INTERNATIONAL STANDARD

ISO/IEC 29179

First edition
2012-02-01

Information technology — Mobile item identification and management — Mobile AIDC application programming interface

Technologies de l'information — Gestion et identification d'élément mobile — Interface de programmation pour application AIDC mobile

Reference number
ISO/IEC 29179:2012(E)

COPYRIGHT PROTECTED DOCUMENT

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 29179 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 31, *Automatic identification and data capture techniques*.

Introduction

Mobile AIDC is a version of automatic identification and data capture (AIDC) technology that is combined with mobile communications for the purpose of creating novel services that provide information to Mobile AIDC terminal users just by pointing the terminal at the objects of interest. Some emerging services supported by Mobile AIDC technology are wine authentication, movie reviews and trailers, and tourist information. Mobile AIDC terminals are equipped with various AIDC devices such as a radio frequency (RF) interrogator or optically readable media (ORM) reader, and interact with data carriers around the terminal by activating embedded AIDC devices. The data carriers contain the uniform resource identifiers (URIs) of the information associated with the objects to which the data carriers are affixed, or contain user data that the Mobile AIDC terminal users intend to use. If data carriers include the URIs of associated information, the Mobile AIDC applications try to connect to the content servers to get the necessary information. From a service viewpoint, the Mobile AIDC service users interact with the information infrastructure by using Mobile AIDC terminals in a way similar to how web users use the web by mouse-clicking. This International Standard intends to cover the generic mobile application programming interfaces for Mobile AIDC application development and running.

A Mobile AIDC application is a type of mobile application that runs on a mobile application platform. The mobile application platform is the development and running environment for mobile applications for the management and coordination of activities and sharing of the limited resources of the mobile terminal. There are many kinds of mobile application platforms providing various environments or mobile applications. Mobile AIDC applications need access to data carriers such as RF tags, barcodes, etc. for reading and/or writing content. For this purpose, Mobile AIDC applications use embedded Mobile AIDC devices such as Mobile radio frequency identification (RFID) interrogators or Mobile ORM readers. To make this possible, mobile application platforms provide appropriate application programming interfaces to mobile applications.

This International Standard defines Mobile AIDC application programming interfaces such as *open, close, read, write, lock*, etc. This International Standard does not specify implementation level interfaces, but abstract level interfaces, because of the diversity of mobile application platforms. Different mobile application platforms have their own particular application programming interface sets or data dictionaries.

This International Standard uses a Unique Item Identifier (UII) as an identifier for the object to which the data carrier is affixed. Implementers may use an instance of Mobile Item Identifier (MII) defined in ISO/IEC 29174 as a UII or may use an instance of other code schemes as a UII. Implementers who wish to implement GS1 standards using the Electronic Product Code (EPC) as the UII are encouraged to contact GS1.

Information technology — Mobile item identification and management — Mobile AIDC application programming interface

1 Scope

This International Standard provides a description of Mobile AIDC applications and specifies the functional requirements of Mobile AIDC application interfaces. It defines abstract Mobile AIDC application interfaces to provide a standardized functional view over various Mobile AIDC application platforms.

2 Conformance

To claim conformance with this International Standard, a Mobile AIDC application platform shall comply with all relevant clauses of this International Standard, except those marked as “optional”.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 19762-1, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 1: General terms related to AIDC*

ISO/IEC 19762-3, *Information technology — Automatic identification and data capture (AIDC) techniques — Harmonized vocabulary — Part 3: Radio frequency identification (RFID)*

ISO/IEC TR 29172, *Information technology — Mobile item identification and management — Reference architecture for Mobile AIDC services*

4 Terms, definitions and abbreviated terms

4.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 19762-1, ISO/IEC 19762-3, ISO/IEC TR 29172 and the following apply.

4.1.1
correspondent
communication participant by means of telecommunication networks

4.1.2
Mobile AIDC device identifier
data used for identification of Mobile AIDC devices in a Mobile AIDC terminal

4.1.3
Mobile AIDC application interface
generic application interface to handle AIDC devices attached to mobile terminals

4.1.4
mobile application
computer program designed to operate in a mobile terminal

4.1.5
mobile application platform
set of computer instructions as the place to launch a mobile application in a mobile terminal

4.1.6
mobile terminal
handheld electronic device, e.g. a mobile telephone, a PDA, etc., used for voice or data communication over a wireless communication network

4.1.7
unique item identifier
UII
identification that uniquely identifies a specific entity (e.g. package, transport unit, returnable asset) during its life within a particular domain and scope of a code system

NOTE 1 When used with this data protocol, the particular object identifier that defines the unique item identifier relies on the fact that each instance of its object is unique and unambiguous with all other related objects.

NOTE 2 As the object is unique, its use in the RF tag confers uniqueness to the RF tag itself. [ISO/IEC 19762-1].

NOTE 3 For the purposes of this International Standard the concept of MII is included within UII.

4.1.8
UII resolution server
process of converting UII into its associated information by means of a designated resolution method for the UII

NOTE When an instance of MII is used as UII, then Object Directory Service (ODS) is a UII resolution server. While EPC code is used as UII, then Object Naming Service (ONS) is a UII resolution server.

4.2 Abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO/IEC TR 29172 and the following apply.

ONS Object Naming Service

URI Uniform Resource Identifier

5 Overview

5.1 Mobile AIDC applications description

Generally, mobile applications run on the platforms that provide the running environment for mobile terminals. To provide value-added services, mobile applications can use various peripherals (e.g. camera, Bluetooth, or mp3 player) attached to the mobile terminals via mobile application platforms. The Mobile AIDC application uses a device such as an RFID interrogator, or an ORM reader. By using Mobile AIDC devices, the applications interact with data carriers such as RFID tags, bar codes or two-dimensional symbols.

More specifically, if a Mobile RFID interrogator is embedded in a mobile terminal, the application uses the interrogator to get the data from RF tags around the Mobile AIDC terminal user. Or, if an ORM reader is

embedded in a mobile terminal, the application uses the ORM reader to read data from symbols nearby. Mobile AIDC applications may use the data by itself or they may use the network to get relevant information from the content servers. When using a network, Mobile applications use the network access utility provided by the application platforms. In some cases, Mobile AIDC applications try to write data to data carriers or to erase data from other data carriers.

The characteristics of the Mobile AIDC applications are summarized as follows:

- They use Mobile AIDC devices to interact with data carriers
- Supporting operations are restricted within the capabilities of Mobile AIDC devices and data carriers
- They may use a network to obtain information associated with the UIIs read from data carriers
- Their implementations are highly dependent on Mobile AIDC application platforms. Therefore, it is almost impossible to define common application programming interfaces at the implementation level for all Mobile AIDC devices and Mobile AIDC application platforms

5.2 Overview of the Mobile AIDC application interfaces

As mentioned in Clause 5.1, a characteristic of Mobile AIDC applications is that they use an appropriate Mobile AIDC device to interact with data carriers around a Mobile AIDC terminal. To support the applications' operations, interfaces are needed to control the devices and to process the data obtained by the devices. But there are different types of Mobile AIDC devices and their functions and handling methods are all different. Some devices provide only UIIs read from data carriers. Others may provide other functions such as reading user data from data carriers or writing user data to data carriers. Therefore, in this International Standard, possible operations are enumerated and are marked as mandatory or optional.

The main function of a Mobile AIDC application is to read UIIs from data carriers by using Mobile AIDC devices attached to a Mobile AIDC terminal in order to obtain the relevant information via a network. To support this function, the interface to read the UIIs shall be defined. Generally, all devices shall be opened beforehand. After using the device, the device shall be closed. Therefore, the following interfaces shall be defined and implemented on the application platforms regardless of device types:

— *Open Device (Mobile AIDC device Identifier): To use the Mobile AIDC device, a Mobile AIDC application shall open the device*

— *Close Device (Mobile AIDC device Identifier): After using the Mobile AIDC device Identifier, a Mobile AIDC application shall close the device*

— *Read UIIs (Mobile AIDC device identifier): Read UIIs from surrounding data carriers*

Some types of Mobile AIDC devices provide functions such as reading user data written in data carriers, writing user data on certain data carriers, or erasing data from certain data carriers. To support these functions, the following interfaces shall be defined and implemented on the application platforms:

— *Read user data (Mobile AIDC device identifier, Mobile AIDC media identifier): Read user data from a certain data carrier*

— *Write data (Mobile AIDC device identifier, Mobile AIDC media identifier, user data, repeat count [, access password, new access password, new Mobile AIDC media identifier]): Write data on a certain data carrier*

— *Erase user data (Mobile AIDC device identifier, Mobile AIDC media identifier, repeat count [, access password]): delete user data from a certain data carrier*

Some types of Mobile AIDC devices provide a security facility to protect user data from overwriting. To provide this function, the following interface shall be implemented.

— *Lock/unlock (Mobile AIDC device identifier, Mobile AIDC media identifier, operation type, memory bank [, access password]): Protect (or release) data from overwriting*

Some types of Mobile AIDC devices read multiple UIIs from multiple data carriers around the terminal simultaneously. In this case, reducing target UIIs is reasonable, considering the processing costs. If the devices support a filtering capability or the application platforms support a filtering capability, the following interface shall be implemented.

— *Filtering UIIs (Mobile AIDC device identifier, filter type, mask): filter UIIs read by a Mobile AIDC device using an appropriate mask*

Annex A shows where this International Standard works. A Mobile AIDC application command takes the form of this International Standard and uses appropriate parameters to exchange necessary information between a Mobile AIDC application and data carriers through an appropriate Mobile AIDC application platform. In this International Standard, a UII uniquely identifies a data carrier within a Mobile AIDC service domain. For example, an instance of ISO/IEC 29174 (MII) may be used as an instance of UII within a Mobile AIDC service domain. User data in this International Standard contains values for a specific Mobile AIDC service and may take a form of ISO/IEC 29175.

An application command of a Mobile AIDC application is delivered from an application platform to a specific device and is transformed into a data carrier-specific command. Communication interfaces between a Mobile application platform and a Mobile device and between a Mobile device and data carriers are dependent on an application platform implementation and data carrier technology. For instance, if a Mobile AIDC device is a Mobile RFID interrogator, then a Mobile AIDC application platform may use ISO/IEC 29173-1 to exchange information with the RFID interrogator and ISO/IEC 18000-6 or ISO/IEC 29143 may be used between a Mobile AIDC device and the data carriers.

This international standard supports only ISO/IEC 18000-6 Type C, ISO/IEC 29143 and Mobile ORM.

6 Mobile AIDC application programming interfaces

6.1 General

Mobile AIDC applications may run on various Mobile AIDC application platforms. Each mobile application platform has unique API sets and data dictionaries. Due to this diversity, it is impossible to make mobile application implementations interoperable among the various mobile application platforms.

This International Standard defines abstract Mobile AIDC application interfaces to give a standardized functional view of the various Mobile AIDC application platforms. Implementation-level application interfaces shall be defined from this abstract set of Mobile AIDC application interfaces. This international standard does not cover the security aspect of the interface. Security is defined in the air interface.

Generally, a device registration at the mobile terminal is the first step to use that device. The exact method to register devices and the device names are highly dependent on mobile application platform implementations. The registration process is outside the scope of this International Standard.

6.2 Open (mandatory)

To use a Mobile AIDC device attached to a Mobile AIDC terminal, a Mobile AIDC application shall open the device first. *Open* connects an application to a device and then the device gets ready to operate.

Exact input parameter(s), data type(s) of input parameter(s), return value(s) and data type(s) of return value(s) shall comply with the application platform implementation on which the application executes.

- input parameter
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
- return value
    - success/fail

6.3 Close (mandatory)

After using a Mobile AIDC device, a Mobile AIDC application shall close the device to conserve energy. *Close* disconnects the application from the device. At the same time, the application platform instructs the device to power down.

Exact input parameter(s), data type(s) of input parameter(s), return value(s) and data type(s) of return value(s) shall comply with the application platform implementation on which the application executes.

- input parameter
  - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
- return value
  - success/fail

6.4 Read (mandatory)

To get UIIs from surrounding data carriers, a Mobile application directs the Mobile AIDC device to read UIIs. If the user data are written in data carriers and the device provides a reading function for user data, then the application may get both UIIs and user data at the same time. If TID is written in the data carriers and the device provides a reading function for TID, then the application may get both UIIs and TIDs at the same time. If TIDs and user data are written in the data carriers and the device provides a reading function for them, the application may get all of the data at the same time.

Exact input parameter(s), data type(s) of input parameter(s), return value(s), data type(s) of return value(s) and response shall comply with the application platform implementation on which the application executes.

The *Read* operation usually takes time to complete. Therefore, it is recommended to implement in an asynchronous way. A Mobile AIDC application platform may process other jobs while awaiting a response(s). When *Read* is implemented in an asynchronous way, then the return value is returned immediately to notify the application of a successful calling of the interface and a successful validation check by the platform. Actual data will be returned as a response(s).

- input parameters
  - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
  - read_type:
    - 00: read UIIs
    - 01: read UIIs and user data
    - 10: read UIIs and TIDs
    - 11: read all
- return value
  - success/fail
- response
  - data read from data carriers

    NOTE: The returned data corresponding to the UIIs consist of entire UII memory content except StoredCRC of 18000-6:2010. Parsing of the returned data in order to extract the UII is dependent upon functionality of the applications.

6.5 ReadUserData (optional)

To get user data from a specific data carrier, a Mobile AIDC application requests a Mobile AIDC device to read user data from the specific data carrier.

Exact input parameter(s), data type(s) of input parameter(s), return value(s), data type(s) of return value(s) and response shall comply with the application platform implementation on which the application executes.

The *ReadUserData* operation usually takes time to complete. Therefore, it is recommended to implement in an asynchronous way. A Mobile AIDC application platform may process other jobs while awaiting a

response(s). When *ReadUserData* is implemented in an asynchronous way, the return value is returned immediately to notify the application of a successful calling of the interface and a successful validation check by the platform. Actual data will be returned as a response(s).

- input parameters
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
    - Mobile AIDC media identifier: UII of a target data carrier (ex. Content of MB01 except StoredCRC of 18000-6:2010)
- return value
    - success/fail
- response
    - user data read

6.6 Write (optional)

To record data in the specific data carrier, a Mobile AIDC application directs the Mobile AIDC device to write data to the data carrier. *Write* records data to the specific data carrier. *Write* is performed numerous times set by the application program to meet the operational need.

Exact input parameter(s), data type(s) of input parameter(s), return value(s), data type(s) of return value(s), data type(s) of response, and data value(s) of response shall comply with the application platform implementation on which the application executes.

The *Write* operation usually takes time to complete. Therefore, it is recommended to implement in an asynchronous way. An application platform may process other jobs while awaiting a response(s). When *Write* is implemented in an asynchronous way, the return value is returned immediately to notify the application of a successful calling of interface and a successful validation check by the platform. The actual result will be delivered as a response(s).

- input parameters
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
    - Mobile AIDC media identifier: UII written in the data carrier (ex. Content of MB01 except StoredCRC of 18000-6:2010)
    - access password: if programmed and data carrier with UII is locked, the access password written in the data carrier
    - user data: user data to write
    - new access password: new access password to overwrite existing one
    - new Mobile AIDC media identifier: new UII which is to be written in the data carrier for security reasons
    - repeat count: repeat count
- return value
    - success/fail
- response
    - success/fail

6.7 Erase (optional)

To delete user data from a certain data carrier, a Mobile AIDC application directs the Mobile AIDC device to erase data from the specific data carrier.

Exact input parameter(s), data type(s) of input parameter(s), return value(s), and data type(s) of return value(s) shall comply with the application platform implementation on which the application executes.

The *Erase* operation usually takes time to complete. Therefore, it is recommended to implement in an asynchronous way. A Mobile AIDC application platform may process other jobs while awaiting a response(s). When *Erase* is implemented in an asynchronous way, the return value is returned immediately to notify the

application of a successful calling of interface and a successful validation check by the platform. The actual result will be delivered as a response(s).

- input parameters
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
    - Mobile AIDC media identifier: UII written in the data carrier (ex. Content of MB01 except StoredCRC of 18000-6:2010)
    - access password: if programmed and data carrier with UII is locked, the access password written in the data carrier
    - repeat count: repeat count
- return value
    - success/fail
- response
    - success/fail

6.8 Lock (optional)

To protect user data written in the specific data carrier from overwriting, a Mobile AIDC application locks the data carrier. If a certain data carrier is locked, the write operation is not permitted. If necessary, the data carrier can be unlocked.

Exact input parameter(s), data type(s) of input parameter(s), return value(s) and data type(s) of return value(s) shall comply with the Mobile AIDC application platform implementation on which a Mobile AIDC application executes.

The *Lock* operation usually takes time to complete. Therefore, it is recommended to implement in an asynchronous way. A Mobile AIDC application platform may process other jobs while awaiting a response(s). The actual result will be delivered as a response(s). When *Lock* is implemented in an asynchronous way, the return value is returned immediately to notify the application of a successful calling of the interface and a successful validation check by the platform. The actual result will be delivered as a response.

- input parameters
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
    - Mobile AIDC media identifier: UII written in the data carrier (ex. Content of MB01 except StoredCRC of 18000-6:2010)
    - op_type: lock or unlock
    - access password: if programmed, the access password written in the data carrier
    - memory_bank: bank number to lock or unlock

- return value
    - success/fail
- response
    - success/fail

6.9 Filter (optional)

To discard UIIs that are not intended, a Mobile AIDC application uses a filtering mask in either way, inclusive or exclusive.

Exact input parameter(s), data type(s) of input parameter(s), return value(s) and data type(s) of return values shall comply with the application platform implementation on which the application executes. When it comes to a mask, it shall take code type of the UIIs into consideration. To filter unintended UIIs, a Mobile AIDC application shall know the structure of the code and shall then organize the mask to filter them out effectively.

- input parameter
    - Mobile AIDC device identifier: a Mobile AIDC device identifier defined at the Mobile AIDC application platform at registration time
    - filter type: inclusive or exclusive
    - mask: a mask to filter UIIs
- return value
    - success/fail

Annex A
(informative)

Mobile AIDC standard architecture

Figure A.1 describes where this International Standard works.

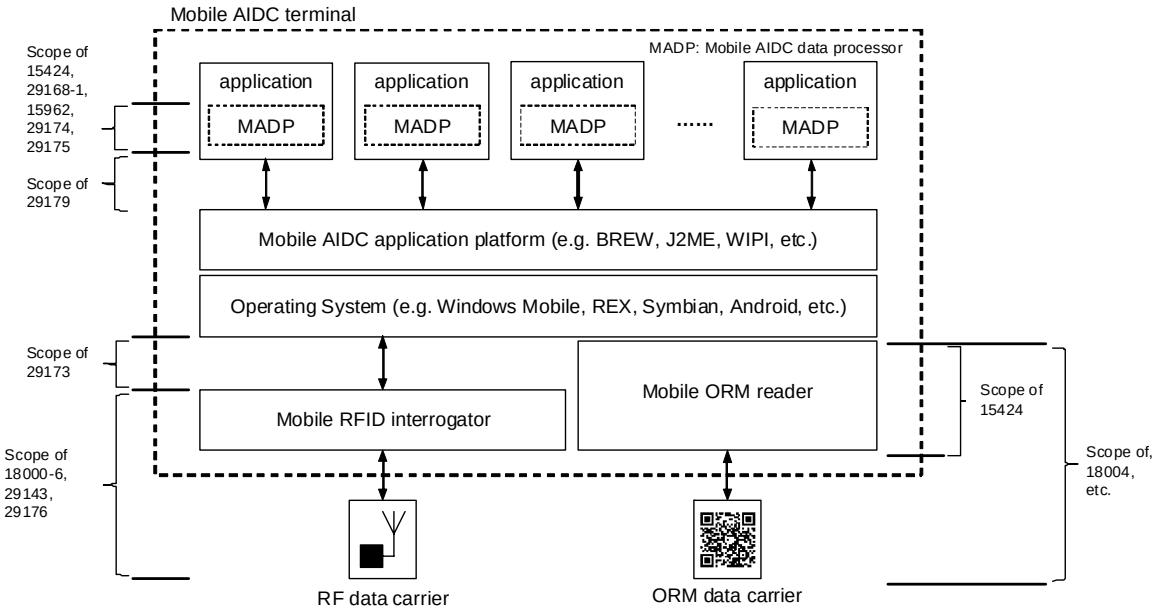

Figure A.1 — Mobile AIDC standard architecture [*ISO/IEC 29172*]

Annex B
(informative)

Mobile AIDC service scenarios

B.1 Introduction

There are various types of Mobile AIDC applications such as wine authentication, movie reviews and trailers, tourist information or simply calling a friend. For the private use of data carriers after purchasing objects, some Mobile applications may manipulate data carriers as if the data carriers were a private data storage device. For example, after purchasing a product, a terminal user might want to encode messages on the data carrier affixed to the product so that every time the data carrier is read, the message can be retrieved. In this case, some security measures would be required to protect from overwriting the original message.

In Annex B, three cases of Mobile AIDC service scenarios are described, depicting possible operation flows for Mobile AIDC applications.

B.2 Scenario: Obtaining content through UII resolution

Mobile AIDC applications provide information to the Mobile AIDC terminal users based on UIIs read by the Mobile AIDC devices. In this service, Mobile applications read UIIs from data carriers around the Mobile terminal users. The terminal users then use the network for UII resolution and for relevant information acquisition. Figure B.1 shows possible operation flows.

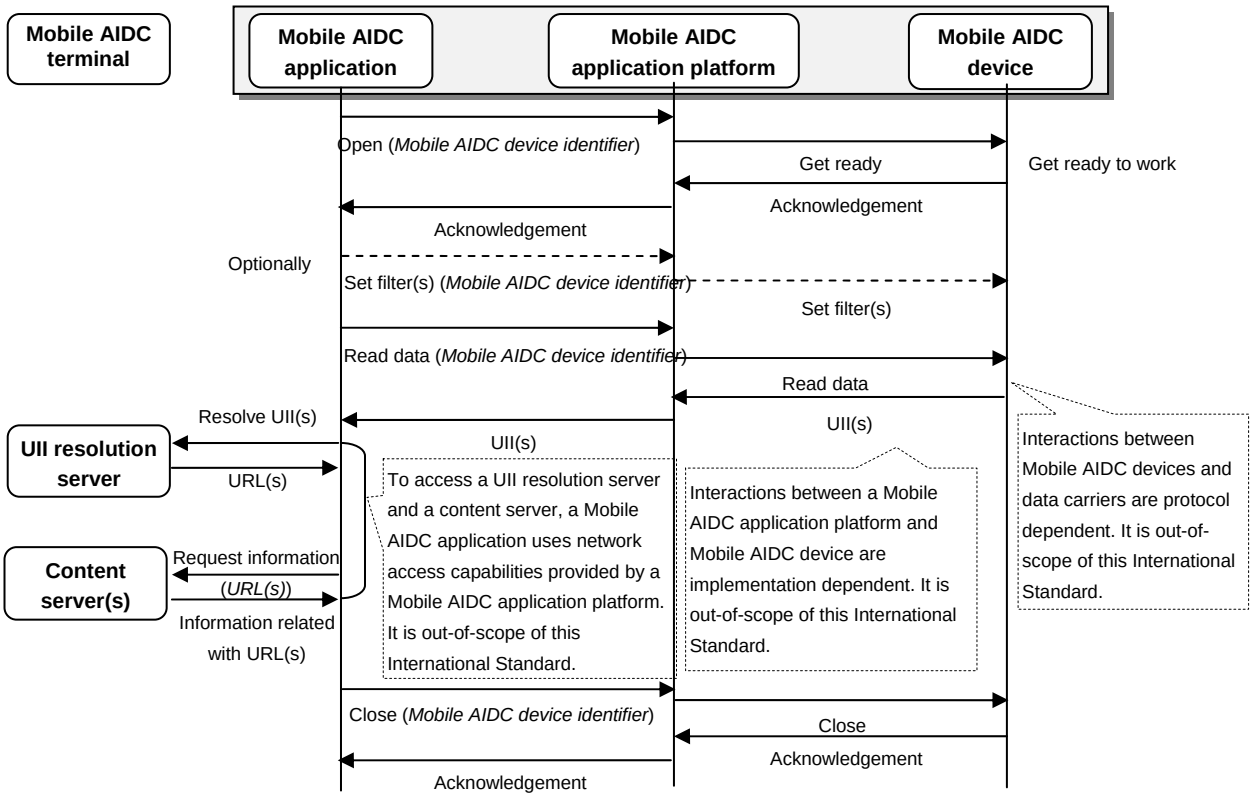

Figure B.1 — Mobile AIDC service scenario: Obtaining relevant information through UII resolution

All implementation level operations, input parameters, output parameters and the method of acknowledgement are different based on the Mobile AIDC application platforms. And all interactions between the Mobile applications and the Mobile devices are performed via Mobile application platforms. Abstract operation flows are as follows:

- A Mobile AIDC application directs a Mobile AIDC device to get ready to operate through the Mobile AIDC application platform.
- The Mobile device sends an appropriate acknowledgement to the Mobile application.
- The application directs the device to read UIIs.
- The device reads the UIIs within a reading range. Usually, it reads multiple UIIs from data carriers around the Mobile AIDC device. Therefore, a mobile application may use appropriate filters to get only the intended UIIs. Then, the Mobile device returns the UIIs to the application.
- The Mobile application performs the UII resolution by using a UII resolution server. Then the resolution server returns the relevant URLs of the content servers to the application. The Mobile application uses the network access facilities provided by the application platform to access a UII resolution server.
- The application sends an information-request to the content servers based on the URLs to get UII-relevant information. The application uses the network access facilities provided by the application platform to access the content servers.
- The content server returns the information to the application.
- After using the Mobile device, the application directs the device to close.

B.3 Scenario: Obtaining content without UII resolution

Mobile AIDC applications provide information to the Mobile AIDC terminal users based on data read by a Mobile AIDC device. In this service, Mobile applications read user data from data carriers around the Mobile AIDC terminal users. The Mobile terminal users then use the network for relevant information acquisition. Figure B.2 shows possible operation flows.

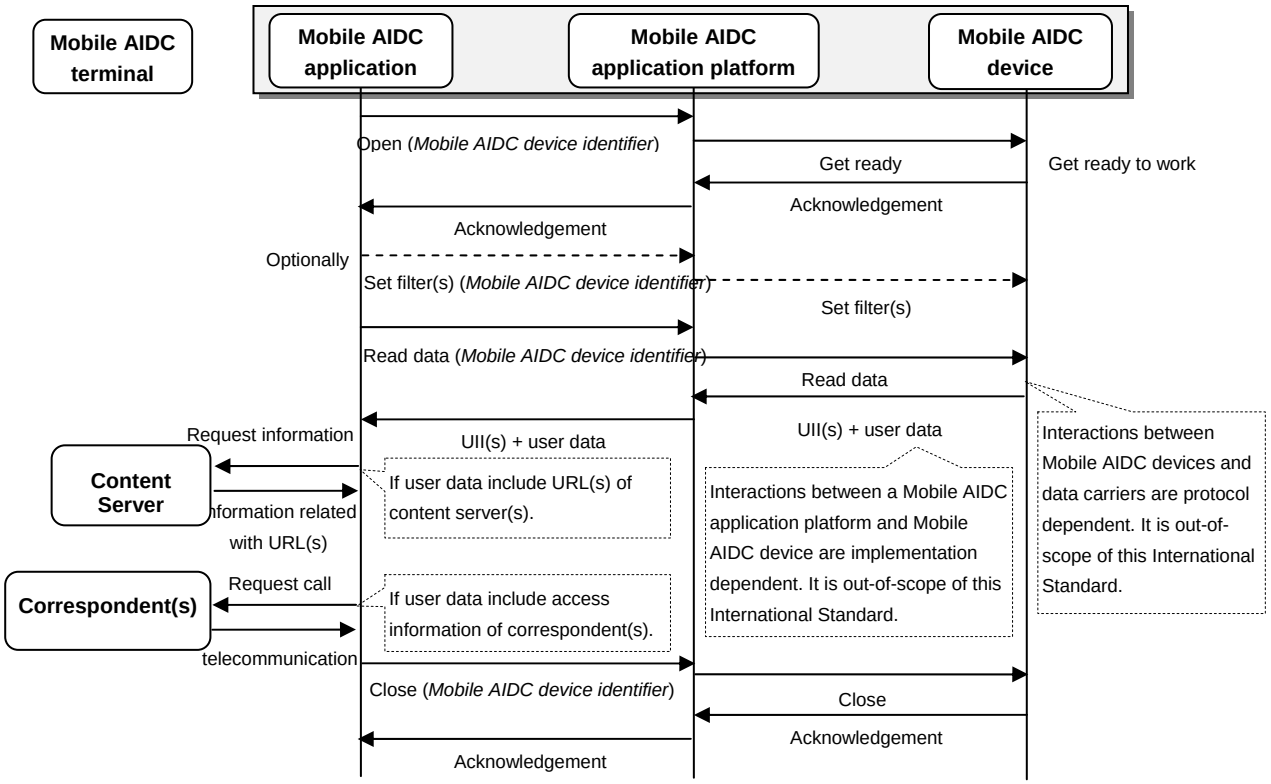

Figure B.2 — Mobile AIDC service scenario: Obtaining relevant information without UII resolution

All implementation level operations, input parameters, output parameters and the method of acknowledgement are different based on the implementation of the Mobile AIDC application platforms. In this case, user data includes either URL(s) of content server(s) or access information of correspondent(s) such as a phone number(s). If user data includes URL(s) of content server(s), the Mobile AIDC applications connect to the content server(s) to get the relevant information. Otherwise if the user data includes access information of correspondent(s), the Mobile AIDC applications make calls to the correspondents by using acquired access information. Abstract operation flows are as follows:

- A Mobile AIDC application directs a Mobile AIDC device to get ready to operate through the Mobile AIDC application platform.
- The Mobile device sends an appropriate acknowledgement to the Mobile application.
- The application directs the device to read data {UII + user data}s.
- The device reads data within a reading range. To extract possible candidates from all data, the application may set appropriate filters before reading.
- The Mobile AIDC application analyses user data.
    - o If the user data includes a URL(s) of content server(s), the application sends an information-request to the content server(s) based on the URL(s) to get the relevant information. The application uses network access facilities provided by the application platform to access a content server.
    - o If the user data includes access information of the correspondent(s) (*in this case, it is a phone number*), the Mobile application tries to connect to the correspondent(s). The application uses call-setup facilities provided by the application platform to make a call to the correspondent.

    NOTE: if multiple correspondents' information is written in user data area, then a Mobile AIDC application calls the first number. If the call fails, the application tries to connect to next number. It is implementation-dependent.
- After using the Mobile device, the application directs the device to close.

B.4 Scenario: Using a data carrier as a storage device

Another possible service is to use a data carrier as a storage device. This service can be possible only when both Mobile AIDC devices and data carriers are programmed appropriately. In this case, a service framework consists of data carriers and a Mobile AIDC terminal on which a Mobile AIDC application runs. The operations supported are dependent on the capabilities of the Mobile devices and the data carriers. For example, if both the Mobile devices and the data carriers comply with ISO 18000-6 type C air interface, the following operations are supported:

- Read user data
- Write data
- Erase data
- Lock/unlock

And if the Mobile devices provide a filtering function, the filtering operation may be performed. Figure B.3 shows possible operation flows.

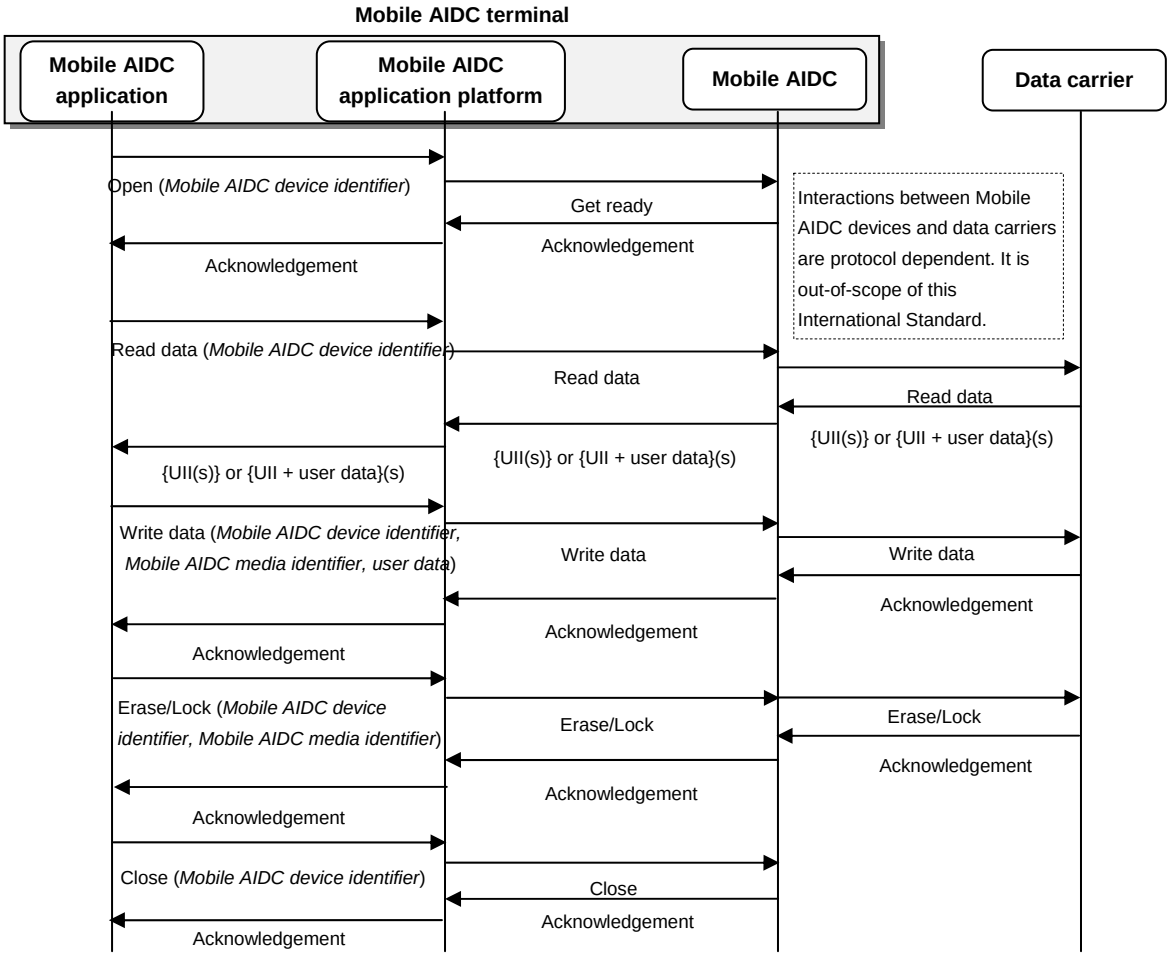

Figure B.3 — Mobile AIDC service scenario: user data storage

The operation flows are as follows:

- A Mobile AIDC application directs a Mobile AIDC device to get ready to operate through the Mobile AIDC application platform.
- The Mobile AIDC device sends an appropriate acknowledgement to the Mobile AIDC application.
- The Mobile AIDC application directs the Mobile AIDC device to read data (UIIs or {UII + user data}s).

- The Mobile AIDC device reads data within a reading range. To extract possible candidates from all data, the Mobile AIDC application may use appropriate filters before reading.
- Or, the Mobile AIDC application directs the Mobile AIDC device to write data on the specific data carrier.
- Or, the Mobile AIDC application directs the Mobile AIDC device to erase data from the specific data carrier.
- Or, the Mobile AIDC application directs the Mobile AIDC device to lock/unlock the specific data carrier.
- After using the Mobile AIDC device, the Mobile AIDC application directs the Mobile AIDC device to get closed.

Annex C
(informative)

Sample APIs

Annex C describes sample APIs which are derived from this International Standard.

```
package org.iso.aidc.miim;

import org.iso.aidc.miim.exception.MIIMException;

/**
 * Additional information can be exchanged between a Mobile AIDC application platform and a Mobile AIDC application by adding more parameters (I/O) according to implementation.
 */
public interface MIIM29179APIs {

    /**
     * Connect the Mobile AIDC device to use
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @throws MIIMException
     */
    public void open(String deviceId) throws MIIMException;

    /**
     * Disconnect the Mobile AIDC device after use it
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @throws MIIMException
     */
    public void close(String deviceId) throws MIIMException;

    /**
     * Read data from tags in a form of {UIIdata, user data, TID}s. UIIdata is whole content of UII memeory bank except CRC
     * Data types of UIIdata, user data and TID are byte stream in this sample APIs
     * length information is not included in the sample APIs. This sample APIs focus on main parameters only.
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @param readType if readType="00", then call ISO/IEC 29173-1 "Read tag once" with (MB=0x01). In this case, response is the content stored at UII memory bank except CRC <br />
     *                 if readType="01", then call ISO/IEC 29173-1 "Read tag once" twice per each tag; the first read is for UIIdata(MB=0x01) and next read is for user data(MB=0x03) <br />
     *                 if readType="10", then call ISO/IEC 29173-1 "Read tag once" twice per each tag; the first read is for UIIdata(MB=0x01) and next read is for reading TID(MB=0x02)<br />
```

```
     *          if readType="11", then call ISO/IEC 29173-1 "Read tag once" three times per each tag; the first read is for UII(MB=0x01)data, next is for user data(MB=0x03) and final read is for TID(MB=0x02)<br />
     * @return if a Mobile AIDC application platform implements this interface in a synchronous way, return value is the data read <br />
     *          if readType="00", then return {UII, null, null} list <br />
     *          if readType="01", then return {UII, user data, null} list <br />
     *          if readType="10", then return {UII, null, TID} list <br />
     *          if readType="11", then return {UII, user data, TID} list  <br />
     * @throws MIIMException
     */
    public byte[] read(String deviceId, byte readType) throws MIIMException;

    /**
     * Read user data from the specific tag identified by mediaId
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @param mediaId the Mobile AIDC media identifier from which a Mobile AIDC application reads user data
     * @return user data(s)
     * @throws MIIMException
     */
    public byte[] readUserData(String deviceId, byte[] mediaId) throws MIIMException;

    /**
     * Write data to a specific tag identified by mediaId
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @param mediaId the Mobile AIDC media identifier on which a Mobile AIDC application writes data
     * @param accessPassword if an access password is programmed on a tag, set accessPassword with the same access password or set this parameter with a null value
     * @param userData if write is for writing user data, then set this parameter with user data or set this parameter with a null value.
     * @param newAccessPassword if write is for writing a new access password, then set this parameter with a new access password or set this parameter with a null value
     * @param newMediaId if write is for writing a new UII, set this parameter with a new UIIdata(UII memeory content except CRC) or set this parameter with a null value
     * @param repeatCount for ensuring write, do it repeatCount times repeatedly
     * @throws MIIMException
     */
    public void write(String deviceId, byte[] mediaId, byte[] accessPassword, byte[] userData, byte[] newAccessPassword, byte[] newMediaId, int repeatCount) throws MIIMException;

    /**
     * Erase data from a specific tag identified by mediaId
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @param mediaId a Mobile AIDC media identifier from which a Mobile AIDC application erases data
     * @param accessPassword if an access password is programmed on a tag, set accessPassword with the same
```

```
access password or set this parameter with a null value
     * @param repeatCount for ensuring erase, do it repeatCount times repeatedly
     * @throws MIIMException
     */
    public void erase(String deviceId, byte[] mediaId, byte[] accessPassword, int repeatCount) throws MIIMException;

    /**
     * Lock a specific tag identified by mediaId
     * @param deviceId the Mobile AIDC device identifier assigned by a Mobile AIDC application platform
     * @param mediaId a Mobile AIDC media identifier on which a Mobile AIDC application locks data
     * @param opType "lock" or "unlock"
     * @param accessPassword if an access password is programmed on a tag, set accessPassword with the same access password or set this parameter with a null value
     * @param memoryBank memory bank to lock
     * @throws MIIMException
     */
    public void lock(String deviceId, byte[] mediaId, String opType, byte[] accessPassword, byte memoryBank) throws MIIMException;

    /**
     * Set filter values
     * @param deviceId a MIIM device identifier assigned by a Mobile AIDC application platform
     * @param filterType "inclusive" or "exclusive"
     * @param mask for filtering in a form of UII
     * @throws MIIMException
     *
     */
    public void filter(String deviceId, String filterType, byte[] mask) throws MIIMException;

}
```

Bibliography

[1] ISO/IEC 15424:2008, *Information technology — Automatic identification and data capture techniques — Data Carrier Identifiers (including Symbology Identifiers)*

[2] ISO/IEC 15961:2004, *Information technology — Radio frequency identification (RFID) for item management — Data protocol: application interface*

[3] ISO/IEC 15962, *Information technology — Radio frequency identification (RFID) for item management — Data protocol: data encoding rules and logical memory functions*

[4] ISO/IEC 15963, *Information technology — Radio frequency identification for item management — Unique identification for RF tags*

[5] ISO/IEC 18000-3, *Information technology — Radio frequency identification for item management — Part 3: Parameters for air interface communications at 13,56 MHz*

[6] ISO/IEC 18000-6, *Information technology — Radio frequency identification for item management — Part 6: Parameters for air interface communications at 860 MHz to 960 MHz*

[7] ISO/IEC 18004:2006, *Information technology — Automatic identification and data capture techniques — QR Code 2005 bar code symbology specification*

[8] ISO/IEC 29143, *Information technology — Automatic identification and data capture techniques — Air interface specification for Mobile RFID interrogators*

[9] ISO/IEC 29168-1 | ITU-T X.672, *Information technology — Open systems interconnection — Object identifier resolution system*

[10] ISO/IEC 29173-1, *Information technology — Mobile item identification and management — Part 1: Mobile RFID interrogator device protocol for ISO/IEC 18000-6 Type C*

[11] ISO/IEC 29174 (all parts), *Information technology — UII scheme and encoding format for Mobile AIDC services*

[12] ISO/IEC 29175, *Information technology — Mobile item identification and management — User data for Mobile AIDC services*

[13] ISO/IEC 29176, *Information technology — Mobile item identification and management — Consumer privacy-protection protocol for Mobile RFID services*

[14] ISO/IEC 29177, *Information technology — Mobile item identification and management — Object Directory Service for Mobile AIDC services*

ICS 35.040

Price based on 18 pages