

CC/PP Negotiation of a mobile station in MExE service environment

Markus Sihvonen

VTT Electronics

PL 1100

90750 Oulu, Finland

markus.sihvonen@vtt.fi

Abstract: MExE (Mobile Execution Environment) is a standardised middleware platform in an MS (mobile station) where to execute 3rd generation applications and services regardless of the operating system. It is one of the 3GPP's (third generation partnership project) VHE (Virtual Home Environment) toolkits, perhaps the most complex but also the most flexible and diversified one. It enables a dynamic download of applications to an MS at a users' request. Currently, applications can be written in Java and WML (Wireless mark-up language), later possibly with many other programming languages. CC/PP (Composite Capability / Preference Profile) negotiation normally occurs prior to a download of an application to an MExE capable MS and it is used to define the right version of the requested service or application to that particular MS. Even though the technology is being under development particularly for 3rd generation mobile networks and MSs, it can be also applied to 2nd generation mobile technology. This paper focuses on CC/PP negotiation of the 3rd generation all IP (Internet protocol) MSs. The purpose is to identify problems of the CC/PP negotiation and to propose a solution for the negotiation strategy. At the very beginning, the problem area will be introduced to the reader.

1 Introduction

MExE is a part of a larger entity called VHE, which is being standardised by the 3GPP. The VHE is defined as a PSE (personal service environment) that expands beyond boundaries of single network and between different MSs. The underlying concept of the VHE is that users are able to gain access to the same personalised features, including customisation of the user interface and selection of services, regardless of the physical location of a user, a used network or an MS. Naturally, the limitations are imposed by the actual capabilities of the used MS and the network. The 3GPP currently identifies OSA (Open Service Architecture), MExE, SAT (SIM Application Toolkit) and CAMEL (Customised Applications for Mobile network Enhanced Logic) among the most important mechanisms to support the VHE concept. [3G00C]

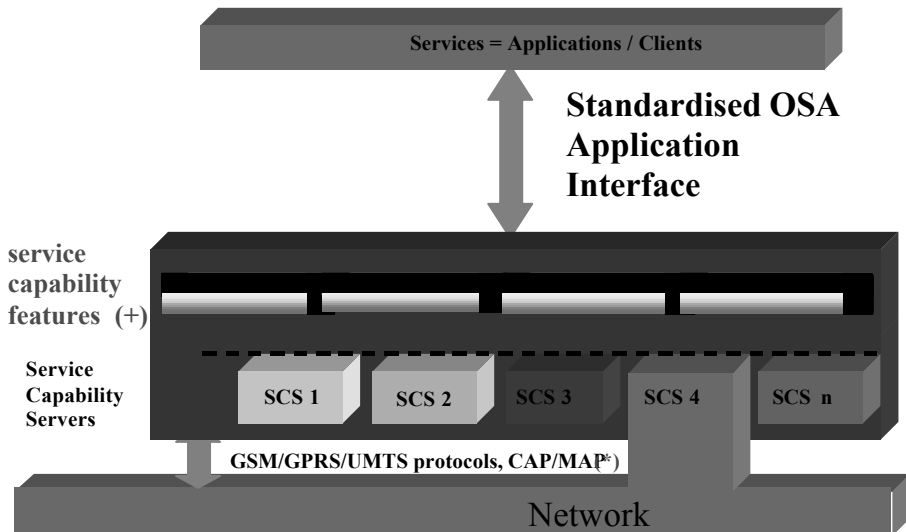


Figure 1.1 Realisation of the framework for services

The function of the OSA is to define the overall architecture that enables services to utilise the network functionality via an open standardised API (Application Protocol Interface). The OSA is located between services and SCs (service capabilities) provided by the network and its mission is to enable services to be independent from the underlying network technology. The services are at the top most level in the OSA, as illustrated in Figure 1.1. The OSA API connects the SCSs (Service Capability Servers) and services. The SCSs' purpose is to hide the network's complexity from the services by bringing together the underlying telecom specific protocols, such as MAP (Mobile Application Part), CAP (Camel Application Part), and OSA API.

Services can be executed either in an MS or in a server that is located somewhere in the network. The 3GPP's standardised execution platforms enabling service execution in an MS are MExE and SAT. The services executed in a network utilise service capability features provided by the OSA API, and also some legacy mechanisms such as CAMEL may be applied. [3G00C]

The research work has been focused on the MExE toolkit, which is a framework that enables the execution of dynamically downloadable services in an MS. The Figure 1.2

illustrates the general MExE service architecture, including a list of possible services it can provide for end users. The MSE (MExE Service Environment) can be composed of several service nodes each providing MExE services that can be transferred to the MS via different transfer mechanisms, such as fixed network protocols, mobile network protocols, Bluetooth and standard Internet protocols. The list of the transfer mechanisms is not exhaustive. The service nodes can be located in the CS (Circuit Switched) domain, PS (Packet Switched) domain, IMS (IP Multimedia Subsystem) or in the Internet space. The MSE may also include a proxy server to translate the content defined in standard Internet protocols into their wireless optimised derivatives. A wireless network shall provide an MS with access to a range of bearer services on the radio interface to support application control and transfer from MSE. Since the MExE toolkit can be utilised in fixed and cordless environments, an MExE MS may also access MExE services via non-wireless access mechanisms.[3 G01]

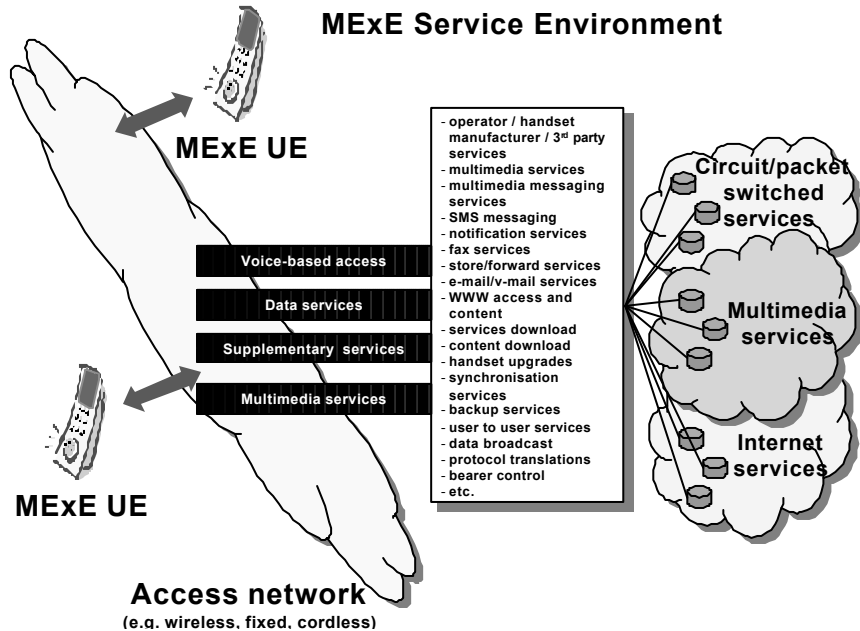


Figure 1.2 The generic MExE architecture

Currently, MExE's technical specification determines three different classmarks and a group of generic requirements that are applicable across the classmarks. The classmark 1 is based on WAP (Wireless Application Protocol) and it is aimed for less powerful MSs. The classmark 2 includes the Personal JAVA with the addition of JAVA Phone API. This classmark is focused for consumer electronic devices such as PDA equipment and powerful mobile phones e.g. NOKIA communicator. The classmark 3 is a new addition to the specification 23.057 in release 4 and it is based on CLDC (Connected Limited Device Configuration) with the added MIDP (Mobile Information Device Profile) requirements. This classmark is tailored for mobile devices that have very limited resources available, but still are required to run JAVA executables. The most important generic MExE

requirements are CC/PP negotiation and security domains in MExE capable MS. The function of CC/PP negotiation is to optimise the content of the downloadable service for the MS. The CC/PP negotiation occurs after the service has been discovered but prior to downloading it to the MS. The CC/PP negotiation engine is located in the MExE server that is a node supporting MExE services in MSE. The downloaded service is placed into the relevant security domain in the MS, which defines the available functions for the service. The mission of MExE is to provide a secure means to dynamically download 3rd generation services into the MS, in a manner of the content of the requested service being tailored for the particular needs of the user and the used MS. The MExE service can be anything from a simple call forwarding service to new core software that reconfigures the MS. Once a user does not need the service anymore, it can be removed from the MS, which gives it, practically, unlimited memory resources.[3G01]

2 Technology overview of CC/PP negotiation

Regardless of the used MExE classmark, the CC/PP negotiation must occur prior to downloading of any service. Content negotiation represents the means by which the MExE MS and the MSE inform each other of the requested and available form of content. It is used to select the best representation of an entity when there are multiple representations available by an MSE. The entity, which can be a service, an image, an application etc, is located behind a URI. The best representation of an entity can be decided by an MExE Server or in some cases by the client application. If needed, the content negotiation may take place following a capability negotiation between an MSE and MExE MS. The methods for content negotiation are the basic HTTP/1.1 (Hyper Text Transfer Protocol) for classmarks 2 and 3, and WSP (Wireless Session Protocol) for classmark 1. Both the capability and the content negotiation have the same purpose: to optimise the content according to the client's capabilities. [3G00A]

Preference profile negotiation represents the MS's user profile transmission to an MSE. The user profile describes preferences requested by the user such as selected language, sound on/off etc. The connection protocols are the same as for the content negotiation. [Fr98]

The CC/PP negotiation method used by the MExE specification is made by W3C. The properties and the actual schema are based on the WAP UAProf group specification. The CC/PP framework is intended to provide an efficient mechanism for enabling enhanced content and service negotiation through a standardised format for user agent profiles. The use of RDF (Resource Description Framework) in CC/PP allows for interoperable encoding of the profile metadata in XML (Extensible Mark-up Language) and supports multiple vocabularies to provide for future extensibility. The WAP UAProf (User Agent Profile) specification is based on the CC/PP framework. The purpose of the UAProf is to specify an RDF based schema and vocabulary for CC/PP in the context of WAP UAProf,

and guidelines for schema extensibility to support a composite profile that enables future additions to the vocabulary and schema. [3G01]

2.1 Resource Description Framework

The RDF's mission is to support the interoperability of metadata. The RDF allows descriptions of different data resources, which are usually located on WWW (World Wide Web), to be made available in machine understandable form. It is based on a concrete formal model utilising directed graphs that conform to the semantics of resource description. The basic concept is that a resource is described through a collection of properties called a RDF description. Each of these properties has a property type and value. Any resource can be described within the RDF. The basic RDF data model is composed of three different object types, which are:

1. Resources

All things that are described by the RDF expression are called resources. A resource can be an HTML document, XML element, collection of Web pages, entire Web site or it might not be accessible via Internet at all. But resources always have an URI and they can have optional anchor ids. The extensibility of URI allows the introduction of identifiers for any entity imaginable.

2. Properties

A property is a particular aspect, characteristic, attribute, or relation, which is used to illustrate a resource. A property has always a particular meaning which defines its allowed values, the types of resources it can describe and its relationship with other properties.

3. Statements

A particular resource along with a named property and the value of that property for that resource compose an RDF statement. These three individual parts of a statement are called, respectively, the subject, the predicate, and the object. The object of a statement can be another resource or it can be a literal resource or a simple string or other primitive data type defined by XML [FF99].

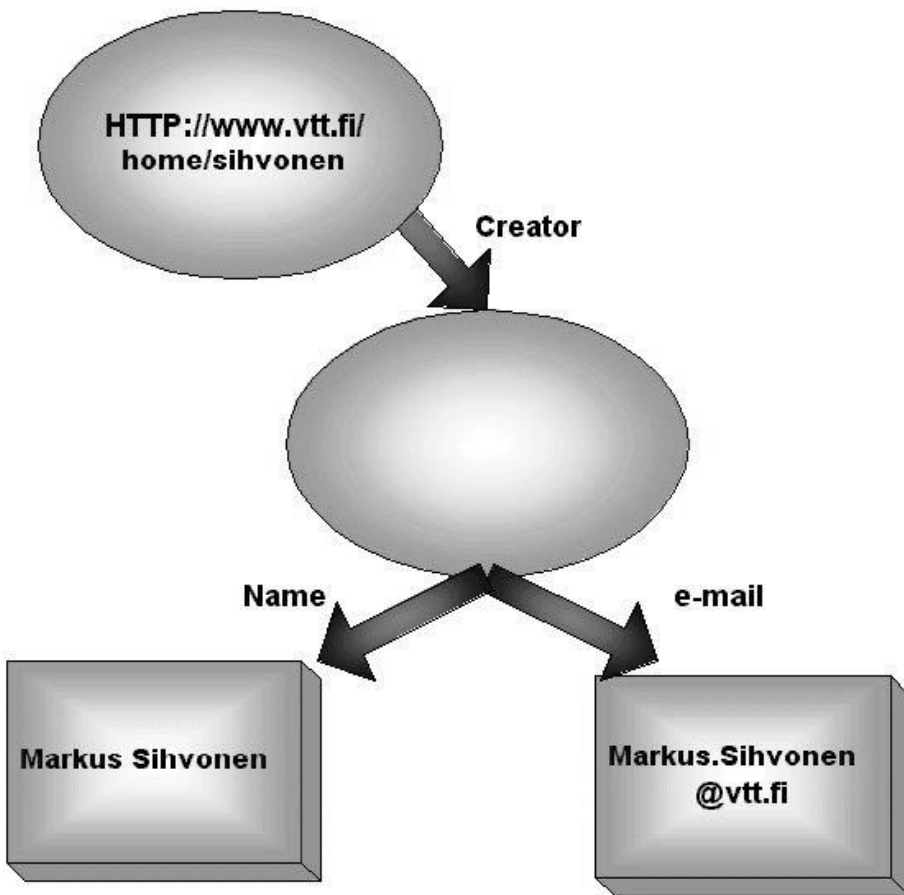


Figure 2.1 An RDF data model

The RDF data model is illustrated in Figure 2.1. The nodes in the figure are drawn as ovals and they represent resources. The named properties are presented as arrows. The nodes that are drawn in the shape of rectangles represent string literal. The direction of the arrows is important. Arrows always start from the subject, which is the resource, and point to the object of the statement. The diagram illustrated in Figure 2.1 can also be read as: “*The Person whose name is Markus Sihvonen, e-mail<markus.sihvonen@vtt.fi>, is the creator of <http://www.vtt.fi/home/sihvonen>*”. The intention of this sentence is to make the value of the creator property, <http://www.vtt.fi/home/sihvonen>, part of a structured entity. In the RDF, this type of entity is represented as another resource. In this example the structured entity does not have a name, which is illustrated as an empty oval in Figure 2.1.

The RDF extends the XML model and syntax for describing resources. It utilises the namespace facility of XML. The XML Namespace, which points to a URI, makes it possible for the RDF to uniquely identify a set of properties. The set of properties is

called a schema and it can be accessed by the URI, which is identified by the namespace. RDF also inherits all the XML syntactic flexibility, which includes white space rules, quoting using either single quote (') or double quote ("), character escaping, case sensitivity, and language tags that enable the support of multi-lingual metadata [BG99].

2.2 Extensible Markup Language

XML describes a class of data objects that are XML documents. It also partially describes the behaviour of a program that processes the XML documents. An XML document's construction is in conformance with the SGML (Standard Generalised Markup Language) document since XML is derived from the SGML, but it is more restricted than the SGML document.

Entities are building blocks of the XML documents. Information, the data, is stored in to the entities either in parsed or unparsed format. Actual characters construct a parsed data. Some of the characters are mark up data and the rest is character data. The purpose of the mark-up is to encode a description of the document's storage layout and logical structure. XML provides a mechanism to impose constraints on the storage layout and logical structure. A software module called an XML processor or XML parser is used to read XML documents and provide access to their content and structure.

A data object is a well-formed and valid XML document if it complies with the following set of constraints. An XML document must always have a logical and physical structure. The physical structure of a document is formed by entities. An entity may refer to other entities to cause their inclusion into the document. A document must always start from the root or document entity. The document is logically composed of declarations, elements, comments, character references, and processing instructions. All logical elements are indicated in the document by explicit mark-up. The logical and physical structures must nest properly, which means that no start-tag, end-tag, empty-element tag, element, comment, processing instruction, character reference, or entity reference can begin in one entity and end in another. [Ru99]

2.3 Transfer protocols

The Classmarks 2 and 3 use HTTP 1.1 to transfer CC/PP information to the MExE server. The Classmark 1, the WAP classmark, utilises WSP (Wireless Session Protocol) to transfer the CC/PP profile to the WAP gateway after which the CC/PP profile is converted from binary format to text format and HTTP 1.1 is used for the rest of the way to the MExE server. It is very likely that in future classmarks the used transfer protocol will be HTTP or SIP (Session Initiated Protocol) which is derived from the HTTP. The IETF has introduced Profile header, Profile diff-header and profile warning header for the SIP,

which enables the profile negotiation with the SIP. However, the current MExE specification does not support SIP for profile negotiation.

3 CC/PP schema with HTTP

There are three different HTTP headers for profile negotiation, which are the Profile header, the Profile-Diff header and the Profile warning header. The Profile header contains a list of references that address CC/PP descriptions. The Profile-Diff header contains the CC/PP description itself. Both the profile header field and the Profile-Diff header field are request header field types. The Profile-warning header, which is a response-header field, is the tool for delivering warning information [FF99].

The request-header fields in general allow a client device to transmit additional information about the particular request and information about the client device to the server. The request-header fields act as request modifiers and they have semantics similar to parameters on a programming language method invocation. The response-header allows the server to transmit additional information about the response that can not be placed in the status line. The response-header provides information about the server and future access to the resource identified by the Request-URI. [FB96]

3.1 The Profile header

The grammar for the profile header is illustrated in Table 3.1. The value of the profile header is a list of references (line 3). Each reference (line 3) in the Profile header represents the corresponding entity of the CC/PP description. The reference can either be an absolute URI or a profile-diff-name (line 3). An entity of a CC/PP description that is represented by an absolute URI exists outside the request, and an entity of a CC/PP description that is represented by a profile-diff-name is included into the request (line 3).

	Profile header field	Value
Line 1	Profile	Profile-field-name ":" 1#reference
Line 2	profile-field-name	"Profile"
Line 3	reference	<"> (absoluteURI profile-diff-name) <">
Line 4	profile-diff-name	Profile-diff-number "-" profile-diff-digest
Line 5	profile-diff-number	1#DIGIT
Line 6	profile-diff-digest	Sp; <MD5message digest encoded by base64>
Line 7	DIGIT	<any US-ASCII digit "0".."9">

Table 3.1 The profile header field's grammar

The entity that is addressed by an absolute URI, in the profile header (line 3), must exist on WWW. It is possible to use multiple absolute URIs, which enable the use of multiple sources for a CC/PP description. An absolute URI is distinguished from a profile-diff-name (line 4) by the presence of a colon (":") in the profile header field-value.

The profile-diff-name (line 3) in the Profile header addresses a CC/PP description in the corresponding Profile-Diff header within the same request. When the Profile header includes a profile-diff-name, the corresponding Profile-Diff header must be included within the same request. The Profile-Diff header will be discussed in the next chapter (3.2).

The purpose of profile-diff-name is to specify the priority of each CC/PP description in the Profile header. The priority is indicated by the order of references (line 3), which can be either absolute URIs or profile-diff-names. The latest reference in the Profile header has the highest priority. Therefore, according to the default rule, a CC/PP description that is represented by the latest reference can override CC/PP descriptions, which are represented by the previous references. The CC/PP schema incorporates an overriding rule for this basic function. When a CC/PP schema is applied, the CC/PP description, which is the latest reference, does not override the previous CC/PP description.

The profile-diff-name (line 4) is constructed from the two different partitions: the profile-diff-number partition (line 4) and the profile-diff-digest partition (line 4). The profile-diff-number indicates the corresponding Profile-Diff header. Multiple Profile-Diff headers can be in the same request and they are separated by the profile-diff-number (line 4). The profile-diff-number (line 5) is formed so that it corresponds to the suffix of the corresponding Profile-Diff header field-name (discussed in chapter 3.2) in the same request.

Using the MD5 message digest algorithm and the Base64 algorithm to the corresponding Profile-Diff header generates the profile-diff-digest value (line 6). The MD5 algorithm takes a message of arbitrary length as input and produces a 128-bit fingerprint of the input as output. Correspondingly, the Base64 algorithm takes arbitrary binary data as input and produces printable encoding data of the input as output. The reason for the existence of the profile-diff-digest (line 6) is the increased efficiency of the cache table to look up gateways, proxies and user agents. When the server uses some headers to select a representation that is subject to server-driven negotiation, these headers should be included in the Vary header field . [FF99]

3.2 The Profile-Diff header

The Profile-Diff header must be used within the Profile header in the same request. When absolute URIs in the Profile header represent all of the profile information, the Profile-Diff header is not added to the request. The other extreme alternative is that the Profile-Diff header contains all the profile information. The grammar for profile-diff header is illustrated in Table 3.2.

	Profile-Diff header field	Value
Line 1	Profile-Diff	Profile-diff-field-name ":" profile-desc
Line 2	profile-diff-field-name	"Profile-Diff-" profile-diff-number
Line 3	profile-desc	< the CC/PP description based on XML/RDF text format (any OCTET except CTLs, but including LWS)>

Table 3.2 The grammar for Profile-Diff header

In the Profile-Diff header, the profile-diff-field-name (line 1 and line 2) is composed of the prefix "Profile-Diff-" and the following profile-diff-number. The profile-diff-number (line 2) indicates the corresponding profile-diff-name (line 4 in the Profile header grammar; chapter 3.1) in the Profile header. One request can contain many Profile-Diff headers, and for this reason the profile-diff-number is introduced to indicate the corresponding profile-diff-name in the Profile header. The profile-diff-number corresponds to the prefix of the profile-diff-name in the Profile header within the same request (line 4 and line 5 in the Profile header grammar; chapter 3.1). The profile-diff-number should be increased by 1 when the Profile-Diff header is being added by a user agent, a gateway, or a proxy.

The profile-diff-field-name should not have zero as a first digit in the profile-diff-number, because it will cause unnecessary ambiguity in transfer sessions. For example, Profile-Diff-02 and Profile-Diff-002 are not allowed in the same request since they can be interpreted to be the same profile-diff-field-name. There can never be similar profile-diff-field-names or profile-diff-numbers included in one request.

The profile-desc field (line 3) contains the XML/RDF based CC/PP description that is negotiated to the MExE server. The field values can be folded onto multiple lines if the continuation of a line starts with a space or horizontal tab. All linear white spaces are treated similarly as a space. [FF99]

3.3 The Profile-warning header

The Profile-warning header's function is to carry warning information. When a client device transmits a request along with the Profile header to an MExE server, the server will request the CC/PP descriptions from the CC/PP repositories by using the absolute URIs

in the Profile header. If any one of the CC/PP repositories is not available, or the server is not able to obtain the most recent CC/PP descriptions, the MExE server should respond to the client with the Profile-warning header, the grammar of which is illustrated in Table 3.3.

	Profile-warning header field	Value
Line 1	profile-warning	profile-warning-field-name ":" 1#warning-value
Line 2	profile-warning-field-name	"Profile-Warning"
Line 3	Warning-value	warn-code SP warn-target SP warn-text [SP warn-date]
Line 4	warn-code	3DIGIT
Line 5	warn-target	(absoluteURI host [":" port])
Line 6	warn-text	quoted-string
Line 7	warn-date	<"> HTTP-date <">

Table 3.3 A grammar for the Profile-warning header

The warn-code (line 4) has three digits. When the first digit is 1, it indicates the status of the CC/PP description. If the first digit is 2, it indicates the type of the content adaptation applied to the message. The warn-target (line 5) indicates either the absolute URI or the host corresponding to the type of the warn-code. When the first digit is 1, it indicates the absolute URI and if the first digit is 2 it indicates the host. [FF99]

4 CC/PP negotiation strategy

The simplified CC/PP negotiation scenario is illustrated in Figure 4.1. The key components include an MS, which has one or more MExE classmark capabilities, an MExE Server that contains a negotiation engine and an Application Server that contains services. In addition, the MExE Server may access third party CC/PP description databases.

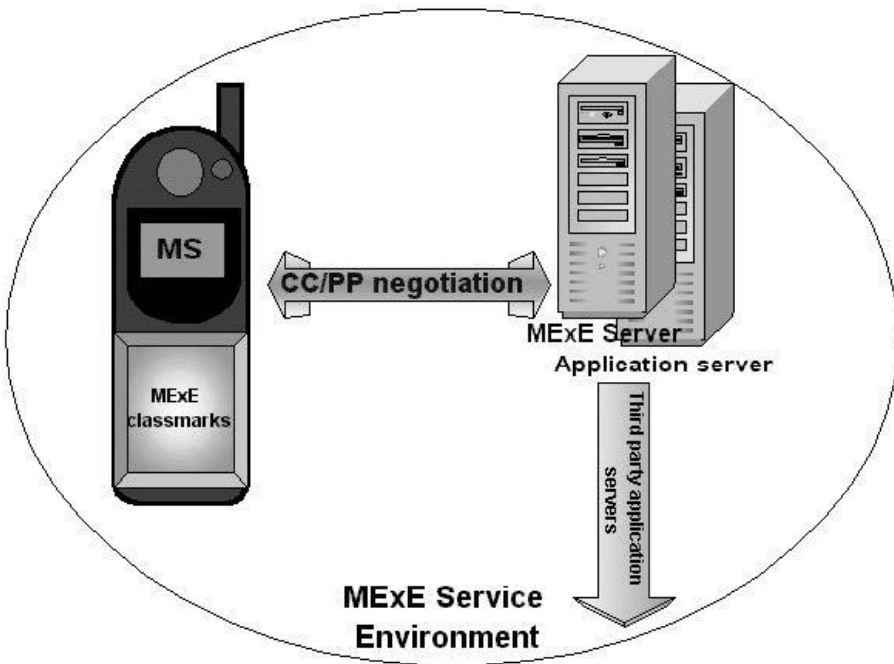


Figure 4.1 CC/PP negotiation scenario

The most important information provided to an MExE server by an MS, is its current CC/PP description. At the beginning of CC/PP negotiation session, a CC/PP negotiation engine, regardless of its location, must have access to the MS's current CC/PP description. [Fr98] Upgrades, such as increase in memory or a new operating system version, must replace the older CC/PP description information. An MExE server must also be sensitive to dynamic changes of an MS's current CC/PP description, for example the user must be able to turn the sound on and off "on the fly".

At the beginning of a CC/PP negotiation session, an MS's CC/PP description can be transmitted to the MExE Server in the following manners.

1. A full CC/PP description is transferred from an MS to an MExE Server.
2. All of the CC/PP description of an MS is transferred to an MExE Server from hardware and software manufacturers' databases, which are physically located outside MSE.
3. Any combination of parts 1 and 2.

When all of the CC/PP description is delivered directly by an MS, some simplifications can be made. If any of the default properties has been changed, there is no need to deliver old values but only the very latest CC/PP description values. [Fr98] This technique obviously diminishes the transmission load. But still, when an MS itself transmits all the profile information required in a session, a great load is imposed on a

network. This may cause a decrease in performance, particularly in relatively slow wireless networks.

Instead of delivering the complete CC/PP description directly from an MS to an MExE Server, a remote reference technique can be used. When the technique is utilised, an MS transmits a URI address of a third party to an MExE Server and the MExE Server retrieves the MS's CC/PP description from the location defined by the URI. This is particularly useful when non-changeable CC/PP default information, such as hardware and software default CC/PP description settings, is being transferred. A separate caching of CC/PP description into functional subsets is enabling this technique.

The remote reference approach leaves the problem of choosing between the default values and modified values for the application that controls the server logic in an MExE Server. The load on wireless networks decreases by using the technique since the MS transfers only the modified CC/PP description when particularly requested and only the third party URIs of the default CC/PP description during the initial session. An MExE server must request additional information from an MS or from a user if it has not received all the required data during the initial session. [Fr98]

A clear advantage of the remote reference technique is that it diminishes the load of a wireless network compared to the first negotiation alternative. A disadvantage is that the implementation is more complex than in the first negotiation model. There is no support for the third strategy, since it is impossible for a client device to know before a server's feedback what information is relevant. In this case the MS either transmits too little information, or irrelevant information. If too little information is being transmitted, the second transmission, where profile information is updated, must occur as in the second transmission strategy. Correspondingly, if irrelevant information is being transmitted, unnecessary network load is generated.

5 Conclusions and future work

The architectures of the different networks vary from one to another. The physical sizes of networks and their properties are different. Some networks are accessible only for a limited number of users or are physically located in a very limited area, which naturally regulates the usage of the network. Some networks are closed without a gate to other networks. The different properties and functionality of networks do affect the decision process of the CC/PP negotiation strategy.

Because of the great number of different network types, the strategy for transmitting the CC/PP description information of an MS should be flexible. In a situation where an MExE server can, via WWW, access third party repositories of default CC/PP descriptions of a given MS, the CC/PP strategy two should be applied – to transmit as little information as possible from an MS. During the initial contact the MS transfers the minimum required CC/PP description, which includes only the URIs of the third party servers. Later during

the connection, the MS must be able to transmit selected portions of its CC/PP descriptions at the request of an MExE server. When MSE does not have a gate to WWW, the MS must itself be able to transmit its full CC/PP description to an MExE server. According to the CC/PP description negotiation strategy, a hardware and a software manufacturer's default settings need to be stored in an MS and into a third party servers which have public Internet access. The servers must be dedicated to the storage of that particular device's default CC/PP description information.

The design of the negotiation engine for an MExE Server is of great interest for future work. An MExE server must be able to define whether a client device has the required resources to run the requested application. It is also the MExE Server's responsibility to know all the available applications for an MS.

The open questions of user profile will be considered in the near future. The vocabulary is yet to be standardised along with the user portability requirements of the 3rd generation networks.

Bibliography

- [3G00A] 3G TS 23.057 V3.3.0 (2000-09); 3rd Generation Partnership Project; Technical Specification Group Terminals; Mobile Station Application Execution Environment (MExE); Functional description; Stage 2 (Release 99)
- [3G01] 3G TS 23.057 V4.0.0 (2001-03) for information only, to be approved by 2001-03; 3rd Generation Partnership Project; Technical Specification Group Terminals; Mobile Station Application Execution Environment (MExE); Functional description; Stage 2 (Release 4)
- [FB96] Freed N. ; Borenstein N.: Mobile Information Device Profile, Java 2ME version 1.0, Multipurpose Internet Mail Extensions (MIME) Part One. Format of Internet Message Bodies. Network Working Group, November 1996.
- [Fr98] Franklin R.; Hjelm J.; Dawkins S.; Singhal S: Composite Capability/Preference Profiles (CC/PP): A user side framework for content negotiations; W3C Note 30 of November 1998
- [3G00C] 3GPP TS 22.121 V4.0.0 (2000-10); 3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Service aspects; The Virtual Home Environment (Release 4).
- [FF99] Franklin R.; Frystyk H. N.: staff members of W3C. CC/PP Exchange Protocol based on HTTP Extension Framework. W3C Note 24th of June 1999
- [BG99] Brickley D.; Guha R.V.: Resource Description Framework (RDF) Schema Specification. W3C Proposed Recommendation 03 March 1999.
- [Ru99] Rusty E. H. 1999. XML Bible. IDG Books Worldwide Inc. Foster City California. ISBN: 0-7645-3236-7