Device Profile Evolution Architecture

Approved Version 1.0 – 05 Jul 2011

Open Mobile Alliance

OMA-AD-DPE-V1_0-20110705-A

Contents

Figures

1. Scope (Informative)

The scope of the Device Profile Evolution architecture document is to define the architecture for the Device Profile Evolution enabler.

This document details the functional description and architecture for properties on the devices within OMA specifications.

This document fulfils the functional capabilities and information flows needed to support this service enabler as described in the Device Profile Evolution Requirements document [DPE-RD].

2. References

2.1 Normative References

[DPE-RD] "Device Profile Requirements", Open Mobile Alliance™, OMA-RD-DPE-V1_0, URL:http://www.openmobilealliance.org/

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997, URL:http://www.ietf.org/rfc/rfc2119.txt

2.2 Informative References

[ARCH-PRINC] "OMA Architecture Principles", Open Mobile Alliance™, OMA-ArchitecturePrinciples-V1_2, URL:http://www.openmobilealliance.org/

[ARCH-REVIEW] "OMA Architecture Review Process", Open Mobile Alliance™, OMA-ORG-ARCHReviewProcess-V1_4, URL:http://www.openmobilealliance.org/

[OMADICT] "Dictionary for OMA Specifications", Version 2.7, Open Mobile Alliance™, OMA-ORG-Dictionary-V2_7, URL:http://www.openmobilealliance.org/

[OMA-CP] "OMA Client Provisioning", Version 1.1,Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[OMA-DM] "OMA Device Management", Version 1.2, Open Mobile Alliance™, URL:http://www.openmobilealliance.org/

[SEC_CF] "Security Common Functions Architecture", Open Mobile Alliance™, OMA-AD-SEC_CF-V1_0-20070213-D, URL: http://www.openmobilealliance.org/

[UAProf] "User Agent Profile", February 2006, OMA-TS-UAProf-V2_0-20060206-A.pdf, URL:http://www.openmobilealliance.org/

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

Service Provider	An entity that manages and distributes software-based services and content to the end user.
Device Property / Capability	A hardware, software, or network characteristic that represents a capability of a device at a given point in time. Throughout the document, the words "property" and "capability" can be used indistinctly. A device property can be either static or dynamic.
Dynamic Device Property / Capability	A device property that may change its value e.g. as a result of hardware, software or configuration changes.
Static Device Property/Capability	A device property that does not change its value. Examples are display resolution, processor type, etc.

3.3 Abbreviations

DPE	Device Profile Evolution
OMA	Open Mobile Alliance
OTA	Over The Air
SP	Service Provider

4. Introduction (Informative)

Mobile applications and services are required to function in varying network environments with users having devices with a wide range of capabilities. The device capabilities and network conditions can vary dynamically and applications and services need to be able to respond to these changes accordingly.

The Device Profile Evolution enabler is intended to provide a standardized solution to convey information on the device capabilities. While other enablers such as UAProf [UAProf] can only convey information on static device capabilities, the DPE enabler can convey information either on static device capabilities or on dynamic device capabilities, the dynamic aspect being the main added-value of the DPE enabler. This information, in the form of property names and values, directly comes from the device and is communicated to a Service Provider (SP), allowing an enhancement of the quality of the services provided to the device. This enhancement could be, for example, a better content adaptation, or services matching to the maximum extent the device capabilities, resulting at the end on a better user experience.

DPE architecture follows a client-server model. Therefore, in the existing OMA architecture two new entities will be created: a DPE Server and a DPE Client.

Any SP willing to get information on the device capabilities of a given user using the DPE enabler has to submit its requests to the DPE Server managing the DPE Client of that given user. An SP can also transmit policies to DPE Server. The policies will be applied at the DPE Client level to the associated device capabilities (see §5.4.2.3 for more information on the policies). The use of policies will be further detailed in the DPE Technical Specifications document.

The DPE Server's main tasks will be to:

- parse and verify the validity of the requests sent by the SP
- verify the coherence of the policies wanted to be applied by the SP
- transmit the validated requests and policies to the DPE Clients
- manage the Authentication and Authorization of the DPE Clients
- manage all the policies assigned to the DPE Clients
- cache the device property values previously retrieved, in order to minimize the impact of the DPE communications on the over the air interface
- communicate the responses to the SP at the origin of the request

The DPE Client, installed on the device, will perform the following tasks:

- retrieve the device’s internal information about the device capabilities, and communicate the values when needed and/or requested
- accept and apply the policies transmitted by the DPE Server

The creation of the Core Vocabulary of device capabilities will take place in the DPE Technical Specification Document. Each device capability will be associated with a validity period, which will be used by the DPE Server for its caching functionality. An infinite validity period means that the device capability is a static device capability.

It is not within the scope of the DPE to provide detailed interfaces to other existing or future Device Description solutions.

4.1 Planned Phases

The specification that is going to be derived from this architecture is DPE 1.0. Therefore, this is the first version of the architecture, and there are no additional phases planned beyond this architecture to cover the requirements described in the DPE 1.0 RD [DPE-RD].

4.2 Security Considerations

This section provides a brief high level description of the DPE Security functions, the details of which will be described in the Technical Specifications document.

4.2.1 Authentication and authorization

A DPE Client must be able to authenticate a DPE Server based on credentials that are provisioned in the DPE Client. These credentials may be just limited to the IP address of the DPE Server. Once a DPE Server is authenticated by a DPE Client, it gets full authorization to send any DPE request to that DPE Client.

A DPE Server must also be able to authenticate any DPE Client it is communicating with. The credentials on which this authentication is based includes at least the DPE Client Id assigned by the DPE Server to the DPE Client during the registration process (see §5.4.1). Once authenticated by a DPE Server, a DPE Client is only authorized to advertise information pertaining to its device properties.

4.2.2 Confidentiality, data integrity and protection against security threats

No data encryption is required for DPE communications, though the DPE enabler provides some mechanisms for confidentiality and data integrity. In addition, the DPE enabler includes mechanisms to protect against security threats such as denial-of-service attacks.

These mechanisms will be detailed in the Technical Specifications document.

5. Architectural Model (NORMATIVE)

5.1 Dependencies

OMA CP [OMA-CP] or OMA DM [OMA-DM] MAY be used to provision the connectivity information for the DPE Client.

It is foreseen, however, that the OMA Security Common Functions [SEC_CF] might be a source of dependencies in the TS phase.

5.2 Architectural Diagram

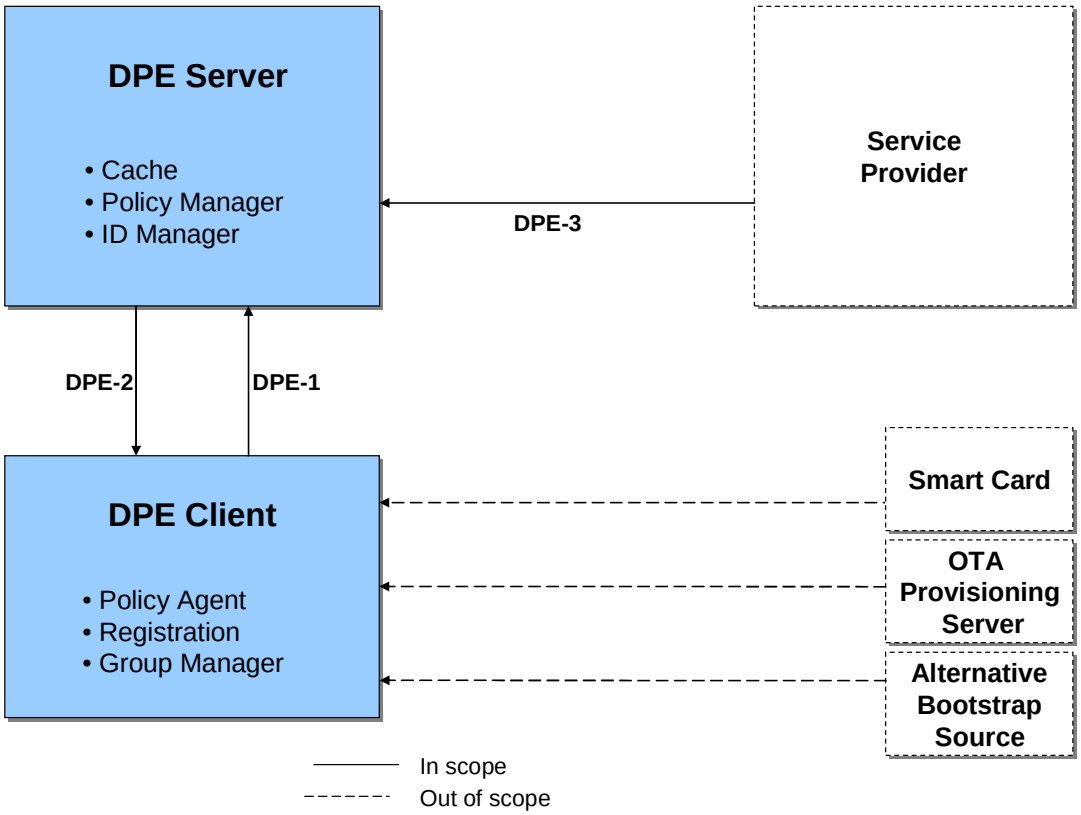

Figure 1: Diagram of DPE Architecture

5.3 Functional Components and Interfaces

5.3.1 Components

5.3.1.1 Internal Components

5.3.1.1.1 The Device Profile Evolution Server

The DPE Server is the entity that communicates with any DPE Client for the purpose of:

- assigning a unique identifier (using Interface DPE-1)
- requesting the value(s) of a single (or a group of) device property (ies) (using Interface DPE-2)
- transmitting one or several policies (using Interface DPE-2)

The DPE Server also communicates with any DPE enabled SP for the purpose of:

- processing any request concerning the retrieval of the value(s) of a single (or a group of) device property (ies) (using Interface DPE-3)
- advertising these values depending on their availability and on the policies applied, if any (using Interface DPE-3)

The DPE Server includes three main functions:

- the ID Manager in charge of assigning a unique identifier to a DPE Client during the registration phase
- the Policy Manager in charge of managing the current policies associated with the DPE Clients
- the Cache, in charge of caching the values of the device properties, according to the predefined duration of validity of each device property (these durations are defined in the DPE Core Vocabulary)

5.3.1.1.2 The Device Profile Evolution Client

At first the DPE Client has to get the necessary information to be able to initiate a connection with a DPE Server (from a Smart Card, an OTA Provisioning Server or any alternative bootstrap solution).

Then, the DPE Client communicates with the DPE Server in order to:

- ask for a unique identifier;
- provide the value(s) of a single (or a group of) device property (ies) according to the potential policies applied.

The DPE Client includes three main functions:

- the Registration function in charge of retrieving a unique identifier;
- the Policy Agent, in charge of handling the policies applied to the device properties in order to advertise their values accordingly;
- the Group Manager, in charge of managing the labelled groups of device properties.

5.3.1.2 External components

5.3.1.2.1 The Service Provider

The SP is an entity that distributes services and delivers content to the end-user. In order to provide the end-user with the content best suited to its device capabilities, the SP communicates to a DPE Server the list of device properties it is interested in and the policy to be followed to advertise the values of these device properties and their changes through time.

Any alternative device description entity wishing to retrieve information on the device capabilities of a given terminal through the DPE enabler would be considered like an SP from a DPE Server's point of view.

5.3.1.3 Smart Card, OTA Provisioning Server, and Alternative Bootstrap Solution

These components are the potential sources from which the DPE Client initially retrieves the necessary information to be able to initiate a connection with a DPE Server.

5.3.2 Interfaces

5.3.2.1 The DPE–1 Interface

The DPE-1 interface allows the DPE Server to assign a unique identifier to each registering requestor. This unique identifier will be used afterwards for all the subsequent DPE communications..

5.3.2.2 The DPE–2 Interface

The DPE-2 interface allows to request the value(s) of a single (or a group of) device property (ies), to transmit policies and to respond to those requests and policies.

5.3.2.3 The DPE–3 Interface

The DPE-3 interface is used to request a DPE Server for the value(s) of a single (or a group of) device property (ies) of a given DPE Client. The DPE-3 interface is also used to transmit policies associated to the device properties it is asking for.

Any entity wishing to retrieve the value(s) of a single (or a group of) device property (ies) of a given DPE Client and to transmit policies must use the DPE-3 interface.

5.4 Flows

5.4.1 Registration process

After the first boot of the device where the DPE Client is installed, the DPE Client needs to get the necessary information to be able to initiate a connection to a DPE Server. The retrieval of this information can be issued from different sources: through factory configuration of the DPE Client, information provisioned via a smartcard, information sent by an OTA Provisioning Server, etc. OMA CP [OMA-CP] or OMA DM [OMA-DM] MAY be used for the provisioning purpose.

Once the DPE Client is able to initiate a communication with a DPE Server, the Registration function asks the ID Manager function of the DPE Server for the allocation of a unique identifier. This unique identifier will be communicated afterwards by the DPE Client to the ASPs, which will use it as a key to retrieve, through the DPE Server, the values of the device properties of the associated DPE Client.

During the registration process, the DPE Client version is communicated to the DPE Server, so that the DPE Server is aware of the list of device properties supported by this given DPE Client.

5.4.2 Retrieval of the device capabilities

5.4.2.1 Retrieval of a single device property value

This flow describes how an SP retrieves the current value of a device property from a DPE Client hosted on the end-user's device:

1. A User Agent (e.g. a browser) hosted on the end-user's device sends a request for content or for a service to an SP. This request includes the address of the DPE Server managing the DPE Client and the DPE Client ID assigned by the DPE Server during the registration process.
2. Upon receiving this information, the SP sends a request to the DPE Server containing the DPE Client ID and the specific device property whose value is needed.
3. After a verification of its validity, the DPE Server forwards the request to the DPE Client.
4. If the requested device property is supported, the DPE Client sends its value to the DPE Server.
5. The DPE Server forwards the DPE Client's answer to the ASP.
6. The SP delivers its content or service to the User Agent based on the response received from the DPE Server.

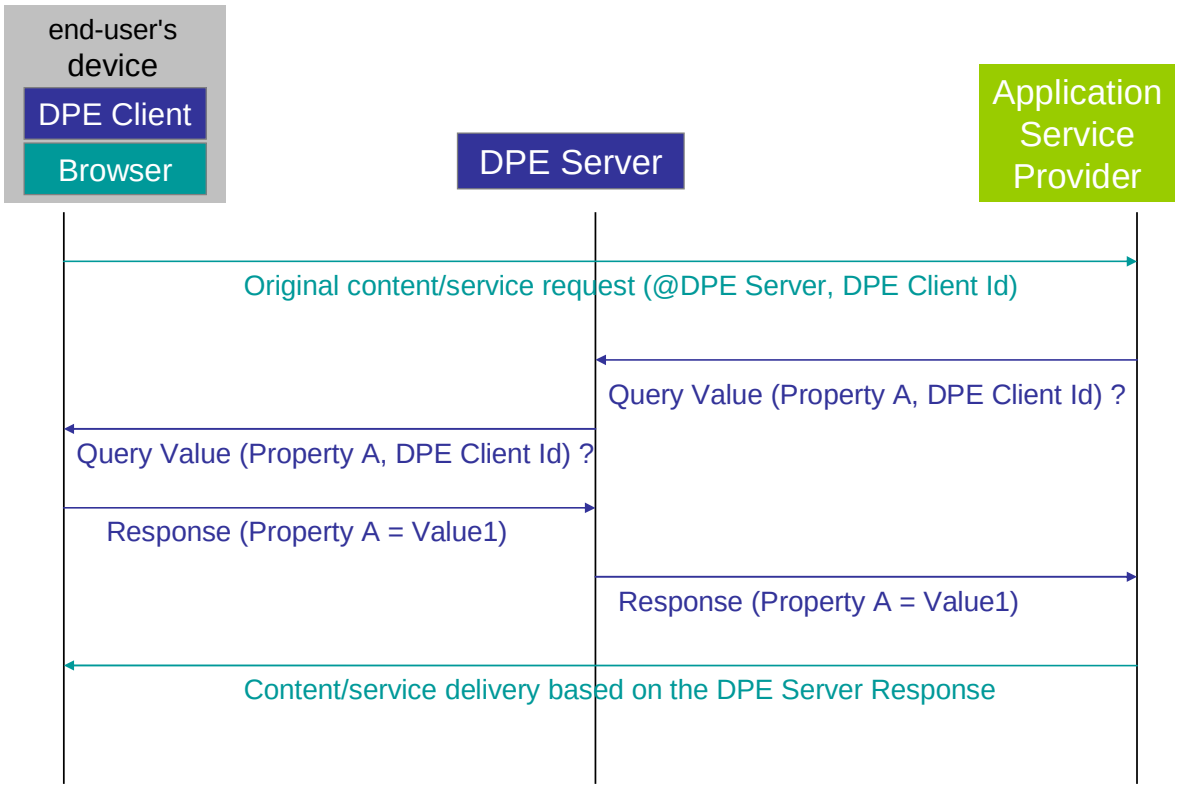

Figure 2: Retrieval of a single device property value

5.4.2.2 Retrieval of an unsupported device property's value

In case an SP requests a device property that is not supported by the version of the DPE Client installed on the end-user's device, the DPE enabler prevents to unnecessarily overload the OTA interface between the DPE Server and the DPE Client with this pointless request. Indeed, as the DPE Server is aware through the registration process (see §5.4.1) of the version of DPE Client installed on the end user's device, it knows the list of device properties supported by the targeted DPE Client. Therefore, it can directly answer to the SP with the adequate error message, as illustrated by the flow below.

This flow describes the DPE messages exchanged when an SP tries to retrieve the value of an unsupported device property from a DPE Client hosted on the end-user's device:

1. A User-Agent (e.g. a browser) hosted on the end-user's device sends a request for content or for a service to an SP. This request includes the address of the DPE Server associated with the DPE Client and the DPE Client ID assigned by the DPE Server during the registration process.
2. Upon receiving this information, the SP sends a request to the DPE Server containing the DPE Client ID and the specific device property whose value is needed.
3. The DPE Server sends an error message to the SP to notify it that the targeted DPE Client doesn't support the requested device property.
4. The ASP delivers the content or service to the User-Agent, with no possible adaptation based on the requested device property.

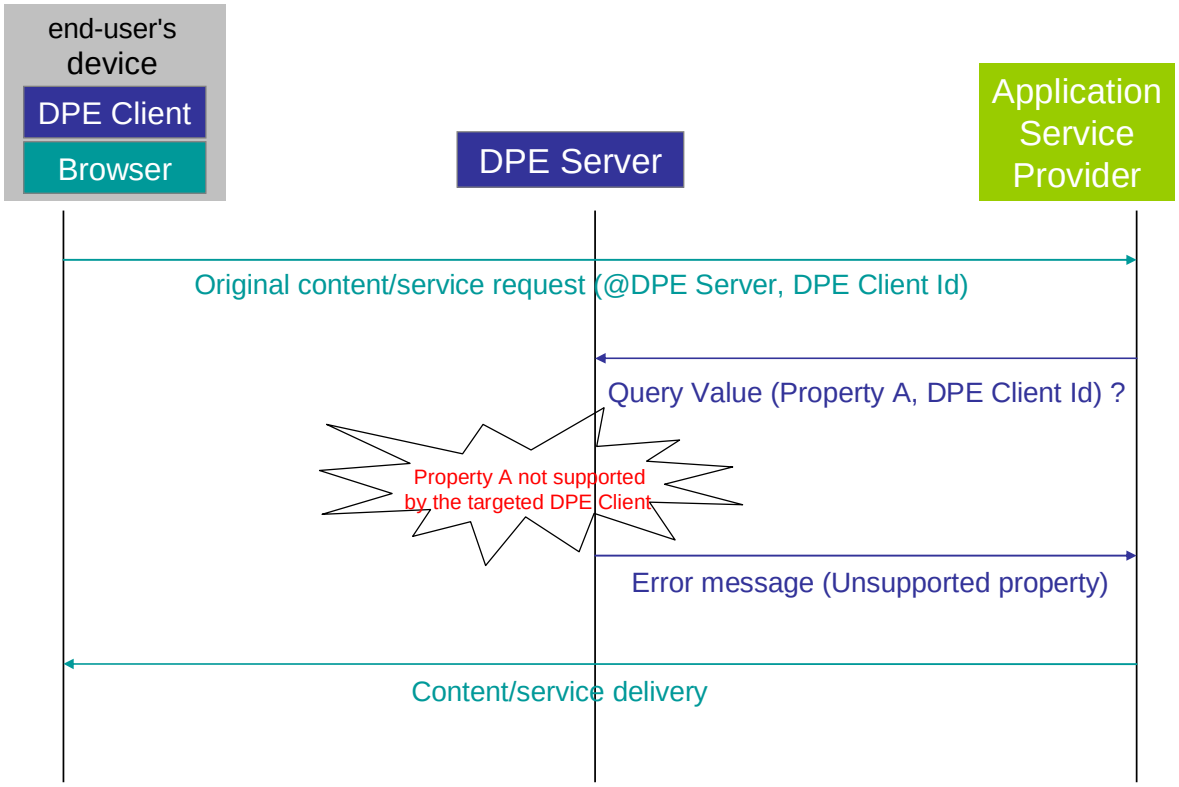

Figure 3: Retrieval of an unsupported device property value

5.4.2.3 Use of the cache functionality of the DPE Server

In order to improve performance and also limit costly over-the-air communications between the DPE Client and the DPE Server, a cache function must be implemented within the DPE Server.

Each attribute of the DPE Core Vocabulary that will be defined in the DPE Technical Specifications will be associated with a predefined cache lifetime. Based on that duration, the DPE Server will store through its cache function the value of any device property previously requested by a SP. Hence any subsequent request of a cached device property coming from the same or from another ASP will not require further communication between the DPE Server and the target DPE Client, as illustrated in the flow described below.

1. A User Agent (e.g. a browser) hosted on the end-user's device sends a request for content or for a service to a SP. This request includes the address of the DPE Server managing the DPE Client and the DPE Client ID assigned by the DPE Server during the registration process.
2. Upon receiving this information, the SP sends a request to the DPE Server containing the DPE Client ID and the specific device property whose value is needed.
3. The DPE Server verifies the validity of the request.
4. If the value of the device property requested is cached, the DPE Server answers directly to the ASP.
5. The ASP delivers its content or service to the User Agent based on the response received from the DPE Server.

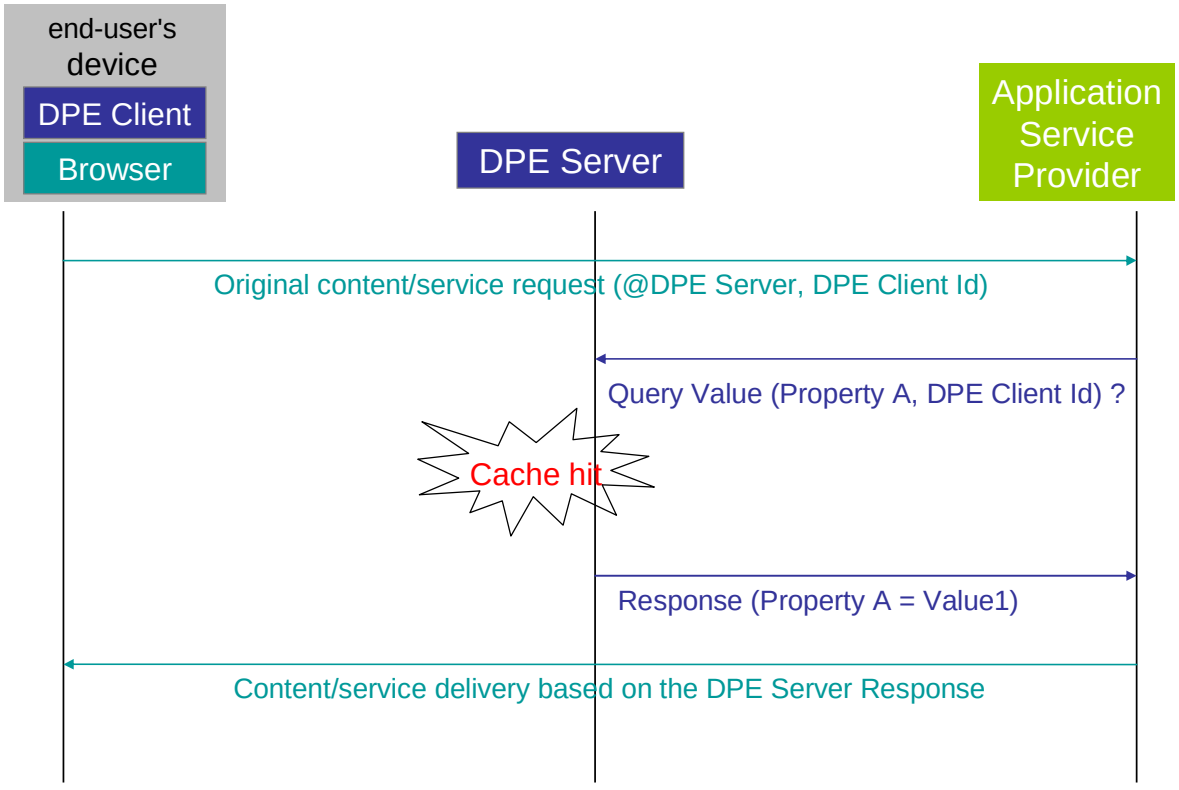

Figure 4: Use of the cache functionality of DPE Server

Remark: The cache functionality is illustrated here through the simplest example, but it can be activated whenever useful, be the request complex (use of several policies, of labelled groups of device properties, etc) or not.

5.4.2.4 Retrieval of a single device property value according to a policy

A policy represents, broadly speaking, the manner in which information is required by a SP. Indeed, a SP may want to retrieve device capabilities information only under certain circumstances: triggered by a specific event (timer expiry, threshold hit, etc.) or periodically. The following propositions further define the use of policies:

- A policy can be assigned for each one of the device properties requested by an SP to a DPE Client. It can also be assigned to a labelled group of properties (see §5.4.2.5 for more information on the labelled groups).
- A given SP may also assign several policies to the same property, as long as the policies are not contradictory. An example of contradictory policies would be to ask for the advertisement of the battery status only if it falls under 10% of the maximum charge and only if it goes over 50% of the maximum charge. These two conditions cannot be true at the same time. The Policy Manager function of the DPE Server is in charge of verifying the coherency of the policies submitted by the SP.
- Several SP can assign policies to the same device property on the same DPE Client. The Policy Agent function of the DPE Client handles the policies for each ASP separately.
- The use of a policy makes sense only for the duration of a service or a content being delivered by a SP to an end user. The policies assigned by the SP should be released at the end of the service or content delivery to prevent the DPE Client's Policy Agent and the DPE Server's Policy Manager from having to manage needless policies. This is achieved either through a release-policy message sent by the SP to the DPE Server, and then forwarded to the DPE Client, or through a predefined timer at the DPE Server's Policy Manager which releases the policy at timer expiry.

This flow illustrates the use of policies.

1. A User Agent (e.g. a browser) hosted on the end-user's device sends a request for content or for a service to a SP. This request includes the address of the DPE Server managing the DPE Client and the DPE Client ID assigned by the DPE Server during the registration process.
2. Upon receiving this information, the SP assigns a policy via the DPE Server to a specific device property and starts delivering content or service in parallel with assigning policy via the DPE Server to a specific device property.
3. After a verification of the syntax and the currency of the policy, the DPE Server forwards the request to the DPE Client.
4. The policy requested by the SP is handled by the Policy Agent function of the DPE Client.
5. The DPE Client sends an acknowledgment of the established policy back to the SP via the DPE Server.
6. If the established policy conditions are met, the Policy Agent function of the DPE Client notifies the DPE Server, which forwards the notification to the SP.
7. The SP adjusts the content or service delivery as appropriate based on the received policy notification.

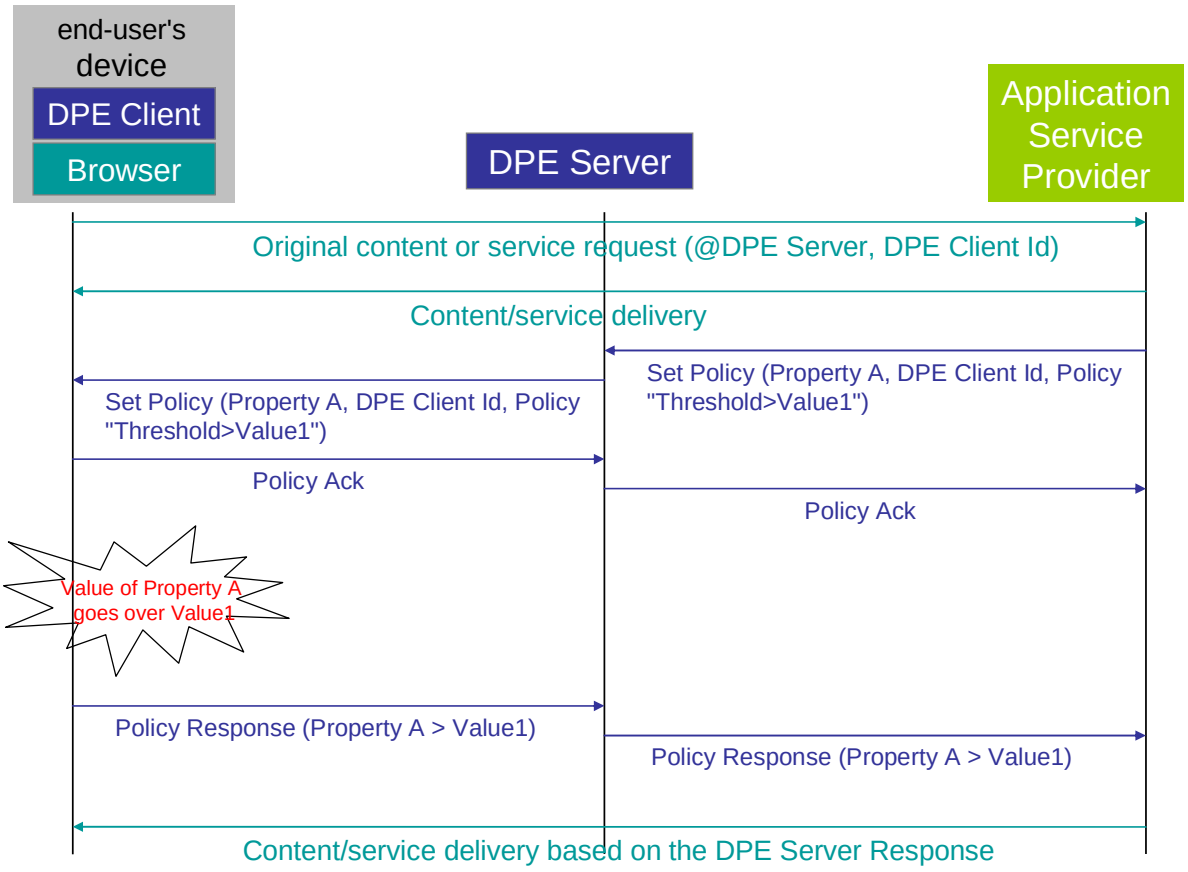

Figure 5: Retrieval of a single device property value according to a policy

Remark: The policy illustrated in this flow is very simple ("if-changed" based), but the policies can be much more complex, based on a periodicity, based on a threshold, etc. Besides, each device property requested can be associated with a distinct policy.

5.4.2.5 Retrieval of a labelled group of device property's values

The ASP can group several device properties and give a label to the group. This flow illustrates the use of labelled groups of device properties.

1. A User Agent (e.g. a browser) hosted on the end-user's device sends a request for content or for a service to a SP. This request includes the address of the DPE Server managing the DPE Client and the DPE Client ID assigned by the DPE Server during the registration process.
2. Upon receiving this information, the SP sends a request about several device properties' values to the DPE Server. The SP also groups all these device properties and labels the group.
3. After a verification of its validity, the DPE Server forwards the request to the DPE Client.
4. If the requested device properties are supported, the DPE Client sends their values to the DPE Server.
5. The DPE Server forwards the DPE Client's answer to the SP.

6. The SP delivers its content or service to the User Agent based on the response received from the DPE Server.
7. After a while, the SP needs to retrieve the current values of the same device properties grouped and labelled under one name.
8. The SP sends a request to the DPE Server including the identifier of the DPE Client and the group label as defined previously.
9. After a verification of its validity, the DPE Server forwards the request to the DPE Client.
10. Upon receiving this request, DPE Client recognizes the group label and sends the current values of all the device properties pertaining to that group.
11. The DPE Server forwards the DPE Client's answer to the SP.
12. The SP continues to deliver its content or service to the User Agent based on the up-to-date property values retrieved through the DPE Server.

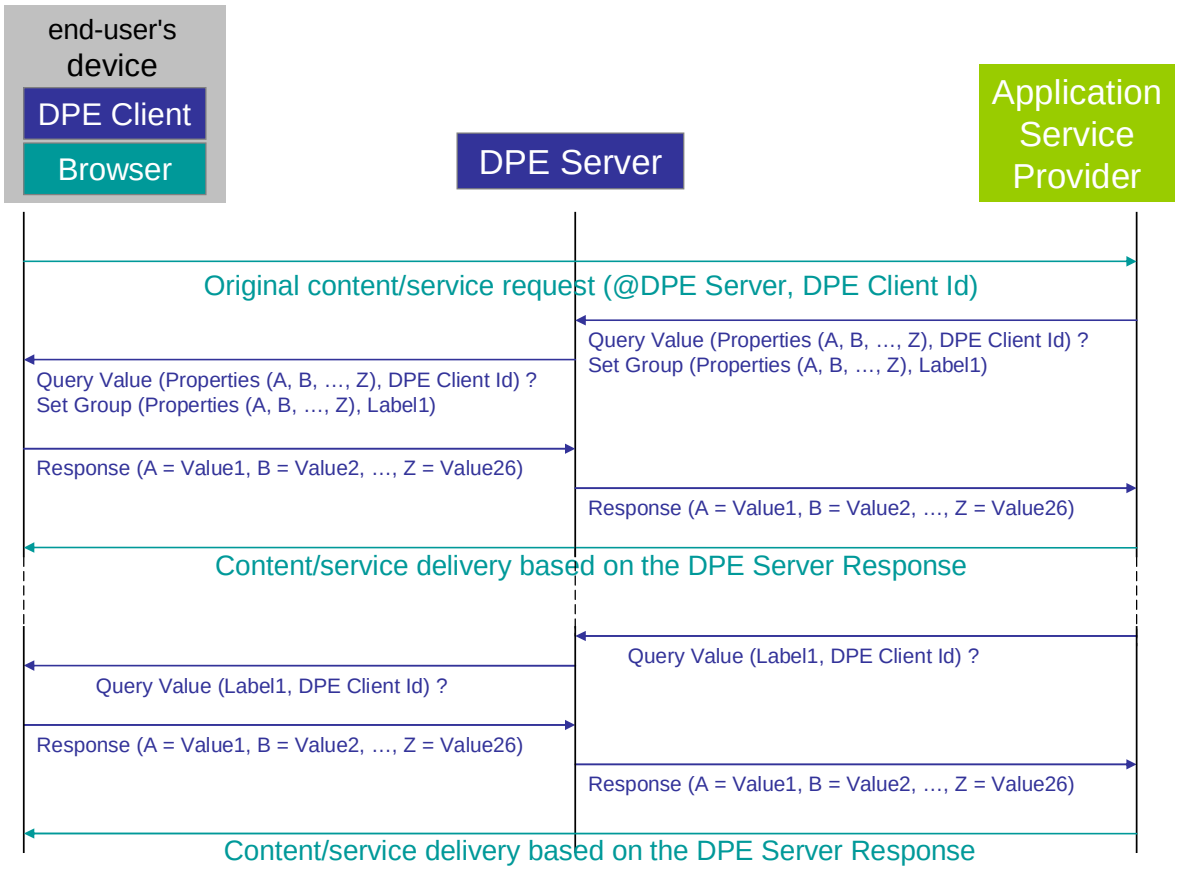

Figure 6: Retrieval of a labelled group of device property's values

Remark: Policies can be associated to a labelled group and/or to the device properties forming the group.

Appendix A. Security assessment (Informative)

A.1 System assets

Asset	Description	Asset importance
DPE server	acts as proxy between ASP and DPE client	Critical
DPE client	manage capability	High

Responses to the checklist of the Security Risk Assessment Guide (OMA-ORG-SRAG-V1_0-20070613-D):

- Components installation

  Other than the Provisioning of the DPE Client, there is no component installation as such.

- Components execution
  - Is the component validated before each execution or while running?
    - To be defined further in the Technical Specifications
  - Does the component rely upon any external components?
    - DPE Client may rely on external component for the provisioning phase.
  - Does the component provide data input checking? (client-side or server-side)
    - DPE Server checks requests for syntax, currency, etc.
  - Does the component accept partial data? sanitize bad data?
    - No.
  - Have very large data input been taken into account?
    - No.
  - Does the component support multiple input formats (e.g. Unicode representations) or extended characters? (e.g. NUL byte injection)?
    - To be defined further in the Technical Specifications
  - Can error management reveal internal component information?
    - N/A

- Authentication sessions
  - How authentication credentials are originally supplied to the client?
    - During provisioning, DPE Client gets DPE Server's IP address. Other credentials might be supplied during the registration phase.
  - Can the credentials be changed, updated?
    - The credentials are fixed and can not be changed without incurring a re-Provisioning step and/or re-Registration step.
  - Which party initiates the session?
    - Registration : DPE Client
    - Query : DPE Server
    - Policy Set : DPE Server
  - Is authentication information stored on the client?
    - Yes
  - Is there a client lockout procedure?
    - To be defined further in the Technical Specifications
  - Are there some timeouts defined (after a period of inactivity, to force re-authenticating…)?
    - To be defined further in the Technical Specifications
  - Can the client initiate a logout procedure?
    - No.

  - Is it possible to deduce other client credentials from a known set of credentials?
    - No.
  - Are confidential authentication details displayed on screen if the user needs typing them in?
    - No. No user typing of anything.

- Data exchange
  - Is the communication channel encrypted end-to-end, partially?
    - Encryption might be needed during registration phase.
  - Is it possible to hijack the communication channel?
    - It might be possible.
  - Is it possible to DoS the communications channel?
    - Yes. Rate limiting logic would be needed on the DPE Server and possibly on the DPE Client.
  - Can input data be obfuscated, and then be used to carry attack payloads?
    - Yes. But response would go to "true" DPE Server, and recovery possible.
  - Are cached content or transfer files stored locally?
    - Yes. For DPE Server.

- Cryptography
  - Are known cryptographic algorithms used to:
    - encrypt local data storage?
      - Out of scope. Implementation choice.
    - encrypt communications?
      - For registration phase if required.
  - Are selected standardized cryptographic algorithms referring to their most recent specification?
    - To be defined further in the Technical Specifications
  - Does the system allow several cryptographic levels for a given asset? Is it possible to select or force the level of cryptography to an insecure level?
    - To be defined further in the Technical Specifications
  - If digital certificates are used, what are their sizes? Are they unique per entity or can they be transported between hosts?
    - Not decided yet.
  - Are checksums or hashes available for verifying the integrity of the application components?
    - Out of scope. It is an implementation choice.

- Operational constraints
  - What type of network domain relations exist between the involved entities (mode of deployment)?
    - Client-Server deployment model.
    - All requests/responses are centralized through the DPE Server.
    - Transport protocol decision hasn't been made yet (but should be over IP)
  - Are critical operations logged?
    - components installation
      - N/A
    - software failures and errors
      - DPE Server might log these but not DPE Client
    - authentication at both client and server hosts
      - DPE Server might log these but not DPE Client
  - Is it possible to retrieve confidential authentication information from logs?
    - Possibly, depending on what is logged.
  - Is there any privacy concern related to DPE client information?
    - To be defined further in the Technical Specifications

A.2 Threats analysis

The table below lists the potential threats that may target the DPE enabler.

Risk #	Assets	Threat category / Description	Attack	Prerequisites	Expertise/Cost/Diffusion
1	DPE Server	Availability: Service disruption	DoS, TCP/IP flooding	Needs to know server IP address	Script kiddies
2	DPE Server	Availability: Service, disruption, Cache saturation	DoS	Needs to know server IP address Valid DPE requests	Hackers
3	DPE Server	Data integrity: fake associations MSISDN-DPE Client ID pushed to the DPE Server, possible deletion of the real user data	DoS	Valid DPE registration requests (radio) network access	Hackers
4	DPE Server	Data integrity: Request (or responses) sent to (or from) the DPE Server are modified, resulting in a degraded service or content delivered to the end-user (since it is based on fake device capability information)	Man-in-the-middle	Valid DPE requests	Hackers
5	DPE Client	Data integrity: fake policy update to force permanent battery-consuming OTA connection	Software-based IP spoofing at IP stack level	SW upload	Script kiddies
6	DPE Client	Same, but attack coming from the radio network	IP spoofing at network level	(radio) network access	Hackers
7	DPE Server/DPE Client	Privacy violation: interception of the information exchanged between the DPE Client and the DPE Server	Man-in-the-middle	(radio) network access	Hackers

A.3 Impact analysis and risk mitigation

The tables below characterize the potential threats identified in section B.2.

Risk #	Likelihood of Occurencey	Impact severity	Risk level
1	High	Major	High
2	Low	Major	Moderate
3	Medium	Critical	High
4	Low	Major	Moderate
5	Medium	Major	High
6	Low	Major	Moderate
7	Medium	Minor	Low

Risk #	Resolution	Observations
1	assumption	Security of underneath network infrastructure is out of the scope of OMA DPE.
2	limitation	
3	avoidance	Security of underneath network infrastructure is out of the scope of OMA DPE.
4	limitation	
5	assumption	Security of the terminal is out of the scope of OMA DPE. Though the specification should clearly emphasize on the need to have certified software installation in a terminal.
6	assumption	Network access security is out of the scope of DPE, and OMA enablers are typically deployed on secured access networks.
7	assumption	Network access security is out of the scope of DPE, and OMA enablers are typically deployed on secured access networks. In addition, device capabilities information is not considered as sensitive data.

A.4 Security recommendations

The SEC Group provided a list of recommendations that could help to resolve potential security threats for DPE.

1. Mutual authentication of the DPE Client and the DPE Server. Credentials are detailed in the Technical Specifications.

2. Link User-Agent request to DPE Client in order for the DPE Server to authenticate the validity of the DPE request coming from the Service Provider and for the DPE Client to validate the request coming from the DPE Server. Possible ways to do that:

   - shared key between DPE server and DPE client to authenticate the request (key derivation based on DPE client ID, provisioned at bootstrapping phase)
   - token based: challenge AND response sent by the User-Agent to allow the DPE Server to avoid DoS attacks
   - replay protection

   Drawbacks:

   - pre-provisioning DPE Client ID and DPE Server IP address may not be sufficient
   - necessity to define an API for challenge/response retrieval from the DPE client by the User-Agent
   - all User-Agents targeted (browser, email client, …) must implement the API
   - management of replay protection at server level

Appendix B. Change History (Informative)

B.1 Approved Version History

Reference	Date	Description
OMA-AD-DPE-V1_0-20110705-A	05 Jul 2011	Status changed to Approved by TP: OMA-TP-2011-0225-INP_DPE_V1_0_ERP_for_Final_Approval