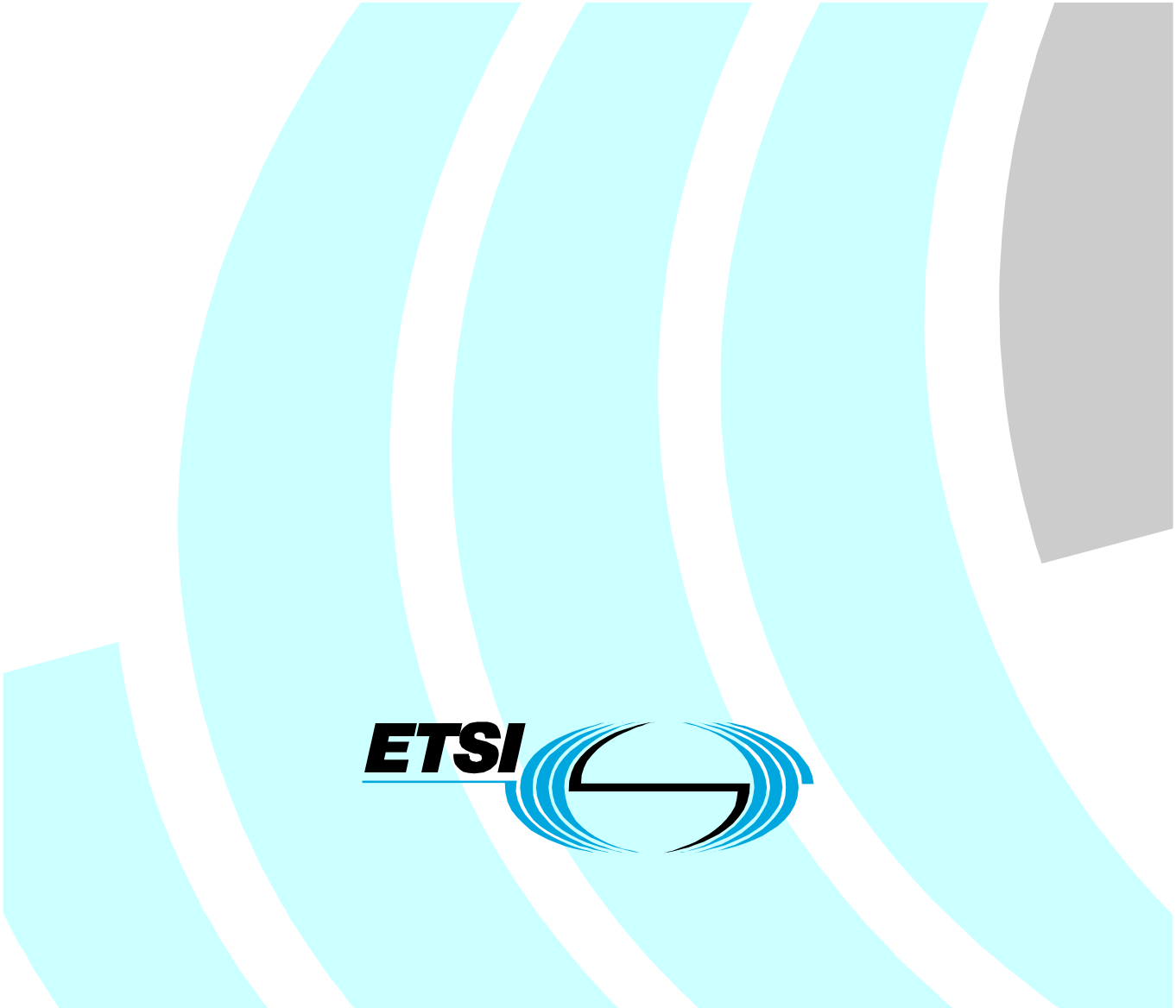
Final draft ETSI ES 202 388-14 V1.1.1 (2005-01)

ETSI Standard

Open Service Access (OSA); Application Programming Interface (API); Test Suite Structure and Test Purposes (TSS&TP); Part 14: Presence and Availability Management SCF; (Parlay 4)



Reference
DES/TISPAN-06004-14-OSA

Keywords
API, OSA, TSS&TP

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://webapp.etsi.org/IPR/home.asp).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 14 of a multi-part deliverable. Full details of the entire series can be found in part 1 [6].

To evaluate conformance of a particular implementation, it is necessary to have a set of test purposes to evaluate the dynamic behaviour of the Implementation Under Test (IUT). The specification containing those test purposes is called a Test Suite Structure and Test Purposes (TSS&TP) specification.

1 Scope

The present document provides the Test Suite Structure and Test Purposes (TSS&TP) specification for the Presence and Availability Management SCF of the Application Programming Interface (API) for Open Service Access (OSA) defined in ES 202 915-14 [1] in compliance with the relevant requirements, and in accordance with the relevant guidance given in ISO/IEC 9646-2 [4] and ETS 300 406 [5].

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

[1] ETSI ES 202 915-14: "Open Service Access (OSA); Application Programming Interface (API); Part 14: Presence and Availability Management SCF (Parlay 4)".

[2] ETSI ES 202 363: "Open Service Access (OSA); Application Programming Interface (API); Implementation Conformance Statement (ICS) proforma specification; (Parlay 4)".

[3] ISO/IEC 9646-1: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 1: General concepts".

[4] ISO/IEC 9646-2: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 2: Abstract Test Suite specification".

[5] ETSI ETS 300 406: "Methods for Testing and Specification (MTS); Protocol and profile conformance testing specifications; Standardization methodology".

[6] ETSI ES 202 388-1: "Open Service Access (OSA); Application Programming Interface (API); Test Suite Structure and Test Purposes (TSS&TP); Part 1: Overview; (Parlay 4)".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 915-14 [1], ISO/IEC 9646-1 [3], ISO/IEC 9646-2 [4] and the following apply:

abstract test case: Refer to ISO/IEC 9646-1 [3].

Abstract Test Method (ATM): Refer to ISO/IEC 9646-1 [3].

Abstract Test Suite (ATS): Refer to ISO/IEC 9646-1 [3].

Implementation Under Test (IUT): Refer to ISO/IEC 9646-1 [3].

Lower Tester (LT): Refer to ISO/IEC 9646-1 [3].

Implementation Conformance Statement (ICS): Refer to ISO/IEC 9646-1 [3].

ICS proforma: Refer to ISO/IEC 9646-1 [3].

Implementation eXtra Information for Testing (IXIT): Refer to ISO/IEC 9646-1 [3].

IXIT proforma: Refer to ISO/IEC 9646-1 [3].

Test Purpose (TP): Refer to ISO/IEC 9646-1 [3].

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

API	Application Programming Interface
ATM	Abstract Test Method
ATS	Abstract Test Suite
ICS	Implementation Conformance Statement
IUT	Implementation Under Test
IXIT	Implementation eXtra Information for Testing
LT	Lower Tester
OSA	Open Service Access
PAM	Presence and Availability Management
SCF	Service Capability Feature
TP	Test Purpose
TSS	Test Suite Structure

4 Test Suite Structure (TSS)

- Presence and Availability Management
  - PAM Access Service
  - PAM Event Handler

5 Test Purposes (TP)

5.1 Introduction

For each test requirement a TP is defined.

5.1.1 TP naming convention

TPs are numbered, starting at 01, within each group. Groups are organized according to the TSS. Additional references are added to identify the actual test suite (see table 1).

Table 1: TP identifier naming convention scheme

Identifier: <suite_id>_<group>_<nnn>		
<suite_id>	= SCG name:	"PAM" for **Presence and Availability Management SCF**
<group>	= group number:	two character field representing the group reference according to TSS
<nn>	= sequential number:	(01 to 99)

5.1.2 Source of TP definition

The TPs are based on ES 202 915-14 [1].

5.1.3 Test strategy

As the base standard ES 202 915-14 [1] contains no explicit requirements for testing, the TPs were generated as a result of an analysis of the base standard and the ICS specification ES 202 363 [2].

The TPs are only based on conformance requirements related to the externally observable behaviour of the IUT and are limited to conceivable situations to which a real implementation is likely to be faced (see ETS 300 406 [5]).

5.2 TPs for the Presence and Availability Management SCF

All ICS items referred to in this clause are as specified in ES 202 363 [2] unless indicated otherwise by another numbered reference.

All parameters specified in method calls are valid unless specified.

The procedures to trigger the SCF to call methods in the application are dependant on the underlying network architecture and are out of the scope of the present document. Those method calls are preceded by the words "Triggered action".

5.2.1 Presence and Availability Management, SCF side

5.2.1.1 PAM Access Service

5.2.1.1.1 IpPAMPresenceAvailabilityManager

Precondition: IpPAMPresenceAvailabilityManager supported.

Test PAM_AS_01

Summary: **IpPAMPresenceAvailabilityManager**, obtainInterface and getAuthToken successful.

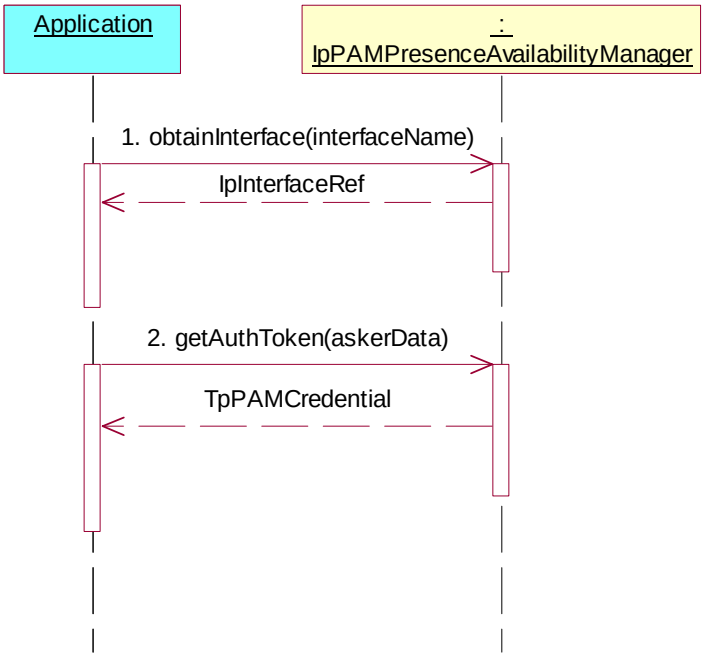
Reference: ES 202 915-14 [1], clause 8.2.1.

Preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMPresenceAvalibilityManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call obtainInterface()
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call getAuthToken()
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Preamble_getAuthToken_AS

Summary: Preamble to obtain the authentication token.

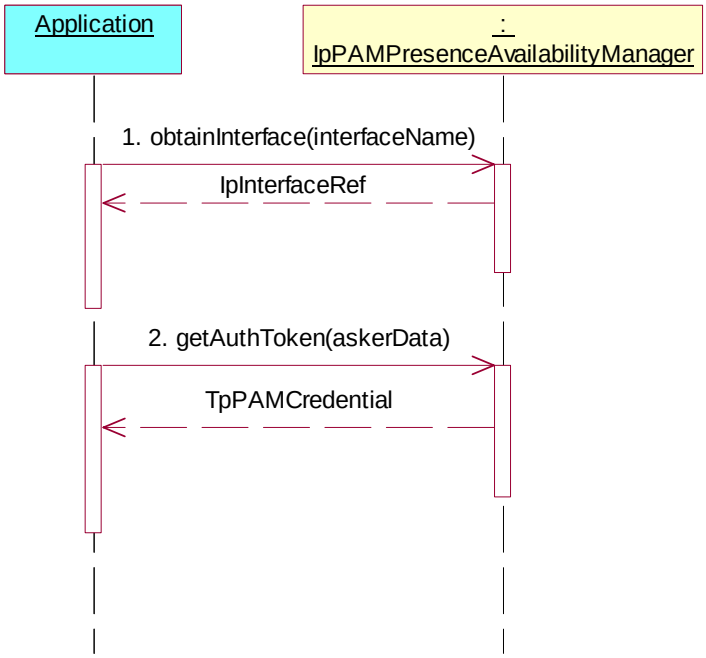
Reference: ES 202 915-14 [1], clause 8.2.

Pre-preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMPresenceAvalibilityManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call obtainInterface()
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call getAuthToken()
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Test PAM_AS_02

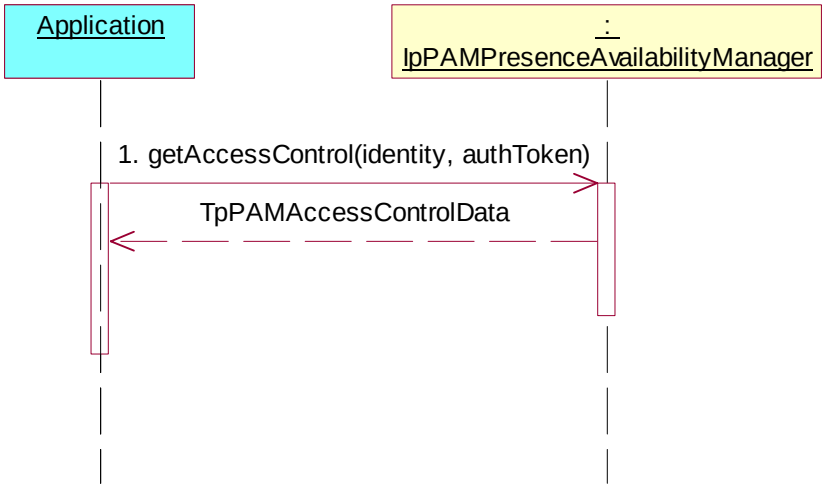
Summary: **IpPAMPresenceAvailabilityManager**, getAccessControl successful.

Reference: ES 202 915-14 [1], clause 8.2.1.3.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call getAccessControl()
   Parameters: identity, authToken as obtained in Preamble
   Check: valid value of TpPAMAccessControlData is returned



Test PAM_AS_03

Summary: **IpPAMPresenceAvailabilityManager**, getAccessControl and setAccessControl successful.

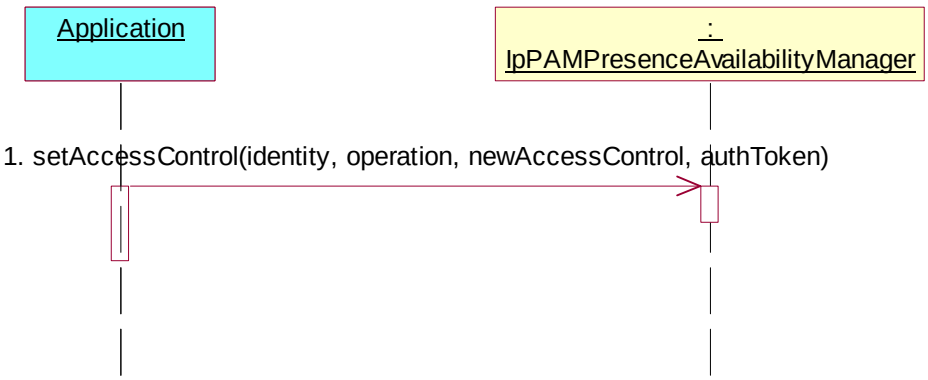
Reference: ES 202 915-14 [1], clause 8.2.1.4.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call setAccessControl()
   Parameters: identity, operation, newAccessControl, authToken as obtained in Preamble
   Check: no exception returned

NOTE: setAccessControl() can be used in conjunction with getAccessControl().



5.2.1.1.2 IpPAMIdentityPresence

Precondition: IpPAMIdentityPresence supported.

Test PAM_AS_04

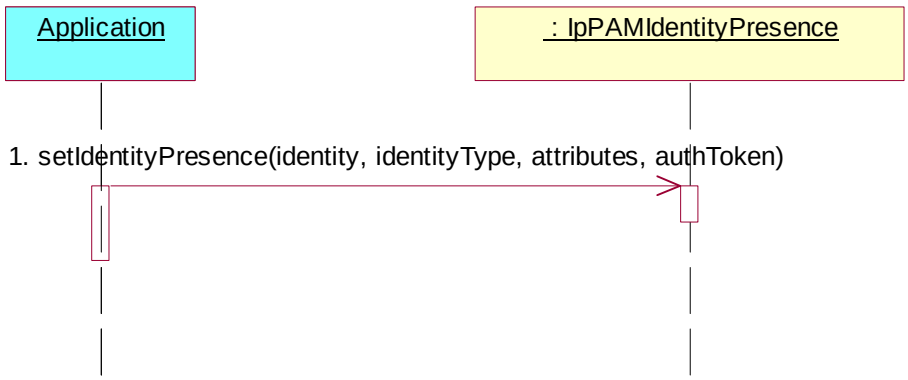
Summary: **IpPAMIdentityPresence**, setIdentityPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.2.1.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setIdentityPresence()**
   Parameters: identity, identityType, attributes, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_05

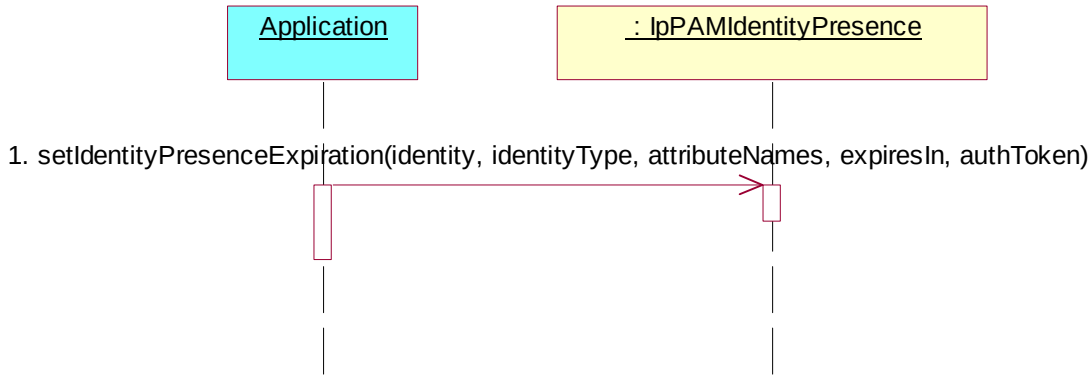
Summary: **IpPAMIdentityPresence**, setIdentityPresenceExpiration successful.

Reference: ES 202 915-14 [1], clause 8.2.2.2.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setIdentityPresenceExpiration()**
   Parameters: identity, identityType, attributeNames, expiresIn, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_06

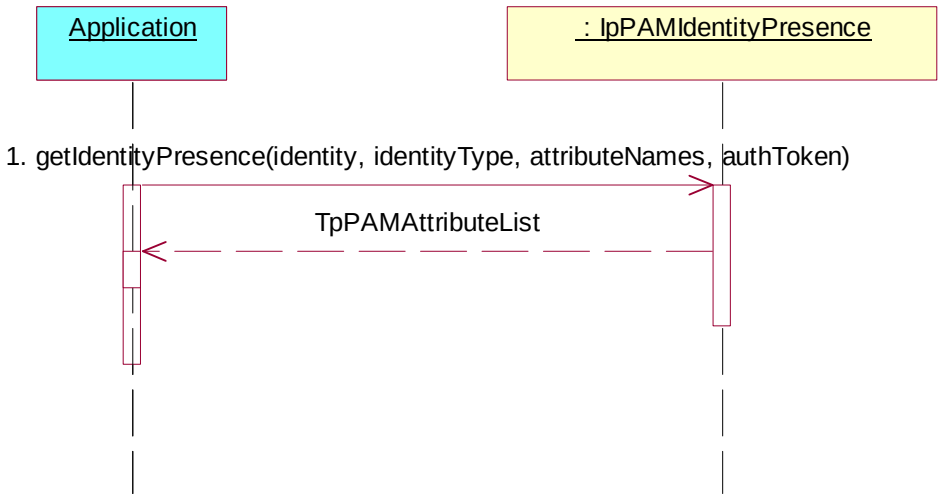
Summary: **IpPAMIdentityPresence**, getIdentityPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.2.3.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getIdentityPresence ()**
   Parameters: identity, identityType, attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned



5.2.1.1.3 IpPAMAvailability

Precondition: IpPAMAvailability supported.

Test PAM_AS_07

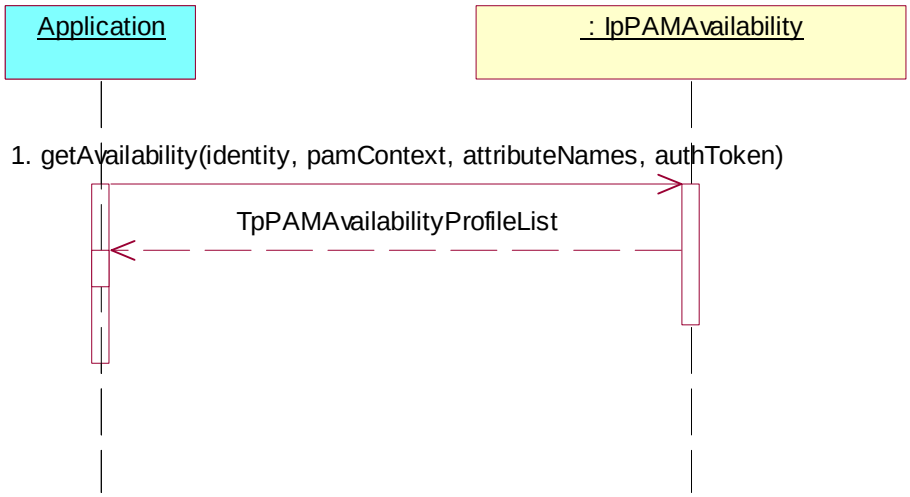
Summary: **IpPAMAvailability**, getAvailability successful.

Reference: ES 202 915-14 [1], clause 8.2.3.1.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getAvailability()**
   Parameters: identity, pamContext, attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAvailabilityProfileList is returned



Test PAM_AS_08

Summary: **IpPAMAvailability**, getAvailability successful.

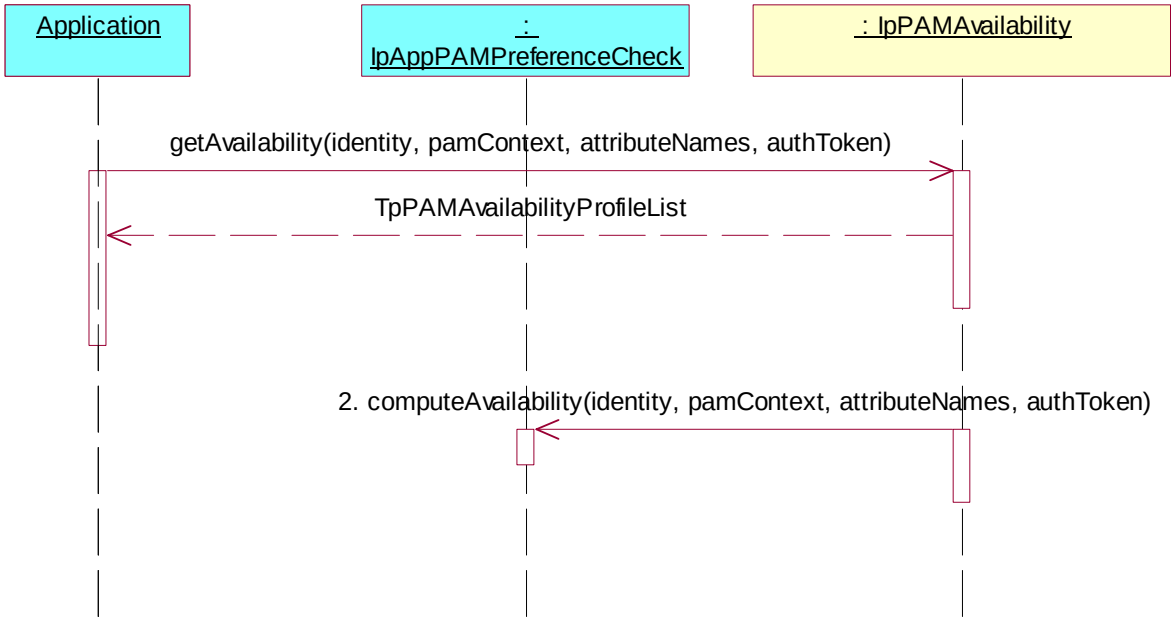
Reference: ES 202 915-14 [1], clause 8.2.3.1.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getAvailability()**
   Parameters: identity, pamContext, attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAvailabilityProfileList is returned

2. Triggered action: cause IUT to call **computeAvailability()** method on the tester's (Application) **IpAppPAMPreferenceCheck** interface.
   Parameters: identity, pamContext, attributeNames, authToken.



Test PAM_AS_09

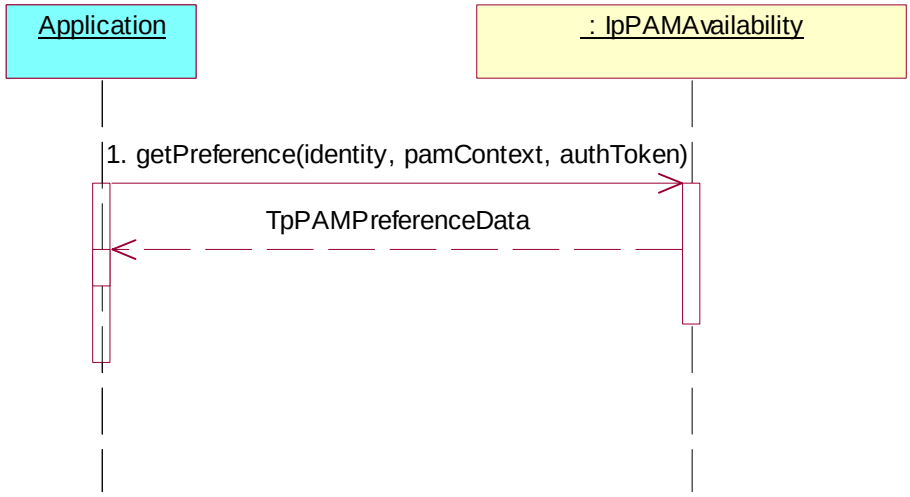
Summary: **IpPAMAvailability**, getPreference successful.

Reference: ES 202 915-14 [1], clause 8.2.3.2.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getPreference()**
   Parameters: identity, pamContext, authToken as obtained in Preamble
   Check: valid value of TpPAMPreferenceData is returned



Test PAM_AS_10

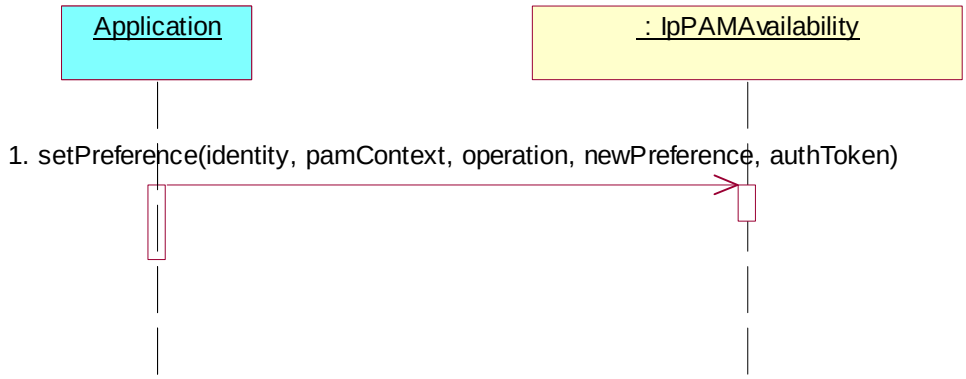
Summary: **IpPAMAvailability**, setPreference successful.

Reference: ES 202 915-14 [1], clause 8.2.3.3.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setPreference()**
   Parameters: identity, pamContext, operation, newPreference, authToken as obtained in Preamble
   Check: valid value of TpPAMPreferenceData is returned



5.2.1.1.4 IpPAMAgentPresence

Precondition: IpPAMAgentPresence supported.

Test PAM_AS_11

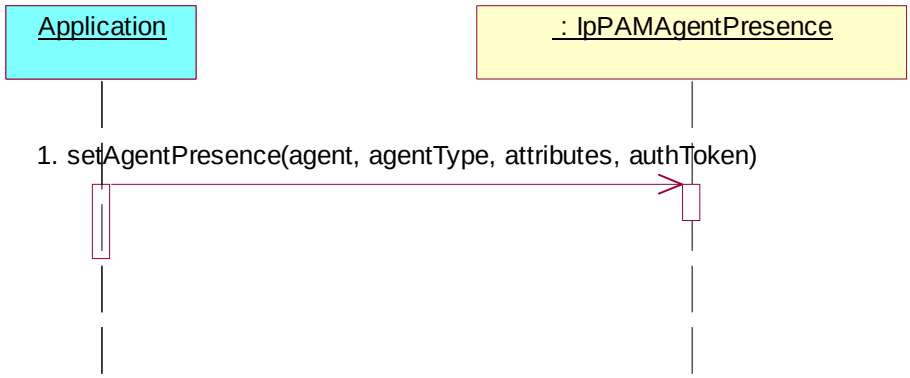
Summary: **IpPAMAgentPresence**, setAgentPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.4.1.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setAgentPresence()**
   Parameters: agent, agentType, attributes, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_12

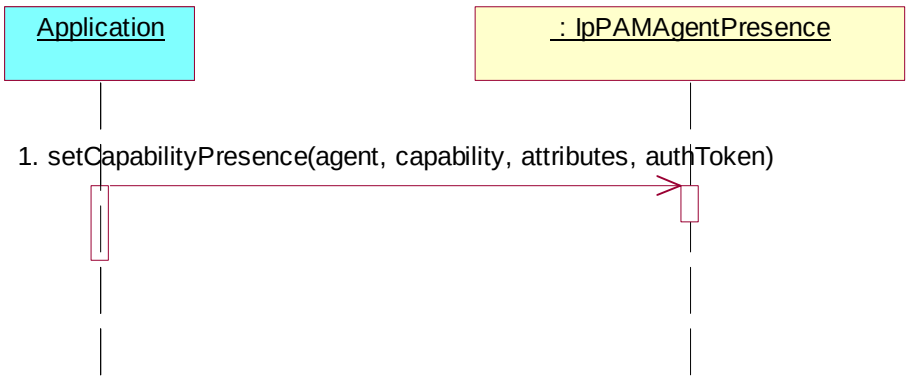
Summary: **IpPAMAgentPresence**, setCapabilityPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.4.2.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setCapabilityPresence()**
   Parameters: agent, capability, attributes, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_13

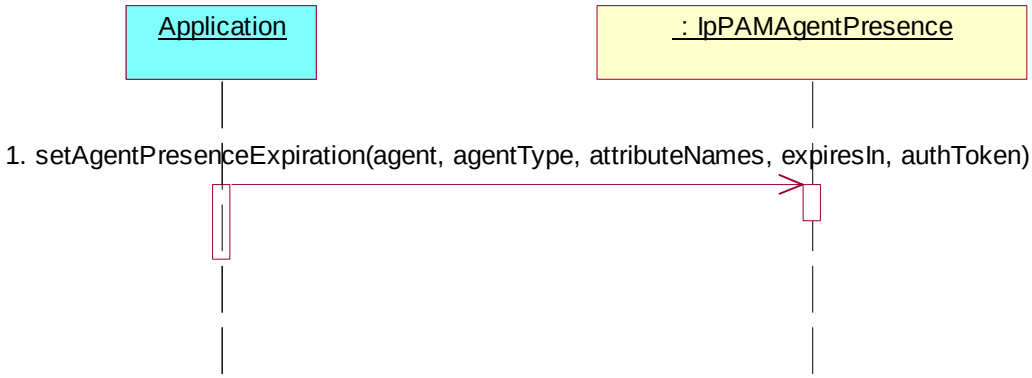
Summary: **IpPAMAgentPresence**, setAgentPresenceExpiration successful.

Reference: ES 202 915-14 [1], clause 8.2.4.3.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setAgentPresenceExpiration()**
   Parameters: agent, agentType, attributeNames, expiresIn, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_14

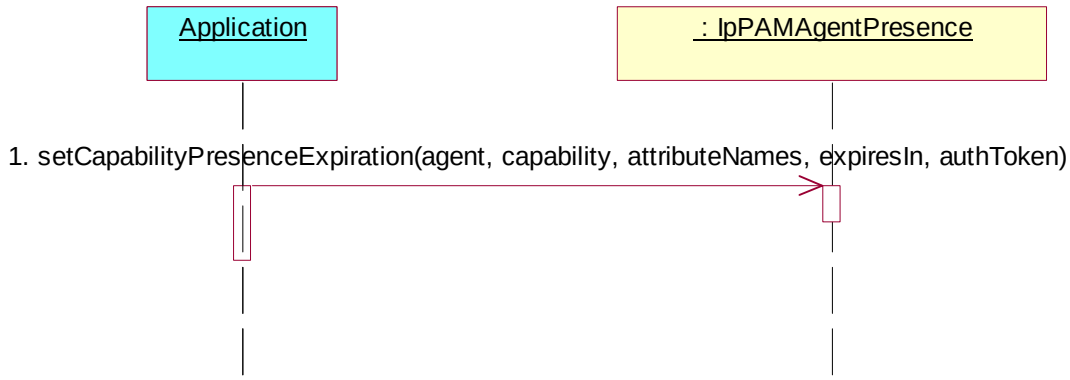
Summary: **IpPAMAgentPresence**, setCapabilityPresenceExpiration successful.

Reference: ES 202 915-14 [1], clause 8.2.4.4.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **setCapabilityPresenceExpiration()**
   Parameters: agent, capability, attributeNames, expiresIn, authToken as obtained in Preamble
   Check: no exception returned



Test PAM_AS_15

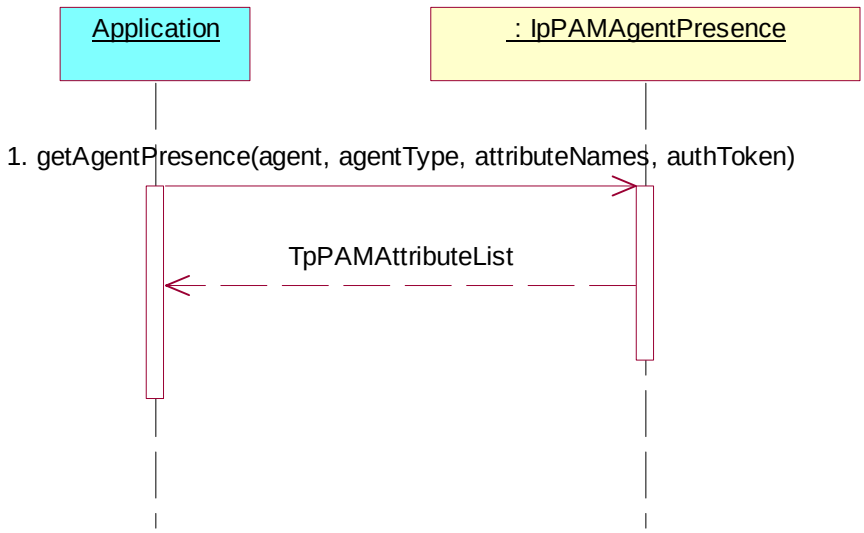
Summary: **IpPAMAgentPresence**, getAgentPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.4.5.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getAgentPresence()**
   Parameters: agent, agentType, attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned



Test PAM_AS_16

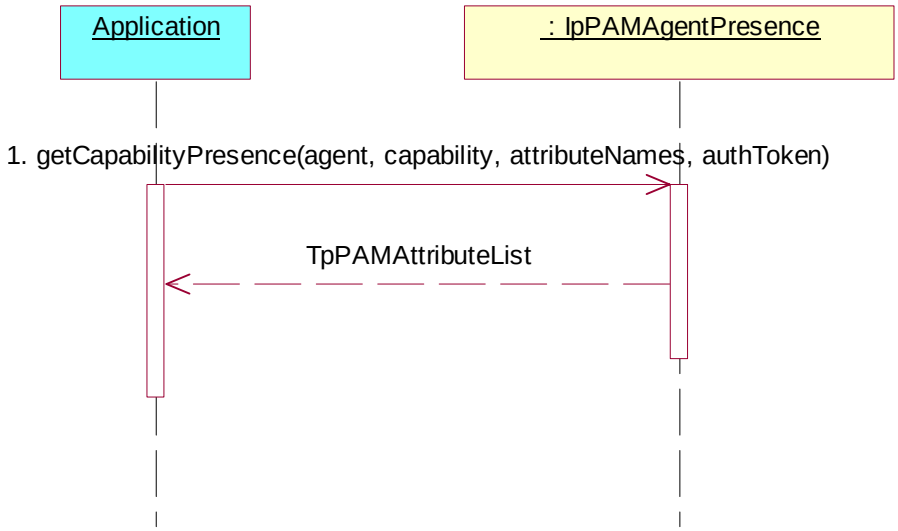
Summary: **IpPAMAgentPresence**, getCapabilityPresence successful.

Reference: ES 202 915-14 [1], clause 8.2.4.6.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call **getCapabilityPresence()**
   Parameters: agent, capability, attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned



5.2.1.2 PAM Event Management Service

5.2.1.2.1 IpPAMEventManager

Precondition: IpPAMEventManager supported.

Test PAM_EM_01

Summary: **IpPAMEventManager**, obtainInterface and getAuthToken successful.

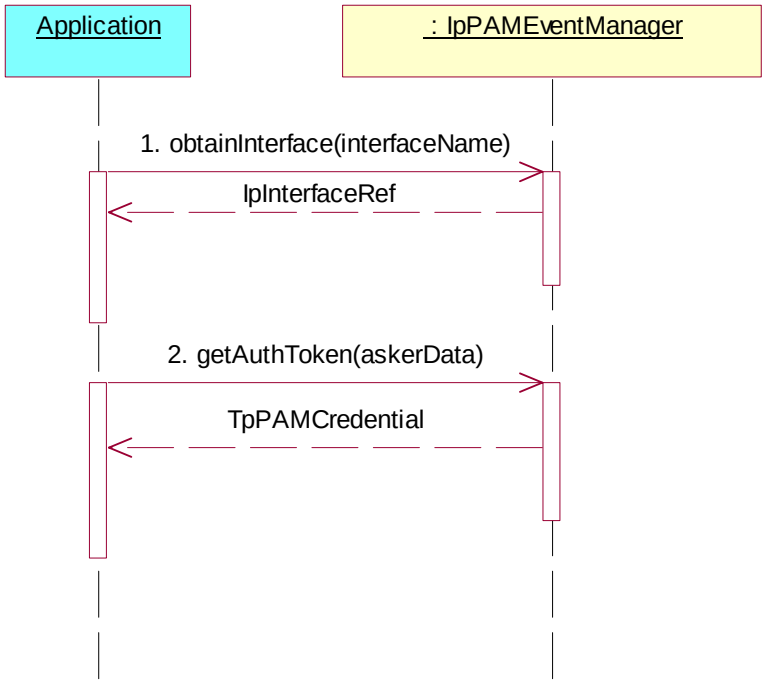
Reference: ES 202 915-14 [1], clause 8.3.1.

Preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMEventManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **obtainInterface()**
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call **getAuthToken()**
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Preamble_getAuthToken_EM

Summary: Preamble to obtain the authentication token.

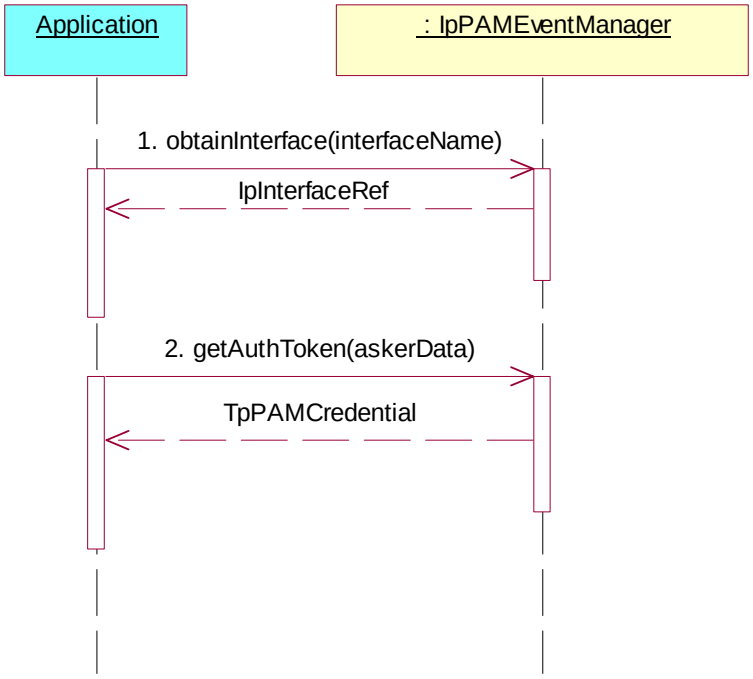
Reference: ES 202 915-14 [1], clause 8.3.1.

Pre-preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMEventManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call **obtainInterface()**
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call **getAuthToken()**
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Test PAM_EM_02

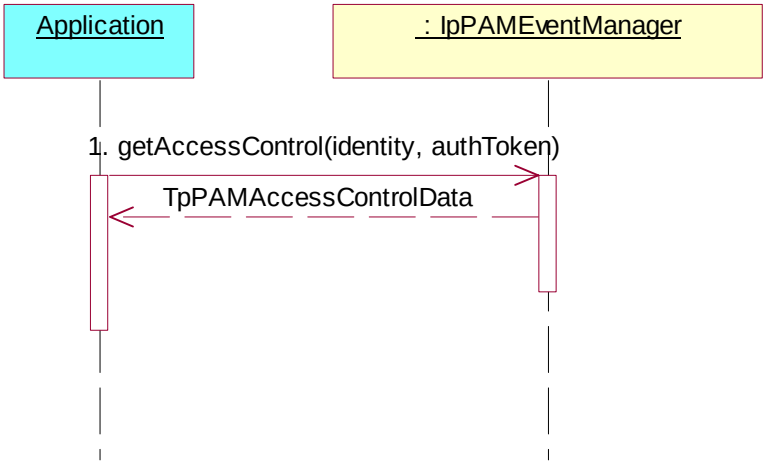
Summary: **IpPAMEventManager**, getAccessControl successful.

Reference: ES 202 915-14 [1], clause 8.3.1.3.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **getAccessControl()**
   Parameters: identity, authToken as obtained in Preamble
   Check: valid value of TpPAMAccessControlData is returned



Test PAM_EM_03

Summary: **IpPAMEventManager**, setAccessControl successful.

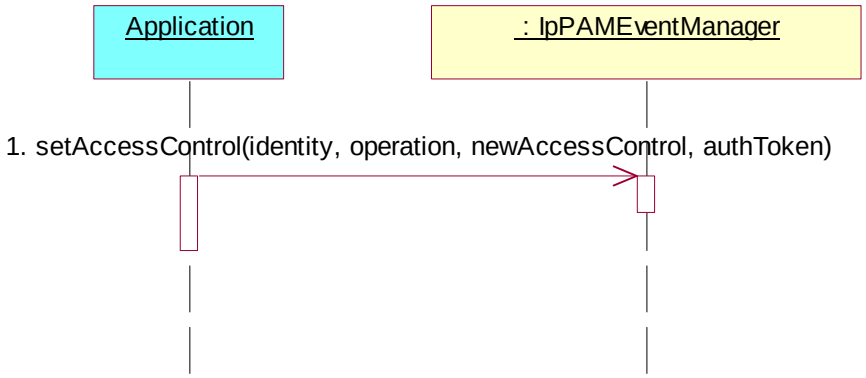
Reference: ES 202 915-14 [1], clause 8.3.1.4.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **setAccessControl()**
   Parameters: identity, operation, newAccessControl, authToken as obtained in Preamble
   Check: no exception returned

NOTE: setAccessControl() can be used in conjunction with getAccessControl().



5.2.1.2.2 IpPAMEventHandler

Precondition: IpPAMEventHandler supported.

Test PAM_EM_04

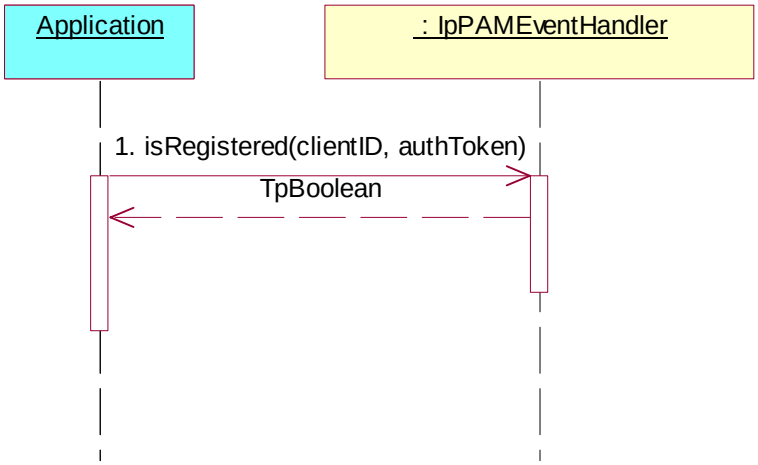
Summary: **IpPAMEventHandler**, isRegistered successful.

Reference: ES 202 915-14 [1], clause 8.3.2.1.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **isRegistered()**
   Parameters: clientID, authToken as obtained in Preamble
   Check: valid value of TpBoolean is returned



Test PAM_EM_05

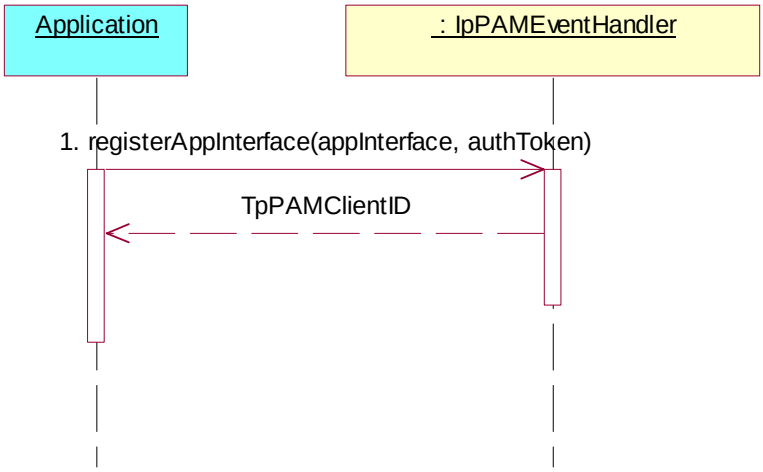
Summary: **IpPAMEventHandler**, regAppInterface successful.

Reference: ES 202 915-14 [1], clause 8.3.2.2.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned



Test PAM_EM_06

Summary: **IpPAMEventHandler**, registerForEvent successful.

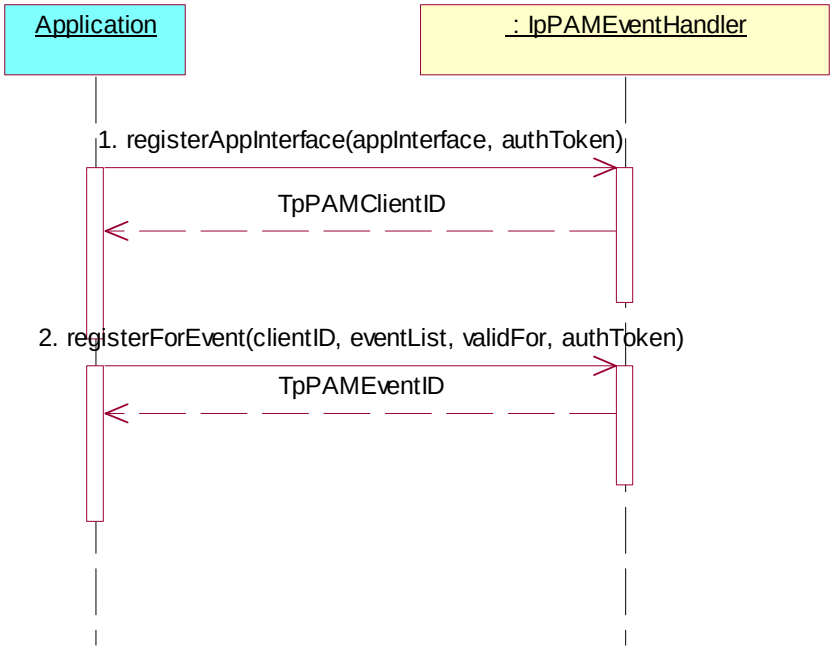
Reference: ES 202 915-14 [1], clause 8.3.2.3.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned

2. Method call **registerForEvent()**
   Parameters: clientID as returned in 1., eventList, valid For, authToken as obtained in Preamble
   Check: valid value of TpPAMEventID is returned



Test PAM_EM_07

Summary: **IpPAMEventHandler**, registerForEvent and eventNotify successful.

Reference: ES 202 915-14 [1], clause 8.3.2.3.

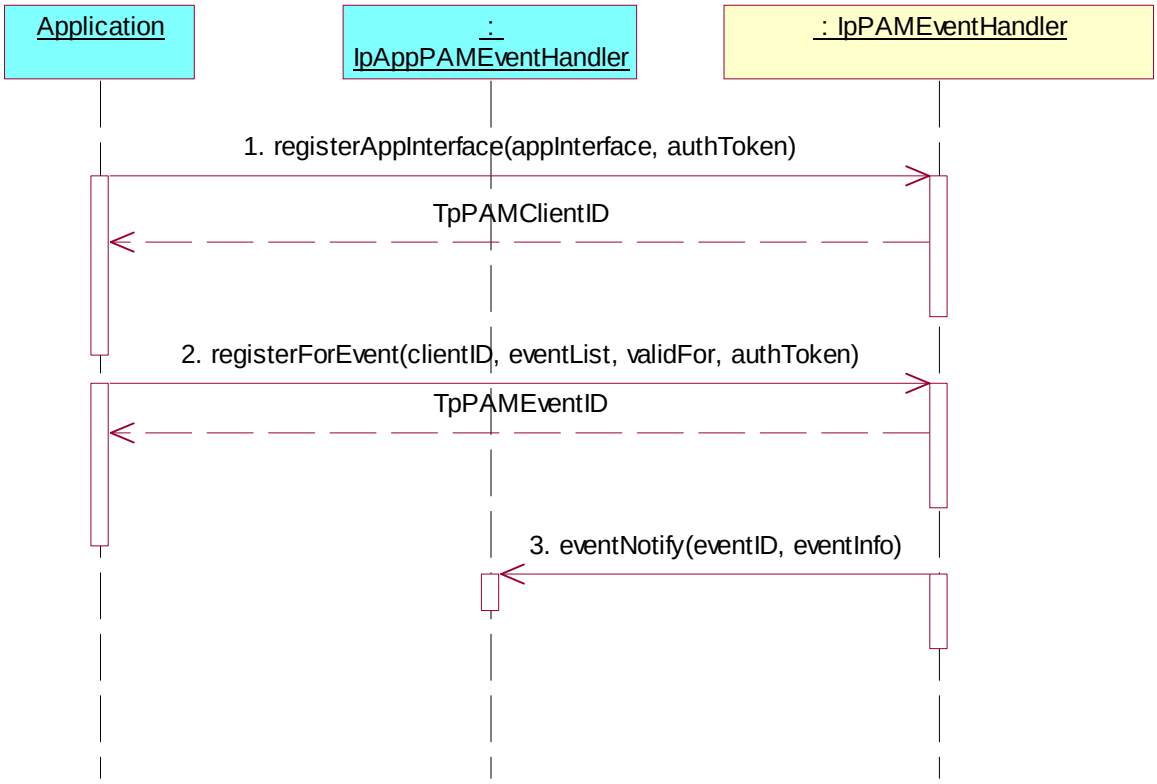
Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned

2. Method call **registerForEvent()**
   Parameters: clientID as returned in 1., eventList, valid For, authToken as obtained in Preamble
   Check: valid value of TpPAMEventID is returned

3. Triggered action: cause IUT to call **eventNotify()** method on the tester's (Application) **IpAppPAMEventHandler** interface.
   Parameters: eventID as returned in 2., eventInfo.



Test PAM_EM_08

Summary: **IpPAMEventHandler**, registerForEvent and eventNotifyErr successful.

Reference: ES 202 915-14 [1], clause 8.3.2.3.

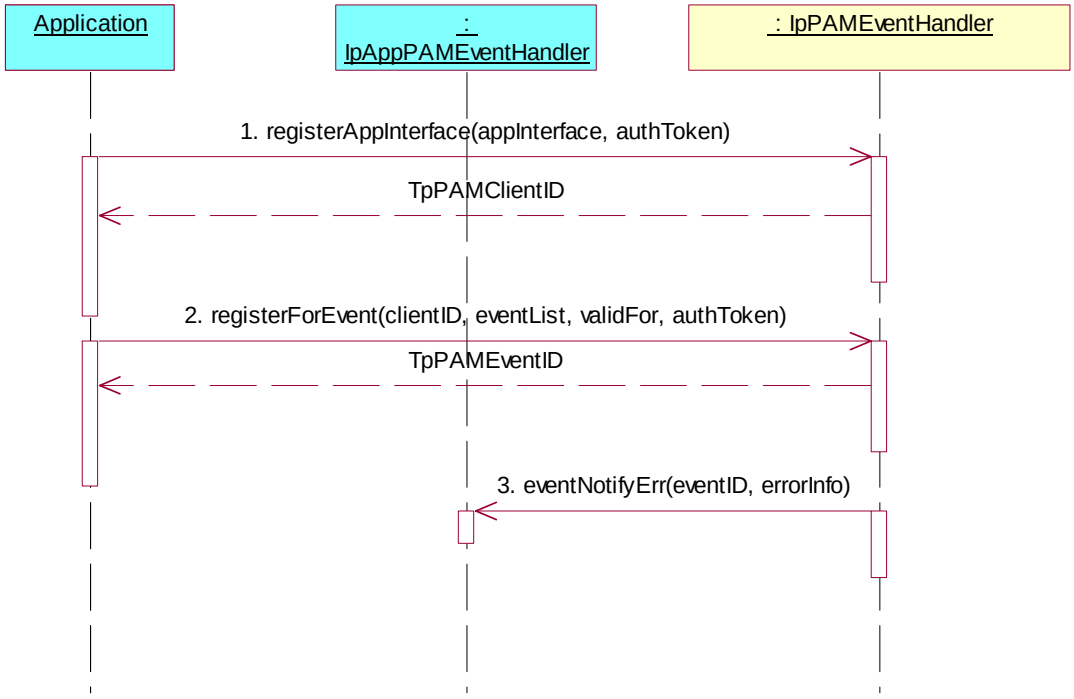
Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned

2. Method call **registerForEvent()**
   Parameters: clientID as returned in 1., eventList, valid For, authToken as obtained in Preamble
   Check: valid value of TpPAMEventID is returned

3. Triggered action: cause IUT to call **eventNotifyErr()** method on the tester's (Application) **IpAppPAMEventHandler** interface.
   Parameters: eventID as returned in 2., errorInfo



Test PAM_EM_09

Summary: **IpPAMEventHandler**, deregisterAppInterface successful.

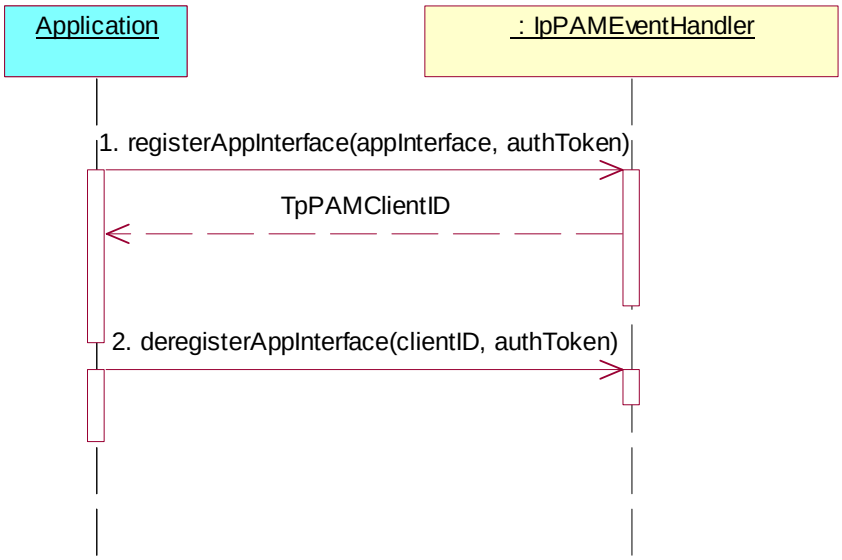
Reference: ES 202 915-14 [1], clause 8.3.2.2.

Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned

2. Method call **deregisterAppInterface ()**
   Parameters: clientID as returned in 1., authToken as obtained in Preamble
   Check: no exception returned



Test PAM_EM_10

Summary: **IpPAMEventHandler**, deregisterFromEvent successful.

Reference: ES 202 915-14 [1], clause 8.3.2.3.

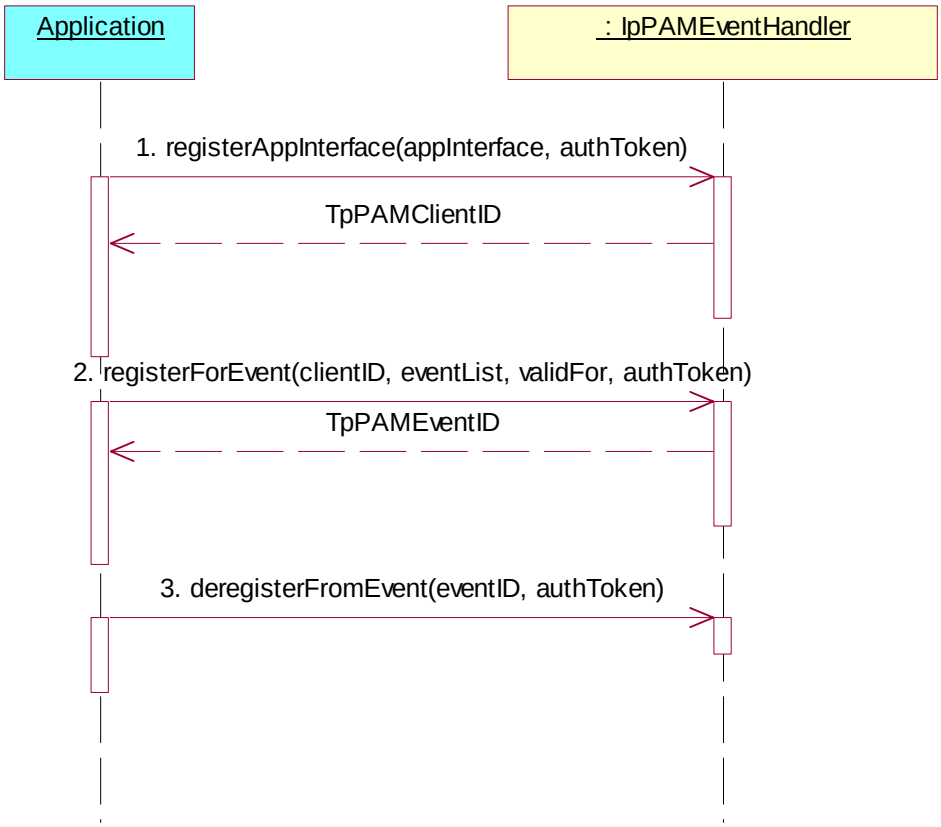
Preamble: **Preamble_getAuthToken_EM.**

Test Sequence:

1. Method call **registerAppInterface()**
   Parameters: appInterface, authToken as obtained in Preamble
   Check: valid value of TpPAMClientID is returned

2. Method call **registerForEvent()**
   Parameters: clientID as returned in 1., eventList, valid For, authToken as obtained in Preamble
   Check: valid value of TpPAMEventID is returned

3. Method call **deregisterFromEvent()**
   Parameters: eventID as returned in 2., authToken as obtained in Preamble
   Check: no exception returned



5.3.1.3 PAM Provisioning Service

5.2.1.3.1 IpPAMProvisioningManager

Precondition: IpPAMProvisioningManager supported.

Test PAM_PM_01

Summary: IpPAMProvisioningManager, obtainInterface and getAuthToken successful.

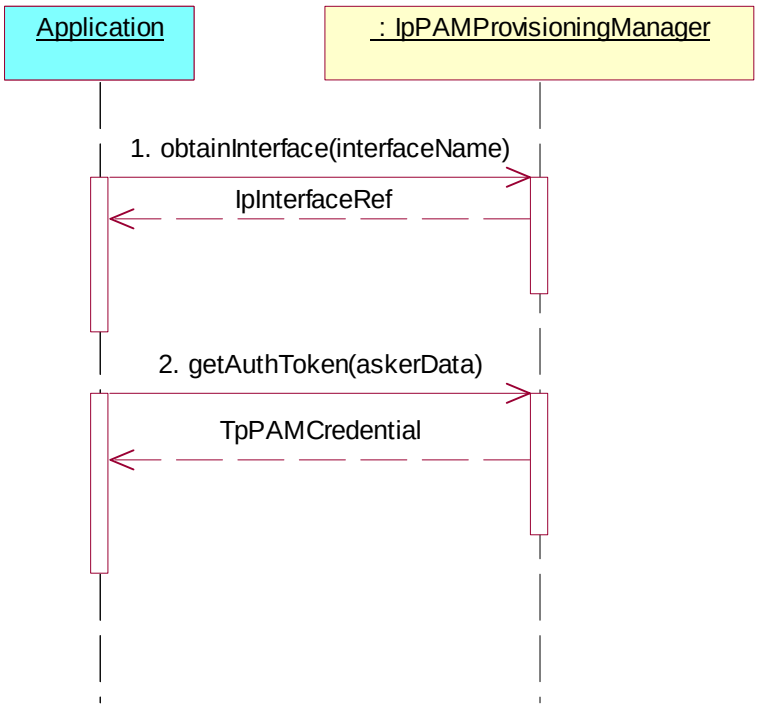
Reference: ES 202 915-14 [1], clause 8.1.1.

Preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMProvisioningManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call obtainInterface()
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call getAuthToken()
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Preamble_getAuthToken_PM

Summary: Preamble to obtain the authentication token.

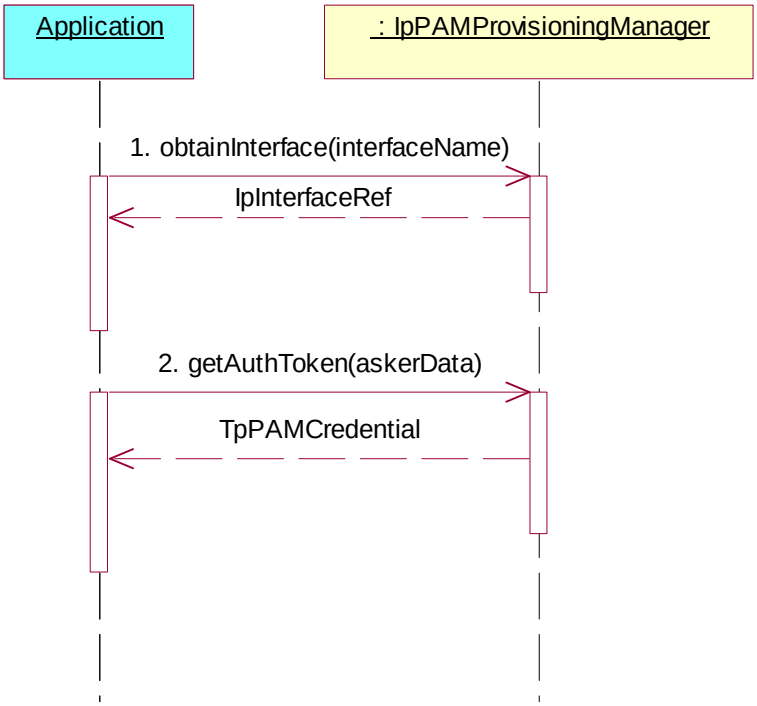
Reference: ES 202 915-14 [1], clause 8.1.

Pre-preamble: Registration of the IUT (PAM SCF) and the tester (application) to the framework. The tester must have obtained a reference to an instance of the IpPAMPresenceAvalibilityManager interface through selecting that service and signing the required service agreement.

Test Sequence:

1. Method call obtainInterface()
   Parameters: interfaceName
   Check: valid value of IpInterfaceRef is returned

2. Method call getAuthToken()
   Parameters: askerData
   Check: valid value of TpPAMCredential is returned



Test PAM_PM_02

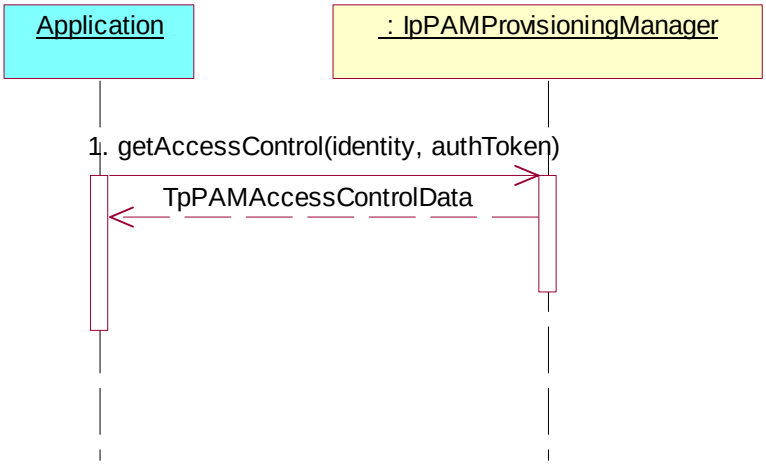
Summary: **IpPAMPresenceAvailabilityManager**, getAccessControl successful.

Reference: ES 202 915-14 [1], clause 8.1.1.

Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call getAccessControl()
   Parameters: identity, authToken as obtained in Preamble
   Check: valid value of TpPAMAccessControlData is returned



Test PAM_PM_03

Summary: **IpPAMPresenceAvailabilityManager**, setAccessControl successful.

Reference: ES 202 915-14 [1], clause 8.1.1.

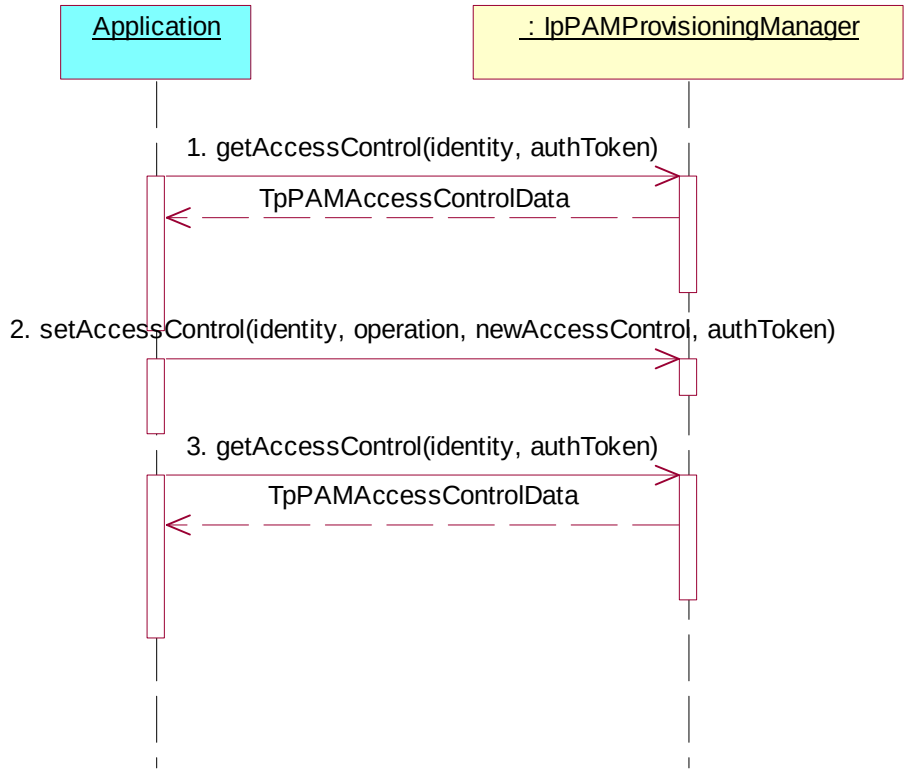
Preamble: **Preamble_getAuthToken_AS.**

Test Sequence:

1. Method call getAccessControl()
   Parameters: identity, authToken as obtained in Preamble
   Check: valid value of TpPAMAccessControlData is returned

2. Method call setAccessControl()
   Parameters: identity, operation, newAccessControl, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call getAccessControl()
   Parameters: identity, authToken as obtained in Preamble
   Check: valid value of TpPAMAccessControlData is returned with modified newAccessControl in 2.



5.2.1.3.2 IpPAMIdentityManagement

Precondition: IpPAMIdentityManagement supported.

Test PAM_PM_04

Summary: **IpPAMIdentityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

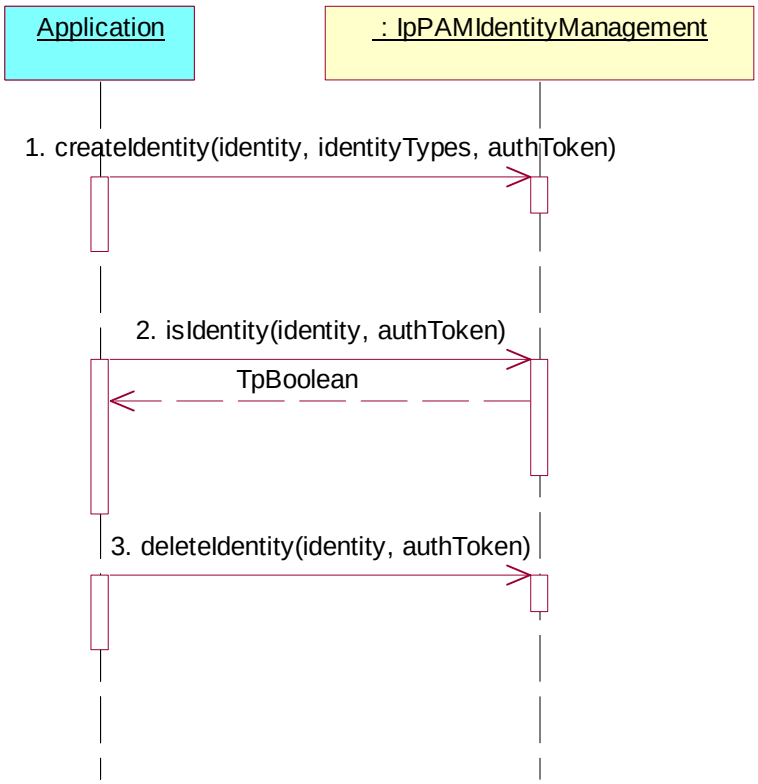
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **isIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: true value of TpBoolean is returned

3. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_05

Summary: **IpPAMIdentityManagement**, isGroupIdentity successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

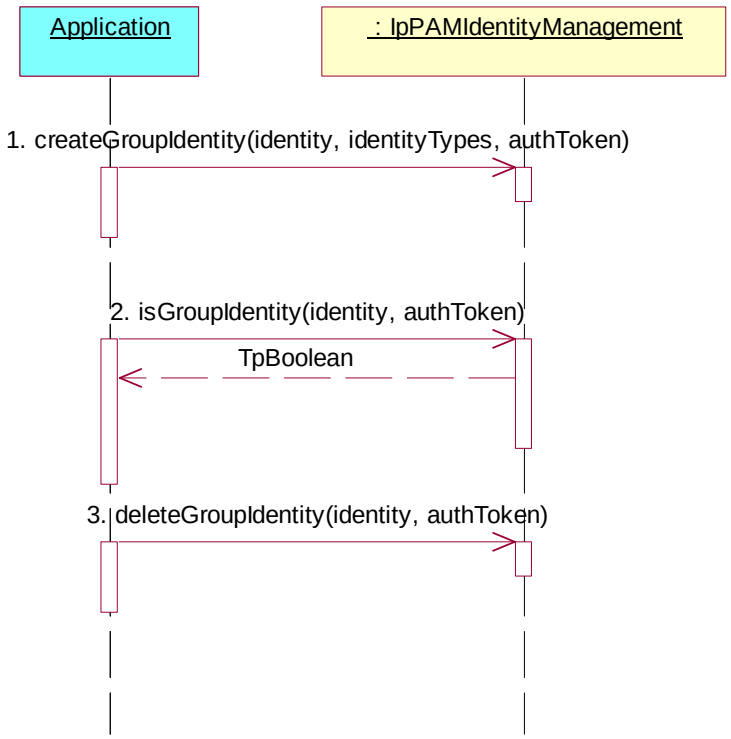
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createGroupIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **isGroupIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: true value of TpBoolean is returned

3. Method call **deleteGroupIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_06

Summary: **IpPAMIdentityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createGroupIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **addToGroup ()**
   Parameters: group given in 2., member given in 1., authToken as obtained in Preamble
   Check: no exception is returned

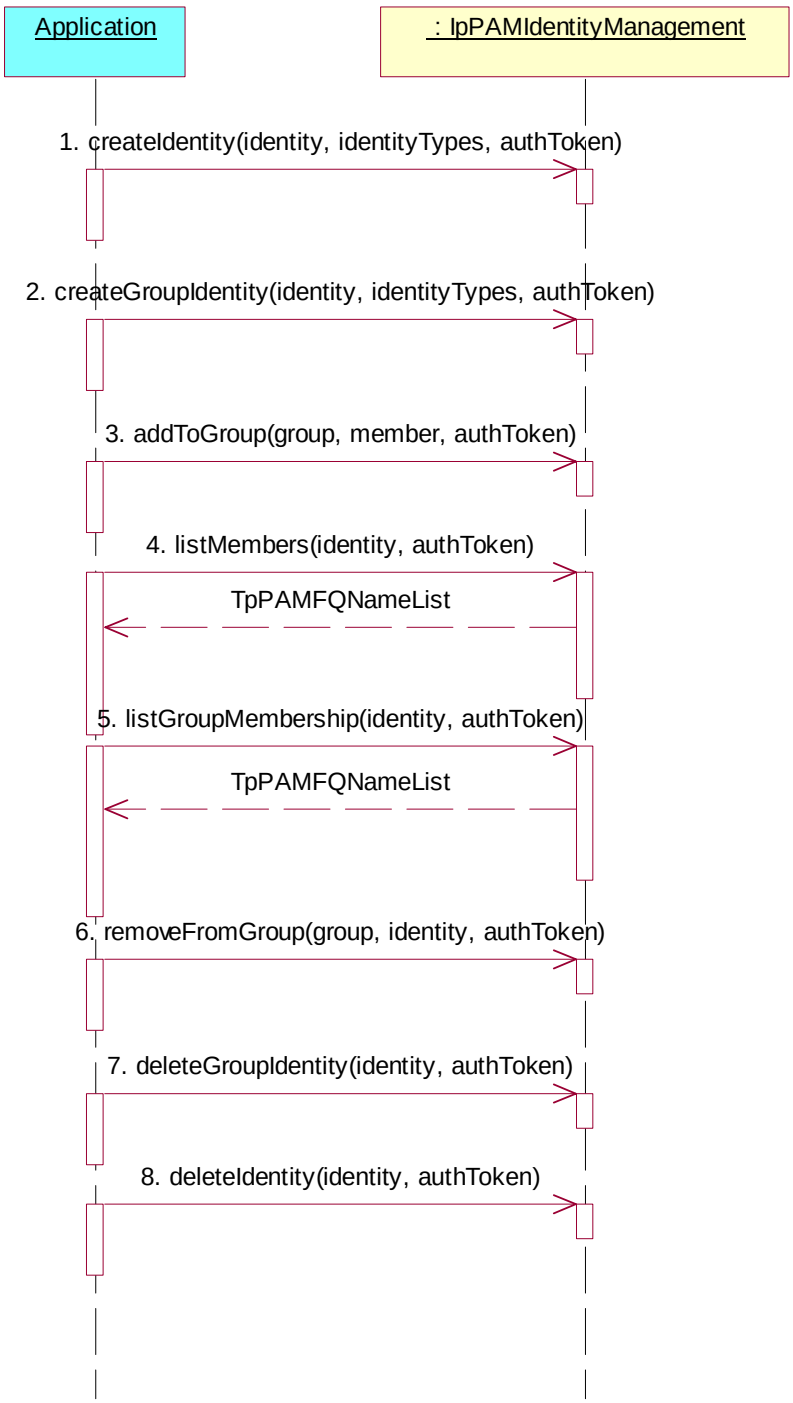
4. Method call **listMembers ()**
   Parameters: group given in 2., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList that include identity given in 1. is returned

5. Method call **listGroupMembership ()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList that include group given in 2. is returned

6. Method call **removeFromGroup ()**
   Parameters: group given in 2., member given in 1., authToken as obtained in Preamble
   Check: no exception is returned

7. Method call **deleteGroupIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMIdentityManagement
1. createIdentity(identity, identityTypes, authToken)
2. createGroupIdentity(identity, identityTypes, authToken)
3. addToGroup(group, member, authToken)
4. listMembers(identity, authToken)
TpPAMFQNameList
5. listGroupMembership(identity, authToken)
TpPAMFQNameList
6. removeFromGroup(group, identity, authToken)
7. deleteGroupIdentity(identity, authToken)
8. deleteIdentity(identity, authToken)

Test PAM_PM_07

Summary: **IpPAMIdentityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

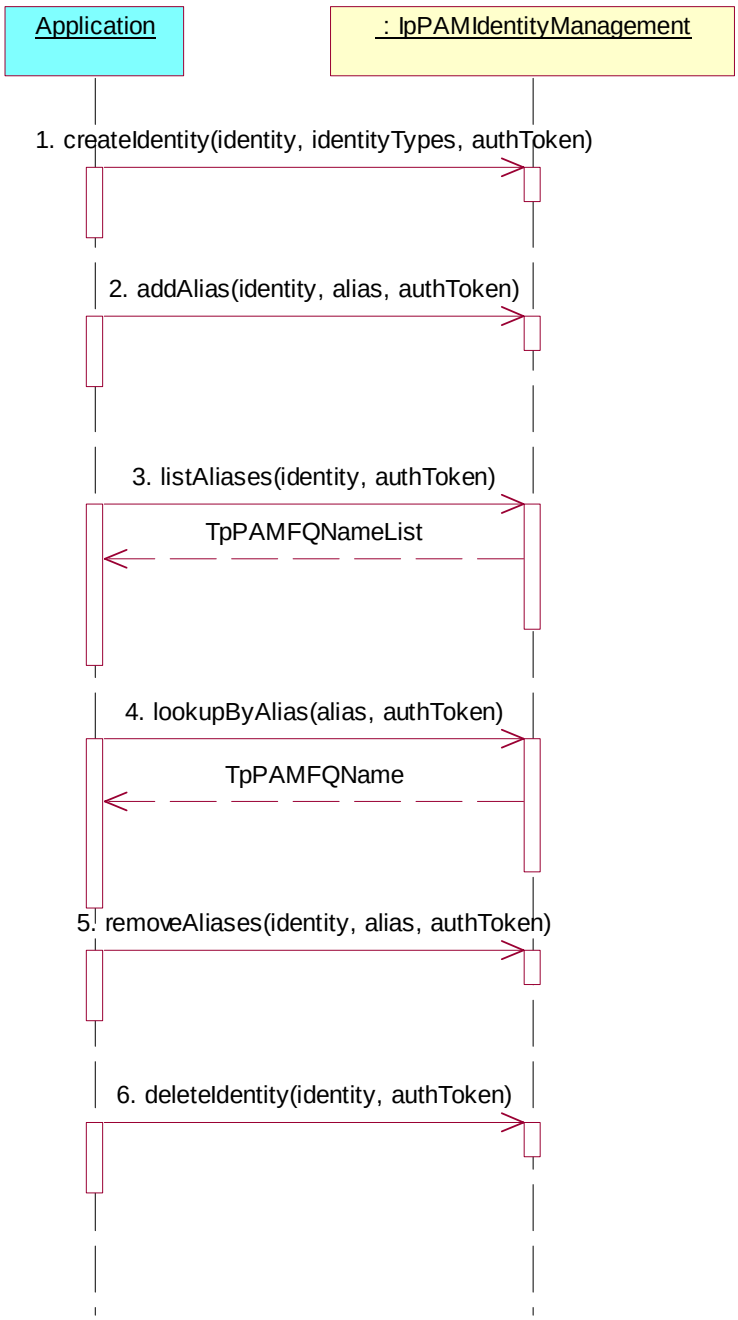
2. Method call **addAlias()**
   Parameters: identity given in 1., alias, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listAliases()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList included alias given in 2. is returned

4. Method call **lookupByAliases()**
   Parameters: alias given in 2., authToken as obtained in Preamble
   Check: valid value of TpPAMFQName corresponding with identity name given in 1. is returned

5. Method call **removeAliases()**
   Parameters: identity given in 1., alias given in 2., authToken as obtained in Preamble
   Check: no exception is returned

6. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_08

Summary: **IpPAMIdentityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

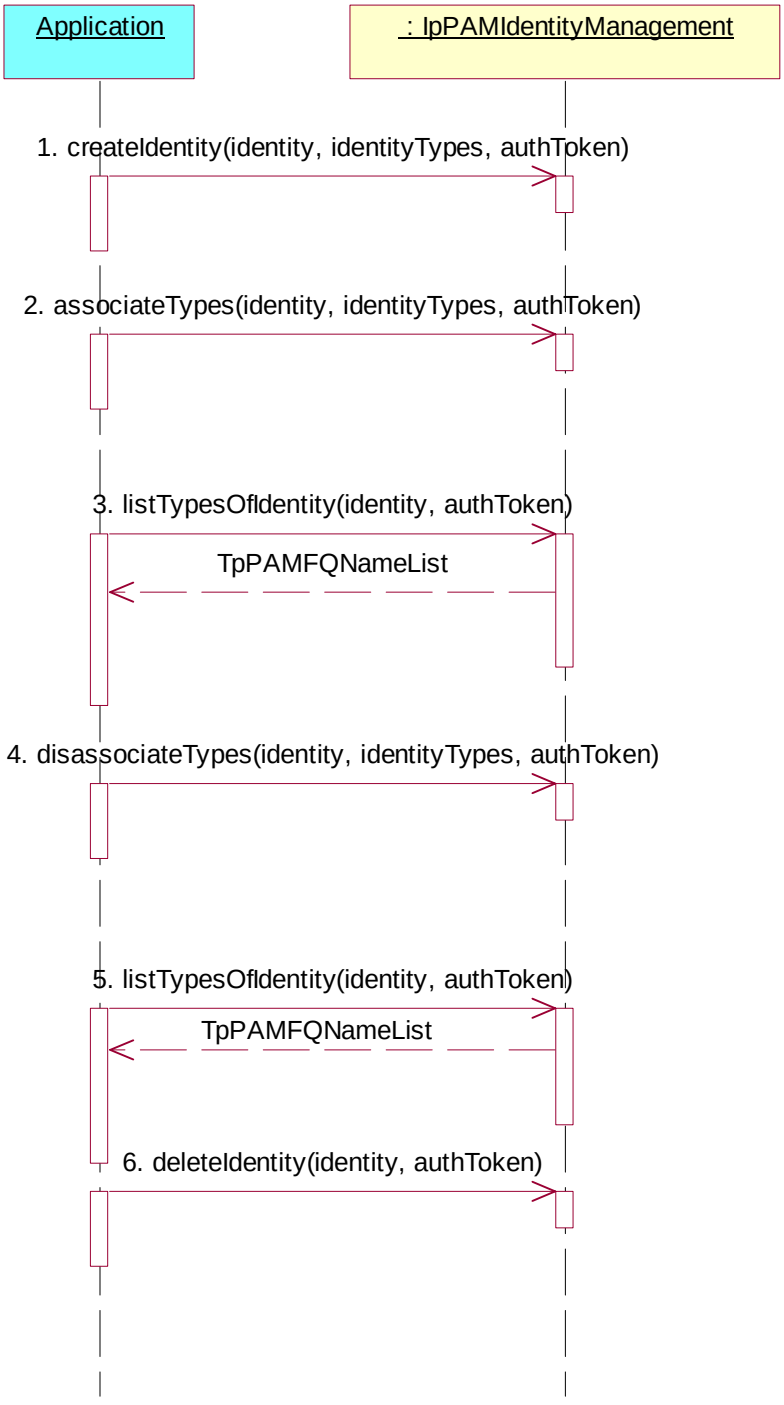
2. Method call **associateTypes()**
   Parameters: identity given in 1., identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listTypesOfIdentity()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList including identyTypes given in 2. is returned

4. Method call **disassociateTypes()**
   Parameters: identity given in 1., identityTypes given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **listTypesOfIdentity()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList without identyTypes given in 4. is returned

6. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_09

Summary: **IpPAMIdentityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

Preamble: **Preamble_getAuthToken_PM.**

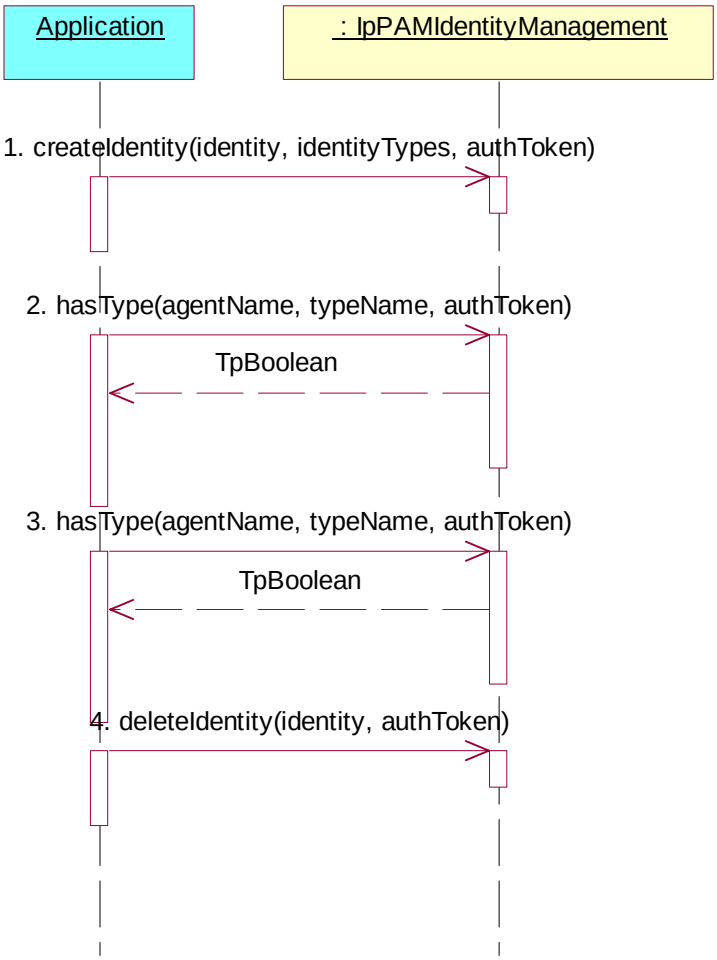
Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **hasType()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., authToken as obtained in Preamble
   Check: true value of TpBoolean is returned

3. Method call **hasType()**
   Parameters: identity given in 1., identityType previous not included in identityTypes given in 1., authToken as obtained in Preamble
   Check: false value of TpBoolean is returned

4. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_10

Summary: **IpPAMIdentityManagement**, getIdentityAttributes successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

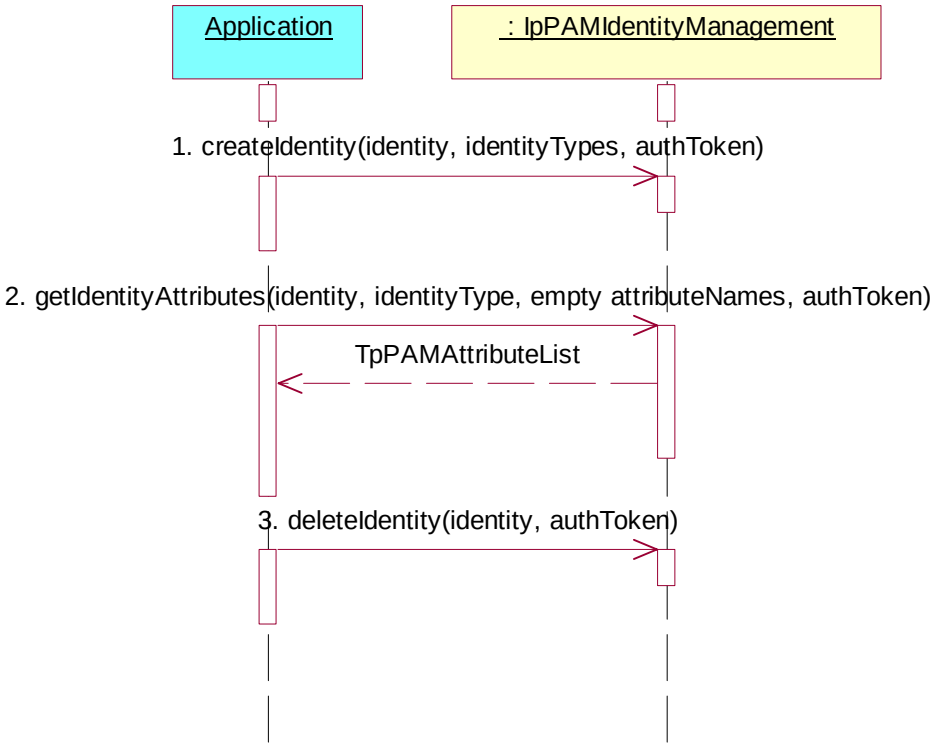
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getIdentityAttributes()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned

3. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_11

Summary: **IpPAMIdentityManagement**, getIdentityAttributes successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

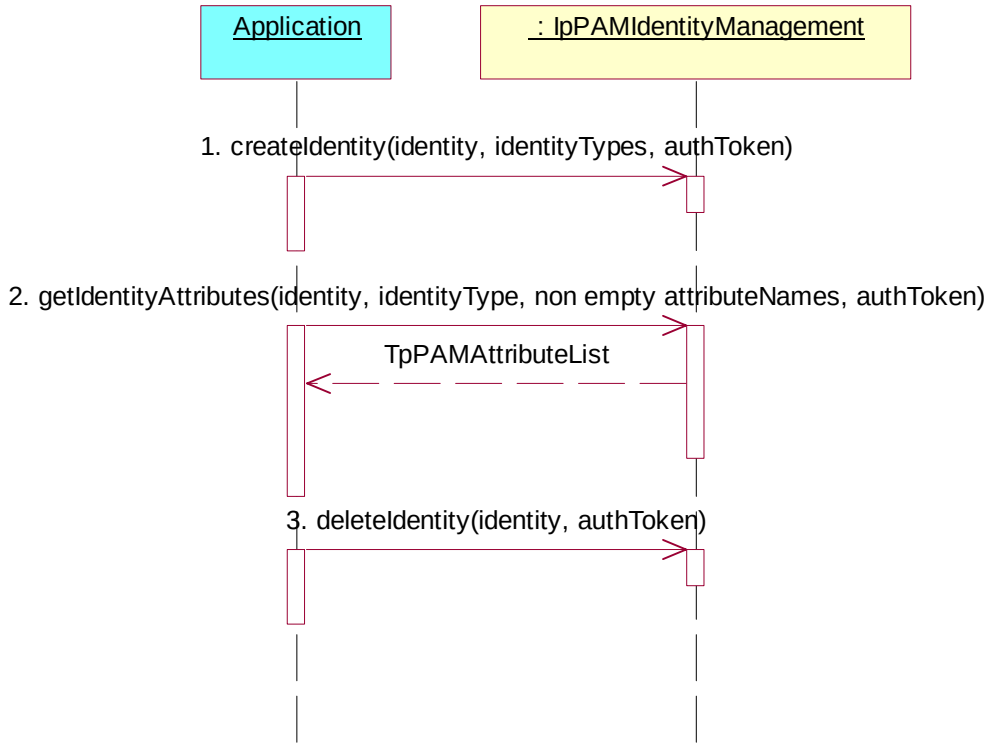
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getIdentityAttributes()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., non empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned included only attributes in attributeNames parameter

3. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_12

Summary: **IpPAMIdentityManagement**, setIdentityAttributes successful.

Reference: ES 202 915-14 [1], clause 8.1.2.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

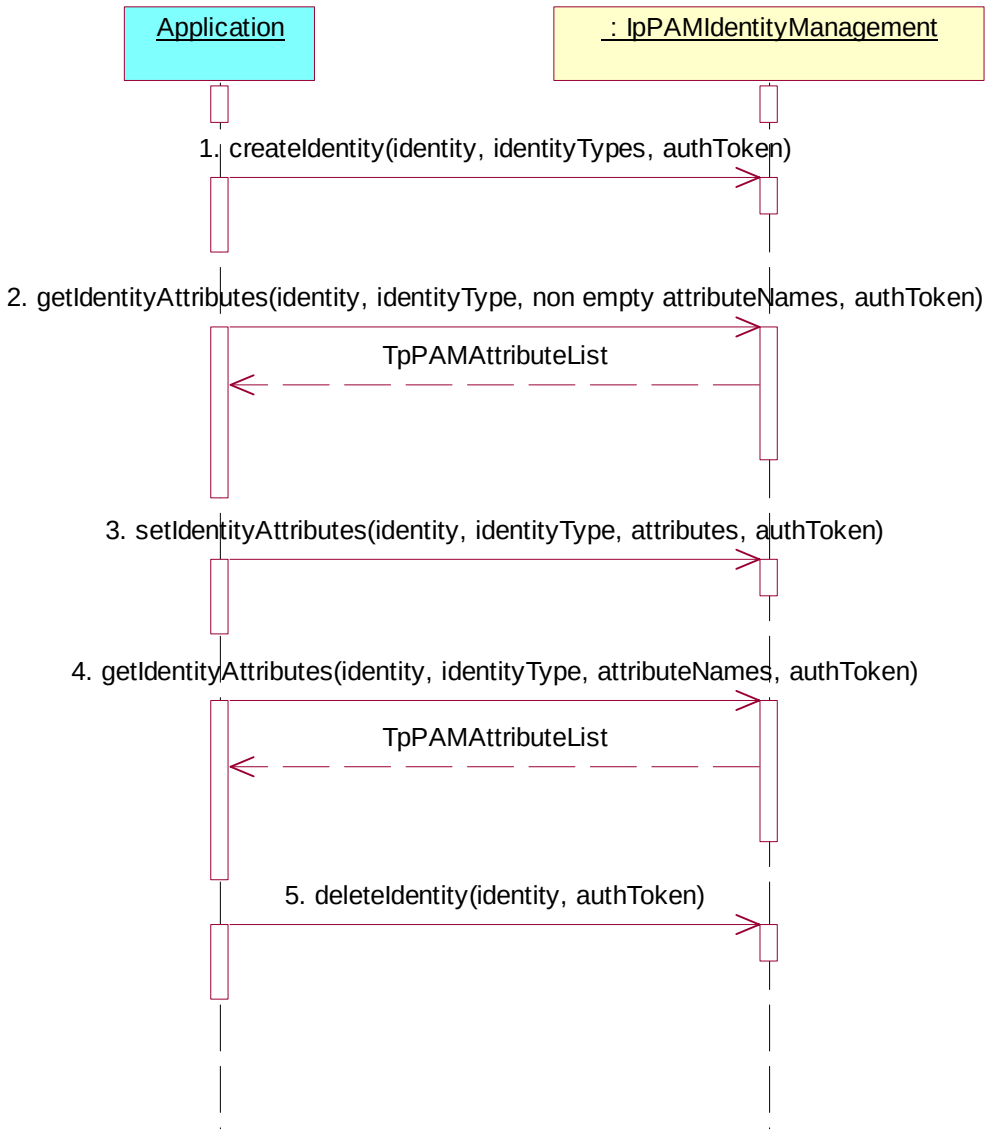
1. Method call **createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getIdentityAttributes()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned

3. Method call s**etIdentityAttributes()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., attributes with different values of TpPAMAttributeList returned in 2., authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned

4. Method call **getIdentityAttributes()**
   Parameters: identity given in 1., identityType previous included in identityTypes given in 1., empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned included attributes and values given in 3.

5. Method call **deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



5.2.1.3.3 IpPAMAgentManagement

Precondition: IpPAMAgentManagement supported.

Test PAM_PM_13

Summary: **IpPAMAgentManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

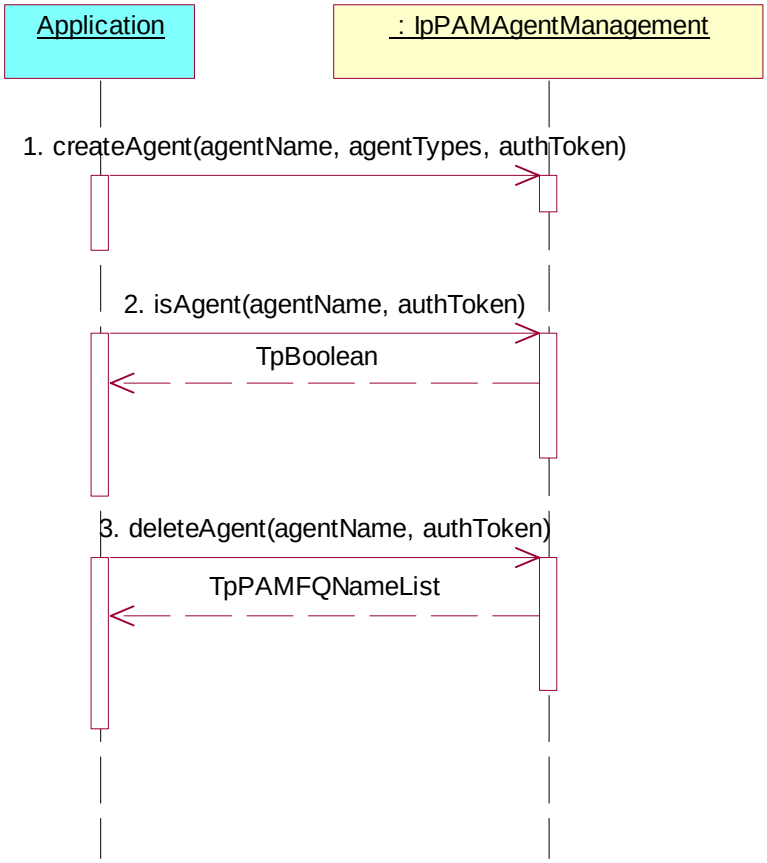
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **isAgent ()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: true value of TpBoolean is returned

3. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_14

Summary: **IpPAMAgentManagement**, disableCapabilities successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

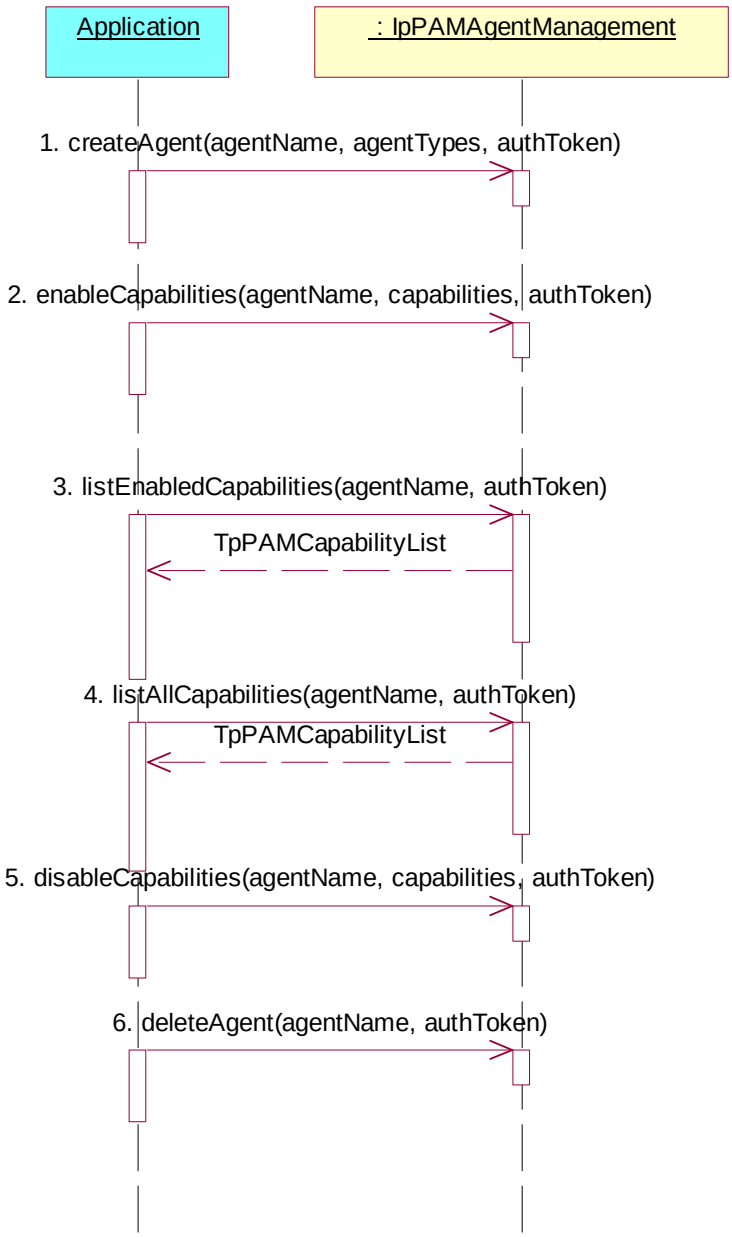
2. Method call **enableCapabilities()**
   Parameters: agentName given in 1., capabilities, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listEnabledCapabilities()**
   Parameters: agentName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMCapabilityList is returned

4. Method call **listAllCapabilities()**
   Parameters: agentName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMCapabilityList is returned

5. Method call **disableCapabilities()**
   Parameters: agentName given in 1., capabilities given in 2., authToken as obtained in Preamble
   Check: no exception is returned

6. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_15

Summary: **IpPAMAgentManagement**, disassociateTypes successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

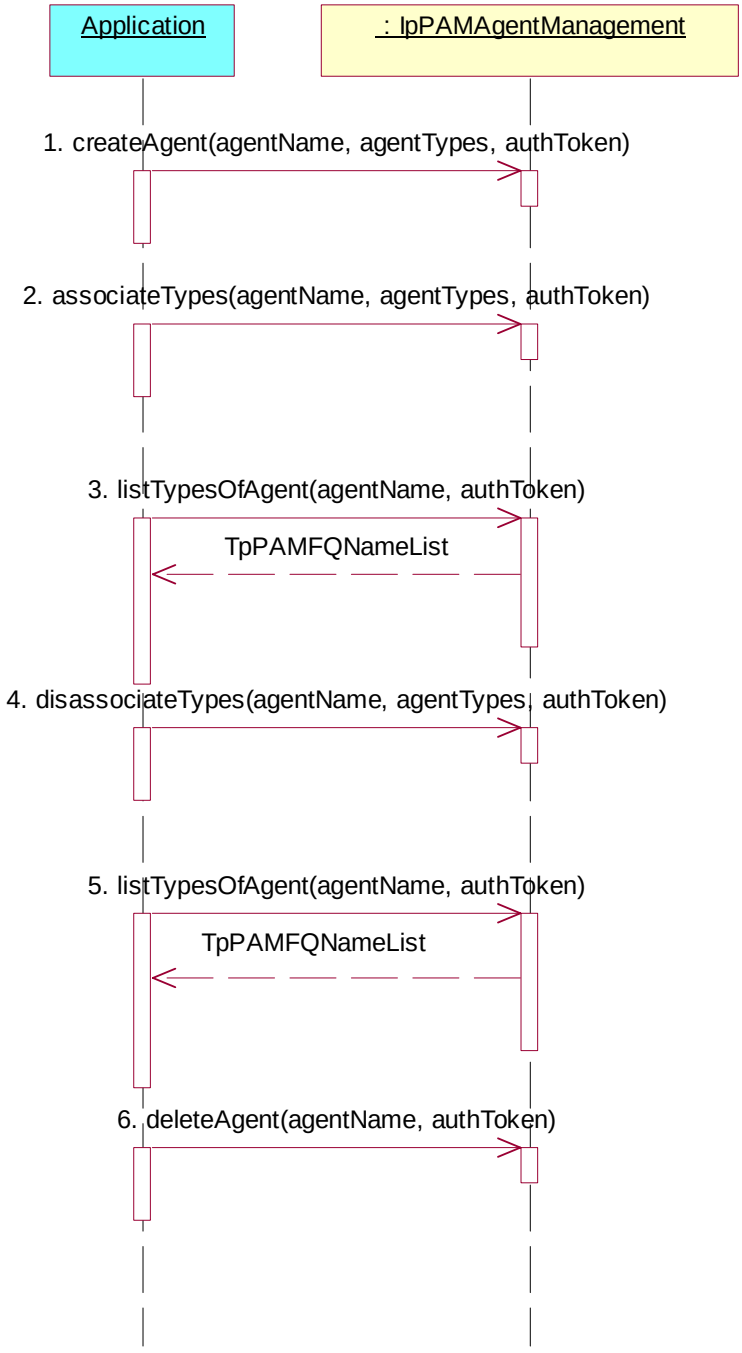
2. Method call **associateTypes()**
   Parameters: agentName given in 1., agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listTypesOfAgent()**
   Parameters: agentName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList including agentTypes given in 2. is returned

4. Method call **disassociateTypes()**
   Parameters: agentName given in 1., agentTypes given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **listTypesOfAgent()**
   Parameters: agentName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList not including agentTypes given in 2. is returned

6. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_16

Summary: **IpPAMAgentManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

Preamble: **Preamble_getAuthToken_PM.**

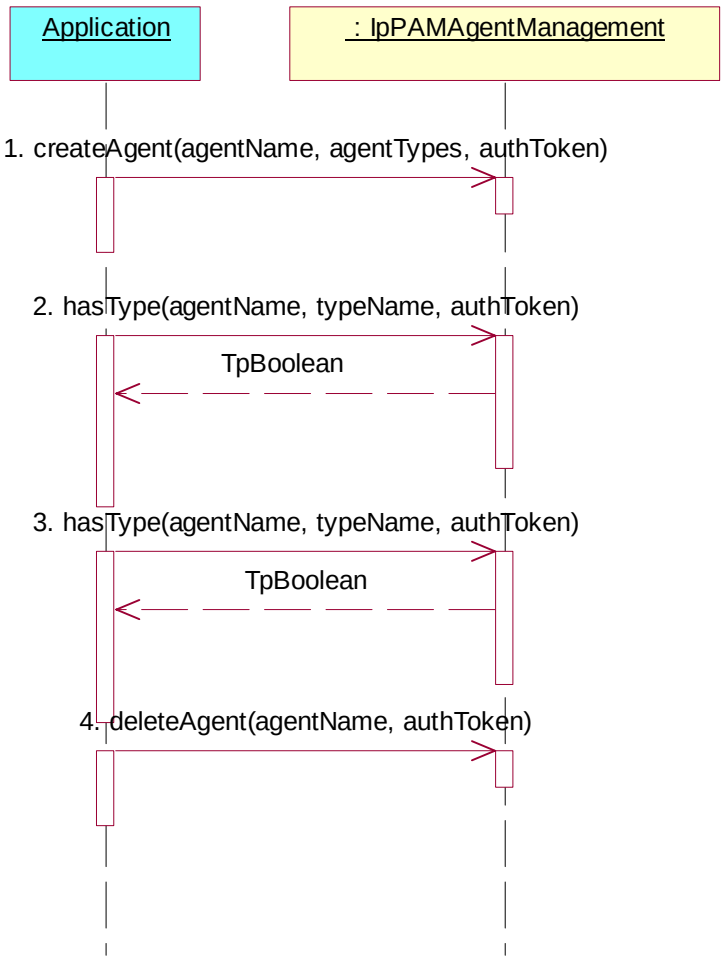
Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **hasType()**
   Parameters: agentName given in 1., agentType previously included in agentTypes given in 1., authToken as obtained in Preamble
   Check: true value of TpBoolean is returned

3. Method call **hasType()**
   Parameters: agentName given in 1., agentType previously not included in agentTypes given in 1., authToken as obtained in Preamble
   Check: false value of TpBoolean is returned

4. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_17

Summary: **IpPAMAgentManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

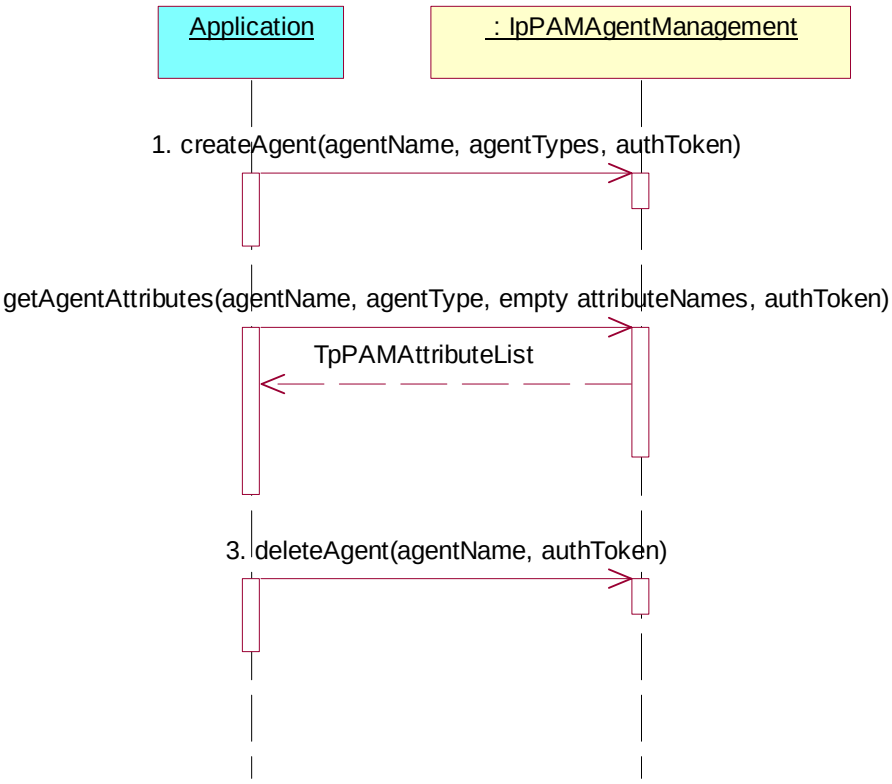
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getAgentAttributes()**
   Parameters: agentName given in 1., agentType previously included in agentTypes given in 1., empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned

2. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_18

Summary: **IpPAMAgentManagement**, getAgentAttributes successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

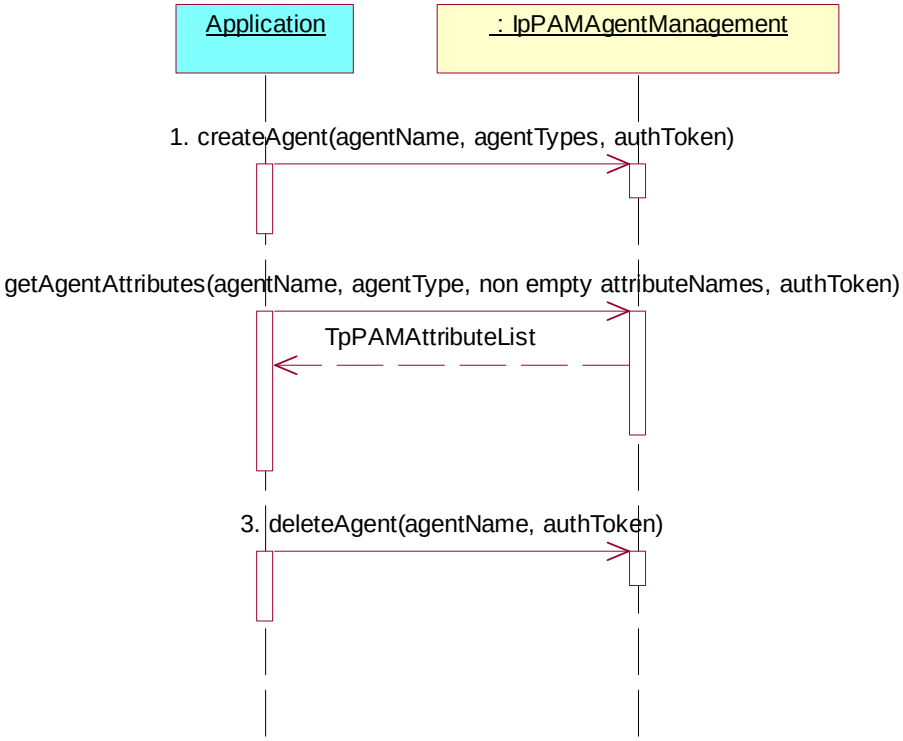
Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getAgentAttributes()**
   Parameters: agent given in 1., agentType previous included in agentTypes given in 1., non empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned included only attributes in attributeNames parameter

3. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_19

Summary: **IpPAMAgentManagement**, setAgentAttributes successful.

Reference: ES 202 915-14 [1], clause 8.1.3.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

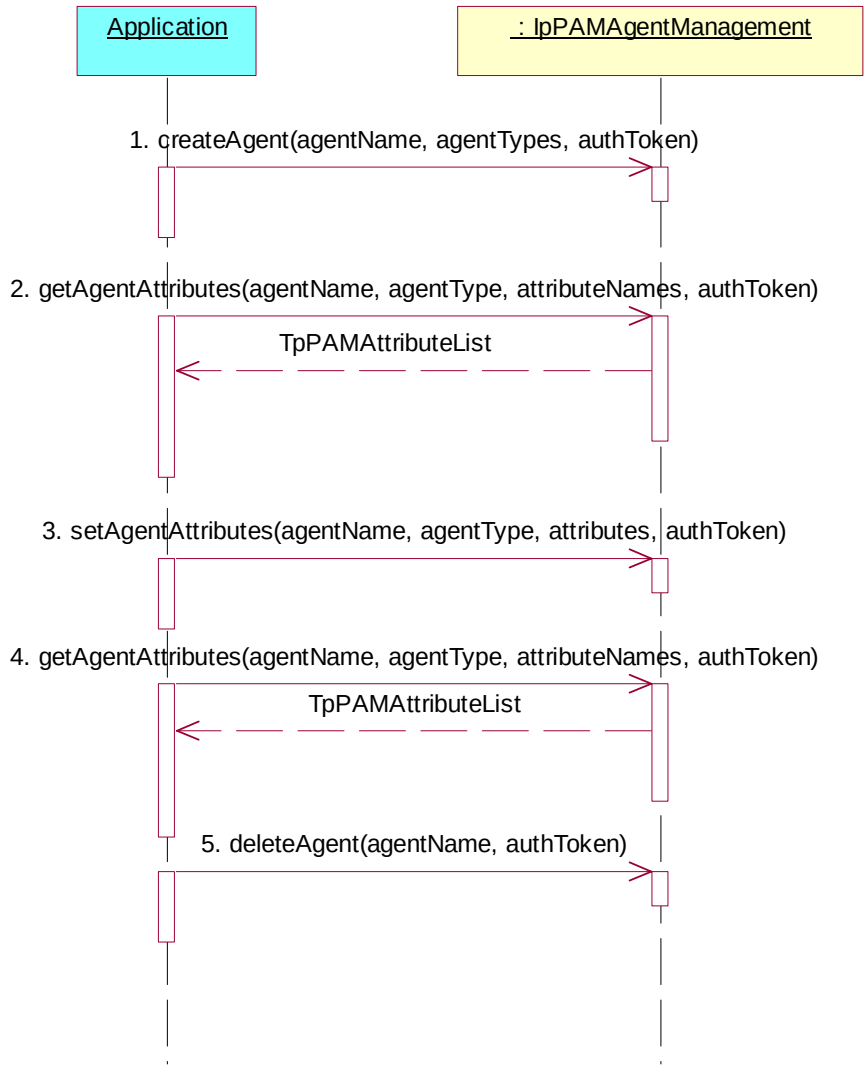
1. Method call **createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getAgentAttributes()**
   Parameters: agent given in 1., agentType previous included in agentTypes given in 1., non empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned included only attributes in attributeNames parameter

3. Method call s**etAgentAttributes()**
   Parameters: agentName given in 1., agentType previous included in agentTypes given in 1., attributes with different values of TpPAMAttributeList returned in 2., authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned

4. Method call **getAgentAttributes()**
   Parameters: agent given in 1., agentType previous included in agentTypes given in 1., non empty list of attributeNames, authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeList is returned included attributes and values given in 3.

5. Method call **deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned



5.2.1.3.4 IpPAMAgentAssignment

Precondition: IpPAMAgentAssignment supported.

Test PAM_PM_20

Summary: **IpPAMAgentAssignment**, listAssignAgent successful.

Reference: ES 202 915-14 [1], clause 8.1.4.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **IpPAMIdentityManagement.createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **IpPAMAgentManagement.createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **assignAgent()**
   Parameters: identity given in 1., agentName,given in 2., authToken as obtained in Preamble
   Check: no exception is returned

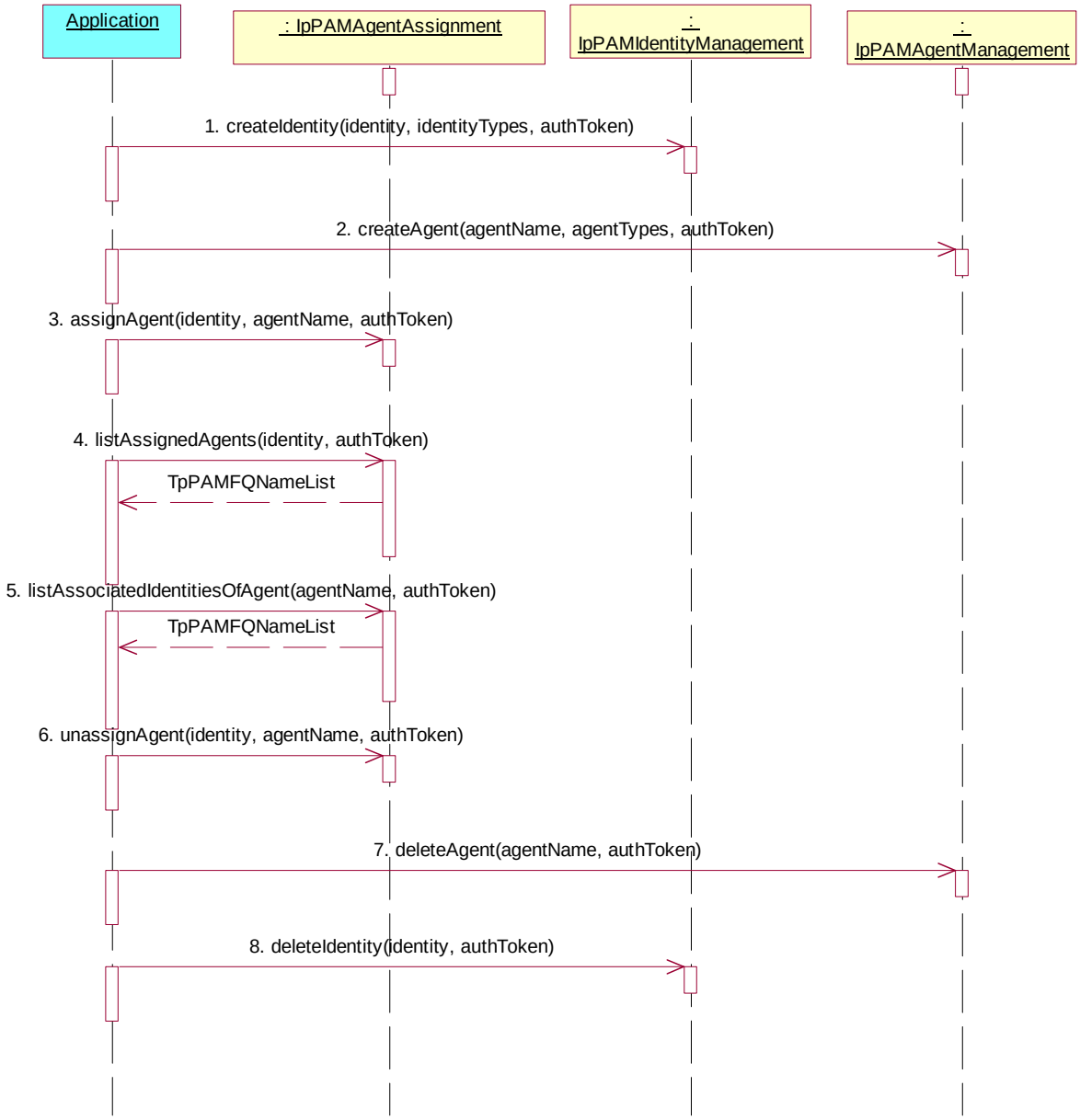
4. Method call **listAssignAgents()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList included agentName given in 2.

5. Method call **listAssociatedIdentitiesOfAgent()**
   Parameters: agentName given in 2., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList included identity given in 1.

6. Method call **unassignAgent()**
   Parameters: identity given in 1., agentName,given in 2., authToken as obtained in Preamble
   Check: no exception is returned

7. Method call **IpPAMAgentManagement.deleteAgent()**
   Parameters: same agentName than given in 2., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **IpPAMIdentityManagement.deleteIdentity()**
   Parameters: same identity than given in 1., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMAgentAssignment
: IpPAMIdentityManagement
: IpPAMAgentManagement
1. createIdentity(identity, identityTypes, authToken)
2. createAgent(agentName, agentTypes, authToken)
3. assignAgent(identity, agentName, authToken)
4. listAssignedAgents(identity, authToken)
TpPAMFQNameList
5. listAssociatedIdentitiesOfAgent(agentName, authToken)
TpPAMFQNameList
6. unassignAgent(identity, agentName, authToken)
7. deleteAgent(agentName, authToken)
8. deleteIdentity(identity, authToken)

Test PAM_PM_21

Summary: **IpPAMAgentAssignment**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.4.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **IpPAMIdentityManagement.createIdentity()**
   Parameters: identity, identityTypes, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **IpPAMAgentManagement.createAgent()**
   Parameters: agentName, agentTypes, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **IpPAMAgentManagement.enableCapabilities()**
   Parameters: agentName given in 2., capabilities, authToken as obtained in Preamble
   Check: no exception is returned

4. Method call **assignAgent()**
   Parameters: identity given in 1., agentName,given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **listAssignedAgentsByCapability()**
   Parameters: identity given in 1., capability included capabilities given in 3., authToken as obtained in Preamble
   Check: valid value of TpPAMFQNameList included agentName given in 2.

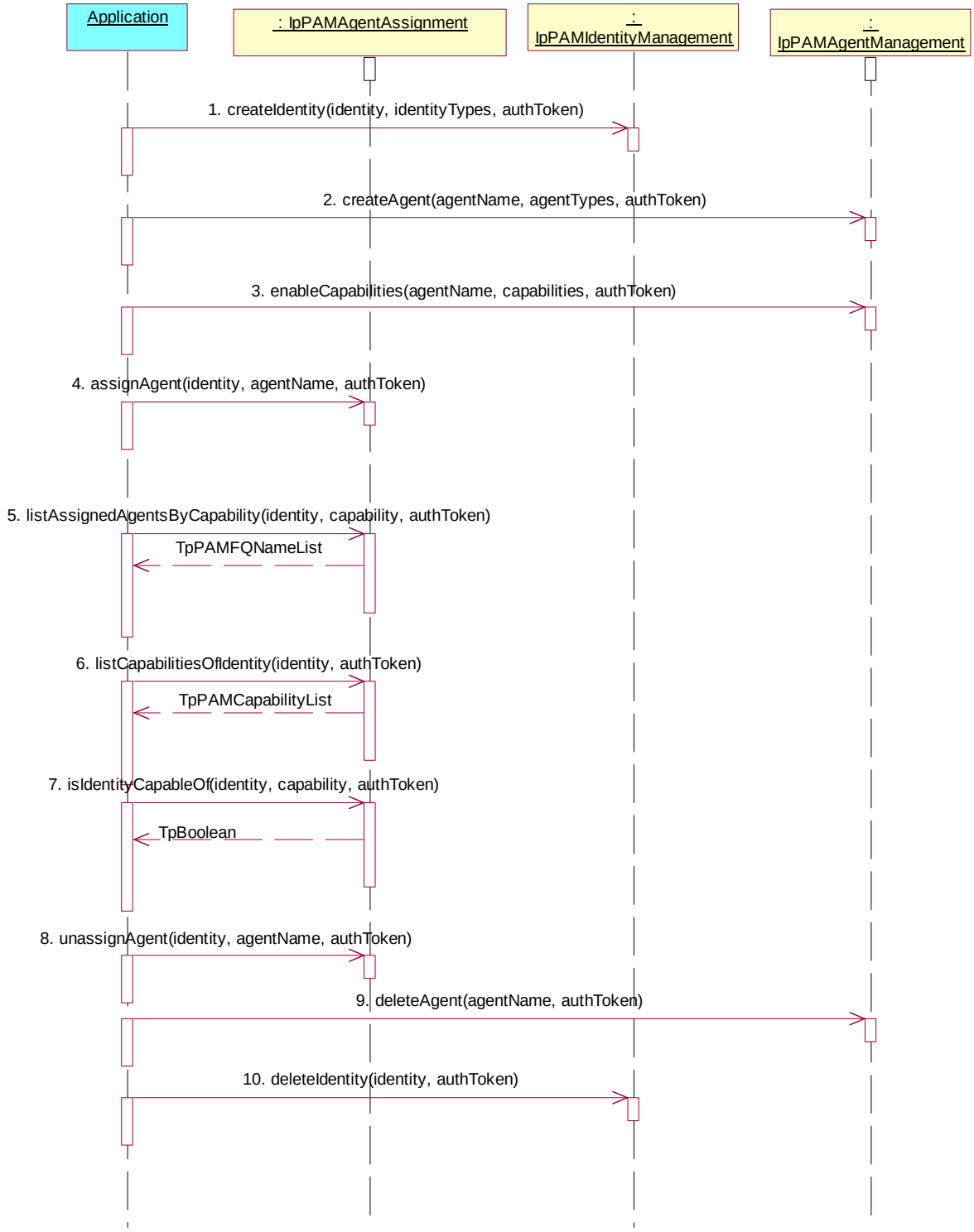
6. Method call **listCapabilitiesOfIdentity()**
   Parameters: identity given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMCapabilityList included capabilities given in 3.

7. Method call **isIdentityCapableOf()**
   Parameters: identity given in 1., one of a capability given in 3, authToken as obtained in Preamble
   Check: true is returned

8. Method call **unassignAgent()**
   Parameters: identity given in 1., agentName,given in 2., authToken as obtained in Preamble
   Check: no exception is returned

9. Method call **IpPAMAgentManagement.deleteAgent()**
   Parameters: same agentName than given in 1., authToken as obtained in Preamble
   Check: no exception is returned

10. Method call **IpPAMIdentityManagement.deleteIdentity()**
    Parameters: same identity than given in 1., authToken as obtained in Preamble
    Check: no exception is returned



5.2.1.3.5 IpPAMIdentityTypeManagement

Precondition: IpPAMIdentityTypeManagement supported.

Test PAM_PM_22

Summary: **IpPAMIdentityTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.5.

Preamble: **Preamble_getAuthToken_PM.**

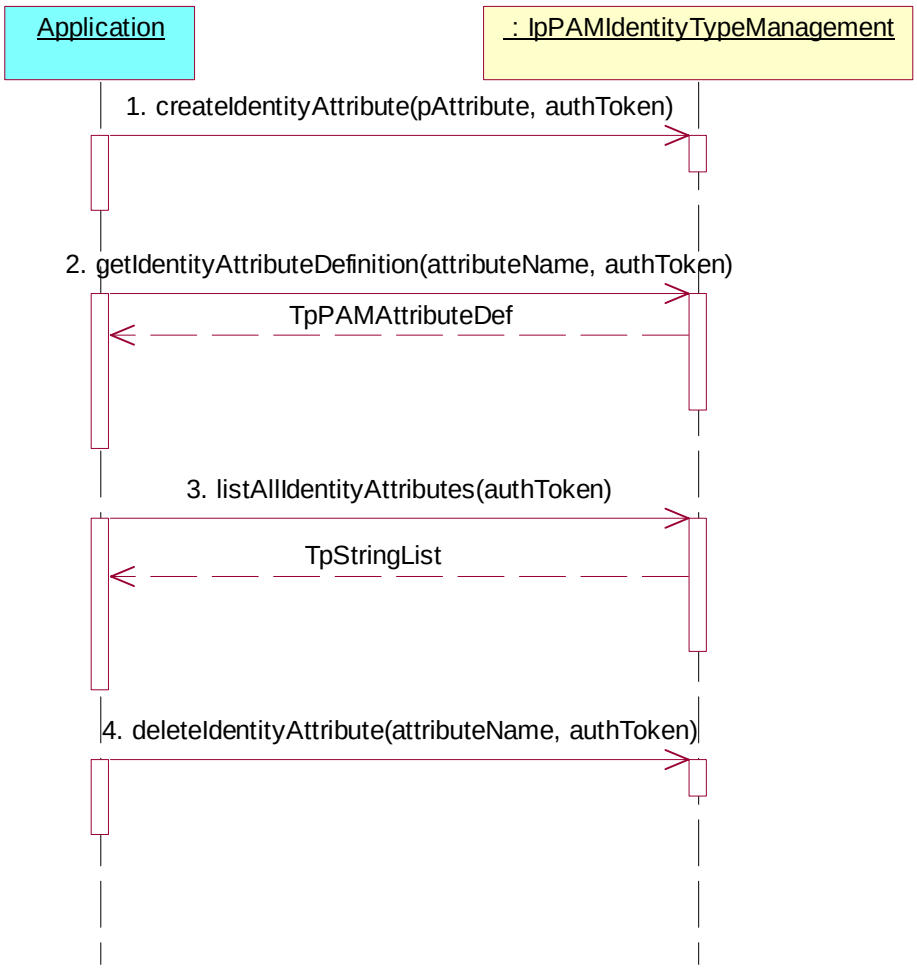
Test Sequence:

1. Method call **createIdentityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getIdentityAttributeDefinition()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeDef is returned

3. Method call **listAllIdentityAttributes()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned with AttributeName given in 1.

4. Method call **deleteIdentityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_23

Summary: **IpPAMIdentityTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.5.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

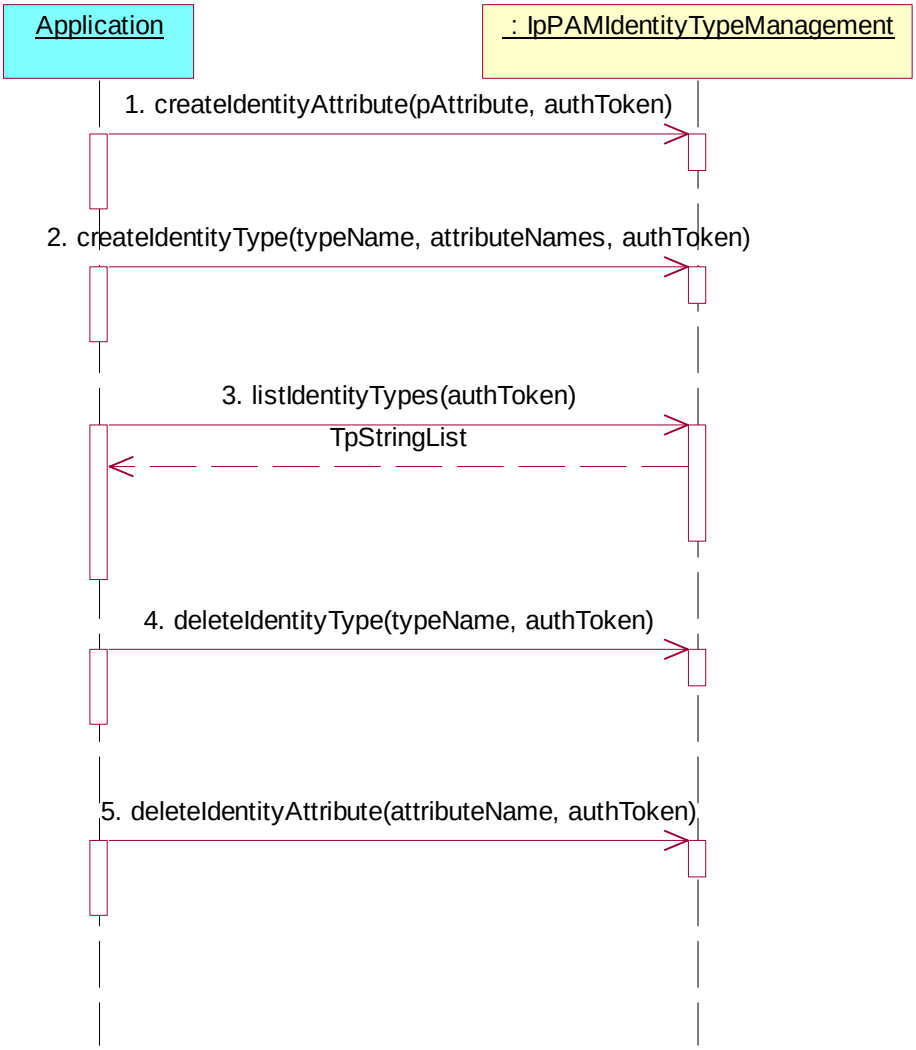
1. Method call **createIdentityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createIdentityType()**
   Parameters: typeName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listIdentityTypes()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned included typeName given in 2.

4. Method call **deleteIdentityType()**
   Parameters: typeName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **deleteIdentityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_24

Summary: **IpPAMIdentityTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.5.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createIdentityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createIdentityType()**
   Parameters: typeName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **createIdentityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

4. Method call **addIdentityTypeAttributes()**
   Parameters: typeName given in 1., attributesNames included attributeName given in 3.,authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **listIdentityTypeAttributes()**
   Parameters: typeName given in 1., authToken as obtained in Preamble
   Check: valid value of TpStringList is returned

6. Method call **removeIdentityTypeAttributes()**
   Parameters: typeName given in 1., attributesNames given in 4.,authToken as obtained in Preamble
   Check: no exception is returned

7. Method call **deleteIdentityType()**
   Parameters: typeName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **deleteIdentityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned

9. Method call **deleteIdentityAttribute()**
   Parameters: attributeName given in 3., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMIdentityTypeManagement
1. createIdentityAttribute(pAttribute, authToken)
2. createIdentityType(typeName, attributeNames, authToken)
3. createIdentityAttribute(pAttribute, authToken)
TpStringList
4. addIdentityTypeAttributes(typeName, attributeNames, authToken)
5. listIdentityTypeAttributes(typeName, authToken)
TpStringList
6. removeIdentityTypeAttributes(typeName, attributeNames, authToken)
7. deleteIdentityType(typeName, authToken)
8. deleteIdentityAttribute(attributeName, authToken)
9. deleteIdentityAttribute(attributeName, authToken)

5.2.1.3.6 IpPAMAgentTypeManagement

Precondition: IpPAMAgentTypeManagement supported.

Test PAM_PM_25

Summary: **IpPAMAgentTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.6.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgentAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getAgentAttributeDefinition()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeDef is returned

3. Method call **listAllAgentAttributes()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned with AttributeName given in 1.

4. Method call **deleteAgentAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned

Test PAM_PM_26

Summary: **IpPAMAgentTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.6.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

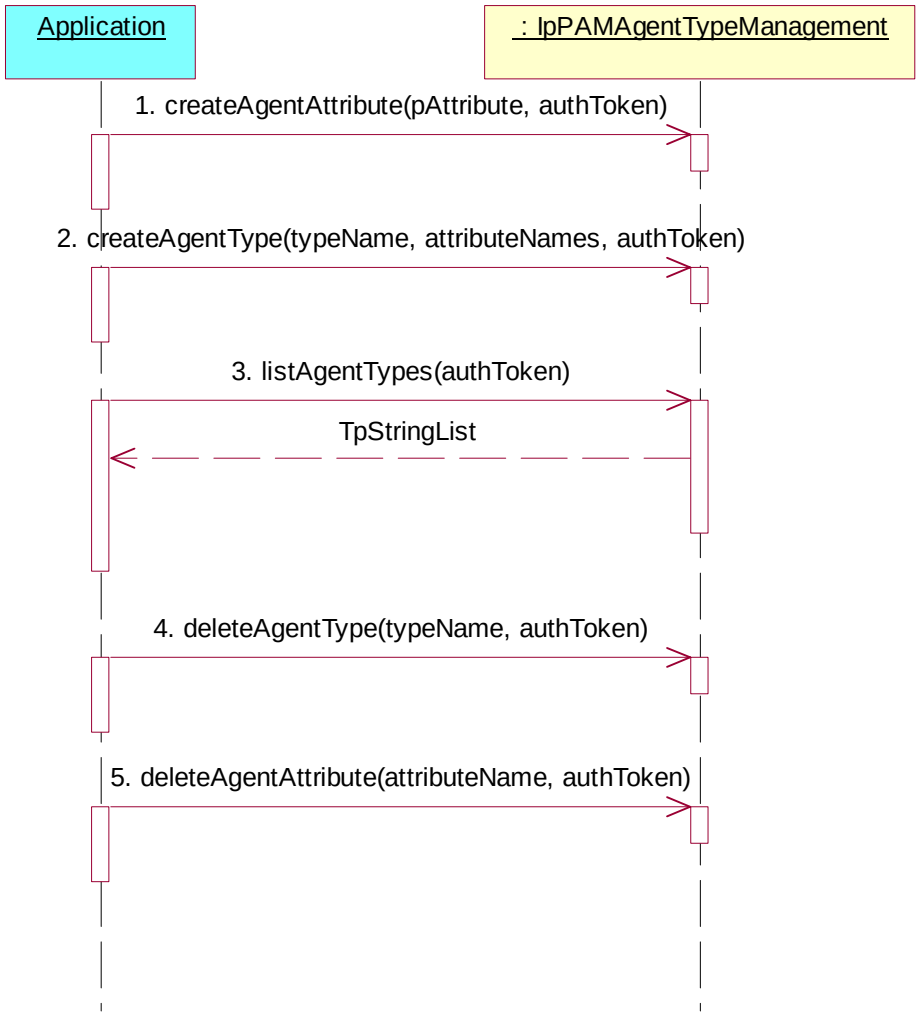
1. Method call **createAgentAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createAgentType()**
   Parameters: typeName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listAgentTypes()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned included typeName given in 2.

4. Method call **deleteAgentType()**
   Parameters: typeName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **deleteAgentAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_27

Summary: **IpPAMAgentTypeManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.6.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createAgentAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createAgentType()**
   Parameters: typeName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **createAgentAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

4. Method call **addAgentTypeAttributes()**
   Parameters: typeName given in 1., attributesNames included attributeName given in 3.,authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **listAgentTypeAttributes()**
   Parameters: typeName given in 1., authToken as obtained in Preamble
   Check: valid value of TpStringList is returned

6. Method call **removeAgentTypeAttributes()**
   Parameters: typeName given in 1., attributesNames given in 4., authToken as obtained in Preamble
   Check: no exception is returned

7. Method call **deleteAgentType()**
   Parameters: typeName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **deleteAgentAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned

9. Method call **deleteAgentAttribute()**
   Parameters: attributeName given in 3., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMAgentTypeManagement
1. createAgentAttribute(pAttribute, authToken)
2. createAgentType(typeName, attributeNames, authToken)
3. createAgentAttribute(pAttribute, authToken)
4. addAgentTypeAttributes(typeName, attributeNames, authToken)
5. listAgentTypeAttributes(typeName, authToken)
TpStringList
6. removeAgentTypeAttributes(typeName, attributeNames, authToken)
7. deleteAgentType(typeName, authToken)
8. deleteAgentAttribute(attributeName, authToken)
9. deleteAgentAttribute(attributeName, authToken)

5.2.1.3.7 IpPAMCapabilityManagement

Precondition: IpPAMCapabilityManagement supported.

Test PAM_PM_28

Summary: **IpPAMCapabilityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.7.

Preamble: **Preamble_getAuthToken_PM.**

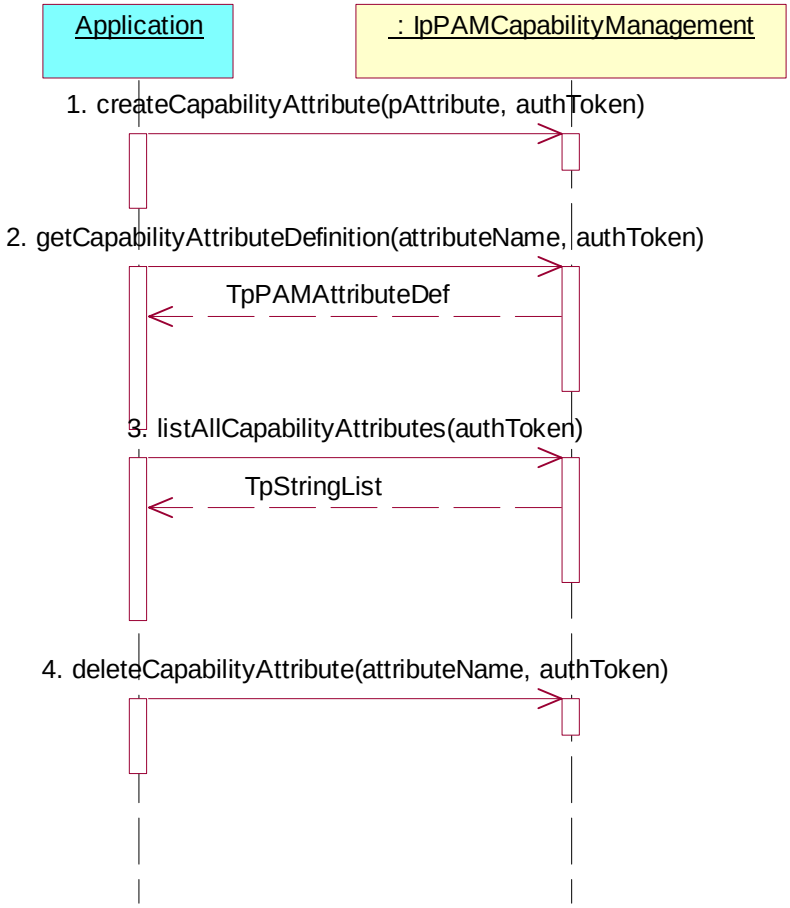
Test Sequence:

1. Method call **createCapabilityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **getCapabilityAttributeDefinition()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: valid value of TpPAMAttributeDef is returned

3. Method call **listAllCapabilityAttributes()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned with AttributeName given in 1.

4. Method call **deleteCapabilityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_29

Summary: **IpPAMCapabilityManagement** all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.7.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

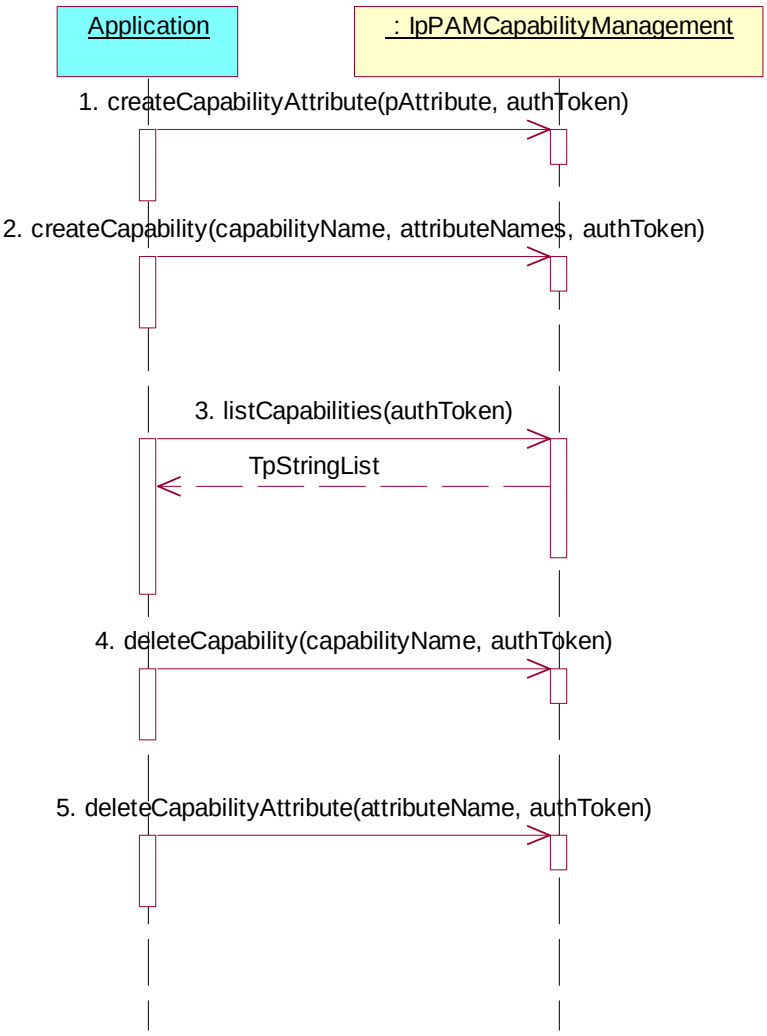
1. Method call **createCapabilityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createCapability()**
   Parameters: capabilityName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **listCapabilities()**
   Parameters: authToken as obtained in Preamble
   Check: valid value of TpStringList is returned included tCapabilityName given in 2.

4. Method call **deleteCapability()**
   Parameters: capabilityName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

5. Method call **deleteCapabilityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned



Test PAM_PM_30

Summary: **IpPAMCapabilityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.7.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createCapabilityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createCapability()**
   Parameters: capabilityName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **createCapabilityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

4. Method call **addCapabilityAttributes()**
   Parameters: capabilityName given in 1., attributesNames included attributeName given in 3., authToken as obtained in Preamble
   Check: no exception is returned

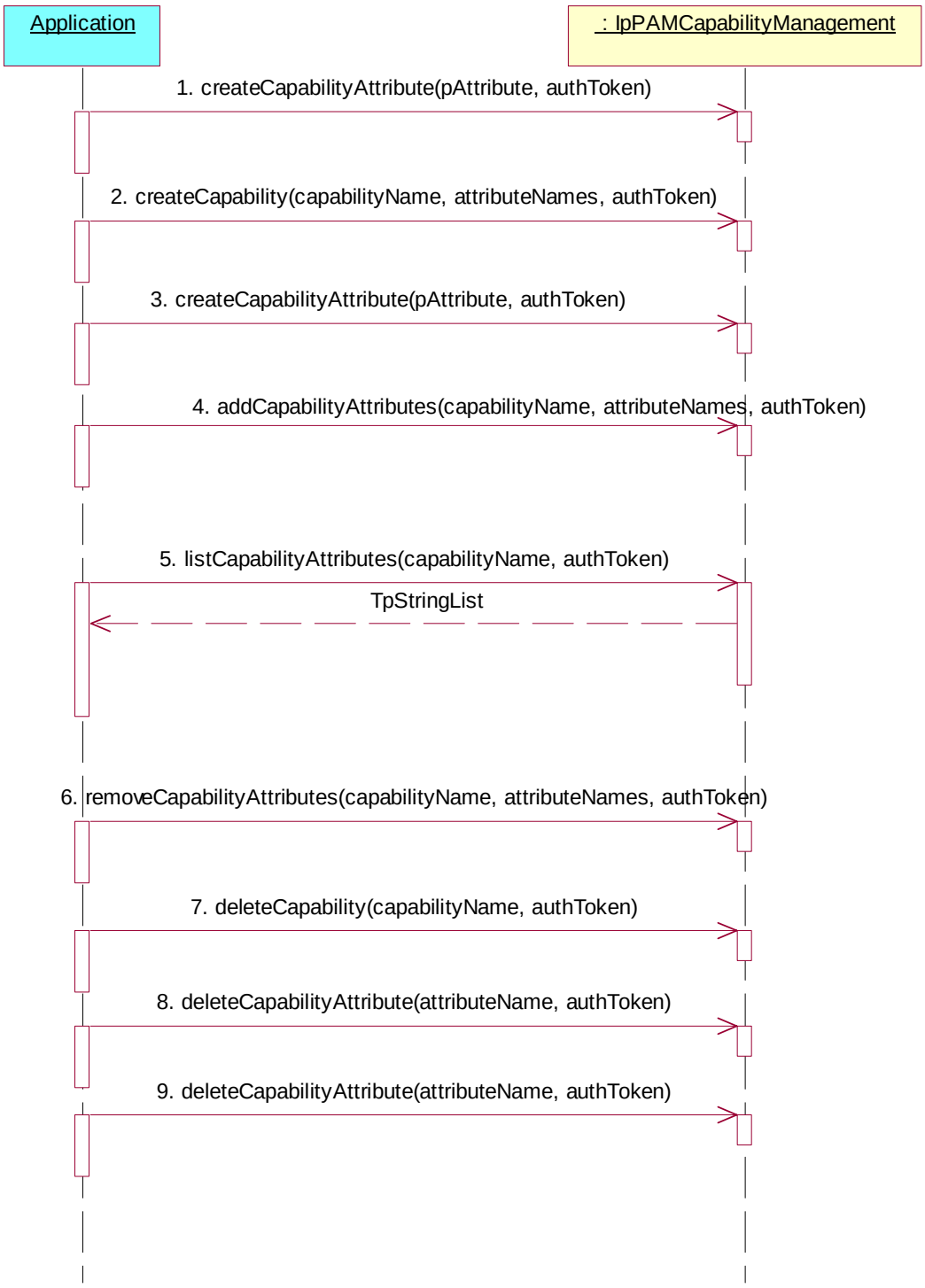
5. Method call **listCapabilityAttributes()**
   Parameters: capabilityName given in 1., authToken as obtained in Preamble
   Check: valid value of TpStringList is returned

6. Method call **removeCapabilityAttributes()**
   Parameters: typeName given in 1., attributesNames given in 4., authToken as obtained in Preamble
   Check: no exception is returned

7. Method call **deleteCapability()**
   Parameters: capabilityName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **deleteCapabilityAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned

9. Method call **deleteCapabilityAttribute()**
   Parameters: attributeName given in 3., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMCapabilityManagement
1. createCapabilityAttribute(pAttribute, authToken)
2. createCapability(capabilityName, attributeNames, authToken)
3. createCapabilityAttribute(pAttribute, authToken)
4. addCapabilityAttributes(capabilityName, attributeNames, authToken)
5. listCapabilityAttributes(capabilityName, authToken)
TpStringList
6. removeCapabilityAttributes(capabilityName, attributeNames, authToken)
7. deleteCapability(capabilityName, authToken)
8. deleteCapabilityAttribute(attributeName, authToken)
9. deleteCapabilityAttribute(attributeName, authToken)

Test PAM_PM_31

Summary: **IpPAMCapabilityManagement**, all methods successful.

Reference: ES 202 915-14 [1], clause 8.1.7.

Preamble: **Preamble_getAuthToken_PM.**

Test Sequence:

1. Method call **createCapabilityAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

2. Method call **createCapability()**
   Parameters: capabilityName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

3. Method call **IpPAMAgenTypeManagement.createAgentAttribute()**
   Parameters: pAttribute, authToken as obtained in Preamble
   Check: no exception is returned

4. Method call **IpPAMAgenTypeManagement.createAgentType()**
   Parameters: typeName, attributeNames, authToken as obtained in Preamble
   Check: no exception is returned

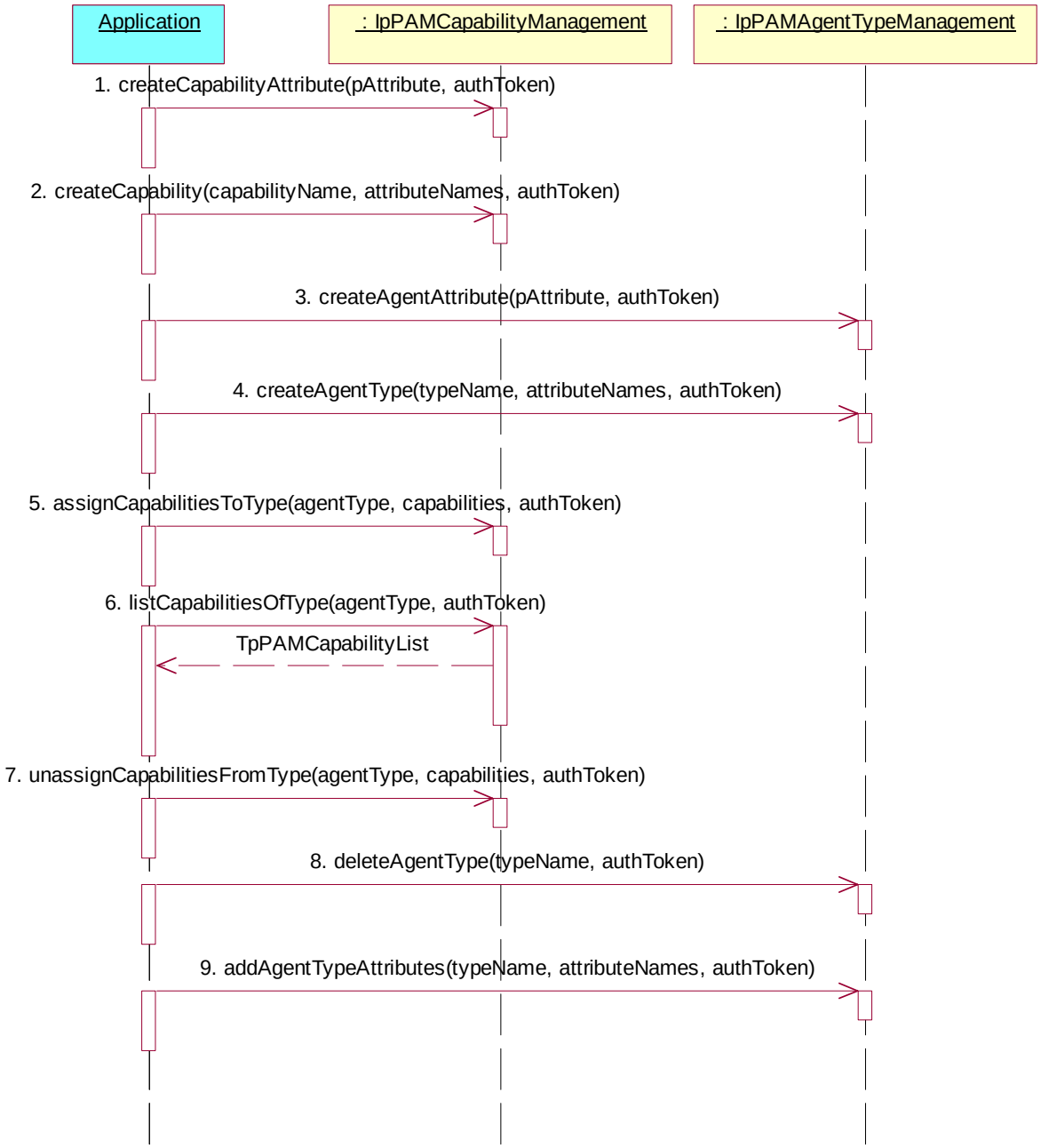
5. Method call **assignCapabilitiesToType ()**
   Parameters: agentType given in 4., capabilities included capability given in 2., authToken as obtained in Preamble
   Check: no exception is returned

6. Method call **listCapabilitiesofType ()**
   Parameters: agentType given in 4., authToken as obtained in Preamble
   Check: valid vale of TpPAMCapabilityList is returned

7. Method call **unassignCapabilitiesToType ()**
   Parameters: agentType given in 4., capabilities included capability given in 2., authToken as obtained in Preamble
   Check: no exception is returned

8. Method call **IpPAMAgenTypeManagement.deleteAgentType()**
   Parameters: typeName given in 2., authToken as obtained in Preamble
   Check: no exception is returned

9. Method call **IpPAMAgenTypeManagement.deleteAgentAttribute()**
   Parameters: attributeName given in 1., authToken as obtained in Preamble
   Check: no exception is returned


Application
: IpPAMCapabilityManagement
: IpPAMAgentTypeManagement
1. createCapabilityAttribute(pAttribute, authToken)
2. createCapability(capabilityName, attributeNames, authToken)
3. createAgentAttribute(pAttribute, authToken)
4. createAgentType(typeName, attributeNames, authToken)
5. assignCapabilitiesToType(agentType, capabilities, authToken)
6. listCapabilitiesOfType(agentType, authToken)
TpPAMCapabilityList
7. unassignCapabilitiesFromType(agentType, capabilities, authToken)
8. deleteAgentType(typeName, authToken)
9. addAgentTypeAttributes(typeName, attributeNames, authToken)

History

Document history			
V1.1.1	January 2005	Membership Approval Procedure	MV 20050311: 2005-01-11 to 2005-03-11