Policy Evaluation, Enforcement and Management Requirements

Approved Version 1.0 – 24 Jul 2012

Open Mobile Alliance

OMA-RD-Policy_Evaluation_Enforcement_Management-V1_0-20120724-A

Contents

Figures

1. Scope (Informative)

This document provides use cases and requirements for policy evaluation, enforcement and management, (PEEM) within OMA.

The PEEM enabler evaluates, (or evaluates and executes) policies. Policies are applied to requests to, or responses from resources or, when explicitly called by a resource.

The requirements in this document are intended to facilitate the development of a set of specifications for defining, exposing, managing, evaluating, and executing policies in a way that is scalable and flexible yet independent of any specific implementation scheme.

Tools to translate enabler specific local policies into the language specified by PEEM may be needed but are out of scope of the PEEM specification. Note also that this RD does not specify individual policies, but rather addresses requirements on how to express policies.

2. References

2.1 Normative References

[ARCH]	"OMA Architecture Requirements, Version 1.0", Open Mobile Alliance™, OMA-RD_Architecture-V1_0 URL: http://www.openmobilealliance.org/
[OMA Dict]	"Dictionary for OMA Specifications", Version 2.6, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_6, URL: http://www.openmobilealliance.org/
[PEEM_AD]	"Policy Evaluation, Enforcement and Management Architecture", Version 1.0, Open Mobile Alliance™, OMA-AD_Policy_Evaluation_Enforcement_Management-V1_0, URL:http://www.openmobilealliance.org/
[PEEM_Callable_Policy_Interface]	"Policy Evaluation, Enforcement and Management Callable Interface (PEM-1) Technical Specification", Verison 1.0, Open Mobile Alliance™, OMA-TS-PEEM_PEM1-V1_0, URL: http://www.openmobilealliance.org/
[PEEM_Management_Interface]	"Policy Evaluation, Enforcement and Management – Management Interface (PEM-2) Technical Specification", Version 1.0, Open Mobile Alliance™, OMA-TS-PEEM_PEM2-V1_0, URL: http://www.openmobilealliance.org/
[PEEM_Policy_Rule_Language]	"PEEM Policy Expression Language Technical Specification", Open Mobile Alliance™, Verison 1.0, OMA-TS-PEEM_PEL-V1_0, URL: http://www.openmobilealliance.org/
[Privacy]	"Privacy Requirements for Mobile Services", Version 1.0, ", Open Mobile Alliance™, OMA-RD_Privacy-V1_0, URL; http://www.openmobilealliance.org/
[RFC 2119]	"Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt
[RFC 3060]	"Policy Core Information Model (PCIM) Extensions" http://www.ietf.org/rfc/rfc3460.txt
[RFC 3198]	"Terminology for Policy-Based Management" http://www.ietf.org/rfc/rfc3198.txt
[RFC 3460]	"Policy Core Information Model (PCIM) Extensions" http://www.ietf.org/rfc/rfc3460.txt

2.2 Informative References

None

3. Terminology and Conventions

3.1 Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All sections and appendixes, except “Scope” and “Introduction”, are normative, unless they are explicitly indicated to be informative.

3.2 Definitions

Common Functions	See [OMA Dict]
Delegate	A delegate is a designated resource that performs specified tasks or functions on behalf of (one or more) other resources. To *delegate* is to designate a resource to perform specified tasks or functions on behalf of (one or more) other resources.
Home Network	The network in which the user’s device is subscribed
Policy	An ordered combination of policy rules that defines how to administer, manage, and control access to resources, [Derived from [RFC 3060], [RFC 3198] and [RFC 3460].
Policy Action	Action (e.g. invocation of a function, script, code, workflow, …) that is associated to a policy condition in a policy rule and that is executed when its associated policy condition results in "true" from the policy evaluation step.
Policy Condition	A condition is any expression that yields a Boolean value.
Policy Enforcement[1]	The process of executing actions, which may be performed as a consequence of the output of the policy evaluation process or during the policy evaluation process.
Policy Engine	A logical entity that evaluates a policy or policies.
Policy Evaluation	The process of evaluating the policy conditions and executing the associated policy actions up to the point that the end of the policy is reached.
Policy Execution	Execution of the action associated to the policy condition selected by policy evaluation
Policy Expression	The process of representing a policy
Policy Expression Language	The language to express policies
Policy Management	The act of describing, creating, updating, deleting, provisioning and viewing policies.
Policy Rule	A combination of **a** condition and **an** action to be performed if the condition is true
Principal	See [OMA Dict]
Request	An articulation of the need to access a resource (e.g. asynchronous events).
Requestor	Any entity that issues a request to a resource.
Responder	Resource that is the target of a request.
Response	An articulation of the results of the processing of a request.
Resource	Any component, enabler, function or application that can receive and process requests.
Visited Network	Any network other than the subscriber’s home network

3.3 Abbreviations

APP	Application
ASP	Application Service Provider
EN	Enabler Implementation
HMO	Home Mobile Operator

ISDN	Integrated Services Digital Network
MMO	Multi-network Mobile Operator
MO	Mobile Operator
MSISDN	Mobile Subscriber ISDN Number
OMA	Open Mobile Alliance
PDP	Policy Decision Point, [RFC 3198]
PEEM	Policy Evaluation, Enforcement and Management
PEP	Policy Enforcement Point, [RFC 3198]
SLA	Service Level Agreement
SMS	Short Message Service
VMO	Visited Mobile Operator

4. Introduction (Informative)

Mobile service environments where different entities, e.g. enterprises, mobile operators, service providers and 3rd party service providers collaborate to provide highly personalised services to mobile subscribers present new opportunities and benefits to the mobile industry. Policy Evaluation, Enforcement and Management, (PEEM) is driven by the need to reduce management complexity whilst introducing consistent new subscriber services with the same or reduced time to market.

Policies are formalisms that are used to express business, engineering or process criteria represented by a combination of conditions and actions. PEEM specifies ways to convey and enforce policies that can be used to manage resources, processes and underlying systems. OMA enablers are expected to re-use PEEM concepts in order to avoid duplication and misalignment. The aim of this document is to collect requirements on a PEEM enabler.

PEEM also enables the delegation of responsibility to other resources:

- This can help avoid the costly duplication of functionality across service enablers and reduce the proliferation of 'silos' in service provider networks;
- This is expected to be an efficient mechanism to re-use resources by providing a systematic way to express and implement the delegation to such other resources.

Policies are associated with resources, and/or requestors and/or requests. Whenever requests are made to a resource, the associated policies are evaluated and enforced by a policy enforcement mechanism on the request and on the associated response.

The PEEM enabler can be used as a function that can be explicitly called by other resources:

- To protect the resource and therefore facilitate its exposure by the service provider;
- To realise workflow or composition.

This requirements document is expected to be neutral in terms of implementation and deployments.

4.1 Actors in the context of PEEM

Figure 1 illustrates the main stakeholders in the context of PEEM.

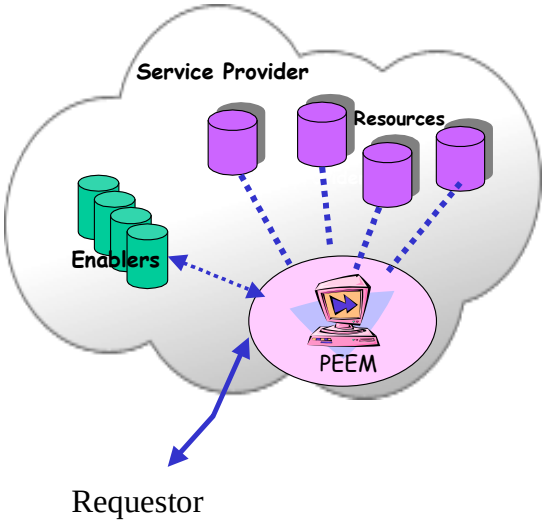

Figure 1: Actors in the context of PEEM

The PEEM enabler invokes relevant policies to process the request and if successful it passes the request to the target resource.

The following discusses further the various actors in OMA and how they view the PEEM enabler.

4.1.1 End Users

The end user wants to personalise his/her services and express his/her preferences at a high level utilising some personalization application (e.g. web-based forms with tables, pull-down menus, etc) in order to use those preferences across available services. In the service provider environment, preferences may be expressed as policies. These policies are applied whenever requests are made that could be affected by those preferences.

4.1.2 Operators

The Operator wants flexible service management that helps managing the requests to resources and protecting the integrity of its resources. Being able to expose resources in a manageable, secure, billable[1], auditable and automatable manner is a key requirement. The PEEM enabler allows Operators to enhance their service portfolio and encourages the uptake of services by other providers.

4.1.3 Service Providers

The service provider wants flexible service management that helps managing the requests to resources and protecting the integrity of its resources. Being able to expose resources in a manageable, secure, billable, auditable and automatable manner is a key requirement. The PEEM enabler allows service providers to enhance their service portfolio and encourages the uptake of services by other providers.

4.1.4 3rd Party Service Providers

The 3rd Party Service Providers will exploit features by accessing resources exposed by Service Providers or Operators. Therefore, the ability to access resources in a secure and automatable way is a key requirement for 3rd Party Service Providers who want to access features required by their deployed applications. Another important requirement is to be able to access resources from multiple Service Providers and Operators.

3rd party service providers can also delegate the PEEM functionality to another service provider:

- To protect their own resources;
- To impose constraints on usage of certain services by some users (e.g. restrict what an employee can do);

To allow 3rd parties to provide policy for accessing hosted resources.

4.1.5 Application Developers

Applications can be developed tailored to specific end-users using policies. The Applications will also exploit network features by accessing resources exposed in the network. Therefore, the ability to request resources in a standardised way is a key requirement for application developers who want to add features to their applications. Another important requirement is to be able to deploy the enabled applications with many Service Providers and 3rd Party Service Providers.

4.2 User settings and resource protection

It is important to note that the PEEM enabler can be used to:

- Protect a resource by ensuring that policies are evaluated and enforced for any message to and from the resource;

[1] Notions of allowing billable and auditable exposure of resources used throughout this document should be understood as examples of conditions that can be enforced before allowing access and usage of the resource. It does not imply that billing must always take place, nor that this is the sole type of condition can be enforced besides authentication, authorization etc… This comments applies throughout this document.

- Allow evaluation and enforcement of preferences or user settings expressed as policies;
- Perform policy evaluation (or evaluation and execution) on request;
- Simplify implementation of resources by allowing delegation to other resources.

4.3 PEEM, Other Enablers and Integration

By providing mechanisms to implement the delegation of responsibility for certain functions the PEEM enabler provides alternative ways to:

- Use other enablers;
- Facilitate integration of resources that can share or reuse other resources.

Because delegation to other enablers is by definition common across most use cases, the PEEM enabler can itself be considered as a key enabler.

4.4 Usage Patterns for PEEM

In this requirement document, two major usage patterns have been identified for PEEM: PEEM as a proxy and a callable PEEM. The following figures provide logical illustrations of these patterns.

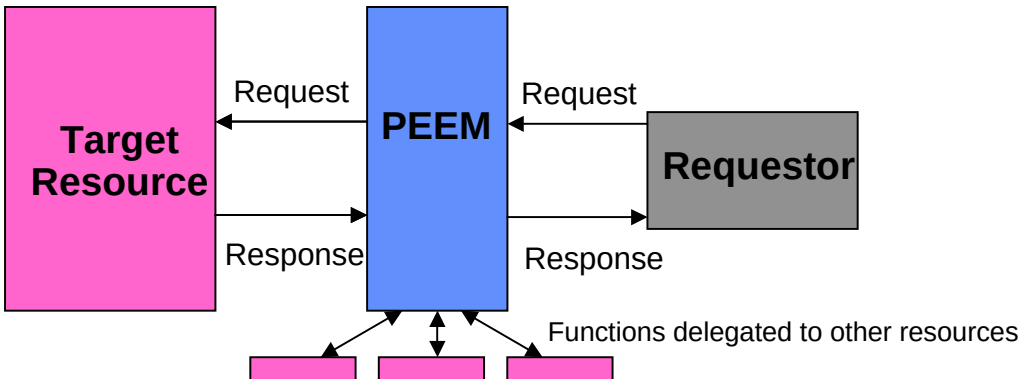

Figure 2: Logical illustration of PEEM as proxy usage pattern.

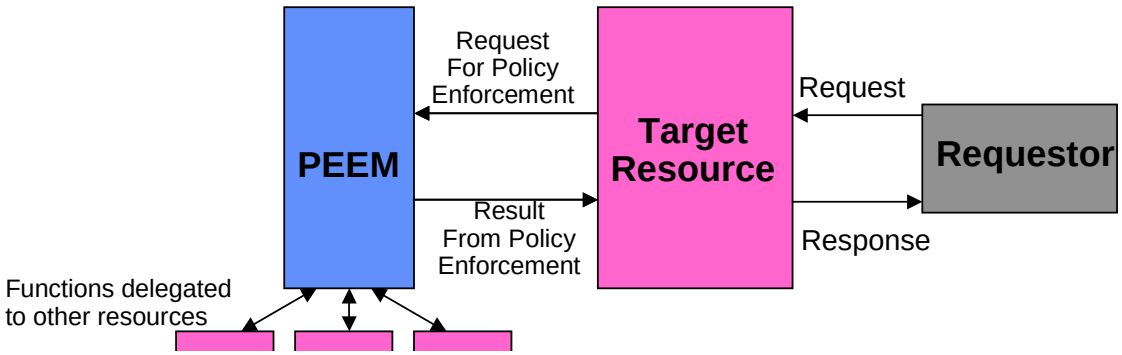

Figure 3: Logical illustration of callable PEEM usage pattern.

Note, a request can be for Policy Execution only, Policy Evaluation only, or for both.

5. Use Cases (Informative)

5.1 Typical Flow in a PEEM Use Case

This section provides an overview of the typical flow associated to the use of the PEEM enabler:

- Owner of a resource protects the resource with a PEEM enabler using a particular valid implementation and deployment model of the PEEM enabler. Examples are discussed in section 4.4;
- Owner of a resource establishes the policy associated to it;
- Owner publishes/registers policy somewhere;
- Requestor discovers (or knows) resource;
- Requestor knows the conditions it must satisfy (e.g. via Service Level Agreement (SLA)[2]);
- Requestor may take prior steps to satisfy the conditions that he/she knows for using the resource;
- Requestor prepares request to resource and provides information/meta-data/credentials to be able to satisfy the conditions that he/she knows for using the resource;
- Policy enforcement is performed on the request;
- Request is passed to resource for action (assuming successful validation of all the steps)[3];
- If specified by policy, response may be similarly processed before being passed to requestor. In such a case, the requestor may also add a policy to apply on the response before letting it reach it (e.g. authentication of the source – i.e. the original responder).

The use cases that follow are considered to provide a good representation of the basic functionality of the PEEM enabler. Other use cases that describe this functionality further can be found in Appendix B.

5.2 Location Application with User Interaction: Location obtained in the phone

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)	X		X	X	
Additional Keywords					

5.2.1 Short Description

User A uses his mobile terminal to obtain User B location through a 3rd party application.

In this Use Case User B handset has the capability to calculate his position.

Policies set by end user B require that explicit confirmation is needed before giving his position to any user/application.

[2] In a non-typical variation of the flow, the requestor could discover the conditions that it must satisfy.

[3] In case of failure of some of the execution or validation steps, the request may be returned with the requestor with an error or a dialog may be established between the requestor and one of the involved intermediate resources (e.g. a new prompt / challenge).

5.2.2 Actors

- User A;
- User B;
- Mobile Operator – acting as network provider and service provider for both end users:
  - Network Provider;
  - Service Provider;
- 3rd party service provider;
- 3rd Party application.

5.2.2.1 Actor Specific Issues

- The Mobile Operator is the Service Provider for end-user A and end-user B;
- The Mobile Operator is the owner of the resources that are protected by PEEM:
  - Enforcing the policies;
  - Providing access to its resource;
  - Enforcing the SLA between the Third Party Service Provider and itself;
  - Negotiates (possibly as part of SLA) billing and interconnect charges between the Third Party Service Provider and itself;
  - Coordinates with the Third Party Service Provider the correct charging of events.
- The Third Party Service Provider is the owner of the Location Application:
  - Uses MO resources to get user B's location;
  - Providing necessary credentials to access the resource;
  - End-user A subscribes to the Location Application through the Mobile Operator. His subscription is paid to the Mobile Operator.

5.2.2.2 Actor Specific Benefits

- The Mobile Operator:
  - Can offer access to resource and its use while enforcing conditions of usage expressed in policies;
  - Knows that resource is appropriately protected;

Makes available a wider range of Applications to their customer base;

- Third Party Service Provider:
  - Can access MO resources to use for their Applications;
  - Can simplify and automate the way to use a resource belonging to the Mobile Operator: need only to know what credential to pass and how.

5.2.3 Pre-conditions

- End-user A and B must have a subscription with a Mobile Operator. In this use-case both A and B are subscribed to the same Mobile Operator;
- End-user A and B may either have a post-paid or prepaid subscription with the Mobile Operator;
- End-user A must have a subscription with the Location Application. Both End-user A and B are identified via a valid address (MSISDN) or an Alias ID or session ID;
- Third Party Service Provider has a contractual agreement with the Mobile Operator. This contractual agreement covers aspects such as terms and conditions, establishing payment method for application consumption and establishing privacy settings if applicable;
- Third Party Service Provider Application is registered with the Mobile Operator and is allowed to submit a "location_request" containing an end-user's identity, e.g. Alias ID;
- The location being requested must be that of a network attached mobile end-user terminal;
- Privacy preference for the targeted end-user (end-user B), and local government legislation must be maintained by the Mobile Operator;
- If end-user A and B are roaming, service experience is not impacted. However, possibly extra charging for message reception must done with the methods employed in the same situation as those described below, i.e. different scenarios should not require special functionality;
- The submission of the message is charged;
- User's B handset is able to calculate its position and to give it to the network, under request;
- User B tells the operator (i.e., via call centre or a web interface) to set and enforce a policy saying that, if somebody asks for his location, he must be asked explicitly for permission (i.e. via a SMS).

5.2.4 Post-conditions

- End-user A is presented with the location of end-user B;
- End-user A is correctly charged with the service event;
- Mobile Operator and Third Party Service Provider are able to fulfil their SLA agreements, i.e. charging and billing between the two parties is correctly handled.

5.2.5 Normal Flow

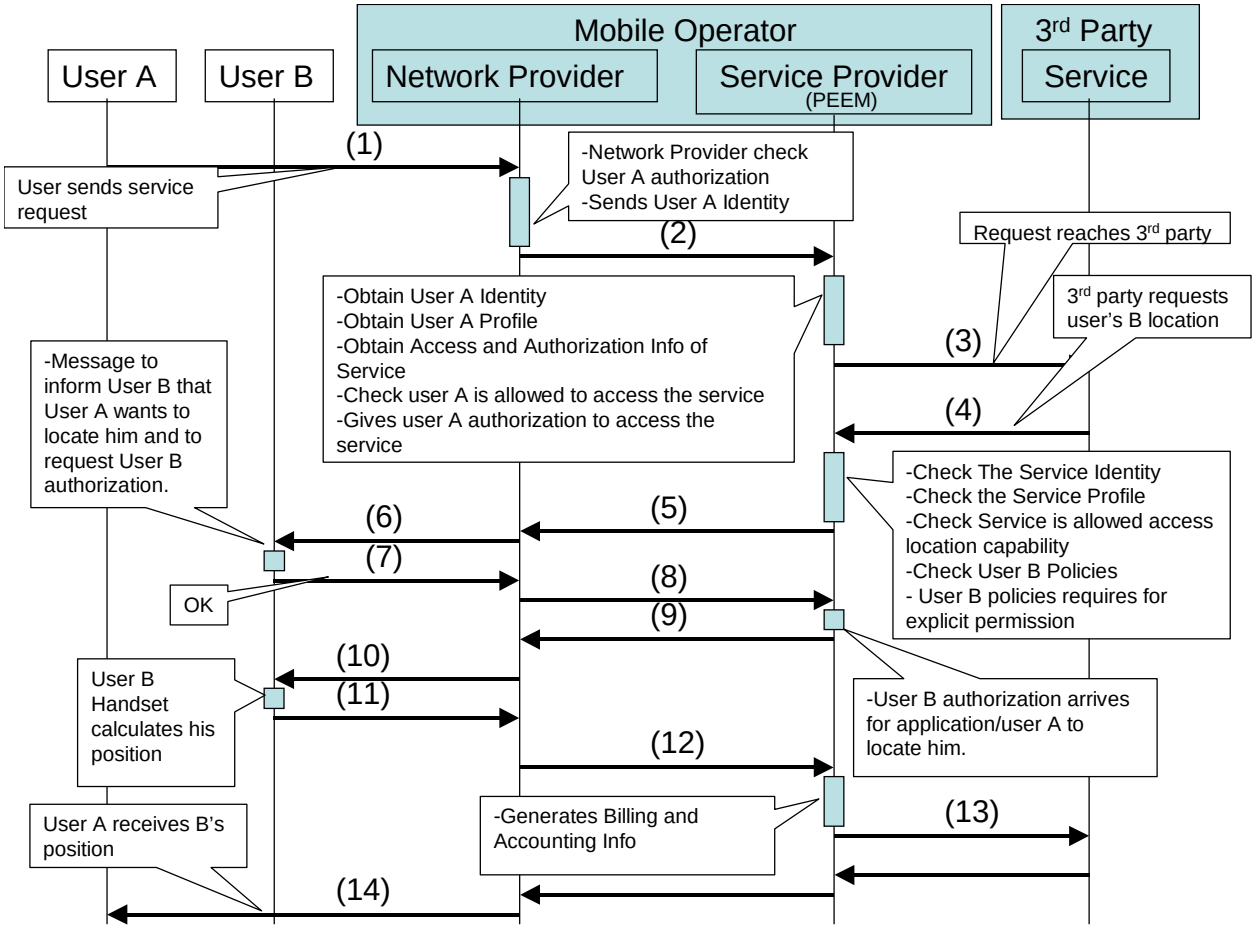

Figure 4: Location Application with user interaction

(1) User A accesses the MO Network in order to access the Service through the Service Provider:

- MO's network authenticates the user (normally done formerly in the attach procedure);
- MO's network sends the user identity (e.g. MSISDN) to the service layer.

(2) MO's Service Provider receives the request via the Network. At this point MO has to:

- Obtain user A Identity;
- Obtains user A Profile;
- Obtains Access & Authorization info for the location application;
- Check user A is allowed to access the location application;
- Gives user A authorization to access the location application.

(3) MO forwards request to the 3rd party location application:

- In order to be able to fulfil the request from User A, 3rd party application must know the position of User B.

(4) 3rd party application requests MO the location of User B. At this moment Service Provider has to:

- Check the identity of the 3rd party;
- Obtain his profile;
- Check 3rd party application is allowed to access the Location Capability;
- Check if User B location information is allowed to be accessed by user A through the location application:
  i. Check user B privacy policies;
  ii. Check regulatory policies;
  iii. Check operator policies.
- Check if User B Handset has the ability to give the Network its position;
- Evaluating policies we have that "User A only can locate User B if User B gives his authorization".

(5) MO uses his messaging capabilities to send a message to inform User B that User A wants to locate him;

(6) Network sends the message to User B;

(7) User B responds to this message and allows application/User A to locate him;

(8) Confirmation reaches the PEEM in the MO Service Provider;

(9) MO requests User B location to its network;

(10) MO's network requests User B handset its position;

(11) User B handset gives its position to Network;

(12) Network gives the location information of User B to the MO Service Provider. MO generates Billing and Accounting Info;

(13) MO gives the location of User B to the 3rd party application;

(14) 3rd party application gives the location information of User B to User A through the MO via its Network.

5.2.6 Alternative Flow

None Identified.

5.2.7 Operational and Quality of Experience Requirements

- The Mobile Operator is able to set different levels of authorisation for accessing different levels of their resource;
- 'The user shall have full control over his personal data;
- The Mobile Operator shall adhere to local government legislation;
- User A experience must not be degraded by the policy enforcement mechanisms or charging mechanisms;
- PEEM shall be able to deal with the absence of a response.

5.3 Location Application. Multi-Network scenario

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X	X	
Additional Keywords					

5.3.1 Short Description

User A uses his mobile terminal to obtain User B location through a 3rd party application.

User A belongs to network provider A, and user B belongs to network provider B.

Both network providers share the same service provider (for location services at least).

Network Providers A and B, as well as Service Provider, are part of the same Mobile Operator (Multi-network Mobile Operator).

Application belongs to a 3rd party, which offers it through the Service Provider.

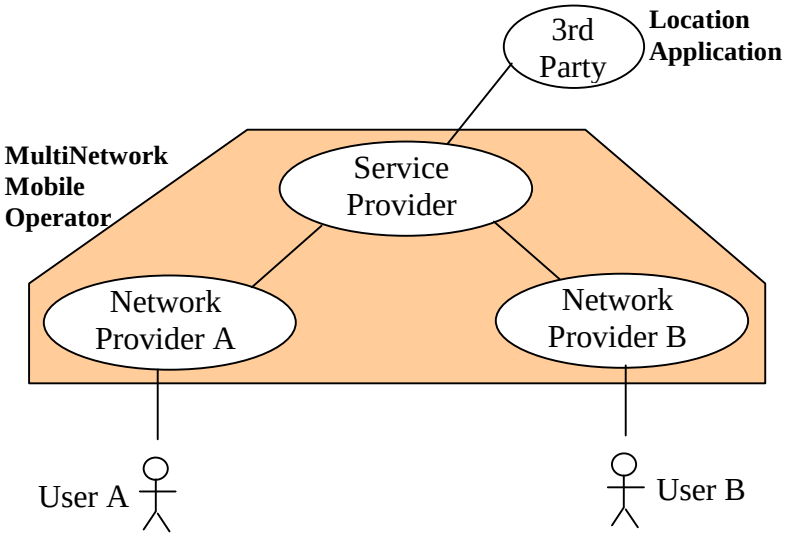

Figure 5: Multi-network Scenario

5.3.2 Actors

- User A;
- User B;
- Multi-network Mobile Operator (MMO or Mobile Operator from now on) – acting as network provider and service provider for both end users:

  - Network Provider A;
  - Network Provider B;
  - Service Provider.
- 3rd party service provider;
- 3rd Party Location application.

5.3.2.1 Actor Specific Issues

- The MMO is the Service Provider for end-user A and end-user B;
- The MMO is the owner of the resource:
  - Enforcing the execution policies;
  - Providing access to its resource;
  - Enforcing the SLA between the Third Party Service Provider and itself;
  - Negotiates (possibly as part of SLA) billing and interconnect charges between the Third Party Service Provider and itself;
  - Coordinates with the Third Party Service Provider the correct charging of events.
- Mobile Operator offers services through different network providers;
- The Third Party Service Provider is the owner of the Location Application:
  - Uses MO resources to get user B's location;
  - Providing necessary credentials to access the resource.
- End-user A subscribes to the Location Application through the Mobile Operator. His subscription is paid to the Mobile Operator;
- The two network providers could be a 2G and a 3G networks operating in the same country and belonging to the same mobile operator, or it could be two 2G mobile networks operating in different countries, but belonging to the same global operator.

5.3.2.2 Actor Specific Benefits

- The Mobile Operator:
  - Can offer access to resource and its use while enforcing conditions of usage expressed in execution policies;
  - Knows that resource is appropriately protected;
  - Make available a wider range of Applications to their customer base;
  - Share policy infrastructure when delivering services through various network providers.
- Third Party Service Provider:

  - Can access MO resources to use for their Applications;
  - Can simplify and automate the way to use a resource belonging to the Mobile Operator: need only to know what credential to pass and how.

- End Users:
  - End User B sees its privacy policies successfully applied.
- End User A can access properly the services he's authorized to use.

5.3.3 Pre-conditions

- End User A is subscribed to Network Operator A;
- End User B is subscribed to Network Operator B;
- End-user A and B are subscribed to the same service provider;
- Service Provider is delivering services to both network providers;
- End-user A and B may either have a post-paid or prepaid subscription with the Mobile Operator;
- End-user A must have a subscription with the Location Application. Both End-user A and B are identified via a valid address (MSISDN) or an Alias ID or session ID;
- Third Party Service Provider has a contractual agreement with the Mobile Operator. This contractual agreement covers aspects such as terms and conditions, establishing payment method for application consumption and establishing privacy settings if applicable;
- Third Party Service Provider Application is registered with the Mobile Operator and is allowed to submit a "location_request" containing an end-user's identity, e.g. Alias ID;
- The location being requested must be that of a network attached mobile end-user terminal;
- Privacy preference for the targeted end-user (end-user B), and local government legislation must be maintained by the Mobile Operator;
- If end-user A and B are roaming, service experience is not impacted. However, possibly extra charging for message reception must done with the methods employed in the same situation as those described below, i.e. different scenarios should not require special functionality;
- The submission of the message is charged;
- User B tells the operator (i.e., via call centre or a web interface) to set and enforce a policy saying that, if somebody asks for his location, he must be asked explicitly for permission (i.e. via a SMS);
- Service Provider manages policies for its subscribers.

5.3.4 Post-conditions

- End-user A is presented with the location of end-user B;
- End-user A is correctly charged with the service event;
- Mobile Operator and Third Party Service Provider are able to fulfil their SLA agreements, i.e. charging and billing between the two parties is correctly handled;
- Privacy policies of user B have been properly applied.

5.3.5 Normal Flow

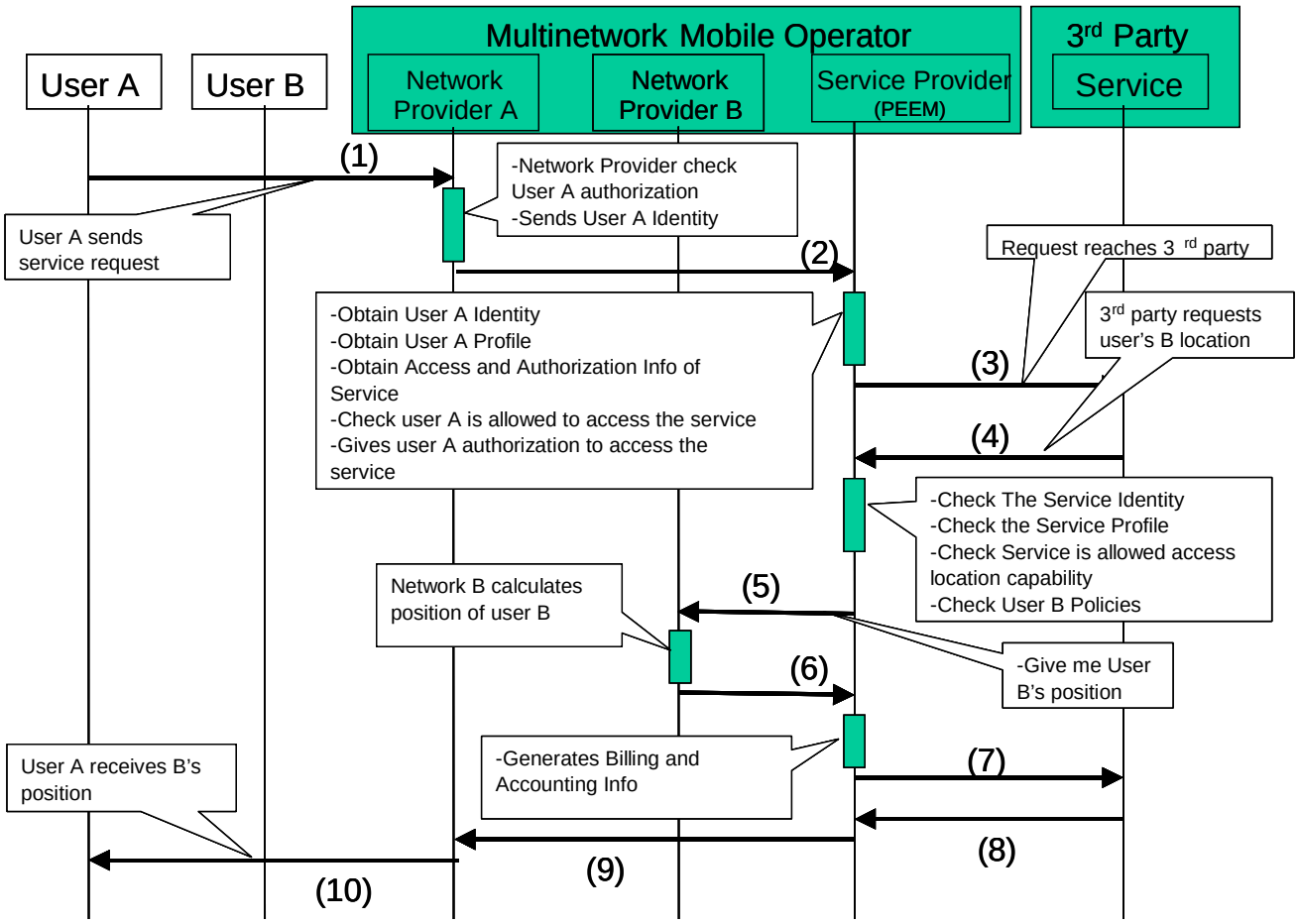

Figure 6: Multi-network Scenario Flow

1. User A accesses the Network A in order to access the Service through the Service Provider:
    - MO's network authenticates the user (normally done formerly in the attach procedure);
    - MO's network sends the basic user identity to the service layer.
2. MO's Service Provider receives the request via the Network A. At this point MO has to:
    - Obtain user A Identity;
    - Obtains user A Profile;
    - Obtains Access & Authorization info for the location application;
    - Check user A is allowed to access the location application;
    - Gives user A authorization to access the location application.
3. MO forwards request to the 3rd party location application:
    - In order to be able to fulfil the request from User A, 3rd party application must know the position of User B.

4. 3rd party application requests MO the location of User B. At this moment Service Provider has to:
    - Check the identity of the 3rd party;
    - Obtain his profile;
    - Check 3rd party application is allowed to access the Location Capability;
    - Check if User B location information is allowed to be accessed by user A through the location application:
        - Check user B privacy policies;
        - Check regulatory policies;
        - Check operator policies.
5. MO requests User B location to network B:
    a. Network B calculates position of user B.
6. Network B gives the location information of User B to the MO Service Provider. MO generates Billing and Accounting Info.
7. MO gives the location of User B to the 3rd party application.

(8), (9) and (10) 3rd party application gives the location information of User B to User A through the MO via its Network A.

5.3.6 Alternative Flow

None.

5.3.7 Operational and Quality of Experience Requirements

- The Mobile Operator is able to set different levels of authorisation for accessing different levels of their resource;
- The user shall have full control over his personal data;
- The Mobile Operator shall adhere to local government legislation.

User A experience MUST NOT be degraded by the policy enforcement mechanisms or charging mechanisms.

5.4 Location Application in Visited Network

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X	X	
Additional Keywords					

5.4.1 Short Description

A user is roaming in a visited operator.

Visited operator wants to push some tourist info to the roamer.

This tourist info is location dependant.

Visited Network asks permission to home operator, to know if user privacy policies (or any other home policies) allow for doing so (locating the user, and push of location dependant tourist info).

5.4.2 Actors

- End user;
- Visited Mobile Operator (Visited MO or VMO);
- Home Mobile Operator (Home MO or HMO);
- 3rd Party Location application, offered through visited.

5.4.2.1 Actor Specific Issues

- The Home Operator is the Service Provider for end-user (when user is at home network);
- Visited Operator acts as Service Provider for the end-user, when roaming in the visited network;
- Home operator enforces and manages end user privacy policies;
- The Third Party Service Provider is the owner of the Tourist Info Application:
    - Uses visited MO resources to get user location and send tourist info;
    - Providing necessary credentials to access the resource.

5.4.2.2 Actor Specific Benefits

- Home Mobile Operator:
    - Grants its customers that they are going to be protected, even when roaming.
- Visited Mobile Operator:
    - VMO can offer services to roamers while respecting their policies (privacy policies of the roamer as well as home operator policies, if required).
- Third Party Service Provider:
    - Can offer its application also to roamers whose policies allow for it.
- End Users:
    - End user sees his privacy policies enforced, even when roaming.

5.4.3 Pre-conditions

- End User is a subscriber of the Home Network;
- Both operators have agreements in order to be able to mutually ask for authorization of operations on roamers;
- End-user may either have a post-paid or prepaid subscription with the Home Mobile Operator;
- Third Party Service Provider has a contractual agreement with the Visited Mobile Operator. Third Party Service Provider Application is registered with the Visited Mobile Operator and is allowed to submit a "location_request" and “send_message”;
- The location being requested must be that of a network attached mobile end-user terminal;
- Privacy preferences of the end-user must be maintained by the Home Mobile Operator;
- Legal regulatory policies at the roaming country must be enforced;

- End User has told the home operator (i.e., via call centre or a web interface) to set and enforce policies regarding his privacy. These policies include also preferences about push of location dependant tourist info when roaming, allowing it;
- Home Mobile Operator manages policies for its subscribers;
- End User has successfully attached to the visited network.

5.4.4 Post-conditions

- End-user is presented with the location dependant tourist info;
- Privacy policies of end user have been properly applied.

5.4.5 Normal Flow

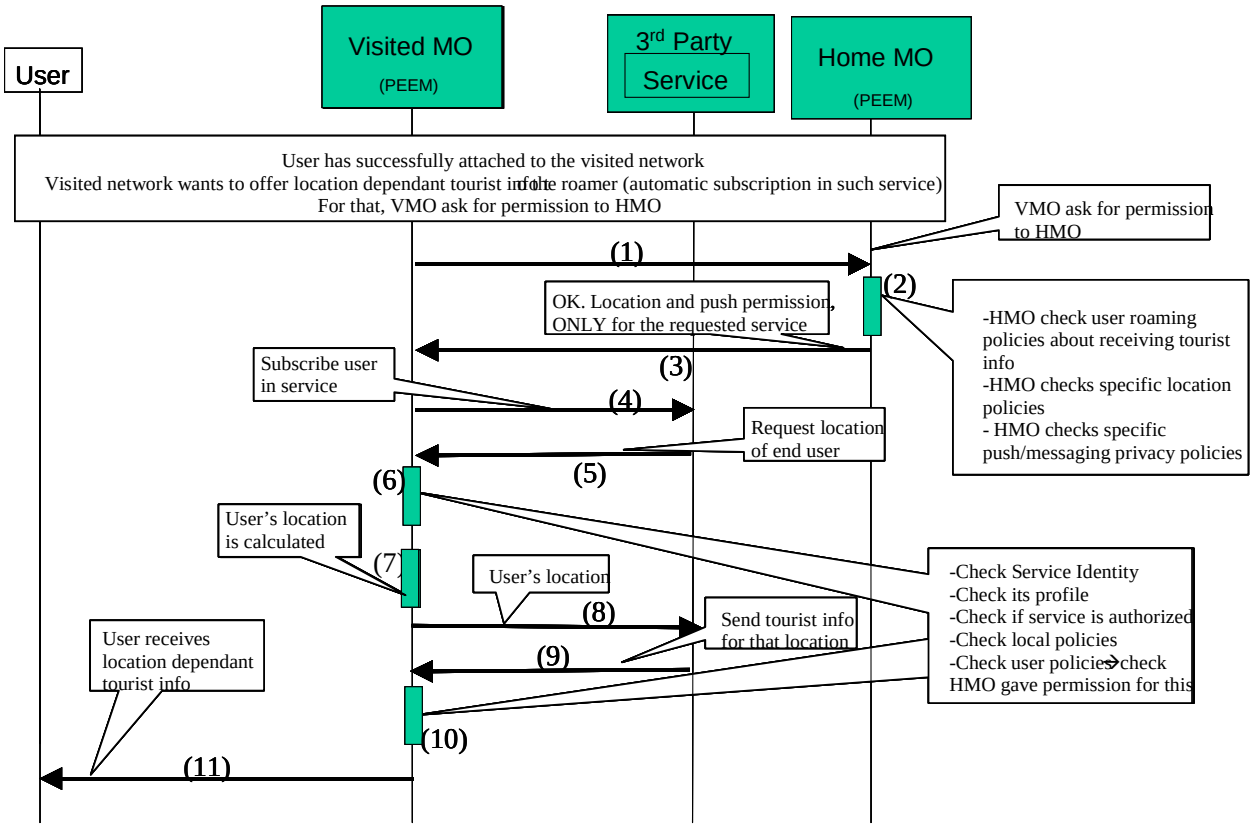

Figure 7: Normal Flow for Visited Network Scenario

As explained in the picture, prior to all the steps, we assume a former flow where:

- the user has successfully attached to the network;
- Visited MO has detected the presence of this new roamer;
- Visited MO wants to offer location dependant tourist info to roamers, so, initiates the process to ask home MO for authorization.

(1) Visited MO requests to HMO for authorization for:

- Automatically subscribe the user for location dependant tourist info push services;

- Describing that the service implies location and push of messages.

(2) Home MO makes the right checks:

- Agreements with the other operator;
- User Policies for services in roaming;
- Location Privacy policies that may override the general roaming policies set by the user;
- Push privacy policies that may override the general roaming policies set by the user;
- Regulator policies that could override any of the previous policies.

(3) HMO grants permission for the requested operation. The service for local dependant tourist info push that exists in the visited network (and only that service) may locate the user and send tourist info to him.

(4) Visited MO subscribes the roamer into the tourist info push application (a kind of automatic subscription into such service, due to the fact that policies allowed for that in step (3)).

(5) Tourist info service at the 3rd party requests location of the roamer.

(6) VMO performs the usual checks: Service Identity, Check if service is authorized to use location enablers, check relevant policies, and check if roamer allows for being located.

(7) Visited MO obtains the position of the roamer.

(8) Roamer's position is given to the 3rd party application.

(9) After calculating the tourist info relevant to the area where the roamer is, 3rd party pushes it to the user, using the VMO messaging capabilities.

(10) VMO performs the same checks as in 6.

(11) Location dependant tourist info is pushed to the user via visited MO's network.

5.4.6 Alternative Flow

- Answer in step (3) could be that explicit permission is requested from the user;
- In such case, before step (4), a message should be sent to the user, asking if he wants to receive local dependant tourist info, explaining the conditions.

5.4.7 Operational and Quality of Experience Requirements

- Both Mobile Operators are able to set different levels of authorisation for accessing different levels of their resource;
- 'The user shall have full control over his personal data;
- The Visited Mobile Operator shall adhere to local government legislation.

5.5 SMS Spam Prevention Policy

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X	X	
Additional Keywords					

5.5.1 Short Description

The Mobile Operator's Short Message system interacts with PEEM to offer a policy-enabled short message service. Subscriber data is used to customize generic privacy rules that are created by the Mobile Operator.

At this stage all the necessary 'facts' specific to the customer (with respect to the policy-enabled SMS service), are known to the Mobile Operator. So when a 'request' is made of the SMS system to send a short message to the subscriber, the request is parsed to extract 'facts' and related 'context information' that allows the system to classify the request type, (e.g., an urgent SMS etc).

5.5.2 Actors

- Mobile subscriber who is the subject of the privacy policies;
- Mobile originator of SMS messages;
- Mobile Operator.

5.5.2.1 Actor Specific Issues

Mobile subscriber:

- Wants privacy from unsolicited SMS messages.

Mobile Operator:

- The Mobile Operator wants to offer a feature rich service that can be flexibly applied to address subscriber privacy concerns.

5.5.2.2 Actor Specific Benefits

Mobile subscriber:

- Is in control of the short messages it receives.

Mobile Operator:

- Can execute policies based on user privacy;
- Can protect its subscribers from unsolicited SMS messages;
- Can implement a flexible service, with enhanced revenue generating potential.

5.5.3 Pre-conditions

- The subscriber has a mobile account with operator and is able to provide information that is used to customize the application of privacy policies/rules to his needs;
- The Mobile Operator provides a means to enter relevant subscriber information. This information along with other data is used to identify, process and enforce relevant privacy policy rules;
- The Mobile Operator has put in place some policies, including privacy and anti-spam, with respect to SMS;
- The subscriber has specified data that is used to customize privacy & anti-spam policies to the subscriber's needs:
    - An example of privacy policies for the subscriber is: "In non-working hours, I only want to receive SMS's from people in my address book".
- The SMS originates from a mobile phone of another subscriber referred to as the originator. Note that the originator could also originate from or be a 3rd party service provider.

5.5.4 Post-conditions

SMS messages may be delivered, withheld, re-routed, rejected, etc., based on Mobile Operator policy rules and subscriber specific data.

5.5.5 Normal Flow

- Originator of SMS sends a message to the Mobile Operator's Short Message Service system requesting that the message is sent to the subscriber who is the subject of the privacy policies, mentioned above;
- Mobile Operator parses the SMS and extracts all facts and context information and checks:
  - if originator is authorized to use the SMS system;
  - if originator is authorized for the requested operation;
  - that policies set by the subscriber allow for receiving messages from the originator.
- Originator request is accepted;
- Message is sent to the subscriber.

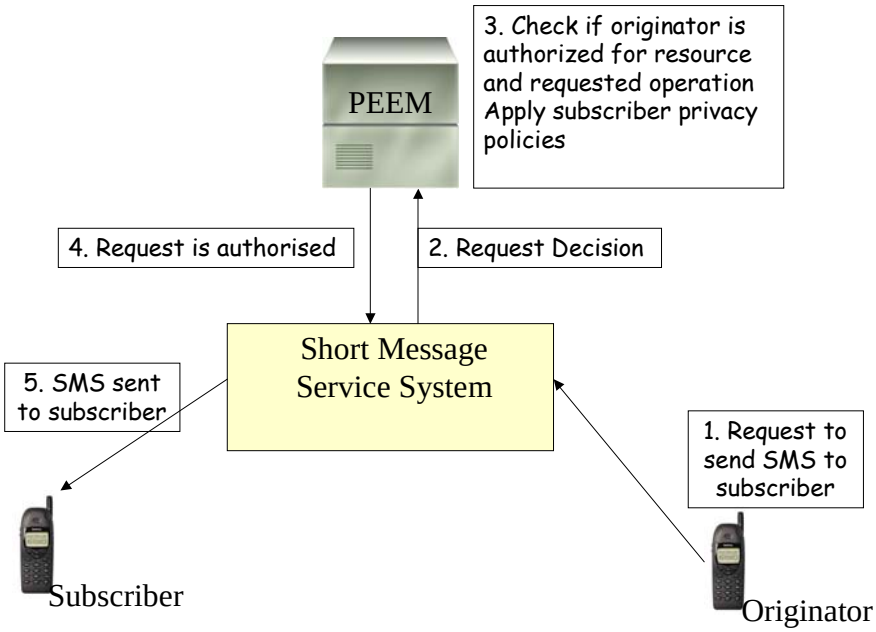

Figure 8: SMS Spam Prevention

5.5.6 Alternative Flows

None identified.

5.5.7 Operational and Quality of Experience Requirements

- A policy-enabled service incorporates or has access to PEEM capabilities;
- The PEEM enforces decisions on behalf of the policy-enabled service;
- The mobile subscriber is able to provision his privacy preferences via an appropriate interface, e.g. through his mobile device or via a web based/GUI;
- The mobile subscriber's contract is always with the Mobile Operator;
- User experience must be uniform, seamless and consistent whenever the user accesses the system;
- User experience using services must not be deteriorated by the use of policy enforcement mechanisms;
- It should be possible always for PEEM mechanisms to comply with local regulatory policies;
- PEEM mechanisms must support "telco reliability and performance", regarding downtime, transactions per second, etc.

5.6 Charging Control using Policies

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X	X	
Additional Keywords					

5.6.1 Short Description

A 3rd party Application Service Provider (ASP) delivers services to end-users. In doing so, the ASP uses resources from a Mobile Operator and the MO will perform settlement for that with the ASP. Mobile Operator charges users on behalf of the 3rd party. Policies are being used to protect the third party Application Service Provider (ASP) from being exposed to charges generated by the Mobile Operator's resources consumption (a session to a pre-paid subscriber of the operator in order to deliver a service) when the pre-paid debit limit of the user has been reached and the user has no money to pay for the service delivered by the ASP. This use case provides two alternatives to protect the ASP against such charges:

1. PEEM acts *after* subscriber has overrun his/her prepaid balance. The third party creates a policy rule to handle subscriber overruns of his/her prepaid balance. The policy rule created for purposes of illustrating this use case is called the "PP_CHECK" policy rule. This use case is an example of how an event can be triggered during an ongoing session. In this case, the event is triggered when the network detects that a subscriber has exceeded his pre-paid limit and run out of funds. Additionally, the use case illustrates also how a 3rd party Service Provider defines his own particular policies. This alternative is described as the 'normal flow' in this use case.

2. PEEM acts *before* subscriber has overrun his/her prepaid balance – it will prevent overrunning of the account balance. The Mobile Operator creates a policy to handle subscriber overruns of his/her prepaid balance: such policy won't allow the resource consumption in case the pre-paid debit limit of the subscriber has been reached. The policy created for purposes of illustrating this use case is called the "ACCOUNT_CHECK" policy. This alternative flow shows how the ASP request for using a resource triggers a Charging Control policy check in addition to a business agreement check. This alternative is described as 'alternative flow' in this use case.

5.6.2 Actors

- Third party ASP;
- Pre-paid subscriber;
- Mobile Operator.

5.6.2.1 Actor Specific Issues

Third party ASP:

- Wants to avoid delivering a service to a user, when user's pre-paid account is empty (avoiding, thus, the expense of using the Mobile Operator's resources when no revenue will be gathered in the end);
- Wants to have a Service Level Agreement (SLA) with the Mobile Operator. This SLA describes to what extent the ASP is entitled to make use of the resources of the Mobile Operator;
- Wants to be able to define his own policies (applies to the normal flow only, this is not applicable to the alternative flow).

Mobile Operator:

- Wants to manage applications across a diverse and distributed set of service providers;
- Wants a flexible service management mechanism, e.g., policy management to manage access to and protect the integrity of network services, where a Service Level Agreement is used to specify policies;
- Wants to define network service policies.

Pre-paid subscriber:

- Wants to be charged according to terms described in his service contract.

5.6.2.2 Actor Specific Benefits

Third party ASP:

- Is protected from unwarranted charges generated when the subscriber's prepaid debit limit is reached.

Pre-paid subscriber:

- Uses services according to the terms of his pre-paid subscription.

Mobile Operator:

- Can offer a feature rich service.

5.6.3 Pre-conditions

- The subscriber has a pre-paid account and his subscription allows him to receive services from the ASPs through the Mobile Operator, at a certain price;
- The ASP has a Service Level Agreement (SLA) with the Mobile Operator that, (1) allows the Mobile Operator to charge the ASP for resource usage, (2) allows the Mobile Operator to charge the subscriber on behalf of the ASP, and (3) obliges the Mobile Operator to provide for resource usage;
- The Mobile Operator has implemented a policy-enabled session initiation that incorporates or has access to PEEM functionality. A specialized case of this pre-condition applies to the alternative flow: The Mobile Operator has implemented a policy-enabled service (e.g. session control) that incorporates or has access to PEEM functionality so that the SLA and the ACCOUNT_CHECK policy are enforced;
- This pre-condition applies to the normal flow only, not to the alternative flow. The 3rd party has defined some policies in place to be triggered when the user's account becomes empty. (The policies may be specified by the 3rd party through an appropriate interface, or the Mobile Operator could do it on behalf of a mutual agreement).

5.6.4 Post-conditions

5.6.4.1 Normal flow – acting on account that was overrun

The network detects that the prepaid account of this called party has overrun its lower bound. This results in a notification to the policy-enabled service that uses PEEM capabilities to process the notification. The PEEM recognizes this as an 'alert'

and takes appropriate action to process the alert. The PEEM identifies the relevant policy, (PP_CHECK), evaluates it and enforces the resulting decision. As a result the call leg to the called party is released.

5.6.4.2 Alternative Flow – preventing the account from overrunning

The policy-enabled network service used the PEEM capability that was invoked on the ASP request for network service. Appropriate information was sent about the requested network service to be assessed in an evaluation process. The PEEM enabler was consulted for a SLA check. Also the PEEM enabler was consulted for the evaluation of the ACCOUNT_CHECK policy and for the enforcement of the ACCOUNT_CHECK decision. As a result the session was initiated and thereafter re-evaluated every two minutes.

5.6.5 Normal Flow

- The Mobile Operator creates the pre-paid balance policy rule, (PP_CHECK) for its call management service. PP_CHECK is associated with a condition ("pre-paid_account.balance <= prepaid_account.lower_bound) and an action ("release.subscriber_call_leg");
- The rule is created for PEEM. As part of the creation process the PEEM is configured to respond appropriately to events associated with the rule;
- As part of the normal operation of the call management service, a call leg is created and routed to a certain called party in the network. After some time elapses, the network detects that the prepaid account of this called party has overrun its lower bound. This event triggers the invocation of the call management service PEEM;
- The PEEM recognizes this as an alert that is to be processed. The PEEM extracts all 'facts' and 'context' information for further processing;
- The PEEM applies the policy rule PP_CHECK whose condition is satisfied by the information passed onto it;
- The PEEM enforces the resulting action. As a result the call leg to the called party is released, hence protecting the third party ASP from being exposed to unwarranted charges generated by this call leg.

5.6.6 Alternative Flows

The Mobile Operator creates the pre-paid balance policy rule (ACCOUNT_CHECK) for its network service. The ACCOUNT_CHECK policy looks like this: If the pre-paid account balance is lower than or equal to the lower bound then the session is discontinued, otherwise it is allowed to be initiated and continue for two minutes.

The PEEM enabler is invoked on a request to initiate a session to the subscriber. The SLA is evaluated. The outcome of the SLA evaluation is enforced (the decision is enforced, e.g. continue). The ACCOUNT_CHECK policy is evaluated. The outcome of the ACCOUNT_CHECK policy evaluation is enforced (the decision is enforced: initiate the session and set the time limit to 2 minutes). The session is now initiated and continued for two minutes. After two minutes have passed a notification is sent which invokes the PEEM enabler. The ACCOUNT_CHECK policy is evaluated again. If the account did not reach the lower bound the session will be continued for two more minutes. This will go on until the account lower bound limit is reached (then the session will be discontinued) or until the ASP or subscriber decides to end the call.

5.6.7 Operational and Quality of Experience Requirements

- Policies may be defined in high-level service terms consistent with a policy information model;
- The high-level representation of a policy is mapped onto an internal representation that is best suited for computations and evaluation;
- The application of a policy may require a decision from external elements;
- User experience must be uniform, seamless and consistent whenever the user accesses the system;
- User experience of using services must not be degraded by the use of policy enforcement mechanisms;
- Policies defined by 3rd parties SHOULD be based on standardised schema and semantics;

5.7 Enforcing Policies

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X		
Additional Keywords					

5.7.1 Short Description

A set of policies has been set up in advance by the owner of a resource. They must be satisfied by a requestor before it can access or use the resource.

In this use case we assume that the owner of the resource also enforces the policies. Other sections will consider the use cases where these are different parties or in different domains.

5.7.2 Actors

- Owner of the resource:
  - It has set up policies on the resource that it controls;
  - Enforces the policies on requests to the resource.
- Requestor:
  - Any issuer of request to access and use the resource:
    - Provide necessary credentials to use the resource as it has been informed of.

5.7.2.1 Actor Specific Issues

- Owner of the resource:
  - Enforcing the policies;
  - Providing access to its resource.
- Requestor:
  - Providing necessary credentials to access the resource;
  - Using the resource.

5.7.2.2 Actor Specific Benefits

- Owner of the resource:
  - Can offer access to resource and its use while enforcing conditions of usage expressed in policies;
  - Knows that resource is appropriately protected.
- Requestor:
  - Can access resources to use within its applications;
  - Can simplify and automate the way to satisfy the conditions to use a resource while requesting the resource: need only to know what credential to pass and how.

5.7.3 Pre-conditions

- Owner of the resource:
  - It has set up policies on the resource that it controls;

    - It has communicated what and how credentials must be passed in a request to the resource to potential requestors:
        - E.g. via SLA or a priori agreements / communications.
- Requestor knows resource;
- Requestor knows the conditions it must satisfy (e.g. via Service Level Agreement (SLA)):
    - E.g. what credentials and how they must be passed with a request.

5.7.4 Post-conditions

- The request from requestor reaches the resource and is executed by or on the resource;
- The response may be treated through additional policy steps if imposed by:
    - Policies of the target resource:
        - E.g. a charging event is logged after successful or failed access of the resource.
    - Or as a repeat of the present use case where the responder becomes the requestor and vice-versa.

5.7.5 Normal Flow

- Requestor prepares request to resource and provides information / meta-data / credentials to be able to satisfy the conditions that he/she knows for using the resource;
- Request is logically processed by the PEEM enabler (logical entity / mechanism):
    - Request and / or credentials[4] are passed to other resources for action and / or validation of the results as specified by the policies (*):
        - E.g. The requestor is first authenticated based on credentials then it is checked for authorization (which may be based on authorization statement/token supplied with the request) to access the resource and then it is passed to a charging systems that generates a billing event;
        - These may be checked to be up-to-date policies. They may or may not depend on the nature of the request and on the requestor.
- Request is passed to resource for action (assuming successful validation of all the steps);
- The action is executed on or by the resource (see post conditions);
- Response is returned to the requestor (see post conditions).

5.7.6 Alternative Flow

- At step (*) above, it is possible that some of the validation fail. In such a case, the following cases may take place:
    - The request to the resource fails and an error message is returned to the requestor;
    - A dialog may be established between the requestor and one of the involved intermediate resource:
        - e.g. please provide a new credential or answer the following challenge.
- Other alternative steps are discussed in the use case sections below.

[4] The credentials may result from previous steps performed by the requestor to acquire these credentials as allowed or specified by the details on the executions policies that it is aware of.

5.7.7 Operational and Quality of Experience Requirements

- The PEEM enabler of the owner of the resource is aware of the policies associated to the resource;
- The resource and network is logically setup such that any request to the resource is processed by the PEEM enabler:
  - Note that this can be done in numerous manners that may not impose a single PEEM entity.

5.7.8 Concrete Examples

Concrete examples include a location-based service exposed by a service provider provided that appropriate authentication, authorization, charging and logging is taking place.

5.8 Delegation

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X		
Additional Keywords					

5.8.1 Short Description

The owner of a resource deploys it by delegating some of the functions (e.g. authentication, charging, logging, …) to other resources. Alternatively, the implementer of a resource implements it by delegating some of the functions to other resources. The delegation is expressed as policies enforced in any messages to and from the resource.

The flows associated to these steps remain the same and do not need to be expanded in this section. The present section focuses solely on the steps that perform delegation of functions, acknowledging that these may include enforcement and validation steps and workflow as discussed above.

5.8.2 Actors

- Owner (or implementer) of the resource:
  - It deploys or implements the resource by delegating some functions to other resources and expresses these via policies;
  - Enforces the policies on requests to the resource.
- Requestor:
  - Any issuer of request to access and use the resource;
    - Provide necessary credentials to use the resource as it has been informed of.

5.8.2.1 Actor Specific Issues

- Owner (or implementer) of the resource:
  - Resource deployment or implementation by delegation;
  - Same as in use cases above.
- Requestor:
  - Using the resource;
  - Same as in use cases above.

5.8.2.2 Actor Specific Benefits

- Owner (or implementer) of the resource:
    - Can simplify implementation or deployment by relying on other resources to provide the delegated functions;
    - Re-use resources;
    - Avoid silos;
    - Simplifies integration:
        - Re-use resources through PEEM and policies.
    - Same as for use cases above.
- Requestor:
    - Can access resources to use within its applications;
    - Same as for use case above.

5.8.3 Pre-conditions

- Owner (or implementer) of the resource:
    - It has implemented or deployed resources by relying on a set of policies for the resource that it controls:
        - Same as for use cases above.
- Same as for use cases above for the requestor.

5.8.4 Post-conditions

- Same as for use cases above.

5.8.5 Normal Flow

- Requestor prepares request to resource and provides information / meta-data / credentials to be able to satisfy the conditions that he/she knows for using the resource;
- Request is logically processed by the PEEM enabler (logical entity / mechanism):
    - Request and / or credentials are passed to other resources that perform in particular the delegated functions for action and / or validation of the results as specified by the policies (*):
- Request is passed to resource for action (assuming successful validation of all the steps);
- The action is executed on or by the resource (see post conditions);
- Response is returned to the requestor (see post conditions).

5.8.6 Alternative Flow

- At step (*) above, it is possible that some of the validation fail. In such a case, the following cases may take place:
    - The request to the resource fails and an error message is returned to the requestor;
    - A dialog may be established between the requestor and one of the involved intermediate resource:
        - e.g. please provide a new credential or answer the following challenge.
- Other alternative steps are discussed in the use case sections below.

5.8.7 Operational and Quality of Experience Requirements

- Delegation may be implemented:
    - Directly by the target resource:
        - PEEM functionality built in the resource.
    - By another logical mechanism:
        - E.g. as a component in front of the resource that intercepts any request to it.
    - Way to provide delegation for legacy system:
        - New conditions are enforced in front of it
        - Conditions already enforced and not expressed in policies.
- Same as for use cases above.

5.9 Enabler composition

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X	X	X
Additional Keywords					

5.9.1 Short Description

This use case describes the PEEM performing enabler composition, which means that PEEM enabler is making use of a number of other enablers within a service transaction. A user accesses a mobile location based game service, where the network operator provides enablers to an application provided by a third party service provider. The user perceives the delivery of the game service as a single transaction. If the transaction proceeds normally, the user receives the service and is charged. If the transaction proceeds abnormally, e.g. if one of the enablers fails perhaps, the user is notified and, since the service was not delivered, the user is not charged.

5.9.2 Actors

- User – end user using a mobile device;
- Network operator – operates a mobile network and provides network enablers to third parties;
- Third party service provider-– provides a mobile location based game.

5.9.2.1 Actor specific Issues

- User:
    - Uses any third party service whilst relying on existing agreements and solutions.
- Network operator:
    - Offers composed services;
    - Enforces SLA in-between user, third party service provider and itself.
- Third party service provider:
    - Provide a high value service sharing the risk and investment.

5.9.2.2 Actor specific Benefits

- User:
    - Perceives and pays only for the successful use of a third party service.
- Network operator:
    - Provides basic values to any third party.
- Third party service provider:
    - Leverage and uses existing NO resources to provide added service value.

5.9.3 Pre-conditions

- User has an account with the network operator;
- Third party service provider has made the necessary business and technical arrangements with the network operator to use several enablers as part of a specific service offering;
- User has an account with the third party service provider. The user has subscribed to the service of Third party service provider. The user has given consent for all charging actions with respect to the service that he/she subscribed to;
- Third party service provider is authorised to receive position information for this user.

5.9.4 Post-conditions

The user has sent a message to the network operator, indicating some action as part of the game play:

- The network operator has provided the users position and the users message to the application service provider;
- The Third party service provider has computed a result and sent a message via the network operator's message enabler to the user;
- The user is charged 1 million woolongs for a successful transaction by the network operator;
- The network operator has settled 500 thousand woolongs with the application service provider (settlement).

5.9.5 Normal Flow

1. The user sends a message "SHOOT WITH PHOTON LASER" to the service account for the game service.
2. The network operator forwards this message to the application service provider.
3. The Third party service provider requests the network operator to reserve 1 million woolongs on the user account.
4. The network operator places a reservation of 1 million woolongs on the user account.
5. The Third party service provider requests the position of the user. Note that the network operator could have sent this position information in conjunction with the user message, depending on the technical solution. Also note that in such case the network operator could have placed a reservation directly, depending on the technical solution.
6. The network operator determines the position of the user.
7. The network operator provides the Third party service provider with the position. The note in the previous step still applies.
8. The Third party service provider computes a result.
9. The Third party service provider sends a result message to the network operator.
10. The network operator forwards the result as a message to the user "DIRECT HIT – 300 POINTS".

11. The Third party service provider requests that the network operator charges the user 1 million woolongs. Note that this information may be sent in conjunction with the previous step depending on the technical solution. Note also that depending on the technical and business arrangements between the network operator and the Third party service provider the network operator may instigate this step directly.

12. The network operator charges the user. Note that depending on the technical and business arrangements between the network operator and the Third party service provider the network operator may instigate this step directly (step 11 would then not be required).

In some later offline stage the network operator will settle the 500,00 woolongs with the Third party service provider.

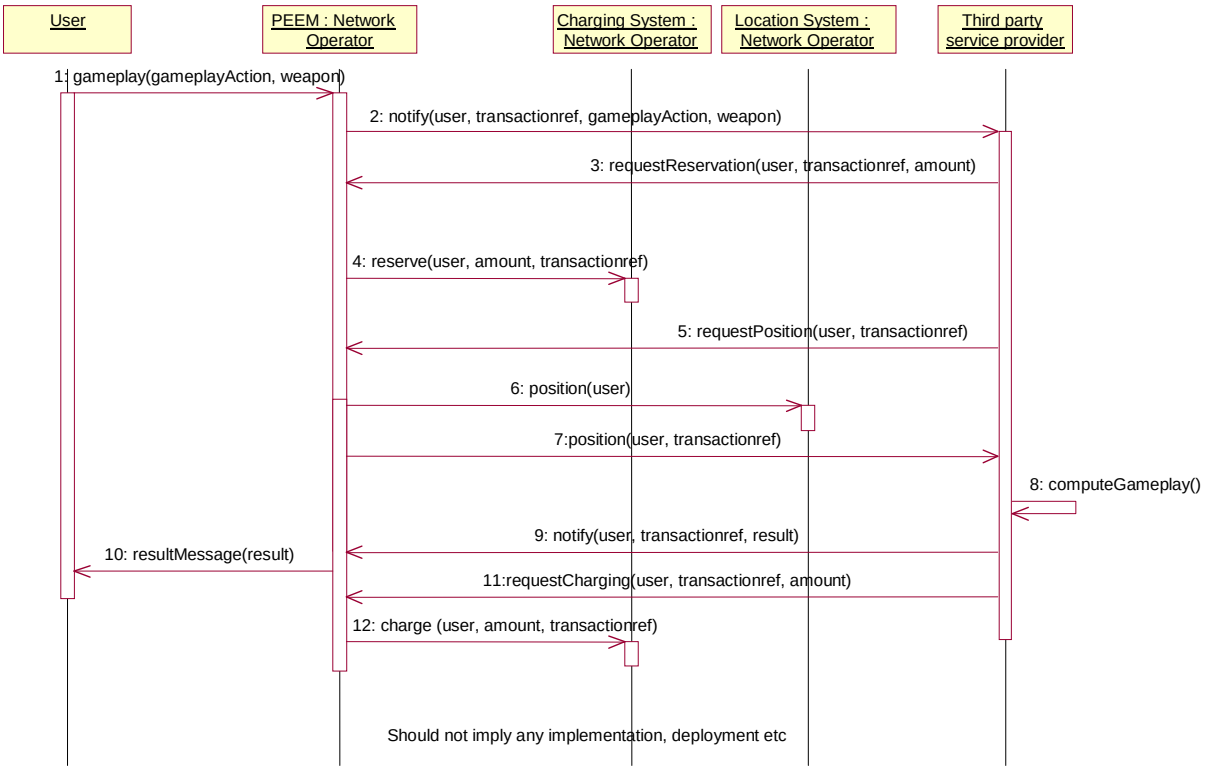

Figure 9: Normal Flow for Composition Use Case

5.9.6 Alternative Flow

This alternative flow is an exception handling flow, and is the substance of this use case, since it leads to specific overall requirements. The post-conditions described above do not apply to these exception flows.

5.9.6.1 Alternative flow 1: Enabler exception

1. As per normal flow.

2. As per normal flow.

3. As per normal flow.
4. As per normal flow.
5. As per normal flow.
6. The network operator cannot retrieve the position of the user due to some technical fault.
7. The network operator advises the Third party service provider to this effect.
8. The Third party service provider requests to cancel the reservation on the user account. Note that depending on the technical and business arrangement between the two actors, this and the next step may be affected by the network operator directly. Also these steps may be performed after message 10 or 11.
9. The network operator cancels the reservation on the user account.
10. The Third party service provider sends a message to the network operator to be forwarded to the user. Note that depending on the technical and business arrangement between the two actors, this and the next step may be affected by the network operator directly.
11. The network operator sends the message “TECHNICAL PROBLEMS – YOU HAVE NOT BEEN CHARGED” to the user.

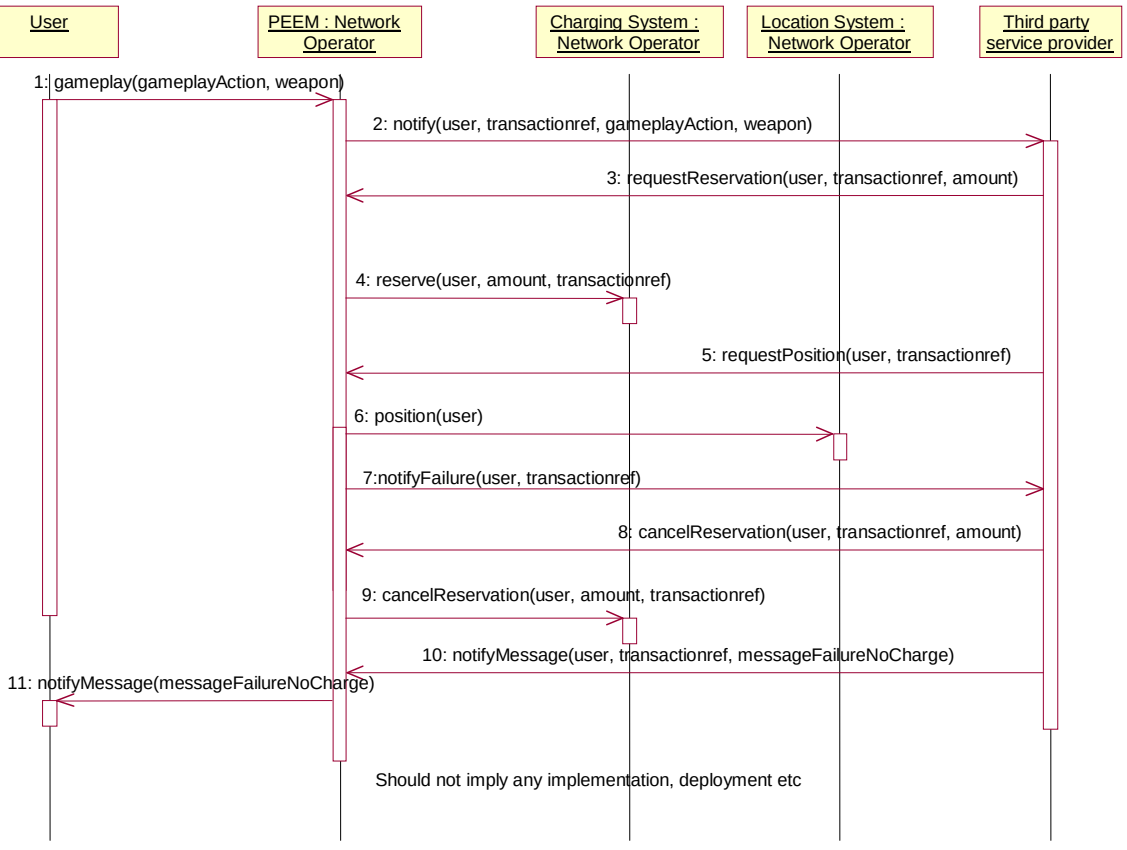

Figure 10: Alternative Flow for Composition Use Case

5.9.7 Operational and Quality of Experience Requirements

The user, in this case also the customer, perceives and values the service, not the underlying enablers. If he receives the service he pays the agreed price. If he does not receive the service, he does not pay for the service, or for the use of the underlying enablers.

5.10 Interaction with Register and Discover

	Affected Areas				
	Device	**Connectivity**	**Enabling Services**	**Applications**	**Content**
Tickmarks (X)			X		
Additional Keywords					

5.10.1 Short Description

In this scenario, an enabler implementation (let's call it EN) is installed into a service provider domain, registers itself, and is then discovered by an application (let's call it APP). This use case describes these processes and how PEEM interacts with them.

5.10.2 Actors

Service Provider having the enabler implementation.

Actor deploying an application attempting to invoke an enabler implementation at an SP.

5.10.2.1 Actor Specific Issues

The SP wants to automate as much as possible the deployment of enabler implementations and their use by applications.

The actor deploying an application wants to dynamically determine how to invoke enabler implementations deployed at different SPs.

5.10.2.2 Actor Specific Benefits

SPs reduce effort required to deploy enabler implementations and have applications access those implementations.

Actor deploying application reduces effort required to invoke enabler implementations at different SPs.

5.10.3 Pre-conditions

SP has enabler implementation that conforms to (OMA's to-be-defined) Discover and Register enabler specifications.

SP has defined the policies for accessing these Discover and Register enabler implementations.

Actor deploying application has adequate relationship with SP to permit the application to access the enabler.

5.10.4 Post-conditions

Application has received description of interfaces (i.e., set of parameters, protocols) needed to access the SP's enabler implementation.

5.10.5 Normal Flow

1. The SP installs the enabler implementation EN.
2. The SP defines to its PEEM implementation what policies should be evaluated/enforced when requests are received destined for EN. The policies might include, but not be limited to, authenticating the requestor using a certificate, charging 3 Baht's for requests to EN, or requests to EN can only be made between 1500 GMT and 1700 GMT.
3. The SP configures the enabler implementation (EN) to know how to access the Register enabler implementation that is already deployed in the SP's domain.
4. The EN uses the OMA-defined protocols to describe its interface (including required parameters) to the Register enabler implementation. EN's interface is the I0 interface as defined by OMA.
5. The application (APP) sends a Discover request to SP's well-known address to find out the interface information for EN
6. The SP's PEEM implementation intercepts the incoming Discover message from APP and applies the SP-defined policies pertaining to the Discover enabler implementation.
7. The Discover enabler implementation analyzes the incoming messages, and returns the interface specification for EN that was supplied by EN in step 4 above.
8. As the Discover response passes out of the SP domain through the PEEM implementation, it is recognized by PEEM as a Discover message. PEEM analyzes the policies that have been specified by the SP for EN and combines the required parameters to satisfy those policies (I1) with the enabler-required parameters (I0) to produce the full interface (I0+I1) that APP will have to use to access EN. PEEM changes the Discover response to include this enhanced interface specification.
9. APP receives the Discover response and uses the information to implement the required interface to invoke EN in the chosen SP's domain.

5.10.6 Alternative Flow 1

An alternative approach is that in step 4 PEEM intercepts the Register message and performs the same policy analysis as described in step 8 above. PEEM then sends the enhanced Register message to the Register enabler implementation. In this case, PEEM does not have to (but may) process the Discover flow in step 8.

5.10.7 Alternative Flow 2 – SP publishes enabler at discovery server

The following scenario describes an alternative for steps 1 to 4. It is the SP that performs the publication of a “new” enabler in the SP’s domain.

1. The SP installs or updates the enabler implementation EN.
2. The SP defines to its PEEM implementation what policies should be evaluated/enforced when requests are received destined for EN. The policies might include, but not be limited to, authenticating the requestor using a certificate, charging 3 Baht's for requests to EN, or requests to EN can only be made between 1500 GMT and 1700 GMT.
3. The SP publishes the enabler implementation including protocol/interface description at the discovery enabler that will be invoked by the APP. This can be an automated part of the installation or updating procedure.

5.10.8 Alternative flow 3 – SP provides service interface description offline

The following scenario describes a scenario involving registration and discovery as typically seen from a 3GPP/Parlay point of view. The scenario is about an application (APP) finding out on how to access an enabler, and accessing an enabler (EN). In this scenario the discovery enabler is called "framework". Both the framework and the service enabler are in the SP domain. Note that the sequence below is an abstracted version of the sequence specified in 3GPP specifications.

1. The SP exchanges protocol description, interface description, authentication and authorization information for accessing the framework *and* the EN off-line with the APP. Such information is exchanged by other means than a discovery enabler (e.g. e-mail).
2. The SP registers the enabler implementation at the framework.
3. The SP creates a Service Level Agreement (SLA) for an APP that wants to access the EN and stores this SLA at the framework.
4. APP authenticates to the framework and signs the SLA for using the EN.
5. The framework returns to the APP the address of where to invoke the EN.

The APP uses the EN. The SLA is enforced during usage (e.g. by a PEEM enabler implementation).

5.10.9 Alternative flow 4 - Enabler registers policies to apply during discovery, a user can define user-specific policies

As a precondition for this alternative flow the service consumer needs to know what information it needs to put in the discovery request to satisfy the SP policies for accessing the (EN) enabler. Such discovery request is sent to find out where to find a specific enabler (EN) implementation.

1. The SP installs the enabler implementation EN.
2. The SP defines to its PEEM implementation what policies should be evaluated/enforced when requests are received destined for EN. The policies might include, but not be limited to, authenticating the requestor using a certificate, charging 3 Baht's for requests to EN, requests to EN can only be made between 1500 GMT and 1700 GMT, the APP needs to provide input data for authorization during the discovery of EN, the APP needs to provide input data for user authentication during the discovery of EN.
3. Users subscribed to use the enabler may define additional personalized policies when applications access the enabler (e.g. don't give location information to foo.com, requests to EN can only be made between 1500 GMT and 1700 GMT).
4. The enabler implementation (EN) knows how to access the Register enabler implementation that is already deployed in the SP's domain, e.g. by configuration or as part of a user interaction to set EN as a provider of a user's information.
5. The EN uses the OMA-defined protocols to register in the Register enabler implementation; a standard EN only needs to indicate its service type (e.g. a urn like "urn:oma:LOC_MLP:2004-08" which might correspond to a standard definition publicly available for application developers), not its detailed interface description. EN's interface is the I0 interface as defined by OMA. The EN may then indicate the policies that it wants the Register implementation to apply when answering Discovery requests (e.g. Authenticate Requester, Authorize Requester) – The results of the checking of these policies are to be processed by the enabler later on.
6. The application (APP) sends a Discover request to the user's discovery enabler to find out where to find a specific enabler implementation for the user who is accessing APP.
7. An SP's PEEM enabler implementation evaluates the incoming Discover message from APP and applies the SP-defined policies pertaining to the Discover enabler implementation (for accessing the Discovery enabler). Also additional policies may be evaluated to check whether the APP can access the EN as well as other policies that the EN may have indicated during the registration of EN.
8. If APP is allowed to access the SP's EN the Discover enabler implementation returns the address for sending requests to EN that was supplied by EN in step 2 above, *accompanied with* all the information that satisfies the policy defined by the EN, e.g. input data for authorization that states that the APP is allowed to access the EN.

9. APP receives the Discover response and uses the information to invoke EN in the chosen SP's domain. The EN invocation contains the information that was returned by the Discovery response after evaluation of the policies defined by the EN.
10. An SP's PEEM enabler implementation or the EN interacting with the SP's PEEM enabler implementation evaluates the incoming EN message from APP, checks the validity of the input data received in the message and it applies user and/or additional SP-defined policies pertaining to the EN enabler implementation (for accessing the EN enabler).
11. EN receives the service request from APP.

6. Requirements (Normative)

6.1 High-Level Functional Requirements

This section contains high-level requirements for the PEEM enabler.

Some of the requirements in the following sub-sections originate from other OMA source material, (see section 2.1) and may have been modified for the PEEM enabler. In some cases, the source requirement from this material is indicated.

The PEEM enabler specification:

1. MUST NOT restrict the technology and deployment options, (e.g. a geographically distributed PEEM, a highly scalable PEEM) that satisfy the requirements. (Motivated by Use Case 5.7).
2. MUST NOT violate any of the requirements in section 6.1 of [ARCH].
3. MUST support the evaluation (or evaluation and execution) of policies for at least the following:
    - Requests from a 3rd party ASP, wanting to utilize Operator's (Service Provider's) resources;
    - Requests from end users, (including end users belonging to different networks) wanting to access services of the Operator, (Service Provider);
    - Requests from end users, wanting to access services of a 3rd party ASP, and;
    - Requests from another Operator, (Service Provider);
    - Response from the resource;
    - Request from an end user wanting to communicate with another end user.
4. MUST support the identification of policies that apply to a request and response. (Motivated by Use Case 5.5).
5. MUST support the receipt of requests for a policy evaluation (or evaluation and execution), e.g. authorisation. (Motivated by Use Case 5.5).
6. MUST support the interaction with other resources to enforce the results of policy evaluation (or evaluation and execution), e.g. security. (Motivated by Use Case 5.5).
7. MUST support policies that allows the request to pass through without any additional policy enforcement, (i.e. this allows other resources to implement their own local policies (Motivated by Use Case 5.5).
8. MAY support policies evaluated (or evaluated and executed) on events (e.g. asynchronous events). (Motivated by section 5.6).
9. MUST specify a language based on standard schema and semantics. (Motivated by almost all Use Cases).
10. MUST support the processing of data provided as input with the processed request or response (e.g. the level of security or QoS). (Motivated by almost all Use Cases).

6.1.1 Security

1. The PEEM enabler specification MUST support secure exchanges between requestor and responder.
2. The PEEM enabler specification SHOULD support policies that allow the use of different trust models [ARCH].
3. The PEEM enabler specification MUST support the evaluation of policies by delegation of the policy execution steps to enablers that may be in different security and administrative domains [ARCH].

6.1.2 Charging

1. The PEEM enabler specification MUST NOT preclude any charging models between different actors. [ARCH]

6.1.3 Administration and Configuration

1. The PEEM enabler specification MUST NOT preclude the establishment of SLAs. (Motivated by Use Cases 5.2, 5.3, 5.4 and 5.6).
2. The PEEM enabler specification MUST define interfaces for a principal to manage policies related to a resource.
3. The following functions related to policy management MUST be supported:
   - To create policies;
   - To update/modify/re-use policies at runtime, (e.g. if the application/service provider adds new functionality that may impact policies);
   - To view policies;
   - To delete policies.

   The following functions related to policy management MAY be supported:
   - To prioritise/sequence policies;
   - To identify inconsistencies.
4. The PEEM enabler specification MUST specify mechanisms that associate policies with:
   - an individual resource;
   - a group of multiple resources;
   - a specific requestor;
   - a specific request.
5. The PEEM enabler specification MUST support ways to include in a policy rule references to input data (i.e. contained in the service request/response), during policy management (with the intent to be replaced by the real input parameter values during the policy enforcement)." (Motivated by Use Case 5.5 and 5.7).
6. The PEEM enabler specification MUST be able, as part of the policy enforcement process or as part of the policy management process, to derive from policies what additional input data a requestor must supply. (Motivated by Use Case 5.5 and 5.7).
7. The PEEM enabler specification MUST enable a resource owner to delegate to other parties the enforcement of policies for such resources. (Motivated by Use Case 5.8 and 5.9).
8. The PEEM enabler specification MUST permit the delegation of policy management to parties other than the Service Provider, e.g. allow a subscriber to set his/her privacy rules.
9. The PEEM enabler specification SHOULD support policy management at run time.
10. The PEEM enabler MUST support policy management by various actors, e.g. service provider, network operator, enterprise, and end-user.

6.1.4 Usability

1. If OMA defines a mechanism to perform registration and discovery, the PEEM enabler specification MUST be compatible with those mechanisms for the owner of a resource to register and discover the data required in order for another party to use a service enabler. (Motivated by Use Cases 5.2, 5.3, 5.4 and 5.7).
2. The PEEM enabler specification MUST be able to support requests done on behalf of principals.

6.1.5 Interoperability

1. Standardized interfaces MUST be defined for the PEEM enabler [ARCH].

6.1.6 Privacy

1. The PEEM enabler specification MUST support enforcement of privacy policies (Motivated by [Privacy]).
2. The PEEM enabler specification MUST be able to support the expression of policies that protect user identity and related user information including privacy preferences, e.g. for anonymity.

6.2 Overall System Requirements

The general characteristics & behaviours specified in this chapter are supported by the PEEM enabler:

1. The PEEM enabler specification MUST provide a mechanism to enforce the policy associated to a resource on any request to that resource and on any associated response.
2. The PEEM enabler specification MUST be able to support delegation to one or more resource (e.g. charging, service discovery).
3. The PEEM enabler specification MUST support requestors and responders located in the same or different domains.
4. The PEEM enabler specification SHOULD NOT preclude the deployment of service enablers in high-availability, high-uptime, scalable environments (e.g. By requiring implementation in ways which disable the use of the functions of this environment) [ARCH].
5. The PEEM enabler specification MUST be able to simultaneously support multiple versions (i.e. multiple instances, defined according to different releases of the OMA specifications) of a target resource's interface [ARCH].
6. The PEEM enabler specification MUST be able to support a mechanism through which the PEEM enabler is made aware of the addition, modification or removal of a resource or an interface to a resource.
7. The PEEM enabler MUST be able to determine the resource that needs to be protected and the policies associated to that resource.
8. The PEEM enabler specification MUST be able to support the interruption of flows and rejection of requests, through enforcement of policies (e.g. failure in authentication or authorization requests, charging failure, etc.).
9. The policy expression language MUST be able to create rules to use at least the following information (Motivated by mostly all of use cases):
    a. Subscriptions of the end user.
    b. End-user segment, (e.g. gold, silver bronze users).
    c. Subscriptions/agreements (SLAs) with 3rd parties.
    d. End users account status.
    e. End user personal data.
    f. Service Provider variables and conditions.
    g. Regulatory/legislative (e.g. user protection) variables and conditions.
10. The PEEM enabler MUST be able to create log information about the flows and events (such as error events) that the PEEM processes, and the associated policy evaluation (or evaluation and execution), that may result (for auditing purposes). Examples of log information are:
    - Statistical information (e.g. failure rate of a particular request for a resource and real time QoS information for a session);

- Information for both inbound requests, (e.g. requests from authorised third parties, and outbound responses).

(Motivated by Use Case 5.8)

11. The PEEM enabler specification MUST be able (by interacting with other enablers) to obtain session information (e.g., user_id) from the information contained in a request/response (Motivated by Use Case 5.6).

12. The policy expression language of the PEEM enabler specification MUST be able to express policies that require the user's consent.

13. The PEEM enabler MUST be able to enforce end-user defined policies even when the end-user is in a visited network (Motivated by Use Cases 5.4).

14. The PEEM enabler specification MUST be able to evaluate (or evaluate and execute) policies that handle authorization requests from other Service Providers.

15. The PEEM enabler specification MUST support the evaluation (or evaluation and execution) of policies for end-users who access resources in a visited network.

16. PEEM enabler specification SHOULD be able to support the override of policies (ones cancelling/pre-empting others) due to different priority levels in different policies.

17. When Policies are established the PEEM enabler SHOULD be able to include mechanisms to facilitate detection of policies incompatible with others already established, i.e. for detection of contradicting policies.

18. The PEEM enabler specification MUST NOT specify any mechanisms for registration and discovery.

19. The PEEM enabler specification MUST support the processing of policy rules according to their priorities.

Appendix A. Change History (Informative)

A.1 Approved Version History

Reference	Date	Description
OMA-RD-Policy_Evaluation_Enforcement_Management-V1_0-20120724-A	24 Jul 2012	Status changed to Approved by TP Ref TP Doc# OMA-TP-2012-0278-INP_PEEM_V1_0_for_Final_Approval

Appendix B. Other Use Cases (Informative)

The following additional use cases have been identified for the PEEM enabler. These are considered incomplete in the sense of not being completely processed or, the essential functionality for PEEM is described by other use cases and requirements already captured in section 5 and 6 of this document.

B.1 Friend-Location-Finder Application

B.1.1 Short Description

This use-case describes the Friend-Location-Finder Application that when requested to do so first checks that end-user A and end-user B are subscribed to the Application, determines if the requesting end-user (end-user A) is a member of end-user B's friend list, retrieves the location of end-user B and then informs end-user A of end-user's B location.

B.1.2 Actors

- Third Party Service Provider;
- Third Party Service Provider Application (generically called Application);
- End-User A;
- End-User B;
- Mobile Operator.

B.1.2.1 Actor Specific Issues

- The Mobile Operator is the Service Provider for end-user A and end-user B;
- The Mobile Operator is the owner of the resource:
    - Enforcing the policies;
    - Providing access to its resource;
    - Enforcing the SLA between the Third Party Service Provider and itself;
    - Negotiates (possibly as part of SLA) billing and interconnect charges between the Third Party Service Provider and itself;
    - Coordinates with the Third Party Service Provider the correct charging of events.
- The Third Party Service Provider is the owner of the Friend-Location-Finder Application:
    - Acts as the Requestor when requesting end-user B's credentials including location;
    - Providing necessary credentials to access the resource;
    - Using the Mobile Operator's resource.

End-user A and B subscribes to the Friend-Location-Finder Application through their Service Provider, i.e. the Mobile Operator. Their subscriptions are paid to the Mobile Operator.

B.1.2.2 Actor Specific Benefits

- The Mobile Operator:
    - Can offer access to resources and their use while enforcing conditions of usage expressed in policies;

  - Knows that resource is appropriately protected;
  - Makes available a wider range of Applications to their customer base;
  - Charging and billing are coordinated per event allowing coordinated charging (or refunds) per event thus enhancing user experience.

- Third Party Service Provider:
  - Can access resources to use for their Applications;
  - Can simplify and automate the way to use a resource belonging to the Mobile Operator: need only to know what credential to pass and how.

- End-user A and B has single point of contact for Application queries and charging and billing issues;
- End-user A and B has a wider range of Applications available for consumption.

B.1.3 Pre-conditions

- End-user A and B must have a subscription with a Mobile Operator. In this use-case both A and B are subscribed to the same Mobile Operator;
- End-user A and B may either have a post-paid or prepaid subscription with the Mobile Operator;
- End-user A and B must have a subscription with the Friend-Location-Finder Application. Both End-user A and B are identified via a valid address (MSISDN) or an Alias ID or session ID;
- Third Party Service Provider has a contractual agreement with the Mobile Operator. This contractual agreement covers aspects such as terms and conditions, establishing payment method for application consumption and establishing privacy settings if applicable;
- Third Party Service Provider Application is registered with the Mobile Operator and is allowed to submit a "is_a_member request" and a "location_request" containing an end-user's identity, e.g. Alias ID;
- The location being requested must be that of a network attached mobile end-user terminal;
- The Third Party Service Provider Application is responsible for converting between coordinate systems belonging to the Mobile Operator and the Third Party Service Provider Application;
- Privacy preference for the targeted end-user (end-user B), and local government legislation must be maintained by the Mobile Operator;
- If end-user A and B are roaming, service experience is not impacted. However, possibly extra charging for message reception must done with the methods employed in the same situation as those described below, i.e. different scenarios should not require special functionality.

The submission of the message is charged.

B.1.4 Post-conditions

- End-user A is presented with the located of end-user B;
- End-user A is correctly charged with the service event.

Mobile Operator and Third Party Service Provider are able to fulfil their SLA agreements, i.e. charging and billing between the two parties is correctly handled.

B.1.5 Normal Flow

1. End-user A accesses the Network of the Mobile Operator in order to access the Third Party Service Provider Application.

2. The Mobile Operator authenticates end-user A (basic authentication to access network).

3, 4, 5, 6 End-user A initiates an "is_a_member" request to the Friend-Location-Finder Application:

- Mobile Operator receives the request via their Network, to access the “Friend-Location-Finder Application”;
- At this point the Mobile Operator:
    - Obtains end-user A Identity information;
    - Obtains end-user A Subscription Profile;
    - Obtains related Access & Authorization information to access Friend-Location-Finder application;
    - Checks that end-user A is allowed to access the Application;
    - Confirms that end-user A is authorized to access the Service;
    - Introduces and associates an Alias ID to the request.

7. End-user A request reaches the Third Party Service Provider Application. End-user A requests location the Third Party Service Provider Application.

8, 9, 10 The Friend-Location-Finder Application identifies the end-user A (possibly through their Alias ID) and ensures that their credentials are authenticated and authorized for the consumption of the Friend-Location-Finder Application.

11, 12, 13 Friend-Location-Finder application initiates a request to the Mobile Operator to check end-user A account details, e.g. adequate funds such as air time credit or money credit (This could be a check for pre-paid subscribers to ensure adequate funds to deliver the service).

14. Mobile Operator initiates an acknowledgement to the Friend-Location-Finder application for successful check for adequate funds.

15, 16, 17 Friend-Location-Finder application prompts end-user "A" to make a request. This request may also contain an Advice of Charge (AoC).

18, 19, 20, 21 End-user A requests location of End-user B from the Third Party Service Provider Application.

22. Friend-Location-Finder Application performs a lookup, through the use of their Alias ID, of the end-user B credentials to determine whether they are registered for the Friend-Location-Finder Application.

23, 24, 25 Friend-Location-Finder application then initiates a request to the Mobile Operator to check end-user's B account details, e.g. Service subscription and adequate funds. This action may require a further level of authorisation (This check may be required to confirm that "B" has still an active subscription to the application, e.g. account not dormant).

26. The Mobile Operator returns a positive acknowledgement to the Friend-Location-Finder Application confirming end-user B account details meet Application usage criteria.

Application requests the location of end-user B from the Mobile Operator. At this moment Mobile Operator has to:

a. Check the identity of the application;

b. Obtain the application Profile information;

c. Check that the application is allowed/authorized to request a charge/reservation/query on end-user A account.

27. The Third Party Service Provider Application initiates a "location_request" for end user "B" towards the Mobile Operator.

28, 29 The Third Party Service Provider Application credentials are authenticated and authorized.

30. The Mobile Operator performs mapping of the provided Alias ID to the Mobile Operator's internally allocated end-user identification, e.g. MSISDN:

- The Mobile Operator checks whether permission has been granted by end-user B for end-user A to find the location of end-use B. This may be done by:

- *Checking end-user B privacy policies;*
- *Checking regulatory policies;*
- *Checking the Mobile Operator's own policies.*

31. The Mobile Operator submits location request to the LES. The Mobile Operator receives the location request report from the LES which includes the geographical co-ordinates of the end-user B mobile terminal.

32. The Mobile Operator returns the location details in "Location_result" response for end-user B to the Friend-Location Finder Application.

33. The Mobile Operator logs the delivery of the request and determines the charge associated with the third party transaction.

34. The Third-Party Service Provider Application receives the Location Report in geographic coordinates.

35. The Third Party Service Provider Application can use other services to transform the geographic location response to the coordinate system requested in the location immediate request.

36, 37, 38 The Third Party Service Provider Application delivers, through the Mobile Operator, the service to end-user "A", i.e. the location of end-user "B".

39, 40 The Mobile Operator provides notification to the Third Party Service Provider Application following the completion and successful delivery of end-user B location.

41. The Mobile Operator charges end-user A for a successful consumption of the service

42. The Third Party Service Provider is charged appropriately according to the SLA.

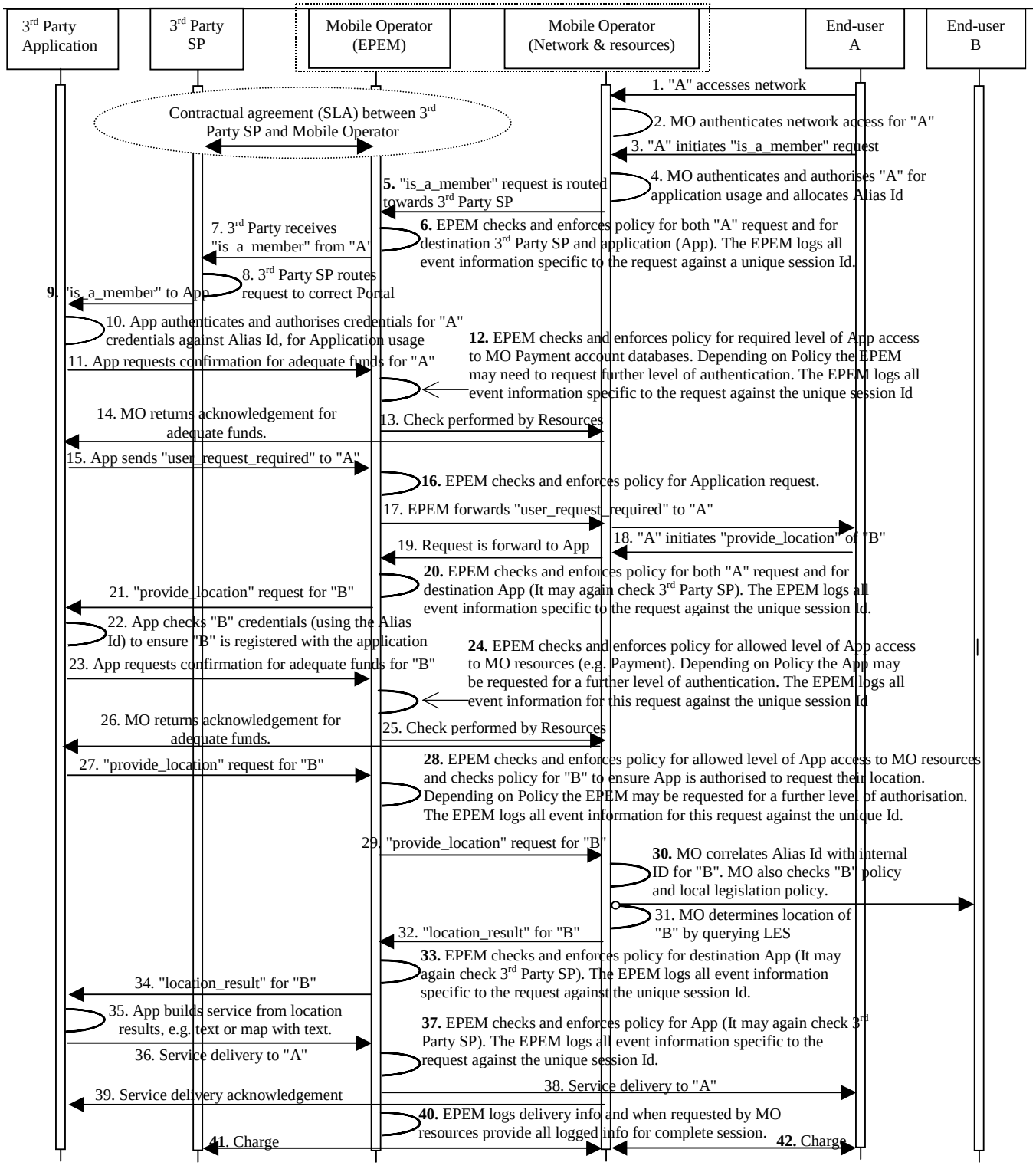

Figure 11: Friend-Location-Finder Application

B.1.6 Alternative Flow

- Authentication fails;
- Authorization failed due to lack of required funds;
- If the MSISDN Alias ID is invalid;
- The third party application may not be able to obtain a location request response due to, for example, Authentication failure;
- In the scenario where the response from the Mobile Operator fails.

The location delivery to end-user A fails and it is therefore not possible for the Mobile Operator to charge the end-user for service consumption.

B.1.7 Operational and Quality of Experience Requirements

- The Mobile Operator is able to set different levels of authorisation for accessing different levels of their resource;
- The user shall have full control over his personal data;
- The Mobile Operator and 3rd party service provider shall adhere to local government legislation.

B.2 Workflow

B.2.1 Short Description

The policies setup by the resource owner implement a set of steps that must be performed by different resources that it owns to provide a particular function or application.

The flows associated to these steps remain the same and do not need to be expanded in this section. The present section focuses solely on the steps that implement a particular function or application, acknowledging that these may include enforcement and validation steps as discussed above.

B.2.2 Actors

- Owner of the function or application:
    - It has set up policies for calls to a particular resource in order to implement the application or function;
    - It provides the function or application by executing the prescribed workflow on requests to the function or application.
- Requestor:
    - Any issuer of request to access and use the function or application:
        - It still provides the necessary credentials to use the resource as it has been informed of.

B.2.2.1 Actor Specific Issues

- Owner of the function or application:
    - Executing the workflow on calls to the function or application in order to implement it.
- Requestor:
    - Using the function or service.

B.2.2.2 Actor Specific Benefits

- Owner of the function or application:
    - Can easily implement services through the workflow specified in the policies;

    - Plus all the benefits enumerated in previous use cases.
- Requestor:
    - Can use the service;
    - Plus all the benefits enumerated in previous use cases.

B.2.3 Pre-conditions

- Owner of the function or application:
    - It has set up workflow to implement the function or application;
    - Plus same steps as in previous use cases.
- Requestor knows function or application.

Requestor knows the conditions it must satisfy if any as in the previous use cases.

B.2.4 Post-conditions

- The request from requestor reaches the function or application and the request is executed.
- The response may be treated through additional policy steps as for use cases above. These may enforce usage conditions or be additional workflow steps to implement the full functionality of the service.

B.2.5 Normal Flow

- Requestor prepares request to function or application and provides information / meta-data / credentials to be able to satisfy the conditions that he/she knows for using the service;
- Request is logically processed by the PEEM enabler (logical entity / mechanism):
    - Request and / or credentials are passed to other resources for action and / or validation of the results as specified by the policies (*) to implement the steps of the workflow.
- Response is returned to the requestor (see post conditions).

B.2.6 Alternative Flow

- At step (*) above, it is possible that some of the validation fail. In such a case, the following cases may take place:
    - The request to the function or application fails and an error message is returned to the requestor;
    - A dialog may be established between the requestor and one of the involved intermediate resource:
        - e.g. please provide a new credential or answer the following challenge.
- Other alternative steps are discussed in the use case sections below.

B.2.7 Operational and Quality of Experience Requirements

- Same as for previous use cases.

B.3 Controlled Exposure of Resources

For all the use cases above, the requestor may be:

- Part of the same domain or system as the resource:
    - To simplify enforcement of policies on any request or to implement delegation and workflows;
    - E.g. another resource within the domain etc.

- A third party requestor part of a system and domain different from the target resource:
    - To implement services or resources and to enforce that the service or resource is securely exposed only to authorized parties;
    - To implement delegation and workflows and enforce steps like billing, logging;
    - E.g. another resource in another domain etcetera.

In all cases, actors and flows remain the same as above and below. Accordingly the other use case sub-sections are skipped.

Note that as already mentioned, the goal is to expose enabler in a controlled manner. It should not be assumed that authentication, authorization, enncryption or charging is always to be enforced or that these are the only conditions that can be enforced.

B.4 Policies for terminal-based Resources

For all the use cases above, the resources may be:

- In the network of the owner domain but not as a terminal;
- On the terminal:
    - When an explicit PEEM entity is present this can be:
        - On the terminal:
            - Actors and flows remain the same as above:
                - Accordingly the other use case sub-sections are skipped for this use case.
        - In the network, processing (by PEEM enabler) any message to and from the terminal including:
            - Within home network;
            - While roaming etcetera;
            - The only differences with respect to all use cases considered before are at the level of the Operational and Quality of Experience Requirements:
                - How does the PEEM enabler determines the policies to enforce;
                - Where are the policies enforced:
                    - Within visited network;
                    - Within the home network, after intercept and redirect to the PEEM enabler:
                        - From the terminal;
                        - From the visited network.

B.5 Discovery of Policies

For all the use cases above, what credentials must be provided and how may be explicitly discovered by the requestor instead of being known from the resource owner through a separate channel.

The pre-conditions do not require any more that the requestor be aware of these conditions. Instead during the normal flow, the requestor can discover this meta-data prior to preparing and generating the request. This typically also involves a registration of the conditions.

The actual policies may be similarly discovered by the PEEM enabler:

- In advance for static policies;
- Prior to any enforcement in all the other cases (or before final validation in the case of section 5.4);

- Again this typically involves registration of the policies.

B.6 Defining the Policies

For all the use cases above, the policies may be:

- Specific to the target resource:
  - Set or derived by user's settings (e.g. derived from privacy considerations);
  - Set or derived by owner's settings;
  - Limited to the delegatable functions that are not performed by the resource.
- Global across all the resources controlled by its owner;
- The result of a combination of policies that are global across all the resources controlled by the owner and policies proper to the target resource.

Using the management interface of the PEEM enabler, the owner of the resource can manage these different policies. The combined policy associated to a resource can be:

- Generated;
- Communicated to requestor as discussed above (in advance or via discovery);
- Communicated to PEEM enabler as discussed above (in advance, via discovery or via update events).

B.7 Debugging the Policies

For all the use cases above, the policies may be debugged:

- For policy expression or logic errors;
- For errors:
  - E.g. by checking dependencies on other resources and availability of these resources.
- For conflicting policies.

B.8 Deploying New Resources

For all the use cases above, in order to satisfy the pre-conditions, the owner of the resource is able to:

- Express/generate the policies as discussed in section C.6;
- Communicate these policies, when needed (as discussed above) to the PEEM enabler;
- Communicate the appropriate subset of conditions to the requestor, when needed (as discussed above).

B.9 Sources of Policies

For all the use cases above, the policies may be:

- Defined by the owner of the resources;
- Derived from settings by others (user's terminal, owner):
  - See for example use case in section 5.5.
- Defined by a third party:

E.g. An enterprise may want to establish particular policies for access of certain resources by its employees or a person wants to let others perform actions on its behalf.

B.10 Prioritization of Policies

Section C.9 indicates that there may be multiple sources of policies. The owner of the PEEM enabler can provide prioritizations rules between these policies.

B.11 PEEM Delegation

For all the use cases above, it is possible that the PEEM that processes messages to and from a resource be provided by a different actor:

E.g. a resource is made available (e.g. exposed through the Operator's network or uploaded) by a third party on an operator's network. PEEM is provided by the operator. Policies are provided by the third party, possibly combined with the global policies of the operator as discussed in section C.6.

B.12 Handling Changes in Policies

B.12.1 Short Description

This use case describes the issues involved with changes of the policy associated to a resource protected by PEEM.

B.12.2 Actors

The involved actors are:

- Service provider that owns a resource (e.g. location server) protected by PEEM;
- Requestor that issue request to the resource.

In addition:

- The PEEM functionality may or may not be provided by the same service provider;
- The requestor may or may or may not be in the same domain as the resource (e.g. an application developer within the service provider domain or a third party application developer).

B.12.2.1 Actor Specific Issues

The issues for the actors are:

- Requestor:
    - Issuing an acceptable request to the resource; independently of the changes of policy (that the requestor should in general not be aware of).
- Service Provider:
    - Ensuring that PEEM is aware of the updated policies;
    - Ensuring that the authorized requestor know how to issue request to the resource at all time.

B.12.2.2 Actor Specific Benefits

The benefits for the actors are:

- Service provider:
    - Being able to manage the policies and change them when dictate by any business or technical reasons;
    - Being able to accommodate cases where users can dynamically change their privacy or service preferences and have this reflected in policies that can be immediately reflected.

- Requestor:
    - Being able to query any resource that the requestor is authorized to query.

B.12.3 Pre-conditions

The required pre-conditions are:

- PEEM protects a resource;
- Policies are set up for the resource;
- Requestor is known of the service provider for example through existing agreement between the requestor and the SP (e.g. SLA):
    - As a result of the agreements above, the requestor is authorized to send requests to the resource;
    - Requestor knows how to issue requests to the resource.

B.12.4 Post-conditions

The required post-conditions are:

- Policies have been changed;
- Requestor has received response to the request that he/she sent to the resource after the change of policies.

B.12.5 Normal Flow

The normal flow for this use case is:

1. The service provider decides to change the policies associated to the resource that he controls.
2. He / she generates a new policies:
    - This can be by editing descriptions of the policies;
    - Or by modifying the policies through a policy management application.
3. The PEEM is provisioned with the new policies.
4. The requestor issues a request to the resource.
5. The request is processed by PEEM.
6. If the policies are satisfied the request is passed to the resource.
7. The request is executed or acted upon.
8. The response is returned to the requestor, possibly further processed by PEEM systems, as defined by the applicable policies.

B.12.6 Alternative Flow

Several alternate flows may take place.

B.12.6.1 Requestor notification

- Prior to step 4, the requestor is informed one way or another (typically when the requestor request the type of information that he / she must provide with a request to the resource)[5] that the policies have been changed and how this may impact the type of request that he/she may have to generate. Depending on how resource interfaces and PEEM is implemented the following alternative exist:
    - The interface communicated to the requestor has been modified to reflect the changes that affect the requestor. This is done in a step 3' before the step introduced above:
        - Typically in such a case the request arguments affected by the changes in policies are not distinguishable from the other request arguments.
    - The interface to the resource is not changed but the requestor is explicitly informed of changes that affect the request that must be issued. No additional step is needed besides the step introduced above;
    - In such a case it is possible to distinguish between arguments that do not depend on the policies and those that do. For example the former can be part of a message payload, possibly encrypted one way agreed with the target resource while.

B.12.6.2 Discovery

- Prior to step 4, the requestor may discover the type of request that he/she may have to generate. Depending on how resource interfaces and PEEM is implemented the following alternative exist:

  The interface registered for the resource and discovered by the requestor reflects the changes of the interface that result from the changes in policies. This is done in a step 3' before step 4. In such a case, the requestor only sees an interface.
    - The interface registered for the resource and discovered by the requestor is not changed but the requestor also discovers one way or another the changes that affect the message that must be issued. Update of these, registration and discovery is done in a step 3" before step 4;
    - In such a case it is easy possible to distinguish between arguments of the request that do not depend on the policies and arguments that do. For example the formers can be part of a message payload, possibly encrypted one way agreed with the target resource while the latter are passed in the header, encrypted in ways understandable by PEEM.

B.12.6.3 Change in the middle of a request

- The change of policies may take place between steps 3 and 4:
    - PEEM may have to reject the request as it may not satisfy the new policies any more (e.g. if the request conditions have changed but the request does not take the changes into account);
    - PEEM may enter a set of exchanges with the requestor to satisfy the new interface if the request does not satisfy the new policies any more (e.g. if the interface have changed because of changes of policies but the request does not take the changes into account).

It is also possible that PEEM does not change its processing of on-going requests and still relies on the older policies.

B.12.6.4 PEEM checks

- The change of policies may take place between steps 5 and 6:
    - The PEEM could check that the policies have not changed in a step 5';

  - If they have changed, the request may be rejected or enter a set of exchange with the requestor as discussed in the case above.

- Step 3 in general could be replaced by having PEEM checking if the policies have changed;
- It is also possible that PEEM does not changes it's processing of on-going requests and still relies on the older policies.

B.12.7 Operational and Quality of Experience Requirements

- If the changes of policies imply that the requestor must provide identity claim, credentials and account information (e.g. for payment) the interface description must describe the need to pass this information and how it should be passed as part of the request or aside of the request. This can be provided as part of the description of the interface to the resource or in a side communication (e.g. meta-information associated to the description of that interface).

  The PEEM should be compatible with the different mechanisms to inform the requestor when the type of requests that must be provided to a resource has changed. This can be because the resource changes (e.g. upgrade) and therefore it has a new interface;
- It should be possible to derive the impacts on the interface from the execution policies:
  - They are in general a subset of the policies or derived from a subset of the policies assertions that they contain (e.g. only the charging, authentication and authentication assertions).
- It should be possible to satisfy the requirements above automatically (i.e. by machine);
- The service provider must be able to express the policies, change them and provision them into the PEEM system.