

GREATER CAIRO METRO UPGRADE LINE 1&2


CONTRACT 73/METRO
LoC: TFCIMPO161404556

Unified Interface Definition Interface Control Document (ICD)

CDRL N°63217055-997

Prepared for:
NATIONAL AUTHORITY FOR TUNNELS

Prepared by:
THALES Communications & Security

 THALES Communications & Security F0057	NUMERO DOCUMENT / DOCUMENT NUMBER	REVISION	FORMAT / SIZE	PAGE
	63369660-558	-B	A4	1/114

Document Management

TCS Revision Number	Date	Modification
--	07/11/2016	First issue
-A	15/06/2017	<ul style="list-style-type: none"> Update following NAT review. Refer to comment sheet in appendix F. Renaming the namespaces : "www.thales.cairo" into www.cairo.com/cairo Adding precisions in paragraph 3.3 UNIFIED INTERFACE CONFIGURATION Removing agent file following review of CBO SSS by NAT and corresponding web service §C7, update of alarm codes.
-B	12/12/2017	Update following NAT review and workshop on open points held on 21/11/17/ Refer to comment sheet in appendix F.

Glossary

List of Abbreviations	
AFC	Automatic Fare Collection
AG	Automatic Gate
API	Application Programmable Interface
AR	Anomaly Report
AR	Audit Register
ASN.1	Abstract Syntax Notation One
AV2 SAM	Ticketing Security Access Module
BO	Back Office
BR	Business Rule
BSN	Blocking Sequence Number
CA	Certification Authority
CBO	Central Back Office
CCB	Configuration Control Board
CCU	Central Control Unit
CDRL	Contractual Data Requirement List
CI	Check-In (so-called Tag-On)
CI	Card Issuer
CIPS	Card Initialization and Personalization System
CIS	Card Initialization System
CO	Clearing Operator
CO	Check-Out (so-called Tag-Off)
CP	Card Producer
CPS	Central Processing System
CPU	Central Processor Unit
CR	Change Request
CRL	Certificate Revocation List
CSC	Contactless Smart Card
CSC-A	Anonymous CSC
CSC-P	Personalized CSC
CSV	Comma Separated Value
CT	Contactless Ticket
DES, 3DES	Data Encryption Standard
DF	Dedicated file
DIS	Detailed Interface Document
DN	Distinguished name
DPS	Depot Processing System
DR	Delivery Review
E2E	End To End
EC	Electrical Cabinet
ECR	Engineering Change Request
EF	Elementary File
EFT	Electronic Fund Transfer
EMI	Electro-Magnetic Interference
EMC	Electro-Magnetic Compatibility
EOD	Equipment Operating Data
EPROM	Erasable-Programmable-Read-Only Memory
ERP	Enterprise resource planner
ETL	Extraction Transformation and Loading
FAT	Factory Acceptance Test
FED	Front-End Device

List of Abbreviations

FMCS	Field Maintenance Computer System
FO	Front Office
FT	Fait Technique (Problem Report)
FTP	File Transfer Protocol
FTP	Foiled Twisted Pair
GIS	Global Interface Specification
GSM	Global System for Mobile communication
GUI	Graphical User Interface
HRS	Hardware requirement Specification
HSM	Hardware Security Module
IADT	Inspection, Analysis, Demonstration, Test
ICD	Interface Control Document
IDD	Interface Data Document
IFM, IFMS	Integrated Fare Management System (Level 0 to 4)
IFS	Integrated Fare System (Level 0 to 3)
ITC	Ignore Time Check
ITF	Integrated Test Facility
IVVQ	Integration, Verification, Validation & Qualification
IVVQ-E	Integration, Verification, Validation & Qualification of Equipment/Subsystem
IVVQ-S	Integration, Verification, Validation & Qualification of System
Kab	3DES R/W keys used for mutual authentication of R/W vs. the controller part of equipment
KCK	3DES Infrastructure Key used as transportation key
KEK	3DES Key Encryption Key used as transportation key
Keq, K'eq	RSA public and private key pair
KMS	Key Management System
KPI	Key Progress Indicator
Ksession	Key created during a communication session
KTR	3DES Transport Key used as a transportation key
LAN	Local Area Network
LCU	Line Control Unit
LDAP	Lightweight Directory Access Control
LREF	List of Reference
LRU	Line Replaceable Unit
MAC	Message Authentication Code
MD5	Message-Digest version 5
MF	Master File
MMI	Man Machine Interface
MCBF	Mean Cycles Between Failures
MSN	Multi-Services Network
MT	Magnetic Ticket
MTBF	Mean Time Between Failures
MTTR	Mean Time To Repair
NOC	Network Operational Centre
NTP	Notice To Proceed
NTP	Network Time Protocol
OTP	One time programming
PAT	Partial Acceptance Test
PCM	Portable Checking Machine
PCR	Problem/Change Report
P-CSC	Personalized Card
PER	Packed Encoding Rule
PIN	Personal Identification Number

List of Abbreviations

PKCS	Public Key Cryptographic Standard
PKI	Public Key Infrastructure
PKI SAM	Public Key Infrastructure Security Access Module
PM	Project Manager
POS	Point Of Sales
POST	Point Of Sale Terminal
PROM	Programmable-Read-Only Memory
PSTN	Public Switched Telephone Network
PTO	Public Transport Operator
PVU	Portable Verifying Unit
QR	Quality Review
R/W	Reader/Writer
RA	Registration Authority
RAM	Random Access Memory
RAMS	Reliability, Availability, Maintainability and Safety
RFU	Reserved for future use
RLAN	Remote Local Area Network
ROM	Read-Only Memory
RSA	Asymmetric Encryption algorithm
RSS	Remote Station Server
SAM	Security Access Module
SAPP	System Assurance Program Plan
SAT	Site Acceptance Tests
SCR	Station Controller Room
SCU	Station Control Unit
SDR	System Design Review
SF	Station Failure
SFTP	Secure File Transfer Protocol
SOAP	Simple Object Access Protocol
SP	Service Provider
SPS	Station Processing System (or call also SCU)
SQL	Structured Query Language
SL	Submittal List
SRR	System Requirements Review
SRT	Single Ride Ticket
SSDD	System/Subsystem Design Document
SSS	System/Subsystem Specification
STP	Shielded Twisted Pair
SVT	Stored Value Ticket
TBD	To Be Defined
TBC	To Be Confirmed
TCF	Ticket Checking File
TCP/IP	Transmission Control Protocol / Internet Protocol
TLS	Transport Layer Security
TOM	Ticket Office Machine
TP	Ticket Processing
TPB	Test Procedures Book
T-Purse	Transport Purse linked to a Transport Application on a contactless card
TRB	Test Results Book
TRR	Test Readiness Review
TSN	Transaction Sequence Number
TVM	Ticket Vending Machine

List of Abbreviations

UD	Usage Data
UIA	Unified Interface Adapter
UPS	Uninterruptible Power System
USN	Unblocking Sequence Number
UTP	Unshielded Twisted Pair
VAL	Validator
VCF	Validator Concentration Function
VDU	Visual Display Unit
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
XDR	External Data Representation
XML	Extensible Markup Language
xPS	Means CPS or SPS or DPS
WAN	Wide Area Network
WSDL	Web Services Description Language

CONFIDENTIAL

TABLE OF CONTENT

1. INTRODUCTION	10
1.1 IDENTIFICATION	11
1.2 DOCUMENT OVERVIEW	12
1.2.1 PROJECT BACKGROUND	12
1.2.2 DOCUMENT STRUCTURE	12
1.3 OVERVIEW	13
1.3.1 CONTEXT	13
1.4 MIGRATION CONSTRAINTS	15
2. REFERENCE DOCUMENTS	16
3. INTERFACE SPECIFICATIONS	17
3.1 UNIFIED INTERFACE OVERVIEW	18
3.2 UNIFIED INTERFACE SECURITY	19
3.2.1 CERTIFICATE CREATION AND DISTRIBUTION	20
3.2.2 CONSTRAINTS ON CERTIFICATES	21
3.2.3 CERTIFICATE REVOCATION	23
3.3 UNIFIED INTERFACE CONFIGURATION	24
3.4 BATCH INTERFACE FROM CBO	28
3.4.1 FILE TRANSFER DESCRIPTION	28
3.4.2 PARAMETER FILE VERSIONNING	31
3.4.3 PARAMETER FILES	31
3.5 BATCH INTERFACE TO CBO	50
3.5.1 FILE TRANSFER DESCRIPTION	50
3.5.2 EVENT DATA FILES	50
3.5.3 EQUIPMENT LIST FILE	66
3.6 ONLINE WEB SERVICES	68
3.6.1 WEB SERVICE TECHNICAL DESCRIPTION	68
3.6.2 CUSTOMER DATA	69
3.6.3 CARD DATA	75
3.6.4 CLAIM DATA	80
3.6.5 AV2 SAM QUOTA	83
3.7 TIME SYNCHRONIZATION	85
APPENDIX A: CARD DATA STRUCTURES	86
A.1 DATA TYPES	86
A.2 CARD HEADER	87
A.3 TPURSECONTEXT	88
A.4 APPCONTEXT	88
A.5 CONTRACTLISTS	88
A.6 CONTRACT	89
A.7 EVENTLOG	91
A.8 ENVIRONMENT	92
A.9 TPURSELOG	93
A.10 SPECIALEVENTLOG	93
A.11 PREVIOUSSEVENTLOG	94
A.12 CARD HOLDER	95
APPENDIX B: CLARIFICATIONS ON XML DATA	96
APPENDIX C: INTERFACE CODES	97
C.1 EQUIPMENT TYPE CODE	97
C.2 ACCOUNTING OPERATION TYPE CODE	97
C.3 VALIDATION TYPE CODE	97
C.4 PENALTY TYPE CODE	98

C.5 PAYMENT MEANS	98
C.6 AGENT PROFILES	98
C.7 ALARM CODIFICATION	99
C.8 BLACKLISTING/BLOCKING REASON	100
C.9 SERVICE PROVIDER	100
C.10 CUSTOMER PROFILES	100
C.11 VALIDATION ERRORS	101
APPENDIX D: SPECIFIC CODES FOR ONLINE WEB SERVICES	102
D.1 WEB SERVICE RESPONSE CODES	102
D.2 OPERATION TYPE CODE	102
D.3 CARD DISABLING REASON CODES	103
D.4 CUSTOMER IDENTIFICATION TYPE	103
D.5 CLAIM STATE	103
D.6 SERVER ERRORS	103
D.7 AV2 SAM WEB SERVICE RESPONSE CODE	104
APPENDIX E: XSD & WSDL	105
E.1 CBO BATCH INTERFACE	105
E.2 ONLINE WEB SERVICES	105
E.3 FILE SIGNATURE SCHEMA	105
APPENDIX F: COMMENT SPREAD SHEET	106

CONFIDENTIAL

TABLE OF FIGURES

Figure 1 :	Project documents	11
Figure 2 :	Overview of interoperability	13
Figure 3 :	Interface compatibility overview	15
Figure 4 :	Unified interface overview	18
Figure 5 :	Unified interface security	19
Figure 6 :	Certificate creation and distribution	20
Figure 7 :	Certificate revocation list	23
Figure 8 :	Time synchronization.....	85

CONFIDENTIAL

1. INTRODUCTION

CONFIDENTIAL

1.1 IDENTIFICATION

This document is the specification of the Unified interface for the Greater Cairo Metro AFC system.

This interface is dedicated to the connection of vendor systems to the Central Back Office (CBO). It participates in the interoperability of the system.

The figure below describes the organization of the project documents (refer to CDRL for details).

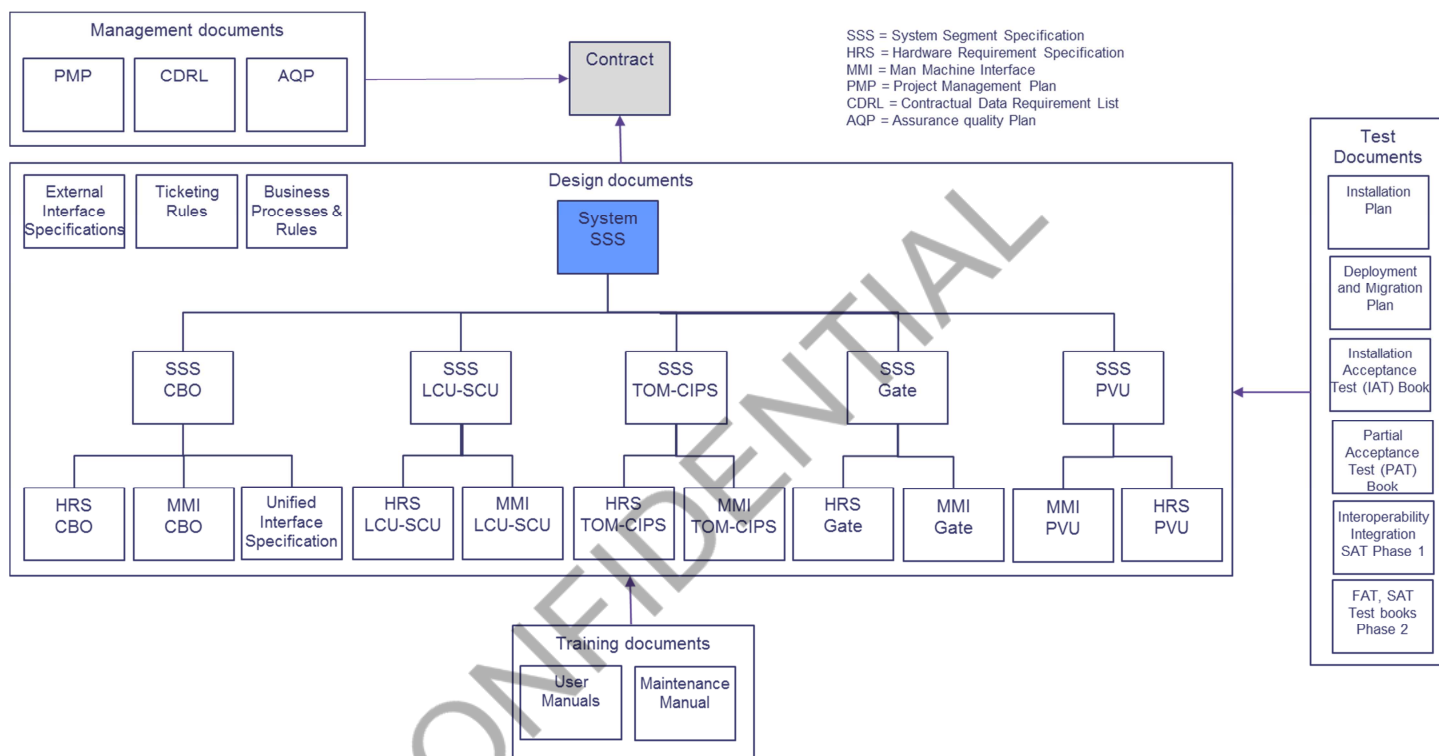


Figure 1 : Project documents

1.2 DOCUMENT OVERVIEW

1.2.1 PROJECT BACKGROUND

A contract has been signed between The National Authority for Tunnels (NAT) and Thales Communications a Security SAS (TCS) for the upgrade of the AFC system for metro lines 1 & 2 (850 gates project).

The project includes the delivery of front-end equipment, Automatic Gates and Ticket Office Machines (TOM) as well as Station Control Units (SCU) and Line Control Units (LCU) for the 2 lines. The objective is the replacement of the old magnetic devices by dual (magnetic / contactless) devices compatible with the dual devices of line 3 and the contactless devices of lines 1&2.

A new equipment type, the Portable Verifying Unit (PVU), dedicated to the roving inspectors will also be delivered.

In addition, Thales will provide a Central Back Office (CBO) that will house the central AFC functions. This 'level 4' system will manage the 3 lines and is connected to Thales and 3rd party systems (Level 3-1 systems: LCU, SCU and equipment).

In order to guarantee the interoperability between the level 4 and level 3-1 systems, this document specifies the Unified interface with the CBO.

1.2.2 DOCUMENT STRUCTURE

This document is structured as follows:

Chapter 1 (Introduction, this chapter) starts by highlighting the purpose and the structure of the document. It then gives an overview of the Unified Interface.

Chapter 2 (References) lists the associated documents.

Chapter 3 (Interface specifications), details the interface specifications.

1.3 OVERVIEW

1.3.1 CONTEXT

The following figure illustrates the interoperability model put in place for Cairo metro and shows the position of the Unified Interface in the global system.

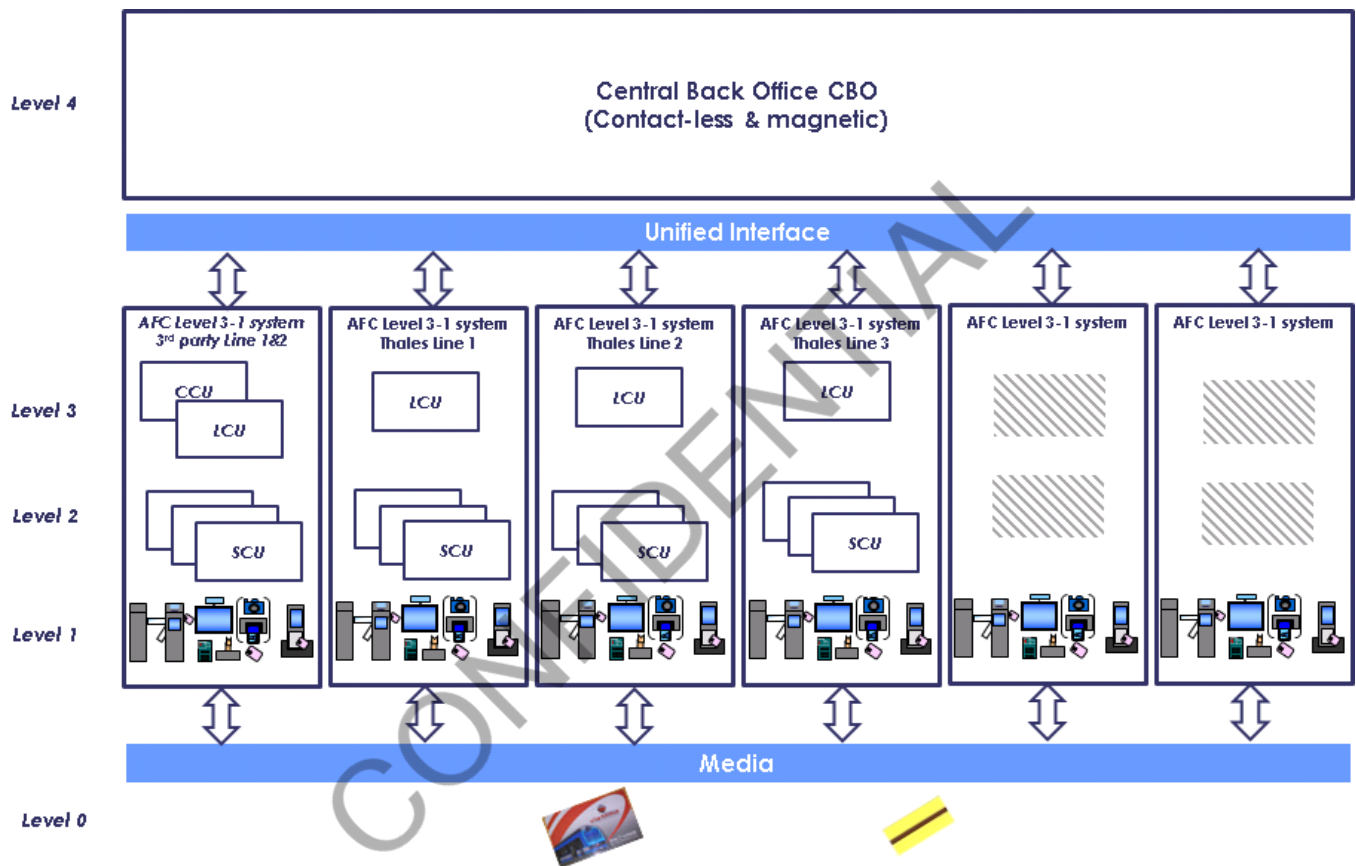


Figure 2 : **Overview of interoperability**

Note: the 'levels' mentioned on the left side are commonly used for describing AFC systems:

- Level 0 is the media.
- Level 1 is the front-end equipment.
- Level 2 is the local concentrator managing the communication with the equipment as well as local functions.
- Level 3 is a departmental back-office ensuring the operation of a subset of the AFC (line, transport service, retail network ...).
- Level 4 is the Central Back Office (CBO) system. The CBO system connects the different Level 3 -1 systems and implements the global functions that ensure the interoperability such as the media and customer management.

The interoperability of the system architecture is based on the blue interfaces:

- The Media.
- The Unified Interface to the CBO.

The unified interface defines the standard of communication between the Level 3-1 systems and the Level 4. The reference of this standard is the unified interface specification (the present document). The Unified Interface includes the following items:

- Interface security
- Configuration
- Batch interface from CBO
 - Business parameters
 - Blacklists
- Batch interface to CBO
 - Transactions and events: accounting, validations, alarms, shifts, stocks, ...
 - Device registration
- On line web services for front end devices
- Time synchronization

Any vendor can provide a Level 3 to 1 system and join the interoperable scheme assuming it complies with the media and unified interfaces.

The documents supporting the interoperability include:

- The media documents.
 - CSC: fare media layout, AV2 SAM specification
 - MT: fare media layout
- The unified interface specification
- The system end to end use case document that allow a global understanding of the system operation and the role of each subsystem.

CONFIDENTIAL

1.4 MIGRATION CONSTRAINTS

The CBO will replace the CCU system in phase 2 of the project. The key drivers for the deployment are:

- Ensure the continuity of the service through a strategy allowing the parallel usage of the existing and the new system.
- Limit the un-availability of the infrastructures.
- Facilitate the end-customer experience during the transition phase
- Assist the personnel during the transition phase.

The following figure illustrates the migration constraints: even if there will be additional data managed by the Unified Interface, the existing CCU elements shall be managed and remained untouched in order to ensure the ascending compatibility, especially for the equipment.

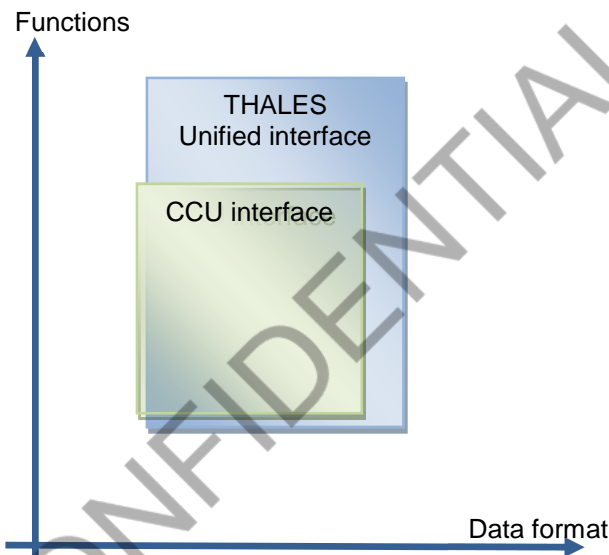


Figure 3 : **Interface compatibility overview**

Note: compatibility with existing CCU interface is not 100% and there might be minor impact on existing equipment. A typical example is the name space in the XML schema that is updated.

2. REFERENCE DOCUMENTS

Refer to (CDRL) N°63217055-997.

CONFIDENTIAL

3. INTERFACE SPECIFICATIONS

Note that in the various tables proposed in the following of this chapter, the default cardinality of 1 when not overridden means that the field is mandatory.

CONFIDENTIAL

3.1 UNIFIED INTERFACE OVERVIEW

The Unified Interface shall guarantee the Central Back Office interoperability with any vendor. The following figure gives an overview of the interface.

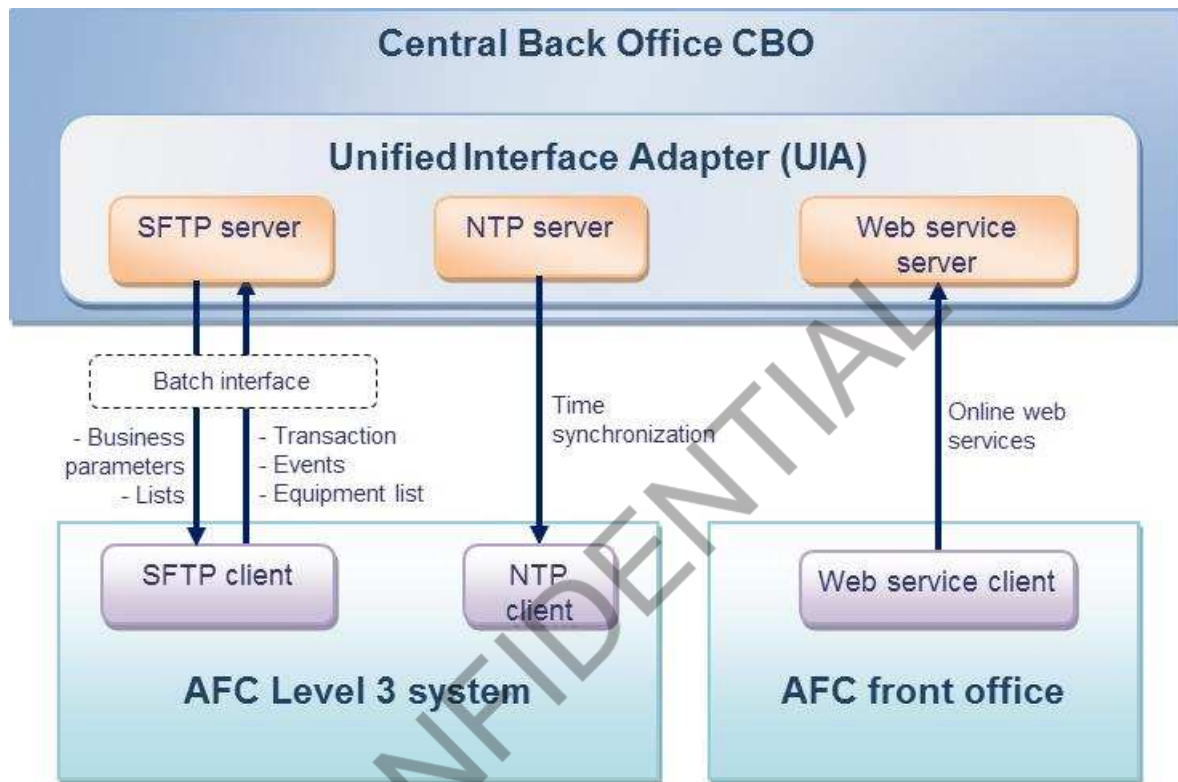


Figure 4 : Unified interface overview

Interface	Description	Protocols
CBO -> Level3 Batch interface from CBO	The CBO provides the configuration files to the Level3 systems. It includes the following files: - Business parameters - Blacklists	SFTP
Level3 -> CBO Batch interface to CBO	The Level3 system provides the event files and device configuration to the CBO. It includes the following files: - Transactions and events: accounting, validations, alarms, shifts, stocks ... - Equipment list (device registration)	SFTP
CBO -> Front office CBO On line web services	The CBO provides web services for front office. The web service client can consult some central back office data and modify or create such data.	SOAP over HTTPs
CBO -> Level3 Time synchronization	The Level3 systems are synchronized with the CBO.	NTP

3.2 UNIFIED INTERFACE SECURITY

The Unified Interface allows connecting AFC vendors to the Central back office. This interface is secured in such manner that new vendors can be added to this interface during the project lifecycle. There are two parts in this interface. Both of them rely on authentication of the client/server sides with X509 certificates.

First, the batch interface allows exchanging files over SFTP. Its security relies on the signature of the exchanged files with the X509 certificates (Refer to §3.4.1.3). It guarantees the file authentication. In addition, of course, the SFTP server uses cryptographic protocol and its access is protected by login/password (Refer to §3.4.1.2).

Second, the online web services security is based upon mutual authentication over HTTPs. This security is based on certificates that allow the mutual identification of the client and the server. Data are encrypted thus guaranteeing message confidentiality, authentication, and non-repudiation.

The next paragraphs describe the certificate and key distribution process as well as the certificate revocation process.

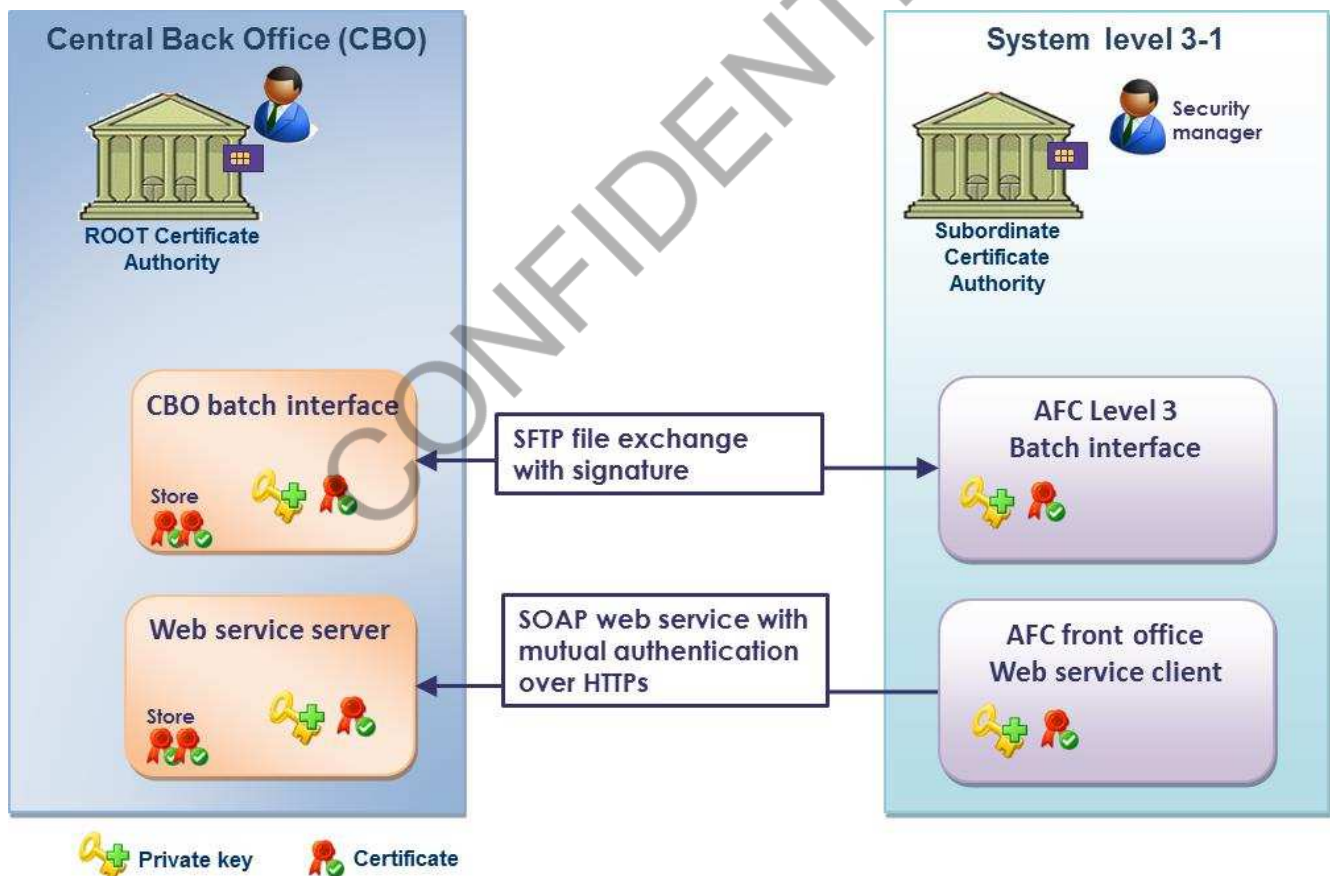


Figure 5 : Unified interface security

Ticketing system security

The security scope addressed here is limited to the CBO interface. The internal security of the level 3-1 system (e.g. equipment to level 2) is under the responsibility of each vendor and the implementation may be specific.

3.2.1 CERTIFICATE CREATION AND DISTRIBUTION

The Root Certificate Authority is part of the Central back office. It is physically isolated from the AFC network and is stored in a secured place. It is responsible of the signature of the certificates for the subordinates CA and of the generation of the certificates for the CBO servers.

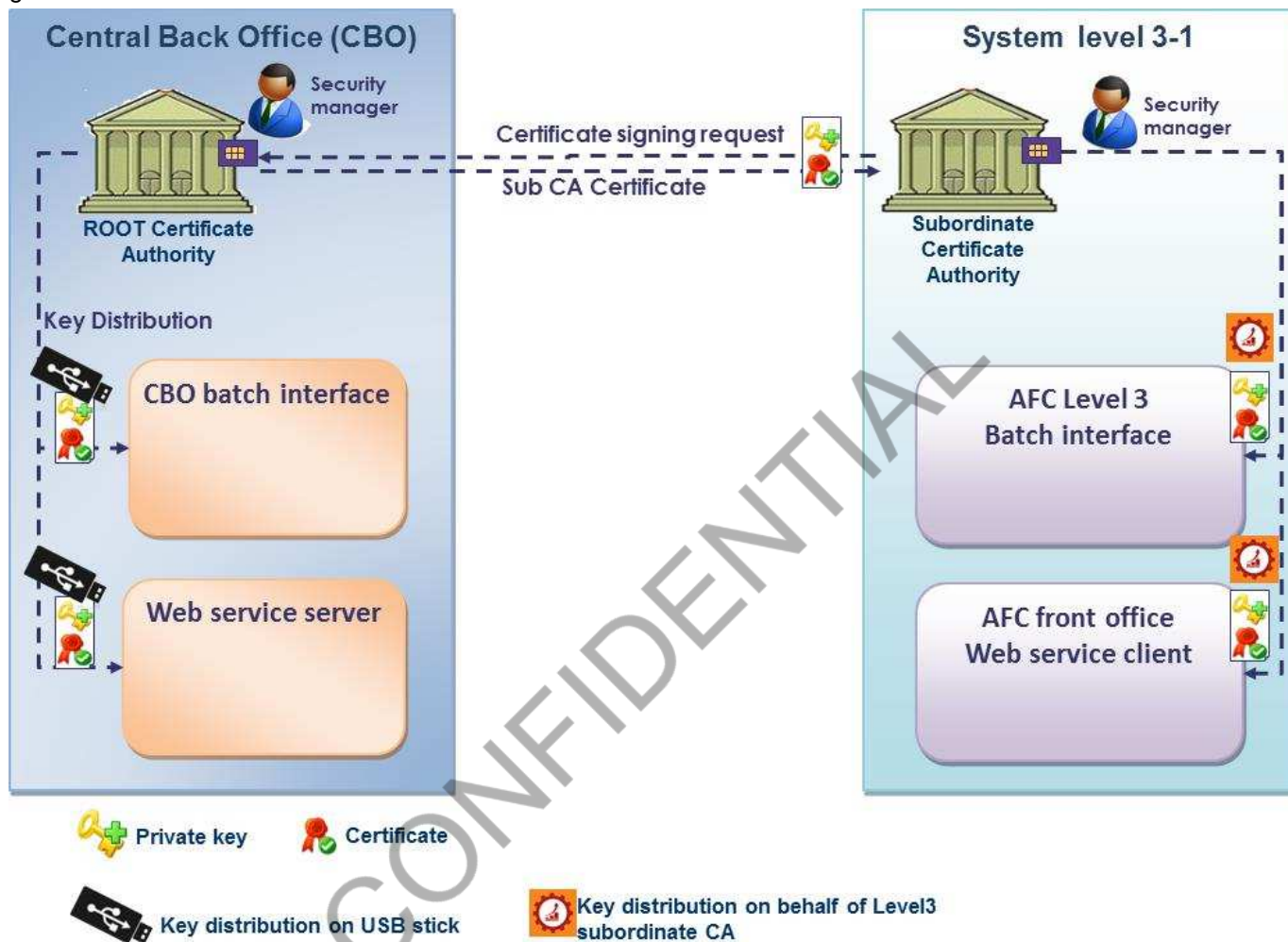


Figure 6 : Certificate creation and distribution

CBO application server certificates

The CBO security manager uses the ROOT CA in order to generate the keys and certificate for each CBO application server. The keys and certificates are exported into a **PKCS#12 V1.0** file. The file is protected by a password and is installed on the CBO server manually during installation process. These steps will be described in installation manual.

Subordinate CA certificates

For each Level3-1 system connected to the CBO, a subordinate Certificate Authority is created. It is in charge of the certificate distribution and signature for the Level3-1 components that are connected to the CBO. It is under the responsibility of the system Level3-1.

The subordinate CA needs to be added into the CBO certification hierarchy. For this purpose, the certificate manager of the Level3-1 system sends a **certificate signing request (PKCS#10 V1.7)** to the CBO security manager. The request subject shall conform to the rule defined for certificates: refer to §3.2.2.

After checking that the request is legitimate, the certificate is signed and provided back to the subordinate CA via a manual procedure.

Note: the automatic distribution of certificates and key pairs requires a trusted relationship between the client and the CA and an enrolment process. A manual process is much simpler

Note: At present installation of certificates is manual on the TOMs. In phase 1 of the project this will continue like this with the CCU certificate. For the CBO deployment, the CCU certificate will still be used in order to smooth the process. For the TOMs that will be installed later, the certificate signature may be automated by including the TOMs in the LCU domain. However, Thales has to check the feasibility. The gates do not access the CBO customer and card services and hence do not need certificate.

3.2.2 CONSTRAINTS ON CERTIFICATES

In order to guarantee the homogeneity of certificate declaration, the certificates must conform to following rules.

Certificate format: x509 V3

Key pair algorithm: RSA PKCS#1

Key length: RSA 1024 bits (minimum) / RSA 2048 bits (recommended)

Signature algorithm: sha1WithRSAEncryption

Subject (distinguished name) of the certificate

The subject field of the certificate must contain:

C=Country, O=Organization, OU=Organizational Unit, CN=Common name, SERIALNUMBER=Serial number

With:

Field	Content
Country (C)	EG (The two-letter ISO code for the country where the organization is located)
Organization (O)	CMO
Organizational Unit (OU)	AFC
Common name (CN)	<p>The value depends on the type of certificate:</p> <ul style="list-style-type: none"> - Root CA: CAIRO CBO Root CA - Subordinate CA CAIRO <system name> Subordinate CA - Front office: CAIRO <system name>_<eqt_model_ID><eqt_model_name>#<eqt_ID> <p>Where:</p> <ul style="list-style-type: none"> - <system name> is the System level3 name as declared in CBO configuration (Refer to §3.4.1.2.1) or an external system name - <eqt_model_ID> is the equipment model ID - <eqt_model_name> is the equipment model name - <eqt_ID> is the equipment serial number

Field	Content
SERIALNUMBER	The value depends on the type of certificate: - Root CA: null - Subordinate CA: null - Front office: <eqt_model_ID>-<eqt_ID>

Examples of subject field:

- C=EG, O=CMO, OU=AFC, CN=CAIRO CBO Root CA
- C=EG, O=CMO, OU=AFC, CN=CAIRO LCUL1Thales Subordinate CA
- C=EG, O=CMO, OU=AFC, CN= LCUL1Thales_10AG#1234567890, SERIALNUMBER=10-1234567890

Issuer of the certificate

The Issuer field of the certificate must contain:

C=Country, O=Organization, OU=Organizational Unit, CN=Common name

With:

Field	Content
Country (C)	EG (The two-letter ISO code for the country where the organization is located)
Organization (O)	CMO
Organizational Unit (OU)	AFC
Common name (CN)	The value depends on the type of certificate: - Root CA: CAIRO CBO Root CA - Subordinate CA CAIRO <system name> Subordinate CA

Examples of Issuer field:

- C=EG, O=CMO, OU=AFC, CN=CAIRO CBO Root CA
- C=EG, O=CMO, OU=AFC, CN=CAIRO LCUL1Thales Subordinate CA

Certificate validity duration

The X509 certificates have validity duration that is fixed at certificate creation time. In accordance with security state of the art, the validity duration is:

Certificate	Validity
ROOT CA	25 years
Subordinate CA	10 years
CBO server certificate	5 years
Component connected to CBO	5 years

Certificate renewal

When the certificates expires, it shall be replaced in the same way as for its initial creation (refer to paragraph above). Of course the replacement shall be anticipated in order to guarantee service continuity. Thus the request must be prepared and further processed at least 3 months before the expiration date.

3.2.3 CERTIFICATE REVOCATION

If a certificate is compromised, it shall be revoked in the system and denied access to the CBO. In such a case, the security manager adds the compromised certificate in the revocation list on the CBO application servers. This is the responsibility of System level3-1 to provide the file of the compromised certificate.

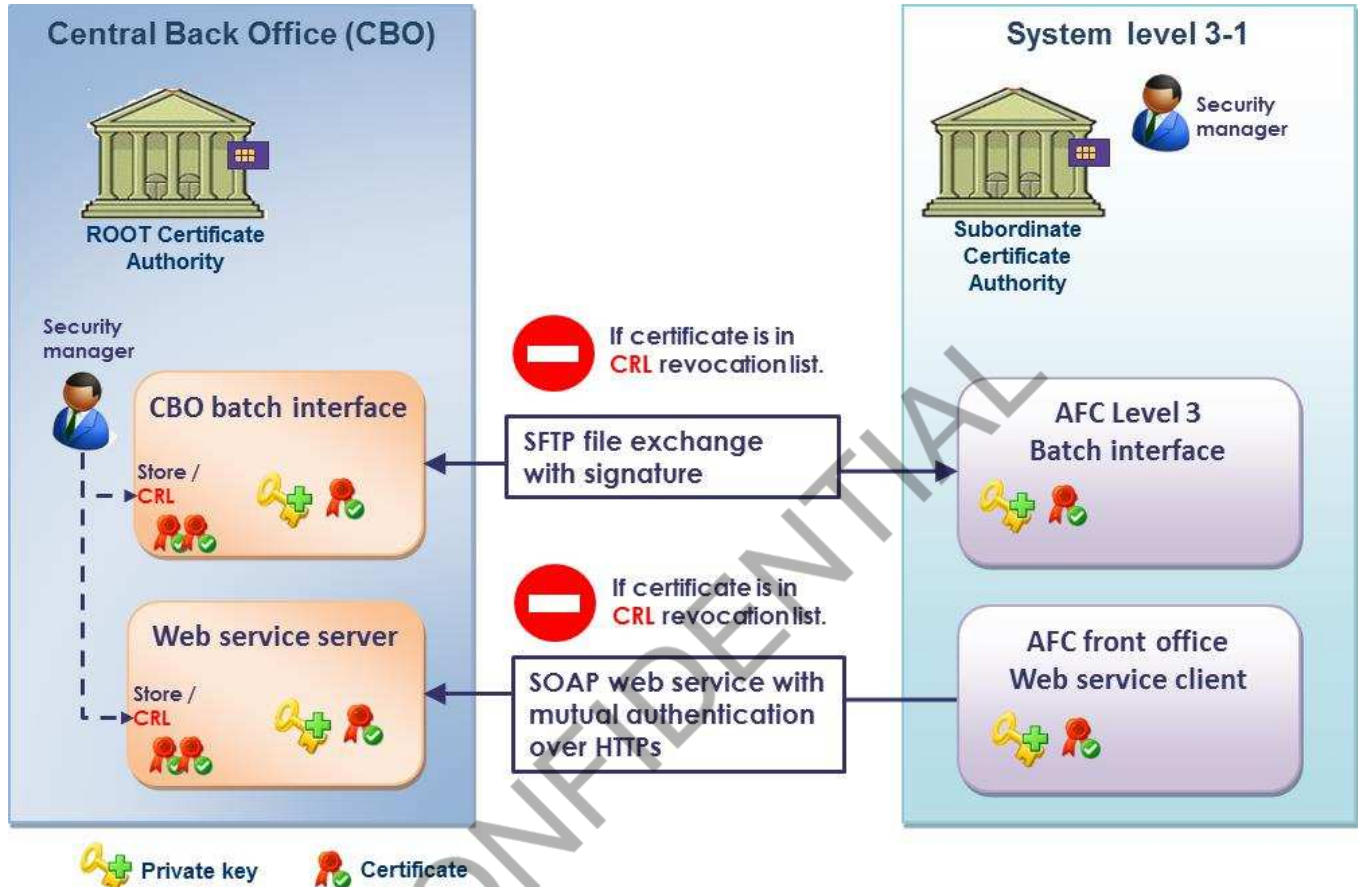


Figure 7 : Certificate revocation list

3.3 UNIFIED INTERFACE CONFIGURATION

The unified interface can be extended with new connected systems and with new interface data codes. This paragraph details the configuration possibilities.

Adding a new System Level3-1

If a new system is added to the AFC network, there are two steps for the configuration:

- The batch interface is configured with new connected system (Refer to details in §3.4.1.2 for configuration file). The Unified Interface Adaptor component is restarted, thus creating dedicated directories for new System Level3-1 on the SFTP server. The SFTP login/password is provided to the Level3-1 system.
- A certificate is delivered to the Level3-1 subordinate CA (refer to §3.2). Thus the components that have a certificate signed by the subordinate CA will be provided access to the Unified interface.

Adding a new Service provider

The service providers are configured at a CBO repository. The configuration file contains following information:

- BusinessEntityID (Code)
- Name
- Abbreviation
- BusinessEntityType: TRANSPORT_OPERATOR, TRANSPORT_AUTHORITY, SYSTEM_PARTICIPANT

For initial list of service providers, refer to appendix C.9.

The modification of the business entities requires calling a web service or restarting the component. For illustration purpose, here is below an example of file.

```
[
{
  "businessEntityId": 1,
  "name": "CAIRO Metro",
  "abbreviation": "CM1",
  "businessEntityType": "TRANSPORT_OPERATOR",
  "businessRoleTypes": []
},
{
  "businessEntityId": 2,
  "name": "Business Entity 2",
  "abbreviation": "BE_2",
  "businessEntityType": "TRANSPORT_OPERATOR",
  "businessRoleTypes": []
},
{
  "businessEntityId": 3,
  "name": "Business Entity 3",
  "abbreviation": "BE_3",
  "businessEntityType": "TRANSPORT_OPERATOR",
  "businessRoleTypes": []
},
{
  "businessEntityId": 100,
```



```

"name": "Business Entity 100",
"abbreviation": "BE_100",
"businessEntityType": "TRANSPORT_AUTHORITY",
"businessRoleTypes": []
},
{
"businessEntityId": 500,
"name": "CardIssuer (BE 500)",
"abbreviation": "BE_500",
"businessEntityType": "SYSTEM_PARTICIPANT",
}
]

```

Adding a new equipment type

The equipment types are configured at CBO repository. The configuration file contains following information:

- deviceTypeId: Value & Label
- accessControlGroup: GATE_DEVICES, PVU_DEVICES (Portable Verifying Unit), POST_DEVICES (Point Of Sale Terminal), VAL_DEVICES (Validation device), TVM_DEVICES (Ticket Vending Machine), DEVICES (other devices)
- deviceConnectivity: ONLINE/OFFLINE
- fixedDevice: true/false (indicates if device is onboard vehicle or fixed on the ground)

For initial list of equipment types, refer to appendix C.1.

The modification of the equipment types requires calling a web service or restarting the component. For illustration purpose, here is below an example of file.

```

[
{
"deviceTypeId": {"value": 1, "label": "AG"},
"accessControlGroup": "GATE_DEVICES",
"deviceConnectivity": "ONLINE",
"fixedDevice": true
},
{
"deviceTypeId": {"value": 2, "label": "TOM"},
"accessControlGroup": "POST_DEVICES",
"deviceConnectivity": "ONLINE",
"fixedDevice": false
},
{
"deviceTypeId": {"value": 4, "label": "SCU"},
"accessControlGroup": "DEVICES",
"deviceConnectivity": "OFFLINE",
"fixedDevice": true
},
{
"deviceTypeId": {"value": 5, "label": "PVU"},

```

```
"accessControlGroup": "PVU_DEVICES",
"deviceConnectivity": "OFFLINE",
"fixedDevice": false
},
{
"deviceTypeId": {"value": 6, "label": "CIS"},
"accessControlGroup": "DEVICES",
"deviceConnectivity": "OFFLINE",
"fixedDevice": true
},
{
"deviceTypeId": {"value": 10, "label": "AG2"},
"accessControlGroup": "GATE_DEVICES",
"deviceConnectivity": "OFFLINE",
"fixedDevice": true
},
{
"deviceTypeId": {"value": 12, "label": "TOM"},
"accessControlGroup": "POST_DEVICES",
"deviceConnectivity": "OFFLINE",
"fixedDevice": true
}
}
```

In addition the dictionary file shall be edited for translation: dictionary_en.properties:

```
DeviceType.1=Automatic Gate
DeviceType.2=Ticket Office Machine
DeviceType.4=Station Control Unit
DeviceType.5=Portable Verifying Unit
DeviceType.6=Card Initialization System
DeviceType.10=Dual Automatic Gate
DeviceType.12=Ticket Office Machine
```

Adding a new penalty type

The penalty type is considered as simple string and, whatever the value, it is accepted and doesn't require any configuration.

Adding a new blacklisting/blocking reason

The blacklisting/blocking reasons are configured at CBO repository. . The configuration file contains following information:

- denyReason: Code and comment
- isInteractive: always true

For initial list of blocking reason, refer to appendix C.8.

The modification of the equipment types requires calling a web service or restarting the component. For illustration purpose, here is below an example of file.

```
[
```

```
{
  "denyReason": {"code": "1", "comment": "Product abuse"},
  "isInteractive": true
},
{
  "denyReason": {"code": "2", "comment": "Card Lost"},
  "isInteractive": false
},
{
  "denyReason": {"code": "3", "comment": "Card Stolen"},
  "isInteractive": false
},
{
  "denyReason": {"code": "4", "comment": "Damaged"},
  "isInteractive": false
},
{
  "denyReason": {"code": "5", "comment": "Faulty"},
  "isInteractive": false
}
}
```

In addition the dictionary file shall be edited for translation: dictionary_en.properties :

```
DenyReasonCode.1=Product abuse
DenyReasonCode.2=Card Lost
DenyReasonCode.3=Card Stolen
DenyReasonCode.4=Damaged
DenyReasonCode.5=Faulty
```

Adding a new Customer profile

The customer profiles are configured in CSC fare parameter file (Refer to §3.4.3.3). For initial list, refer to appendix C.10.

3.4 BATCH INTERFACE FROM CBO

The CBO system is responsible for level4 functionalities that are global to the AFC system. The result of these functions is materialized into files that are forwarded to the Level3 systems in order to manage the behavior of the front end devices.

These files can be used by any vendor who wants to integrate into the AFC system.

It includes:

- **The business parameters** that allow the fare computations and the management of fare products:
 - Topology parameters
 - Time parameters
 - CSC fare parameters
 - MT fare parameters
- **Card blacklist** that allows blocking card that is considered fraudulent in the system.
- **SAM blacklist** that allows blocking a device for fare product sale.

3.4.1 FILE TRANSFER DESCRIPTION

3.4.1.1 Protocol for file transfer

The file transfer is performed via **SFTP** over TCP-IP.

The SFTP server is deployed on the CBO. The AFC Level-3 system connects to this server in order to get the parameter files.

The files are provided in **XML** format.

3.4.1.2 Configuration

The configuration of the interface is performed on both CBO and Level 3 system sides.

3.4.1.2.1 CBO configuration

Configuration for the list of connected Level3 Systems

The list of Level3 systems connected to the CBO shall be configurable. The configuration shall contain a name for each Level3 system. For instance:

```
LCUL1Thales
SystemL1Vendor2
LCUL2Thales
SystemL2Vendor2
LCUL3Thales
..
```

This name is referred to as **#Level3Name#** in the following paragraphs.

Sub FTP directories for Level3 systems

For each Level3 system name, the CBO shall create an FTP directory under the root directory with the corresponding name (**#Level3Name#**).

Directory hierarchy

The files for a Level3 system are placed under the parent folder called #Level3Name#.

SFTP directory	Description
\\#Level3Name#\data\	Contains the event data files uploaded from the Level3 system. There is one sub directory per day that contains the transaction files: <ul style="list-style-type: none"> \\#Level3Name#\data\yyyymmdd\
\\#Level3Name#\cfg\out\	Contains the parameter files generated by CBO for the Level3 system: <ul style="list-style-type: none"> \\#Level3Name#\cfg\out\businessParameters\x\fareTable.xml \\#Level3Name#\cfg\out\businessParameters\x\MTfareTable.xml \\#Level3Name#\cfg\out\businessParameters\x\timeTable.xml \\#Level3Name#\cfg\out\businessParameters\x\stationNetwork.xml \\#Level3Name#\cfg\out\blacklistCard\x\blackListCard.xml \\#Level3Name#\cfg\out\blacklistSAM\x\blackListSAM.xml <p>NB:</p> <ul style="list-style-type: none"> \x\ is a directory named with the version number of the file. Refer to §3.4.2. The business parameters (fareTable.xml, MTfareTable.xml, timeTable.xml, stationTable.xml) are generated together; they are placed in the same directory.
\\#Level3Name#\cfg\in\	Contains the parameter files generated by Level3 system and processed by the CBO. It contains the equipmentList: <ul style="list-style-type: none"> \\#Level3Name#\cfg\in\equipment\x\equipmentListYYMMDD.xml <p>NB:</p> <ul style="list-style-type: none"> \x\ is a directory named with the version number of the file. Refer to §3.4.2.

Adding new Level3 system

The Level3 system list configuration shall be taken into account at each server restart. If a new #Level3Name# is detected then the current files for that Level3 system are automatically generated in \\#Level3Name#\cfg\out\ directory (business parameter, lists).

Security (directory secured with login/password)

The directories \\#Level3Name#\ shall be secured with SFTP login/password with following access rights:

SFTP directory	Access for Level3 system login/password
\\#Level3Name#\data\	<ul style="list-style-type: none"> READ/WRITE
\\#Level3Name#\cfg\out\	<ul style="list-style-type: none"> READ
\\#Level3Name#\cfg\in\	<ul style="list-style-type: none"> READ/WRITE

3.4.1.2.2 Level3 system configuration

Directory configuration

The Level3 system shall be configured with FTP directories where to get and put the files:

- \#Level3Name#\data\
- \#Level3Name#\cfg\out\businessParameters\
- \#Level3Name#\cfg\out\user\
- \#Level3Name#\cfg\out\blacklistCard\
- \#Level3Name#\cfg\out\blacklistSAM\
- \#Level3Name#\cfg\in\equipment\

User password configuration

The Level3 system shall be configured with user/password for accessing the CBO SFTP server. The password storage shall be secured.

Parameter directory polling frequency

The CBO parameter files are placed in dedicated directories so that the Level3 system synchronizes the parameters. The Level3 system shall regularly check the SFTP directories in order to find out if new files need to be picked up. The polling frequency shall be configurable and set to 1 minute.

Event data file transfer frequency

The frequency of the event data file transfer shall be configurable. Files shall be transferred at least every 15 minutes.

NB: The front end devices shall also close the file for transfer at least every 15 minutes.

Event data file repositories

The Level3 system shall guarantee that all the interface files are sent to the CBO. The Level3 system shall keep a copy of the files transferred. The number of days kept shall be configurable at Level3 system.

3.4.1.3 File Security

Refer to §3.2 for the global security description.

The files shall be signed with X509 certificates. The certificates are valid and signed by AFC ROOT Certificate Authority for both CBO and Level3 systems.

CBO security

The CBO shall provide files signed with certificate validated by AFC Root Certificate authority. See Appendix E.3

The CBO shall check that the files provided by Level3 system are signed with a certificate validated by AFC Root Certificate authority.

Level3 system security

The Level3 system shall provide files signed with certificate validated by AFC Root Certificate authority. See Appendix E.3

The Level3 system shall check that the files provided by CBO are signed with a certificate validated by AFC Root Certificate authority.

3.4.1.4 Degraded modes

CBO degraded mode: Interruption during file generation

If CBO is stopped during file generation, it shall restore the situation at next startup: it checks the directory for each Level3 system and generates the missing files for the current parameter version.

Level3 system degraded mode: SFTP server not available

If CBO FTP server is not available, the Level3 system shall abort the process and do a retry after one minute. It shall guarantee that no file is lost.

3.4.2 PARAMETER FILE VERSIONNING

The parameter files shall be placed in a directory containing the version number. This version number is an integer incremented with each new version. For instance

- \#Level3Name#\cfg\out\businessParameters\1\fareTable.xml
- \#Level3Name#\cfg\out\businessParameters\2\fareTable.xml
- \#Level3Name#\cfg\out\businessParameters\3\fareTable.xml
- ...
- \#Level3Name#\cfg\out\businessParameters\122\fareTable.xml
- ...
- \#Level3Name#\cfg\out\businessParameters\n\fareTable.xml

3.4.3 PARAMETER FILES

The following paragraphs describe the format of the files generated by the CBO.

The files generated by the CBO contain first a header indicating:

- The file version
- The file generation date
- The file activation date, meaning when the file shall become active on front end devcies.

3.4.3.1 Topology file

FileName	stationNetwork.xml
Directory	\#Level3Name#\cfg\out\businessParameters\x\ <ul style="list-style-type: none"> • \x\ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	StationNetwork.xsd
Description	This file defines: <ul style="list-style-type: none"> • The lines of the transportation network. • The zones of the transportation network. • The stations of the transportation network. • The Inter station matrix that defines the number of stations between origin and destination stations. • The Zone to Zone matrix that defines the number of zones between origin and destination zones.

Field name	Type Cardinality (1 by default)	Description
stationNetwork		
head		Header
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version
activationDate	dateTime	Activation date of configuration version
lines		Network lines
linesNum	integer	Number of lines
line	<u>Cardinality:</u> 1..n	
lineCode	integer	Line identifier
lineNames		
lineName	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens. Len = 60 chars
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars
floatingZones		Floating zone. The fare calculation is based on the number of zones. The number of zones depends on the number of stations between origin and destination.
floatingZone	<u>Cardinality:</u> 1..n	
numberOfZones	integer	Number of zones that apply for fare calculation
numberOfStations	integer	Maximum number of inter-stations to be considered between travel origin and destination.
zones		Geographical zones
zone	<u>Cardinality:</u> 1..n	

Field name	Type Cardinality (1 by default)	Description
zoneCode	integer	Zone identifier
zoneNames		
zoneName	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens Len = 60 chars
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars
stations		Stations
station	<u>Cardinality:</u> 1..n	
stationId	integer	Station identifier
zoneCode	integer	Upper zone identifier that the station belongs to
boundaryZoneCode	integer <u>Cardinality:</u> 0..1	Boundary zone code. RFU (reserved for future use).
stationLines		The Lines that the station belongs to
stationLine	<u>Cardinality:</u> 1..n	
lineCod	integer	Line code
main	boolean	Indicates if this is the main line of the station.
position	integer	Position of this station in this line.
stationNames		
stationName	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens Len = 60 chars

Field name	Type Cardinality (1 by default)	Description
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars
interStationsMatrix		This matrix defines the number of inter-stations between two stations.
interStation	<u>Cardinality:</u> 1..n	
originStation	integer	Origin station
destStation	integer	Destination station
intStation	integer	Number of stations between origin station and destination station
zoneToZoneMatrix		This matrix defines the number of zones between two zones.
interZone		
originZone	integer	Origin zone
destZone	integer	Destination zone
numZones	integer	Number of zones between origin zone and destination zone
route		
zoneCode	Integer <u>Cardinality:</u> 1..n	Indicates the different zones between origin zone and destiny zone
Signature		See Appendix E.3

3.4.3.2 Time parameter file

FileName	timeTable.xml
Directory	\#Level3Name#\cfg\out\businessParameters\lx\ <ul style="list-style-type: none"> lx is a directory named with the version number of the file. Refer to §3.4.2.
XSD	TimeTable.xsd

Description	<p>This file defines:</p> <ul style="list-style-type: none"> The working days The special days (like holidays...) The time periods (Peak-off Hours, Normal Hours, Peak Hours....) for each day type. Time periods can be associated to different fares in the fare parameter file.
--------------------	---

Field name	Type / Cardinality (1 by default)	Description
timeTable		
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
days		
firstDay	integer	Code that define the first Day of the week. Note that in the configuration, this field holds the value 7 (Sunday) <ul style="list-style-type: none"> 1. Monday 2. Tuesday 3. Wednesday 4. Thursday 5. Friday 6. Saturday 7. Sunday
day	<u>Cardinality:</u> 7	
hhmm1	time	Time (hh:mm) defining the start of the working day. The seconds are always 00.
numOfMin	integer	Number of minutes of the working day from hhmm1
dayTypeeld	integer	Day type associated to this day
specialDays		
specilaDay	<u>Cardinality:</u> 1..n	
yyyymmdd	date	The date of the special day with the format yyyy/mm/dd
hhmm1	time	Time (hh:mm) defining the start of the day. The seconds are always 00.
numOfMin	integer	Number of minutes of the working day from hhmm1
dayTypeeld	integer	Day type associated to this day

Field name	Type Cardinality (1 by default)	Description
dayTypes		Defines the time periods for each day type.
dayType	<u>Cardinality:</u> 1..n	
dayTypeeld	integer	Unique identifier for this dayType
description	string	Description free field
slots		
slot	<u>Cardinality:</u> 1..n	Defines a time period
hhmmss	time	Time (hh:mm:ss) that defines the end of the period and the start of the following period. Note: the first period start at 00:00, and the last period finishes at 23:59:59.
periodType	integer	Defines the type of the period: <ul style="list-style-type: none"> • 0: Peak-off Hours. • 1: Normal Hours. • 2: Peak Hours. • 3: No working
Signature		See Appendix E.3

3.4.3.3 CSC Fare parameter file

FileName	fareTable.xml
Directory	\#Level3Name#\cfg\out\businessParameters\ <ul style="list-style-type: none"> • \ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	FareTable.xsd
Description	This file defines: <ul style="list-style-type: none"> • Global fare parameters • Customer profiles (adult, child, student...) • Fares applied for travels with TPurse • Fare reduction applied on time periods for travels with TPurse • Fare media (supports) • Fare tables for the period pass products • Families of fare products • Period pass fare products

Field name	Type Cardinality (1 by default)	Description
fareTable		

Field name	Type / Cardinality (1 by default)	Description
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
globalParameters		Global parameters
freeExitTime	integer	Time in seconds to allow exiting from the same in station without being charged.
penaltyAmount	integer	Amount to pay in case of penalty
maxGroupTime	integer	Maximum time (expressed in seconds) between two validations with the same media for a group travel.
maxGroupNumber	integer	Maximum number of passenger that can travel in group
maxJourneyDuration	integer	Maximum duration of a journey expressed in minutes
samQuotaThreshold	integer	Threshold under which the device shall request SAM quota update.
profiles		
profile	Cardinality: 1..n	
profileCode	integer	Profile code Following the EN1545 standard Codification. See Appendix C.10
names		
name	Cardinality: 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens Len = 60 chars
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars

Field name	Type / Cardinality (1 by default)	Description
tPurseFare		Fare applicable on T-Purse
tariffs		
tariff	<u>Cardinality:</u> 1..n	
numberOfZones	integer	Number of zones.
fullFare	integer	Price to be paid for the number of zones if the customer profile is not defined.
profiles	<u>Cardinality:</u> 0..1	
profile	<u>Cardinality:</u> 1..n	
profileCode	integer	Profile code Following the EN1545 standard Codification. See Appendix C.10
price	integer	Price to be paid for the profile and number of zones.
initialFee	integer	The Initial Fee is an amount withdrawn from the T-purse at the Check-in Validation.
maxTravelPrice	integer	In some abnormal check-in/check-out situations, the maximum travel price shall be deducted from the T-Purse. Fee to apply for expired travel time duration and for expired journey time duration.
minTravelPrice	integer	Price to be paid in case of degraded case fare mode (Entry/Exit Override mode, Incident mode)
maxValue	integer	Tpurse max amount value
minOperationalValue	integer	This is the minimum amount of Tpurse when a card is processed at sale device.
minUsageValue	integer	This is the minimum amount of Tpurse that shall remain on the card after the validation. For instance, at check-in, the amount of the Tpurse must be at least minUsageValue + initial fee.
maxReloadValue	integer	Tpurse maximum reload value
minReloadValue	integer	Tpurse minimum reload value
tPurseFareType	integer	Defines the price computation mode for the TPurse: <ul style="list-style-type: none"> 3: By Zones. 4: Floating zones
timeCharges		Time period fare reductions for travels with Tpurse.
peak_off	integer	Defines the percentage over the fare applicable in this time period.
normal	integer	Defines the percentage over the price applicable in this time period.
peak	integer	Defines the percentage over the price applicable in this time period.

Field name	Type / Cardinality (1 by default)	Description
faresTables		Different fares for the pass products
fare	<u>Cardinality:</u> 1..n	
faresTableCode	integer	Code that uniquely identifies the fare table. This fare table can be used for one or more passes.
tariffs		
tariff	<u>Cardinality:</u> 1..n	
numberOfZones	integer	Number of zones.
fullFare		Price to be paid if the customer profile is not defined.
profiles		
profile	<u>Cardinality:</u> 1..n	
profileCode	integer	Profile code Following the EN1545 standard Codification (see Appendix C.10)
price	integer	Price to be paid for the profile and number of zones.
cardsSupports		Fare media definition
cardSupport	<u>Cardinality:</u> 1..n	
ownCod	integer	Code of the card owner
cod	integer	Code of the fare media (starting from 1)
desc	string	Description
technology	integer	Card chip technology: <ul style="list-style-type: none"> 2: desFire
personalized	boolean	Indicates if the card is personalized: <ul style="list-style-type: none"> false: Not personalized. true: Personalized.
personalizationFee	integer	Fee in order to personalize this support
cardDeposit	integer	Card deposit amount that is returned when card is refunded (except if due to misuse or damage on the card).
cardFee	integer	Card sale fee (it is not refunded)
transactMaxNum	integer <u>Cardinality:</u> 0..1	Information is kept for compatibility with CCU interface. It is optional.
familyPasses		A Product Family is a group of Products that have a set of common behaviors: <ul style="list-style-type: none"> Similar sale and usage rules Similar pricing rules Configurable by the same set of parameters

Field name	Type / Cardinality (1 by default)	Description
familyPass	<u>Cardinality:</u> 1..n	
familyPassCod	integer	Family pass code
names		
name	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens Len = 60 chars
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars
passDefinition		Definition of period pass products
pass	<u>Cardinality:</u> 1..n	
ownCod	integer	Code of the product owner
familyPassCod	integer	Family pass to which the product belongs to
profiles	<u>Cardinality:</u> 0..1	
profile	<u>Cardinality:</u> 1..n	
profileCode	integer	Profiles for which the pass has permission. See Appendix C.10
contractTariff	integer	Contract tariff written on the card. It identifies uniquely the combination of product and profile. Range 0..255 is reserved for magnetic fare products
supports		
cod	Integer <u>Cardinality:</u> 1..n	Lists the fare media on which this product can be sold.
passCod	integer	Unique period pass identifier.
desc		Period pass description

Field name	Type / Cardinality (1 by default)	Description
names		
name	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
name	string	Name to display on the machine screens Len = 60 chars
nameLong	string	Extended name to display on the machine screens Len = 60 chars
display	string	Name for display Len = 60 chars
displayLong	string	Extended name for display Len = 60 chars
receipt	string	Name for the receipt printing Len = 60 chars
receiptLong	string	Extended name for the receipt printing Len = 60 chars
activationValDateType	integer	Defines how the activation of product validity is computed: <ul style="list-style-type: none"> 0: Without beginning of validity date 1: At validation (sliding product) 2: At the moment of the sale 3: Absolute time (a specific date & time)
activationValDate	<u>Date</u> <u>Cardinality:</u> 0..1	Defines the validity date with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ if the ValDateType is 3 (Absolute time)
validityTimeUnitType	integer	Defines the Time Units type (hours, days, months) needed for the next field: <ul style="list-style-type: none"> 0: Minutes. 1: Hours. 2: Days. 3: Months.
validityTimeUnitNum	integer	Defines the time unit number that the pass will be valid for (refer to validityTimeUnitType field).
balanceType	<u>Integer</u> <u>Cardinality:</u> 0..1	It establishes the balance type for the product: <ul style="list-style-type: none"> 1: Non-monetary. 2: None Information is kept for compatibility with CCU interface. It is optional.

Field name	Type / Cardinality (1 by default)	Description
dailyTripCtrl	integer	Defines if there is a daily trip limitation: <ul style="list-style-type: none"> 0: no limitation 1: there is a limitation
maxDailyTrip	integer	Indicates the maximum number of daily trips that it is possible to do with the pass, in case that the DAILY_TRIP_CTRL is active. <ul style="list-style-type: none"> 0: there is no control N: maximum number of trips
entryExitCtrl	integer	Indicates if it is required the cycle of entry /exit validations. <ul style="list-style-type: none"> 0: Not needed. 1: needed
fareType	integer	Defines the type of price computation at sale. It implies some parameters (edition mask) that are selected at product sale: <ul style="list-style-type: none"> 0: Free (No parameter) 1: Fix price (No parameter) 2: Origin-Destination (origin-destination stations) 3: Zones (Center zone + radius number of zones) 4: Floating zones (Number of zones) 5: Zone to zone (Origin & Destination zones)
faresTableCode	integer <u>Cardinality:</u> 0..1	Defines the fare table to use for the product price computation.
bundleTrips	integer	Defines the number of trips of the pass. <ul style="list-style-type: none"> 0: there is no control N: maximum number of trips
maxTripTime	integer	Defines the maximum number of minutes that the pass can be used in the paid area after an entry validation.
passBackCtrl	integer	Defines if anti pass-back control is activated <ul style="list-style-type: none"> 0: anti pass-back control deactivated. 1: anti pass-back control activated
passBackTime	integer <u>Cardinality:</u> 0..1	Defines the anti pass-back duration in minutes in case it is activated (Refer to passBackCtrl field). It defines the time during which a Fare Media cannot be used again on the same station.
slotTimeAccept	integer	Defines the time period for which the pass is valid. <ul style="list-style-type: none"> 0: Only for the peak-off slot. 1: Valid for peak-off and normal slot. 2: Valid for all slots

Field name	Type / Cardinality (1 by default)	Description
fare	integer <u>Cardinality:</u> 0..1	Defines the value of product price if FARE_TYPE is 1 (Fix price).
refundable	integer	Defines the minimum remaining validity of the pass required to do a refund (as defined in the ticketing rules). <ul style="list-style-type: none"> 0: No possibility of refund. N: minimum percentage of remaining time (or remaining trip usage) required to allow a product refund.
renewable	boolean	Determines if a pass can be renewed before being expired.
renewalTime	integer	Number of days before the pass expiration from which the pass can be renewed
specialFare	integer	Determines if the pass has a special fare or not <ul style="list-style-type: none"> 0: normal fare 1: special fare
amount	integer	Percentage applied on the remaining product amount for calculating the refunded amount.
startValPeriod	dateTime <u>Cardinality:</u> 0..1	Sets the fixed validity starting date and time for the pass. It is typical for promotional passes. Xml format dateTime: yyyy-mmddThh:mm:ssZ
endValPeriod	dateTime <u>Cardinality:</u> 0..1	Sets the fixed validity end date and time for the pass. It is typical for promotional passes. Xml format dateTime: yyyy-mmddThh:mm:ssZ
groupPass	boolean	Indicates if the pass is a pass group.
maxGroupSize	integer <u>Cardinality:</u> 0..1	Maximum group size
load_permits		Indicates if tom device type can sell the product
tom_entry	integer	1: Operation allowed to the machine / 0: Not allowed
tom_exit	integer	1: Operation allowed to the machine / 0: Not allowed
setOriginAtSale	boolean	Determines if the sale operator has to choose the origin location at sale time. The origin can be a station or a zone
reload_permits		Indicates if tom device type can reload the product.
tom_entry	integer	1: Operation allowed to the machine / 0: Not allowed
tom_exit	integer	1: Operation allowed to the machine / 0: Not allowed
linesAllowed		Lines where the pass is valid
lineCode	integer <u>Cardinality:</u> 1..n	Indicates line identifier where the pass is valid.

Field name	Type / Cardinality (1 by default)	Description
Signature		See Appendix E.3

3.4.3.4 MT Fare parameter file

FileName	magneticFareTable.xml
Directory	\#Level3Name#\cfg\out\businessParameters\ <ul style="list-style-type: none"> \ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	MagneticFareTable.xsd
Description	This file defines the fare parameters for magnetic tickets: <ul style="list-style-type: none"> singleRideTicket dailyPassTicket weeklyPassTicket monthlyPassTicket seasonPassTicket agentPassTicket freePassTicket The details of Magnetic ticket specification is provided in document 4020_19982 (Line 1&2 Fare Structure - Magnetic tickets)

Limitations for old magnetic gates (refer to document 4020_19817):

- The old magnetic gates implement the following tickets:
 - singleRideTicket
 - seasonPassTicket
 - agentPassTicket
 - freePassTicket
 (Following products are not implemented dailyPassTicket, weeklyPassTicket, monthlyPassTicket)
- Number of zones is limited to 8

Field name	Type / Cardinality (1 by default)	Description
fareTableMagneticTicket		
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ

Field name	Type / Cardinality (1 by default)	Description
<u>magneticTicket*</u>		Magnetic tickets: <ul style="list-style-type: none"> singleRideTicket dailyPassTicket weeklyPassTicket monthlyPassTicket seasonPassTicket agentPassTicket freePassTicket
category	string	Attribute defining the ticket category: <ul style="list-style-type: none"> Single Pass FreePass Agent
refNumber	integer	Reference number identifying the type of ticket. In the ticket data layout (document 4020_19982), it corresponds to the fields "Type of ticket tens" & "Type of ticket units". Note that the unit part of the refNumber defines the fare type: <ul style="list-style-type: none"> 0 : full fare 1 : reduced fare 1 2 : reduced fare 2 3 : reduced fare 3
isOperational		True: ticket fare is active False: ticket fare is not active
names		Defines the ticket product names
name	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
longName	string	Long name Len = 40 characters
shortName	string	Short name Len = 20 characters
abbreviation	string	Abbreviation Len = 5 characters
expiryDuration	integer <u>Cardinality:</u> 0..1	Number of days for period pass validity.
tariffs		

Field name	Type / Cardinality (1 by default)	Description
tariff	<u>Cardinality:</u> 1..16	
numberOfZones	integer	Number of zones for the ticket geographical validity.
price	integer	Price for the number of zones
nbInterStation	integer	Number of inter stations corresponding to the number of zones.
reducedFare	<u>Cardinality:</u> 0..3	
refNumber	integer	Reference number identifying the type of ticket. In the ticket data layout (document 4020_19982), it corresponds to the fields "Type of ticket tens" & "Type of ticket units". Note that the unit part of the refNumber defines the fare type: <ul style="list-style-type: none"> • 0 : full fare • 1 : reduced fare 1 • 2 : reduced fare 2 • 3 : reduced fare 3
isOperational		True: ticket fare is active False: ticket fare is not active
names		Defines the ticket product names
name	<u>Cardinality:</u> 1..2	
language	string	Specifies the language of the next labels. Using the ISO639-2/B code. 'ara' Arabic or 'eng' English
longName	string	Long name Len = 40 characters
shortName	string	Short name Len = 20 characters
abbreviation	string	Abbreviation Len = 5 characters
tariffs		
tariff	<u>Cardinality:</u> 1..16	
numberOfZones	integer	Number of zones for the ticket geographical validity.
price	integer	Price for the number of zones
Signature		See Appendix E.3

The following table lists the refNumber value for the different ticket types:

Magnetic ticket	Constraints
-----------------	-------------

Magnetic ticket	Constraints
singleRideTicket	refNumber=10 reducedFare refNumbers: 11, 12, 13
dailyPassTicket	refNumber=20 reducedFare refNumbers: 21, 22, 23
weeklyPassTicket	refNumber=30 reducedFare refNumbers: 31, 32, 33
monthlyPassTicket	refNumber=40 reducedFare refNumbers: 41, 42, 43
seasonPassTicket	refNumber=50 reducedFare refNumbers: 51, 52, 53
agentPassTicket	refNumber=70 reducedFare refNumbers: NA
freePassTicket	refNumber=70 reducedFare refNumbers: NA

Note: as per existing MT pass implementation, validity can be 1 day, 1 week, 1 month, 3 months (season pass)

Note: There are 8 floating zones (single ride tickets). For passes, there are 3 linear zones defined by entry and exit stations.

3.4.3.5 Card Blacklist file

FileName	blackListCard.xml
Directory	\\#Level3Name#\cfg\out\blacklistCard\x\ <ul style="list-style-type: none"> x\ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	BlackListCard.xsd
Description	This file defines: <ul style="list-style-type: none"> The cards that are in the blacklist The card ranges that are in the blacklist

Field name	Type Cardinality (1 by default)	Description
blackList		
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ

Field name	Type Cardinality (1 by default)	Description
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
cards		Cards included into the blacklist
card	<u>Cardinality:</u> 0..50000	
serialNumber	long	Serial number of the card
sequenceNumber	integer	Sequence Number to manage the blocking/unblocking of the card.
reason	integer	Blacklisting reason. See appendix C.8
cardRanges		Card ranges included in the blacklist
cardRange	<u>Cardinality:</u> 0..100	
initRange	long	Start range of cards that will be blocked
endRange	long	End range of cards that will be blocked
reason	integer	Blacklisting reason. See appendix C.8
Signature		See Appendix E.3

Note: there are at most 100 ranges. This is the limitation in the existing CCU interface. To check with the 3rd party vendor if more ranges can be added.

3.4.3.6 SAM Blacklist file

FileName	blackListSAM.xml
Directory	\#Level3Name#\cfg\out\blacklistSAM\x\ <ul style="list-style-type: none"> \x\ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	BlackListSAM.xsd
Description	This file defines: <ul style="list-style-type: none"> The SAM AV2 that are in blacklist.

Field name	Type Cardinality (1 by default)	Description
blackListSAM		
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ

Field name	Type / Cardinality (1 by default)	Description
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
sams		SAM AV2 included in blackList
sam		
samSerialNumber	long	Blacklisted SAM serial number
activationDate	dateTime	Date when the SAM was blacklisted. Xml dateTime: yyyy-mmddThh:mm:ssZ
reason	integer	Blacklisting reason.
Signature		See Appendix E.3

CONFIDENTIAL

3.5 BATCH INTERFACE TO CBO

The front end devices generate an event for each operation that they execute. These events are forwarded to the Level3 system that is responsible to forward them to the Central Back Office. Thus the Level3 systems send the **event data files** on the CBO SFTP server. The CBO then processes the events. In particular it maintains the card image at back office and provides some reporting functionalities on the whole network.

Another responsibility of the Level3 system is the front end device management. This means that the Level3 system declares and controls the front end devices. It shall forward to the Central back office the **equipment list file** that contains the front end devices deployed on the transportation network. This allows the CBO to link the event data to the corresponding station equipment.

3.5.1 FILE TRANSFER DESCRIPTION

The interface is based on the same principles than the batch interface from CBO. That's why the following paragraphs also apply for this interface:

- § 3.4.1.1 **Protocol for file transfer**
- § 3.4.1.2 **Configuration**
- § 3.4.1.3 **File Security**
- § 3.4.1.4 **Degraded modes**

3.5.2 EVENT DATA FILES

FileName	MM_NNNNN_YYMMDDSEQ.xml Refer to description in table below
Directory	\#Level3Name#\data\yyyymmdd\ <ul style="list-style-type: none"> • yyyymmdd is a sub directory per day that contains the event data files: <ul style="list-style-type: none"> ○ yyyy: year ○ mm: month ○ dd: day
XSD	LCUEventsDataFile.xsd
Description	<p>The event data files are created by the front end devices. They contain the operations executed on the fare media or on the device itself. The event data file can contain:</p> <ul style="list-style-type: none"> • Fare media transactions: <ul style="list-style-type: none"> ○ Sale and services (product sale, Tuprse reload, product refund ...) ○ Validation events • Device events: <ul style="list-style-type: none"> ○ Filtered device alarms ○ Stock status ○ Agent shift open / close ○ Device startup ○ Device shutdown (End of file)

The event data files are named with following convention:

MM_NNNNN_YYMMDDSEQ.xml
MM : Number that identifies the equipment model
NNNNN: Specifies the equipment serial number
YY: Represents the year of the exploitation day (0-99).
MM: Represents the month of the exploitation day (1-12).
DD: Represents the day of the exploitation day (1-31).
SEQ: Represents the sequence of the day. (one every 15 minutes)

Field name	Type / Cardinality (1 by default)	Description
dataRecordsCollection		
xs:choice	<u>Cardinality:</u> 0..n	The file can contain several events. The event types are chosen from the list (Refer to § 3.5.2.1): <ul style="list-style-type: none"> • startUp • accountingOperation • alarm • beginShift • endShift • cardAction • cardStockUpdate • dataCollectionComplete • penalty • personalization • personalizationReconstruction • unitReconstruction • validation • samChanged • QuotaUpdated • MTaccountingOperation • MTvalidation
endOfFile	<u>Cardinality:</u> 0..1	Optional part that can be generated at device shutdown or at the end of the operational day. Refer to § 3.5.2.1.19
Signature		See Appendix E.3

3.5.2.1 Events

The following paragraphs describe the different event types.

Event	Description
Start-up	This event is generated each time the device is switched on.

Event	Description
Alarm	This event is generated when an alarm is started or ended on the device.
Card stock update	This event is generated when an operator of SCU or TOM updates the card stock.
Validation	This transaction is generated when a customer validates his CSC on front end device. It can be generated event when passage is rejected.
Accounting operation	This transaction is generated when a front end operation involves a payment for a CSC. For instance charge, recharge, cancellation and refund operations.
Personalization	This transaction is generated when the fare media is personalized.
Penalty	This event is generated by an inspection device when the operator decides that the card owner committed fraud.
Beginning of Shift	This event is generated by manned devices when an operator initiates a new shift.
End of Shift	This event is generated by manned devices when an operator closes a shift.
Card Action	This transaction is generated when any equipment blocks or unblocks a card.
Personalization data reconstruction	This transaction is generated when a personalized fare media is reconstructed.
Unit data reconstruction	This transaction is generated for each item reconstructed on a fare media. In one reconstruction action over one card, one transaction is generated: <ul style="list-style-type: none"> Emission of the new card Tpurse reconstruction action Each reconstructed contract
Data collection complete	This event is <u>optionally</u> generated when the data collection is complete for one operational day of a device.
Sam Changed	This event is generated when the AV2 SAM is changed on the device.
Sam Quota Updated	This event is generated when an AV2 SAM quota is updated (in equipment).
MT Accounting operation (Magnetic ticket)	This transaction is generated when a front end operation involves a payment for a Magnetic ticket.
MT Validation (Magnetic ticket)	This transaction is generated when a customer validates his magnetic ticket on front end device. It can be generated event when passage is rejected.
End of file	This event is generated when the device is switched off.

Note that some events use data structures that are taken from the Fare Media Electrical Data Layout specifications. These structures are detailed in paragraph 3.7.

Note that when a defective equipment is replaced, the last data elements are likely to be not uploaded. So in this case, the data collection complete is not generated in order to highlight such a situation. This is the reason why data collection complete is optional.

3.5.2.1.1 Event header

The event data records contain a general header that is common to all the records. The header fields are detailed in the following table:

Field name	Type / Cardinality (1 by default)	Description
head		
recordId	integer	The record code. Each event type has its own code: <ul style="list-style-type: none"> • 1 : Start-up • 2 : Alarm • 3 : Card stock update • 4 : Validation • 5 : Accounting operation • 6 : Personalization • 7 : Penalty • 8 : Beginning of Shift • 9 : End of Shift • 10 : Card Action • 11 : Personalization data reconstruction • 12 : Unit data reconstruction • 14 : Data collection complete • 15 : Sam Changed • 16 : Sam Quota Updated • 255: End of file
version	integer	The record version / 1 It is increased when the XSD is changed.
date	dateTime	Date and time when the message was created. Format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
universalSeq	integer <u>Cardinality:</u> 0..1	Universal sequence number. Information is kept for compatibility with CCU interface. It is optional.
dailySeq	integer <u>Cardinality:</u> 0..1	Sequential number assigned to the event per device and per day. The start-up registry must be daily sequence number 1 Information is kept for compatibility with CCU interface. It is optional.
serviceProvider	integer <u>Cardinality:</u> 0..1	Service provider information provided by the device: See Appendix C.9
equipmentType	integer	Equipment type: See Appendix C.1
equipmentModel	integer	Reserved for Future Use when subtypes of equipment will be available. For the moment, it has the same value as the equipment type.
serialNumber	integer	Unique serial number assigned to this equipment. Limited to 5 digits

Field name	Type Cardinality (1 by default)	Description
line	integer	Code of the line where the equipment is located. If value is not known, the value = 0
station	integer	Code of the station where the equipment is located. If value is not known, the value = 0
hall	integer	Code of the hall where the equipment is located. If value is not known, the value = 0
position	integer	Position of the machine within the hall. If value is not known, the value = 0

3.5.2.1.2 Start-up

This event is generated each time the device is switched on.

Field name	Type Cardinality (1 by default)	Description
startUp		
head		Refer to event header § 3.5.2.1.1
hostName	string	Hostname of the machine.
IPAddress	string	IP address of the machine.
SAM1	long Cardinality: 0..1	AV2 SAM serial number (if the equipment has two SAM's this will be the entry one)
SAM1quota	integer Cardinality: 0..1	The current quota for the SAM1
SAM2	long Cardinality: 0..1	AV2 SAM serial number (if the equipment has 2 SAM's this will be the exit one)
SAM2quota	integer Cardinality: 0..1	The current quota for the SAM2

Note: 2 SAMs is typically the case of a bi-directional gate having 2 readers.

3.5.2.1.3 Alarm

This event is generated when an alarm is started or ended on the device.

Field name	Type Cardinality (1 by default)	Description
alarm		

Field name	Type Cardinality (1 by default)	Description
head		Refer to event header § 3.5.2.1.1
alarmCode	integer	Code of the alarm. See Appendix C.7
closingFlag	boolean	Closing or opening flag <ul style="list-style-type: none"> False: Opening flag True: Closing flag
openingDate	dateTime	Opening alarm date and time. Format Xml dateTime: yyyy-mmddThh:mm:ssZ - T indicates the start of the time section. - Z indicates the time zone. Here there is no offset. UTC time is used

3.5.2.1.4 Card stock update

This event is generated when an operator of SCU or TOM updates the card stock.

Field name	Type Cardinality (1 by default)	Description
cardStockUpdate		
head		Refer to event header § 3.5.2.1.1
Operator	integer	Identifier of the operator who makes the operation.
supportState	integer	0: correct media 1: defective media
typeOfVariation	integer	0: increment the stock 1 : decrement the stock
Stock	integer	Number of items that were added or removed from stock.

Note: in case of minimum stock reached, the TOM raises automatically an alarm (refer to C.7)

3.5.2.1.5 Validation

This transaction is generated when a customer validates his CSC on front end device. It can be generated event when passage is rejected.

Field name	Type Cardinality (1 by default)	Description
validation		
head		Refer to event header § 3.5.2.1.1
validationType	integer	See Appendix C.3
validationError	integer	See Appendix C.11
cardId	long	Card serial number

Field name	Type / Cardinality (1 by default)	Description
trips	integer <u>Cardinality:</u> 0..1	Number of trips discounted because of the operation for non-monetary passes.
cardTrips	integer <u>Cardinality:</u> 0..1	Number of trips remaining after the operation for non-monetary passes.
operationBalance	integer	Amount charged on to the TPurse
cardBalance	integer	TPurse Balance remaining after the operation
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
tPurseContext		Tpurse context. See Card Data structures § A.3
tPurseValue	integer	Value of TPurse
appContext		Application context. See Card Data structures § A.4
contractLists		List of contract instances. See Card Data structures § A.5
contract	<u>Cardinality:</u> 0..1	Contract instance used for the operation. It is absent if no contract is used (when using Tpurse for instance). See Card Data structures § A.6
eventLog		Event log. See Card Data structures § A.7
previousEventLog	<u>Cardinality:</u> 0..1	Previous event log. See Card Data structures § A.11

3.5.2.1.6 Accounting operation

This transaction is generated when a front end operation involves a payment for a CSC. For instance charge, recharge, cancellation and refund operations.

Field name	Type / Cardinality (1 by default)	Description
accountingOperation		
head		Refer to event header § 3.5.2.1.1
operationType	integer	See Appendix C.2
operationNumber	integer <u>Cardinality:</u> 0..1	Sequential number of accounting operation in this machine. Information is kept for compatibility with CCU interface. It is optional.

Field name	Type / Cardinality (1 by default)	Description
personalizationFlag	integer <u>Cardinality:</u> 0..1	Indicates if the operation implies card personalization. This implies that a personalization event is mandatory, following accounting event Information is kept for compatibility with CCU interface. It is optional.
customerId	long <u>Cardinality:</u> 0..1	Customer id If the operation is attachment to a customer. In other cases the value is NULL or omitted.
cardID	long	Card serial number
trips	integer <u>Cardinality:</u> 0..1	Number of trips involved (charged or returned) in the operation for non-monetary pass
cardTrips	integer <u>Cardinality:</u> 0..1	Number of trips remaining after the operation for non-monetary passes.
operationBalance	integer	Amount involved (charged or returned) in the operation for Tpurse
cardBalance	integer	Amount remaining after the operation for Tpurse.
passPrice	integer <u>Cardinality:</u> 0..1	Price paid or returned for the pass
personalizationFee	integer <u>Cardinality:</u> 0..1	Fee to personalize the support
cardDeposit	integer <u>Cardinality:</u> 0..1	Card deposit
totalAmount	integer	Total amount of the operation (operationBalance+passPrice+personalizationFee+cardDeposit)
payment	integer	Payment means. See Appendix C.5
serviceNumber	integer	Shift identifier
userCode	integer	Identifier of the operator who makes the operation (AgentID)
sequenceNumber	integer <u>Cardinality:</u> 0..1	Action list sequence Number, when the event corresponds to the execution of an action-list
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
tPurseContext		Tpurse context. See Card Data structures § A.3
tPurseValue	integer	Value of TPurse

Field name	Type / Cardinality (1 by default)	Description
tPurseLog		See Card Data structures § A.9
appContext		Application context. See Card Data structures § A.4
contractLists		List of active contract instances. See Card Data structures § A.5
contract	Cardinality: 0..1	Contract instance used for the operation. It is absent if no contract is used (when using Tpurse for instance). See Card Data structures § A.6
environment		See Card Data structures § A.8

3.5.2.1.7 Personalization

This transaction is generated when the fare media is personalized.

Field name	Type / Cardinality (1 by default)	Description
personalization		
head		Refer to event header § 3.5.2.1.1
operationNumber	Integer Cardinality: 0..1	Sequential number of accounting operation in this machine. This number must be the same than the accounting operation this one is attached to. Information is kept for compatibility with CCU interface. It is optional.
profileCode	integer	Profile code Following the EN1545 standard Codification
cardID	long	Card serial number
customerId	long	The value of the customerId
update	boolean	True if the record is a update of a personalized data card.
result	boolean	Operation result
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
appContext		Application context. See Card Data structures § A.4

3.5.2.1.8 Penalty

This event is generated by an inspection device when the operator decides that the card owner committed fraud.

Field name	Type / Cardinality (1 by default)	Description
penalty		

Field name	Type / Cardinality (1 by default)	Description
head		Refer to event header § 3.5.2.1.1
cardID	long	Card serial number
passCode	integer	Code of the pass involved in the operation.
penaltyReason	integer	Penalty reason. See Appendix C.4
penaltyAmount	integer	Amount charged to the client who has been fined.
payment	integer	Payment means. See Appendix C.5
serviceNumber	integer	Shift identifier
userCode	integer	Identifier of the operator who makes the operation (AgentID)
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
appContext		Application context. See Card Data structures § A.4
eventLog		Event log. See Card Data structures § A.7
specialEventLog		Special event log. See Card Data structures § A.10

Note: in case of a CSC holding a penalty event (special event), the gate at exit generates a validation event of type penalty (refer to C3). Precision will be added.

3.5.2.1.9 Beginning of Shift

This event is generated by manned devices when an operator initiates a new shift.

Field name	Type / Cardinality (1 by default)	Description
beginShift		
head		Refer to event header § 3.5.2.1.1
operator	integer	Identifier of the operator opening the shift (AgentID)
mode	integer	Agent profile (sales, maintenance ...). See Appendix C.6
identificationType	integer	Identification source: <ul style="list-style-type: none"> 1: identification by keyboard 2: identification by contact less reader
shiftID	integer	Unique Shift Identifier on the machine
stock	integer	Number of available CSC in the device
erroneous	integer	Number of cards detected as erroneous

3.5.2.1.10 End of Shift

This event is generated by manned devices when an operator closes a shift.

Field name	Type / Cardinality (1 by default)	Description
endShift		
head		Refer to event header § 3.5.2.1.1
operator	integer	Identifier of the operator closing the shift (AgentID)
mode	integer	Agent profile (sales, maintenance ...). See Appendix C.6
shiftID	integer	Unique Shift Identifier on the machine
totalOperations	integer	Number of operations performed in the current shift.
totalAmount	integer	Amount from operations in the current shift.
emissionOperations	integer	Number of emissions performed in the current shift.
emissionAmount	integer	Amount of emissions performed in the current shift.
loadOperations	integer	Number of card loads performed in the current shift.
loadAmount	integer	Amount of card loads performed in the current shift.
reloadOperations	integer	Number of card reloads performed in the current shift.
reloadAmount	integer	Amount of card reloads performed in the current shift.
cancelOperations	integer	Number of operations cancelled in the current shift.
cancellationAmount	integer	Amount of operations cancelled in the current shift.
penaltyOperations	integer	Number of penalties in the current shift.
penaltyAmount	integer	Amount of penalties in the current shift.
numRefunds	integer	Number of refunds performed in the current shift.
shiftRefundsAmount	integer	Amount from refunds in the current shift.
numCustomizations	integer	Number of cards customizations performed in the current shift
customizationsAmount	integer	Amount of cards customizations performed in the current shift
stock	integer	Number of available card
erroneous	integer	Number of cards detected as erroneous

3.5.2.1.11 Card Action

This transaction is generated when any equipment blocks or unblocks a card.

Field name	Type / Cardinality (1 by default)	Description
cardAction		
head		Refer to event header § 3.5.2.1.1
shiftID	integer	Unique Shift Identifier on the machine (apply just for TOM)

Field name	Type / Cardinality (1 by default)	Description
cardId	long	Card serial number.
action	integer	Action performed over the card: <ul style="list-style-type: none"> 1: blocking of card 4: unblocking of card
blockingReason	integer	Blocking reason if the action is 1,2 or 3. See Appendix C.8
blackListVersion	integer	Indicates the black list version if reason is 1 or 2
sequenceNumber	integer	Sequence Number to manage the action list
result	boolean	Operation result <ul style="list-style-type: none"> True : successful False : unsuccessful
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
tPurseContext		Tpurse context. See Card Data structures § A.3
tPurseValue	integer	Value of TPurse
appContext		Application context. See Card Data structures § A.4
contractLists		List of active contract instances. See Card Data structures § A.5
contract	<u>Cardinality:</u> 0..1	Contract instance used for the operation. It is absent if no contract is used (when using TPurse for instance). See Card Data structures § A.6

3.5.2.1.12 Personalization data reconstruction

This transaction is generated when a personalized fare media is reconstructed.

Field name	Type / Cardinality (1 by default)	Description
personalizationReconstruction		
head		Refer to event header § 3.5.2.1.1
operationNumber	integer <u>Cardinality:</u> 0..1	Sequential number of accounting operation in this machine. This number must be the same than the accounting operation this one is attached to. Information is kept for compatibility with CCU interface. It is optional.
profileCode	integer	Profile code Following the EN1545 standard Codification
customerId	integer	Is the value of the customerId attachment to the personalized card
result	boolean	Operation result

Field name	Type / Cardinality (1 by default)	Description
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardID	long	Card serial number of a new card
cardTagId	long	Card engraved number of a new card
oldCardID	long	Card serial number of old card
oldCardTagId	long	Card engraved number of old card
appContext		Application context. See Card Data structures § A.4

3.5.2.1.13 Unit data reconstruction

This transaction is generated for each item reconstructed on a fare media. In one reconstruction action over one card, one transaction is generated:

- Emission of the new card
- Tpurse reconstruction action
- Each reconstructed contract

Field name	Type / Cardinality (1 by default)	Description
unitReconstruction		
head		Refer to event header § 3.5.2.1.1
operationType	integer	See Appendix C.2
operationNumber	Integer <u>Cardinality:</u> 0..1	Sequential number of accounting operation in this machine. This number must be the same than the following customization record when appropriate. Information is kept for compatibility with CCU interface. It is optional.
customerId	long	Indicates the customer id If the operation is attachment to a customer in other case the value is NULL
cardID	long	Card serial number of new card
oldCardID	long	Card serial number of old card
trips	integer	Number of trips involved (charged or returned) in the operation for non-monetary pass
cardTrips	integer <u>Cardinality:</u> 0..1	Number of trips remaining after the operation for non-monetary passes.
operationBalance	integer	Amount involved (charged or returned) in the operation for Tpurse
cardBalance	integer	Amount remaining after the operation for Tpurse.

Field name	Type / Cardinality (1 by default)	Description
passPrice	integer	Price paid or returned for the pass
personalizationFee	integer	Fee to personalize this support
cardDeposit	integer	Card deposit
totalAmount	integer	Total amount of the operation (operationBalance+passPrice+personalizationFee+cardDeposit)
payment	integer	Payment means. See Appendix C.5
serviceNumber	integer	Shift identifier
userCode	integer	Identifier of the operator who makes the operation (AgentID)
cardData		Card related data. Based on "Fare Media Electrical Data Layout Specification"
cardDataVersion	integer	
cardTagId	long	Card engraved number
tPurseContext		Tpurse context. See Card Data structures § A.3
tPurseValue	integer	Value of TPurse
tPurseLog		See Card Data structures § A.9
appContext		Application context. See Card Data structures § A.4
contractLists		List of active contract instances. See Card Data structures § A.5
contract	Cardinality: 0..1	Contract instance used for the operation. It is absent if no contract is used (when using TPurse for instance). See Card Data structures § A.6
environment		See Card Data structures § A.8

3.5.2.1.14 Data collection complete

This event is optionally generated when the data collection is complete for one operational day of a device. It is kept for compatibility with former CCU interface.

Field name	Type / Cardinality (1 by default)	Description
dataCollectionComplete		
head		Refer to event header § 3.5.2.1.1
completedDay	dateTime	Date for which the data has been completed. Xml dateTime: yyyy-mm-ddThh:mm:ssZ

3.5.2.1.15 Sam Changed

This event is generated when the AV2 SAM is changed on the device.

Field name	Type / Cardinality (1 by default)	Description
samChanged		
head		Refer to event header § 3.5.2.1.1
oldSamSerialNumber	long	Old SAM serial number
newSamSerialNumber	long	New SAM serial number
oldSAMquota	integer	Old SAM Quota
newSAMquota	integer	New SAM Quota

3.5.2.1.16 Sam Quota Updated

This event is generated when an AV2 SAM quota is updated (in equipment).

Field name	Type / Cardinality (1 by default)	Description
samQuotaUpdated		
head		Refer to event header § 3.5.2.1.1
samSerialNumber	long	SAM serial number
oldSAMquota	integer	SAM Old Quota
newSAMquota	integer	SAM New Quota

3.5.2.1.17 MT accounting operation

This transaction is generated when a front end operation involves a payment for a magnetic ticket.

Field name	Type / Cardinality (1 by default)	Description
MTaccountingOperation		
head		Refer to event header § 3.5.2.1.1
operationType	integer	See Appendix C.2: <ul style="list-style-type: none"> Charge : ticket sale Contract reconstruction : MT fare media exchange
ticketType	integer	Reference number identifying the type of ticket.
entryStation	integer <u>Cardinality:</u> 0..1	Code of the entry station defining the geographical validity for period passes.
exitStation	integer <u>Cardinality:</u> 0..1	Code of the exit station defining the geographical validity for period passes.
numberOfZones	integer	Number of zones for the ticket geographical validity.

Field name	Type / Cardinality (1 by default)	Description
expiryDate	date Cardinality: 0..1	Ticket expiry date.
totalAmount	integer	Amount of the operation
payment	integer	Payment means. See Appendix C.5
serviceNumber	integer	Shift identifier
userCode	integer	Identifier of the operator who makes the operation (AgentID)

3.5.2.1.18 MT validation

This transaction is generated when a customer validates his magnetic ticket on front end device. It can be generated event when passage is rejected.

Field name	Type / Cardinality (1 by default)	Description
MTvalidation		
head		Refer to event header § 3.5.2.1.1
validationType	integer	See Appendix C.3
validationError	integer	See Appendix C.11
ticketType	integer	Reference number identifying the type of ticket.
entryStation	integer Cardinality: 0..1	Code of the entry station defining the geographical validity for period passes.
exitStation	integer Cardinality: 0..1	Code of the exit station defining the geographical validity for period passes.
numberOfZones	integer	Number of zones for the ticket geographical validity.
expiryDate	date Cardinality: 0..1	Ticket expiry date.

3.5.2.1.19 End Of File

This event is optionally generated when the device is switched off. It is kept for compatibility with former CCU interface.

Field name	Type / Cardinality (1 by default)	Description
endOfFile		
head		Refer to event header § 3.5.2.1.1

3.5.3 EQUIPMENT LIST FILE

FileName	equipmentListYYMMDD.xml Refer to description in table below
Directory	\\#Level3Name#\cfg\in\equipment\%x\ <ul style="list-style-type: none"> %x\ is a directory named with the version number of the file. Refer to §3.4.2.
XSD	EquipmentList.xsd
Description	Level3 system is responsible of stations equipment management. The list of devices is transmitted to the CBO. This file shall be generated every day a change occurs on stations configuration. Equipment list always contains all equipment, not only those modified (added or removed).

The equipment list file is named with following convention:

equipmentListYYMMDD.xml
YY: Represents the year of new configuration generation (0-99).
MM: Represents the month of the exploitation generation (1-12).
DD: Represents the day of the exploitation generation (1-31).

Field name	Type Cardinality (1 by default)	Description
equipmentList		
head		
formatVersion	integer	File format version
confVersion	integer	File configuration version
generationDate	dateTime	Generation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
activationDate	dateTime	Activation date of configuration version with format Xml dateTime: yyyy-mm-ddThh:mm:ssZ
stations		Station equipment definition
station	<u>Cardinality:</u> 1..n	
stationId	integer	Station identifier for following equipments
line	integer <u>Cardinality:</u> 0..1	Main line where the station is located
equipments		Equipment definition
equipment	<u>Cardinality:</u> 1..n	
equipmentType	integer	Equipment type: See Appendix C.1
serialNumber	integer	Unique serial number (deviceId)

Field name	Type / Cardinality (1 by default)	Description
hall	integer <u>Cardinality:</u> 0..1	Hall location.
ordinal	integer <u>Cardinality:</u> 0..1	Ordinal at hall and station.
name	string	Device name
Signature		See Appendix E.3

CONFIDENTIAL

3.6 ONLINE WEB SERVICES

The online web services are used by front end devices. These services concern following back office data:

- Customer data
- Card data
- Claim data
- AV2 SAM quota

3.6.1 WEB SERVICE TECHNICAL DESCRIPTION

3.6.1.1 Web services protocols

The web services are based on following protocols

- SOAP 1.1
- HTTPs
- XML 1.0

3.6.1.2 Web services security

Refer to §3.2.

3.6.1.3 Web service versioning

The web service version is defined by the XSD schema version attribute.

3.6.1.4 Configuration

The following parameters shall be set for the client part:

The web service access points shall be configured with the web service URI including Server Name, port number and operation name.

Parameter	Comment
EXECUTION_TIMEOUT	Max time allowed for one Web Service call.
MAX_MESSAGE_SIZE	Max size of message allowed for one request or response message.
Web service URIs	The web service access points shall be configured with the web service URI including Server Name, port number and operation name.

3.6.1.5 Degraded modes

If the Web Service call fails, the client shall catch the error and retry the request if necessary.

3.6.1.6 Server errors

When the server encounters an error, it returns a <soap:Fault> element. See Appendix D.6 for details.

3.6.2 CUSTOMER DATA

These web services allow managing the Customer information.

WSDL	customerdatabase.wsdl
XSD	customerdatabase_schema1.xsd
Web service	Operation message name
Customer creation	customerRecordCreation
Customer enquiry	customerRecordEnquiry
Customer update	customerRecordUpdate
Attach card to customer	customerCardAttachment

3.6.2.1 Customer creation

This web service creates a new Customer record in CBO database.

3.6.2.1.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
customerCreationRequest		
firstName	string	Customer first name
lastName	string	Customer last name
docNumber	string	Number of document
docType	integer	Type of document used to prove the customer identity (see Appendix D.4)
address	string <u>Cardinality:</u> 0..1	
postZipCode	string <u>Cardinality:</u> 0..1	
city	string <u>Cardinality:</u> 0..1	
phoneNumber	string <u>Cardinality:</u> 0..1	Phone Number (home, mobile, work)

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
email	string <u>Cardinality:</u> 0..1	
birthDate	date	Customer birthdate
profile	integer	Customer Profile (see Appendix C.10)
profileEndDate	dateTime	End of profile validity
language	string <u>Cardinality:</u> 0..1	Preferred language (ISO639-2/B codes 'ara' Arabic or 'eng' English)
compData	string <u>Cardinality:</u> 0..1	Complementary data (free text)

3.6.2.1.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
customerResponse		
result	integer	Result (see Appendix D.1)
customerId	long <u>Cardinality:</u> 0..1	Customer Id created by CBO
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.2.2 Customer enquiry

This web service returns Customer information.

3.6.2.2.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
customerEnquiryRequest		
docType ¹⁾	integer	Type of document used to prove the customer identity (see Appendix D.4)
docNumber ¹⁾	string	Number of document
cardTagId ¹⁾	long	Card engraved number

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
cardId ¹⁾	long	Card serial number
customerId	long <u>Cardinality:</u> 0..1	Customer ID
firstName	string <u>Cardinality:</u> 0..1	Customer first name
lastName	string <u>Cardinality:</u> 0..1	Customer last name

¹⁾ One of the following parameter shall be provided in the request (the others can be set to zero):

- docType
- docNumber
- cardTagId
- cardId

3.6.2.2.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
customerEnquiryResponse		
result	integer	Result (see Appendix D.1)
customerId	long <u>Cardinality:</u> 0..1	Customer Id
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)
firstName	string <u>Cardinality:</u> 0..1	Customer first name
lastName	string <u>Cardinality:</u> 0..1	Customer last name
docNumber	string <u>Cardinality:</u> 0..1	Number of document
docType	integer <u>Cardinality:</u> 0..1	Type of document used to prove the customer identity (see Appendix D.4)
address	string <u>Cardinality:</u> 0..1	

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
postZipCode	string <u>Cardinality:</u> 0..1	
city	string <u>Cardinality:</u> 0..1	
phoneNumber	string <u>Cardinality:</u> 0..1	Phone Number (home, mobile, work)
email	string <u>Cardinality:</u> 0..1	
birthDate	date <u>Cardinality:</u> 0..1	Customer birthdate
profile	integer <u>Cardinality:</u> 0..1	Customer Profile (see Appendix C.10)
language	string <u>Cardinality:</u> 0..1	Preferred language (ISO639-2/B codes 'ara' Arabic or 'eng' English)
profileEndDate	dateTime <u>Cardinality:</u> 0..1	End of profile validity
compData	string <u>Cardinality:</u> 0..1	Complementary data (free text)
cards	<u>Cardinality:</u> 0..1	
card	<u>Cardinality:</u> 1..n	
cardId	long	Card serial number
cardTagId	long	Card engraved number
disabled	boolean	Indicates if card is disabled
claimList	<u>Cardinality:</u> 0..1	
claim	<u>Cardinality:</u> 1..n	
claimId	long	Claim identifier
claimText	string	Description of the claim: initial text + cumulated updates
state	integer	Claim state. See Appendix D.5
lastUpdate	string	Last update of the claim
claimDate	dateTime	Claim creation date

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
lastUpdateDate	dateTime	Last update of the claim
closedDate	dateTime	Date of claim closure

3.6.2.3 Customer update

This web service updates a customer record by providing new parameter values.

3.6.2.3.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
customerUpdateRequest		
customerId	long	Customer identifier (Mandatory)
firstName	string <u>Cardinality:</u> 0..1	Customer first name
lastName	string <u>Cardinality:</u> 0..1	Customer last name
docNumber	string <u>Cardinality:</u> 0..1	Number of document
docType	integer <u>Cardinality:</u> 0..1	Type of document used to prove the customer identity (see Appendix D.4)
address	string <u>Cardinality:</u> 0..1	
postZipCode	string <u>Cardinality:</u> 0..1	
city	string <u>Cardinality:</u> 0..1	
phoneNumber	string <u>Cardinality:</u> 0..1	Phone Number (home, mobile, work)
email	string <u>Cardinality:</u> 0..1	

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
profile	integer <u>Cardinality:</u> 0..1	Customer Profile (see Appendix C.10)
birthDate	date <u>Cardinality:</u> 0..1	Customer birthdate
language	string <u>Cardinality:</u> 0..1	Preferred language (ISO639-2/B codes 'ara' Arabic or 'eng' English)
profileEndDate	dateTime <u>Cardinality:</u> 0..1	End of profile validity
compData	string <u>Cardinality:</u> 0..1	Complementary data (free text)

Note: firstName, lastName, docNumber, birthDate and docType are fixed cannot be changed after the customer creation. This is managed at the TOM

3.6.2.3.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
customerResponse		
result	integer	Result (see Appendix D.1)
customerId	long <u>Cardinality:</u> 0..1	Customer Id
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.2.4 Attach card to customer

This web service attaches a non-personalized card to an existing customer.
If card is already attached, an error code is returned (see Appendix D.1).

3.6.2.4.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
customerCardAttachmentRequest		
customerId	long	Customer identifier
cardId	long	Card serial number

3.6.2.4.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
customerResponse		
result	integer	Result (see Appendix D.1)
customerId	long <u>Cardinality:</u> 0..1	Customer Id created
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.3 CARD DATA

These web services allow managing the CSC information. In particular they can be used in order to recover necessary data for Card Reconstruction.

WSDL	cardinformation.wsdl
XSD	cardinformation_schema1.xsd
Web service	Operation message name
Card information enquiry	cardInformationEnquiry
Card T-Purse enquiry	cardTPurseEnquiry
Card product enquiry	cardProductEnquiry
Disable card	disableCard

3.6.3.1 Card information enquiry

This web service returns the CSC information.

3.6.3.1.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
cardEnquiryRequest		
xs:choice		CardID or CardTagId is used as searching criterion.
cardId	long	Card serial number
xs:choice		
cardTagId	long	Card engraved number

3.6.3.1.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
cardInformationEnquiryResponse		
result	integer	Result (see Appendix D.1)
cardId	long <u>Cardinality:</u> 0..1	Card serial number
cardDataVersion	integer <u>Cardinality:</u> 0..1	
cardHeader	<u>Cardinality:</u> 0..1	Card header. See Card Data structures § A.2
appContext	<u>Cardinality:</u> 0..1	Application context. See Card Data structures § A.4
environment	<u>Cardinality:</u> 0..1	Environment. See Card Data structures § A.8
holder	<u>Cardinality:</u> 0..1	Card holder. See Card Data structures § A.12
holderInformation	<u>Cardinality:</u> 0..1	
firstName	string	Customer first name
lastName	string	Customer last name
customerId	long	Customer Id
blockingReason	integer <u>Cardinality:</u> 0..1	Blocking reason. See appendix C.8
blockingDate	dateTime <u>Cardinality:</u> 0..1	Blocking date
blackListingDate	dateTime <u>Cardinality:</u> 0..1	Blacklisting date
lastTransactions	<u>Cardinality:</u> 0..1	
transaction	<u>Cardinality:</u> 0..10	
date	dateTime	Transaction date and time

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
stationId	integer	Identifier of the station where the transaction was executed.
deviceId	integer	Identifier of the device that executed the transaction.
eqTypeid	integer	Device type
passCod	integer Cardinality: 0..1	Period pass associated to the transaction
contractTariff	integer Cardinality: 0..1	It identifies uniquely the combination of product and profile.
opType	integer	Operation type. Refer to appendix D.2
xs:choice		The transaction either contains an amount or a blocking information
priceAmount	integer	Amount of the transaction. 0 is transaction is zero value.
xs:choice		
blockingReason	integer	Blocking reason. See appendix C.8
blackListingDate	dateTime	Blacklisting date.
errorDescription	string Cardinality: 0..1	Present only when Result=0 (System Error)

3.6.3.2 Card T-Purse enquiry

This web service returns the TPurse information.

3.6.3.2.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
cardEnquiryRequest		
xs:choice		CardID or CardTagId is used as searching criterion.
cardId	long	Card serial number
xs:choice		
cardTagId	long	Card engraved number

3.6.3.2.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
cardTPurseEnquiryResponse		
result	integer	Result (see Appendix D.1)
cardId	long <u>Cardinality:</u> 0..1	Card serial number
cardTagId	long <u>Cardinality:</u> 0..1	Card engraved number
tPurseContext	<u>Cardinality:</u> 0..1	TPurse context. See card data structure §A.3
tPurseValue	integer <u>Cardinality:</u> 0..1	Tpurse value
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.3.3 Card product enquiry

This web service returns the active products for a CSC.

3.6.3.3.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
cardEnquiryRequest		
xs:choice		CardID or CardTagId is used as searching criterion.
cardId	long	Card serial number
xs:choice		
cardTagId	long	Card engraved number

3.6.3.3.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
cardProductEnquiryResponse		
result	integer	Result (see Appendix D.1)
contracts	<u>Cardinality:</u> 0..1	
contract	<u>Cardinality:</u> 1..2	Product instance. See Card Data structures § A.6

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.3.4 Disable Card (Lost, stolen, damaged and faulty card declaration)

This web service disables the card that is added in the CBO blacklist.

3.6.3.4.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
cardEnquiryRequest		
xs:choice		CardID or CardTagId is used as searching criterion.
cardId	long	Card serial number
xs:choice		
cardTagId	long	Card engraved number
date	dateTime	Disabling date and time.
reason	integer	Disabling reason code: see Appendix D.3

3.6.3.4.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
disableCardResponse		
result	integer	Result (see Appendix D.1)
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.4 CLAIM DATA

These web services allow managing the Claim information for a traveller complaint.

WSDL	claimmanagement.wsdl
XSD	claimmanagement_schema1.xsd
Web service	Operation message name
Creation of a claim	customerClaimCreation
Update of a claim	customerClaimUpdate
Customer claim list enquiry	customerClaimListEnquiry
Customer claim record enquiry	customerClaimEnquiry

3.6.4.1 Creation of a claim

This web service creates a new claim record at CBO.

3.6.4.1.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
claimCreationRequest		
customerId	long	Customer identifier
claimText	string	Claim description (free text)

3.6.4.1.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
claimResponse		
result	integer	Result (see Appendix D.1)
claimId	long <u>Cardinality:</u> 0..1	Claim identifier created by CBO
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.4.2 Update of a claim

This web service updates customer claim information.

3.6.4.2.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
claimUpdateRequest		
claimId	long	Claim identifier
claimText	string	Claim description (free text)

3.6.4.2.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
claimResponse		
result	integer	Result (see Appendix D.1)
claimId	long <u>Cardinality:</u> 0..1	Claim identifier created by CBO
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.4.3 Customer claim list enquiry

This web service returns the list of claims registered for a customer. It includes the closed claims.

3.6.4.3.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
claimListEnquiryRequest		
customerId	long	Customer identifier

3.6.4.3.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
claimListEnquiryResponse		
result	integer	Result (see Appendix D.1)
claimList	<u>Cardinality:</u> 0..1	
claim	<u>Cardinality:</u> 1..n	
claimId	long	Claim identifier
claimText	string	Description of the claim: initial text + cumulated updates
state	integer	Claim state. See Appendix D.5
lastUpdate	string	Last update of the claim
claimDate	dateTime	Claim creation date
lastUpdateDate	dateTime	Last update of the claim
closedDate	dateTime	Date of claim closure
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.4.4 Customer claim record enquiry

This web service returns information for a claim record.

3.6.4.4.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
claimRecordEnquiryRequest		
claimId	long	Claim identifier

3.6.4.4.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
claimRecordEnquiryResponse		
result	integer	Result (see Appendix D.1)

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
claim	<u>Cardinality:</u> 0..1	
claimId	long	Claim identifier
claimText	string	Description of the claim: initial text + cumulated updates
state	integer	Claim state. See Appendix D.5
lastUpdate	string	Last update of the claim
claimDate	dateTime	Claim creation date
lastUpdateDate	dateTime	Last update of the claim
closedDate	dateTime	Date of claim closure
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)

3.6.5 AV2 SAM QUOTA

These web services allow reloading the AV2 SAM quotas.

WSDL	samquotaupdate.wsdl
XSD	samquotaupdate_schema1.xsd
Web service	Operation message name
Update AV2 SAM quota	updateSamQuota

3.6.5.1 Update AV2 SAM quota

This web service reloads an AV2 SAM quota.

3.6.5.1.1 Request parameters

REQUEST parameters Field name	Type Cardinality (1 by default)	Description
updateRequest		
equipmentModelId	integer	For the moment, it has the same value as the equipment type: See Appendix C.1
equipmentId	long	Unique serial number assigned to this equipment.
samSerialNumber	long	SAM serial number
currentSamQuota	long	SAM Quota
currentSamValue	long	Counter incremented at each authentication

3.6.5.1.2 Response parameters

RESPONSE parameters Field name	Type Cardinality (1 by default)	Description
updateResponse		
Result	integer	Result (see Appendix D.1D.7)
errorDescription	string <u>Cardinality:</u> 0..1	Present only when Result=0 (System Error)
updateKey	base64Binary <u>Cardinality:</u> 0..1	Cryptogram for updating AV2 SAM Key Usage Counter (KUC)
newQuota	long <u>Cardinality:</u> 0..1	New SAM quota value

CONFIDENTIAL

3.7 TIME SYNCHRONIZATION

The date/time information is essential in the system. Many operations use or store date/time. This is the reason why the whole system must be synchronized with the same date & time. Thus the vendor system shall synchronize with the CBO time server based on NTP protocol.

The CBO deploys a NTP server that the Level3 System shall synchronize with. Then the Level3 System is responsible for the time synchronization of the devices that it manages. The time synchronization uses the following details:

- The version of NTP protocol is version 3, see: <http://www.eecis.udel.edu/~mills/database/rfc/rfc1305/>
- Time Reference: Local Time
- This local time will be used to date all the events generated by the system and the events of the cards
- Manage the time Changes: automatically

The CBO NTP server is synchronized with existing central metro NTP server.

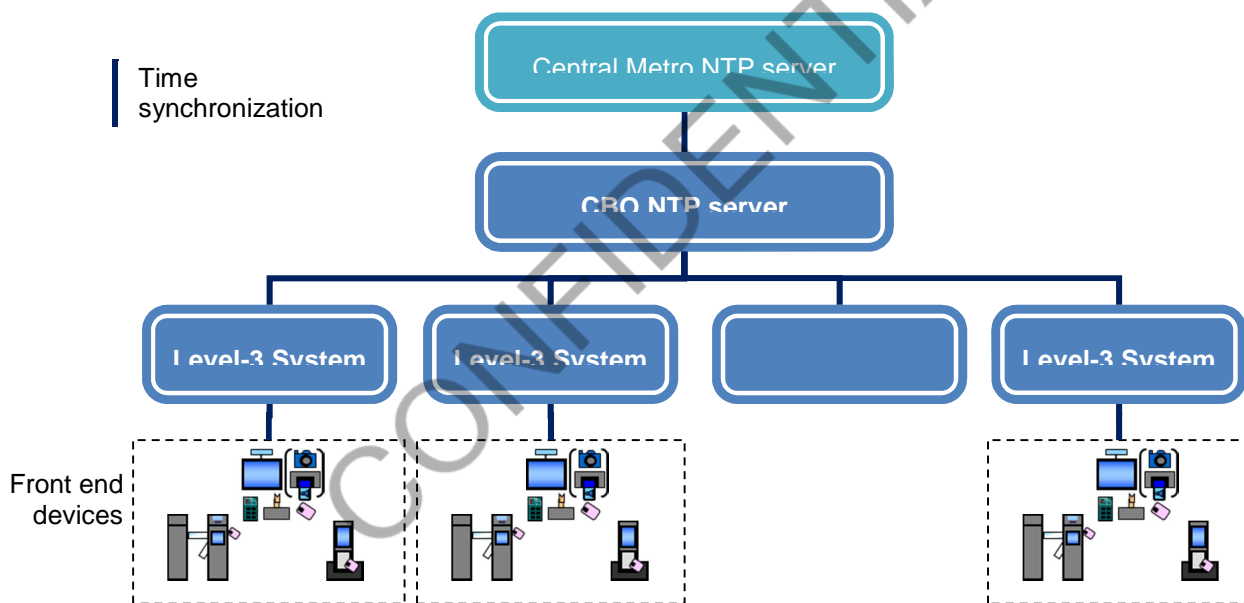


Figure 8 : Time synchronization

APPENDIX A: CARD DATA STRUCTURES

The card data structures contain an extract of the data stored on the card. These structures are reused from the Fare Media Electrical Data Layout and provided in some transactions events.

Refer to Fare Media Electrical Data Layout specifications for further details.

A.1 DATA TYPES

The card data layout implies some constraints on the ranges of some specific fields:

Field	Range	Description
contractTariff	0..65535	It identifies uniquely the combination of product and profile.
company	0...65535	Identifier of a company (service provider).
device DevcelD Device serialNumber	$0 \dots 2^{24} - 1$ (16777215)	Device identifier
eventCode	0..31	Event code: <ul style="list-style-type: none"> • Selection = 0: points any contract to be validated next time. • Entry = 1: journey first Check-in • Exit = 2: any Check-out but Service Disruption • Inspection = 4: inspection mark • Interchange = 6: any transfer Check-In
locationId	$0 \dots 2^{16} - 1$	Network registered location (bus stop, station, zone, area radius, etc.).
transportMode	0...31	Transport mode: <ul style="list-style-type: none"> • notSpecifiedFurther (0) • urbanBus (1) • parking (8) • interUrbanBus (2) • taxi (9) • lightTrain(metro) (3) • highSpeedTrain (10) • tram (4) • ruralBus (11) • train (5) • express (12) • waterborne(ferry) (6) • regionalTrain (19) • toll (7) • intercity (20)

Field	Range	Description
passengerClass	0..15	Travel class: <ul style="list-style-type: none"> Unspecified = 0 (unknown): Not specified further. FirstClass = 1 (first): first Class (Gold). SecondClass = 2 (second-standard-traveller): second Class (Regular). Others: Reserved for Future Use.

A.2 CARD HEADER

Field name	Type / Cardinality (1 by default)	Description
cardHeader	<u>Cardinality:</u> 0..1	Card header.
country	integer	Identifies the country from which the card format is being used, according to ISO 3166-1 alpha3: 818 (EGY).
format	integer	Identifies the card electrical format being used. This field, in addition with the country code gives information about the card format i.e. what are the following fields implemented in the card. Currently 0.
directory	<u>Cardinality:</u> 0..1	
applicationId	integer	The identifier of the application, as determined by the application owner. Only one of each Application Id is present on the card.
applicationLayout	integer	This field gives information about the Application Layout.
applicationKeyVersion	integer	Version of the security key set currently stored within the application. The version is managed in a cyclic way wrapping to 0 after 15 (modulo 16).
cardTagId	long	The engraved card ID.
cardIdIteration	integer	0 means the card is very new. Each time the card is recycled; this number is increased by one. Personalized cards cannot be recycled; Card Iteration will always be 0.
cardArtwork	integer	Specifies which card artwork (graphical mask) has been printed.
cardValidityEndDate	date	Specifies the last day on which the card is still valid.

A.3 TPURSECONTEXT

Field name	Type / Cardinality (1 by default)	Description
tPurseContext		
status	integer	TPurse status <ul style="list-style-type: none"> Enabled = 0 (ok) Initialised = 1 (blocked-undefined) Deactivated = 4 (blocked-refunded) Blocked = 9 (blocked-misuse)
tPurseAutoReloadActive	boolean	Auto-Reload feature is deactivated when FALSE
sequenceNumber	integer	Specifies the current Transaction Sequence Number (TSN).
unblockingNumber	integer	Specifies the Unblocking Sequence Number (USN).

A.4 APPCONTEXT

Field name	Type / Cardinality (1 by default)	Description
appContext		
appStatus	integer	Application status <ul style="list-style-type: none"> Enabled = 0 (ok) Initialised = 1 (blocked-undefined) Deactivated = 4 (blocked-refunded) Blocked = 9 (blocked-misused)
appUnblokNum	integer	Application unblocking sequence number
appLastSrvOrd	integer	Interoperated Action List sequence number that defines the last served order for this Application.
appSeqNum	integer	Application transaction sequence number. TSN starts with zero (initialisation), and increases by one each time a transaction is made on the Application.

A.5 CONTRACTLISTS

Field name	Type / Cardinality (1 by default)	Description
contractLists		
length	integer	Number of contracts (0..8)
contractList	<u>Cardinality:</u> 0..8	Product instance
contractTariff	integer	Identifies the product for which a product instance is stored on the card. This product must only be the current active contract used for the transaction.

Field name	Type / Cardinality (1 by default)	Description
isUsed	boolean	Set to false when the Product instance is sold, and raised when used (i.e. at the first Check-In)
pointer	integer	Points the Contract record (1..8) in the media structure.

A.6 CONTRACT

Field name	Type / Cardinality (1 by default)	Description
contract		
contractPointer	integer	Points the Contract record (1..8) in the media structure.
fixedPart		
contractNetWorkId	integer Cardinality: 0..1	Network ID. When absent, it is assumed to be the Application Network.
provider	integer	The field is the Product Owner identifier. See Appendix C.9
contractTariff	integer	Identifies the product for which a product instance is stored on the card. Range 0..255 is reserved for magnetic fare products
serialNumber	integer	This field is the number of the Product on this card. The association of this field with the Card Serial number identifies the Product uniquely.
passengerTotal	integer Cardinality: 0..1	(0..255) Group validation count allowed.
priceAmount	integer Cardinality: 0..1	Contract sale amount
validityStartDate	date	The first day the contract is valid.
validityStartTime	time Cardinality: 0..1	The first second at which the product becomes valid on StartDate.
validityEndDate	date	The last day the contract is valid.
validityEndTime	time Cardinality: 0..1	The last second of the validity end date at which the contract is valid.
journeyType	integer Cardinality: 0..1	If =0 : GeographicalValidty = OriginDestination If =1 : GeographicalValidty = CentreAndRadius If =2 : GeographicalValidty = ListValidity (ZoneMap)
journeyOrigin	integer Cardinality: 0..1	May be any topologic point or zone were the Product is allowed.

Field name	Type Cardinality (1 by default)	Description
journeyDestin	integer <u>Cardinality:</u> 0..1	May be any topologic point or zone were the Product is allowed.
journeyCentre	integer <u>Cardinality:</u> 0..1	May be any topologic point or zone were the Product is allowed.
journeyRadius	integer <u>Cardinality:</u> 0..1	(0..255) May be any distance relative to Origin. The radius unit is defined by Contract Tariff and/or Origin: e.g. kilometres, inter-stations, zones, etc.
journeyZoneMap	integer <u>Cardinality:</u> 0..1	16 zones flagged individually are allowed (true) or forbidden (false) for the Product.
saleDate	date <u>Cardinality:</u> 0..1	Sale date
saleTime	time <u>Cardinality:</u> 0..1	Sale time
saleAgent	integer	Product Retailer Identification. See Appendix C.9
saleDevice	integer	Identify the device that performed the sale of the Product.
saleSAM	long <u>Cardinality:</u> 0..1	Identify the ticketing SAM that performed the sale of the Product.
suspensionStartDate	dateTime <u>Cardinality:</u> 0..1	The first day the Product is suspended.
suspensionEndDate	dateTime <u>Cardinality:</u> 0..1	The last day the Product is suspended.
contractUnblocking	integer	Defines the Contract Unblocking Sequence Number.
varPart		
counterStatus	integer <u>Cardinality:</u> 0..1	Product status: <ul style="list-style-type: none"> Initialised = 0 (never-validated) Enabled = 2 (validated) Blocked = 19 Terminated = 11
counterEndDate	date <u>Cardinality:</u> 0..1	The last day the sliding contract is valid.
counterEndTime	time <u>Cardinality:</u> 0..1	The last second of the last day the sliding contract is valid. This field may be used for short period contract

Field name	Type / Cardinality (1 by default)	Description
counterPeriods	integer <u>Cardinality:</u> 0..1	(0..1023) Number of periods remaining before exhaustion of the Product.
periodJourneys	integer <u>Cardinality:</u> 0..1	(0..1023) Number of valid remaining journeys for the current period.

A.7 EVENTLOG

The event log contains some information stored on the media for a validation event.

Field name	Type / Cardinality (1 by default)	Description
eventLog		
dateStamp	date	Date when this event occurred.
timestamp	time	Time when this event occurred.
transportMode	integer <u>Cardinality:</u> 0..1	Transport mode. Refer to §A.1
eventCode	integer <u>Cardinality:</u> 0..1	Event code. Refer to §A.1
serviceProvider	integer	Machine owner. See Appendix C.9
serialNumber	integer	Sequence number that identifies the event.
origin	integer <u>Cardinality:</u> 0..1	Location where the journey started, if different from the event location.
locationId	integer	Location where the event took place.
device	integer	The device that generated the event
SAM	long <u>Cardinality:</u> 0..1	Identify the ticketing SAM that performed the event.
vehicleId	integer <u>Cardinality:</u> 0..1	Vehicle id in which the event took place
elapsedJourneyMinutes	integer <u>Cardinality:</u> 0..1	Effective minute count elapsed between the first Check-in (start of current journey) and current date/time stamp (if a last event referring to this contract is available, otherwise not relevant).
passengerTotalMax	integer <u>Cardinality:</u> 0..1	(2..255) The group validation maximum count.

Field name	Type / Cardinality (1 by default)	Description
passengerTotalCount	integer <u>Cardinality:</u> 0..1	(0..255) Current validation count, starting from one when checking in or out in a new location.
priceAmount	integer <u>Cardinality:</u> 0..1	The debited amount related to current travel (initial fee on Check-in, the effective amount on Check-out).
contractPointer	integer <u>Cardinality:</u> 0..3	The used Contract record.
numPenalty	integer <u>Cardinality:</u> 0..1	
checkOutExpected	boolean <u>Cardinality:</u> 0..1	True if the journey mode is Check-In/Check-Out
firstCheckIn	boolean <u>Cardinality:</u> 0..1	If true indicates the first Check-In
contractRollBack	integer <u>Cardinality:</u> 0..1	Contains the value subtracted from number of remaining periods in the Product variable part.
passengerClass	integer <u>Cardinality:</u> 0..1	Passenger class. Refer to §A.1

A.8 ENVIRONMENT

Field name	Type / Cardinality (1 by default)	Description
environment		
transportAppVersion	integer	Transport Application Version, the currently 0.
appNetwork	integer	The Transport Application base network, which is the Application default network for any object when missing.
deposit	integer	Deposit set when card is sold.
appRetailer	integer	Identifies the application retailer. See Appendix C.9
appEndDate	date	Last day the Application instance is still valid. It is at most the cardValidityEndDate.
appType	integer	Application type: <ul style="list-style-type: none"> 0→ Anonymous travel card, no other data; 1→ Personnalized travel card, no other data; 2→ Test card, no other data; 3→ Intervention card 4..15→ R.F.U.

A.9 TPURSELOG

The TpurseLog contains some information stored on the media for a Tpurse add-value event.

Field name	Type Cardinality (1 by default)	Description
tPurseLog		
dateStamp	date	Date when this event occurred.
timestamp	time	Time when this event occurred.
serviceProvider	integer	Equipment owner. See Appendix C.9
serialNumber	integer	Sequence number that identifies the event.
locationId	integer	Location where the event took place.
device	integer	The device that generated the event.
SAM	long	Identify the ticketing SAM that performed the event.
priceAmount	integer	Specifies the amount that was added to the t-Purse.

A.10 SPECIALEVENTLOG

The special contains some information stored on the media for the following events:

- Penalty event

Field name	Type Cardinality (1 by default)	Description
specialEventLog		
dateStamp	date	Date when this event occurred.
timestamp	time	Time when this event occurred.
serviceProvider	integer	Equipment owner. See Appendix C.9
serialNumber	integer	Sequence number that identifies the event.
locationId	integer	Location where the event took place.
device	integer	The device that generated the event.
SAM	long <u>Cardinality:</u> 0..1	Identify the ticketing SAM that performed the event.
passengerTotalMax	integer <u>Cardinality:</u> 0..1	(2..255) The group validation maximum count.
passengerTotalCount	integer <u>Cardinality:</u> 0..1	(0..255) Current validation count, starting from one when checking in or out in a new location.
contractPointer	integer <u>Cardinality:</u> 0..3	The used Contract record.
numPenalty	integer <u>Cardinality:</u> 0..1	

A.11 PREVIOUS EVENT LOG

The previous event log contains some information stored on the media for the previous validation event.

Field name	Type / Cardinality (1 by default)	Description
previousEventLog		
dateStamp	date	Date when this event occurred.
timestamp	time	Time when this event occurred.
eventCode	integer <u>Cardinality:</u> 0..1	Event code. Refer to §A.1
origin	integer <u>Cardinality:</u> 0..1	Location where the journey started, if different from the event location.
locationId	integer	Location where the event took place.
device	integer	The device that generated the event
elapsedJourneyMinutes	integer <u>Cardinality:</u> 0..1	Effective minute count elapsed between the first Check-in (start of current journey) and current date/time stamp (if a last event referring to this contract is available, otherwise not relevant).
passengerTotalMax	integer <u>Cardinality:</u> 0..1	(2..255) The group validation maximum count.
passengerTotalCount	integer <u>Cardinality:</u> 0..1	(0..255) Current validation count, starting from one when checking in or out in a new location.
priceAmount	integer <u>Cardinality:</u> 0..1	The debited amount related to current travel (initial fee on Check-in, the effective amount on Check-out).
contractPointer	integer <u>Cardinality:</u> 0..3	The used Contract record.
checkOutExpected	boolean <u>Cardinality:</u> 0..1	True if the journey mode is Check-In/Check-Out
firstCheckIn	boolean <u>Cardinality:</u> 0..1	If true indicates the first Check-In
contractRollBack	integer <u>Cardinality:</u> 0..1	Contains the value subtracted from number of remaining periods in the Product variable part.
counterPeriods	integer <u>Cardinality:</u> 0..1	(0..1023) Number of periods remaining before exhaustion of the Product.

Field name	Type Cardinality (1 by default)	Description
periodJourneys	integer <u>Cardinality:</u> 0..1	(0..1023) Number of valid remaining journeys for the current period.
tPurseValue	integer	Value of TPurse

A.12 CARD HOLDER

Field name	Type Cardinality (1 by default)	Description
holder	<u>Cardinality:</u> 0..1	Card holder.
birthDate	date	Card holder birthdate
language	string	Preferred language (ISO639-2/B codes 'ara' Arabic or 'eng' English)
profileData	<u>Cardinality:</u> 0..1	
number	integer	Customer Profile (see Appendix C.10)
endDate	date <u>Cardinality:</u> 0..1	End of profile validity

APPENDIX B: CLARIFICATIONS ON XML DATA

- The Date format in all xml is: dd/mm/yyyy
- The Time format in all files is: hh24:mm:ss
- The prices are multiplied by 10^2 to avoid any decimal.
- Xml and xsd files are encoded in UTF-8
- Integer fields in this document can be encoded into xs:int or xs:integer for the translation in XSD. The integer fields are anyway limited to 32 bits.
- All amounts in batch interface to CBO (§3.4) are in 0.01 EGP unit

CONFIDENTIAL

APPENDIX C: INTERFACE CODES

C.1 EQUIPMENT TYPE CODE

The equipment types are configured at CBO. The following table defines the initial code list:

Value	Equipment	Comment
1	AG	Contactless Automatic gate
2	TOM	Contactless Ticket Office Machine
4	SCU	Station Control Unit
5	PCM / PVU	Portable Checking Machine / Portable Verifying Unit
6	CIS	Card Initialization System
10	AG	Dual Automatic gate
12	TOM	Dual Ticket Office Machine

C.2 ACCOUNTING OPERATION TYPE CODE

The following table defines the codes for accounting operations:

Value	Description
1	Charge
2	Charge cancellation
3	Recharge
4	Recharge cancellation
5..10	Reserved
11	Pass Refund
12	Tpurse Refund
13	Card Refund
14	Activation
15	Charge and Activation
16	Contract reconstruction
17	Tpurse reconstruction
18	Reconstruction Activation

C.3 VALIDATION TYPE CODE

The following table defines the codes for the validation type:

Value	Description
0	Refused
1	First Check-in
2	Group Check-in
3	Close Check-in at check-in

Value	Description
4	Close Check-out at check-in
5	First Check-out
6	Group Check-out
7	First Free Exit
8	Group Free Exit
9	First Penalty Exit
10	Group Penalty Exit
11	CSC Failed reading
12	Entry penalty because of missing exit validation
13	Entry penalty because of missing entry validation

C.4 PENALTY TYPE CODE

The penalty types are configured at CBO. The following table defines the initial code list:

Value	Description
1	Standard penalty
	Other reasons not yet defined

C.5 PAYMENT MEANS

The payment means are configured at CBO. The following table defines the initial code list:

Value	Description
1	Cash
2	Credit
3	EPurse
10	TPW

C.6 AGENT PROFILES

The agent profiles are configured at CBO. The following table defines the initial code list per device type:

AG

Code	Name
1	<i>Reserved</i>
2	Maintenance

TOM

Code	Name
1	Sales
2	Maintenance
3	Supervisor

PCM / PVU

Code	Name
1	Sales
2	Maintenance
3	Supervisor
4	Inspection

C.7 ALARM CODIFICATION

The following table defines the alarm codes for front end devices:

Code	Record
0	Communication failure
1	Out Of Service
2	Minimum stock reached
3	Zero stock reached
4	Access. Identification attempts exceeded
5	Power failure
6	No agents configuration
7	No topology configuration
8	No time parameters configuration
9	No fare parameters configuration
10	Intrusion
11	SAM reader alarm
12	CSC reader alarm
13	Data memory alarm
14	Time synchronization alarm
15	EOD alarm (activation or presence)
16	Ticketing keys alarm
17	Printer alarm
18	Card printer alarm
19	Gate door alarm
20	MT reader alarm
21	MT reader Jam alarm
22	MT reader Consecutive Jams alarm
23	Rejected ticket alarm
24	MT dispenser alarm
25	Camera alarm
26	PID alarm

C.8 BLACKLISTING/BLOCKING REASON

The blacklisting/blocking reasons are configured at CBO. The following table defines the initial code list:

Code	Description
1	Product abuse
2	Card Lost
3	Card Stolen
4	Damaged
5	Faulty

C.9 SERVICE PROVIDER

The service providers are configured at CBO. The following table defines the initial code list:

Code	Description
1	CM (for L1, L2, L3)
2	ENR (for future)
3	

C.10 CUSTOMER PROFILES

The following table defines the customer profiles:

Code	Description
0	Non Profile
1	Adult
2	Child
3	Student
4	Pensioner
5	Handcam
9	ECM employees
10	Military
14	ENR employees
21	Police
72	Public sector
73	Family
74	ECM worker wives
75	ECM worker children
76	Gold agents
77	Old soldiers
78	VIP
79	Special Fare
80	Retired ENR Workers

Code	Description
81	Administrative Control

C.11 VALIDATION ERRORS

The following table defines the validation errors:

Code	Description
1	Transaction with error or incomplete (unable to execute the operation)
2	Error for reading or writing the card

CONFIDENTIAL

APPENDIX D: SPECIFIC CODES FOR ONLINE WEB SERVICES

D.1 WEB SERVICE RESPONSE CODES

The following table defines the web service response codes:

Code	Description
0	System Error
1	OK
2	The Request does not match the XML Schema
3	Document not found
4	Card not found
5	Customer not found
6	Claim not found
8	Only the claims with status 'Open' can be modified
9	The customer has not claims
10	The user already exists
14	The card belongs to other user

D.2 OPERATION TYPE CODE

The following table defines the operation types performed on front end device:

Code	Description
1	Charge
2	Charge cancellation
3	Recharge
4	Recharge cancellation
5	Entry validation
6	Exit validation
8	Card Blocked
11	Pass Refund
12	Tpurse Refund
13	Card Refund
14	Activation
15	Charge and Activation
16	Contract reconstruction
17	Tpurse reconstruction
18	Reconstruction Activation
19	Card UnBlocked

Code	Description
20	Product UnBlocked
21	Tpurse UnBlocked

D.3 CARD DISABLING REASON CODES

The following table defines the codes for disabling a CSC:

Code	Description
0	Lost
1	Stolen
2	Damaged
3	Faulty

D.4 CUSTOMER IDENTIFICATION TYPE

The following table defines the types of customer identification:

Code	Description
1	Identity Card
2	Passport
3	Work permit number

D.5 CLAIM STATE

The following table defines the claim states:

Code	Description
1	Open (The initial status)
2	Pending (already being solved by a operator)
3	Resolved (Pending the user confirmation)
4	Closed

D.6 SERVER ERRORS

Additional errors out of the web service control can happen in some specific cases, like server internal errors, wrong data types, etc. In these cases the server responses with a defined structure as follows:

```
<soap:Fault>
  <faultcode>soap:Client</faultcode>
  <faultstring>Unmarshalling Error: Not a number: 3q4</faultstring>
</soap:Fault>
```

The structure of these responses is always the same, it only changes the content of the <faultcode> and <faultstring> fields. In the above example the error comes because of the client has placed a string value in a numeric field.

D.7 AV2 SAM WEB SERVICE RESPONSE CODE

Code	Description
0	System error
1	OK
2	Request does not match the XML Schema
3	Threshold not reached
4	Sam blacklisted
5	Equipment changed
6	Sam secret key not found
7	Updates disabled for this SAM
8	Sam is not-activated
9	KUC period is not reached
20	SSM Signature key not found
21	SSM Certificate not found

CONFIDENTIAL

APPENDIX E: XSD & WSDL

E.1 CBO BATCH INTERFACE



XSD_CBOBatchInterface A1.zip

E.2 ONLINE WEB SERVICES



XSD_WSDL_CBOOnlineServices A1.zip

E.3 FILE SIGNATURE SCHEMA

XML data files exchanged between CBO and level3 systems shall contain a digital signature at the end of the file which follows the next schema definition: <http://www.w3.org/TR/2002/REC-xmlsig-core-20020212/xmlsig-core-schema.xsd>

CONFIDENTIAL

APPENDIX F: COMMENT SPREAD SHEET

Revision --

No	Reader	Page	Chapter	Comment	Date	Thales answer		Nat reply	state	Update tracking
1	NAT	11	1.1 Identification	Installation plan and installation test procedure not part of training documents, add them to testing documents and add PAT / SAT to test documents		Agreed. Document will be updated.		ok		§1.1
2	NAT	12	1.2.1 Project Over ground	CBO shall manage the 3 lines currently and future lines as ... lines 4, 5 and 6 and other transportation means like Tram, train and buses.		For clarity, 1.2.1 is referring to the scope of the 850 gates project which is broken down into : 1. Phase 1: the upgrade of lines 1 & 2. 2. Phase 2: the delivery of the CBO. As per final clarifications : 'The CBO is designed to managed all transactions from Line 1, Line 2 & Line 3 (all 4 phases). Its is also scalable in order to manage further transactions coming from coming Lines subject to purchasing extra data processing capability'		Software should be modular for any other transport ticketing System		Refer to System SSS release -C § 3.6 interoperability use cases have been added and are consistent with the implementation of the UIA.
3	NAT	15	1.4 Migration Demonstration	There is will be quit modification to at the equipment, SCU and LCU because interface document still not updated and may be related some comments raised during integration and interoperability.		Confirmed that compatibility with existing CCU interface is not 100% and there might be minor impact on existing equipment. A typical example is the name space in the XML schema that is updated.		ok		§1.4

4	NAT	22	3.2.2; Certificate renewal	The request of the new certificate must be prepared by automatic/Manual way.	The generation of the certificates for the CBO servers is a manual process. There is no real operational impact here since the number of servers is limited. The signature of the subordinate certificates is also a manual process. There is a single subordinate CA per line (potentially 3rd party providers). Then the subordinate CA is in charge of certificate distribution and signature for all equipment of the line.		Why it is not also automatic generation?	§3.2.1 The automatic distribution of certificates and key pairs requires a trusted relationship between the client and the CA and an enrolment process. A manual process is much simpler.
5	NAT	24	5.2.4 External Interfaces	Interface with Thales equipment Please clarify main power is 230V AC or 220V AC.	(Comment related to gate HRS)			
6	NAT	26	5.3 Gate Barrier Mechanism	Operating Environment: Operating Temperature range shall be from 0°C to 60°C not 0°C to 45°C	(Comment related to gate HRS)			
7	NAT	32	3.4.3.2 Time parameter file	The first Day of the week in Egypt is Sunday not Monday	Will be corrected.		ok	§3.4.3.2
8	NAT	35	5.9 Batteries	The temperature (charge and storage) shall be 60°C.	(Comment related to gate HRS)			
9	NAT	39	5.13.3 Technical Specification	Operation temperature shall be from -10°C to +60°C	(Comment related to gate HRS)			
10	NAT	41	3.4.3.4	What is the capability to implement daily Pass Ticket, weekly Pass Ticket, and monthly Pass Ticket on the MT?	This is part of the existing MT pass implementation. Validity can be 1 day, 1 week, 1 month, 3 months (season pass)		ok	§3.4.3.4

11	NAT	41	3.4.3.4	The limited 'number of zones in MT is 8 floating or fixed zones, please clarify.	This is 8 floating zones (single ride tickets). Note that for passes, there are 3 linear zones defined by entry and exit stations.	ok	\$3.4.3.4
12	NAT	44	3.4.3.5	The blacklist must divided into two category primary blacklist (from 0 to 100000) and secondary blacklist	Refer to item # 56 on CBO SSS review. This point is under discussion.	open	
13	NAT	45	3.4.3.5	what do you mean by card range card Range Cardinality: 0..100 ? If the number of cards needed to	100 is the existing limit for the CCU interface. This can be extended but there might be a compatabily problem with the existing 3rd party equipment (refer to point #3)	Clarify the compatabily problem	\$3.4.3.5. At most 100 ranges. This is the limitation in the existing CCU interface. To check with the 3rd party vendor if more ranges can be added.
14	NAT	50	3.5.2.1 Events	Data collection complete and End of file is a mandatory event not option.	There are certain situations (typically equipment failure and replacement) when those events will not be generated. This is the reason why they are marked as optional.	not ok, please refer to remarks recorded in phase 1 and phase 2, concerning the comment (data collection complete and end of file is a maindatory event)	\$3.5.2.1 When a defective equipment is replaced, the last data elements are likely to be not uploaded. So in this case, the data collection complete is not generated in order to highlight such a situation. This is the reason why data collection complete is optional. Confirmed that end of file is not optional.

15	NAT	50	3.5.2.1 Events	Alarm for SAM Quota threshold shall be added.	We don't think there is a need for such alarm since the equipment anticipates the reloading request (the gap used for the reloading request is part of the fare parameters, the current value is 400). So basically, the SAM will never be locked (provided the equipment is connected to the communication network)	Answer refused	Note that there is a dedicated SAM Quota Updated event.
16	NAT	51	3.5.2.1.1 Event Header	Clarify the meaning of "1" by default in type cardinality for example station and line putted as "1" or No. of line or station.	This means that the field is mandatory (constraint in the XSD schema).	ok	§3
17	NAT	52	3.5.2.1.2	Define the type and description of SAM2 and SAM3.	In the existing implementation we have SAM1 and SAM2. This is in case there are 2 AV2 SAMs in the equipment. Note that in Thales equipment there is a single SAM.	more clarification is needed (answer is a little bit vague)	§3.5.2.1.2 This is typically the case of a bi-directional gate having 2 readers. This is not the case of Thales gate.
18	NAT	53	3.5.2.1.3 Alarm	Clarify what's meant by T&Z in opening date	This is Xml dateTime encoding. - T indicates the start of the time section. - Z indicates the time zone. Here there is no offset. UTC time is used.	ok	§3.5.2.1.3
19	NAT	53	3.5.2.1.4 Card Stock update	This event shall be generated automatically when TOM reached to its threshold for card stock not manual requesting> by operator.	In case of minimum stock reached, the TOM raises automatically an alarm (refer to C.7)	ok	§3.5.2.1.4
20							
21	NAT	56	3.5.2.1.8 Penalty	The gate shall generate penalty event also according to business rules & operational data.	In case of a CSC holding a penalty event (special event), the gate at exit generates a validation event of	ok	§3.5.2.1.8

					type penalty (refer to C3). Precision will be added.				
22									
23	NAT	71	3.6.2.3 Customer update	The following parameters shall be fixed (first name, last name, type of document, birth date and No. of document).	Understood that the need is to forbid the modification of those fields in the 'update' web service. We think this would be an issue in case of error at creation (no possibility to modify the customer record)		ok		No update needed
24	NAT	71	3.6.2.3 Customer update	The national ID of customer shall be limited to 14 digits and parameterized at CBO level to control the No. digits if changed from government.	The limitation is checked at the TOM. There is no limitation at the Unified Interface in order to keep the CBO implementation open.		NAT asked , in case of the national ID digits is increased over the 14 digits at the level of interface that necessitates to modify TOM software		No update, comment below In case the number of digits of the national ID were to be increased, the TOM software would have to be updated since this is checked there.
25	NAT	78	3.6.4.4 Request Parameter s	Reason for disabling card shall be shown in reports or MMI by text not integer.	Confirmed. However in the web service this is a code		ok		No update needed.

26	NAT	85	A.1 Data Types	Remove limitation of amount field because the annual adult cards more than 650 pounds.	The limitation will be removed here. Note: This field mirrors the card data layout. The limitation in the layout is a separate issue not linked to the Unified Interface .	ok	§A.1 Correction compared to the initial answer. - At first the transaction amounts (accounting events) have not this limitation. - Here, since the field mirrors the card layout and the layout cannot be extended, this cannot be changed. The issue has to be addressed globally, likely to be the change of the unit price (0.1 EGP instead of 0.01) This is a separate discussion.
27	NAT	85	A.1 Data Types	Remove limitation of card balance field amount in case of increasing t-Purse charge.	The limitation will be removed here. Note: as the previous point, the limitation was mirroring the card layout (T-Purse Value field). However here, the range in the layout is [-327 EGP, 10158 EGP]. We suggest that this is acceptable.	ok	Limitation not removed (card filed mirroring), no issue since we go up to 10158 EGP.
28	NAT	87	A.4 App Context	Take into consideration for application and transaction sequence No. to follow the existing interoperability documents.	This is included (appSeqNum field, i.e. TSN)	ok	No update needed.

29	NAT	89	A.6 Contract	For journey zone map take into consideration the total No. of zones after opening 6 metro lines and interface with other transportation means.	The journey zone map (layout field) is not used in the existing metro fare structure. As a reminder, we have the following fare methods for CSC : - 1. Free - 2. Flat fare - 3. Station origin/destination - 4. Fixed zone center/radius - 5. Floating zones - 6. Fixed zone origin/destination.	NAT asked, what about fare method for magnetic?	Refer to remark #11.
----	-----	----	--------------	--	---	---	----------------------

Revision –A

N°	Reader	Page	Chapter	Comment	Date	Answer	State	Update tracking
3	NAT	15	1.4 Migration Constraints	There is will be quit modification toat the equipment, SCU and LCU because interface document still not updated and may be related some comments raised during integration and interoperability. [30/7/2017] remove statement "The main migration constraint is that there will be no change at the equipment, the SCU and the LCU (level 3-1 systems); thus the CBO interface shall be compatible with existing CCU interface"	30/07/17	Confirmed that compatibility with existing CCU interface is not 100% and there might be minor impact on existing equipment. A typical example is the name space in the XML schema that is updated. Sentence is removed		§1.4

4	NAT	22	3.2.2; Certificate renewal	The request of the new certificate must be prepared by automatic/Manual way. [30/7/2017] The system must support automatic and manual	30/07/17	Referring to open point #19 and decision recorded on the 21/11/17. - The point is related to the TOM certificate. At present installation is manual. In phase 1 of the project this will continue like this with the CCU certificate. For the CBO deployment, the CCU certificate will still be used in order to smooth the process. - For the TOM that will be installed later, the certificate may be automated by including the TOM in the LCU domain. However, Thales has to check the feasibility - Clarify whether certificate is required for gate -> NO as the gate does not access the CBO customer and card services. Precision is added in the document.	\$3.2.2
15	NAT	50	3.5.2.1 Events	Alarm for SAM Quota threshold shall be added. [30/7/2017] This is mandatory request	30/07/17	Referring to open point #22 and decision recorded on the 21/11/17. - Alarm does exist at the TOM - no change required	na
23	NAT	71	3.6.2.3 Customer update	The following parameters shall be fixed (first name, last name, type of document, birth date and No. of document). [30/7/2017] This is Mandatory request	30/07/17	Confirmed that these parameters are fixed cannot be changed after the customer creation. This is managed at the TOM. Refer to comment # 20 on the TOM SSS.	3.6.2.3
26	NAT	85	A.1 Data Types	Remove limitation of amount field because the annual adult cards more than 650 pounds.	30/07/17	Agreed. There is no special constraint at the Unified interface. However the management of the card layout is a separate issue	A.1, B
27	NAT	85	A.1 Data Types	Remove limitation of card balance field amount in case of increasing t- Purse charge.	30/07/17	[28/08/17]. Thales Same as #26.	A.1, B

END

CONFIDENTIAL