Detailed Specification

Supplementary Service Notifications

Department:	Aalborg Wireless Center
Creation Date:	8 May, 2003
Last Modified:	23 June, 2003 by Anders B. Jørgensen
ID and Version:	8462.716.03.003
Status:	Submitted

0 Document Control

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Texas Instruments reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Texas Instruments. Texas Instruments accepts no liability for any loss or damage arising from the use of any information contained in this document.

The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Texas Instruments.

0.1 Document History

ID	Author	Date	Status
8462.716.03.001	HHV	8 May, 2003	Planned
8462.716.03.002	HHV	22 May, 2003	Submitted
8462.716.03.003	ANBJ	23 June, 2003	Submitted

0.2 References, Abbreviations, Terms

[TI 8010.801] 8010.801, References and Vocabulary, Texas Instruments

Table of Contents

1 Introduction

This document describes the Supplementary Service Notification handling in ACI. The SS Notifications are received from the CC entity and are to be conveyed to the TE via the +CSSI and +CSSU responses. Typically these SS Notifications informs the TE about the current status of the various subscribed services, like if call forwarding is active, a call is waiting etc.

The SS Notification handling described in this document focus on two parts:

- The sending of the correct result code to the TE/MMI with respect to the call direction (MOC/MTC).
- Ensuring that in case of MTC no +CSSU and/or other SS related unsolicited result codes are sent to the TE prior to RING, +CRING, or +CLIP.

2 Sending result codes

2.1 Interpretation of the specification

According to 07.07 the +CSSN AT command enables or disables the unsolicited (+CSSU) and/or the intermidiate (+CSSI) result codes. The application of the result codes is as follows:

- +CSSI: Only applicable for mobile originated calls (MOC).
- +CSSU: Only applicable for mobile terminated calls (MTC) and in case where a final result code is already sent to the TE.

Further 07.07 states that for MT calls the +CSSU result code must be sent after the +CLIP result code. This is interpreted as if the CLIP SS is active the +CSSU result code is sent after the +CLIP result code, otherwise the +CSSU result code is sent after RING/+CRING, but never before RING/+CRING.

2.1.1 Conclusion

The intermediate result code is only used for when an AT command is currently being executed. The only case is for MO calls where the ATD is executed. In all other cases the unsolicited result codes are used since no command is currently being executed.

In case of MO calls the OK result code is not transmitted immediately if the +CSSI result code is enabled in the +CSSN command. The OK result is sent to the TE/MMI when the call is answered as in case the +COLP service is active.

2.1.2 AT command examples

Reception of notifications for MOC:

```
AT+CSSN=1,1 /* Show +CSSI and +CSSU result codes */
OK
ATD12345    /* Dial */
+CSSI: 1    /* CFU Active */
+CSSI: 8    /* Call is deflected */
OK          /* The call is answered */
+CSSU: 2    /* The call is put on hold */
+CSSU: 3    /* The call is retrieved */
```

Reception of notifications for MTC:

```
AT+CSSN=1,1 /* Show +CSSI and +CSSU result codes */
OK
RING        /* Incoming call */
+CSSU: 0    /* This is a forwarded call */
ATA         /* Answer the call */
OK
+CSSU: 2    /* The call is put on hold */
+CSSU: 3    /* The call is retrieved */
```

2.1.3 MSC's

Here are four situations where the MSC's describe the functionality of the CSSI and CSSU.

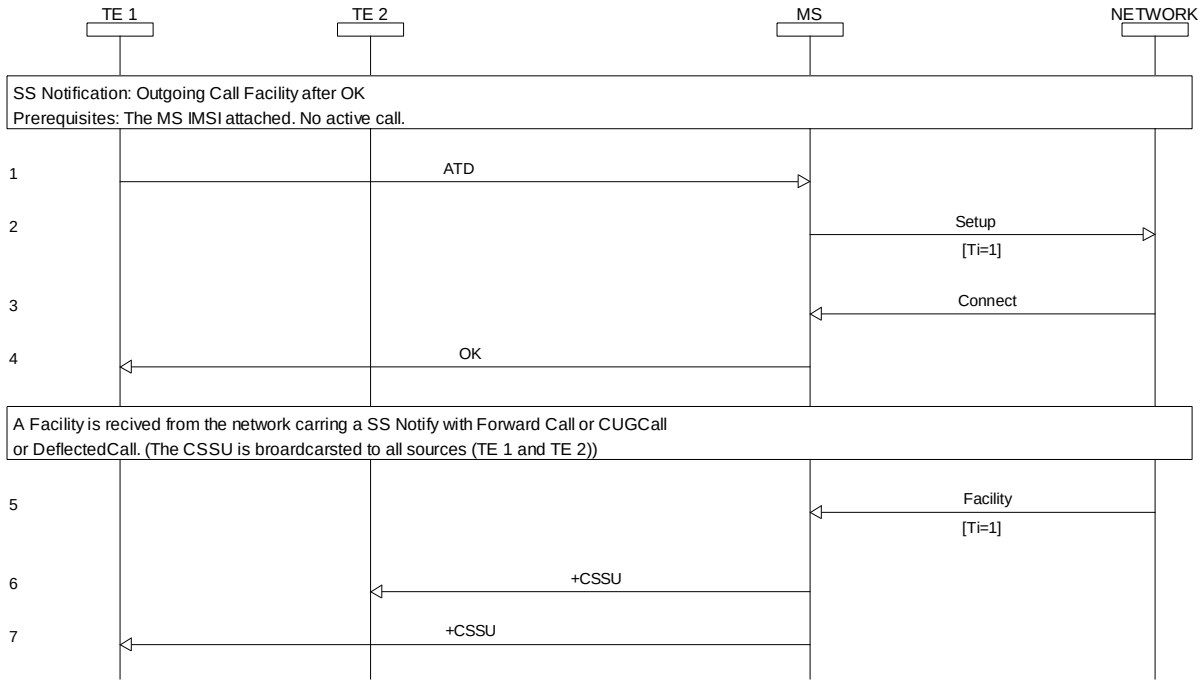

Figure 1 Outgoing Call Facility after OK

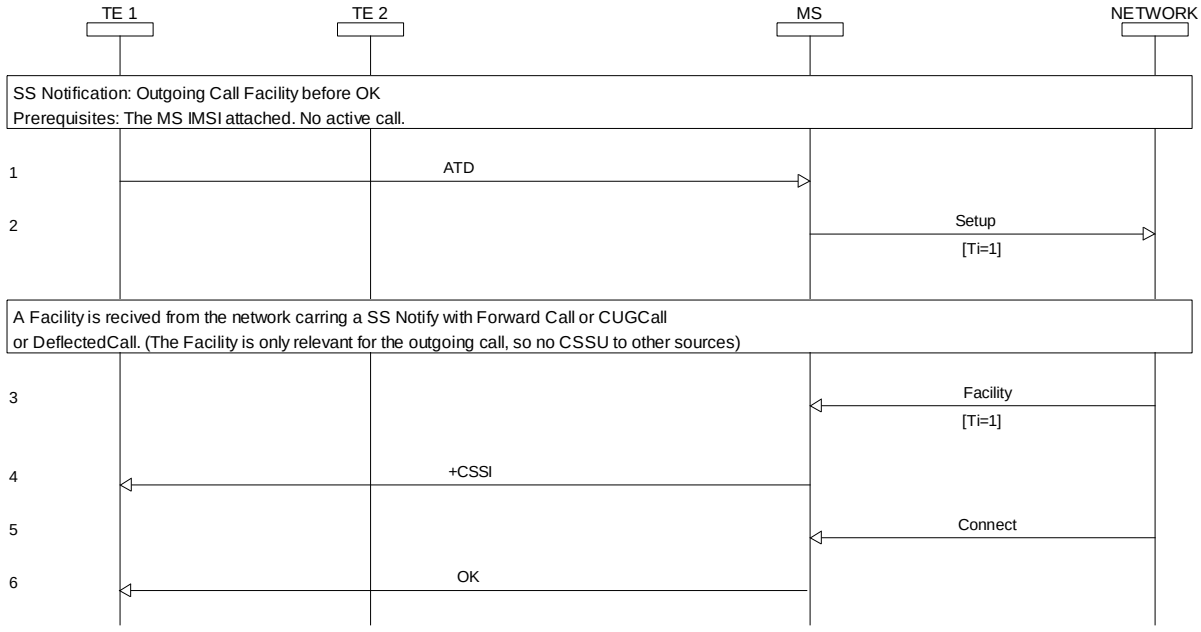

Figure 2 Outgoing Call Facility before OK

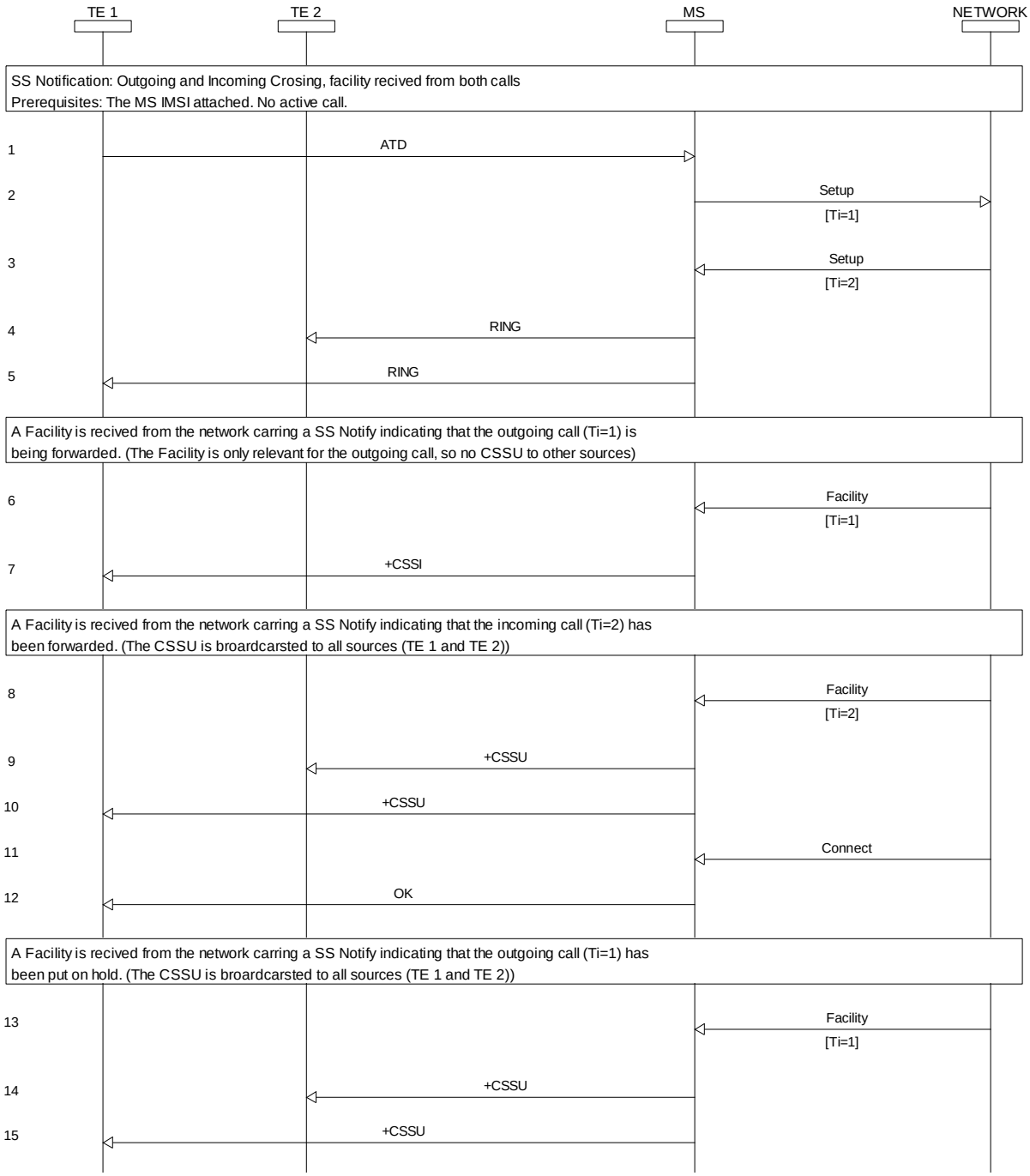

Figure 3 Outgoing and Incoming crossing

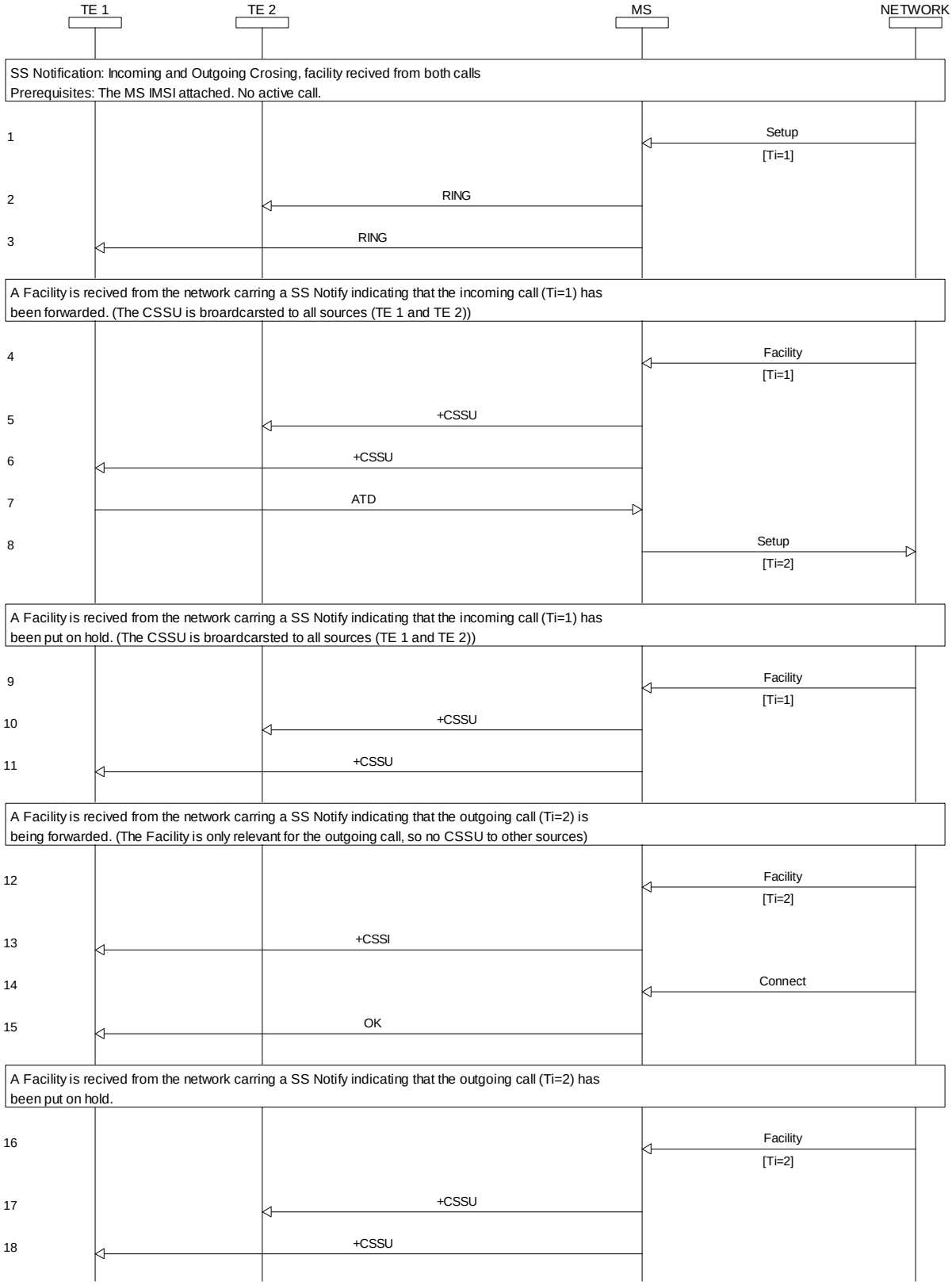

Figure 4 Incoming and Outgoing crossing

2.2 Implementation

2.2.1 Overview

When a MNCC_FACILITY_IND is processed in PSA_CC it is partially CCD decoded to determine the component type (INVOKE, RETURN_RESULT, RETURN_ERROR, RETURN_REJECT). Only the INVOKE component is of interest with respect to the sending of intermediate and unsolicited result codes to the TE/MMI.

As shown in Figure 5, the INVOKE component is disassembled (psa_CC_dasmInvokeCmp) to get the OPERATION code. Only the OPC_NOTIFY_SS and OPC_FWD_CHECK_SS_IND are able to send +CSSx result codes to the TE/MMI, thus the OPC_FWD_CHARGE_ADVICE is of no interest in this document. The OPC_FWD_CHECK_SS_IND operation handling is simply to convey the component parameters to the TE/MMI in a +CSSU result code, and is not treated further in this document. This leaves the OPC_NOTIFY_SS (cmhCC_NotifySS function) for further study.

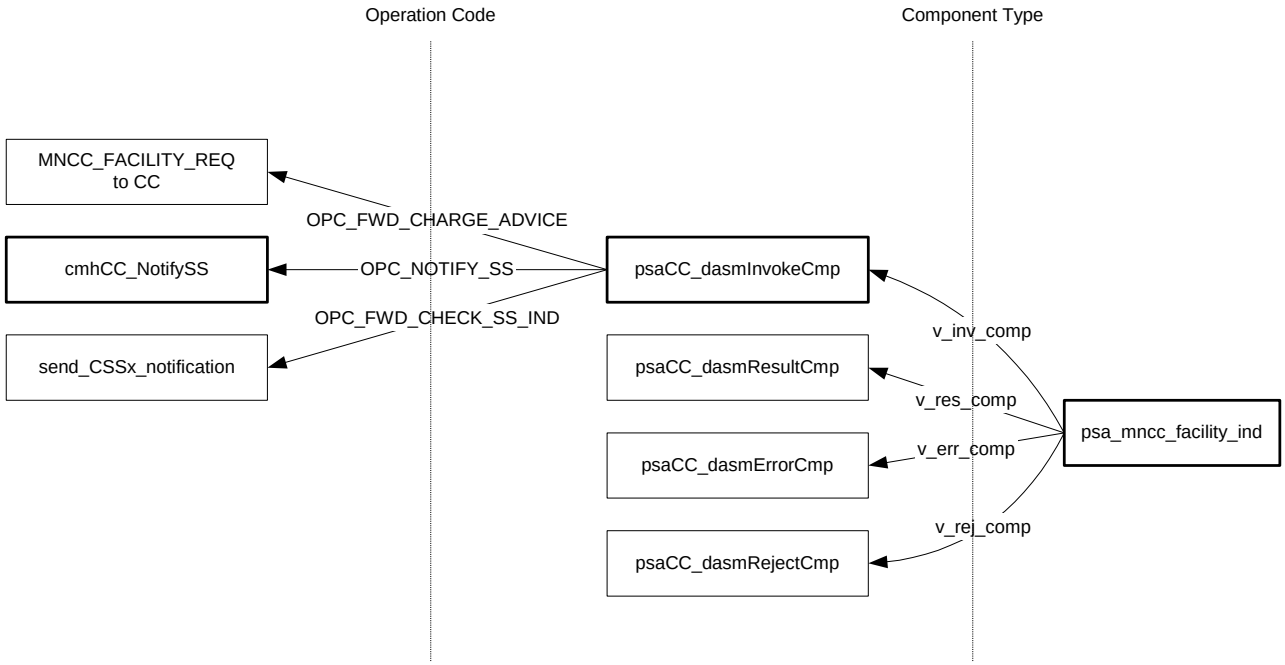

Figure 5 Overview of the funtion path for a SS Notifiy Invoke component

Figure 6 shows the decomposition of the current implementation for the cmhCC_NotifySS function. The function is able to branch for each of the CCD decoded ARGUMENT type for the "NotifySS" OPERATION.

Some of the ARGUMENTs indicates that a service is active just be the presence of the ARGUMENT itself. The handling of these is implemented directly in the cmhCC_NotifySS function. Other ARGUMENTs have additional parameters and are implemented in a function. The latter type conveys supplementary service specific information, thus the handling (transmission of +CSSI or +CSSU) is as shown in Figure 6.

The type of result code (+CSSI or +CSSU) to be transmitted to the TE/MMI is independent from the supplementary. The send_CSSx_notification function is called which then check the call direction and currently executing command to determine the type of result code to transmit.

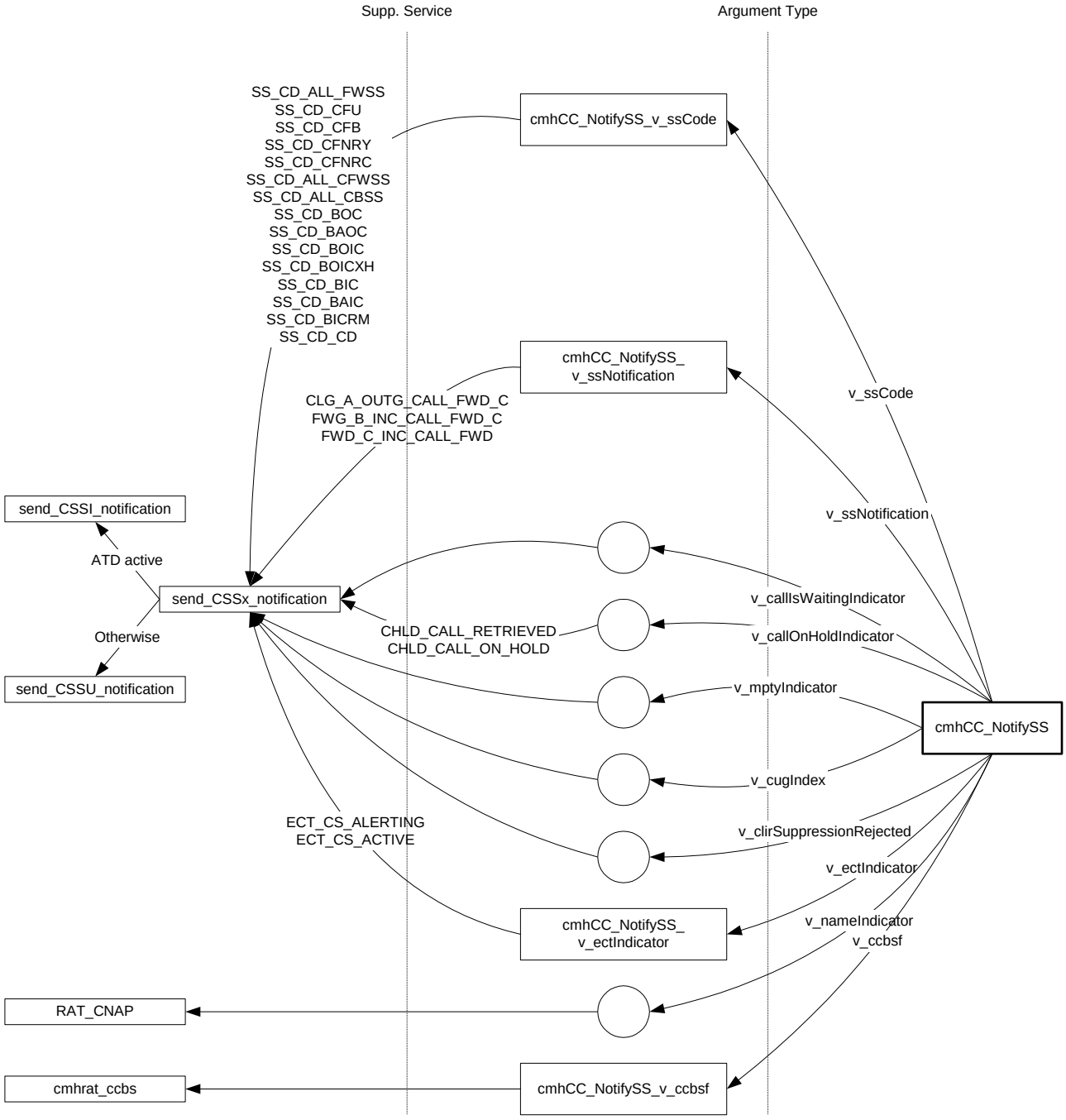

Figure 6 Decomposition of the NotifySS INVOKE component handling

2.2.2 Retaining the OK for MOC

For Mobile Originated speech calls the OK result code is normally sent to the TE/MMI immediately. This would however prohibit the use of +CSSI result codes since it is only applicable between the ATD and the OK result code. The transmission of the OK result code is the same as for the COLP service where the cmhCC_atdsendok function is called to determine whether OK should be sent immediately or not.

2.2.3 send_CSSx_Notification

The following is the function prototype for the send_CSSx_Notification:

```
void send_CSSx_Notification( T_ACI_CSSX_CODE cssx_code,
                             SHORT           index,
                             CHAR            *number,
                             T_ACI_TOA       *toa,
                             CHAR            *subaddr,
                             T_ACI_TOS       *tos )
```

Table 1 Shows the mapping between the internal CSSX codes and the external +CSSI/+CSSU codes according to 07.07 (or 27.007). In some cases mapping is not possible and the result code is not sent to the TE/MMI.

T_ACI_CSSX_CODE	+CSSI code	+CSSU code	Comments
CSSx_CODE_NotPresent	Not mapped	Not mapped	-
CSSx_CODE_CFUActive	0	Not mapped	-
CSSx_CODE_SomeCCFActive	1	Not mapped	-
CSSx_CODE_ForwardedCall	2	0	-
CSSx_CODE_CallWaiting	3	Not mapped	-
CSSx_CODE_CUGCall	4	1	Index parameter is CUG Index
CSSx_CODE_OutCallsBarred	5	Not mapped	-
CSSx_CODE_IncCallsBarred	6	Not mapped	-
CSSx_CODE_CLIRSupRej	7	Not mapped	-
CSSx_CODE_DeflectedCall	8	9	Redirecting number allowed for +CSSU
CSSx_CODE_OnHold	Not mapped	2	-
CSSx_CODE_Retrieved	Not mapped	3	-
CSSx_CODE_Multiparty	Not mapped	4	-
CSSx_CODE_HeldCallRel	Not mapped	5	-
CSSx_CODE_FwrdCheckSS	Not mapped	6	OPERATION = OPC_FWD_CHECK_SS_IND
CSSx_CODE_ECTAlert	Not mapped	7	ECT number allowed for +CSSU
CSSx_CODE_ECTConnect	Not mapped	8	ECT number allowed for +CSSU
CSSx_CODE_AddIncCallForwarded	Not mapped	10	Release ‘99

Table 1 Mapping between internal CSSX codes to external +CSSI or +CSSU codes

3 Facility buffering for MT calls

In some situations the TE/MMI is not alerted (RING, +CRING etc.) on the reception of an incoming call indication from CC (MNCC_SETUP_IND). The RING or +CRING is first sent to the TE/MMI when the TCH is assigned and synchronised (MNCC_SYNC_IND). In this situation one or more Facility informations can be received after the MNCC_SETUP_IND but before the MMI/TE is alerted. It is however not possible to send any call related information at this point (see section 2.1).

Section 3.1 has a detailed description of the implementation and the section 4 holds all the various MSC scenarios.

3.1 Facility Linked List in the Call Table

When an MNCC_FACILITY_IND is received the call direction and TCHasg is checked to see whether to buffer the primitive or to proceed with the normal facility handling. If the facility needs to be buffered (call direction is MTC and TCHasg is FALSE) it is put in the linked facility list (FIFO buffer) in the call table of the CC Shared Parameters store as shown in Figure 7. The primitive is NOT de-allocated, as would be the normal case.

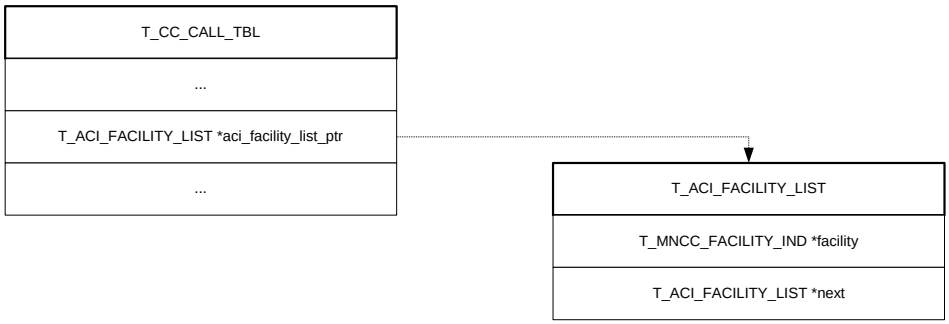

Figure 7 Facility linked list in the CC call table

When the TCH is assigned CMH_CC is notified by PSA_CC calling the cmhCC_IncomingCall. After sending the RING/+CRING and optionally +CLIP in the cmhCC_IncomingCall it is checked if any Facility primitives are stored. If this is the case they are send to PSA_CC by calling the psa_mncc_facility_ind function with the buffered primitive as parameter, until no more Facility primitives are left. The psa_mncc_facility_ind function automatically de-allocates the primitive as this is handled in the same way as if it was just received.

In case of network initiated call release any buffered Facility primitives are de-allocated when releasing the call table entry (psaCC_FreeCtbNtry).

Affected procedures:

- psa_mncc_facility_ind
- cmhCC_IncomingCall
- psaCC_FreeCtbNtry

4 Message Sequence Charts

The following sections shows how SS Notification is handled for MO and MT CS voice calls. Note that only relevant signalling is shown on the MSCs.

4.1 Mobile Originated Call

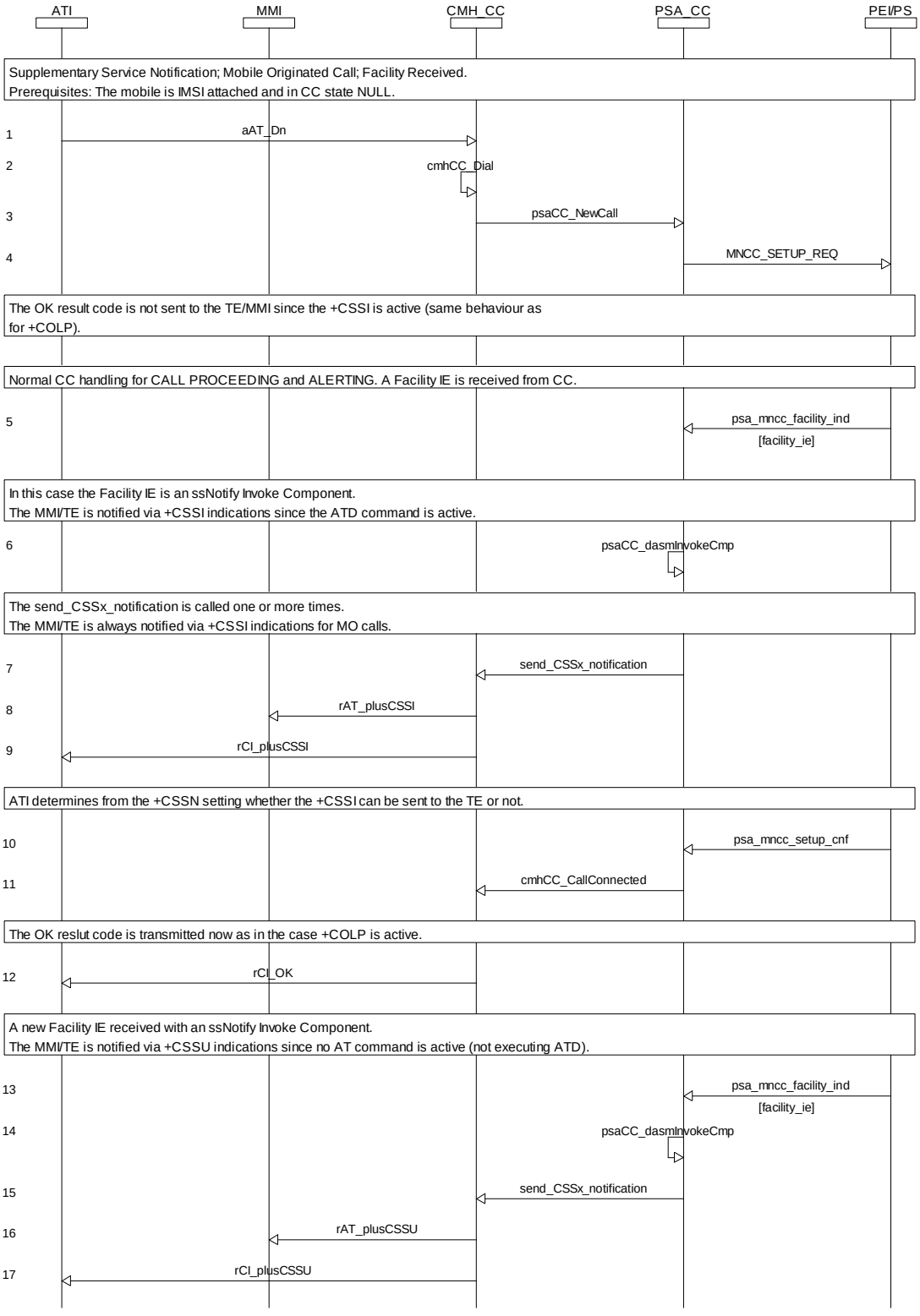

Figure 8 MOC: Facility received after MNCC_SETUP_REQ

4.2 Mobile Terminated Call

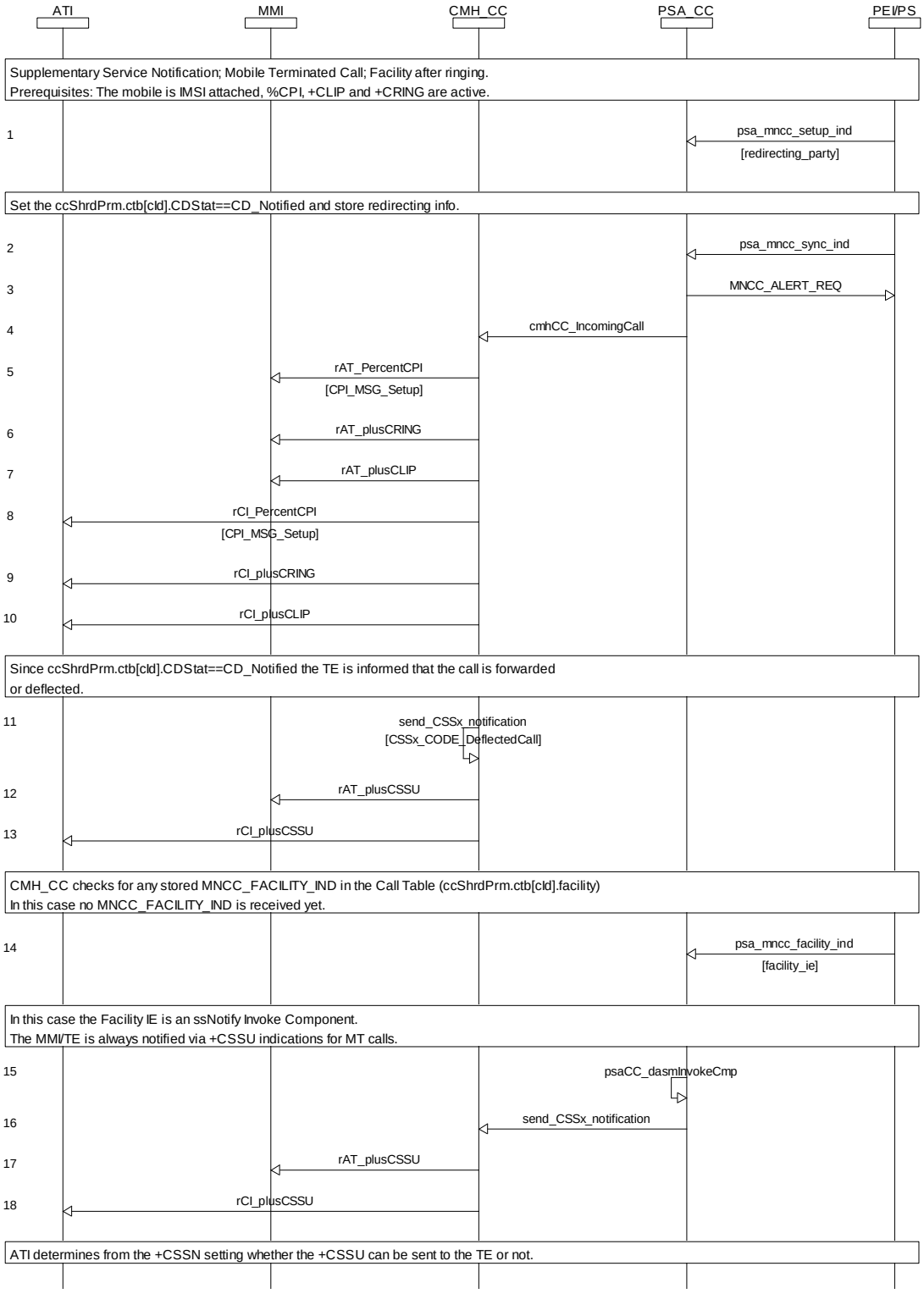

Figure 9 Facility received after TCH assignment

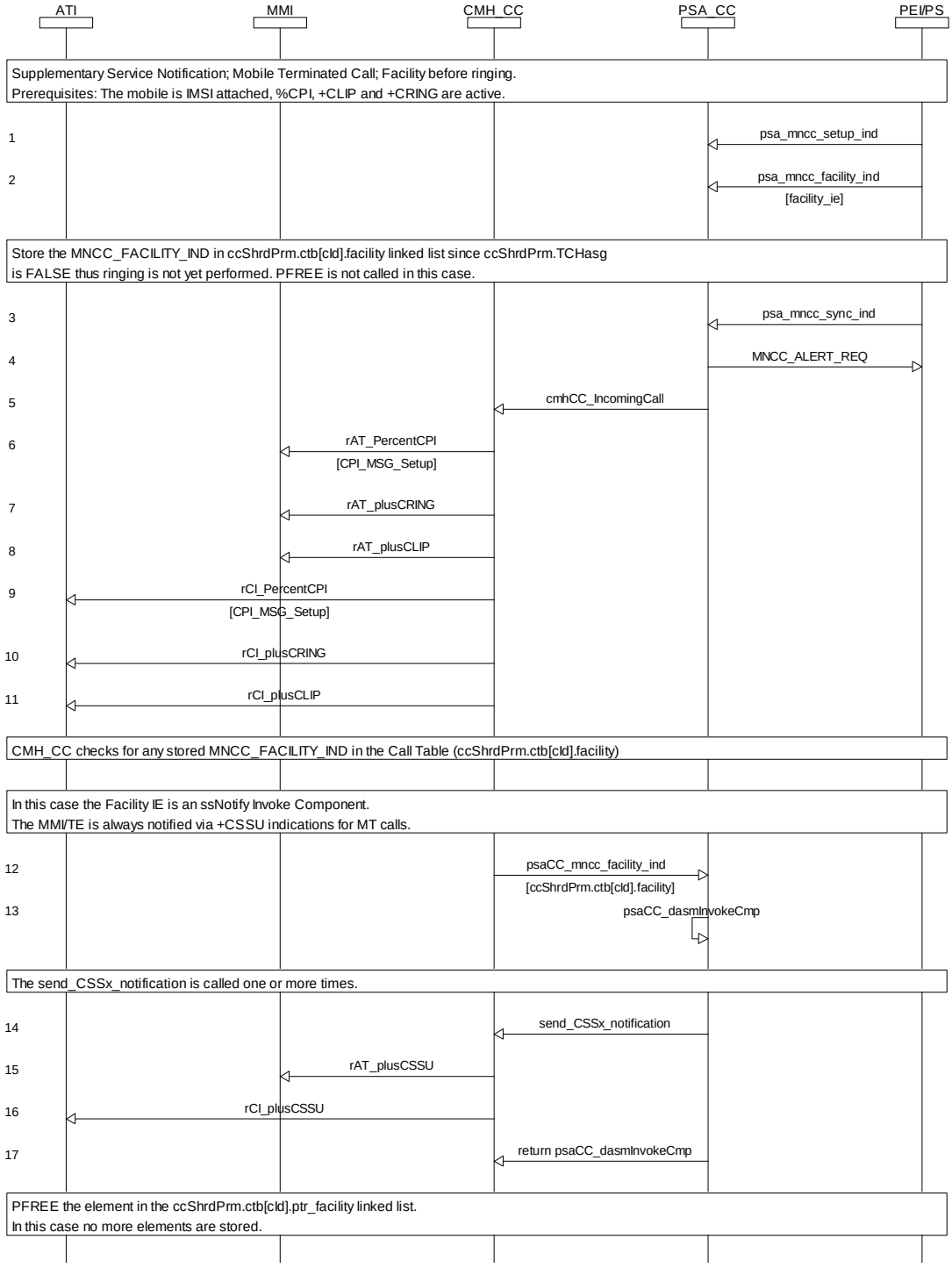

Figure 10 Facility received before TCH assignment

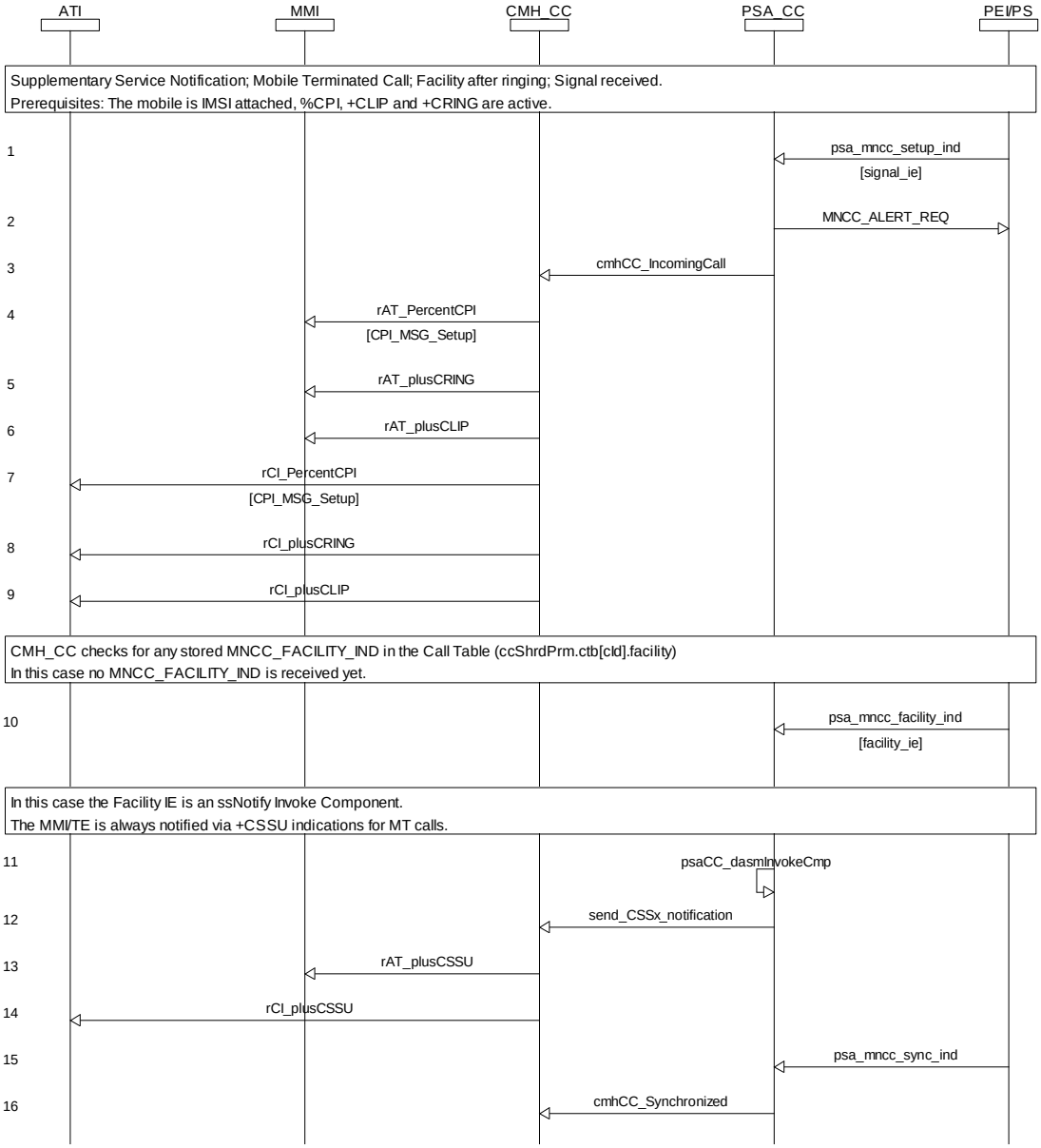

Figure 11 Facility received before TCH assignment; Signal Information Element received

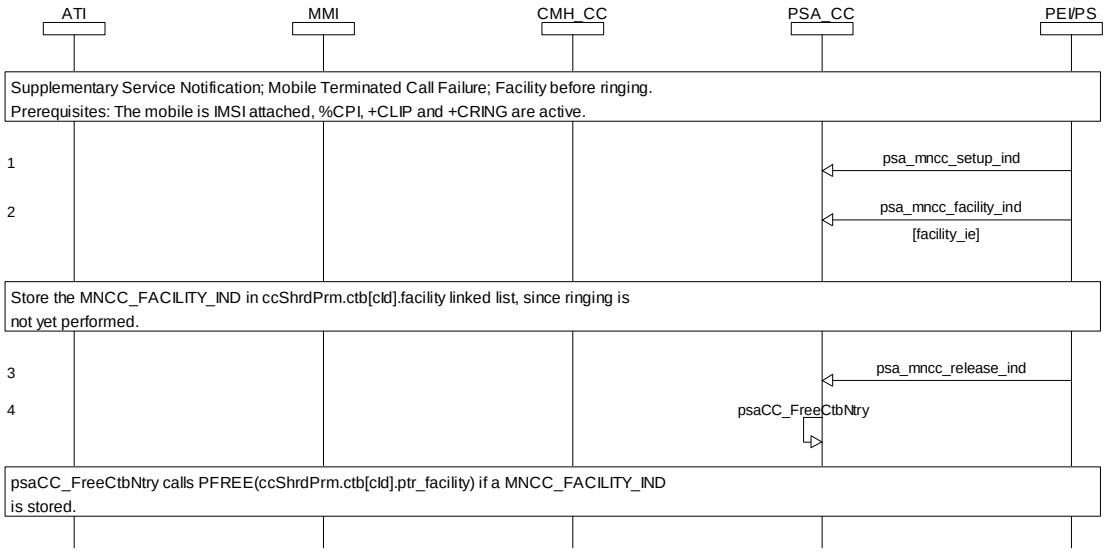

Figure 12 Facility received before TCH assignment; call establishment fails

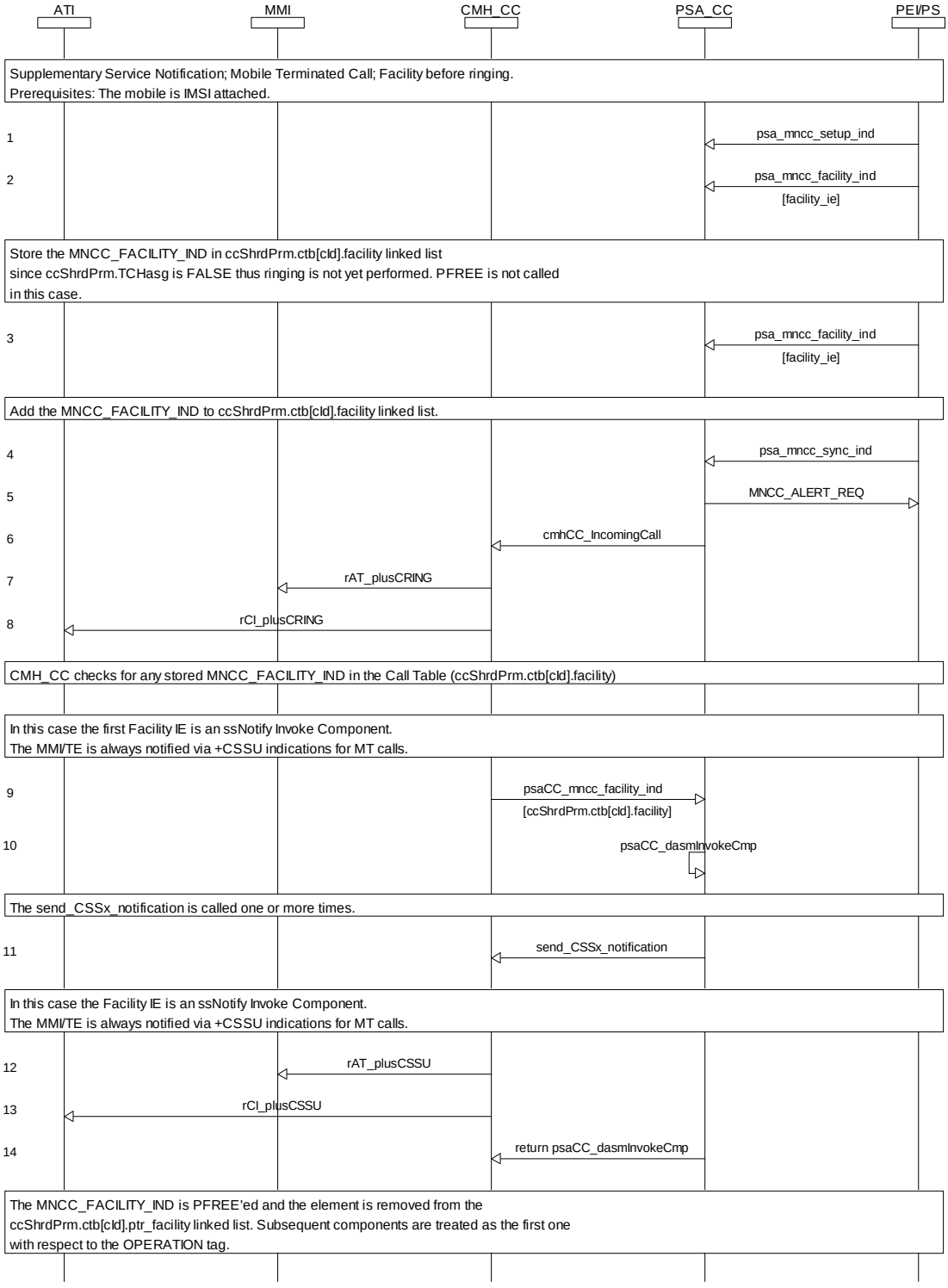

Figure 13 Facility before TCH assignment; multiple Facility IEs received