3GPP TSG CN Plenary Meeting #26 **NP-040486**
08-10 December 2004, Athens, GREECE

Source: **CN5 (OSA)**

Title: **2 Rel-6 CR 29.198-xy OSA API SCF**

Agenda item: **9.7 (OSA Enhancements [OSA3])**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-040486	29.198-02	048	--	Rel-6	Correct the Exception Hierarchy Annex to include new exceptions used in MultiMediaMessaging SCF	F	6.2.0	N5-040789	OSA3
NP-040486	29.198-04-2	027	--	Rel-6	Correct GCC for Multiservice Network Support	F	6.2.0	N5-040723	OSA3

CR-Form-v7

CHANGE REQUEST

⌘ **29.198-04-2** **CR** **027** ⌘ **rev** **-** ⌘ Current version: **6.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network **X**

Title:	⌘	Correct GCC for Multiservice Network Support
Source:	⌘	CN5 AePONA (Eamonn Murray)
Work item code:	⌘	OSA3
Date:	⌘	05/11/2004
Category:	⌘	**F**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Release 6 of the MPCC service has introduced additional explanatory text to outline how the MPCC service may operate with multiservice networks. Multiservice networks may allow more than one service or application to gain control of the call at given point in time, and therefore the description of the notification provisioning process for MPCC and the checking of notification criteria overlap has been expanded to allow the MPCC service to take advantage of the capabilities that exist within multiservice networks. As the capability of Multiservice networks to allow multiple applications or services to gain control of a call at a given point in time is a feature of the underlying network, such capability or behaviour should not be restricted to MPCC. It is therefore necessary to extend the description of Multiservice network support to GCC.
Summary of change:	⌘	Intoduce text to GCC enableCallNotification consistent with current MPCC createNotification text, in order to allow GCC to support multiservice networks.
Consequences if not approved:	⌘	The OSA APIs shall not be truly network agnostic, as different levels of network capability shall be restricted to an arbitrary subset of the eligible OSA specifications.

Clauses affected:	⌘	6.1

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

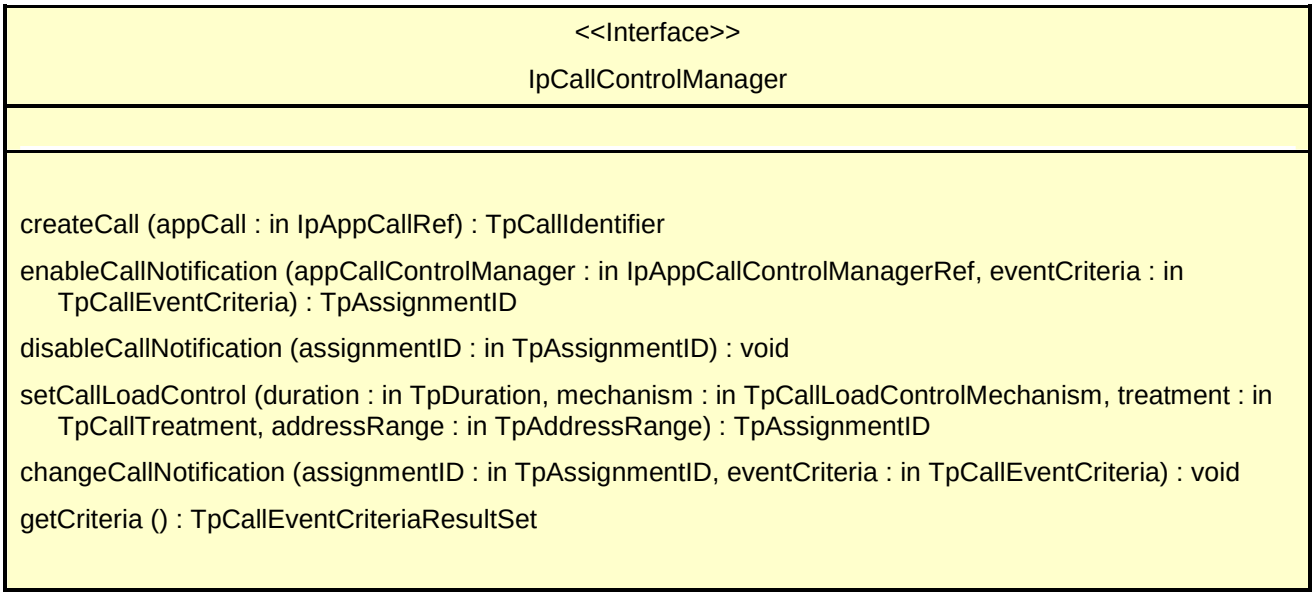
Change in Clause 6.1

6.1 Interface Class IpCallControlManager

Inherits from: IpService

This interface is the 'service manager' interface for the Generic Call Control Service. The generic call control manager interface provides the management functions to the generic call control service. The application programmer can use this interface to provide overload control functionality, create call objects and to enable or disable call-related event notifications.

This interface shall be implemented by a Generic Call Control SCF. As a minimum requirement either the createCall() method shall be implemented, or the enableCallNotification() and disableCallNotification() methods shall be implemented.



6.1.1 Method createCall()

This method is used to create a new call object.

Call back reference:

An IpAppCallControlManager should already have been passed to the IpCallControlManager, otherwise the call control will not be able to report a callAborted() to the application. The application should invoke setCallback() prior to createCall if it wishes to ensure this.

Returns callReference: Specifies the interface reference and sessionID of the call created.

Parameters

`appCall : in IpAppCallRef`

Specifies the application interface for callbacks from the call created.

Returns

TpCallIdentifier

Raises

TpCommonExceptions, P_INVALID_INTERFACE_TYPE

6.1.2 Method enableCallNotification()

This method is used to enable call notifications so that events can be sent to the application. This is the first step an application has to do to get initial notification of calls happening in the network. When such an event happens, the application will be informed by callEventNotify(). In case the application is interested in other events during the context of a particular call session it has to use the routeReq() method on the call object. The application will get access to the call object when it receives the callEventNotify(). (Note that the enableCallNotification() is not applicable if the call is setup by the application).

The enableCallNotification method is purely intended for applications to indicate their interest to be notified when certain call events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a call is made to any number starting with 800.

If some application already requested notifications with criteria that overlap the specified criteria, the request is refused with P_GCCS_INVALID_CRITERIA. The criteria are said to overlap if both originating and terminating ranges overlap and the same number plan is used and the same CallNotificationType is used.

If a notification is requested by an application with the monitor mode set to notify, then there is no need to check the rest of the criteria for overlapping with any existing request as the notify mode does not allow control on a call to be passed over. Only one application can place an interrupt request if the criteria overlaps.

If a notification is requested by an application with an event type that is mutually exclusive compared to existing requested event types, then there is no need to check against the rest of the criteria for overlap. An example could be one application that trigger on "user busy" together with another application that trigger on "answer" - both requests should be allowed as only one can occur on the same call or session.

The overlap criteria have been defined to prevent multiple points of control, leading to possible interaction problems in networks that have no multi service support. Notice that dynamic aspects cannot be taken into account in the overlap criteria check. Therefore where dynamic event arming from an application causes a persistent control relationship it can prevent other applications to be invoked in the case single point of application control applies in the network.

However, the criteria check for overlap may as a network option be overruled by Multi Service networks allowing more services or applications to gain control of the same call or session at the same point in time. Refer to Call Control Common Definitions subpart of this specification (TS 29.198-4-1) for further details on application control over a call or session.

Setting the callback reference:

The call back reference can be registered either in a) enableCallNotification() or b) explicitly with a separate setCallback() method depending on how the application provides its callback reference.

Case a:

From an efficiency point of view the enableCallNotification() with explicit immediate registration (no "Null" value) of call back reference may be the preferred method.

Case b:

The enableCallNotfication() with no call back reference ("Null" value) is used where (e.g. due to distributed application logic) the call back reference is provided subsequently in a setCallback().

In case the enableCallNotification() contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback(). See example in 6.1.6

Set additional callback:

If the same application requests two notifications with exactly the same criteria but different callback references, the second callback will be treated as an additional callback. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. See example in 4.1.

Returns assignmentID: Specifies the ID assigned by the generic call control manager interface for this newly-enabled event notification.

Parameters

appCallControlManager : in IpAppCallControlManagerRef

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

eventCriteria : in TpCallEventCriteria

Specifies the event specific criteria used by the application to define the event required. Only events that meet these criteria are reported. Examples of events are "incoming call attempt reported by network", "answer", "no answer", "busy". Individual addresses or address ranges may be specified for destination and/or origination.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_CRITERIA, P_INVALID_INTERFACE_TYPE, P_INVALID_EVENT_TYPE

End of Change in Clause 6.1
End Of Document

CR-Form-v7

CHANGE REQUEST

⌘ 29.198-02 CR 048 ⌘rev - ⌘ Current version: 6.2.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network ☐ Core Network X

Title:	⌘	Correct the Exception Hierarchy Annex to include new exceptions used in MultiMediaMessaging SCF
Source:	⌘	CN5 Ultan Mulligan, ETSI PTCC
Work item code:	⌘	OSA3
Date:	⌘	12/08/2004
Category:	⌘	**F**
Release:	⌘	*REL-6*

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)

Reason for change:	⌘	New exceptions have been introduced in the new MultiMediaMessaging SCF, in part 15. All exceptions are described hierachially in an informative annex in part 2, as this hierarchy is used when building the Java code for each part. This annex was not updated at the time of introducing the new exceptions.
Summary of change:	⌘	Include new hierarchy diagrams which include the new exceptions in the correct hierarchy. Introduce a new abstract exception TpMultiMediaMessagingException.
Consequences if not approved:	⌘	The exception hierarchy in part 2 will not reflect the hierarchy used in the Java code for MultiMediaMessaging. By not keeping part 2 up to date, we introduce inconsistancy across the specifications, and this can lead to confusion for developers.

Clauses affected:	⌘	Annex D Exception Hierarchy

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

Change in Annex D

Annex D (normative): Exception Hierarchy

This clause arranges the OSA exceptions as a set of hierarchies that, depending upon the technology realisation, may or may not be utilised to simplify software developers' code.

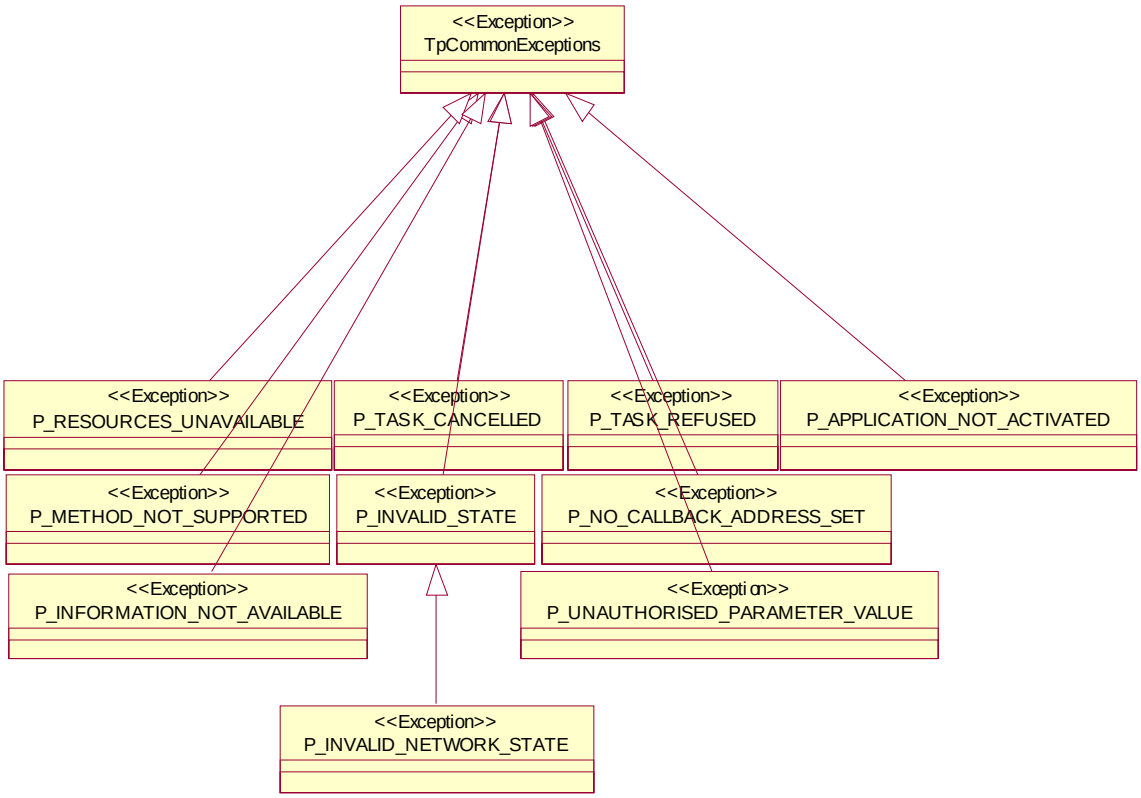
If the exception hierarchy is used in a particular realisation, the following lists all the OSA abstract exceptions:

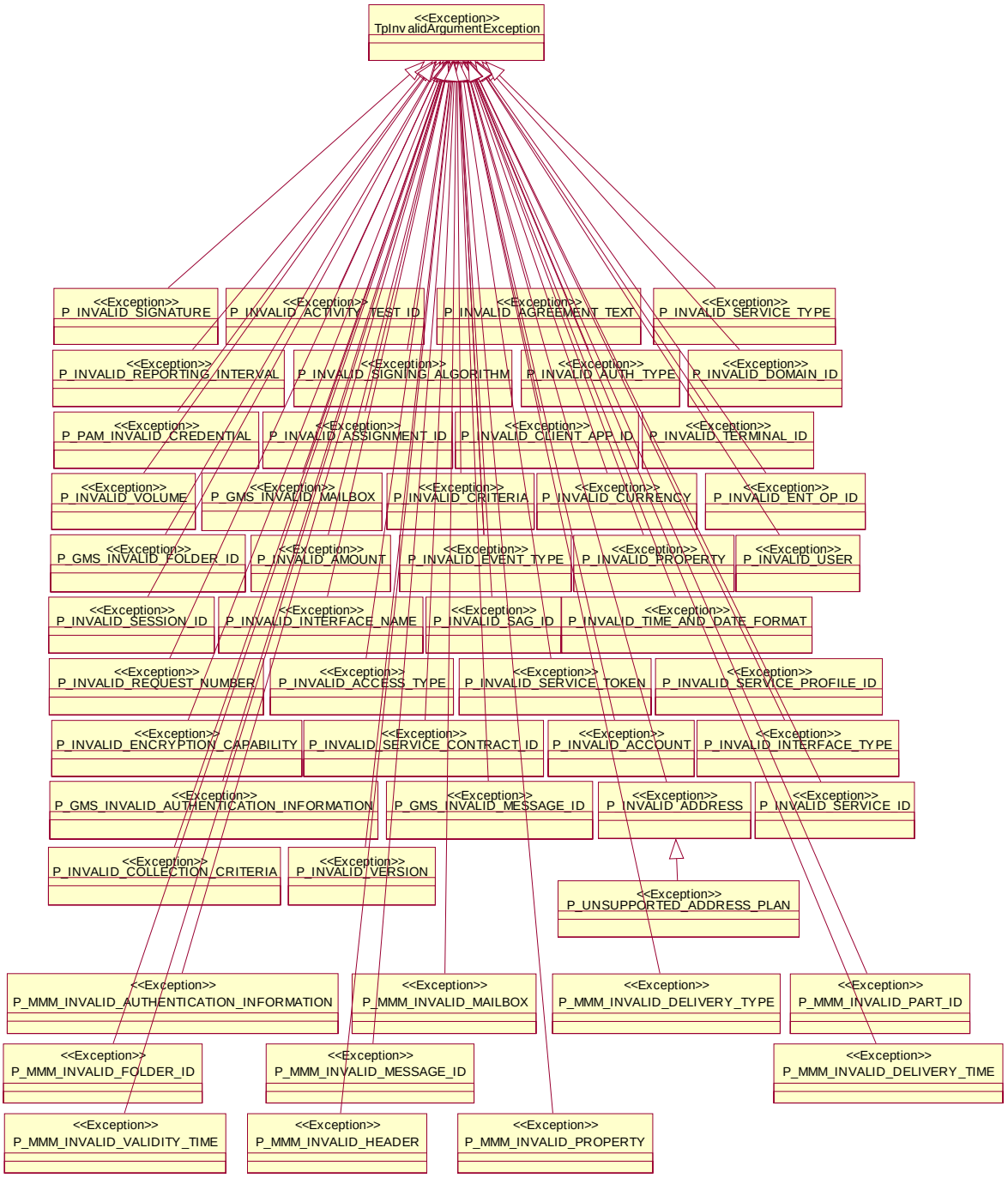
- TpCommonExceptions
- TpInvalidArgumentException
- TpDataSessionException
- TpAccountException
- TpConnectivityException
- TpFrameworkException
- TpMobilityException
- TpMessagingException
- TpPamException
- TpPolicyException
- TpMultiMediaMessagingException

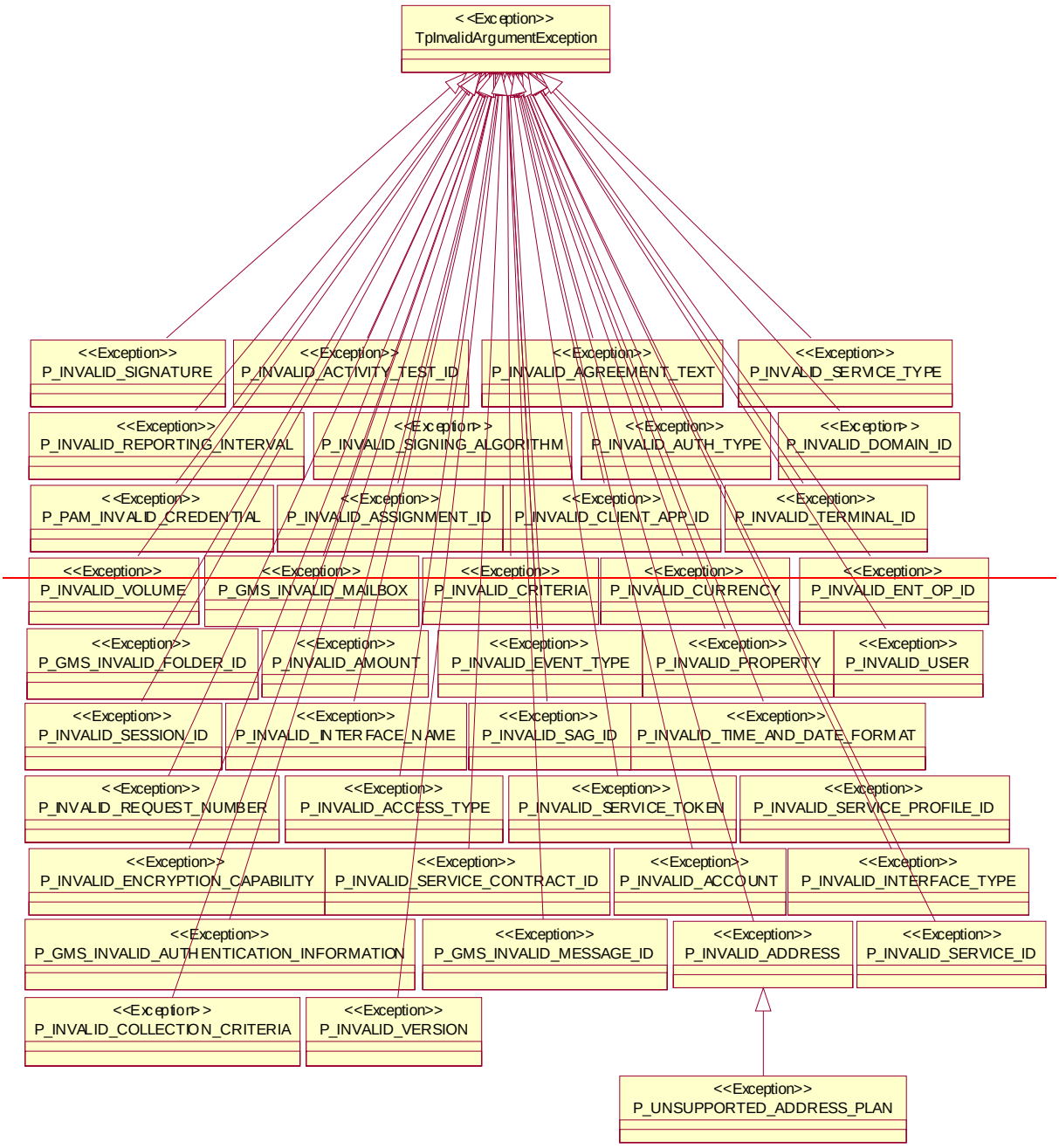
If the exception hierarchy is being used in a particular realisation, a software developer will have the option to catch these abstract exceptions and/or the detailed exceptions which extend them.

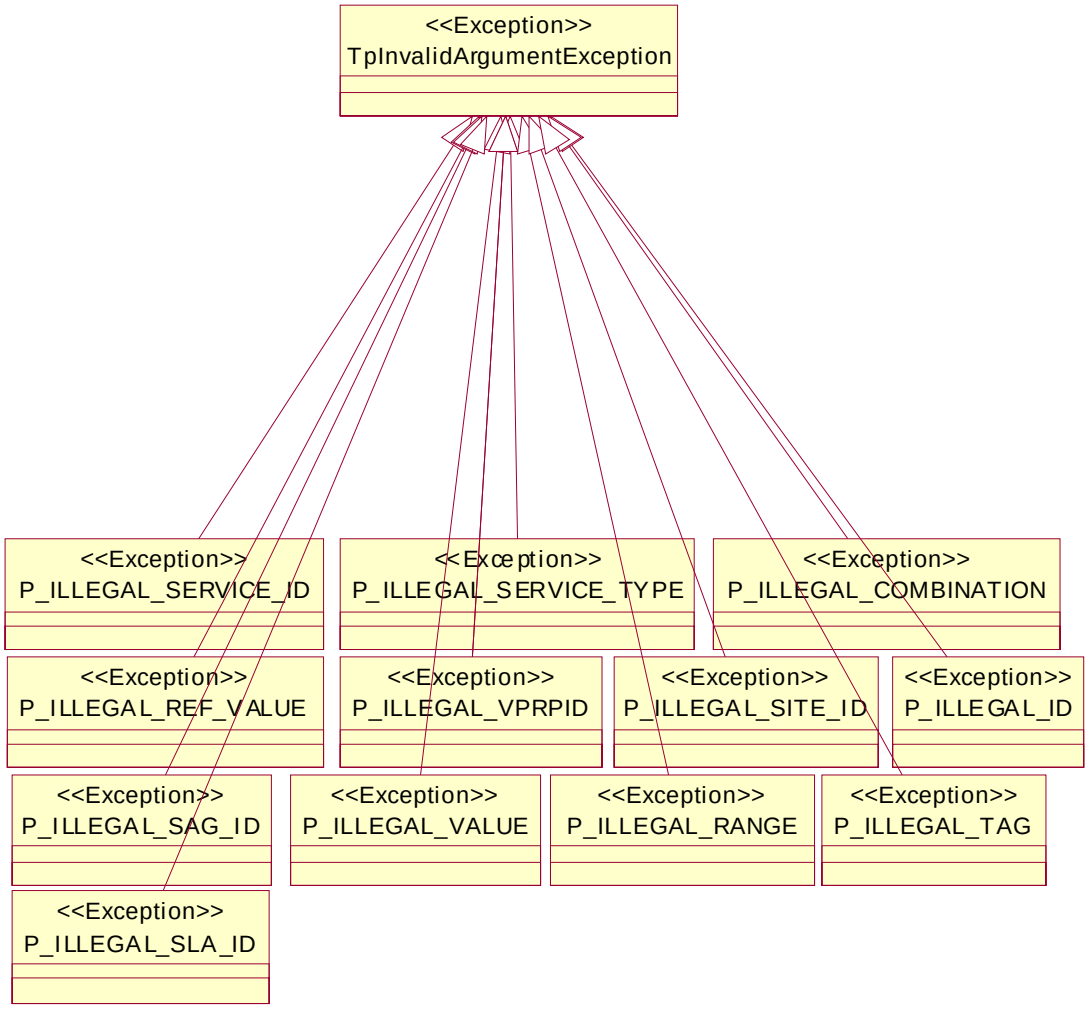
The following diagrams show all the OSA detailed exceptions, and how they relate to the abstract exceptions shown previously.

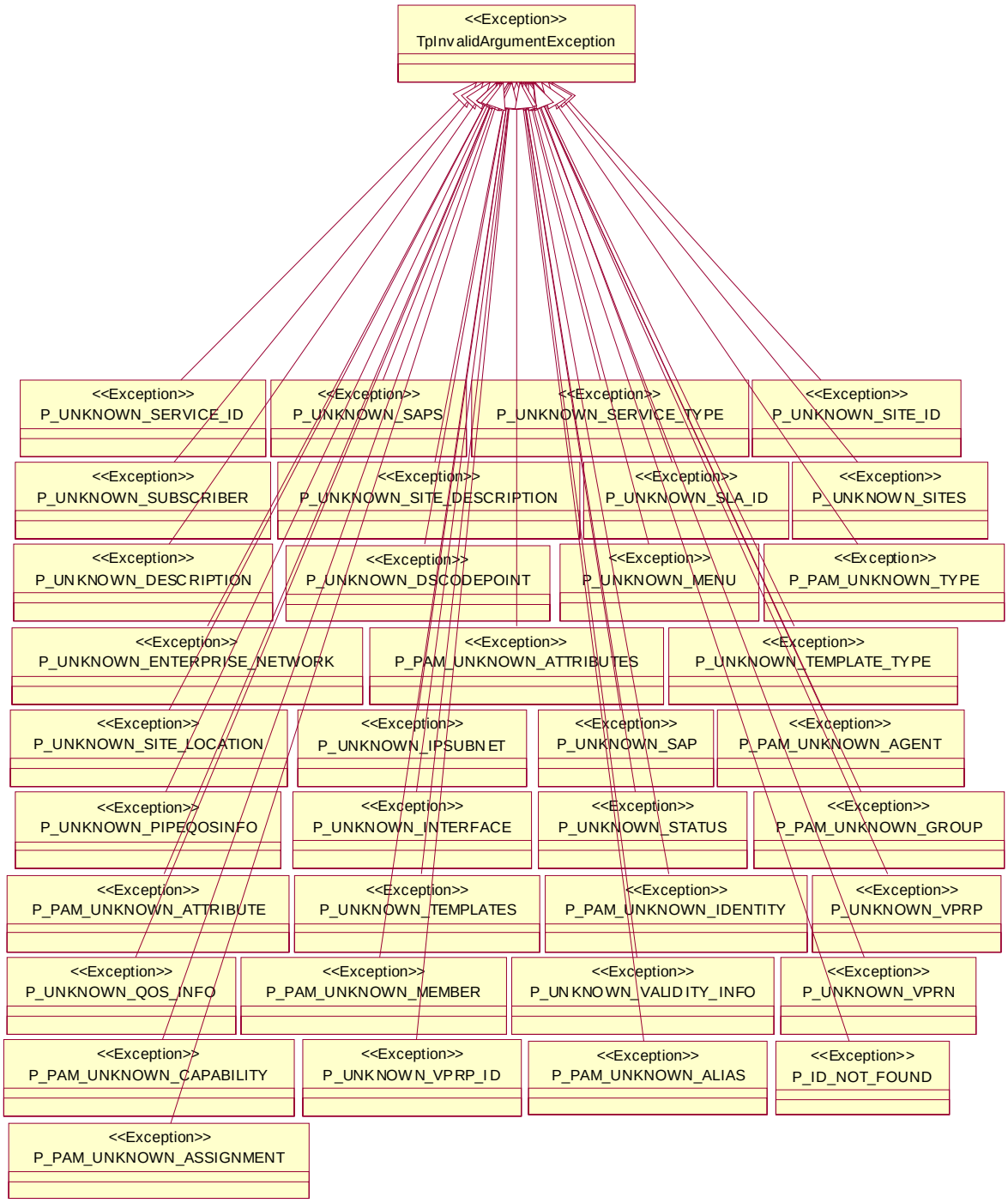
It should be noted that for those OSA methods that raise TpCommonExceptions, the P_RESOURCES_UNAVAILABLE, P_TASK_CANCELLED, P_TASK_REFUSED, P_METHOD_NOT_SUPPORTED, P_INVALID_STATE and P_NO_CALLBACK_ADDRESS_SET detailed exceptions should be raised individually in the method signature. The software developer will thus have the option of catching them individually or catching the TpCommonExceptions abstract exception.

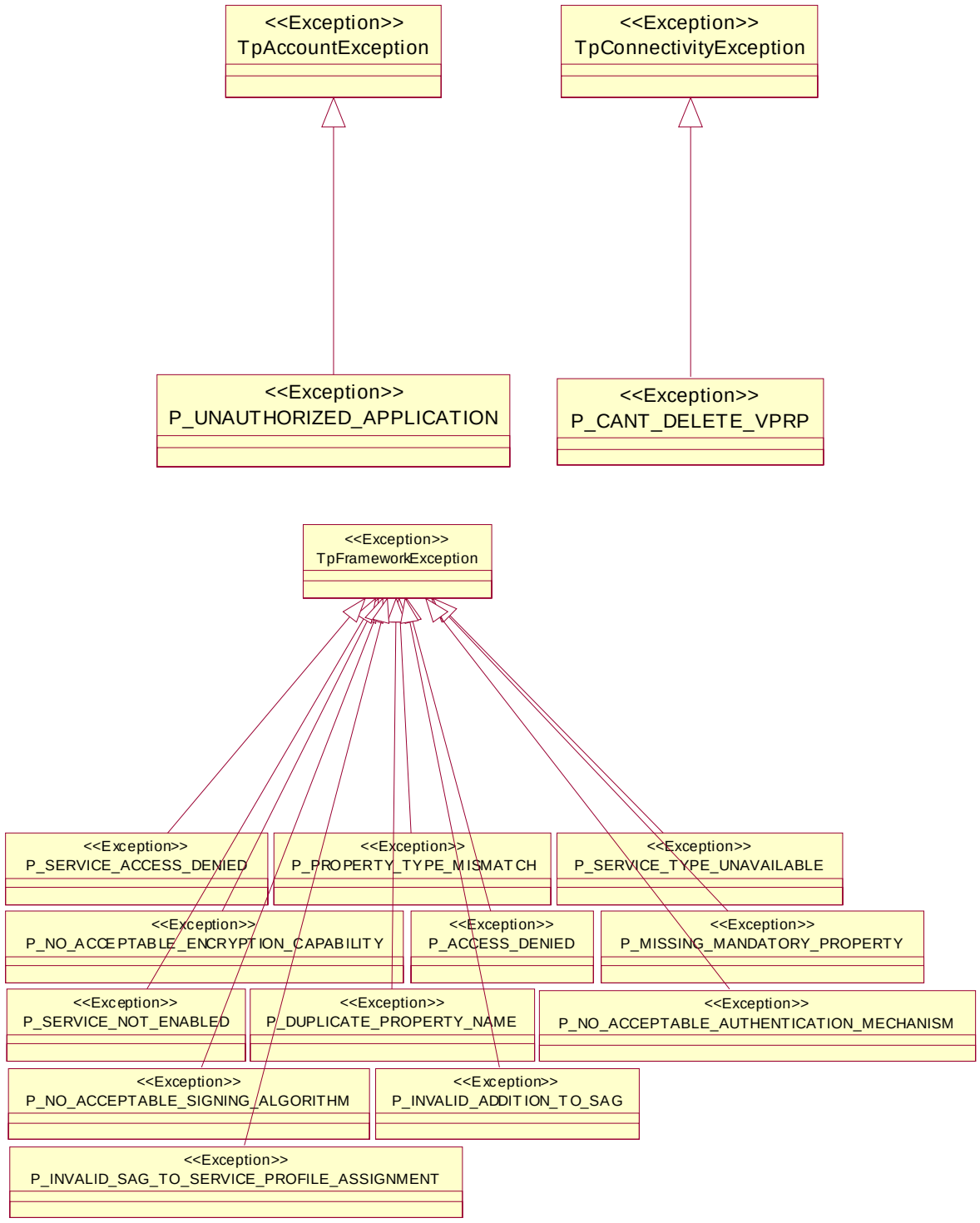

<<Exception>>
TpCommonExceptions
<<Exception>>
P_RESOURCES_UNAVAILABLE
<<Exception>>
P_TASK_CANCELLED
<<Exception>>
P_TASK_REFUSED
<<Exception>>
P_APPLICATION_NOT_ACTIVATED
<<Exception>>
P_METHOD_NOT_SUPPORTED
<<Exception>>
P_INVALID_STATE
<<Exception>>
P_NO_CALLBACK_ADDRESS_SET
<<Exception>>
P_INFORMATION_NOT_AVAILABLE
<<Exception>>
P_UNAUTHORISED_PARAMETER_VALUE
<<Exception>>
P_INVALID_NETWORK_STATE

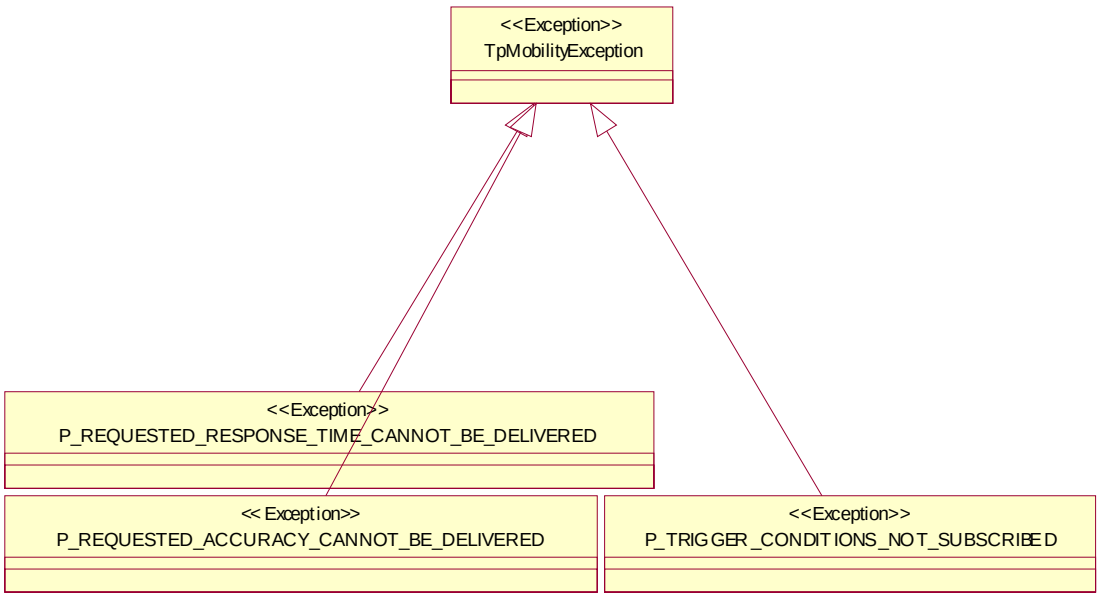

<<Exception>>
TpInvalidArgumentException
<<Exception>>
P_INVALID_SIGNATURE
<<Exception>>
P_INVALID_ACTIVITY_TEST_ID
<<Exception>>
P_INVALID_AGREEMENT_TEXT
<<Exception>>
P_INVALID_SERVICE_TYPE
<<Exception>>
P_INVALID_REPORTING_INTERVAL
<<Exception>>
P_INVALID_SIGNING_ALGORITHM
<<Exception>>
P_INVALID_AUTH_TYPE
<<Exception>>
P_INVALID_DOMAIN_ID
<<Exception>>
P_PAM_INVALID_CREDENTIAL
<<Exception>>
P_INVALID_ASSIGNMENT_ID
<<Exception>>
P_INVALID_CLIENT_APP_ID
<<Exception>>
P_INVALID_TERMINAL_ID
<<Exception>>
P_INVALID_VOLUME
<<Exception>>
P_GMS_INVALID_MAILBOX
<<Exception>>
P_INVALID_CRITERIA
<<Exception>>
P_INVALID_CURRENCY
<<Exception>>
P_INVALID_ENT_OP_ID
<<Exception>>
P_GMS_INVALID_FOLDER_ID
<<Exception>>
P_INVALID_AMOUNT
<<Exception>>
P_INVALID_EVENT_TYPE
<<Exception>>
P_INVALID_PROPERTY
<<Exception>>
P_INVALID_USER
<<Exception>>
P_INVALID_SESSION_ID
<<Exception>>
P_INVALID_INTERFACE_NAME
<<Exception>>
P_INVALID_SAG_ID
<<Exception>>
P_INVALID_TIME_AND_DATE_FORMAT
<<Exception>>
P_INVALID_REQUEST_NUMBER
<<Exception>>
P_INVALID_ACCESS_TYPE
<<Exception>>
P_INVALID_SERVICE_TOKEN
<<Exception>>
P_INVALID_SERVICE_PROFILE_ID
<<Exception>>
P_INVALID_ENCRYPTION_CAPABILITY
<<Exception>>
P_INVALID_SERVICE_CONTRACT_ID
<<Exception>>
P_INVALID_ACCOUNT
<<Exception>>
P_INVALID_INTERFACE_TYPE
<<Exception>>
P_GMS_INVALID_AUTHENTICATION_INFORMATION
<<Exception>>
P_GMS_INVALID_MESSAGE_ID
<<Exception>>
P_INVALID_ADDRESS
<<Exception>>
P_INVALID_SERVICE_ID
<<Exception>>
P_INVALID_COLLECTION_CRITERIA
<<Exception>>
P_INVALID_VERSION
<<Exception>>
P_UNSUPPORTED_ADDRESS_PLAN
<<Exception>>
P_MMM_INVALID_AUTHENTICATION_INFORMATION
<<Exception>>
P_MMM_INVALID_MAILBOX
<<Exception>>
P_MMM_INVALID_DELIVERY_TYPE
<<Exception>>
P_MMM_INVALID_PART_ID
<<Exception>>
P_MMM_INVALID_FOLDER_ID
<<Exception>>
P_MMM_INVALID_MESSAGE_ID
<<Exception>>
P_MMM_INVALID_DELIVERY_TIME
<<Exception>>
P_MMM_INVALID_VALIDITY_TIME
<<Exception>>
P_MMM_INVALID_HEADER
<<Exception>>
P_MMM_INVALID_PROPERTY

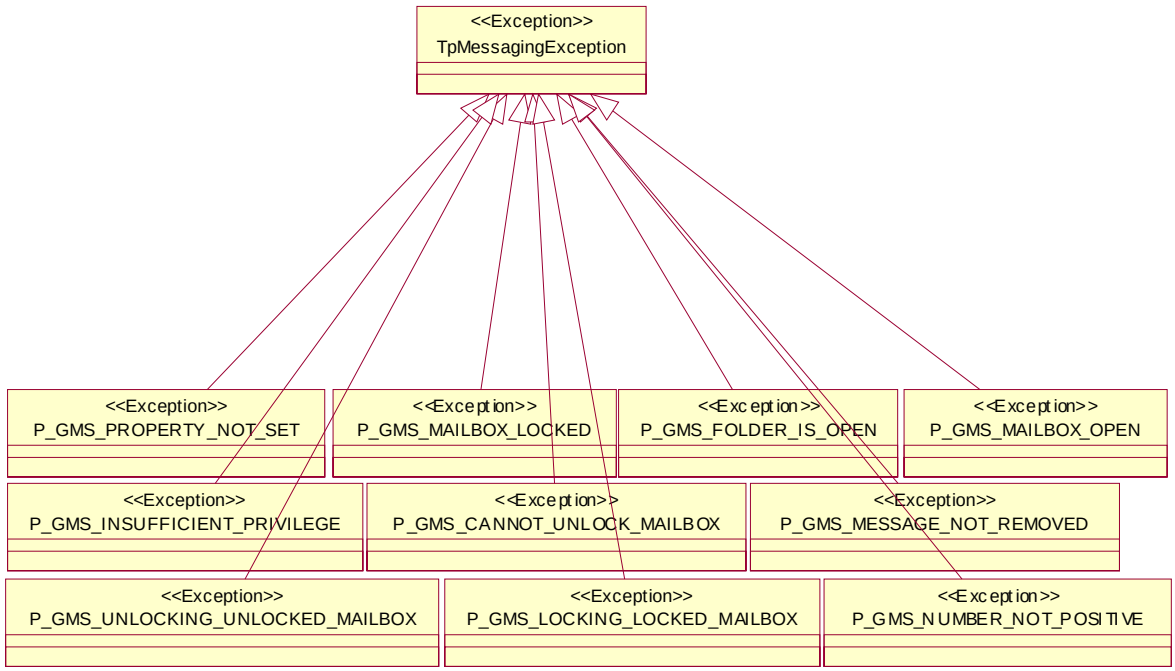

<<Exception>>
TpInvalidArgumentException
<<Exception>>
P_INVALID_SIGNATURE
<<Exception>>
P_INVALID_ACTIVITY_TEST_ID
<<Exception>>
P_INVALID_AGREEMENT_TEXT
<<Exception>>
P_INVALID_SERVICE_TYPE
<<Exception>>
P_INVALID_REPORTING_INTERVAL
<<Exception>>
P_INVALID_SIGNING_ALGORITHM
<<Exception>>
P_INVALID_AUTH_TYPE
<<Exception>>
P_INVALID_DOMAIN_ID
<<Exception>>
P_PAM_INVALID_CREDENTIAL
<<Exception>>
P_INVALID_ASSIGNMENT_ID
<<Exception>>
P_INVALID_CLIENT_APP_ID
<<Exception>>
P_INVALID_TERMINAL_ID
<<Exception>>
P_INVALID_VOLUME
<<Exception>>
P_GMS_INVALID_MAILBOX
<<Exception>>
P_INVALID_CRITERIA
<<Exception>>
P_INVALID_CURRENCY
<<Exception>>
P_INVALID_ENT_OP_ID
<<Exception>>
P_GMS_INVALID_FOLDER_ID
<<Exception>>
P_INVALID_AMOUNT
<<Exception>>
P_INVALID_EVENT_TYPE
<<Exception>>
P_INVALID_PROPERTY
<<Exception>>
P_INVALID_USER
<<Exception>>
P_INVALID_SESSION_ID
<<Exception>>
P_INVALID_INTERFACE_NAME
<<Exception>>
P_INVALID_SAG_ID
<<Exception>>
P_INVALID_TIME_AND_DATE_FORMAT
<<Exception>>
P_INVALID_REQUEST_NUMBER
<<Exception>>
P_INVALID_ACCESS_TYPE
<<Exception>>
P_INVALID_SERVICE_TOKEN
<<Exception>>
P_INVALID_SERVICE_PROFILE_ID
<<Exception>>
P_INVALID_ENCRYPTION_CAPABILITY
<<Exception>>
P_INVALID_SERVICE_CONTRACT_ID
<<Exception>>
P_INVALID_ACCOUNT
<<Exception>>
P_INVALID_INTERFACE_TYPE
<<Exception>>
P_GMS_INVALID_AUTHENTICATION_INFORMATION
<<Exception>>
P_GMS_INVALID_MESSAGE_ID
<<Exception>>
P_INVALID_ADDRESS
<<Exception>>
P_INVALID_SERVICE_ID
<<Exception>>
P_INVALID_COLLECTION_CRITERIA
<<Exception>>
P_INVALID_VERSION
<<Exception>>
P_UNSUPPORTED_ADDRESS_PLAN

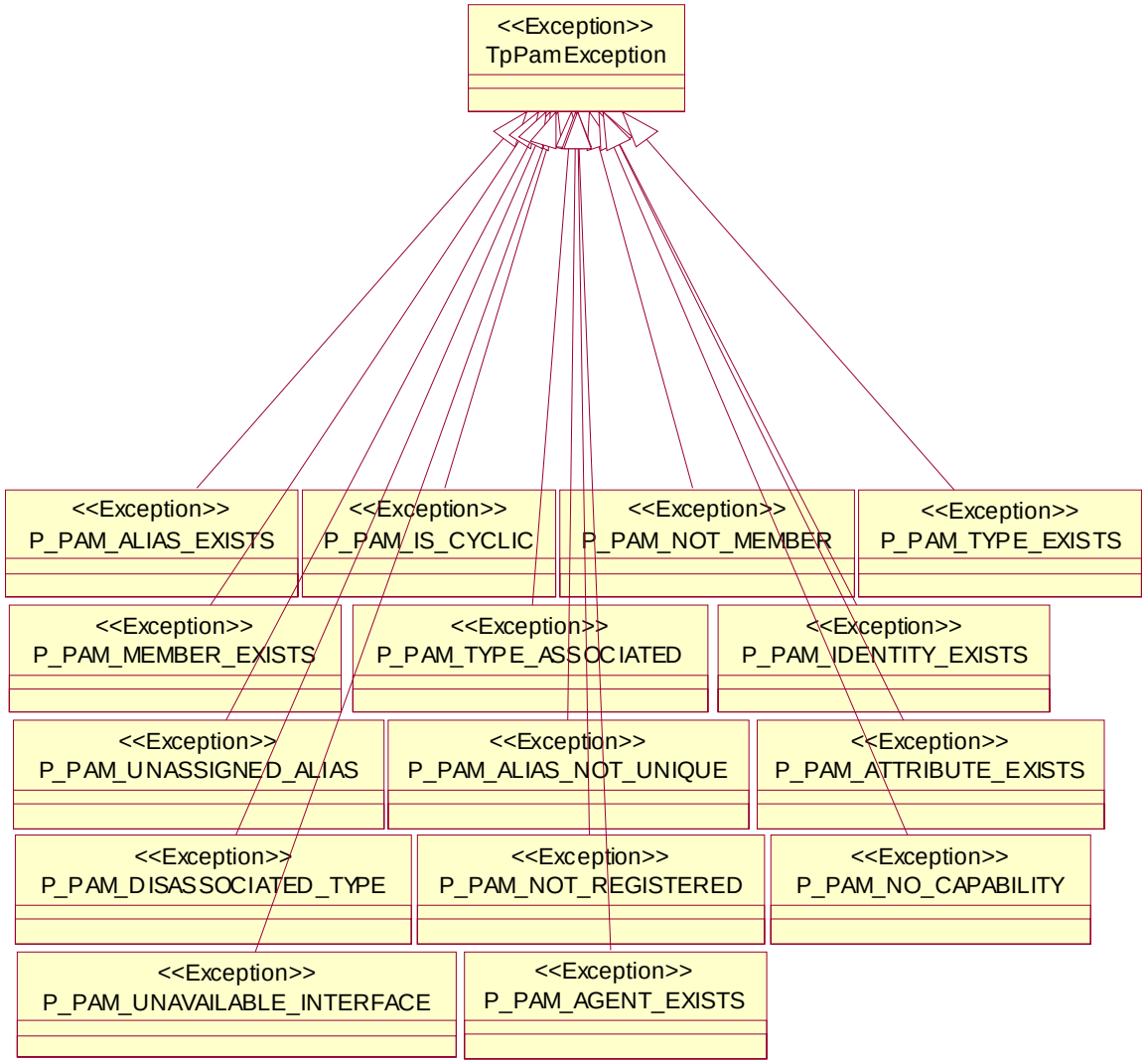

<<Exception>>
TpInvalidArgumentException
<<Exception>>
P_ILLEGAL_SERVICE_ID
<<Exception>>
P_ILLEGAL_SERVICE_TYPE
<<Exception>>
P_ILLEGAL_COMBINATION
<<Exception>>
P_ILLEGAL_REF_VALUE
<<Exception>>
P_ILLEGAL_VPRPID
<<Exception>>
P_ILLEGAL_SITE_ID
<<Exception>>
P_ILLEGAL_ID
<<Exception>>
P_ILLEGAL_SAG_ID
<<Exception>>
P_ILLEGAL_VALUE
<<Exception>>
P_ILLEGAL_RANGE
<<Exception>>
P_ILLEGAL_TAG
<<Exception>>
P_ILLEGAL_SLA_ID

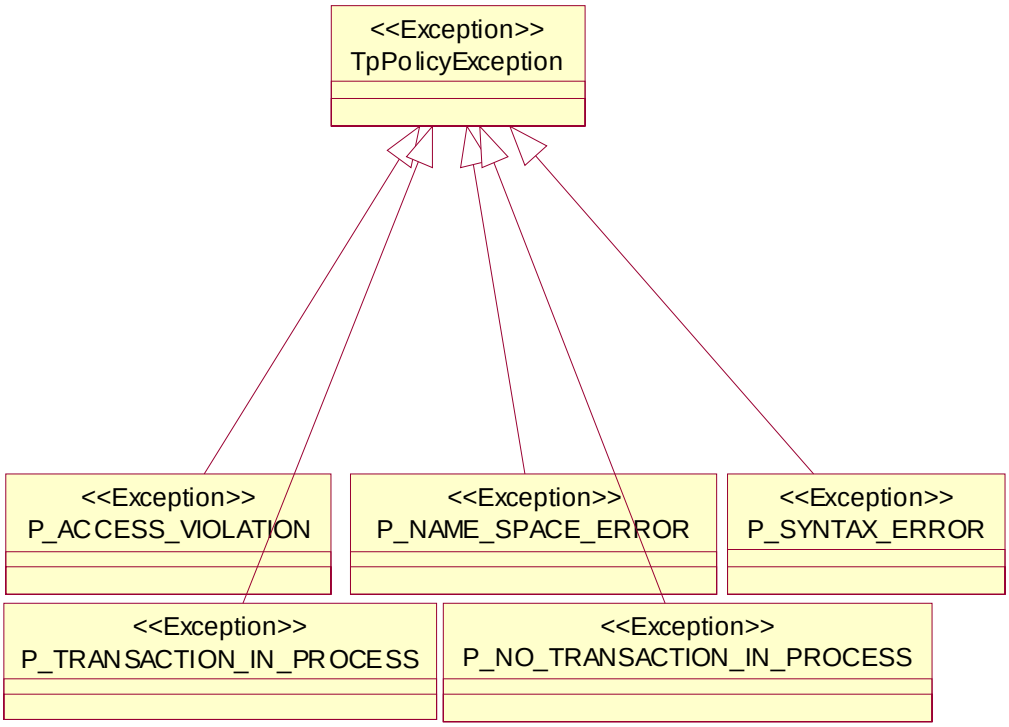

<<Exception>>
TpInvalidArgumentException
<<Exception>>
P_UNKNOWN_SERVICE_ID
<<Exception>>
P_UNKNOWN_SAPS
<<Exception>>
P_UNKNOWN_SERVICE_TYPE
<<Exception>>
P_UNKNOWN_SITE_ID
<<Exception>>
P_UNKNOWN_SUBSCRIBER
<<Exception>>
P_UNKNOWN_SITE_DESCRIPTION
<<Exception>>
P_UNKNOWN_SLA_ID
<<Exception>>
P_UNKNOWN_SITES
<<Exception>>
P_UNKNOWN_DESCRIPTION
<<Exception>>
P_UNKNOWN_DSCODEPOINT
<<Exception>>
P_UNKNOWN_MENU
<<Exception>>
P_PAM_UNKNOWN_TYPE
<<Exception>>
P_UNKNOWN_ENTERPRISE_NETWORK
<<Exception>>
P_PAM_UNKNOWN_ATTRIBUTES
<<Exception>>
P_UNKNOWN_TEMPLATE_TYPE
<<Exception>>
P_UNKNOWN_SITE_LOCATION
<<Exception>>
P_UNKNOWN_IPSUBNET
<<Exception>>
P_UNKNOWN_SAP
<<Exception>>
P_PAM_UNKNOWN_AGENT
<<Exception>>
P_UNKNOWN_PIPEQOSINFO
<<Exception>>
P_UNKNOWN_INTERFACE
<<Exception>>
P_UNKNOWN_STATUS
<<Exception>>
P_PAM_UNKNOWN_GROUP
<<Exception>>
P_PAM_UNKNOWN_ATTRIBUTE
<<Exception>>
P_UNKNOWN_TEMPLATES
<<Exception>>
P_PAM_UNKNOWN_IDENTITY
<<Exception>>
P_UNKNOWN_VPRP
<<Exception>>
P_UNKNOWN_QOS_INFO
<<Exception>>
P_PAM_UNKNOWN_MEMBER
<<Exception>>
P_UNKNOWN_VALIDITY_INFO
<<Exception>>
P_UNKNOWN_VPRN
<<Exception>>
P_PAM_UNKNOWN_CAPABILITY
<<Exception>>
P_UNKNOWN_VPRP_ID
<<Exception>>
P_PAM_UNKNOWN_ALIAS
<<Exception>>
P_ID_NOT_FOUND
<<Exception>>
P_PAM_UNKNOWN_ASSIGNMENT

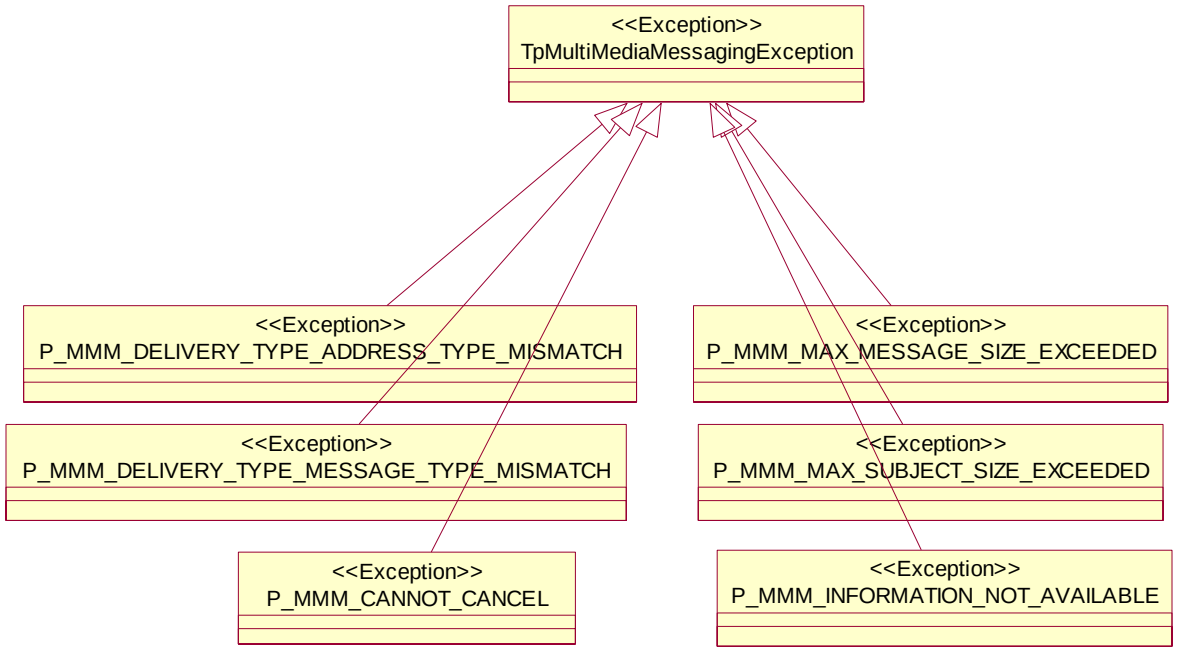

<<Exception>>
TpAccountException
<<Exception>>
TpConnectivityException
<<Exception>>
P_UNAUTHORIZED_APPLICATION
<<Exception>>
P_CANT_DELETE_VPRP
<<Exception>>
TpFrameworkException
<<Exception>>
P_SERVICE_ACCESS_DENIED
<<Exception>>
P_PROPERTY_TYPE_MISMATCH
<<Exception>>
P_SERVICE_TYPE_UNAVAILABLE
<<Exception>>
P_NO_ACCEPTABLE_ENCRYPTION_CAPABILITY
<<Exception>>
P_ACCESS_DENIED
<<Exception>>
P_MISSING_MANDATORY_PROPERTY
<<Exception>>
P_SERVICE_NOT_ENABLED
<<Exception>>
P_DUPLICATE_PROPERTY_NAME
<<Exception>>
P_NO_ACCEPTABLE_AUTHENTICATION_MECHANISM
<<Exception>>
P_NO_ACCEPTABLE_SIGNING_ALGORITHM
<<Exception>>
P_INVALID_ADDITION_TO_SAG
<<Exception>>
P_INVALID_SAG_TO_SERVICE_PROFILE_ASSIGNMENT


<<Exception>>
TpMobilityException
<<Exception>>
P_REQUESTED_RESPONSE_TIME_CANNOT_BE_DELIVERED
<< Exception>>
P_REQUESTED_ACCURACY_CANNOT_BE_DELIVERED
<<Exception>>
P_TRIGGER_CONDITIONS_NOT_SUBSCRIBED


<<Exception>>
TpMessagingException
<<Exception>>
P_GMS_PROPERTY_NOT_SET
<<Exception>>
P_GMS_MAILBOX_LOCKED
<<Exception>>
P_GMS_FOLDER_IS_OPEN
<<Exception>>
P_GMS_MAILBOX_OPEN
<<Exception>>
P_GMS_INSUFFICIENT_PRIVILEGE
<<Exception>>
P_GMS_CANNOT_UNLOCK_MAILBOX
<<Exception>>
P_GMS_MESSAGE_NOT_REMOVED
<<Exception>>
P_GMS_UNLOCKING_UNLOCKED_MAILBOX
<<Exception>>
P_GMS_LOCKING_LOCKED_MAILBOX
<<Exception>>
P_GMS_NUMBER_NOT_POSITIVE


<<Exception>>
TpPamException
<<Exception>>
P_PAM_ALIAS_EXISTS
<<Exception>>
P_PAM_IS_CYCLIC
<<Exception>>
P_PAM_NOT_MEMBER
<<Exception>>
P_PAM_TYPE_EXISTS
<<Exception>>
P_PAM_MEMBER_EXISTS
<<Exception>>
P_PAM_TYPE_ASSOCIATED
<<Exception>>
P_PAM_IDENTITY_EXISTS
<<Exception>>
P_PAM_UNASSIGNED_ALIAS
<<Exception>>
P_PAM_ALIAS_NOT_UNIQUE
<<Exception>>
P_PAM_ATTRIBUTE_EXISTS
<<Exception>>
P_PAM_DISASSOCIATED_TYPE
<<Exception>>
P_PAM_NOT_REGISTERED
<<Exception>>
P_PAM_NO_CAPABILITY
<<Exception>>
P_PAM_UNAVAILABLE_INTERFACE
<<Exception>>
P_PAM_AGENT_EXISTS


<<Exception>>
TpPolicyException
<<Exception>>
P_ACCESS_VIOLATION
<<Exception>>
P_NAME_SPACE_ERROR
<<Exception>>
P_SYNTAX_ERROR
<<Exception>>
P_TRANSACTION_IN_PROCESS
<<Exception>>
P_NO_TRANSACTION_IN_PROCESS



End of Change in Annex D