White Paper on Mobile Codes

Approved – 24 Oct 2008

Open Mobile Alliance

OMA-WP-MobileCodes-20081024-A

Contents

Figures

Tables

1. Scope

This white paper aims to stimulate a global market in which barcodes act as enablers for camera-equipped handsets to access content and services. Some technologies already exist; for example, in Japan, 2D barcode scanning is in widespread use. However there is fragmentation in the worldwide market currently, due to the variety of approaches to the questions of which barcode symbologies should be supported, what format of data they should contain, and how reader software should behave when barcodes are read. The Open Mobile Alliance aims to halt fragmentation by providing guidelines on existing standards as well as creating specifications to address interoperability needs as they arise. Once enough mobile code readers that follow those guidelines are deployed on consumer handsets, marketing organisations and publishers will be able to include mobile codes as links to online content and services with confidence, in advertising and promotional campaigns, and in printed and displayed media of many kinds.

This white paper does not address barcodes displayed on mobile devices as coupons or tickets (for reading by other devices).

2. References

[DATAMATRIX]	"Information technology — International symbology specification — Data Matrix", ISO/IEC 16022:2000.
[EAN/UPC]	"Information technology — Automatic identification and data capture techniques — Bar code symbology specification — EAN/UPC", ISO/IEC 15420.
[FLASHCODE]	"Flashcode Reader International Specification", Version 1.0 http://www.mobiletag.com/beta/en/contactspecification.html
[MIME]	"Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046 http://www.ietf.org/rfc/rfc2046.txt
[NDEF]	"NFC Data Exchange Format (NDEF) Technical Specification", NFC Forum http://www.nfc-forum.org/specs/
[NFCRTD]	"NFC Record Type Definition (RTD) Technical Specification", NFC Forum
[NTTDOCOMOGUIDE]	"Guidelines and criteria for creating QR codes compatible with all terminals", NTT DoCoMo, http://www.nttdocomo.co.jp/english/service/imode/make/content/barcode/about/#p02
[NTTDOCOMOFUNC]	"Bar Code Function", NTT DoCoMo, http://www.nttdocomo.co.jp/english/service/imode/make/content/barcode/function/
[OMAURI]	"URI Schemes for the Mobile Applications Environment", Version 1.0, Open Mobile Alliance™, OMA-TS-URI_Schemes-V1_0-20070718-D, URL:http://www.openmobilealliance.org/.
[QR]	"Information technology — Automatic identification and data capture techniques — QR Code 2005 bar code symbology specification", ISO/IEC 18004:2006.
[SPRTD]	"NFC Smart Poster RTD Technical Specification", NFC Forum
[TAGURI]	"RFC 4151. The 'tag' URI Scheme", IETF, http://www.faqs.org/rfcs/rfc4151.html.
[TEXTRTD]	"NFC Text RTD Technical Specification", NFC Forum
[URI]	"RFC 3986. Uniform Resource Identifier (URI): Generic Syntax", IETF, http://www.ietf.org/rfc/rfc3986.txt.
[URIRTD]	"NFC URI RTD Technical Specification", NFC Forum
[URNRES]	"RFC 2169 - A Trivial Convention for using HTTP in URN Resolution", IETF, http://www.faqs.org/rfcs/rfc2169.html.

3. Terminology and Conventions

3.1 Conventions

This is an informative document, which is not intended to provide testable requirements to implementations.

3.2 Definitions

Code	See Mobile Code
Direct Code	A mobile code that contains either (1) content for direct consumption for the handset, or (2) the address of the service to be accessed (typically a URI [URI])
Indirect Code	A mobile code that contains an identifier. The identifier has to be resolved (looked up) in order to access the identified content or service. See Resolution.
Mobile Code	A 1D or 2D barcode as read by camera-equipped handsets
Resolution	The process of mapping an identifier supplied from an indirect code into either content to be consumed directly by the handset, or the address of content (or a service) to be consumed by the handset. Typically, resolution is performed by a network service.
Symbology	The algorithm by which data is encoded as visual elements (typically arrangements of lines or squares), and the resultant "look and feel" for the user.

3.3 Abbreviations

EAN/UPC	Barcode symbology family including EAN-8, EAN-13, UPC-A, and UPC-E [EAN/UPC].
JAN	Japanese Article Number. A barcode of the EAN symbology, used in Japan.
NDEF	NFC Data Exchange Format
NFC	Near Field Communications
OMA	Open Mobile Alliance
QR	Quick Response, a type of barcode symbology [QR].
UPC	See EAN/UPC.
URI	Uniform Resource Identifier [URI]

4. Introduction

Mobile codes – 2D and 1D barcodes – have emerged as a promising enabler of the mobile Internet in some markets. Camera-equipped handsets now have good enough optics, image resolution and processing capacity to read mobile codes on printed materials and electronic displays. These symbols encode information such as URLs, phone numbers, and in-line content such as business cards.

There is, however, still a lack of interoperability between different markets and players. The majority of consumers are unlikely to adopt the technology before it comes pre-installed on their devices. Similarly, marketing, publishing and other industries that are otherwise motivated to provide mobile codes will not adopt them without adequate potential for consumer take-up. That in turn would entail deployment on a large variety of devices, and interoperability between different service providers.

.

Example			
Symbology	QR [QR]	Data Matrix [DATAMATRIX]	EAN-13 [EAN/UPC]
Data	http://www.openmobilealliance.org	http://www.openmobilealliance.org	5901234123457

Figure 1. Three common types of mobile code.

4.1 Usage scenarios for mobile codes

Figure 1shows examples of mobile codes in three common, standard symbologies: The QR and Data Matrix codes are 2D symbologies, and the EAN-13 code is a 1D symbology. Each symbology is an algorithm for encoding data as visual elements, typically but not necessarily squares and lines. Each has a resultant "look and feel" for the user.

When read and decoded by a camera-equipped handset, data from mobile codes can be used to access content and services in several ways. These include but are not restricted to:

- Initiating a Web download
- Sending an SMS or email message
- Dialling a number
- Consuming data such as business card information directly from the code.

We now give four corresponding scenarios of use.

4.1.1 Web

Outside, in a café, a mobile handset camera is pointed at an advertisement, poster, leaflet or beer-mat. In just one click, the user arrives at a webpage designed specifically for that location. No struggle with the navigational systems of mobile websites; no wait – just the instant fulfilment of the user's needs. The spontaneity of the response encourages an internet connection there and then; the internet content is relevant to the precise time and location of the user; the advertiser can track exactly which piece of paper generated the user response – and the mobile handset has enabled a trouble-free and relevant experience of the web that is potentially more useful to website provider and user alike. And of course, the mobile industry benefits from increased usage of the internet over mobile handsets.

4.1.2 SMS

At the breakfast table, Mary finds day 2 of a holiday promotion in her copy of the Times newspaper. To enter, she can either text 'Holiday 2' to '55555', or she can read the mobile code. The latter is much quicker and easier. The response explains that she needs to do this over three days to complete her entry, and that she can enter for yesterday from the Times web pages. That lunchtime, she finds yesterday's entry on the web, and reads the code from the screen. The next day, at 3pm, she still hasn't bought a copy of the Times. She receives an SMS reminding her that she needs just one more entry to complete her entry. She buys the Times and reads the code when she gets home. Mary has been provided with easy ways to enter the promotion – which might have been too much trouble, otherwise. The newspaper established two-way connections with its readers. If they consent, there are further ways in which it can reach them.

4.1.3 Dial

Jane is going though the yellow pages for a plumber. It's an emergency. She goes for the first ad with a mobile code, points and clicks to dial the number – there's no time to waste. The plumbing company has found they're receiving more calls since they placed this larger ad with the mobile code. The yellow pages company is more than happy to accommodate them.

4.1.4 Business card

Sarah is closing a face to face conversation with a potential client. She takes the client's card and reads the mobile code on it with her phone, to store the contact details. There is no need for error-prone transcription. The contact details are on her mobile phone, exactly where she needs them to be.

4.2 Benefits for consumers, marketing, mobile industries

The above scenarios illustrate the advantages of mobile codes for consumers, marketing and the mobile industries.

4.2.1 Consumers

Reading a mobile code is faster and more convenient than transcribing data such as URLs, telephone numbers and SMS messages into handsets. It's less error-prone, because the error correction in the codes guarantees against entering incorrect data.

Consumers do not need to navigate to the appropriate application on their handset. It is implicit in the type of data in the code. This is especially helpful with respect to web browsers, which many users find difficult to locate and operate on their handsets.

4.2.2 Marketing

By lowering the effort required for a user to participate in promotional campaigns compared to manual transcription of URLs etc., mobile codes increase response rates.

Mobile codes also make print advertising measurable. By including unique identifiers in the mobile codes targeted to different placements of the same advertisement in media formats, markets and physical locations, marketers can measure which placements were more effective than others.

4.2.3 Mobile industries

Wireless Operators: Mobile data is still too hard, despite everyone's best efforts. SMS doesn't scale in usability beyond a few premium applications such as ringtone downloads. With mobile codes, mobile internet growth will accelerate as more connections are made direct from advertisement to web. By adding value to the consumer the industry can increase ARPUs through more and more rewarding internet connections. Operators can help drive the reach and scale of the mobile code scanning opportunities by ensuring that scanning clients and applications are pre-installed in the mobile devices. Moreover, while user triggering of mobile access to content and services can be reduced to “scan the code with the camera-and click”, operators may – where involved in the transaction – be able to associate such user inquiries with the user’s preferences based on his/her subscription profile so that advertisers may benefit from target marketing with deeper profile data of prospective customers. Privacy concerns need to be considered as applicable.

Handset Manufacturers: Feedback from mobile code adoption in Japan suggests that, once established in the market, mobile code readers become a “must have”. Handset companies will want to provide a compelling reason to upgrade to their mobile code-enabled terminals.

Application and Service Providers: There is room for innovation in many aspects of user interaction with mobile codes, and in new mobile code support services. The mobile phone’s code reader is the new web browser in terms of the impact it could have on mobile content and service provision. The service at the end of the code provides the new internet experience. Both application and service sectors have the opportunity for significant growth, perhaps enabling different actors in the value chain, as is discussed in Appendix A.

4.3 The need to select from existing standards

This white paper discusses existing standards and de facto standards that apply to key aspects of reading mobile codes.

By using existing standards wherever possible and appropriate – as applied to the usage scenarios in this white paper– work will be saved and the path to adoption will be accelerated. One of the roles of the OMA is to critically examine the suitability of existing standards, and to determine whether there is a gap between what they specify and the functionality that is required for mobile codes.

5. Symbologies

This section describes standard symbologies that are already in widespread use in business processes such as product identification where dedicated reader hardware and software is used; and, to a greater or lesser extent, as mobile codes, that is, read by readers on handsets. They are described on the basis that they are fully specified and approved standards, and that reader technologies for handsets are already mature, with several providers in the marketplace.

In addition to open standard symbologies, support of proprietary symbologies may be important in certain markets. Examples of such proprietary symbologies include EZCode, Colorzip, BeeTagg and Shotcode. This list is not exhaustive.

5.1 Requirements and basis for comparison

Mobile codes should satisfy certain basic requirements. It is desirable that they appear in print and on electronic displays in the following ways. This is not an exhaustive list.

1. provide rich enough functionality to justify their inclusion;
2. be robustly readable by a critical mass of camera-equipped handsets;
3. have acceptable impact on layouts and aesthetics.

When comparing symbologies with respect to those requirements, the following factors are relevant:

Data capacity. Some symbologies, such as the EAN/UPC family [EAN/UPC], have a data capacity in the order of ten bytes. Such symbologies cannot be used to store in-line content or addresses such as URLs. Instead, their use is restricted to storing identifiers, which need to be looked up to provide associated content or services. Other symbologies, such as QR [QR] and Data Matrix [DATAMATRIX], can in principle be used to store thousands of bytes, although in practice such quantities of data cannot be resolved currently, given the limited imaging capacity of handsets. A typical limit for high-end camera phones at the time of writing is a few hundred bytes of alphanumeric or binary data.

Error correction. Some degree of error detection and correction is essential, given that users may try to read partially occluded codes, codes displayed on imperfect surfaces, and codes read in imperfect lighting conditions.

Space efficiency. Symbologies define a smallest visual unit, sometimes called a module, out of which the variable part of a mobile code is constructed as an encoding of the data to be encoded along with error correction bits. Symbologies also include visual elements (constructed from modules) for the purposes of registration in the visual field, and supplying certain metadata needed for decoding. Those include a 'quiet zone' at the borders of the code, and certain fixed and variable elements within it. One can compare symbologies by asking, for a given module dimension (chosen to be resolvable by a typical handset) and a given amount of data of a given type to be encoded (e.g. numeric vs. alphanumeric vs. binary), and a given level of error correction, what are the overall dimensions of the corresponding mobile code? Obviously, the better the space efficiency in both axes of the display plane, the better.

Look and feel. Any barcode symbology, by its nature, consists of a regular arrangement of visually noisy elements. But the overall appearance of different symbologies to the human eye isn't necessarily equal.

5.2 Standard symbologies

This section will discuss QR codes [QR] and Data Matrix [DATAMATRIX], which are currently deployed standardised symbologies in the mobile space. Other standardised symbologies exist, e.g. Aztec; and more symbologies might be standardised in the future.

5.2.1 QR

QR codes [QR] (Figure 2) are already widely used in Japan in industrial processes and as mobile codes, and have been used in mobile code pilots in the rest of the world. They are an ISO standard. The variant of QR codes in widespread use are Model 2 (with some minor enhancements in the QR 2005 variant).

(a) (b) (c)

Figure 2. QR codes encoding alphanumeric data "a123456789a12…" of length (a) 40, (b) 100, (c) 200 characters

QR codes have native support for the Kanji character set, and efficient support for all-alphanumeric data. QR codes currently support alphanumeric data capacities of up to about 400 bytes and binary capacities of up to about 270 bytes, in formats that are readable by handsets that have macro focus and resolutions of several megapixels. Such handsets are typical in Japan but represent high-end handsets outside Japan. The capacity is closer to 100 bytes of alphanumeric characters or 70 bytes of binary data when read by a typical camera phone without macro focus, but with resolution of one megapixel or more.

The QR symbology supports error correction using the Reed-Solomon method. There are four levels of error correction, selectable according to the operating environment. The capability ranges from correcting about 7% of the data in a code, to about 30%.

A quiet zone of at least four modules width is required around QR codes.

5.2.2 Data Matrix

Data Matrix [DATAMATRIX] (Figure 3) is an ISO standard. We restrict our discussion here to the ECC200 variant of Data Matrix codes, which are widely used in industrial processes and have already been used as mobile codes in several pilots.

(a) (b) (c)

Figure 3. Data Matrix codes encoding alphanumeric data "a123456789a12…" of length (a) 40, (b) 100, (c) 200 characters.

Data Matrix, like QR codes, uses the Reed-Solomon algorithm for error correction. Data Matrix codes have a redundancy of approximately 33%, which ensures a good trade-off between code efficiency and correction capacity.

A quiet zone of at least one module width is generally needed around the 2D barcodes, as shown by the following examples:

5.2.3 EAN/UPC

The EAN/UPC family of 1D symbologies [EAN/UPC] (Figure 4) consists of EAN-13, EAN-8, UPC-A and UPC-E. They are all ISO standards. They are in widespread use as product tags.

Figure 4. An EAN-13 code.

The data capacity of these codes is small and fixed according to the symbology: between 8 and 13 numeric digits. They carry one error correction digit. The space efficiency of these codes is relatively poor.

Aesthetically, EAN/UPC codes are a "given" on the vast majority of products.

5.3 Printing and display considerations

Irrespective of the symbology used, three further factors need to be taken into account when printing mobile codes or presenting them on an electronic display:

Size. A given code may be printed or displayed relatively large or small, by varying the module size. However, relatively large codes may be aesthetically disruptive. Relatively small codes may be unreadable by many handsets. A relatively small or large code may also cause the user to adopt a reading position that is respectively too close to or too far from the display surface.

Rendering fidelity. Mobile codes must be printed with sufficient accuracy, use colours of sufficient contrast, and have a 'quiet area' of suitable dimensions around them, if they are to be robustly readable. Similar considerations apply when presenting codes on electronic displays. In addition, screen flicker (caused by the lack of synchronisation between the display's refresh pulse and the handset's camera) may interfere with code reading.

Colour. Colour is relevant to aesthetics, and not just readability. With Data Matrix, the possibility to use white on black provides a convenient way of integrating a code on a layout with black background.

The standards' specifications [QR, DATAMATRIX, EAN] provide some general guidelines on the above issues, and NTT DoCoMo also publish guidelines [NTTDOCOMOGUIDE].

6. Data Format

This section first describes the data required in mobile codes, and then describes existing standards that specify a syntax for representing it. Both the choice of data structure and the way the data is processed are independent of the symbology choice, except in respect of limited data capacity.

6.1 Data requirements

6.1.1 Direct vs. Indirect codes

There is a basic distinction between 'Direct' and 'Indirect' mobile codes, which have differing requirements.

- **Direct**: the code contains either the address (URI) of the content or service, or the content itself, in-line. The requirement here is to support a range of URI and content types that correspond to scenarios such as those described in Section 4.1.

- **Indirect**: the mobile code contains an identifier, which needs to be resolved to obtain the content or service. Resolving an identifier means looking it up, typically at a network service, to determine the corresponding content or service. To avoid ambiguity, any identifier retrieved from a code must be unique. Several URI schemes have already dealt with those issues, e.g. 'tag' URIs [TAGURI]. Note that certain symbologies, such as those in the EAN/UPC family, contain uniquely allocated identifiers. The combination of the symbology name (e.g. its ISO identifier) and the identifier read from the code is therefore unique.

6.1.2 User Feedback

Experiments with the public and reactions from marketing support the provision of feedback when a mobile code is read that is specific to the mobile code itself. This may be simply text to pop up when the user reads the code, or it may, for example, be a short menu of options. Other sorts of immediate user feedback could also be envisioned. This provides good responsiveness to the user. It also provides publishers and marketers with a total of three levels of interaction with the user: in the content in which the code is embedded, in the code's specific feedback, and on the network.

6.2 Syntax

Below we consider 3 examples of syntax for mobile codes. Other specifications exist, and the list of examples below is non-exhaustive.

6.2.1 NDEF

This subsection describes the applicability of NFC Forum's NDEF standard as an appropriate syntax for data in mobile codes. It accommodates both Direct and Indirect use cases. It can also accommodate proprietary applications and foreseeable extensions to standards (for example, for Bluetooth pairing).

Near Field Communications (NFC) is a short range radio technology based on radio frequency identification (RFID). It can be used for a new kind of human-device interaction, where a user can initiate actions on a mobile device by touching physical objects.

NFC use cases can be divided into 3 sub categories, based on the role the device plays. In some of the most popular use cases of NFC, the device is acting as an RFID tag, being read by an external reader. Examples include using the phone as a mass transit ticket, a credit card, or a key. Another mode of NFC phones is peer-to-peer, where two users can exchange data such as a business card by touching the devices together. The last mode of an NFC phone is when the device acts as the RFID reader (/writer), acting on an external, and usually passive, RFID tag. This last mode is most relevant to the discussion that follows, as these use cases much resemble the typical use of mobile codes.

The NFC Data Exchange Format (NDEF) is a data format agreed by the NFC Forum for achieving interoperability between different players in the mobile industry. Version 1.0 of the NDEF specification was published by the NFC Forum on the 7th of March 2006 [NDEF]. The format wraps content transferred over NFC, adding information about the type of the transferred data (helping the device decide what to do with the data) and the length of the data (useful for separating multiple records, as explained in the UI section below).

The NDEF format was designed with similar aims and constraints that apply to mobile codes. The format provides a flexible way of encoding both common industry standard data formats – that can support the most common use cases – as well as being open for company-specific proprietary data. As the first commercially available NFC tags vary in size from about 50 bytes up to some kilobytes – similar to the data capacity achievable with mobile codes – the NDEF format has been designed for low overhead.

On top of NDEF, the NFC Forum has published a number of Record Type Definitions [NFCRTD], which provide a well known way of encoding data for some of the most common use cases. As of November 2007, the NFC Forum has published RTDs for plain text [TEXTRTD], URIs [URIRTD] and Smart Posters [SPRTD].

Reusing the established NDEF format for mobile codes should help reduce the time to market, Moreover, devices that implement both NFC and mobile code technologies may be able to benefit by re-using the NDEF format for mobile codes

6.2.1.1 URI record

The NFC Forum URI Record Type Definition 1.0 from the 7th of March 2006 [URIRTD] defines an efficient way of encoding URIs (both URLs and URNs) into NDEF records. To save space on the tag, this specification abbreviates the URI protocol field into the first byte of the record data. As an example, "http://www." is abbreviated into the digit '1', while "tel:" is abbreviated as the digit '5'. The specification lists 35 such abbreviations, in addition to which it allows other URI types written in their unabbreviated form.

The URI RTD does not specify how a device should behave when encountering a specific URI. For guidelines on that, please refer to the OMA URI schemes document [OMAURI].

6.2.1.2 Support for inline data

In addition to the NFC Forum RTDs, the NDEF specification can contain content of types identified by RFC 2046 (MIME Media Types) [MIME]. This provides a means of encoding business cards, small images, etc.

6.2.1.3 Smart Poster records

The NFC Forum Smart Poster RTD 1.0 from the 24th of July 2006 [SPRTD] provides a means of encoding a URI together with a popup text to guide the user. To do this, the smart poster contains a URI record, and a simple text record [TEXTRTD], after each other. The popup text can be provided in multiple languages, from which the mobile device chooses the one most appropriate for the user. Optionally, the smart poster can also contain other records, including a title, an image or an action to take – such as bookmarking a URL.

Figure 5 shows SmartPosters with links to www.openmobilealliance.org with the popup text "Hi!" encoded as Data Matrix and QR code.

Figure 5. Link to www.openmobilealliance.org encoded as a Smart Poster record (and as Data Matrix and QR)

6.2.2 NTT DoCoMo

NTT DoCoMo's standard for mobile codes is by far the most widespread of the mobile code deployments, as of November 2007.

The NTT DoCoMo standards are based on the QR code symbology, but the data structures described below could equally well be written on Data Matrix. As of November 2007, these specifications are available on NTT DoMoCo's web pages [NTTDOCOMOFUNC].

6.2.2.1 Marked-up data

In these specifications, there are a number of data structures specified for writing different data types into a mobile code.

- A mobile code starting with MEBKM: is a command to the phone to bookmark a URL. A title for the bookmark is provided in the data, together with the URL. This is similar to an NDEF Smart Poster with the bookmark action.
- A mobile code starting with MECARD: signifies a business card. This is similar to an NDEF record with a vCard as MIME typed data.
- A mobile code starting with MATMSG: is used for encoding email messages onto a mobile code.
- A mobile code starting with CNTS: encodes images, music or ToruCa messages. This is similar to having the data as MIME typed content on an NDEF message [MIME].
- A mobile code starting with LAPL: indicates data used for i-appli (DoCoMo's Java services) synchronization.

6.2.2.2 Plain data

A widespread current use of mobile codes, in Japan as well as elsewhere, is to encode weblinks, phone numbers and email addresses as plain text into a mobile code, and heuristically try to detect these. This detection follows the same pattern as is common in messaging systems on mobile phones and computers, where the device will let the user click on anything that looks like a weblink.

On encountering something that looks like a phone number, the device will offer the user the option of initiating a voice or video phone call, or sending an SMS message to that number. If the device finds a message containing an email address in the form *somebody@example.com*, it can suggest that the user sends an email to the address. Lastly, if the device finds a URI with the prefix *http* or *https*, the device can invoke the browser to access the resource designated by the URI. In all the foregoing cases, the device first obtains user confirmation before the operation can proceed.

6.2.3 Flashcode data syntax

This data syntax is used by some companies belonging to the i-mode alliance. It is specified in the Flashcode Reader International Specification [FLASHCODE]. The syntax is already commercially used in Spain, and a commercial launch is planned for France in early 2008, providing interoperability between French operators. There have been trials in other European countries.

6.2.3.1 Use cases

Both direct and indirect modes are addressed through this data structure which covers a large number of use cases including, but not limited to, web redirection and download, call initiation, SMS/MMS sending and business card information retrieval.

6.2.3.2 Overview of the data structure

The data extracted from the mobile code is a concatenation of 2 fields:

- "Service Type" field
  - this is a 2-digit field indicating the type of service provided (for instance direct call initiation or indirect web redirection)
  - 100 service types may therefore be achieved and this field currently allows a very high number of additional use cases to be added

- "Details" field
  - The format of the "Details" is dependent on whether the service type is direct or indirect:

For direct services, the "Details" field contains formatted data which is specific to the service type and consists of list of fields separated by a delimiter. Here is an example of a direct phone call initiation in the Details field:

- Formatted-Data = NUMBER "|" TITLE

For indirect services, the "Details" field is the concatenation of an optional action and an identifier. The "Action" part is a single digit specifying the way the client application should interact with the content linked to the tag. If not specified in the tag, the barcode reader application shall apply a default value which depends on the "Service Type". The "ID" part is a 13-digit number containing the index that will be resolved by the redirection server to reach the end service.

7. Client behaviours

Now that mobile codes themselves have been dealt with, we describe factors concerning the behaviour of the 'code reader' client on the handset (downloaded or preinstalled).

7.1 Support for Direct and Indirect architectures

One of the most important factors affecting the way in which the code reader behaves with respect to a given mobile code is whether the code is Direct or Indirect (see Section 6.1.1). This section describes in more detail the steps taken according to the method used, and lists some of their advantages and disadvantages.

7.1.1 Direct method

The Direct method was introduced in Section 6.1.1 above. Figure 6describes the logical process of the Direct method, including related steps.

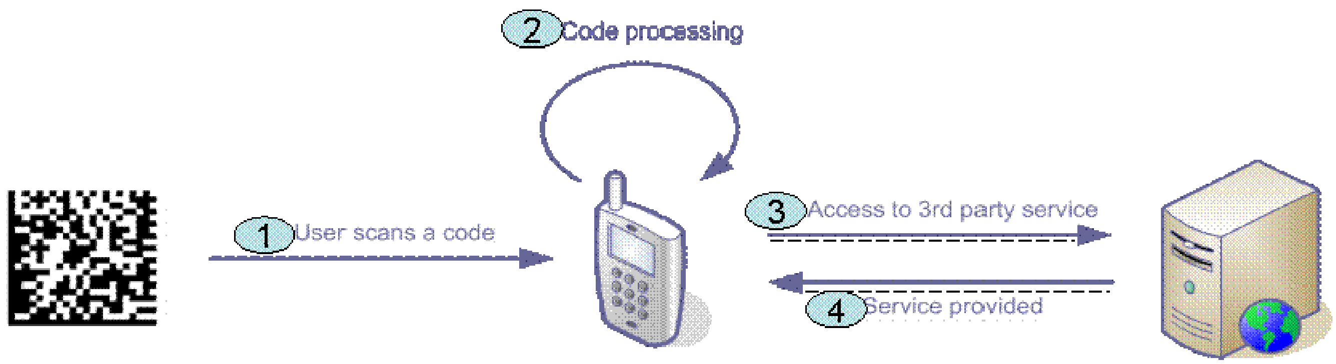

Figure 6. Direct method

Step 1: a software client on a mobile handset acquires, recognises and decodes a mobile code.

Step 2: the decoding is completely performed by the client on the mobile handset. In some cases, the decoded data is content that is provided for direct consumption on the handset, such as personal contact details.

Step 3 (optional): the result of the decoding (as performed in step 2) is a URI to a 3rd party service.

Step 4 (optional): if step 3 occurs, the 3rd party provides the required services.

Note that clients reading a URI from a mobile code in Step 3 will process it in accordance with [OMAURI].

7.1.2 Indirect method

The Indirect method was introduced in Section 6.1.1 above. Figure 7describes the logical process of the Indirect method, including related steps.

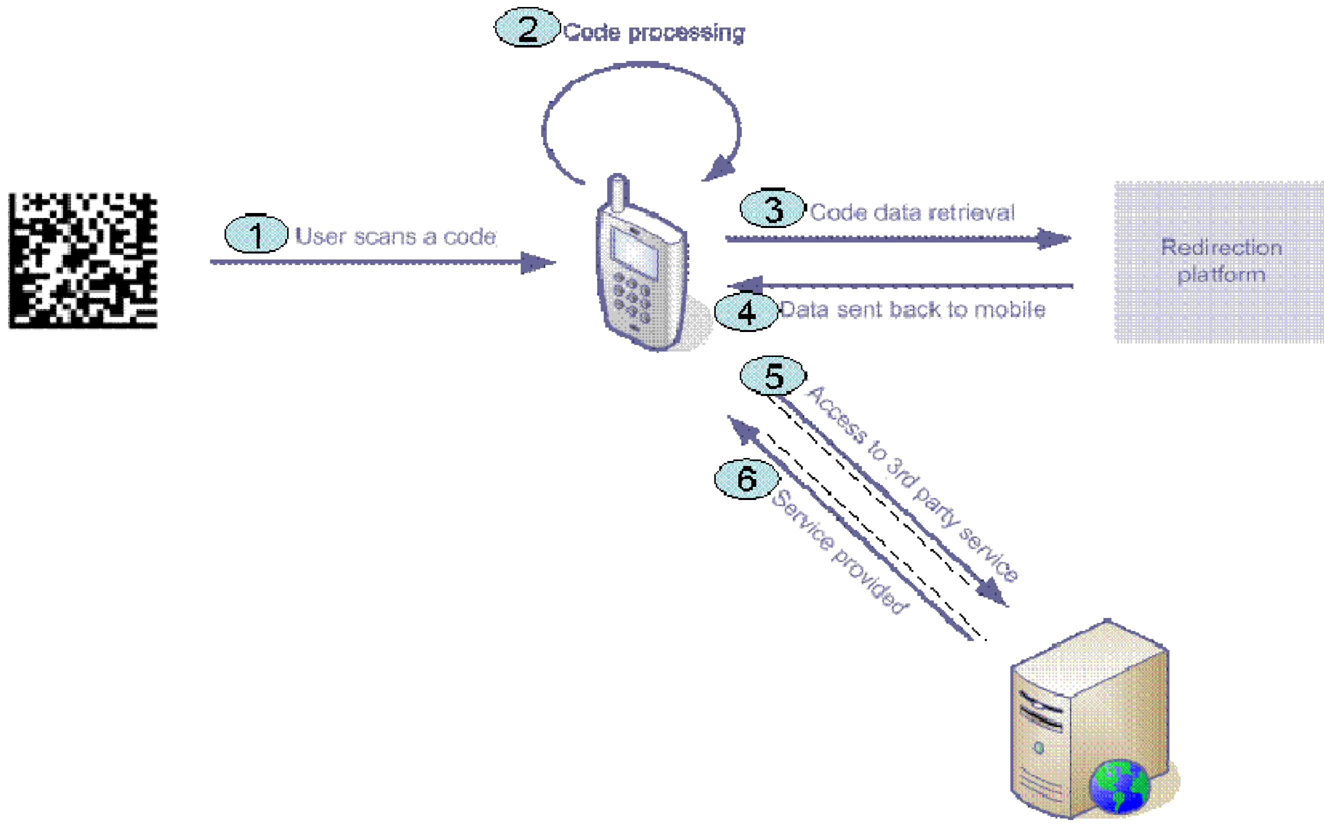

Figure 7. The Indirect method.

Step 1: a software client on a mobile handset acquires, recognises and decodes a mobile code.

Step 2: the processing of the data in the mobile code is not fully performed by the client on the mobile handset; the client reads the identifier embedded in the mobile code and then it connects to a redirection platform.

Step 3: the redirection platform resolves the identifier from the mobile code. The redirection platform may apply rules governing both access restrictions and the type of service provided to mobile users.

Step 4: the redirection platform sends either content or a service URI to the mobile handset.

Step 5 (optional): the data returned to the client in step 3 is a URI to a 3rd party service.

Step 6 (optional): if step 5 occurs, the 3rd party will provide the required services.

Note that clients receiving a URI from a redirection platform in Step 5 will process it in accordance with [OMAURI].

7.1.3 Achieving interoperability with the Indirect method

For commercial applications designed to reach a wide consumer base, a key requirement for the Indirect infrastructure is interoperability: the guarantee that a given mobile code will lead to consistent service provision across makes and models of handsets, across operators, and across redirection platforms. We now discuss some of the challenges involved in creating an interoperable Indirect infrastructure. This discussion is intended only to introduce some of the issues and to point out some existing standards that may prove relevant to the solution.

There are two main problems to solve in order to achieve interoperability: managing the identifier namespace and resolving the identifiers.

1. Managing the identifier namespace

The identifier obtained from an indirect code should be unique; otherwise, the effect of reading a code may be ambiguous. We first describe two examples for decentralised allocation of identifiers, and then consider centralised allocation.

Example U1: There is an existing standard for the decentralised creation of globally unique identifiers: 'tag' URIs [TAGURI], e.g.:

tag:example.com,2007:832481364312894

Users may place such URIs in NDEF URI records or in the free text of an NTT DoCoMo code.

Example U2: In the special case of EAN/UPC codes, where the symbology is strongly tied to a managed namespace of identifiers, the identifier extracted from the code is prefixed by a symbology-specific unique string, such as:

tag:openmobilealliance.org,2008:bt:mc:sym:<symbology identifier>:

In this example, the client is pre-configured with the unique names of the UPC/EAN symbologies. It automatically constructs the resulting globally unique identifier for the mobile code, by appending the identifier it has read from the code to the symbology's identifier.

An alternative to decentralised namespace management (as in examples U1 and U2) is to create a centralised authority for managing the allocation of identifiers. In principle, a centralised authority could allocate shorter identifiers, and thus facilitate smaller mobile codes. Moreover, the scope of uniqueness maintained by such an authority could be, say, national rather than global, enabling the identifiers to be shorter still while achieving geographic market interoperability. But then a factor to be considered is that conflicts might arise for those travelling into or out of the country or region concerned.

2. Resolving the identifier

A system architecture is required for resolving the identifiers extracted from mobile codes into data that the client can consume directly or use to access third-party services, such as a URI. We consider two examples that differ markedly. These are merely taken as the two examples out of a multitude of possibilities.

Example R1: Centralised resolution. This example follows the architecture shown in Figure 7

In this example, every handset is configured with the domain name of a centralised resolution (redirection) service. To access this service, the client may construct a query URL according to the standard for URN resolution [URNRES], using the 'N2L' example. That service then performs the redirection.

Example R2: Code reader applications. In this example, consumers are free to download and install applications that utilise their code readers by processing the identifiers read from codes, and which respond to identifiers beginning with specific tag prefixes. For example, an application for product information could take all URIs beginning *tag:openmobilealliance.org,2008:bt:mc:sym:EAN-13:* read from the code reader, and send them to an application-specific service for resolution. In the case of an enterprise application used by staff in the Example corporation, their clients incorporate an application that sends all tags beginning *tag:example.com* for in-house resolution.

7.1.4 Security

Mobile codes raise security issues, since they may lead the user to a malicious service or to one with an undesirable side-effect such as a hidden charge. Such codes may be presented in apparently benign situations, and they may even be placed so as to cover *bona fide* codes.

In the case of indirect codes, the redirection platform can place constraints on the possible results when a given user reads a mobile code. Firstly, only codes containing an identifier sanctioned via that platform will produce any result at all. Secondly, trusted information about the charge that will be billed to the user can be generated and sent to the user before completion of the transaction. Thirdly, if the user is registered with the platform, then the platform can apply policies concerning the types of content that the user can access – based, for example, on the user's age.

Whether the mobile code is Direct or Indirect, in some markets, user feedback is an important issue in helping the user avoid inadvertent service or network accesses. For example, an implementation may display the details of the third-party service address (for example, its URI) and require confirmation from the user before accessing the service. For general consumer use, in some markets, this is considered to be good practice.

8. Examples of direct mode encoding based on existing standards

This section collects the preceding discussion to consider examples, based on existing standards.

Table 1 shows that two data formats, NDEF and NTT DoCoMo, may be used as independent alternatives, and that each format may be encoded in either the QR or Data Matrix symbology.

About five years of experience exists with the NTT DoCoMo standard. The NDEF standard has not been applied to mobile codes hitherto, but there is no reason in principle why it should not be used for this purpose.

Even though the NTT DoCoMo format is normally associated exclusively with QR codes, there is no technical reason why NTT DoCoMo data should not be represented in Data Matrix encoding. This gives a total of four combinations for implementing mobile codes based on well established standards.

Symbology: / Data format:	QR, Data Matrix
NDEF	Smart Poster, URI, Text, MIME-typed inline data
NTT DoCoMo	MECARD, MATMSG, MEBKM, LAPL, CNTS, Text including URIs[1], phone numbers, email addresses

Table 1. Four combinations of data format and symbology, and the types of data encoded

Representation of Direct mobile codes is immediate for any of the four combinations in Table 1. The NDEF standard allows arbitrary URIs and MIME-typed inline data. The NTT DoCoMo standard includes *http(s)*, *tel*, *sms* and *mailto* URIs, and inline data.

The OMA has defined terminal behaviours for various types of URI in [OMAURI], and this would apply to code reader behaviour on reading these types of URI. All such URIs can already be represented in NDEF. In principle, the NTT DoCoMo standard could be naturally extended so that any types of URI in [OMAREF] could appear in free text.

[1] Includes a proprietary URI scheme "tel-av:" to initiate video phone calls.

9. Conclusions

Barcodes have been discussed as enablers for camera-equipped mobile devices to access content and services. In this context, the barcodes are called mobile codes and include (ordinary) 1D barcodes (e.g. EAN/UPC) as well as 2D matrix code type symbols (e.g. QR Code or Data Matrix).

The existing landscape around mobile codes has been described with its various aspects, including usage scenarios, symbologies to produce mobile code symbols, mobile code data formats, client behaviours when a mobile device encounters a mobile code, and options for the system architecture. Two classes of architectural options have been identified, namely Direct and Indirect architectures. It has been identified that, in some markets, support of proprietary codes may be desirable.

Assessment of the existing landscape has led to insights about the areas where further agreement needs to be made to achieve interoperability between all essential parts of a mobile codes infrastructure, including e.g. code symbol generation, client code reading software, and the mobile network through which services will eventually be provided. Also some standards from other areas have been identified and discussed which may prove useful as candidates for data formats for both the Direct and the Indirect architecture, e.g. NDEF, URI, and tag URIs.

Interoperability is achievable within the Direct architecture, where mainly the choice of symbology and data structure need to be agreed. Proven *de facto* standards and practices exist for both symbologies and data structures. The NTT DoCoMo system in Japan is a prominent example.

Interoperability for the Indirect architecture may require some additional agreement, for example with respect to identifier namespace management and resolution procedures. Standardised interfaces between the functional roles in the Indirect architecture need further consideration.

It is recommended that the normative specifications to be developed ensure backward compatibility with relevant 2D barcode systems, so that mobile devices that are compliant to the specifications to be developed shall be able to recognise and process such existing 2D barcodes.

Based on the analysis in the white paper, the normative specifications should address the following:

- Common components for both Direct and Indirect methods, which consist of:
  - Barcode symbology
    - Support of Data Matrix and QR
    - Optional support of proprietary codes
  - Data structure in the mobile codes (considering namespace issues)
  - Code reader functions
  - Mobile device code reader – server interface
  - Backward compatibility with relevant deployed mobile codes

- Components specifically for the Direct method
  - At the time of writing, no work is expected

- Components specifically for the Indirect method
  - Network architecture comprising of functional roles
  - Standardised interfaces between functional entities (including the protocols)
  - Routing and connectivity between service entities
  - Security and privacy issues
  - Interoperability issues
  - Charging
  - Policy enforcement
  - Statistics generation
  - Namespace allocation to facilitate efficient routing

Appendix A. Example ecosystem roles

The following is a high-level description of relationships between functional roles in an ecosystem, as envisaged for some markets (this model is based on the Indirect method). See Figure 8. Note that an Actor may assume multiple roles in this service architecture both vertically (i.e. where it participates at multiple levels of the value chain by performing multiple functions) and/or horizontally (i.e. where it participates in multiple operator domains: local, regional, or global), subject to market dynamics and the scope of competition as governed by applicable regulatory constraints.

Interoperability between these various functions is a key objective of this ecosystem.

1) **Code Publisher.** This is a brand owner (businesses or individual) who chooses to acquire and present mobile codes in print or electronic media. Code Publishers may wish to execute advertising campaigns by directly acquiring codes themselves or may do so via a campaign agency that manages their interests. Therefore, they may interact with the Code Sales Agency, Code Management Platform or Code Registry.

2) **Code Sales Agency.** The Code Sales Agency acquires the rights to market codes obtained from a Code Management Platform provider and/or from a Code Registry, and sells them to Code Publishers.

3) **Code Management Platform.** The Code Management Platform function obtains the rights to distribute codes from the Code Registry, and adds technical value to the code generation and management process (e.g., code design enhancement, measurement and reporting of success rates of the advertising campaign). This entity also interacts with the Code Sales Agency. The Code Management entity may also develop a variety of "Ad Mechanics" that can be fulfilled by the code, such as "Launch Browser and dereference a URI", or "Launch SMS client, address it, and populate it with this message"; it could also design and host wireless web pages on behalf of the code publishers/code sales agency, or arrange for their access to user demographic data collected based on the campaign.

4) **Code Registry and Resolution.** This entity is comprised of 2 functional roles: a) the code registry, and b) the resolution (redirection) server. In the registry role, this entity authorises different actors performing at the same layer (i.e. multiple players of Code Sales Agencies, or multiple Code Management Platform providers) to activate and distribute codes. Other key responsibilities of the Code registry may include:

i. Ensuring that common codes work across operator domains and that no duplicates exist across the entire ecosystem.
ii. Providing a mechanism for each operator to approve these common codes for use in their networks.
iii. Synchronising multiple Code Registries, where applicable.

In the resolution role, this entity receives an identifier read from a mobile code and provides the content or URI corresponding to that identifier. The Code Registry could interact with the Code Publisher, the Code Sales Agency and/or the Code Management Platform.

5) **Code Clearing House.** This is an entity that sits between the Operator and the Code Management entity and is equipped with a database facility to ensure that the identifier from the Code Reader is *partially resolved* and sent to the appropriate Code Management Server for *full resolution*. The Clearing House can be independent of the Operator, can serve one Operator, or, subject to business agreements, can serve multiple Operators. In addition to routing of scanned codes to the appropriate Code Management Platform via the Code Registry, the Code Clearing House can also provide value-added functions (e.g. accounting functions, scanned code traffic monitoring, logging and reporting) to Code Publishers.

6) **Operator.** The Operator (or an entity performing an equivalent role) supports all required functions toward the End Users, including requiring Code Readers to support desired symbologies relevant to its markets. Towards Code Publishers, the Operators are responsible for connectivity to one or more Clearing Houses, or alternatively, connecting directly to one or more Code Registries.

7) **Code Reader.** Code Readers extract data from mobile codes and processes it accordingly. Code Readers in mobile devices are configured according to individual Operator requirements. Economies of scale should motivate the Code Reader to support symbologies that are both relevant and exclusive to the market.

8) **End User.** The End User is the consumer of services accessed by mobile codes. The End User device may include a pre-installed or downloaded Code Reader.

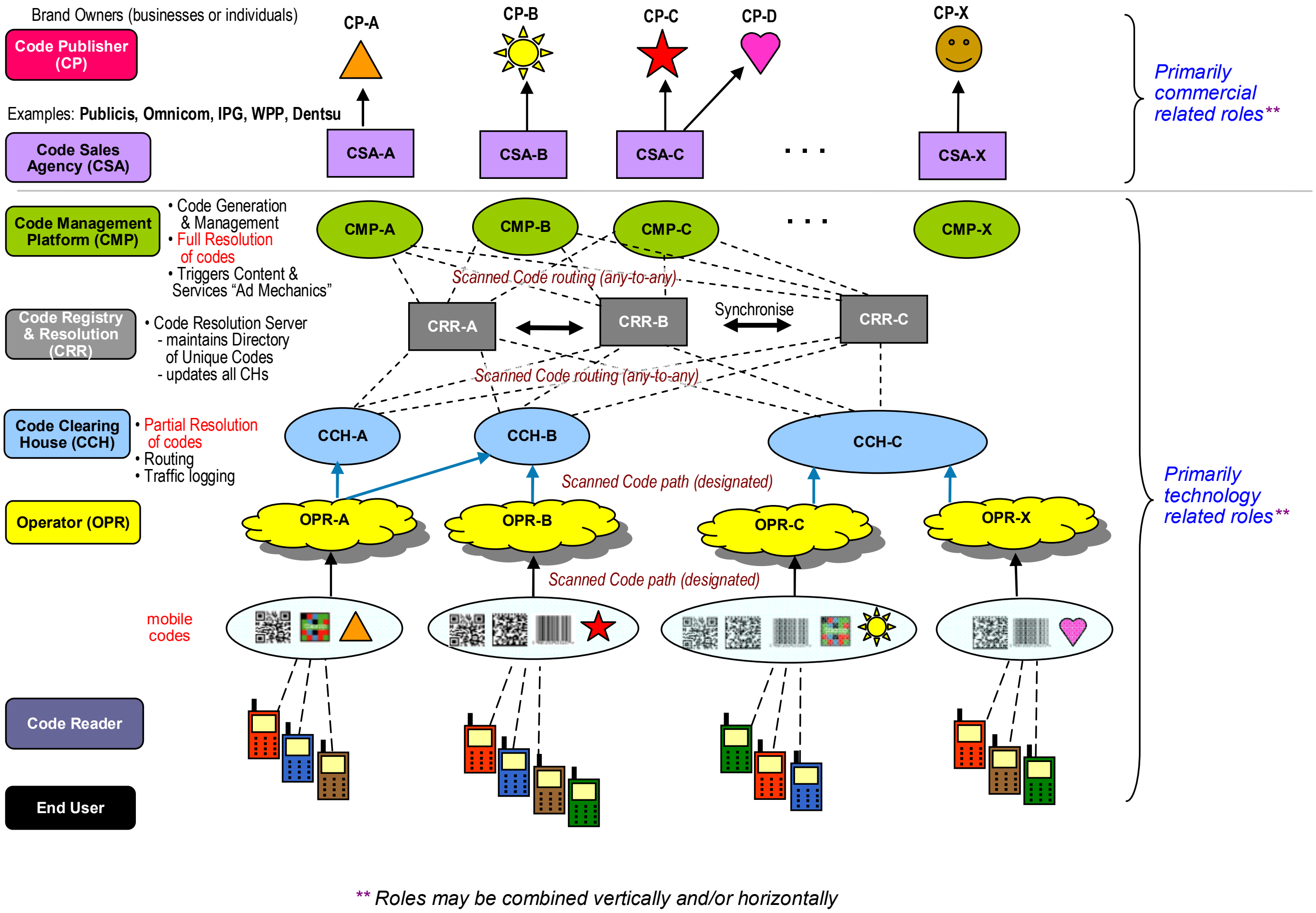

Figure 8. Indirect mobile code decoding conceptual service architecture (possible roles)

Appendix B. Change History (Informative)

Document Identifier	Date	Sections	Description
OMA-WP-MobileCodes-20071019-D	19 Oct 2007	all	initial outline added by Tim Kindberg (HP)
OMA-WP-MobileCodes-20080115-D	15 Jan 2008		Changes marked up from London f/f meeting Nov 2007; also integrated material on Data Matrix from OMA-BT-MC-2007-0002R02-INP_Datamatrix_ISO_standard, also discussed in London.
OMA-WP-MobileCodes-20080122-D	22 Jan 2008		Editorial updates - implementation of the 2008 template - cover page to display the correct WP title - References and Abbreviations sorted alphabetically - styles fixed - Figures and Tables caption fixed - bookmarks added
OMA-WP-MobileCodes-20080208-D	08 Feb 2008	5.2.1, 6.2.2, 7.2.1, 7.2.2, 8	Accepted all changes from London f/f meeting Nov 2007. Integrated change requests agreed 20080204.
OMA-WP-MobileCodes-20080215-D	15 Feb 2008		Same as OMA-WP-MobileCodes-20080208-D except highlighted changes made in deriving it from OMA-WP-MobileCodes-20080122-D
OMA-WP-MobileCodes-20080305-D	03 Mar 2008	2, 4.2, 4.3, 5.0, 5.2, 5.3, 5.4, 6.0, 6.1.2, 6.2.0, 6.2.1, 6.2.2.2, 6.2.3, 7, 8, 9, Appendix A	Changes made during Beijing f/f meeting 25-27 Feb 2008. Subsequent editorial fixes.
OMA-WP-MobileCodes-20080313-D	13 Mar 2008		Changes shown in OMA-WP-MobileCodes-20080305-D accepted. Figure captions fixed. Figure in Appendix A replaced with original PowerPoint slide. Spellings made consistent. Fixed broken references to DoCoMo information. Tables of contents updated.
OMA-WP-MobileCodes-20080404-D	04 Apr 2008	6.2.3, ToC	Fixed formatting so that Flashcode section appears in ToC
OMA-WP-MobileCodes-20080617-C	17 Jun 2008	All	Status changed to Candidate by TP: OMA-TP-2008-0231-INP_MobileCodes_V1_0_RRP_for_Candidate_Approval.
OMA-WP-MobileCodes-20081024-A	24 Oct 2008	All	Approved by OMA TP OMA-TP-2008-0393-INP_MobileCodes_V1_0_RRP_for_Final_Approval