	<p>IST-2001-34340</p> <p>Draft of D 4.3:</p> <p>Internal Functional Specification</p>
Distribution List:	Project Partners
Author:	WP4 Members
Authorised by:	---
Date of Issue:	26.11.2002
Issue:	Version 1.0
File name:	D4.3-draft-1.0.doc
Work package:	WP4 Specifications
Deliverable Number:	4.3
Deliverable Type:	Public
Deliverable Nature:	Specification
Total Number of Pages:	102
Contact Details for EUPKI:	christoph.schiller@qi-de.de martin.rauch@qi-de.com

Copyright © The EU-PKI Consortium

0 Document Control

<i>Issue</i>	<i>Date of Issue</i>	<i>Comments</i>
0.01	14 Oct 2002	First version, sceleton and specification of CSP modules
0.02	21 Oct 2002	specification of some KGS modules added
0.03	24 Oct 2002	Function table inserted
0.04	05 Nov 2002	specification of CA-DB, some CA modules and some KGS modules added
0.05	11 Nov 2002	specification of CA-DB and some CA/KC/KGS modules added or changed
0.06	16 Nov 2002	specification of KGS-DB
0.07	19 Nov 2002	added RA module specification
0.08	20 Nov 2002	added RA-DB diagram
0.09	21 Nov 2002	changes after review meeting
0.10	26 Nov 2002	Corrections
1.0	02 Dec 2002	Validated Version

0.1 Abstract

This document D4.3 is the third of three deliverables of the specification phase of the EU-PKI project.

D4.3 contains the internal functional specifications for each component of the EU-PKI system.

D4.3 is based on D4.1 which specifies the global architecture of the EU-PKI system and D4.2 which specifies the external communication interfaces.

0.2 Table of Contents

0	DOCUMENT CONTROL	2
0.1	ABSTRACT.....	2
0.2	TABLE OF CONTENTS.....	3
1	INTRODUCTION	6
1.1	FUNCTION TABLE.....	6
1.2	PROGRAMMING LANGUAGE.....	7
1.3	INTERNATIONALISATION.....	7
1.4	MISCELLANEOUS.....	7
2	DATABASE TABLES	8
2.1	RA-DB.....	8
2.2	CA-DB.....	16
2.3	KGS-DB.....	24
3	MODULE SPECIFICATIONS	29
3.1	MODULE NUMBER FO-1 / RA-1: USER INTERFACE.....	29
3.2	MODULE NUMBER FO-2: COMMUNICATION WITH BO.....	29
3.3	MODULE NUMBER RA-2: VALIDATION.....	30
3.4	MODULE NUMBER RA-3: LOG.....	31
3.5	MODULE NUMBER RA-4: RA DATABASE.....	31
3.6	MODULE NUMBER RA-5: ENTRY / RENEWAL.....	33
3.7	MODULE NUMBER RA-7: COMMUNICATION TO CA.....	33
3.8	MODULE NUMBER RA-8: AUDIT.....	34
3.9	MODULE NUMBER RA-10: RA-API.....	34
3.10	MODULE NUMBER RA-13: REVOCATION.....	34
3.11	MODULE NUMBER RA-14: ENTITY/CERT BROWSER.....	35
3.12	MODULE NUMBER RA-15: ACCESS CONTROL.....	35
3.13	MODULE NUMBER RA-16: ADMIN INTERFACE.....	36
3.14	MODULE NUMBER RA-17: ADMIN-API.....	36
3.15	MODULE NUMBER RA-18: RA PK ACCEPTOR.....	37
3.16	MODULE NUMBER CA-1: USER REQUEST HANDLER.....	37
3.17	MODULE NUMBER CA-2: WRITEMAIL.....	38
3.18	MODULE NUMBER CA-3: LOG.....	39
3.19	MODULE NUMBER CA-5: ADMIN INTERFACE.....	42
3.20	MODULE NUMBER CA-6: ADMIN-API.....	45
3.21	MODULE NUMBER CA-8: AUDIT.....	48
3.22	MODULE NUMBER CA-9: OTP DISTRIBUTOR.....	48
3.23	MODULE NUMBER CA-10: CA-API.....	50

3.24	MODULE NUMBER CA-11: OTP GENERATOR	59
3.25	MODULE NUMBER CA-12: OTP AUTHENTICATOR	61
3.26	MODULE NUMBER CA-13: COMMUNICATION TO CSP	63
3.27	MODULE NUMBER CA-14: CA DATABASE	63
3.28	MODULE NUMBER CA-15: ACCESS CONTROL	64
3.29	MODULE NUMBER CA-16: PUBLICATION	68
3.30	MODULE NUMBER CA-17: COMMUNICATION TO KGS.....	69
3.31	MODULE NUMBER CA-18: CRL FACTORY	69
3.32	MODULE NUMBER CA-20: CERTIFICATE UPDATE AGENT	71
3.33	MODULE NUMBER CSP-1: CSP-API	73
3.34	MODULE NUMBER CSP-2: CERTIFICATE SIGNER	73
3.35	MODULE NUMBER CSP-3: CRL SIGNER	74
3.36	MODULE NUMBER CSP-4: CRYPTO ENGINE	74
3.37	MODULE NUMBER KC-2: CERTIFICATE FACTORY.....	74
3.38	MODULE NUMBER KC-5: ADMIN INTERFACE	74
3.39	MODULE NUMBER KC-6: ADMIN API.....	77
3.40	MODULE NUMBER KC-15: ACCESS CONTROL	80
3.41	MODULE NUMBER KC-26: KEY PAIR FACTORY	80
3.42	MODULE NUMBER KC-30: SECRET EXPORT	81
3.43	MODULE NUMBER KC-31: SECRET PRINTING.....	81
3.44	MODULE NUMBER KC-32 SECRET IMPORT	81
3.45	MODULE NUMBER KGS-3: LOG.....	81
3.46	MODULE NUMBER KGS-5: ADMIN INTERFACE	82
3.47	MODULE NUMBER KGS-6: ADMIN-API	83
3.48	MODULE NUMBER KGS-8: AUDIT	84
3.49	MODULE NUMBER KGS-10: KGS API JOB SCHEDULER	86
3.50	MODULE NUMBER KGS-14: KEY STORE	89
3.51	MODULE NUMBER KGS-15: ACCESS CONTROL (CENTRAL KGS).....	89
3.52	MODULE NUMBER KGS-22: CERTIFICATE STORAGE.....	90
3.53	MODULE NUMBER KGS-23: EXPORT CERTIFICATE REQUEST TO CA	91
3.54	MODULE NUMBER KGS-24: IMPORT CERTIFICATE REPLIES FROM CA	92
3.55	MODULE NUMBER KGS-25: USER INTERFACE	93
3.56	MODULE NUMBER KGS-26: KEY PAIR FACTORY	94
3.57	MODULE NUMBER KGS-27: USER KGS API.....	95
3.58	MODULE NUMBER KGS-28: ACCESS CONTROL (USER KGS).....	96
3.59	MODULE NUMBER KGS-29: EXPORT PKCS#12	97
3.60	MODULE NUMBER KGS-30: KGS DATABASE.....	98
4	APPENDIX	102

4.1 TERMS AND DEFINITIONS, GLOSSARY 102

4.2 REFERENCES 102

1 Introduction

1.1 Function Table

This revised function table replaces the function tables from D3.6 and D4.1.

The functions F14, F20, F21, F22 and F25 from the function table from D3.6 were identified to be out of scope for the EuPKI project.

The function F24 is renamed and would not be implemented in the current release of EuPKI.

The functions F16, F17, F18 and F19 are combined in one function F16-F19 "Manage operator profiles and privileges".

The function F12 is renamed to "Create certificate profile/template" and a sub function F12.1 "Sending the profile to the RA" is added. It was defined that the profile/template is created in the CA and afterwards sent to the RA.

The function F13 is renamed to "Key Ceremony" and the sub functions F13.1, F13.2 and F13.3 are added.

The function F15 is renamed to "Create and manage RA account".

It was realised that the functions F20, F22, F23 and F25 are all covered by the function F10. For this the function F23 are changed to a sub function F10.1 and the others are defined to be out of scope (e.g. must be realised by third party programs).

FUNCTIONS	
Number	Name
F1	Generate keys and certificate
F1.1	Generate keys
F1.2	Generate certificate
F2	Recover private encryption key
F3	Revoke certificate
F4	Repudiate keys
F5	Publish certificate in a directory
F6	Publish certificate in a CRL
F7	Suspend certificate
F8	Reactivate suspended certificate
F9	Update information in a certificate
F9.1	Renew certificate (F9 must have been executed prior to F9.1)
F10	View event log
F10.1	Consult alarms

F11	Recover certificate
F12	Create a Certificate profile/template
F12.1	Sending the
F13	Key Ceremony
F13.1	CA/sub CA management
F13.2	Administrator management
F13.3	Secrets import/export
F15	Create and manage RA account
F16-19	Manage operator profiles and privileges

See also the extended function_module table at the end of document D8.2.

1.2 Programming Language

The specification of the modules in this document is not based on a particular programming language. The choice of the programming language should be made result oriented and the focus should be set on the easiness of implementation.

For the modules which interacts with each other, the interface is also of note for the implementation. This is especially the case where already available open source products are deployed.

The focus of the implementation should be defined on the quality of the software product on the end and the time of implementation.

1.3 Internationalisation

The language for the implementation of the EuPKI project will be English.

The implementation of the RA will be web based. For that an adaptation to another language could be made easily.

The CA interfaces are only for technical users and therefore an adaptation to another language is not mandatory.

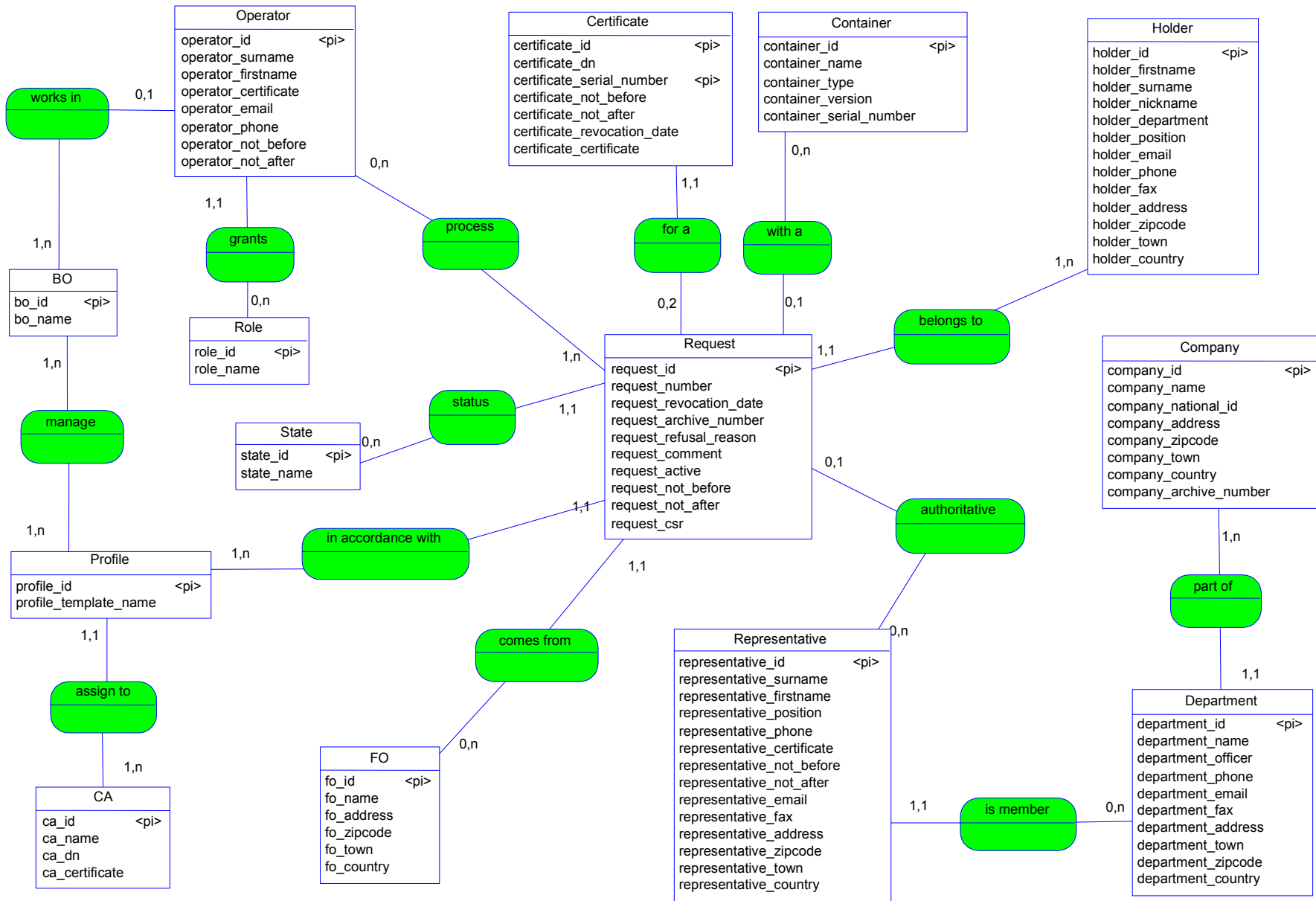
The User KGS module is the only module which would be delivered to the end users. For that it must provide an mechanism for internationalisation. This for example could be made with a resource based technology. All strings and messages which are displayed or visible for the user are stored in a resource database (e.g. resource file) and referenced in the application. They will be loaded while runtime. An internationalisation could be made by translating the strings and messages in the resource database.

1.4 Miscellaneous

Other requirements to the system like cleanup of the database and a monitoring of the hardware and processes are not integral part of the EuPKI and must be realised by using third party programs.

2 Database Tables

2.1 RA-DB



2.1.1 Table: BO

List of the Back Offices where operator's can validate requests.

Field	Usage
BO_ID	Identifier
BO_NAME	Name

2.1.2 Table: CA

List of the Certification Authority where the requests can be sent.

Field	Usage
CA_ID	Identifier
CA_NAME	Name
CA_DN	Distinguished Name
CA_CERTIFICATE	X509 Certificate

2.1.3 Table: COMPANY

List of companies that can ask for certificates.

Field	Usage
COMPANY_ID	Identifier
COMPANY_NAME	Name
COMPANY_NATIONAL_ID	National Identifier (like SIREN in France)
COMPANY_ADDRESS	Address
COMPANY_ZIPCODE	Zip Code
COMPANY_TOWN	Town
COMPANY_COUNTRY	Country
COMPANY_ARCHIVE_NUMBER	Reference in the Archive Folder

2.1.4 Table: CONTAINER

List of containers distributed with certificates.

Field	Usage
CONTAINER_ID	Identifier
CONTAINER_NAME	Name
CONTAINER_TYPE	Type (Smartcard, Token, HSM)
CONTAINER_VERSION	Version
CONTAINER_SERIAL_NUMBER	Serial Number

2.1.5 Table: DEPARTMENT

Department informations within a company.

Field	Usage
DEPARTMENT_ID	Identifier
COMPANY_ID	Foreign Key to Company
DEPARTMENT_NAME	Name
DEPARTMENT_OFFICER	Name of the Officer
DEPARTMENT_PHONE	Telephone number
DEPARTMENT_EMAIL	Email address
DEPARTMENT_FAX	Facsimile number
DEPARTMENT_ADDRESS	Address
DEPARTMENT_TOWN	Town
DEPARTMENT_ZIPCODE	Zip code
DEPARTMENT_COUNTRY	Country

2.1.6 Table: FO

List of front offices where operators can input entity informations.

Field	Usage
FO_ID	Identifier
FO_NAME	Name
FO_ADDRESS	Address
FO_ZIPCODE	Zip code

FO_TOWN	Town
FO_COUNTRY	Country

2.1.7 Table: HOLDER

List of certificates holder.

Field	Usage
HOLDER_ID	Identifier
HOLDER_FIRSTNAME	First name
HOLDER_SURNAME	Surname
HOLDER_NICKNAME	Nickname
HOLDER_DEPARTMENT	Department name
HOLDER_POSITION	Position
HOLDER_EMAIL	Email address
HOLDER_PHONE	Telephone number
HOLDER_FAX	Facsimile number
HOLDER_ADDRESS	Address
HOLDER_ZIPCODE	Zip code
HOLDER_TOWN	Town
HOLDER_COUNTRY	Country

2.1.8 Table: PROFILE

List of available certificate profiles in Certification Authority.

Field	Usage
PROFILE_ID	Identifier
CA_ID	Foreign Key to Certification Authority
PROFILE_TEMPLATE_NAME	Certificate Template Name in CA

2.1.9 Table: REPRESENTATIVE

List of representative within a company.

Field	Usage
REPRESENTATIVE_ID	Identifier
DEPARTMENT_ID	Foreign Key to Representative's department
REPRESENTATIVE_SURNAME	Surname
REPRESENTATIVE_FIRSTNAME	First name
REPRESENTATIVE_POSITION	Position
REPRESENTATIVE_PHONE	Telephone number
REPRESENTATIVE_CERTIFICATE	X509 Certificate
REPRESENTATIVE_NOT_BEFORE	Begin of validity of the certificate
REPRESENTATIVE_NOT_AFTER	End of validity of the certificate
REPRESENTATIVE_EMAIL	Email address
REPRESENTATIVE_FAX	Facsimile number
REPRESENTATIVE_ADDRESS	Address
REPRESENTATIVE_ZIPCODE	Zip code
REPRESENTATIVE_TOWN	Town
REPRESENTATIVE_COUNTRY	Country

2.1.10 Table: ROLE

List of operator's role, in order to restrict services access.

Field	Usage
ROLE_ID	Identifier
ROLE_NAME	Name

2.1.11 Table: STATE

List of possible states of a request.

Field	Usage
STATE_ID	Identifier
STATE_NAME	Name

2.1.12 Table: REQUEST

List of certification requests

Field	Usage
REQUEST_ID	Identifier
REPRESENTATIVE_ID	Representative that allow the request
FO_ID	FO where the request comes from
CONTAINER_ID	Container needed for the certificate
STATE_ID	Current state of the request
PROFILE_ID	Type of certificate requested
HOLDER_ID	The one who will hold the certificate
REQUEST_NUMBER	Request number
REQUEST_REVOCATION_DATE	Date of revocation
REQUEST_ARCHIVE_NUMBER	Reference of the archive number
REQUEST_REFUSAL_REASON	Reason of refusal
REQUEST_COMMENT	General comment
REQUEST_ACTIVE	Is the certificate active?
REQUEST_NOT_BEFORE	Begin of validity of the certificate
REQUEST_NOT_AFTER	End of validity of the certificate
REQUEST_CSR	The request itself (PKCS10, ...)

2.1.13 Table: OPERATOR

List of the granted operators.

Field	Usage
OPERATOR_ID	Identifier
ROLE_ID	Operator's role
BO_ID	BO where the operator is.
OPERATOR_SURNAME	Surname
OPERATOR_FIRSTNAME	First name
OPERATOR_CERTIFICATE	X509 Certificate
OPERATOR_EMAIL	Email address

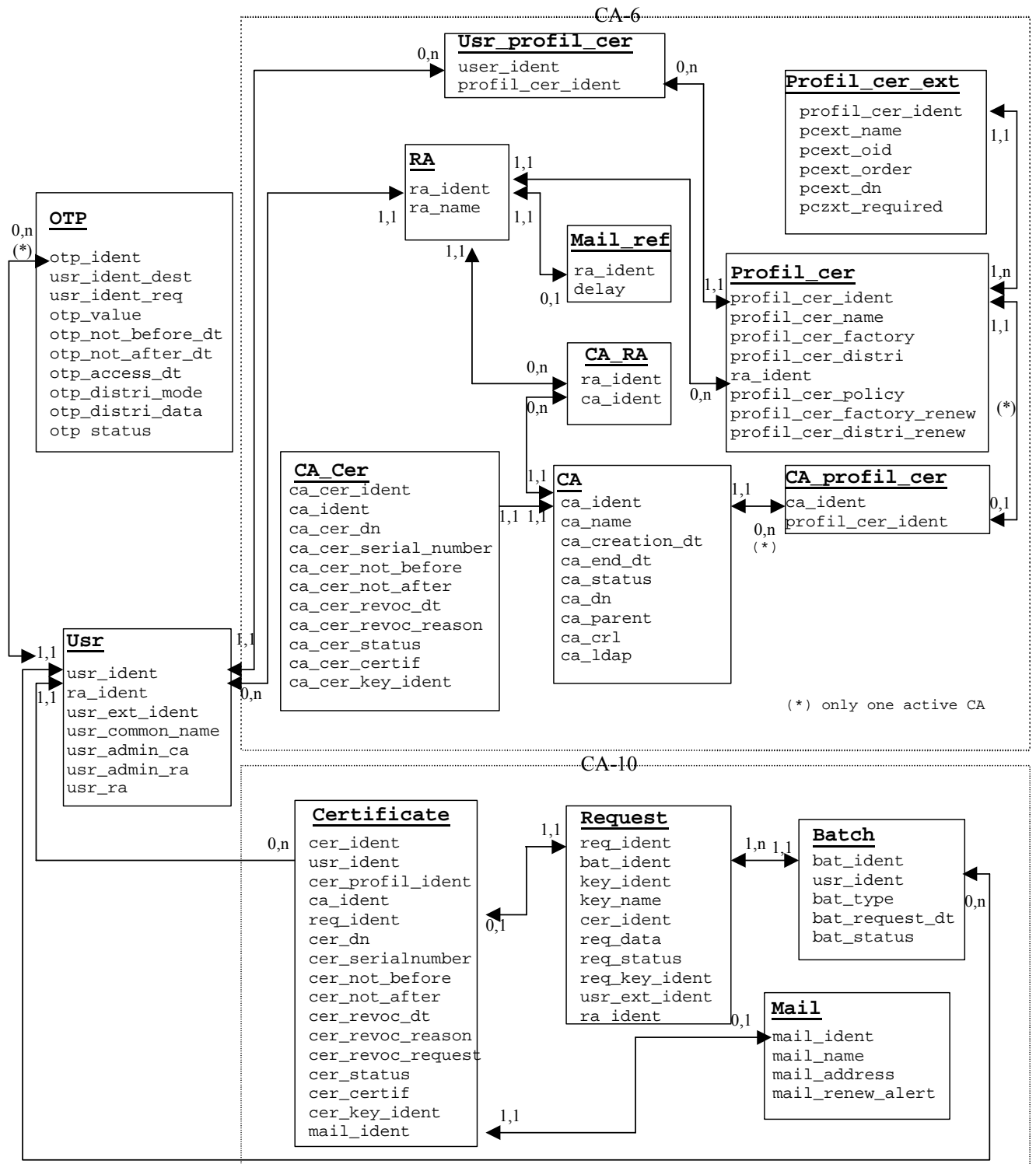
OPERATOR_PHONE	Telephone number
OPERATOR_NOT_BEFORE	Begin of validity of the certificate
OPERATOR_NOT_AFTER	End of validity of the certificate

2.1.14 Table: CERTIFICATE

List of the produced certificates.

Field	Usage
CERTIFICATE_ID	Identifier
CERTIFICATE_DN	Distinguished name
CERTIFICATE_SERIAL_NUMBER	Serial Number
REQUEST_ID	Request that has produced the Certificate
CERTIFICATE_NOT_BEFORE	Begin of validity of the certificate
CERTIFICATE_NOT_AFTER	End of validity of the certificate
CERTIFICATE_REVOCATION_DATE	Date of revocation
CERTIFICATE_CERTIFICATE	X509 Certificate

2.2 CA-DB



2.2.1 Table RA : RAs List.

Field	Usage	Rules
ra_ident	RA ident	NOT NULL

ra_name	RA name	NOT NULL
---------	---------	----------

A RA can be a domain or a company.

2.2.2 Table Mail_ref : Delay for renew alert.

Mail reference is used to obtain information about the delay to alert usr to renew his certificate.

Field	Usage	Rules
ra_ident	RA ident	NOT NULL
delay	in days NULL if no mail alert for renew	

2.2.3 Table Usr : Users list and their rights.

Field	Usage	Rules
usr_ident	User ident	NOT NULL
ra_ident	RA ident (a company for example)	
usr_ext_ident	User ident in his RA (a company for example)	
usr_common_name	User common name	
usr_admin_ca	Administrator CA : - Access to the CA Admin IHM (management user, creation CA, audit, creation OTP)	NOT NULL Y Yes N No (default)
usr_admin_ra	- Validity OTP (by RA17) - Access to some function of the CA Admin IHM: <ul style="list-style-type: none"> • Link profile's certificate (linked with his RA domain) to user ra • Determinate the field Mail_ref.delay for his RA's domain 	NOT NULL Y Yes N No (default)
usr_ra	- validity OTP (by RA17)	NOT NULL Y Yes N No (default)

ra_ident is used to know which RAs are linked with an user.

usr_extident is used to identify an user within his RA (unique within a domain or company).

If user_admin_ca equals "Y", he is not restricted to his RA's domain to administrate the CA and can access to all profile's certificate. If he is a user RA too, his ra_ident and user_ext_ident must be completed and he can just use his profile's certificate to do certificate request.

2.2.4 Table CA : CAs list.

Field	Usage	Rules
ca_ident	CA ident	NOT NULL
ca_name	CA name	NOT NULL
ca_creation_dt	Creation date	NOT NULL
ca_end_dt	End date	
ca_status	Status	V Valid S Suspend R Revoked E Expired
ca_dn	Distinguish name	NOT NULL
ca_parent	His parent CA	NULL if it is a CA root
ca_crl	Address of his CRL	NOT NULL
ca_ldap	Address of his LDAP	

ca_ident identify the CA and can be used to inform the CSP the private key to use for signature.

The status V (valid) allows to use the CA to create certificate.

The status S (suspend) suspends the CA and all his sub Cas which become out of order ; moreover, certificates they have signed are still valid.

The status R (Revoked).

The status E (Expired) revokes all the sub CAs and all the certificates they have signed.

2.2.5 Table Ca_cer : CA certificates list

Field	Usage	Rules
ca_cer_ident	Certificate ident	NOT NULL
ca_ident	CA ident	

ca_cer_dn	Certificate dn	NOT NULL
ca_cer_serial_number	unique for one CA	NOT NULL
ca_cer_not_before	Not valid before	NOT NULL
ca_cer_not_after	Not valid after	NOT NULL
ca_cer_revoc_dt	Revocation date	
ca_cer_revoc_reason	Revocation reason	
ca_cer_status	Certificate status	N New (Not yet valid – or not yet published) V Valid R Revoked S Suspended E Expired
ca_cer_certif	Store the certificate	NOT NULL
ca_cer_key_ident	Ident key to identify the generated key by the central KGS	NOT NULL

2.2.6 Table CA_RA : Links between RA and CA.

Field	Usage	Rules
ca_ident	CA Ident	NOT NULL
ra_ident	RA Ident	NOT NULL

2.2.7 Table Profil_cer : Certificate profiles list.

Field	Usage	Rules
profil_cer_ident	Profile certificate ident	NOT NULL
profil_cer_name	Profile certificate name	
profil_cer_factory	Factory certificate method	PKCS12
profil_cer_distrib	Distribution certificate method	Email Smartcard
ra_ident	RA Ident	
profil_cer_policy	Link with the default profile of CA	NOT NULL: policy_admin policy_admin_ra

		policy_ra policy_any
profil_cer_factory_renew	Factory certificate method to renew	PKCS12
profil_cer_distrib_renew	Distribution certificate method to renew	Email

2.2.8 Table Profile_cer_ext : Certificate profile extensions list.

Field	Usage	Rules
profil_cer_ident	Profile certificate ident	NOT NULL
pnext_name	Extension name	C, O, ORGDN, ST, L, CN, SSL, EMAIL, KeyUsage, KeySize, KeyType...
pnext_oid	Extension OID	
pnext_order	Indicates the extension order	
pnext_dn	Indicates if the extension is used in the dn	Y Yes N No
pnext_required	The value of the extension is compulsory	Y Yes N No

2.2.9 Table Usr_profil_cert : Certificate profiles an user can use

Field	Usage	Rules
user_ident	User ident	NOT NULL
profil_cer_ident	Profile certificate ident	NOT NULL

The user must be a RA to have links in this table.

2.2.10 Table Ca_profil_cer : determines which CA to use for certificate signature.

Field	Usage	Rules
ca_ident		NOT NULL
profil_cer_ident		NOT NULL

This table is used to know the CA which will sign the certificate depending on the certificate profile.

2.2.11 Table Certificate : Certificates list.

Field	Usage	Rules
cer_ident	Certificate Ident	NOT NULL
usr_ident	User Ident	NOT NULL
cer_profil_ident	Certificate Profile Ident	NOT NULL
ca_ident	Signing CA Ident	NOT NULL
req_ident	Request Ident	NOT NULL
cer_dn	Certificate dn	NOT NULL
cer_serialnumber	Serial Number (unique for a CA)	NOT NULL
cer_notbefore	Valid not before	NOT NULL
cer_notafter	Valid not after	NOT NULL
cer_revocationdt	revocation date	
cer_revocationreason	revocation reason	
cer_revrequest	Request Ident	
cer_status	Certificate Status	N New (Not yet valid – or not yet published) V Valid R Revoked S Suspended E Expired
cer_certif	Store the certificate and can be used for the renew process	NOT NULL
cer_key_ident	Ident key to identify the generated key by the central KGS – NULL if the key is transmitted by the RA.	
mail_ident	Mail ident to the renew alert	

2.2.12 Table 11 Batch : RA demands List.

This table contains information about RA job requests which contains one or many certificate requests.

Field	Usage	Rules
--------------	--------------	--------------

bat_ident	Batch ident	NOT NULL
usr_ident	Ident of the seeker	NOT NULL
bat_type	Type of demand	C Creation R Revocation N Renew S Suspend A reActivate
bat_request_dt	Date of the demand	NOT NULL
bat_status	Status of the demand	P Processed D Done E Error

A bat status is "Done" when all its request are done (all request in the batch are processed and all the certificate or crl are published correctly).

It is "Error" when one (at least) request is in ERROR.

It is "Process" while all the request status are not "Done" or "Error".

2.2.13 Table Request : Certificate requests list.

Field	Usage	Rules
req_ident	Request ident	NOT NULL
bat_ident	Batch ident	NOT NULL
key_ident	To link the request with the RA request	NOT NULL
key_name	To link the request with the RA request	NOT NULL
cer_ident	Ident of certificate	
req_data	Request data	NOT NULL
req_status	Status of the demand	For reation, renew request: V Received, Validated P Processed R Ready_to_send S Sent D Published, Done E Error

		For revocation, Suspend, Reactivate request: V Received, Validated P Processed D Published, Done E Error
--	--	---

If the request is a creation request, the fields cer_ident is completed during the creation process.

If the request is a revocation request, a renew request, a suspend request or a reactivate request, the cer_ident is known or can be known with the certificate serial number and its ca.

Req_status is updated during the process.

Req_data contains all the information of the request in a given format (XML can be a solution).

2.2.14 Table Mail : Mail addresses used for certificate renewal.

Field	Usage	Rules
mail_ident	Mail ident	NOT NULL
mail_name	Address name	
mail_address	Mail address used by CA-2 Writemail	NOT NULL
mail_renew_alert	NULL if mail for renew is not sent	NULL S Sent

2.2.15 Table OTP.

Field	Usage	Rules
usr_ident	User ident	NOT NULL
otp_ident	OTP ident	NOT NULL
usr_ident_dest	End user ident	NOT NULL
usr_ident_req	Requester ident	NOT NULL
otp_value	OTP value	NOT NULL
otp_not_before_dt	Creation date	NOT NULL Creation date by

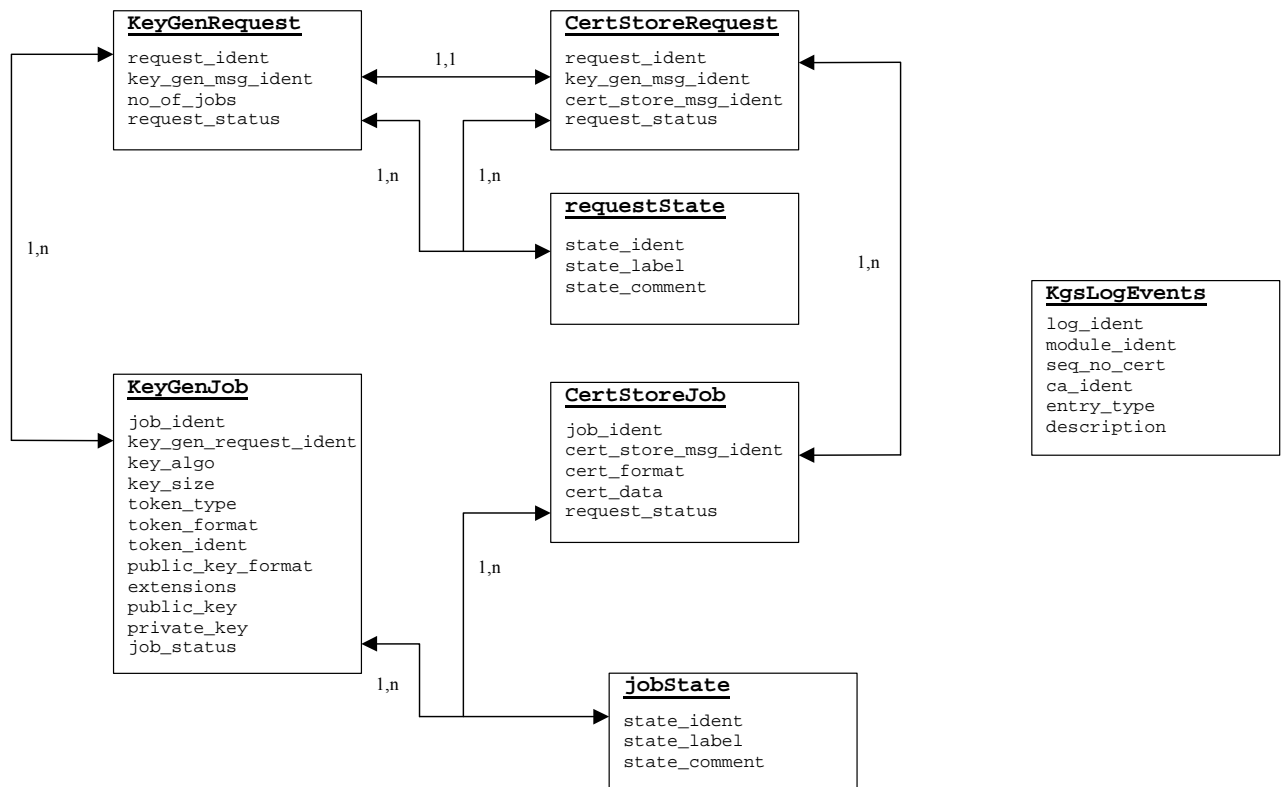
		default
otp_not_after_dt	NULL if no validity delay	
otp_access_dt	Date access (the OTP is valid if this field is NULL and otp_not_after is verify)	
otp_distri_mode	Distribution mode	NOT NULL M MAIL L LETTER
otp_distri_data	Data to distribute the OTP	NOT NULL
otp_status	OTP status	N New S Sent C Cancelled

If the end user does not exist, he is created.

If the OTP requester is a admin RA, this user is automatically in his admin RA domain.

If the OTP requester is a admin CA, this user can be in any RA domain. If this RA domain does not exist, it is created.

2.3 KGS-DB



2.3.1 *Table KeyGenRequest: Key generation requests.*

Field	Usage	Rules
request_ident	Key generation request ident	NOT NULL
key_gen_msg_ident	Key generation request message ident	NOT NULL
no_of_jobs	Number of key generation jobs	NOT NULL > 0
request_status	Status of request (reference to requestState table)	NOT NULL

A new entry is created when module KGS-10 KGS API Job Scheduler receives a Key Generation Request from the module CA17. The KGS-10 creates also the specified number of job entries in the table KeyGenJob.

After all related jobs are finished the status of the request is changed to 'P' (Processed). After the "Key Generation Completed" message was send to the CA the status is changed to 'D' (Done).

The status of 'E' (Error) indicates, that a error occurred during the key generation resp. certificate storage in one of the related jobs. The error will be logged in the KGS Log database.

2.3.2 *Table CertStoreRequest: Certificate storage requests*

Field	Usage	Rules
request_ident	Certificate storage request ident	NOT NULL
key_gen_msg_ident	Key generation request message ident (relation to table KeyGenRequest)	NOT NULL
cert_store_msg_ident	Certificate storage request message ident	NOT NULL
request_status	Status of request (reference to requestState table)	NOT NULL

A new entry is created when module KGS-10 KGS API Job Scheduler receives a Certificate Storage Request from the module CA17.

After all related jobs are finished the status of the request is changed to 'P' (Processed). After the "Certificate Storage Completed" message was send to the CA the status is changed to 'D' (Done).

The status of 'E' (Error) indicates that a error occurred during the key generation resp. certificate storage in one of the related jobs. The error will be logged in the KGS Log database.

2.3.3 Table KeyGenJob: Key generation jobs

Field	Usage	Rules
job_ident	Key generation job ident	NOT NULL
key_gen_request_ident	Key generation request ident (relation to table KeyGenRequest)	NOT NULL
key_algo	Type of algorithm to use (RSA)	NOT NULL
key_size	Size of the key (compatible with the algorithm)	NOT NULL
token_type	Type of token (file, smartcard)	
token_format	Format to store (PKCS12, PKCS7)	
token_ident	Identifier of the token where the key was (generated and) stored	
public_key_format	Public key format (format of the data key value to return or PKCS10)	
extensions <i>(For Future use)</i>	List of extension + value (optional: if the public key format is PKCS10)	
public_key	Public key data	
private_key	Private key data	
job_status	Status of job (reference to requestState table)	NOT NULL

The entries are generated when a "Key Generation Request" message is received.

The field token_ident is a reference to the token object where the key pair is stored. This could be a serial number of a smart card or a filename.

After the key generation is completed, the status of the job entry is changed to 'P' (Processed). After the "Key Generation Completed" message is send to the CA, the status is changed to 'D' (Done).

The status of 'E' (Error) indicates that a error occurred during the key generation resp. certificate storage. The error will be logged in the KGS Log database.

2.3.4 Table CertStoreJob: Certificate storage jobs

Field	Usage	Rules
job_ident	Certificate storage job ident	NOT NULL

cert_store_msg_ident	Certificate storage request message ident	NOT NULL
cert_format	Certificate format (X509 v3, RFC2459)	NOT NULL
cert_data	Certificate data	NOT NULL
job_status	Status of job (reference to jobState table)	NOT NULL

The entries are generated when a "Certificate Storage Request" message is received.

After the certificate storage is completed, the status of the job entry is changed to 'S' (Certificate Stored). After the "Certificate Storage Completed" message is send to the CA, the status is changed to 'D' (Done).

The status of 'E' (Error) indicates that a error occurred during the key generation resp. certificate storage. The error will be logged in the KGS Log database.

2.3.5 Table KgsLogEvents: Log Events of the Central KGS

Field	Usage	Rules
log_ident	Certificate storage job ident	NOT NULL
module_ident	Identifier of module where the log entry was produced	NOT NULL
seq_no_cert	Sequence number of certificate related to the log entry	
entity_ident	Entity identifier related to the log entry	
ca_ident	Certification authority identifier related to the log entry	
entry_type	Log entry type (Event, Warning, Error, etc.)	NOT NULL
description	Short text which describes the event, warning, etc.	NOT NULL

This table of the KGS database is for logging all events, warnings, errors, etc. occurring during the operation of the central KGS. The KGS-8 Audit module receives the information from this database table.

2.3.6 Table requestState: Request States

Field	Usage	Rules
--------------	--------------	--------------

state_ident	State ident	NOT NULL
state_label	State label	NOT NULL
state_comment	Short description of the state	

Possible Values are:

Label	Description
V	Request received and validated
P	Request processed
D	Done, status of request reported to sender CA
E	Error

2.3.7 Table jobState: Job States

Field	Usage	Rules
state_ident	State ident	NOT NULL
state_label	State label	NOT NULL
state_comment	Short description of the state	

Possible Values are:

Label	Description
V	Job extracted from request and validated
P	Key generation processed
S	Certificate stored
D	Done, status of job reported to sender CA
E	Error

3 Module specifications

3.1 Module Number FO-1 / RA-1: User Interface

3.1.1 Functionality

The RA-1 module is the only entry point for interaction between the operator and the RA. The objective of this module is to generate the GUI and display it at the operator. Thanks to it the operator can choose the operations which it wishes to carry out.

The module FO-1 is used in the Front Office, the module RA-1 in the Back Office.

3.1.2 API

Various functions have to be developed according the kind of graphical user interface (web based).

User interface can be made up for example with JSP and Java Beans.

WP5 has to define the necessary functions.

3.2 Module Number FO-2: Communication with BO

3.2.1 Functionality

The Communication module including the Spool is meant to assure a functional RA in the case of a temporary network blackout. It may be scaled to provide functionality when network access is occasional.

An RA Front Office is independent of the RA Back Office or the network up to the point where approved certification requests are ready to be transmitted to the main repository. When network access is not available, these ready-to-be-sent packages are passed to the spool, which in turn handles the safe recording of these packages on a hard media. Once network access is back up, the contents of the spool are fed in one single operation to the master repository.

The communication module provides the network status at the time of package transmission (connected or disconnected). When network communications are re-established, a notification event is issued, and the entire spool gets sent immediately. If additional packages are to be transmitted at the same time, no special action is taken; the communication module opens a new channel and transmits the package in the usual (connected) fashion, without interfering with the spool synchronisation.

The spool is always accompanied by a check sum value, and its successful transmission and removal must always be explicitly confirmed by the receiving end.

The receiving module accepts the entire spool entity, verifies the integrity of the package, synchronizes the package on a hard media and responds with a success or failure message. It then further processes each individual message according to the local policies.

3.2.2 API

3.2.2.1 Connect

- Input:
- Output:
- Process:

3.2.2.2 Send

- Input:
- Output:
- Process:

3.2.2.3 Disconnect

- Input:
- Output:
- Process:

3.3 Module Number RA-2: Validation

3.3.1 Functionality

The RA-2 module is responsible for handling the action of "validation" which is defined by approving certification requests received through the RA-10 module. It transfers pending requests from a "waiting" into an "approved" state.

The use of that module is optional. The RA-API is allowed to create requests in an "approved" state. If it is used, the module presents a list of registration events that shall cause the creation of certificates waiting for approval, and accepts or denies requests based on an operator decision.

3.3.2 API

3.3.2.1 Allow

- Input: Entity's information to build the Register Document.
- Output: XML Register Document signed by the Validator.
- Process: The state of the request is changed to Validated, so the communication (RA-7) process will send the register document to the CA.

3.3.2.2 Deny

- Input: Entity's information an email.
- Output: An email is sent to notice the entity of the refusal.
- Process: The state of the request is changed to Refused.

3.4 Module Number RA-3: Log

3.4.1 *Functionality*

This module is used as interface between the RA and the logs.

All the requests of generation of logs are sent to this module. Logs are considered in the broad meaning. This module manages the logs recorded in a data base and the emission of events network like a SNMP traps.

The logs recorded in a data base are divided into three types:

- Access logs trace all the authentications or attempt at authentication. The profile of each user is also recorded.
- Audit logs trace all the significant events of the application.
- Errors logs record all the errors which occur within the application. The errors can be internal or external. Theoretically, these errors should only be external of the type "such external module does not answer any more".

3.4.2 *API*

3.4.2.1 Error

- Input: Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment.
- Output: An error entry is the log database.

3.4.2.2 Audit

- Input: Time, Module's name, Operator's Id, Entity Id, Action performed.
- Output: An audit entry is the log database.

3.4.2.3 Access

- Input: Time, Module's name, Operator's Id, Authentication result, Option: The reason of the failure.
- Output: An access entry is the log database.

3.5 Module Number RA-4: RA Database

3.5.1 *Functionality*

This module provides access to the RA DB data model and supports creation, modification and query of information stored in the RA DB.

For full specifications, see JDBC 3.0 API.

The caller must maintain a connection to the database. The connection is not Thread Safe, so the caller must take care of it.

3.5.2 API

3.5.2.1 Select

- Input: Connection, Table name, field's name, conditions
- Output: ResultSet

3.5.2.2 Insert

- Input: Connection, Table name, field's name, values
- Output: Status

3.5.2.3 Update

- Input: Connection, Table name, field's name, values, conditions
- Output: Status

3.5.2.4 Delete

- Input: Connection, Table name, conditions
- Output: Status

3.5.2.5 Exec

- Input: Connection, SQL statement
- Output: Status

3.5.2.6 Call

- Input: Connection, Procedure name, parameter's value
- Output: ResultSet

3.5.2.7 Connect

- Input: Connect String, User's name, User's login
- Output: Connection

3.5.2.8 Disconnect

- Input: Connection
- Output: Status

3.5.2.9 Commit

- Input: Connection
- Output: Status

3.5.2.10 Rollback

- Input: Connection
- Output: Status

3.6 Module Number RA-5: Entry / Renewal

3.6.1 Functionality

This module allows the creation of an input (End user, Server...) seized by an operator. Both new entries and updates of existing entries are handled by this module. Updates for existing entries causes new associated data entries for an existing entity to be created.

This module uses the module RA-4 to find and update information which it needs.

This module issues a certificate production job at the CA through RA-7.

3.6.2 API

3.6.2.1 Insert

- Input: Entity's information from operator inputs.
- Output: Entry in database.
- Process: A new Entity is inserted in the RA-DB.

3.6.2.2 Renew

- Input: Entity's information from a previous entity.
- Output: Entry in database.
- Process: A new Entity is inserted in the RA-DB with the reference to the previous one.

3.7 Module Number RA-7: Communication to CA

3.7.1 Functionality

This module is used as interface between the RA and the CA. All the requests addressed to the CA pass by this module which is charged to format the requests with the good format before transmitting them to the CA. This module is also charged to recover the answers of the CA

3.7.2 API

3.7.2.1 Send

- Input: Signed Document from database.
- Output: Acknowledgement from CA.
- Process: The XML Document is send to the CA with a secure connection. The state of the request is changed in accordance with the result of the communication.

3.7.2.2 Retrieve

- Input: State of a request in the CA.
- Output: Data retrieve according to the state.
- Process: The state of the pending requests are retrieved, and if the state has changed, the data bounded are retrieved and store in the RA-DB (example, "certificate produced" is bounded to a X509 Certificate)

3.8 Module Number RA-8: Audit

3.8.1 Functionality

This module allows the consultation of the logs and the induced statistics.

3.8.2 API

3.8.2.1 View

- Input: Log entry id.
- Output: Log entry content.

3.8.2.2 Find

- Input: criteria (date, email, action type ...)
- Output: Log entries matching criteria.

3.9 Module Number RA-10: RA-API

3.9.1 Functionality

This module is the heart of the RA. It establishes the link between the actions seized by the operators or other servers (for bulk input) and the functionalities of the RA.

For the whole of the Operator services, the RA-API module calls the suitable functional modules. All information necessary to the functional module are sought in this module and are transmitted in parameter of the call.

3.9.2 API

The API is described by the XML schema that invokes the different services. Document D4.2 some example of the different invocations.

WP5 should complete it depending on the technical choices.

3.10 Module Number RA-13: Revocation

3.10.1 Functionality

This module generates and emits a request for revocation or suspension or reactivation from suspension of a certificate to a given CA.

3.10.2 API

3.10.2.1 Revoke

- Input: Entity's information to build the Register Document.
- Output: XML Register Document signed by the Validator.
- Process: The state of the request is changed to Invalidated, so the communication (RA-7) process will send the register document to the CA.

3.10.2.2 Suspend

- Input: Entity's information to build the Register Document.
- Output: XML Register Document signed by the Validator.
- Process: The state of the request is changed to Suspended, so the communication (RA-7) process will send the register document to the CA.

3.11 Module Number RA-14: Entity/Cert Browser

3.11.1 Functionality

Thanks to this module an operator can know the list of all the certificates of an end-user and their state. This functionality is particularly useful for the revocation or the suspension of a certificate.

This module uses the module RA-4 and RA-7 (for CA DB access) to find information which it needs.

3.11.2 API

3.11.2.1 Find

- Input: criteria (state, dn, email ...)
- Output: Entity's information (contains the certificate if available).

3.12 Module Number RA-15: Access Control

3.12.1 Functionality

This module ensures the service of authentication of the operators and administrators. The adopted solution is the authentication of the operators starting from the presentation to the RA or the Admin Interface of their certificate.

This module ensures the service of Access Control of the operators and administrators. The adopted solution is based on the consultation of DB Access to know the privileges of the authenticated operators and administrators.

3.12.2 API

3.12.2.1 IsGranted

- Input: (Certificate or Serial Number with Issuer DN, or Finger Print), Module to access.
- Output: access control list. (Read, Write, Delete ... (empty list means none))
- Process: Look up in the database the different access granted to the user.

3.13 Module Number RA-16: Admin Interface

3.13.1 Functionality

The RA-16 module is the only entry point for interaction between the administrator and the RA. The objective of this module is to generate the IHM and display it at the administrator. Thanks to it the administrator can choose the operations which he wishes to carry out.

3.13.2 API

Various functions have to be developed according the kind of graphical user interface (web based).

Admin interface can be made up for example with JSP and Java Beans.

WP5 has to define the necessary functions.

3.14 Module Number RA-17: Admin-API

3.14.1 Functionality

This module is like the RA-API module except that it is dedicated to the administrators. It establishes the link between the actions seized by the administrators and the administration's functionalities of the RA.

For the whole of the Admin services, the Admin-API module calls the suitable functional modules. All information necessary to the functional module are sought in this module and are transmitted in parameter of the call.

3.14.2 API

The API is described by the XML schema that invokes the different services. Document D4.2 some example of the different invocations.

WP5 should complete it depending on the technical choices.

3.15 Module Number RA-18: RA PK Acceptor

3.15.1 Functionality

The RA-18 module allows RA operators to assign a public key to a registered entity. This module allows the handling of certification request received from user based key generation on behalf of the user. The operating RA officer verifies the origin and integrity of the request and the forward the certification request to the PKI for completion of a certificate production job. The result (the certificate) is as well received by the RA-18 module and then returned to the user. Requests and responses are transported from and to the entity by offline media.

3.15.2 API

3.15.2.1 Register

- Input: Public key within an XML Document or PKCS10.
- Output: Same document as input if it contains a proof of possession, or a signed Document with the PKCS10.
- Process: If the document is already in XML syntax with a proof of possession, it is send directly to the CA, otherwise this is a PKCS10 and there is an automatic signature of the XML register document then it is.

3.16 Module Number CA-1: User Request Handler

This module doesn't exist ? A end user does not directly access to the CA. All certificate request are messages transmit to the CA.

If he want to generate a certificate for him, he can use a OTP to connect to the RA interface and request a certificate.

Then the request is received by the CA with the CA-10 CA API module and the right of the user are checked by the CA-15 Access control module

His rights on profile certificate he can use are determinate by his RA admin :

- his RA admin create a user in the table usr (this end user exist)
- links this user to a RA (this end user is in a society or domain)
- allow to use some profile certificates to use (this user is a RA and can generate some certificate).

His OTP is given by the admin RA (or the admin CA) with the CA-5 admin CA interface.

When the end user wants to revoke his certificate, he can connect to the RA interface with its certificate and revoke it. Also, his rights in the user table are `usr_admin_ca = usr=admin_ra = usr_ra = N (NO)` and allow just to revoke his certificate.

3.17 Module Number CA-2: Writemail

Writemail is a technical module who just serves to send a mail to someone.

This module can be changed if the mail system is changed.

The content of the of the mail, the sender and the address are given by the caller (in this case, only by CA-20 module).

CA2_WriteMail	
Input parameter :	
SENDER	Sender mail address
ADDRESS	Mail address
CONTENT	Mail content
Output parameter :	
STATUS	Status

The field SENDER :

- The Writemail address (a address where a admin CA can read a response)

The field ADDRESS :

- A mail address (that can be the end user address, the RA address...)

The field CONTENT :

- A text

The field STATUS returns information about the service :

- A status (Error, Done)
- A error Id
- A error comment.

Each call to this module is logged with its results :

Module trace of CA-2 module Writemail :

- (Audit) – Audit ident, Time, Module name, Module call, Log content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'WRITEMAIL'

- Module call
- Log_content contains (depends of module call) :
If the calling module is CA-20 :
 - Certificate id (the field Certificate.cer_ident to renew)
 - Sender
 - Addressee
 - Mail content
 - **RA Id (consistency with the D4.1)**
 - Mail's id
 - **Comment (consistency with the D4.1)**
- Error content contains
 - **RA id (consistency with the D4.1)**
 - **Request (consistency with the D4.1)**
 - **Error origin (consistency with the D4.1)**
- Error id is given by the STATUS
- Error comment is given by status

Error	WRITEMAIL_1	Unknown address
Error	WRITEMAIL_100	"Error system"
Done	-	-

3.18 Module Number CA-3: Log

CA-3 module log is a technical module which allows any CA module to access to the log database.

This module can be changed if the database is changed (for example mySQL to postgresQL).

In the case database does not support some functions like sequence, rollback, it can support them.

If this data base is the same as the CA DB (not the same instance), it use the same interface:

CA_AccesBase	
Input parameter :	
DATABASE	CA DB or AUDIT

ACTION	LOG_SELECT LOG_WRITE ERROR_SELECT ERROR_WRITE
DATA	Action data
Output parameter :	
STATUS	Status

- Table audit :

Field	Usage	Rules
audit_ident		NOT NULL
audit_time		NOT NULL
audit_module		NOT NULL
audit_call		
audit_content	Module depending – format xml for example	

- Table errors :

Field	Usage	Rules
audit_ident		NOT NULL
error_time		NOT NULL
error_module		NOT NULL
error_call		NOT NULL
error_content	Module depending – format xml for example	
error_id		NOT NULL
error_comment		

The different module can be :

- WRITEMAIL
- ADMINAPI
- AUDIT
- CAAPI
- OTPGENERATOR

- OTPDISTRIBUTOR
- OTPAUTHENTICATOR
- ACCESSCONTROL
- PUBLICATION
- CRLFACTORY
- CERTIFICATEUPDATEAGENT

Error	LOG_1	Cannot access to the data base (the data base is stopped...)
Error	LOG_2	Access denied (password error...)
Error	LOG_100	"Data base error"
Done	-	-

Remark: This errors are not logged.

- Some errors cannot be logged (LOG_1 and LOG_2).
- Other errors can be error or not. It is only the calling which can determine it.

Here is a example to complete the audit tables. See for each CA module for other example.

Writemail module:

```

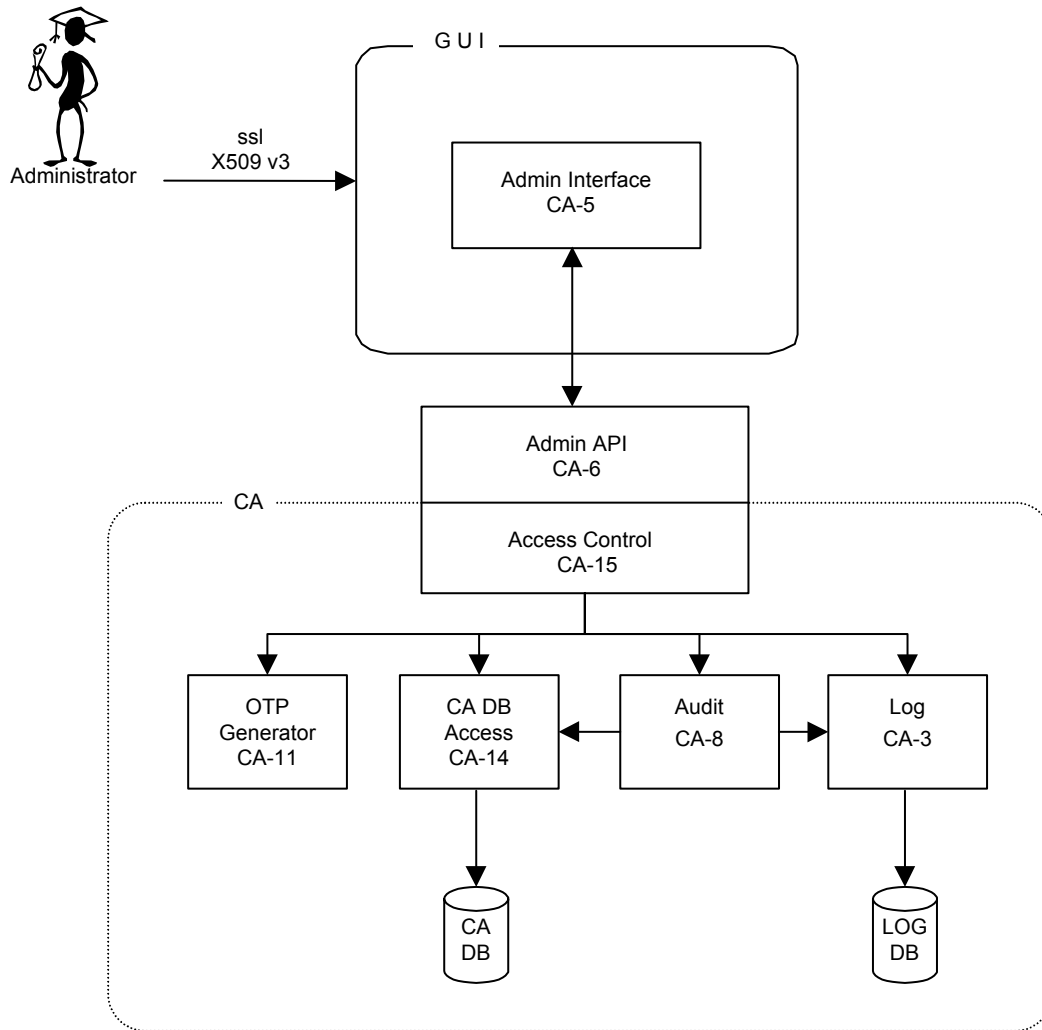
Audit ident      A data base sequence
Time            SYSDATE
Module name     WRITEMAIL
Module call     CERTIFICATEUPDATEAGENT
Log content     <LOG_COMMENT>
                <CER_IDENT>...</CER_IDENT>
                <SENDER>...</SENDER>
                <ADDRESS>...</ADDRESS>
                <MAIL_IDENT>...</MAIL_IDENT>
                <MAIL_CONTENT>...</MAIL_CONTENT>
                <RA_IDENT>...</RA_IDENT>
                <COMMENT>...<COMMENT>

```

<LOG_COMMENT>

Audit ident	A links with the audit table
Time	SYSDATE
Module name	WRITEMAIL
Module call	CERTIFICATEUPDATEAGENT
Error content	<ERROR_CONTENT> <RA IDENT>...</RA IDENT> <REQUEST>...</REQUEST> <ERROR_ORIGIN>...</ERROR_ORIGIN> <ERROR_CONTENT>
Error id	Error id is given by the STATUS
• Error comment	Error comment is given by STATUS

3.19 Module Number CA-5: Admin Interface



3.19.1 Functionality:

The CA-Admin Interface offers the following functions to the administrators :

- CA5_StatsAndLogs to consult data of CA database and Log database with different criteria;
- CA5_ManageAdmin permits to the administrators to manage his database (create RA, create user, create profile, create CA, assign user privilege ...);
- CA11_GenerateOTP is called to permits an OTP generation for different entities.

3.19.2 Function CA5_StatsAndLogs :

This function provide information coming from CA_DataBase and Log_DataBase to the administrator via the CA Audit (CA-8) through the CA API.

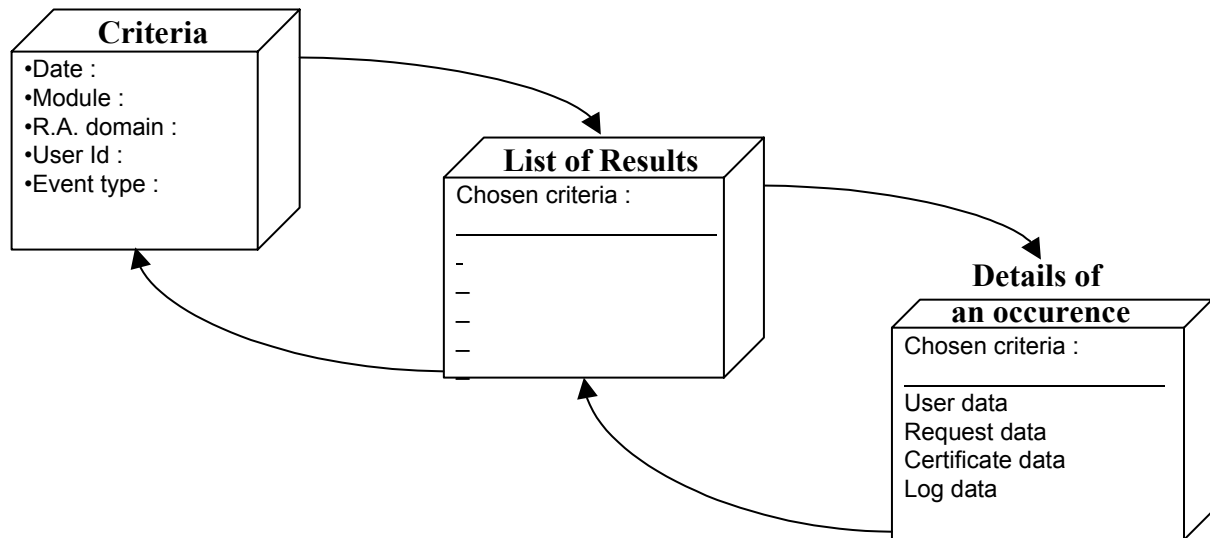
With the admin interface GUI, the administrator has the possibility to query all events specified in the first screen. He also has the possibility to display events with multi criteria.

The criteria to select the events are :

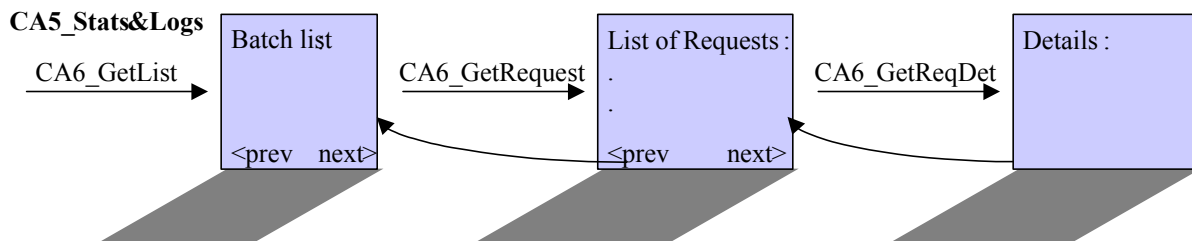
- beginning and ending timestamps

- module name
- Certification Agent
- Registration Agent
- User
- Event type.

In the list of results, the administrator can choose one of them to display all the data concerning that event.



For each criteria, the GUI propose the screens :



The associated functions CA6 are described forward.

3.19.3 Function CA5_ManageAdmin :

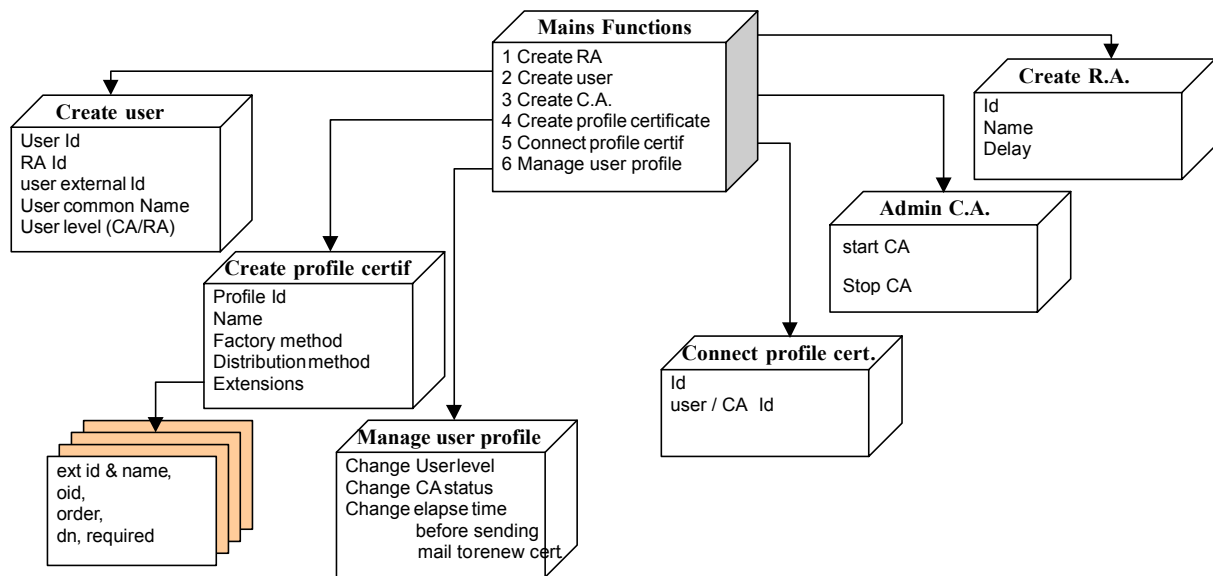
This function supplies the data necessary for the CA database.

The administrator can also change the privilege of a user, the status of a CA or the period of time before a mail is sent for renewing a certificate.

The certificate profile and their extensions can be created from a model. In the function "creation profile certif", all its extensions (required or not) should be created.

When a new profile certificate is created, a message (format XML) is send to the RA to initialise its data base.

All required data are sent to the CA DB module (CA-14) creating a new database occurrence.



3.20 Module Number CA-6: Admin-API

3.20.1 Functionality:

The CA-API is the interface between the services offered by the CA modules and the CA administration GUI.

Before calling the modules "OTP-Generator, CA-DB, Audit or Log", dedicated to the administration of CA, the CA-API module calls the access control module (CA-15) and verifies that the administrator calling is authorised to perform the operation.

Errors are sent to the calling module and to the Log module.

3.20.2 API:

CA6_GetList	
Input parameter:	
ADMIN ID	Administrator Ident
REQUEST STATUS	Status of the job request
REQUEST REQUESTER	Requester
TYPE OF BATCH	bat_type to be selected (in a list of values : C, N, ...)
DATE FROM	Dates between occurs to be listed
DATE TO	
RA IDENT	ra_ident

NB_MAX_OCCURS	100 by default	
Output parameter:		
BATCH LIST	BAT_IDENT	
	BAT_TYPE	
	BAT_REQUESTER	
	BAT_REQUEST_DT	
	BAT_STATUS	
	NUMBER OF REQUEST	
STATUS	Status	

Rules :

When the administrator id is a RA administrator, the occurs to be listed are selected **only** in his RA domain.

When the administrator id is a CA administrator, he can chose in a list of RA domain to be scanned.

CA6_GetRequest

Input parameter:

BATCH IDENT	bat_ident
-------------	-----------

Output parameter:

REQUEST LIST	REQ_IDENT	
	REQ_STATUS	
	KEY_NAME	
	CER_IDENT	if exist
STATUS	Status of the job request	

CA6_GetReqDet

Input parameter:

REQ_IDENT	Request ident
-----------	---------------

Output parameter:

REQ_DATA	Request data
STATUS	Status of the job request

CA6_ManageAdmin	
Input parameter:	
ADMIN ID	Administrator Ident
FUNCTION ID	Function Ident (for example : create user, change user privilege)
CALLING MODULE NAME	Name of calling module
REQUEST DATA	Data set of the administrator's request : <ul style="list-style-type: none"> • Name of requester (usr_common_name of the batch ident) • Name of the user (usr_common_name of the user ident) • Type of batch (bat_type) • Date of request (bat_request_type) • Status of the request (req_status)
Output parameter:	
STATUS	Status of the job request
REQUEST DATA	As in input parameter

This following functions allows to stop or start the CA's module by start or stop the CA-10 job scheduler, the CA-17 job scheduler and the CRL factory jobs.

CA6_StartCA	
Input parameter :	
ADMIN ID	Administrator Ident
CA IDENT	CA Ident
Output parameter :	
STATUS	The field STATUS returns information about the service : <ul style="list-style-type: none"> - A status (Error, Done) - A error Id - A error comment.

CA6_StopCA	
Input parameter :	
ADMIN ID	Administrator Ident
CA IDENT	CA Ident

Output parameter :	
STATUS	The field STATUS returns information about the service : <ul style="list-style-type: none"> - A status (Error, Done) - A error Id - A error comment.

This function forces to generate the CRL.

CA6_ForceCRL	
Input parameter :	
ADMIN ID	Administrator Ident
CA_IDENT	to determine which CRL the process must force
Output parameter :	
STATUS	The field STATUS returns information about the service : <ul style="list-style-type: none"> - A status (Error, Done) - A error Id - A error comment.

3.21 Module Number CA-8: Audit

This module allows the admin interface to display the log entries from the log database and other data from the CA database. The module queries the events stored in the log database and CA database.

For that purpose, it uses the access modules "CA-3 and CA-14".

The module queries the entries specified by the input parameters.

3.22 Module Number CA-9: OTP Distributor

This module distributes the OTP to the end user according to the distribution mode and logs automatically. The OTP status change to S (Sent).

If a errors occurs, the OTP become invalid. Its OTP status change to C (cancelled).

Each call to this module is logged with its results :

Module trace of CA-9 module OTP Distributor :

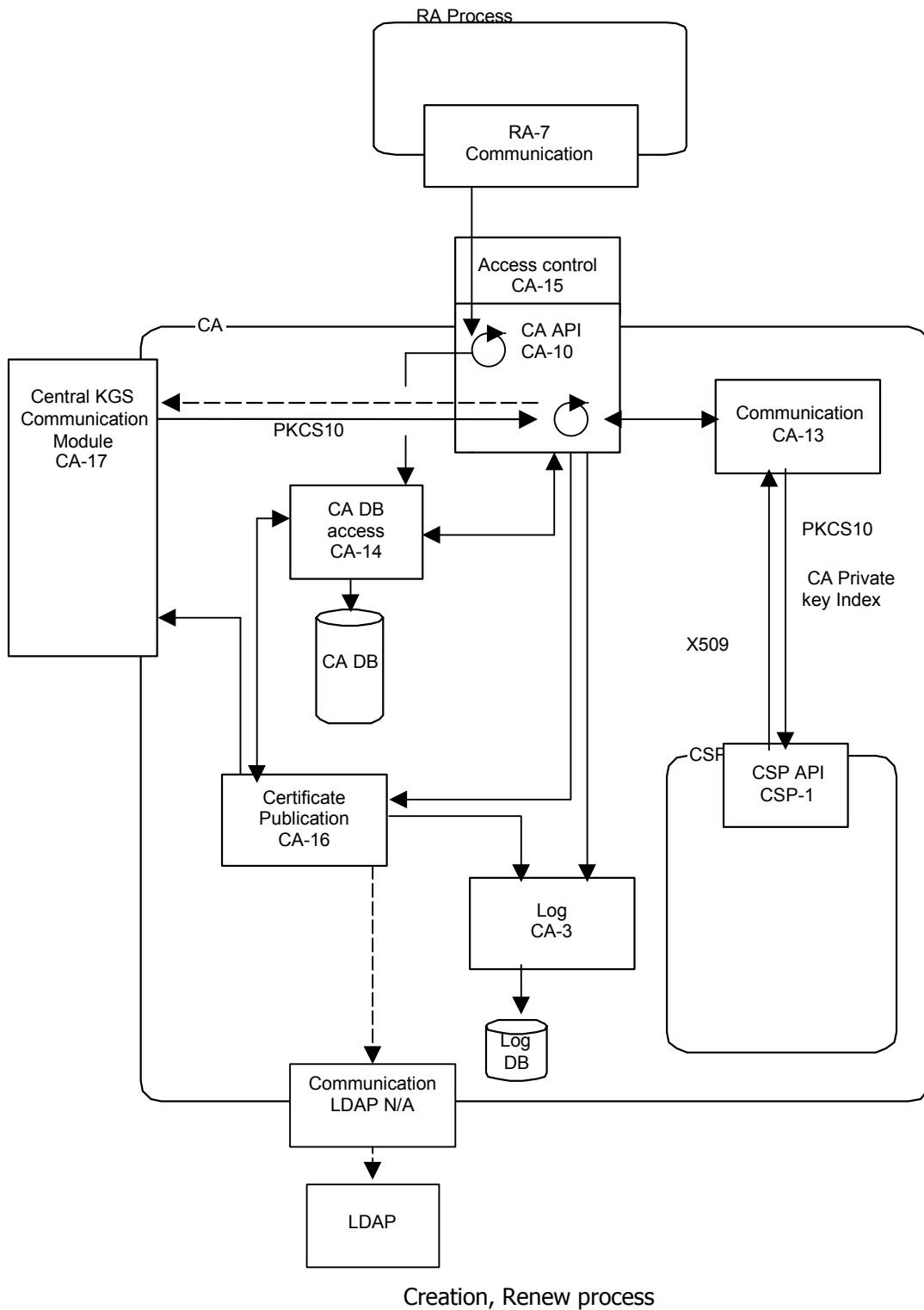
- (Audit) – Audit ident, Time, Module name, Module call, Log content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

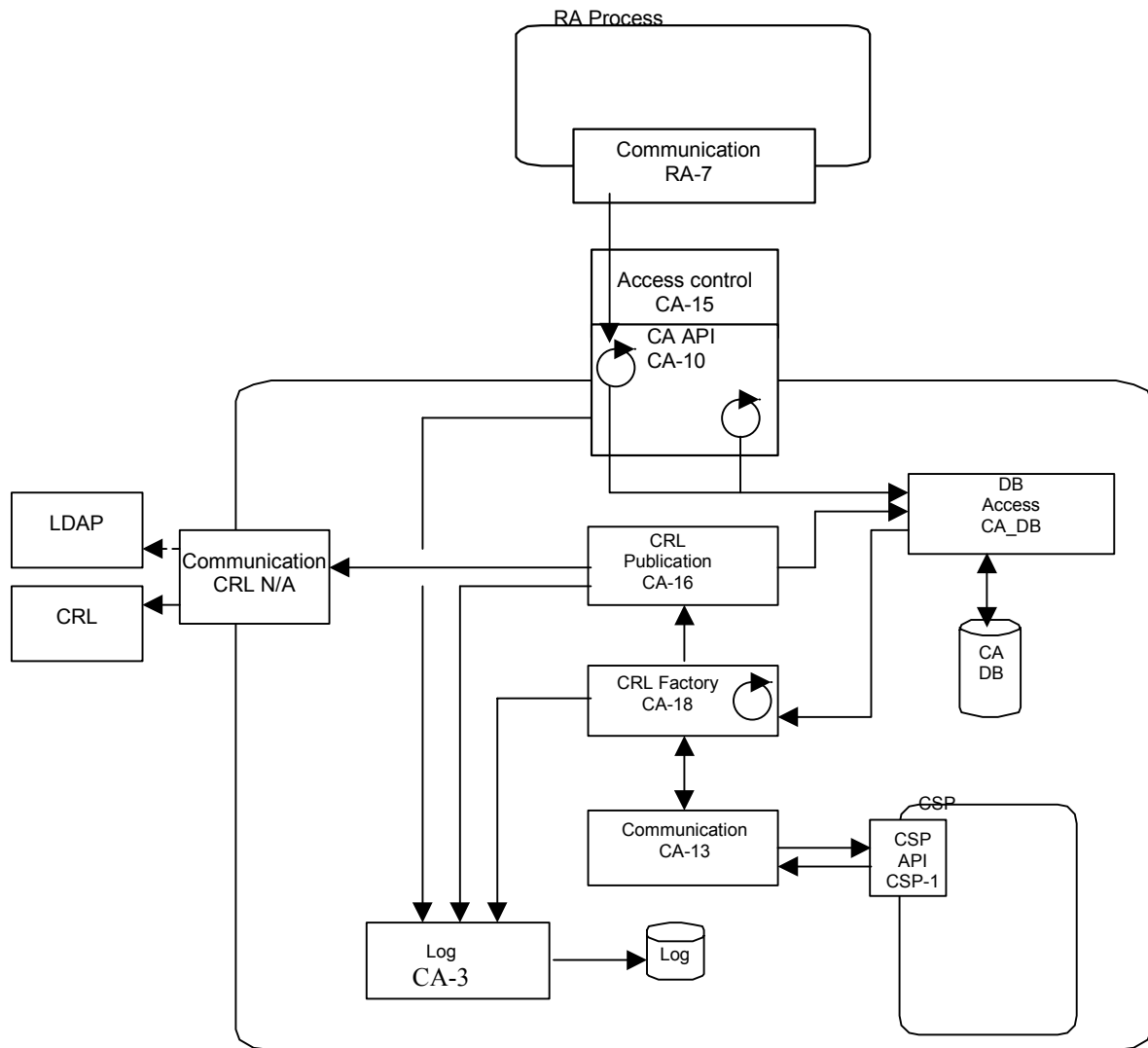
where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'OTPDISTRIBUTOR'
- Module call
- Log_content contains :
 - OTP id
 - End user ident
- Error id is given by the STATUS
- Error comment is given by the STATUS
- Error content contains is given by the STATUS

Error	OTPDISTRIBUTOR_1	Cannot send the mail
Error	OTPDISTRIBUTOR_2	Unknown mail address
Error	OTPDISTRIBUTOR_11	Unknown address
Error	OTPDISTRIBUTOR_100	Internal error (Stack trace if java)
Done	-	-

3.23 Module Number CA-10: CA-API





Suspend, Revocation, Reactivate process.

3.23.1 Functionality

The CA-10 API offers the following services to the RA-7 Communication module through the CA10_JobRequest function :

- Generate Certificate with key generation
- Generate Certificate without key generation
- Renew certificate (with same key)
- Renew certificate (with a new key)
- Reactivate Suspended Certificate.
- Revoke Certificate
- Suspend Certificate

The RA has to transmit his requests in a job file containing one or more requests of the same type. Each request contains this data:

REQUEST_CERTIFICATE or OTP	Certificate of the seeker or OTP of the seeker
REQUEST_SIGNATURE or PROOFOFPOSSESSION	Signature of the demand or The user has the private key
REQUEST_TYPE	Type of request
REQUEST_IDENT	Ident of the request
KEY_NAME	Name of the key
If creation	
CERTIFICATE_PROFILE	Profile of the requested certificate
ENTITY INFORMATION	Information for construct the certificate
PUBLIC_KEY_TO_BIND	Optional – if generated by the requester
If renewal	
REQUEST_IDENT_RENEW	Request ident to use for the renewal
VALIDITY_INTERVAL	new validity interval
PUBLIC_KEY_TO_BIND	Optional – if renewal with a new key pair
If revocation, suspension and reactivation	
REQUEST_IDENT_CREATION	Ident of the creation certificate request

The field REQUEST_CERTIFICATE serves to authenticate the seeker and check the integrity of the data.

The field REQUEST_TYPE serves to identify the type of request and parse the other request information (Creation, Renew, Revocation, Suspend, Reactivation).

The field REQUEST_IDENT and KEY_NAME are transmit by the requester to identify his request. (Its not the certificate ident).

The field CERTIFICATE_PROFILE directs to the set of rules to construct the certificate.

The field ENTITY INFORMATION contains the values of the informations needed to construct the certificate and compliant to the certificate profile. The RA has to check this compliance before the request.

The field PUBLIC_KEY_TO_BIND contains the public key if it is transmit by the requester (not do in central).

The field REQUEST_IDENT_RENEW serves to identify the request to use for the renewal.

The field VALIDITY_INTERVAL serves to defined the new validity interval.

The field PUBLIC_KEY_TO_BIND is optional and allows to change the key.

The field REQUEST_IDENT_CREATION identifies the certificate the RA wants to revoke, suspend or reactivate.

3.23.2 CA10_JobExaminer

Function CA10_JobExaminer is a permanent process like a daemon.

It will examine continuously the job files entry queue from RA.

First, the function will check the format of the job files.

Then, the function will check through module CA-15 Access control module :

- If authentication is made by certificate
 - the authentication of the seeker (and validity of his certificate)
 - the integrity of the data (the signature)
 - the authorisation to ask this request (to verify in the CA data base that the user can use this certificate profile and generate certificate for the user in the request).
- If authentication is made by OTP
 - the authentication of the seeker
 - the proof that the user has the private key
 - the authorisation to ask this request (to verify in the CA data base that the user can use this certificate profile – in this case, the user can just generate certificate for himself).

If all the checks are correct :

- CA10_JobExaminer will write every request of the file in the REQUEST and BATCH table (req_status = V, bat_status = P)
- CA10_JobExaminer will return an OK status to the RA through the CA10_JobRequest function

Else,

CA_10JobExaminer will return an error status to the RA through the CA10_JobRequest function. A occurs is wrote in the log database but not in the CA database.

3.23.3 CA10_RequestScheduler

Function CA10_RequestScheduler is also a permanent process like a daemon.

It will examine continuously the REQUEST table in which CA10_JobExaminer writes the requests to process.

The process depends on the type of the request.

It writes occurs in BATCH and REQUEST tables: req_status=P.

3.23.3.1 Certificate generation with key generation

CA10_RequestScheduler will call CA-17 Central KGS Communication module to ask the KGS to generate a key-pair.

When the CA-17 will return the public key generated, it will complete the certificate request with the public key and call the CA-13 CSP Communication module to sign the certificate with the CA private key. It update the req_status field to R (Ready to send).

When the CA-13 returns the signed certificate, it will call CA-16 Certificate Publication

- to store the certificate on the CA Data Base (req_status = R if OK, insert one occurs in CERTIFICATE table, cer_status = N)
- to call the KGS for the certificate factory (req_status = S if OK)
- optionally transmit the certificate for a publication on a directory (req_status=D if OK or if the certificate doesn't be published : ca.ldap is NULL) and update the cer_status=V or stay to N if a error occurs (include interval validity expired).

If the certificate request is a renew request, a new certificate is generated with the data of the specified request. That means the CA-10 API can extract data from a other request and modify some information as the validity date. For a renew request, the process use the fields profil_cer_factory_renew and profil_cer_distri_renew in the table profil_cer.

3.23.3.2 Certificate generation without key generation

In this case CA10_JobScheduler module call directly the CA-13 CSP Communication module to sign the certificate with the CA private key.

The next step are the same as 3.23.3.1.

3.23.3.3 Certificate renewal with new key

The CA10_RequestScheduler use the initial request to obtain all necessary information. Then it changes the validity interval and get a new key pair.

If it is transmitted by the RA, the public key is in the request, else CA10_RequestScheduler will call CA-17 Central KGS Communication module to ask the KGS to generate a key-pair.

The next step are the same as 3.23.3.1.

Remark: A certificate can be renewed if it is valid. That means its certificate status is V (valid) and can not be E (expired), R (revoked) or S (Suspended).

3.23.3.4 Certificate renewal with the same key

The CA10_RequestScheduler use the initial request to obtain all the necessary information. Then it change the validity interval

The next step are the same as 3.23.3.1.

Remark: A certificate can be renewal if it is valid. That means its certificate status is V (valid) and can not be E (expired), R (revoked) or S (Suspended) or N (New).

3.23.3.5 Certificate revocation

CA10_JobScheduler will call :

- CA-14 CA DB Access to update the status of the certificate on the CA Data Base (change from valid or suspend to revoked).
- If the certificate status is Expired or Revoked, a error is write in the log database and the process are stopped.

3.23.3.6 Certificate suspension

CA10_JobScheduler will call :

- CA-14 CA DB Access to update the status of the certificate on the CA Data Base (change from valid to suspended)
- If the certificate status is Expired, Suspend or Revoked, a error is write in the log database and the process is stopped.

3.23.3.7 Suspended certificate reactivation

CA10_JobScheduler will call :

- CA-14 CA DB Access to update the status of the certificate on the CA Data Base (change from suspended to valid)
- If the certificate status is Expired, Valid or Revoked, a error is write in the log database and the process is stopped.

3.23.3.8 Processes common to all type of requests

If all batch requests are processed, CA10_JobScheduler update the bat_status with D (Done) or E (Error).

At each step of the process, it will update the request table with the called module and its response.

At the end of each step, it will log the results of the called module via the Log Module CA-3.

If there is no response after a time-out to be defined, it will write an error in the request table and in the log.

If an error occurs while the process of a request, the administrator CA has to restart the process (see CA-5 Admin interface). For this, the request status will be update from E (Error) to V (Valid) and the CA-10 CA API can re-execute the process.

Each call to this module is logged :

Module trace of CA-10 module CA API:

- (Audit) – Audit ident, Time, Module name, Module call, Log content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

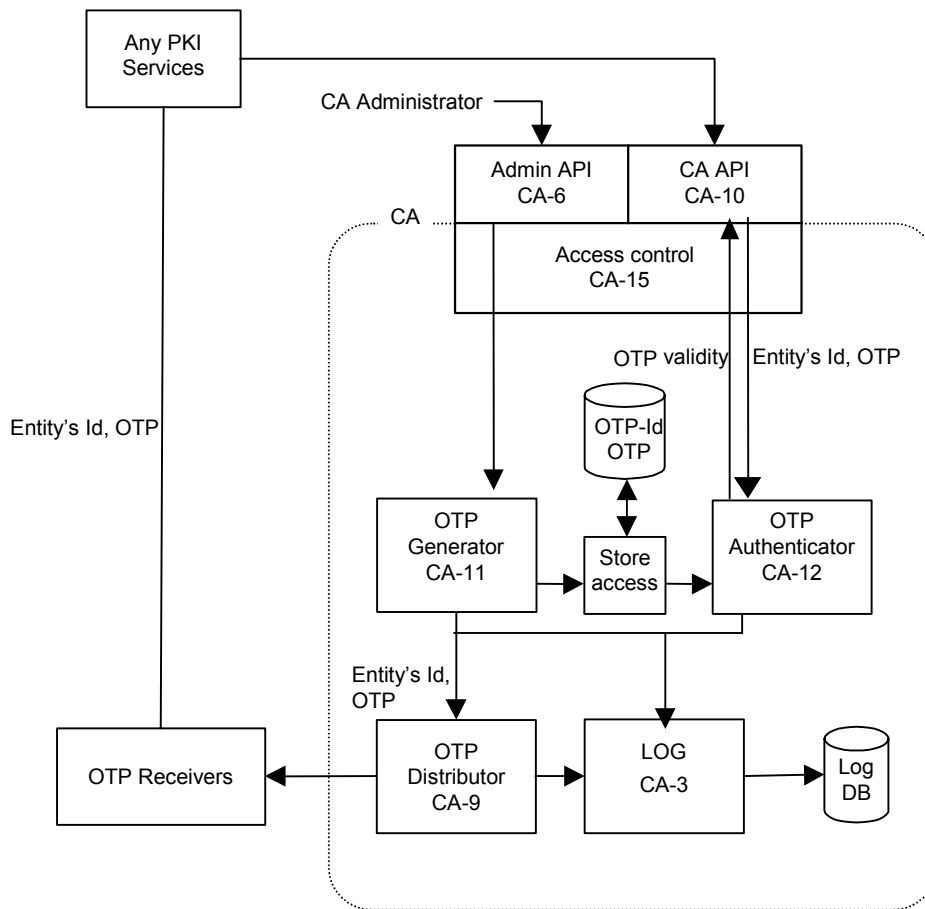
- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'CAAPI'
- Module call
- Log_content contains :
 - requester ident (**ie RA ident for D4.1 consistency**)
 - request data (**ie Request for D4.1 consistency**)
 - **comment (for D4.1 consistency)**
- Error id is given by the STATUS
- Error comment is given by the STATUS
- Error content contains
 - **RA ident**
 - **Request**
 - **Error origin**

Receipt certificate request		
Error	CAAPI_1	Authentication failed (return of CA-15 module)
Error	CAAPI_2	Access data base error (return of CA-14 module) Store request
Key generation		
Error	CAAPI_10	Cannot send message to KGS (key generation)
Error	CAAPI_11	KGS timeout expired (no KGS response)
Error	CAAPI_12	Key generation error (return of KGS)
Ask certificate creation		
Error	CAAPI_20	Signature error (return of CSP module)

Error	CAAPI_21	Access data base error (return of CA-14 module) Update process step signature
Certificate factory		
Error	CAAPI_30	Cannot send message to KGS (store certificate)
Error	CAAPI_31	KGS timeout expired (no KGS response)
Error	CAAPI_32	Access data base error (return of CA-14 module) Update process step store certificate
Certificate publication		
Error	CAAPI_40	Certificate already expired (probably a interval validity error)
Error	CAAPI_41	Cannot publish (return of CA 16)
Error	CAAPI_42	Access data base error (return of CA-14 module) Update process step publishing
Certificate renew		
Error	CAAPI_50	Access data base error (return of CA-14 module) Cannot get the request to renew
Error	CAAPI_51	Certificate to renew does not exist (not exists or status is New)
Error	CAAPI_52	Certificate to renew is Revoked
Error	CAAPI_53	Certificate to renew is Expired
Error	CAAPI_54	Certificate to renew is Suspended
Certificate revocation		
Error	CAAPI_60	Access data base error (return of CA-14 module) Cannot get the certificate to revoke
Error	CAAPI_61	Certificate does not exist.
Error	CAAPI_62	Certificate status is already Revoked.
Error	CAAPI_63	Certificate status is Expired.
Error	CAAPI_64	Access data base error (return of CA-14 module) Update process step revocation
Certificate suspension		
Error	CAAPI_60	Access data base error (return of CA-14 module) Cannot get the certificate to suspend
Error	CAAPI_61	Certificate does not exist (not exists or status is New).

Error	CAAPI_62	Certificate status is already Suspended.
Error	CAAPI_63	Certificate status is Revoked.
Error	CAAPI_64	Certificate status is Expired.
Error	CAAPI_65	Update process step suspension
Certificate reactivation		
Error	CAAPI_70	Access data base error (return of CA-14 module) Cannot get the certificate to reactivate
Error	CAAPI_71	Certificate does not exist (not exists or status is New).
Error	CAAPI_72	Certificate status is already Valid.
Error	CAAPI_73	Certificate status is revoked.
Error	CAAPI_74	Certificate status is expired.
Error	CAAPI_75	Update process step reactivation
Internal CA10 error		
Error	CAAPI_100	Internal error (Stack trace if java)
Done	-	-

3.24 Module Number CA-11: OTP Generator



A admin CA or admin RA can request a OTP for an end user by the admin interface (the authentication is made by the connection to the Admin interface).

If this user does not exist, he is created in the user table. By default, he has no rights to generate certificate (no links between usr and usr_profil_cer) and his status is end user (usr_admin_ca= usr_admin_ra=usr_ra=N).

If this OTP is generated for a user who need to request certificate, the administrator must authorise this user to access to one or more certificate profile and optionally change his status.

If the requester is a admin RA, he can just create a OTP for a user in his RA domain.

If the requester is a admin CA, he can choose a RA domain.

To authorise a end user to use one or several profile certificates, they must be linked to the RA domain of the end user.

CA11_OTPGenerator requests an OTP.

CA11_OTPGenerator	
Input parameter :	
REQUESTER_IDENT	Requester ident

ENTITY_IDENT	Entity ident
USR_IDENT	End user ident
VALIDITY	Interval validity
MODE_DISTRIB	Distribution mode
DATA_DISTRIB	Data used for the distribution
Output parameter:	
OTP	An OTP
STATUS	Status

The field REQUESTER_IDENT:

- usr_ident of the requester

The field ENTITY_IDENT :

- ra_ident NULL if the requester is Admin RA

The field USR_IDENT :

- usr_common_name can be NULL if user already exists
- usr_ext_ident not NULL

The field validity :

- not_before_dt if NULL, SYSDATE will be used in the data base
- not_after_dt NULL if no delay

The field MODE_DISTRI :

- Indicate the distribution mode (MAIL, LETTER)

The field DATA_DISTRI :

- Contains information to send the OTP via the asked distribution mode - the data format can be XML and different for each distribution mode.

The field OTP :

- OTP_ID
- OTP_VALUE

The field STATUS :

- A status (Error, Done)
- A error Id
- A error comment.

Each call to this module is logged with its results :

Trace module CA-12 Certificate Update Scheduler:

- (Audit) – Audit ident, Time, Module name, Module call, Log_content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'OTPGENERATOR'
- Module call
- Log_content contains :
 - Requester ident
 - RA Ident (Entity id for D4.1 consistency)
 - User ext_ident
 - OTP ident (OTP for D4.1 consistency)
- Error id is given by the STATUS
- Error comment is given by the STATUS
- Error content
 - **User id (for D4.1 consistency)**
 - **Error origin (for D4.1 consistency)**

Error	OTPGENERATOR_0	Access data base error (return of CA-14 module)
Error	OTPGENERATOR_1	RA domain does not exist
Error	OTPGENERATOR_2	otp_not_after_dt > otp_not_before_dt or SYSDATE > otp_not_before_dt
Error	OTPGENERATOR_3	Unknown mode distribution
Error	OTPGENERATOR_4	Distribution data not complete – not well formatted
Error	OTPGENERATOR_5	There is already a valid OTP for this user
Error	OTPGENERATOR_100	Error (Stack trace if java)
Done	-	-

3.25 Module Number CA-12: OTP Authenticator

CA12_OTPAuthenticator checks the validity of a user ID with OTP. Once the validation obtained, the module returns the positive answer.

CA12_OTPAuthenticator	
Input parameter:	
ENTITY_IDENT	Entity ident
OTP	An OTP
Output parameter:	
STATUS	Status

The field RA_IDENT :

- ra_ident

The field OTP :

- OTP_ID
- OTP_VALUE

The field STATUS :

- A status (Error, Done)
- A error Id
- A error comment.

Each call to this module is logged with its results :

Trace module CA-12 Certificate Update Scheduler:

- (Audit) – Audit ident, Time, Module name, Module call, Log_content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'OTPAUTHENTICATOR'
- Module call
- Log_content contains :
 - otp_ident
 - **ra ident** (for D4.1 consistency)
 - **status field of STATUS** (for D4.1 consistency)
- Error id is given by the STATUS

- Error comment is given by the STATUS.

Error	OTPGENERATOR_0	Access data base error (return of CA-14 module)
Error	OTPGENERATOR_1	OTP does not exist
Error	OTPGENERATOR_2	Requester's OTP not authorised
Error	OTPGENERATOR_3	Not a valid validity interval OTP
Error	OTPGENERATOR_4	OTP not sent to the end user
Error	OTPGENERATOR_100	Error (Stack trace if java)
Done	-	-

A valid OTP must verify:

- the OTP requester is authorised (Verify requester rights. He rights can be changed since the OTP creation).
- the OTP is found in the data base.
- the interval validity OTP is valid :

$otp_not_before_dt < SYSDATE < otp_not_after_dt$ if $otp_not_after_dt$ is not NULL

$otp_not_before_dt < SYSDATE$ if $otp_not_after_dt$ is NULL

- the otp status is S (Sent)
- Its status can be changed to U (used) in the data base.

3.26 Module Number CA-13: Communication to CSP

See D4.2.

3.27 Module Number CA-14: CA Database

CA-14 module CA Database is a technical module who allows any CA module to access to the CA database.

This module can be changed if the database is changed (for example mySQL to postgreSQL).

Each call to this module is logged with its results :

Trace of module CA-14 CA DB :

- (Audit) – Audit ident, Time, Module name, Module call, Log content

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'ACCESSCONTROL'
- Module call: WRITEMAIL, ADMINAPI, AUDIT, CAAPI, OTPGENERATOR, OTPDISTRIBUTOR, OTPAUTHENTICATOR, ACCESSCONTROL, PUBLICATION, CRLFACTORY, CERTIFICATEUPDATEAGENT
- Log_content contains :
 - **Ra ident** (for D4.1 consistency but optionally NULL)
 - **Data to be stored or to be read**

Error	LOG_1	Cannot access to the data base (the data base is stopped...)
Error	LOG_2	Access denied (password error...)
Error	LOG_100	"Data base error"
Done	-	-

This module can write log when it is accessed but does not write error in the data base.

For example, a 'NO_DATA_FOUND' error can be a error if the calling module is the CA-15 module and it try to identify a usr, and not a error if the calling module is the CA-6 module which try to get a list which can be empty.

3.28 Module Number CA-15: Access Control

3.28.1 *Functionality:*

This module checks all accesses to the CA :

- All requests by the admin interface (CA-5 and CA-6) for :
 - Audit
 - User management
 - OTP generation
 - CA creation
- All requests by the RA (CA-10) for :

- Certificate request
 - Revocation request
 - Renew Request
 - Suspend request
 - Reactivate request
- All requests by a PKI module (CA-10) for :
- OTP Validity request

All other external accesses to the CA will be denied except by the Central KGS Communication CA-17.

To access to the CA module by CA-5 and CA-6, the CA-15 access control module checks :

- The validity of the connected user certificate
- Authentication of the seeker with the certificate
- His rights in the CA database (Is he a admin CA, an admin RA or a RA).

To access to the CA module by CA-10, the CA-15 access control module checks :

- The validity of the connected user certificate or the user OTP
- Authentication of the seeker with the certificate or the OTP
- His rights in the CA database
 - He is in his entity RA
 - He has rights to use this profile certificate
 - He has right to generate certificate for the user in the request:
 - if the requester is a RA, admin RA or admin CA, he can generate a certificate for a user
 - else he can just generate a certificate for himself and must exist in the ca data base (he has a OTP or at least one certificate)
- The integrity of the data by checking the signature or the proof of possession.

The function checks the rights of the specified operator to perform the particular operation. The function returns if the operation is granted or if the execution of the operation is denied.

CA15_checkAccessRight	
Input parameter:	
Certificate or OTP	Requester's authentication
Signature or Proofofpossession	Data integrity

Output parameter:	
STATUS	Status

The field Certificate :

- Requester's certificate or his OTP which serves to identify the requester and his right in the data base.

The field Signature:

- Verify the data integrity.

The field STATUS :

- A status (Error, Done)
- A error Id
- A error comment.

Each call to this module is logged with its results :

Trace of module CA-16 Publication :

- (Audit) – Audit ident, Time, Module name, Module call, Log content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'ACCESSCONTROL'
- Module call: CAAPI, AUDIT
- Log_content contains :

if the calling module is AUDIT :

- **Ra ident** (for D4.1 consistency but optionally NULL)
- Certificate
- Rights : Admin_CA, Admin_RA (not access to the same Admin CA interface function)
- **Authentication result** (for D4.1 consistency but we cannot know the result before the process end)
- **The reason of the failure** (for D4.1 consistency but we cannot know the result before the process end)

if the calling module is CAAPI:

- **Ra ident** (for D4.1 consistency)
- Certificate or OTP
- Type of request
 - **Authentication result** (for D4.1 consistency but we cannot know the result before the process end)
 - **The reason of the failure** (for D4.1 consistency but we cannot know the result before the process end)
- Error content
- **Ra ident** (for D4.1 consistency but can be NULL)
- **Error origin** (for D4.1 consistency)
- Error id is given by the STATUS
- Error comment is given by the STATUS

Error	ACCESSCONTROL_0	Access denied: Not a valid OTP (return of CA-12)
Error	ACCESSCONTROL_1	Access denied: Not a valid certificate – certificate signature altered
Error	ACCESSCONTROL_2	Access denied: Not a valid certificate – certificate expired
Error	ACCESSCONTROL_3	Access denied: Not a valid certificate – certificate revoked
Error	ACCESSCONTROL_4	Access denied: Not a valid certificate – certificate suspended
Error	ACCESSCONTROL_5	Access denied: Not a valid certificate – certificate authority not valid
Error	ACCESSCONTROL_6	Access denied: Insufficient right – not authorised to use this certificate profile
Error	ACCESSCONTROL_7	Access denied: Insufficient right – not authorised to create certificate for this user.
Error	ACCESSCONTROL_8	Access denied: Unknown requester
Error	ACCESSCONTROL_9	Access denied: Data altered
Error	ACCESSCONTROL_10	Access denied: Bad format data – cannot be read.
Error	ACCESSCONTROL_11	Access denied: Missing required data.
Error	ACCESSCONTROL_100	Internal error (Stack trace if java)
Done	-	-

3.29 Module Number CA-16: Publication

This module is called by the CA-10 CA API module for creation or renew certificate.

In this case, it calls the KGS10 module

- to store in a specified format by `profil_cert_factory_method`
- to send the certificate to the user by the method `profil_cert_distri_method`

When the KGS module response to the request the `req_status` is updated to 'S' (Sent) or 'E' if error

If the field `ca_ldap` is not NULL in the CA database, the CA-16 module publish the certificate in the LDAP.

At the end of this step, `req_status` is updated to D (Published - Done) or 'E' if a error occurs.

When the request status get the 'D' value, the certificate status is update from 'N' (New) to 'V' valid.

This module is called by the CA-18 CRL factory module for revocation, suspend and reactivate certificate.

At the end of this step, `req_status` is updated to D (Published - Done) or E if a error occurs. The certificate status change :

- from S to V (reactivate)
- or from S to R (revocation)
- or from V to S (suspend)
- or from V to R (revocation)

If all the request in a batch are done, `bat.status` is update to 'D'

It update the LDAP too to insure the coherence between CRL and LDAP.

Each call to this module is logged with its results :

Trace of module CA-16 Publication :

- (Audit) – Audit ident, Time, Module name, Module call, Log_content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'PUBLICATION'
- Module call: CRLFACTORY, CAAPI
- Log_content contains :

- Cer_ident
- Process type: CRL, LDAP, Purge_LDAP
- **RA ident** (for D4.1 consistency)
- **Mail ident** (for D4.1 consistency)
- Error content
 - **RA ident** (for D4.1 consistency)
 - **Request** (for D4.1 consistency)
 - **Error origin** (for D4.1 consistency)
- Error id is given by the STATUS
- Error comment is given by the STATUS.

Error	PUBLICATION_0	Access data base error (return of CA-14 module)
Error	PUBLICATION_1	Cannot publish
Error	PUBLICATION_100	Internal error (Stack trace if java)
Done	-	-

3.30 Module Number CA-17: Communication to KGS

See D4.2

3.31 Module Number CA-18: CRL Factory

It publish the CRL:

- add revoked and suspend certificates
- erases reactivate certificates

If the field ca.ldap is not null in the database, the module CA-16 publication

- erases revoked, suspend and expired certificate from the LDAP
- publish reactivate certificate

At regularity scheduled intervals the CRL Factory module builds a CRL from the certificates stored in the CA DB.

For each valid CA :

- The daemon get the revoked and suspended certificate list (which are valid validity interval) to publish it in the CRL and to remove from the LDAP.

It use ca and certificate tables to obtain the complete list (Replace the old CRL).

- The daemon get the reactivate certificate list which are valid validity interval to publish it in the LDAP (mandatory if ca.ldap is not null to insure the integrity with the CRL).

It use ca, batch, request and certificate tables to obtain the list : request.status = P, batch.status = P and bat_type = reActivate because the other certificates are already published.

- The daemon get the valid certificate list which has validity interval expired. Then it change the certificate status to E (expired) in the data base and if ca.ldap is not null remove the certificate from the LDAP.

It use ca and certificate tables to obtain the list.

- The daemon get the suspended certificate which has validity interval expired. Then it pass its status to E (expired). There is no modification in CRL or LDAP in this case, just in the database.

It use ca and certificate tables to obtain the list.

Each call to this module (one call by CA) is logged with its results :

Trace of module CA-18 CRL Factory :

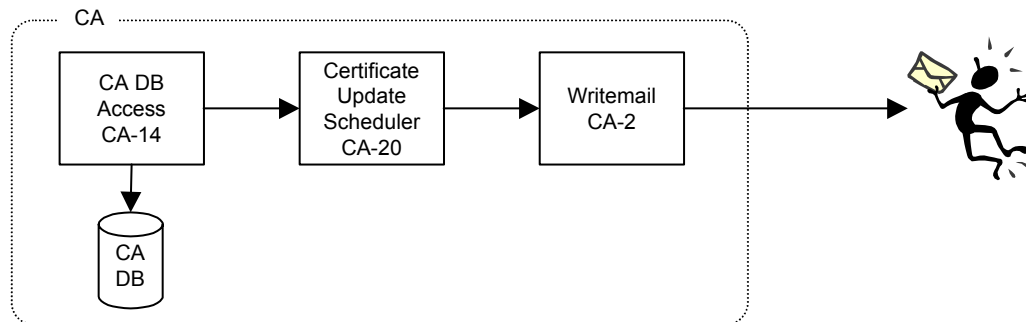
- (Audit) – Audit ident, Time, Module name, Module call, Log_content
- (Error) – Audit ident, Time, Module name, Module call, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'CRLFACTORY'
- Module call : ADMINAPI, daemon
- Log_content contains :
 - Ca_ident
 - Nbe_certif_crl
 - Nbe_certif_ldap
 - Nbe_certif_ldap_purge
 - Operator Ident (not NULL if a admin CA force the CRL)
 - Event : CRL generated **(for D4.1 consistency but cannot know before the process start)**
- Error content
- **Error origin (for D4.1 consistency)**
- Error id is given by the STATUS

- Error comment is given by the STATUS.

3.32 Module Number CA-20: Certificate update agent



Certificate Update Scheduler CA-20 is operated regularly to obtain the list of certificate who will be expired and alert the user by mail (field mail_name, mail_address in the CA database).

The delay to operate the CA-20 module is stored in the CA database by RA domain in the mail_ref table. It can be made too on demand by the CA admin.

It calls the function **CA20_CertificateUpdateScheduler**.

CA20_GetCertificateUpdateScheduler requests the CA database to obtain the list of certificates to be renewed.

Then it builds the content of the mail and sends a mail to a user for each list of certificates by calling the **CA2_WriteMail** function.

This list must verify :

- All certificates will expire on :
 $\text{SYSDATE} + \text{Mail_ref.delay} \geq \text{certificate.cer_not_after}$
 (Mail_ref.delay can be different by RA – see the CA database)
- There is no already a renewal request for a certificate of this list
- The certificate must not be revoked
- Mail_ref.delay must not be NULL (the admin CA has requested a renew alert for all certificates of user in a given RA).
- A mail address is known to send a mail : There exists a occur in mail table for the list certificate.
- A mail has not already sent (mail.renewalert IS NULL).

CA20_GetCertificateUpdateScheduler
Input parameter :

Output parameter :	
LISTE_RENEW	List of renewal certificate
STATUS	Status

The field LISTE_RENEW :

- cert_ident : certificat ident to renew
- mail_name : user name to send the mail
- mail_address : user mail address

The field STATUS :

- A status (Error, Done)
- A error Id
- A error comment.

Each call to this module is logged with its results :

Trace of module CA-20 Certificate Update Scheduler :

- (Audit) – Audit ident, Time, Module name, Module call, Log_content
- (Error) – Audit ident, Time, Module name, Module call, Error content, Error id, Error comment

where :

- Audit ident is a unique ident for the table audit, a links to the audit table for the table error
- Time is SYSDATE
- Module name is 'CERTIFICATE_UPDATE'
- Module call: daemon
- Log_content contains :
 - If calling module is CA-6 Admin API
 - CA administrator ident
 - If calling module is a scheduler :
 - empty
- Error id is given by the STATUS
- Error comment is given by the STATUS.

3.33 Module Number CSP-1: CSP-API

3.33.1 Functionality:

The CSP-API provides the interface to the services offered by the CSP modules towards the CA. Before calling the modules CSP-2 or CSP-3, dedicated to the signing of certificates and CRLs respectively, the CSP-API module calls the access control module (CA-15) and verifies that the caller is authorised to perform the requested operation.

3.33.2 API:

Function name:

CSP1_Sign_Certificate

Input Parameters:

TBS X.509 certificate

Key Index

Output Parameters:

Signature

Function name:

CSP1_Sign_CRL

Input Parameters:

TBS X.509 CRL

Key Index

Output Parameters:

Signature

3.34 Module Number CSP-2: Certificate Signer

3.34.1 Functionality:

Producing the signature for a given "TBS certificate", using a key identified by key index (call of Hash function, retrieving RSA key, calling signature function, returning signature).

The key(s) to be used by the CA are managed via a the CA-DB. For each key index known in this CA the CA-DB keeps information on where the key is to be found (e.g. a crypto device serving as crypto engine or just a file to be used in a pure software implementation).

3.34.2 API:

Function name:

CSP2_Sign_Certificate

Input Parameters:

TBS X.509 certificate
Key Index

Output Parameters:

Signature

3.35 Module Number CSP-3: CRL Signer

as CSP-2, but with function name CSP2_Sign_CRL and X.509 TBS CRL as input

3.36 Module Number CSP-4: Crypto Engine**3.36.1 Functionality:**

This module implements the cryptographic functions needed by the EU-PKI CA, i.e. for signing X.509 certificates and X.509 certificate revocation lists. The EU-PKI CA will use the SHA-1 as hash algorithm and the RSA as signature algorithm. So in detail the necessary cryptographic functionality is:

SHA-1 hash algorithm

RSA: Generation and verification of RSA signatures as defined in PKCS #1 v2.0 [RFC 2437], key length at least 1024 bit, up to 2048 bit should also be supported.

3.36.2 API:

according to OpenSSL Crypto library interface (see especially sha.h, rsa.h and engine.h) as defined in OpenSSL version 0.9.6g-engine - if version 0.9.7 is released in time it can be used instead).

see <http://www.openssl.org/docs/crypto/crypto.html>

This allows to have the actual crypto functionality in a hardware device (engine) or implemented in software.

3.37 Module Number KC-2: Certificate Factory

Reuse central KGS modules.

3.38 Module Number KC-5: Admin Interface**3.38.1 Prerequisite**

To perform the KC, you need a complete environment :

- The central KGS module
- The CSP module
- One database
- One CA (the different modules are installed et configured)
- To have defined six default certificate profile :

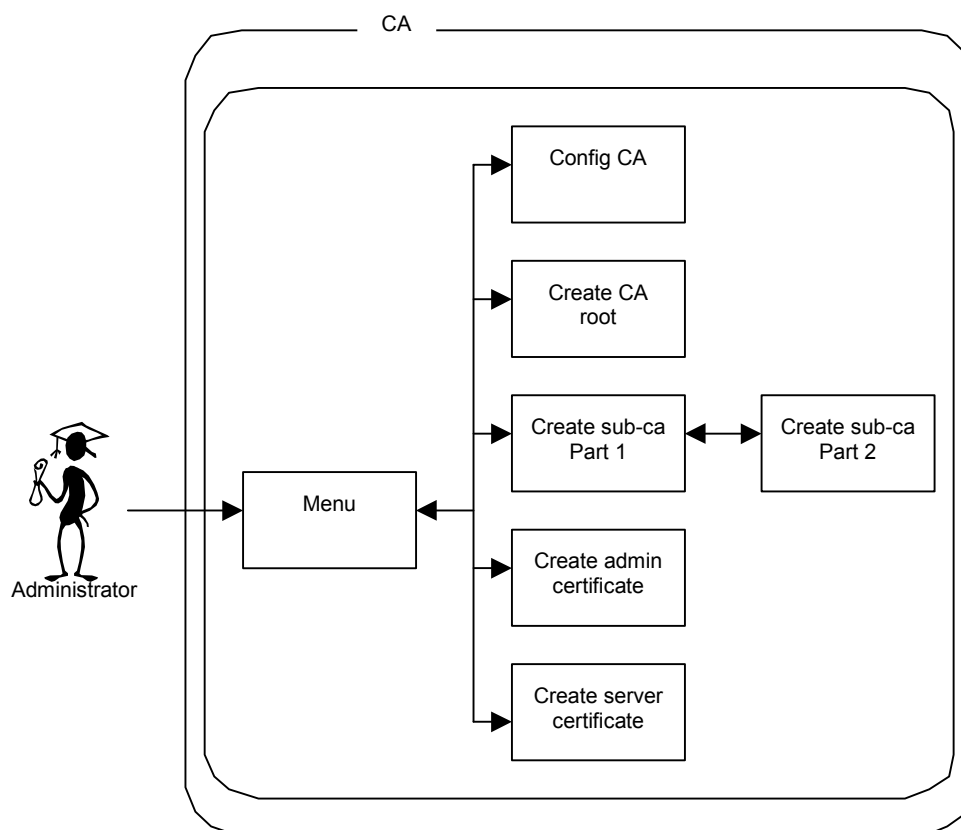
- policy_ca to create a CA certificate
- policy_server to create a server certificate (to use the admin IHM CA-5)
- policy_admin to create a administrator's CA certificate
- policy_admin_ra to create a administrator's RA certificate
- policy_ra to create a RA's certificate
- policy_any to create any policy defined by the Admin Interface CA-5.

The CSP module must be able to access to the private key stored and protected in the Centrale KGS with a reference. This key is protected (by password, crypto or any another system).

That means the different key of the CA cannot be generated out of the data centre where is the CA.

The GUI and the API can be a unique executable. But it work with the KGS, so it will be asynchrony and work with messaging system.

3.38.2 Key ceremony interface



A first screen display five options a administrator can use:

- One screen serves to configure the CA (root or not)

- One screen allows to create a private key and a self signed certificate (use to initiate the root CA).
- Two screens allow to create a private key and generate one certificate signed by the root CA (use to initiate a sub-ca).
- One screen to create a administrator's CA certificate (to access to the CA-5 module)
- One screen to create a server certificate (for the server which display the administrator's interface CA-5)

3.38.2.1 Configuration's screen

This screen allow to see and change the configuration of the CA module.

For example, if the CA uses openssl, this screen can display the openssl.cfg.

3.38.2.2 Creation root certificate's screen

This screen allows to generate a private key in the Central KGS module and introduces the PKCS#10 to the CSP module with a reference to the private key to self signed the PKCS#10.

The policy policy_ca is implicitly used to create this certificate.

Then, it initializes the CA by importing the certificate, the private key and insert occurs in the CA database tables (CA_cert and CA).

For example: If it uses openssl, this last step consists to copy the certificate in a directory (see openssl.cfg).

If it uses openssl, it is mandatory to modify it

- to use the CA DB and not the index.txt file
- to use a private key index to access the private key in the KGS (not in the openssl directory /ca/private)

one other solution consist to extract the private key on a token (Central KGS) then to copy manually in the directory /ca/private/

When you valid this screen data, the function KC_GetCARootCertificate is called.

3.38.2.3 Creation sub ca certificate screen

This two screens allow to generate a private key in the Central KGS module and introduce the PKCS#10 to the CSP module with a reference to the root CA private key to signed the PKCS#10.

The policy policy_ca is implicitly used to create this certificate.

Then, there initialize the CA by importing the certificate, the private key and insert occurs in the CA database tables CA_cert and CA).

First screen: Create a key pair and generate a PKCS#10.

When you valid this screen data, the function KC_GetCAKey is called.

Second screen: Create the certificate and initiate the PKI.

When you valid this screen data, the function KC_GetCACertificate is called.

3.38.2.4 Administrator CA certificate creation screen.

This screen allows to generate a private key in the Central KGS module and introduce the PKCS#10 to the CSP module with a reference to the CA private key to signed the PKCS#10.

The policy policy_admin is implicitly used to create this certificate.

Then it calls the Central KGS to store the certificate in a PKCS#12 to be distribute to the administrator CA.

It inserts occurs in the CA database tables User, Certificate.

user.ra_ident and user.user_ext_ident are NULL

user.admin_ca=Y

there are no occurs in tables batch and request

When you valid this screen data, the function KC_GetAdminCACertificate is called.

3.38.2.5 Server certificate creation screen

This screen allows to create a ssl certificate for the CA server.

It is mandatory to have a certificate to use the admin ca interface (CA-5) but this certificate can be issue by any CA.

This step get a certificate and a private key usable by the server (apache/modssl is the WEB server is Apache).

The policy policy_server is implicitly used to create this certificate.

When you valid this screen data, the function KC_GetServerCertificate is called.

3.39 Module Number KC-6: Admin API

KC_GetCARootCertificate	
Input parameter :	
DN	Distinguished name of the root CA
FILE_OUTPUT	Contains the root certificate
Output parameter :	

STATUS	Status
--------	--------

The field DN :

- Distinguished name of the root'CA

The field FILE_OUTPUT :

- The name of the output file which contains the root certificate

The field STATUS returns information about the service :

- A status (Error, Done)
- A error Id
- A error comment.

KC_GetCAKey	
Input parameter :	
DN	Distinguished name of the sub CA
FILE_OUTPUT	Contains the PKCS#10
Output parameter :	
REF_KEY	A reference to the private key
STATUS	Status

The field DN :

- Distinguish name of the CA

The field FILE_OUTPUT :

- The name of the output file which contains the PKCS#10 to do sign

The field STATUS returns information about the service :

- A status (Error, Done)
- A error Id
- A error comment.

KC_GetCACertificate	
Input parameter :	
FILE_INPUT	The name of the file which contains the PKCS#10
REF_KEY	Reference to the private key
CA_ROOT	Root CA Ident to use to sign the PKCS#10

FILE_OUTPUT	Contains the sub CA certificate (.cer)
Output parameter :	
STATUS	Status

The field FILE_INPUT :

The name of the file which contains the PKCS#10 to sign

The field REF_KEY :

Reference to the private key attached with the PKCS#10

The field CA_ROOT :

- Root CA Ident to use to sign the PKCS#10

The field FILE_OUTPUT :

- The file output which contains the sub CA certificate (.cer)

The field STATUS returns information about the service :

- A status (Error, Done)
- A error Id
- A error comment.

KC_GetAdminCACertificate	
Input parameter :	
DN	Distinguished name of the admin CA
PASS_PHRASE	The pass phrase for protect the PKCS#12
FILE_OUTPUT	The output file name which contains PKCS#12
Output parameter :	
STATUS	Status

The field DN :

- Distinguished of the admin CA

The field PASS_PHRASE :

- The pass phrase for protect the PKCS#12 (must be enter by the Admin CA)

The field FILE_OUTPUT :

- The file output which contains the PKCS#12 certificate

The field STATUS returns information about the service :

- A status (Error, Done)

- A error Id
- A error comment.

KC_GetServerCertificate	
Input parameter :	
DN	Distinguished name
FILE_OUTPUT_CERT	The name of the output file which contains the certificate
FILE_OUTPUT_KEY	The name of the output file which contains the private key
Output parameter :	
STATUS	Status

The field DN :

- Distinguished name

The field FILE_OUTPUT_CERT :

- The name of the file which will contain the server's certificate

The field FILE_OUTPUT_KEY :

- The name of the file which will contain the private key.

The field STATUS returns information about the service :

- A status (Error, Done)
- A error Id
- A error comment.

3.40 Module Number KC-15: Access Control

All the KC ceremony is realised in the data centre on the CA server without extern network. The access control is supported by the data centre.

The rights to execute this application is determined by the exploitation system: the operator must be a root user for example.

3.41 Module Number KC-26: Key Pair Factory

For this part, the central KGS does support HSM function:

- it insure this function itself
- it use a HSM (nCipher for example)

The CA key generation can be made by the same method a certificate key.

We just add two tag to indicate

- it is a CA key (the key cannot be exported in one support)
- pass phrase list (a pass phrase list to protect the export key)
 - file name
 - pass phrase (5 for example)

3.42 Module Number KC-30: Secret Export

The generate key is shared in several file protected by the pass phrase. Each file can be save in an other support (floppy disk...) and are removed from the CA server.

3.43 Module Number KC-31: Secret Printing

The different administrator are responsible of their pass phrase.

3.44 Module Number KC-32 Secret Import

Some administrator are mandatory to recreate the key (3 on 5 for example).

3.45 Module Number KGS-3: Log

3.45.1 Functionality:

This module is used as a interface between the Central KGS and the log database. The log module accepts requests from the Central KGS module and forwards it to the database.

The event, which should be logged, includes the information about the performed action or the occurred error. This information includes the related entity, certificate and the certification authority. Additional to that a parameter specifies the type of the log entry (*event, warning, error* etc.).

The KGS database module is used as interface between the KGS log module and the KGS database. The log information are stored in the KgsLogEvents of the KGS database.

3.45.2 API:

Function name:

KGS3_errorLog

Input Parameters:

time

source module identifier

CA/operator identifier

Error origin
error identifier
comment

Output Parameters:

status code

Function name:

KGS3_auditLog

Input Parameters:

time
source module identifier
CA/operator identifier
entity identifier (if known)
action performed

Output Parameters:

status code

Function name:

KGS3_accessLog

Input Parameters:

time
source module identifier
CA/operator identifier
authentication result
reason of failure (optional)

Output Parameters:

status code

3.46 Module Number KGS-5: Admin Interface

3.46.1 Functionality:

This module is the interface between the administrator and the Central KGS. The objective of this module is to give the administrator the possibility to consult the statistics and the logs of the Central KGS unit. Further the module generates a interface for the management of users and rights.

The module requests the log entries via the module KGS-6 Admin-API from the log database. The entries will be produced and displayed in a human readable way. The administrator has the possibility to query the log entries for a specified certificate, a specified entity and/or for a specified CA. The interface must have the possibility to enter the query parameters.

Beside the consultation of the log entries, the interface displays the statistics about the activities of the Central KGS.

Further the admin interface gives the possibility to administrate the rights of users on the Central KGS and to define for example a new administrator.

3.47 Module Number KGS-6: Admin-API

3.47.1 Functionality:

This module provides the functionality for the admin interface. It is the interface between the admin interface, which displays the information on the admin screen, and the audit module and the user database.

3.47.2 API:

Function name:

KGS6_addAdminRights

Input Parameters:

user identifier

Output Parameters:

status code

Description:

Adds administrator privileges to the specified user.

Function name:

KGS6_delAdminRights

Input Parameters:

user identifier

Output Parameters:

status code

Description:

Removes administrator privileges of the specified user.

Function name:

KGS6_dispStatistics

Input Parameters:

none

Output Parameters:

status code

Description:

Receives the statistics information from the audit module and displays it in a human readable way.

Function name:

KGS6_dispLogEntries

Input Parameters:

source module identifier (optional)

entity identifier (optional)

CA identifier (optional)

log entry type (optional)

Output Parameters:

status code

Description:

Receives the log entries from the audit module and displays them in a human readable way.

3.48 Module Number KGS-8: Audit

3.48.1 Functionality:

This module allows the admin interface to display the log entries from the log database. The module queries the events stored in the log database. With the admin interface the administrator has the possibility to query all events of a specified source module, entity or CA. He also has the possibility to display events of a specific type (e.g. *error*).

The module queries the entries specified by the input parameters. The events received from the database can be iterated by the functions **KGS8_firstEvent** and **KGS8_nextEvent**.

At run time several instances of this module can exist. Each instance stores the results of the specified queries. If a new query is executed (by the command **KGS8_queryEvents**) or the instance is terminated, the data is freed. Each instance of the KGS-8 module represents one browse action by the admin interface.

It also provides the statistics of the Central KGS.

3.48.2 API:

Function name:

KGS8_queryEvents

Input Parameters:

source module identifier (optional)
entity identifier (optional)
CA identifier (optional)
log entry type (optional)
number of entries requested (optional)

Output Parameters:

status code

Description:

Queries the log event entries from the database specified by the input parameters. The maximum number of requested entries could be specified by the input parameter.

Function name:

KGS8_firstEvent

Input Parameters:

none

Output Parameters:

event information

Description:

Returns the information of the first event entry. The event entries must be received first by a call of the function **KGS8_queryEvents**. If no entries are received from the database or a error occurred, no information would be returned.

Function name:

KGS8_nextEvent

Input Parameters:

none

Output Parameters:

event information

Description:

Returns the information of the next event entry in the list of event entries. The event entries must be received first by a call of the function **KGS8_queryEvents** and the iteration must be initialised by a call of **KGS8_firstEvent**. If no entries are received from the database or no more entries are in the event list, no information would be returned.

Function name:

KGS8_getStatistics

Input Parameters:

source module identifier (optional)

entity identifier (optional)

CA identifier (optional)

log entry type (optional)

Output Parameters:

statistic information

Description:

Returns the statistics of the Central KGS. The statistics are generated from the information in the log database.

3.49 Module Number KGS-10: KGS API Job Scheduler

3.49.1 Functionality:

The KGS-10 module operates in a batch-oriented communication mode with the CA-17 module as the remote communication partner and controls the job entries in the database. The KGS API Job Scheduler operates a processing loop, which checks periodically for pending messages from the CA-17 module and for finished Jobs.

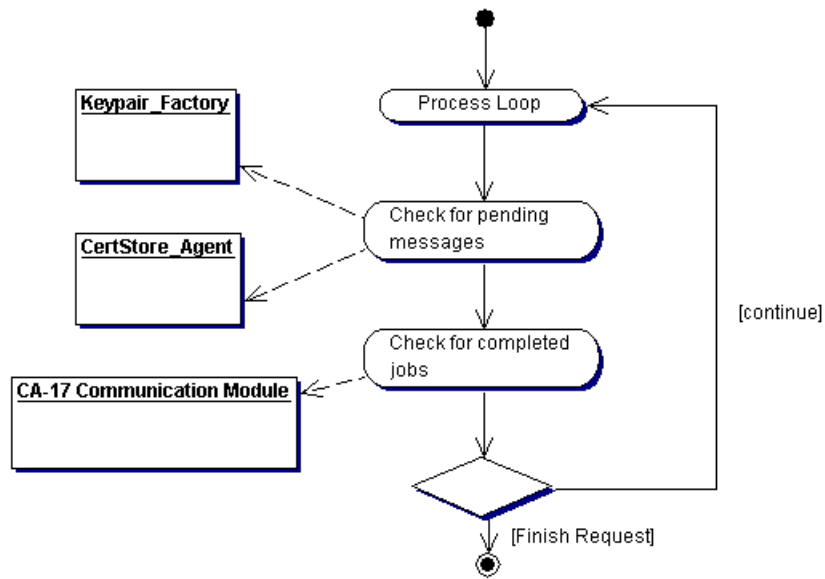


Figure 1: KGS API Job Scheduler Process Loop

If a message from the CA-17 module is received a new job entry in the database is created and the job is forwarded to the KGS-26 Key Pair Factory module or to the KGS-22 Certificate Storage.

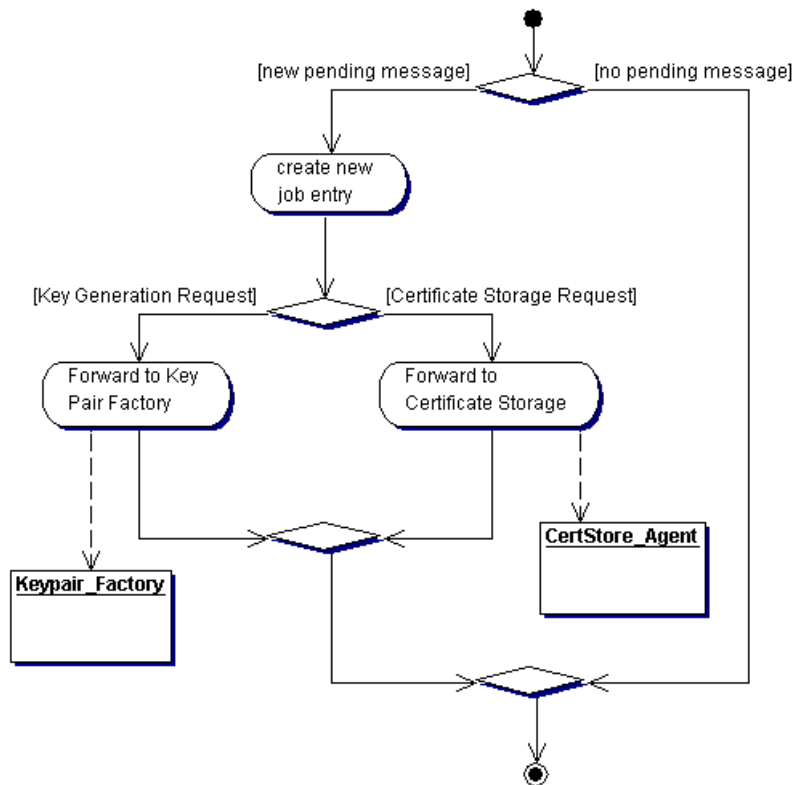


Figure 2 Check for pending message

The module also administrates the job entries in the database. If a key generation request is marked as complete, a "Key generation completed" message is send to the CA-17 module. If a certificate storage request is marked as complete in the database, a "Certificate storage completed" message is send to the CA-17 module.

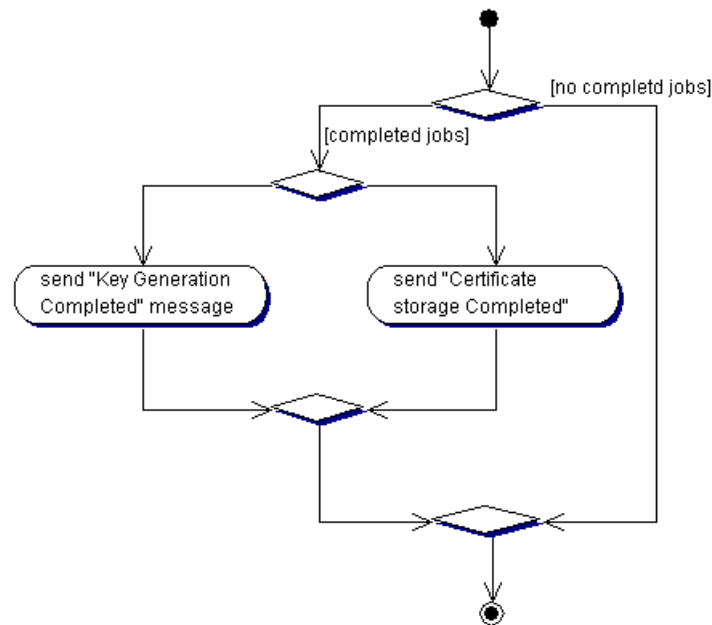


Figure 3 Check for completed jobs

3.49.2 API:

Function name:

KGS10_processLoop

Input Parameters:

none

Output Parameters:

status code

Description:

The function polls for incoming messages and checks the job database for completed jobs. If messages are received from the CA-17 module, new job entries are created in the database and the job is forwarded to the responsible module (Key Pair Factory KGS-26 and Certificate Storage KGS-22).

If the status of a job is set to completed, the function forwards the result message to the responsible module (Communication to KGS CA-17).

3.50 Module Number KGS-14: Key Store

3.50.1 *Functionality:*

The module provides a device-independent interface for key generation and data storage on a per token base. It can be used for user based key and token generation as well as for central token production. A token in the sense of this module is any storage for key pairs and certificates (hardware or software).

3.50.2 *API:*

According to PKCS #11 Cryptographic Token Interface Standard.

See <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html> for details. A open source software token implementation is available from the GNU PKCS#11 implementation <http://gpkcs11.sourceforge.net>.

The module KGS-21 Key Pair Generator is implemented by the functions **C_GenerateKey** resp. **GenerateKeyPair**. The real implementation depends on the realisation of the tokens (software or hardware).

A java wrapper for the PKCS #11 may be available for example from <http://www.mozilla.org/projects/security/pki/jss>.

3.51 Module Number KGS-15: Access Control (Central KGS)

3.51.1 *Functionality:*

This module ensures the service of authentication of the administrators. The adopted solution is the authentication of the administrators starting from the presentation of the their certificate to the admin interface.

This module ensures the service of Access Control of the administrators. The adopted solution is based on the consultation of DB Access to know the privileges of the authenticated administrators.

The access control is on a per operation base. An administrator must have the right to execute the operation.

3.51.2 *API:*

Function name:

KGS15_checkAccessRight

Input Parameters:

administrator certificate

operation id

Output Parameters:

operation granted resp. denied

Description:

The function checks the rights of the specified administrator/operator to perform the particular operation (specified by the operation id). The function returns if the operation is granted or if the execution of the operation is denied.

Function name:

KGS15_setAccessRight

Input Parameters:

administrator certificate

operation id

Output Parameters:

status code

Description:

Set the access right for the specified administrator/operator for a particular operation.

Function name:

KGS15_removeAccessRight

Input Parameters:

administrator certificate

operation id

Output Parameters:

status code

Description:

Removes the access right for the specified administrator/operator for a particular operation.

3.52 Module Number KGS-22: Certificate Storage

3.52.1 Functionality:

The module stores certificates on a per token base, if the certificates are to be delivered to the user (end entity) together with the centrally generated key pairs.

The certificate are stored together with the key pairs in the token (hardware or software). The module provides an interface to the token object, independent of the realisation as a software storage solution or a hardware device.

3.52.2 API:

According to PKCS #11 Cryptographic Token Interface Standard. The certificates could be stored as certificate objects in the cryptoki.

See <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html> for details. A open source software token implementation is available from the GNU PKCS#11 implementation <http://gpkcs11.sourceforge.net>.

3.53 Module Number KGS-23: Export Certificate Request to CA

3.53.1 Functionality:

This module package the OTP and the public key, generated by the key store module in a single message and transmit it accordingly to the CA. The module will notify the API of the success or failure of the operation.

3.53.2 API:

Function name:

KGS23_exportCSR

Input Parameters:

public key

private key

OTP

target

Output Parameters:

status code

one time URL

Description:

The function will create a certificate signing request (CSR), which includes the public key and the OTP of the user and is signed by the private key. The target is a URL (Uniform Resource Locator), where a script will receive the CSR and starts the certification process.

The one time URL will be send to the User KGS as answer to the export of the export of the certification request. The User KGS can retrieve from this URL the certificate after the enrolment was successful.

3.54 Module Number KGS-24: Import Certificate Replies from CA

3.54.1 Functionality:

The module is responsible for receiving the data like certificate and key pairs via the network (e.g. internet) or via a API (e.g. triggered via the user interface). It also delivers

3.54.2 API:

Function name:

KGS24_storeCertificate

Input Parameters:

certificate

certificate storage identifier

Output Parameters:

status code

Description:

This function stores the given certificate in the specified certification store on the users computer. This function is called from the user KGS, after receiving a certificate from the certification authority.

Function name:

KGS24_storeKeys

Input Parameters:

public key

private key

key store identifier

Output Parameters:

status code

Description:

This function stores the given key pair in the specified key store on the users computer. This function is called from the user KGS, after receiving a newly generated key pair (e.g. form the central KGS) from the certification authority.

Function name:

KGS24_receiveCertificate

Input Parameters:

source URL for certificate
authentication token
certificate storage identifier

Output Parameters:

status code

Description:

This function will request the certificate from the CA specified by the first parameter and store it in the certification store specified by the third parameter. The user must authenticate himself to the certification authority, this is done by using the one time URL and representing the authentication token to the CA. The authentication token includes the OTP.

The one time URL will be send to the User KGS as answer to the export of the certification request. The User KGS can retrieve from the on time URL the certificate after the enrolment was successful.

Function name:

KGS24_receiveKeys

Input Parameters:

source URL for key pair
authentication token
key store identifier

Output Parameters:

status code

Description:

The function requests a key pair from e.g. the Central KGS and stores it in the specified key store. The user must authenticate himself to the Central KGS, this is done by using the one time URL and representing the authentication token. The authentication token includes the OTP.

The one time URL will be send to the User KGS as answer to the key generation request. The User KGS can retrieve from this one time URL the key pair.

3.55 Module Number KGS-25: User Interface

3.55.1 Functionality:

This module is the graphical interface between the user and the other modules of the user KGS. It should guide the user automatically through the certificate enrolment

process. The main purpose of the interface is to collect the user data and send it to the RA to start the certificate enrolment.

The module will collect the all necessary data (like common name, organisation, location, country etc.) from the user and creates together with the OTP and the generated the Certificate Signing Request (CSR) which is send to the RA. This will start the certification enrolment process and the CA will issue the new user certificate.

If the enrolment process is successful, the user interface module will receive the certificate and will transfer it to the users certificate storage. Otherwise the user interface will display a error message and will tell the user the reason, why the certification enrolment failed.

A further purpose of the user interface is to renewal certificates. After a certificate has expired, the user has to request a new certificate from the RA. The user has to receive a new OTP from the RA. For this he has to send his actual data to the RA. The user interface will be read the data from the certificate and present them to the user. The user can correct them, if some information are not longer actual (like location etc.).

After he/she has received the new OTP, the user can create a new CSR and send it to the RA. The CA will create a new certificate and send it to the user.

3.56 Module Number KGS-26: Key Pair Factory

3.56.1 Functionality:

The module produces upon request by the KGS-API Job Scheduler KGS-10 key pairs. It utilises the key generation Key Pair Generator KGS-21 module for the generation of each key pair set related to one token and reports the created public key values back to KGS-10.

When the KGS-API Job Scheduler receives a Key Generation Request from the module CA-17, it creates a new job entry in the database and stores the job information there. Then it calls the function KGS26_generateKey from the Key Pair Factory module and starts the key generation process.

The Key Pair Factory module activates the Key Pair Generator module, which will create a key pair set related to a token. After completion of the key pair generation process, the Key Pair Factory module sets the current job state to *completed*.

3.56.2 API:

Function name:

KGS26_generateKey

Input Parameters:

job identifier

Output Parameters:

status code

Description:

Reads the job information from the database and starts the key pair generation process.

3.57 Module Number KGS-27: User KGS API

3.57.1 Functionality:

The module is the core of the User KGS and provides the functionality for the interface and renders public all the resources of underlying modules.

The module functionality is event based, with events coming in from the user interface. The module provides the following functionality:

- Store access
- Key generation
- Access control
- Key/OTP transmission
- Certificate import

3.57.2 API:

According to PKCS #11 Cryptographic Token Interface Standard the functions Store Access, Key generation and Access control could be realised.

Under MS Windows (NT/2000/XP) a Cryptographic Service Provider CSP implementation could be used instead. The CSP library includes all functionality necessary for the described functions.

See <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-11/index.html> for details. A open source software token implementation is available from the GNU PKCS#11 implementation <http://gpkcs11.sourceforge.net>.

For the functions Key/OTP transmission and Certificate import additional functions are necessary:

Function name:

KGS27_keyotpTransmission

Input Parameters:

public key
user OTP
target

Output Parameters:

status code

Description:

This function retrieves a previously user OTP from the user interface, uses the public key generated by PKCS #11 module and transmits the final package to the CA in order to start the certificate creation process.

The target where the data package should be send to, are specified by the target parameter. This could be a URL (Uniform Resource Locator) of a script, which retrieves the data and starts the certificate creation process.

Function name:

KGS26_importCertificate

Input Parameters:

filename (optional)

data (optional)

store identifier

Output Parameters:

status code

Description:

This function will import the certificate included in the file specified by the filename parameter or given by the parameter data. The certificate will be stored in the certificate storage specified by the parameter store identifier.

To automatically import certificates received via mail or web a application, which makes use of this function, must be associated (under MS Windows in the registry database) with the file type extension or the mime type.

3.58 Module Number KGS-28: Access control (User KGS)

3.58.1 Functionality:

This module handles the input and the storage of the password or PIN. If the user has not input his/her password or PIN during the actual session, the module will display a dialog to ask the user to input the password or PIN.

After the user has input the password or PIN it will be (encrypted before and) compared with the one stored in the database. If a smart card is used the PIN will be send to the smart card and the smart card will grant or deny the access to the data stored on it.

A further functionality is the decryption of pass phrase encrypted data.

The password or PIN will be stored inside the module for further usage.

3.58.2 API:

Function name:

KGS28_accessCtrl

Input Parameters:

none

Output Parameters:

access denied or granted

Description:

The function will ask the user to input his/her password or PIN, if the user has not already done this during the actual session. The password will be (encrypted and) compared with the one stored in the user database.

Function name:

KGS28_decrypt

Input Parameters:

encrypted data

Output Parameters:

decrypted data

status code

Description:

Decrypts pass phrase protected data.

3.59 Module Number KGS-29: Export PKCS#12***3.59.1 Functionality:***

This module will export the certificate and key pair into a PKCS #12 format. PKCS #12 is a common format for personal information including the private key of a key pair. It is accepted by Microsoft Internet Explorer and Netscape families of browser.

For security reasons the contents of the PKCS #12 structure is password protected and the password must be different from the master password.

The module must request the key pair (public and private key) and the certificate (and optional the certificate chain) from the key store. To request the private information the master password must be requested from the user.

3.59.2 API:

Function name:

KGS29_exportPKCS12

Input Parameters:

entity id
export certificate chain
filename
pkcs #12 export password

Output Parameters:

status code

Description:

The function saves the key pair and the certificate (with certificate chain) in the file specified by the input parameter *filename*. The content is for security reasons password protected. The password is specified by the input parameter *pkcs #12 export password*. The input parameter *export certificate chain* specifies if the whole certificate chain or only the user certificate should be exported.

3.60 Module Number KGS-30: KGS database

3.60.1 Functionality:

The KGS database module offers an interface for the access of the KGS database and the there stored entries.

A possible implementation of this module could be based for example on a JDBC 3.0 API implementation.

3.60.2 API:

Function name:

KGS30_connect

Input Parameters:

database server name or ip address
database server port
database name
database user
database password

Output Parameters:

connection

Function name:

KGS30_disconnect

Input Parameters:

connection

Output Parameters:

status code

Function name:

KGS30_commit

Input Parameters:

connection

Output Parameters:

status code

Function name:

KGS30_rollback

Input Parameters:

connection

Output Parameters:

status code

Function name:

KGS30_select

Input Parameters:

connection

table name

field names

conditions

Output Parameters:

result set

Function name:

KGS30_insert

Input Parameters:

connection

table name

field names

field values

Output Parameters:

status code

Function name:

KGS30_update

Input Parameters:

connection

table name

field names

field values

conditions

Output Parameters:

status code

Function name:

KGS30_delete

Input Parameters:

connection

table name

conditions

Output Parameters:

status code

Function name:

KGS30_exec

Input Parameters:

connection

SQL statement

Output Parameters:

status code

Function name:

KGS30_call

Input Parameters:

connection

procedure name

parameter values

Output Parameters:

status code

4 Appendix

4.1 Terms and Definitions, Glossary

EUPKI EUPKI, the libre software Public Key Infrastructure (project name)

WP4 Work Package 4 = Specifications

WP3 Work Package 3 = Requirements

4.2 References

The words "should", "may" etc. are used in this document according to: Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP14, RFC 2119, March 1997.

This document refers to the following external documents:

Reference	Document
D3.6	Perimeter and requirements of the project
[RFC 2459]	Internet X.509 Public Key Infrastructure Certificate and CRL Profile (RFC 2459) www.ietf.org/rfc/rfc2459.txt
[RFC 2510]	Internet X.509 Public Key Infrastructure Certificate Management Protocols (RFC 2510) www.ietf.org/rfc/rfc2510.txt
[RFC 2511]	Internet X.509 Certificate Request Message Format (RFC 2511) www.ietf.org/rfc/rfc2511.txt
[RFC 2559]	Internet X.509 Public Key Infrastructure Operational Protocols - LDAPv2 (RFC 2559) www.ietf.org/rfc/rfc2559.txt
[RFC 2587]	Internet X.509 Public Key Infrastructure LDAPv2 Schema (RFC 2587) www.ietf.org/rfc/rfc2587.txt
[RFC 2875]	Diffie-Hellman Proof-of-Possession Algorithms (RFC 2875) www.ietf.org/rfc/rfc2875.txt
[RFC 3279]	Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRI Profile (RFC 3279) www.ietf.org/rfc/rfc3279.txt
[RFC 3280]	Internet X.509 Public Key Infrastructure Certificate and CRL Profile (RFC 3280) www.ietf.org/rfc/rfc3280.txt