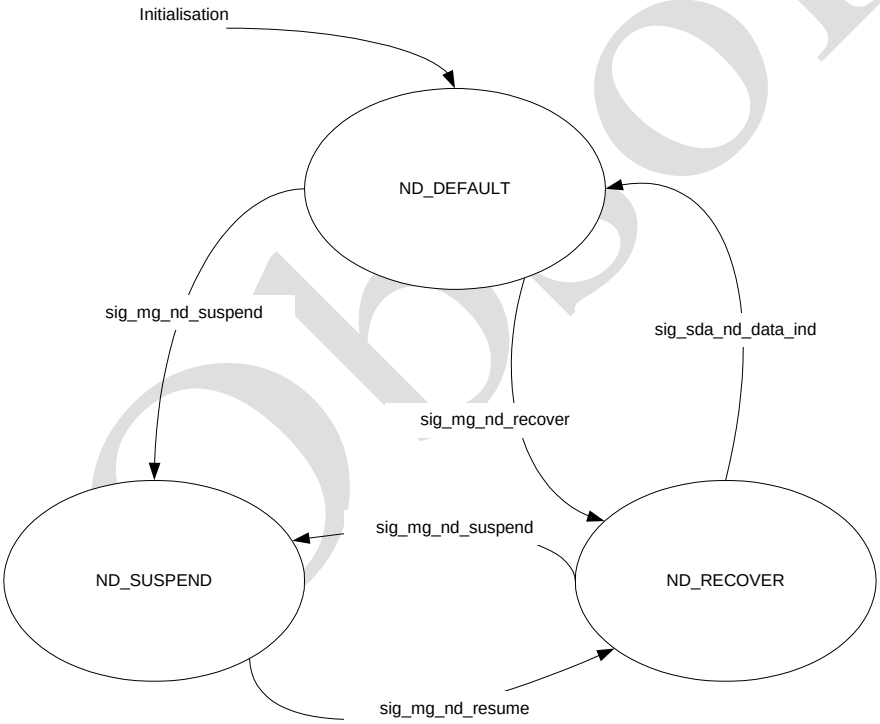

Figure 5 ND module FSM

Below the states are described for the NU-module:

ND_DEFAULT

This is the initial state of the ND module. This state is the normal working state of the ND module. The state is valid for both unacknowledged and acknowledged data primitives.

The state can be reached when receiving signal *sig_sda_nd_data_ind*.

ND_RECOVER

This is a state that only is used in connection with acknowledged mode of operation for the ND module.

The state can be entered when signal *sig_mg_nd_recover* or *sig_mg_nd_resume* is received.

ND_SUSPEND

This state is indicating that the downlink NSAPI is suspended.

The state can be entered when signal *sig_mg_nd_suspend* is received.

3.1.3.2 SAPI Downlink unacknowledged (SD) module

The states defined for the SD module reflects to some extent the ETSI defined states for LLC downlink data handling. The ETSI defined states are included in the state machine.

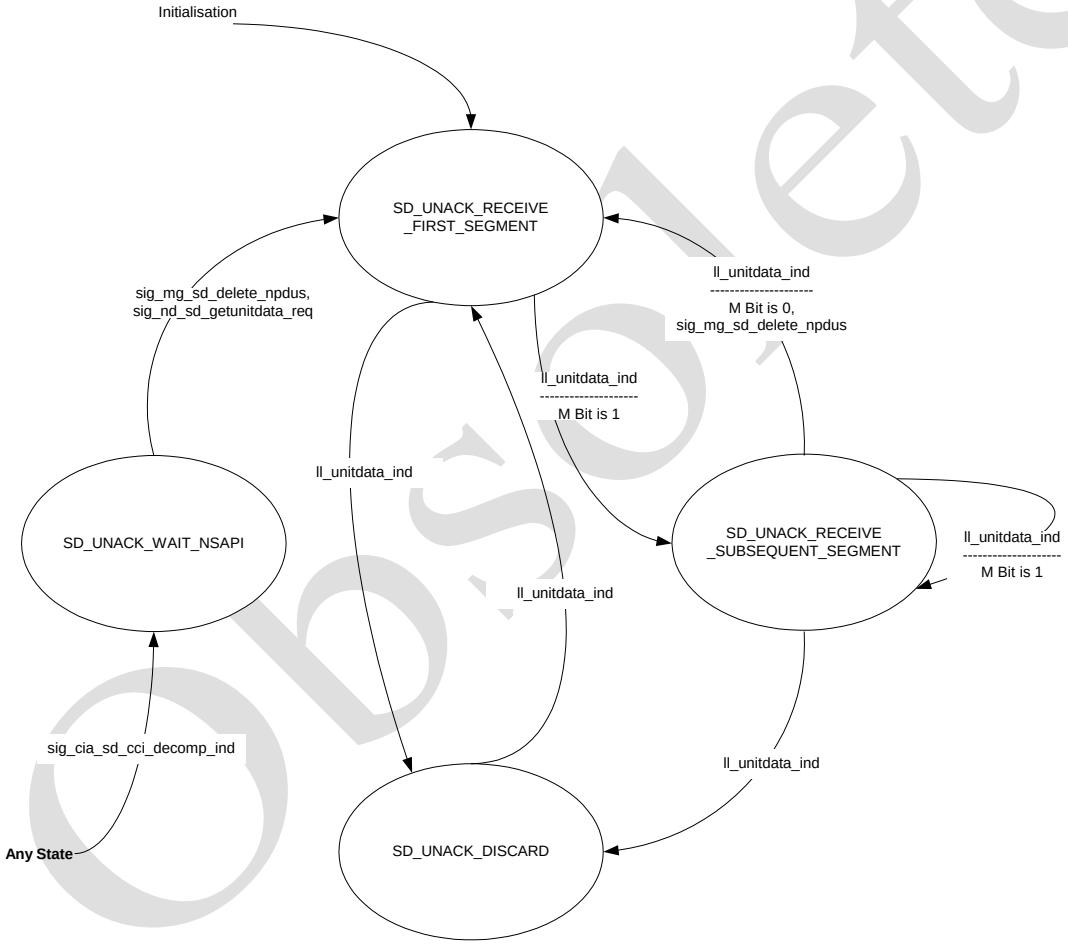

Figure 6 SD module FSM

SD_UNACK_RECEIVE_FIRST_SEGMENT

This is the initial state of the SD module. It is also an ETSI defined state. In this state a data packet indicating that it is the first packet of an NPDU is expected.

The state can be entered when ll_unitdata_ind primitive is received from LLC; with indication of that it is the last segment of an NPDU (M bit is 0). Another possibility for entering this state is reception of signal sig_mg_sd_delete_npdus from the MG module or signal sig_nd_sd_getunitdata_req from the ND module.

SD_UNACK_RECEIVE_SUBSEQUENT_SEGMENT

This state shows that the SD module is expecting at least one more segment belonging to the NPDU currently being received from LLC. The state is again derived from the ETSI specification for SNDCP.

The state can be entered when primitive ll_unitdata_ind is received from LLC with indication of more segments to come in the segment header (M bit is 1).

SD_UNACK_WAIT_NSAPI

In this state the SD module is waiting for ND module to indicate readiness for being able to receive more data.

The state can be entered when signal sig_cia_sd_cci_decomp_ind is received from the CIA module.

SD_UNACK_DISCARD

ETSI defined state. In this state data packets received from LLC are discarded until a packet with indication of that it the last packet of an NPDU. At this point a transition to FIRST_SEGMENT state is executed.

The state can be entered when primitive ll_unitdata_ind from LLC. There has to be detected an error in connection with the decoding of data packet header otherwise the state will not be entered.

3.1.3.3 SAPI Downlink Acknowledged (SDA) module

The SDA uses to some extent the same states as the SD module. The SDA module does not have a discard state since the LLC must guarantee the delivery of acknowledged packets, both in terms of sequence and eventual data loss.

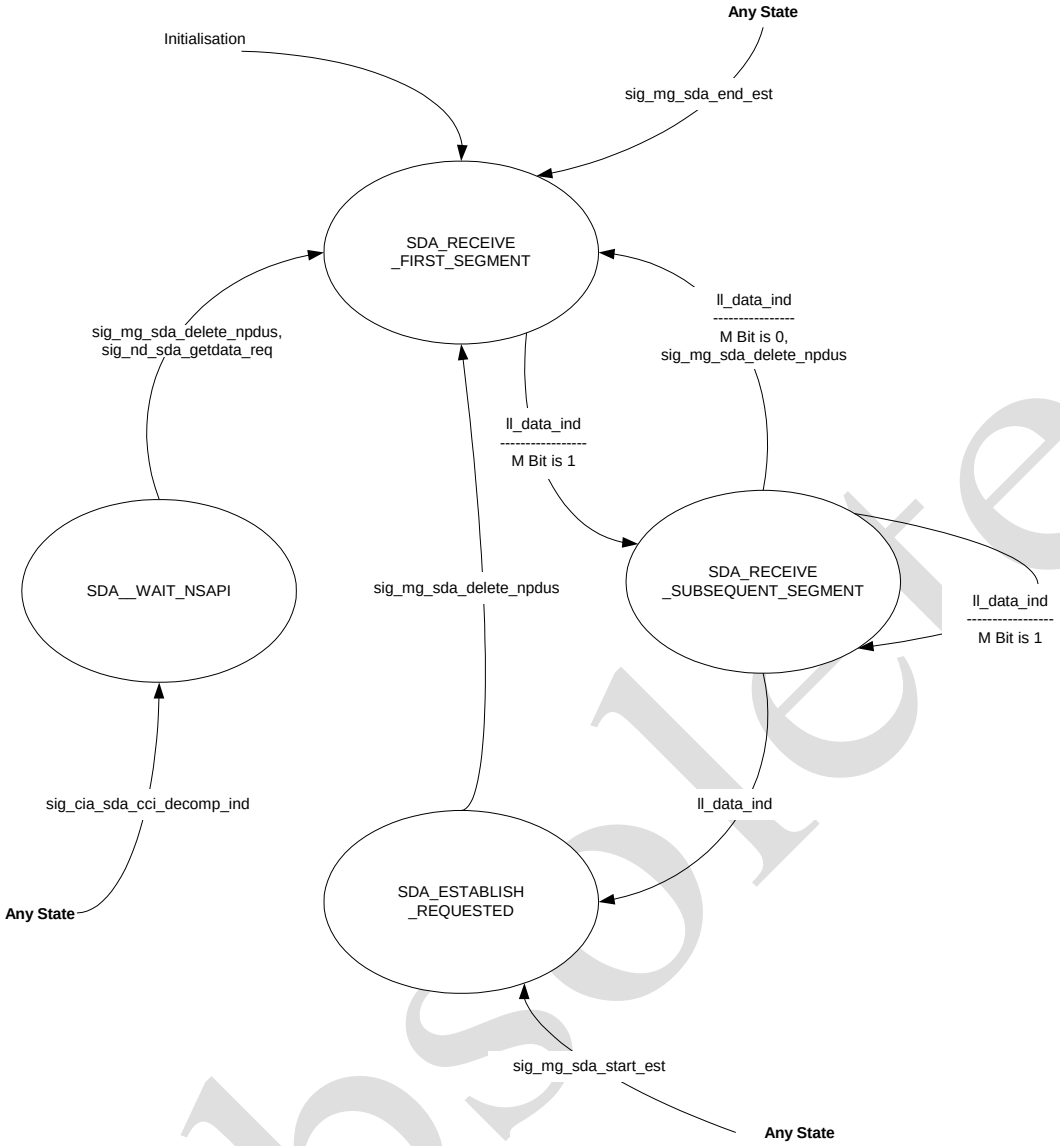

Figure 7 SDA module FSM

SDA_RECEIVE_FIRST_SEGMENT

This is the initial state of SDA module. This is an ETSI defined state and shows that the SDA module is expecting the first segment of an NPDU from the LLC connection.

The state can be entered when receiving ll_data_ind primitive from LLC with indication of that no more segments are to follow to the NPDU (M bit is 0).

It can be entered when receiving signal sig_mg_sda_delete_npdus, sig_mg_sda_end_est or sig_nd_sda_getdata_req.

SDA_ESTABLISH_REQUESTED

This state shows that the SAPI needs an (re)establishment of the acknowledged connection.

The state can be entered when receiving ll_data_ind primitive from LLC or signal sig_mg_sda_start_est from the MG module.

SDA_RECEIVE_SUBSEQUENT_SEGMENT

This state shows that the SDA module is expecting a subsequent segment for an NPDU. The state belongs to the ETSI defined states.

The state can be entered when receiving primitive ll_data_ind from LLC with indication of that more segments to the NPDU are to come (M bit is 1).

SDA_WAIT_NSAPI

In this state the SDA module is waiting for ND module to indicate readiness for being able to receive more data.

The state can be entered when signal sig_cia_sda_cci_decomp_ind is received from the CIA module.

3.2 SNDCP DTI 2 Connect procedures

The MSC in following sections shows the DTI 2 connect procedures. Please note that the actual DTI 2 data transmission is not shown. If information about the data is needed, then please refer to the DTI lib document in reference 2.

Furthermore the XID parameters are not described in any detail, as they are not the focus of this document.

3.2.1 SNDCP context activation, successful

3.2.1.1 Unacknowledged context activation, direction to high layer

Figure 8 below shows an example SNDCP receiving unacknowledged context activation indication from SM. Included in the SNDCP procedure is set-up of DTI 2 connection to the user plane entity. Once the DTI 2 set-up is completed then SNDCP can complete the context activation towards the network. The direction of the set-up indicates that SNDCP communicates with a user plane entity located above SNDCP. (In a kind of layered view.)

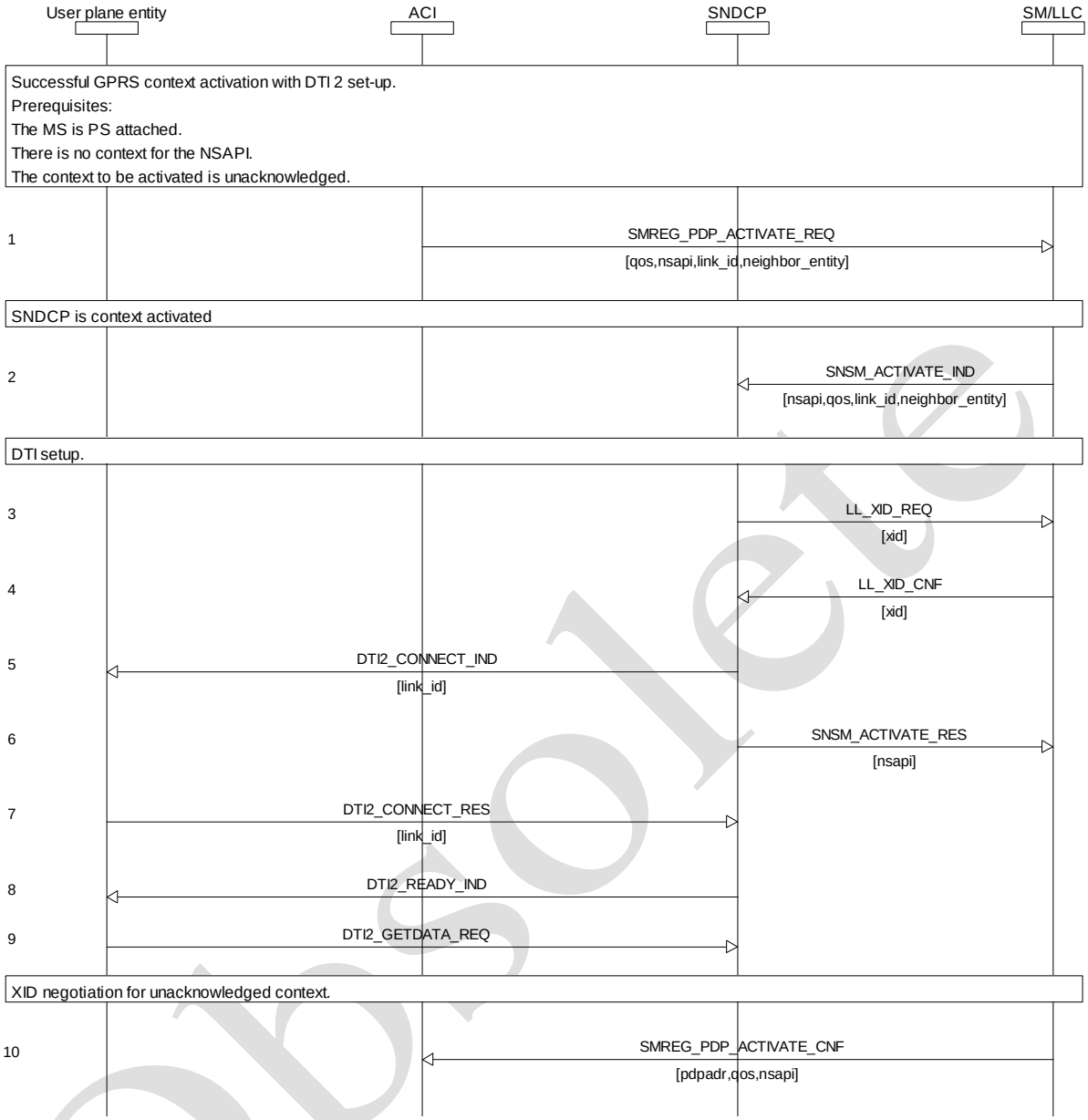

Figure 8 Successful GPRS context activation and DTI 2 set-up.

3.2.1.2 Unacknowledged context activation, direction to low layer

Figure 9 below shows an example of SNDCP receiving unacknowledged context activation indication from SM. Included in the SNDCP procedure is set-up of DTI 2 connection to the user plane entity. Once the DTI 2 set-up is completed then SNDCP can complete the XID negotiation and context activation towards the network. The direction of the set-up indicates that SNDCP communicates with a user plane entity located below SNDCP. (In a kind of layered view.)

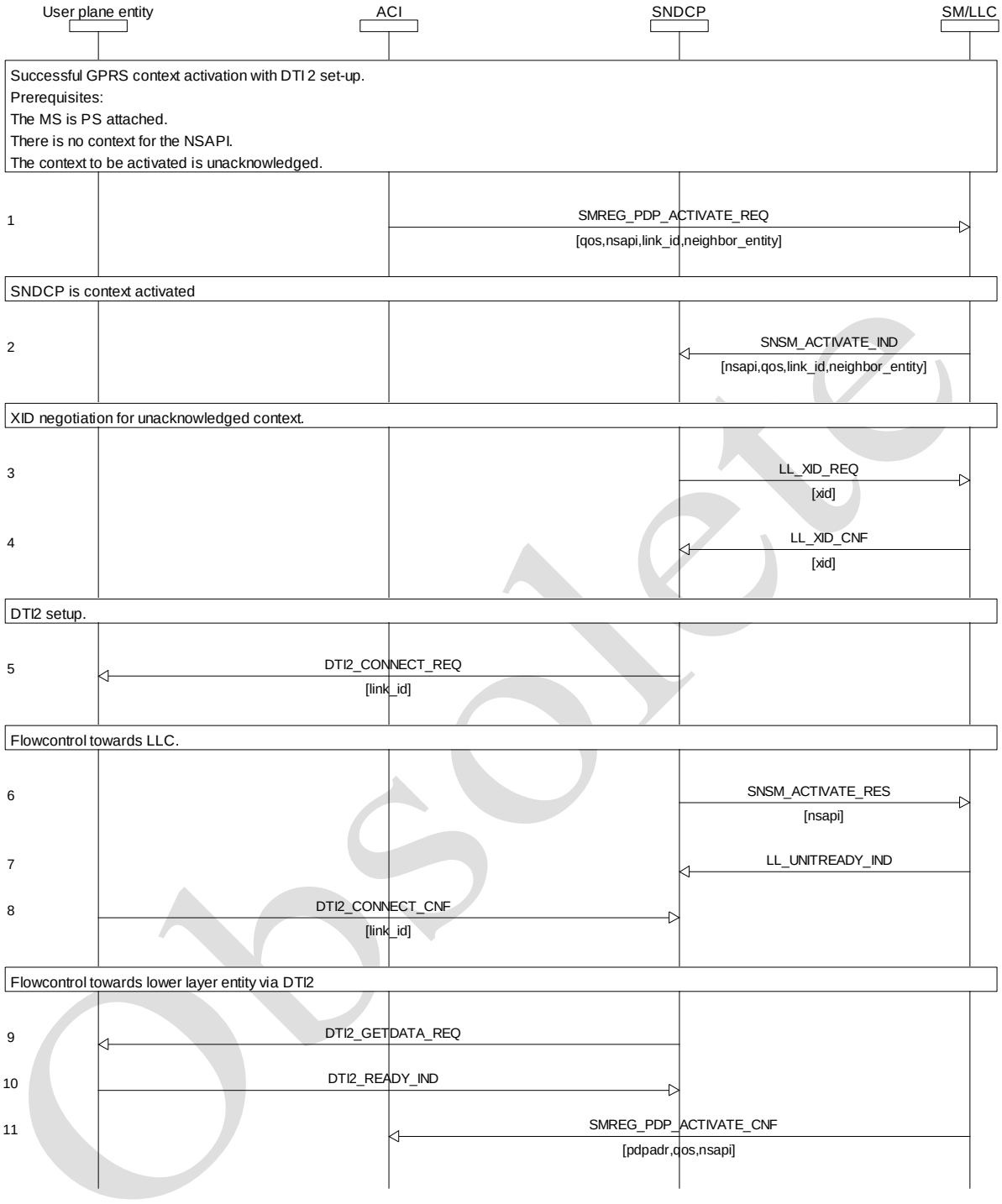

Figure 9 Successful GPRS context activation and DTI 2 set-up

3.2.1.3 Acknowledged context activation, direction to high layer

Figure 10 below shows an example SNDCP receiving acknowledged context activation indication from SM. Included in the SNDCP procedure is set-up of DTI 2 connection to the user plane entity. Once the DTI 2 set-up is completed then SNDCP can complete the establishment and context activation towards the network. The direction of the set-up indicates that SNDCP communicates with a user plane entity located above SNDCP. (In a kind of layered view.)

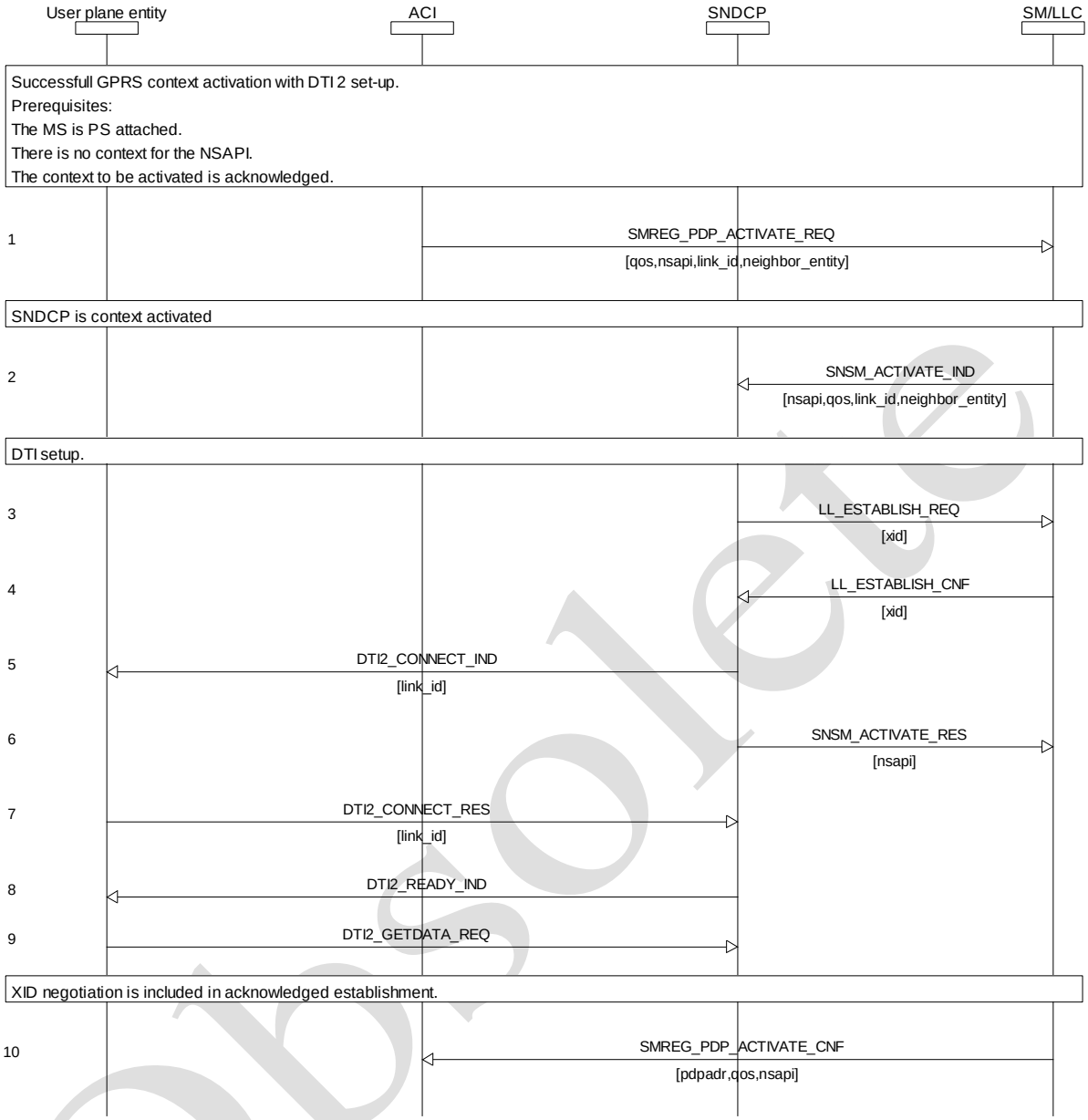

Figure 10 Successful GPRS context activation and DTI 2 set-up

3.2.1.4 Acknowledged context activation, direction to low layer

Figure 11 below shows an example of SNDCP receiving acknowledged context activation indication from SM. Included in the SNDCP procedure is set-up of DTI 2 connection to the user plane entity. Once the DTI 2 set-up is completed then SNDCP can complete the establishment and context activation towards the network. The direction of the set-up indicates that SNDCP communicates with a user plane entity located below SNDCP. (In a kind of layered view.)

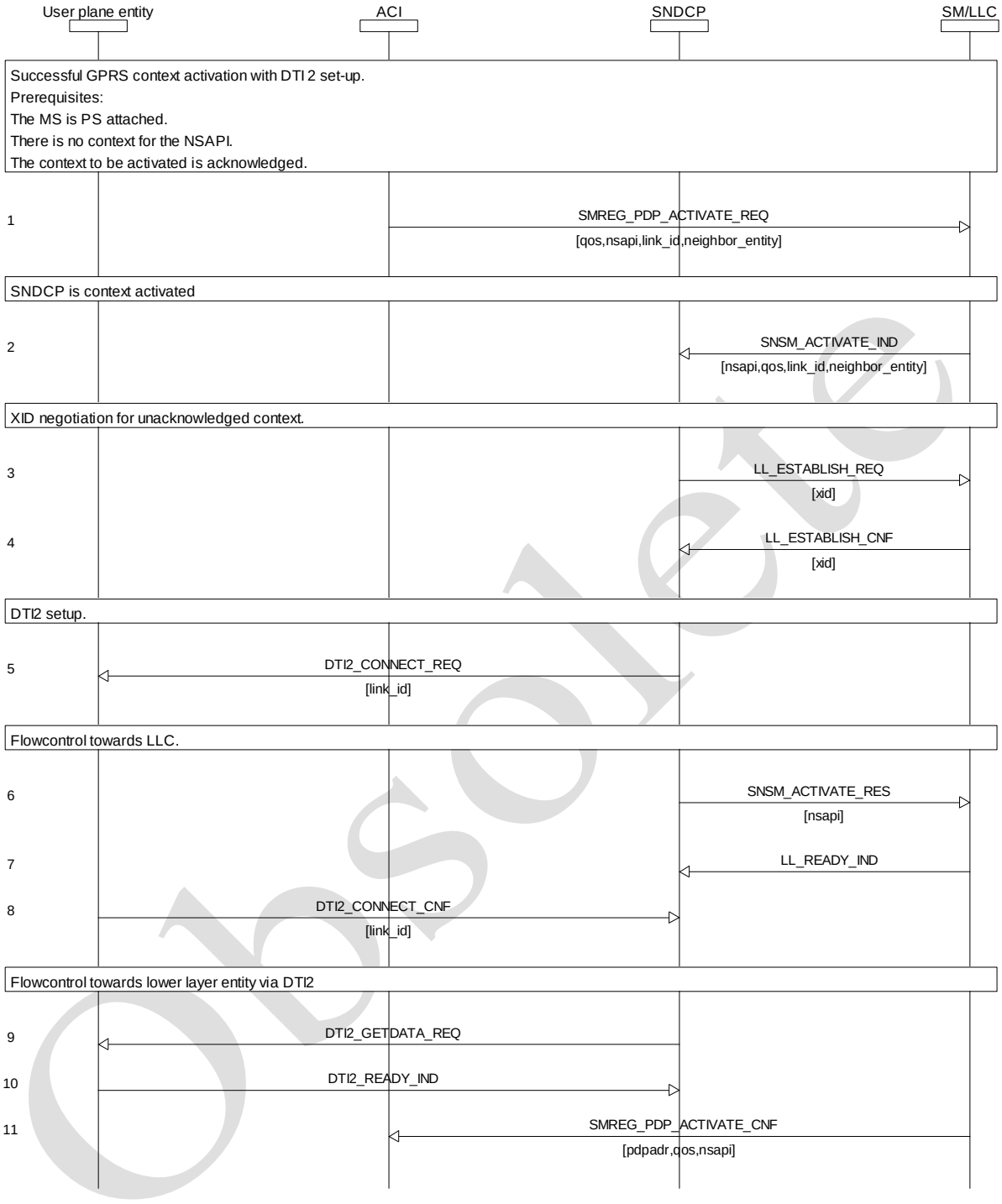

Figure 11 Successful GPRS context activation and DTI 2 set-up

3.2.2 Context activation, unsuccessful, DTI failure

3.2.2.1 Unsuccessful context activation, DTI 2 set-up failure

Figure 12 below shows an example of an unsuccessful PDP context activation where the DTI 2 set-up fails due to disconnection by the user plane entity. SNDCP then expects SNSM_DEACTIVATE_IND from SM to have the context deactivated.

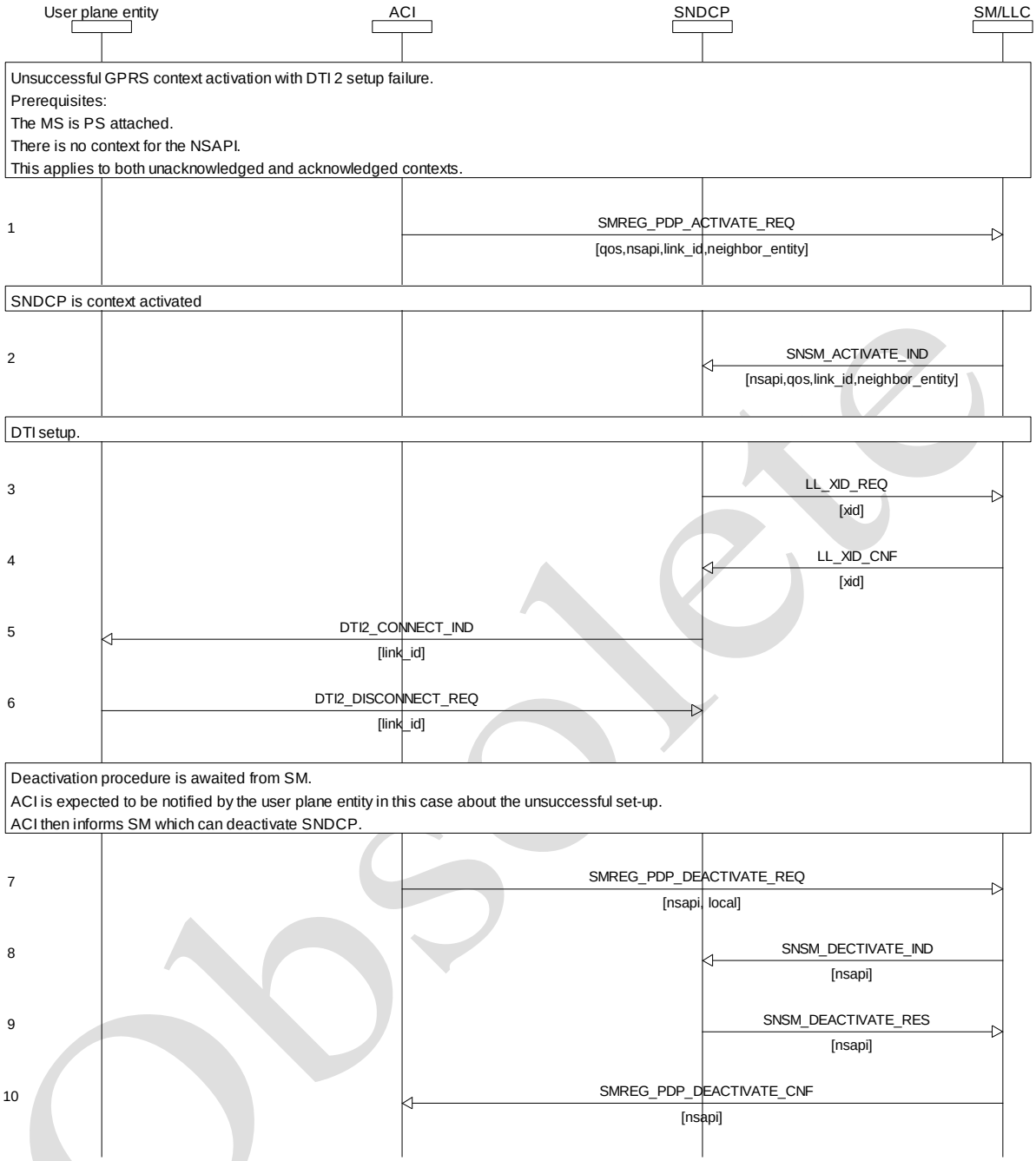

Figure 12 Unsuccessful context activation and DTI 2 set-up failure

3.2.3 Context deactivation, SM initiated

3.2.3.1 Unacknowledged context deactivation

Figure 13 below an unacknowledged context deactivation where DTI_DISCONNECT_IND is sent to the user plane entity from SNDCP.

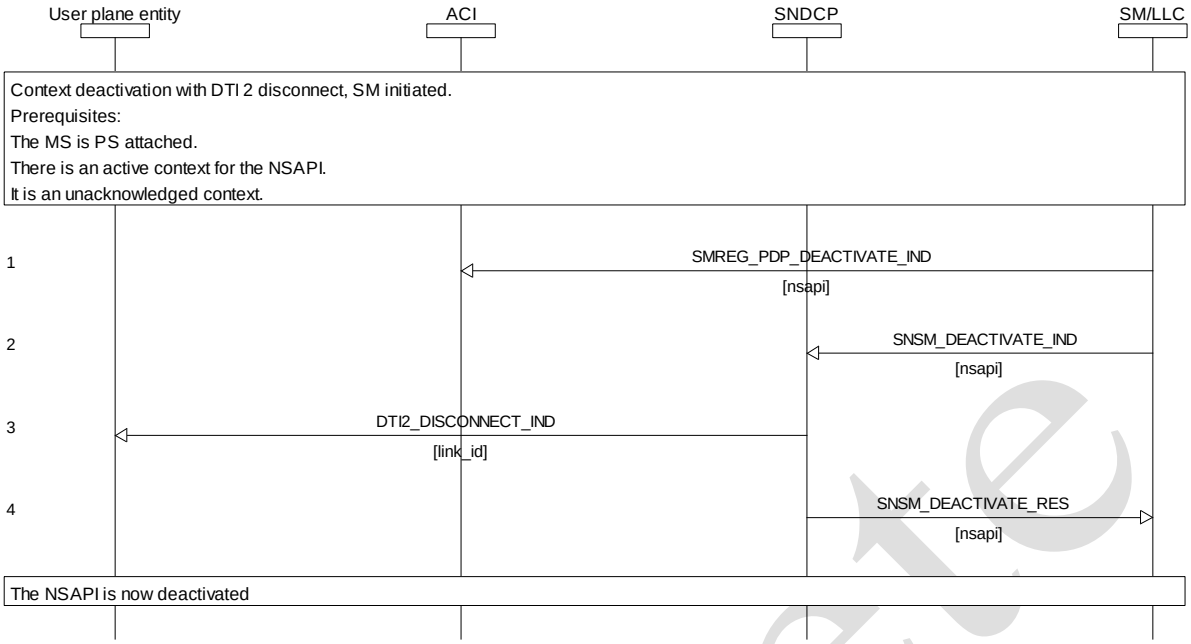

Figure 13 Unacknowledged context deactivation

3.2.3.2 Acknowledged context deactivation

Figure 14 below illustrates, as Figure 13 context deactivation only this time a release of an acknowledged connection is included.

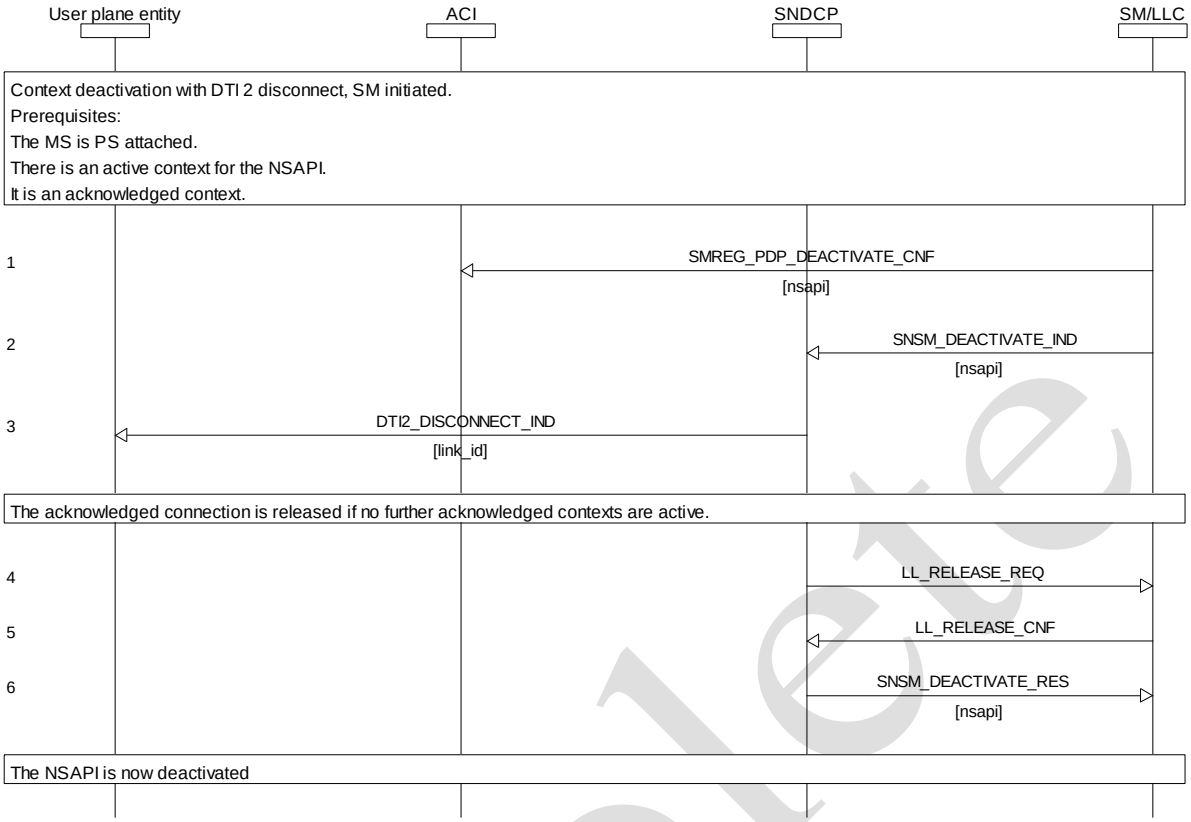

Figure 14 Acknowledged context deactivation