	IST-2001-34340 D 4.1 : Architectural Spec
Distribution List:	Project Partners
Author:	WP4 Members
Authorised by:	---
Date of Issue:	07.10.2002
Issue:	Version 1.1
File name:	EUPKI-WP4-D4.1-1.1.doc
Work package:	WP4 Specifications
Deliverable Number:	4.1
Deliverable Type:	Public
Deliverable Nature:	Specification
Total Number of Pages:	90
Contact Details for EUPKI:	christoph.schiller@gdm.de

Copyright © The EU-PKI Consortium

0 Document Control

Issue	*Date of Issue*	*Comments*
0.10	19 July 2002	First version
0.20	22 July 2002	Consistent Module/Process view, Module List finished
0.21	23 July 2002	Missing central KGS slide reinserted, KGS process diagrams completed
0.22	23 July 2002	New number for Audit module (RA-8) to resolve name collision with revocation
0.32	26 July 2002	Some Central KGS modules completed
0.33	29 July 2002	Included CGEY input on RA modules, Cleanup round on document
0.4	2 Aug 2002	Explained block diagrams with respect to input of EADS Telecom, Cleanup of TOC, Incorporated Intesa comments, Common security section with reference to AQL input
0.41	5 Aug 2002	Incorporated updated RA module descriptions of CGEY, final cleanup of RA module section
0.42	5 Aug 2002	Incorporated user KGS descriptions, applicable standards and various remarks of Axetel
0.43	5 Aug 2002	Incorporated CA module description of CGEY
0.7	5 Aug 2002	Distributed for review meeting
0.8	6 Aug 2002	Added CSP modules, Data model, Distributed for review meeting
0.83	20 Aug 2002	several changes after review meeting
0.84	20 Aug 2002	section 2.11 "Scalability" inserted by GIP-MDS
0.85	22 Aug 2002	several changes (especially KGS) based on input from GIP-MDS and Axetel
0.86	27 Aug 2002	Consistency Checks by CertiNomis
0.9	11 Sep 2002	incorporated review comments from CertiNomis and ON-x. Added scenario descriptions. Document finalized for steering board presentation due for Sep 28th 2002.
0.91	13 Sep 2002	scalability chapter
1.0	17 Sep 2002	Final version released for TB review
1.01	4 Oct 2002	Key Ceremony added by CertiNomis
1.1	7 Oct 2002	Minor changes

0.1 Abstract

This document D4.1 is the first of three deliverables of the specification phase of the EU- PKI project. For clarity reasons security issues are treated in a separate document (D4.1a) which is referenced by this document and belongs to the same deliverable.

D4.1 first provides a general view on the architecture, processes and data models of the EU-PKI system, dealing with the Registration Authority (RA) consisting of front office (FO) and back office (BO) as well as the Certification Authority (CA) and the Key Generation System (KGS). Based on work flow and data flow between these components short module descriptions are provided that deal with the functionality, input and output of the modules identified.

D4.1 is the basis for D4.2 and D4.3 which will contain the detailed specifications of module functionality and interfaces.

0.2 Table of Contents

0 DOCUMENT CONTROL 2
0.1 ABSTRACT 3
0.2 TABLE OF CONTENTS 4
1 INTRODUCTION TO *EU-PKI* 7
1.1 WORKFLOW MODEL A: "BANK" SCENARIO 7
1.2 WORKFLOW MODEL B: "ONLINE" SCENARIO 8
1.3 WORKFLOW MODEL C: "MOBILE" SCENARIO 9
2 BIRD'S VIEW OF GLOBAL ARCHITECTURE 11
2.1 THE THREE MAIN MODULE BLOCKS IN EU-PKI 11
2.2 NOT PART OF THE EU-PKI 11
2.3 GLOBAL INTERACTION BETWEEN MODULES 12
2.4 DETAILED MODULE VIEW 16
2.5 MODULE TABLE 23
2.6 FUNCTION TABLE 25
2.7 PROCESS MODEL 27
2.8 DATA MODEL 36
2.9 COMMON TECHNICAL STANDARDS & PROTOCOLS 41
2.10 COMMON SECURITY REQUIREMENTS 45
2.11 SCALABILITY 46
2.12 CONCEPT FOR ADMINISTRATION AND CONFIGURATION 48
3 MODULE DESCRIPTIONS 50
3.1 MODULE NUMBER FO-1 / RA-1: USER INTERFACE 50
3.2 MODULE NUMBER FO-2: COMMUNICATION WITH BO 50
3.3 MODULE NUMBER RA-2: VALIDATION 51
3.4 MODULE NUMBER RA-3: LOG 52
3.5 MODULE NUMBER RA-4: RA DATABASE 52
3.6 MODULE NUMBER RA-5: ENTRY / RENEWAL 53
3.7 MODULE NUMBER RA-7: COMMUNICATION TO CA 54
3.8 MODULE NUMBER RA-8: AUDIT 54
3.9 MODULE NUMBER RA-10: RA-API 55
3.10 MODULE NUMBER RA-13: REVOCATION 56
3.11 MODULE NUMBER RA-14: ENTITY/CERT BROWSER 56
3.12 MODULE NUMBER RA-15: ACCESS CONTROL 57
3.13 MODULE NUMBER RA-16: ADMIN INTERFACE 58
3.14 MODULE NUMBER RA-17: ADMIN-API 58
3.15 MODULE NUMBER RA-18: RA PK ACCEPTOR 59

3.16 MODULE NUMBER CA-1: USER REQUEST HANDLER 60
3.17 MODULE NUMBER CA-2: WRITEMAIL 60
3.18 MODULE NUMBER CA-3: LOG 61
3.19 MODULE NUMBER CA-5: ADMIN INTERFACE 61
3.20 MODULE NUMBER CA-6: ADMIN-API 62
3.21 MODULE NUMBER CA-8: AUDIT 63
3.22 MODULE NUMBER CA-9: OTP DISTRIBUTOR 63
3.23 MODULE NUMBER CA-10: CA-API 64
3.24 MODULE NUMBER CA-11: OTP GENERATOR 65
3.25 MODULE NUMBER CA-12: OTP AUTHENTICATOR 65
3.26 MODULE NUMBER CA-13: COMMUNICATION TO CSP 66
3.27 MODULE NUMBER CA-14: CA DATABASE 66
3.28 MODULE NUMBER CA-15: ACCESS CONTROL 67
3.29 MODULE NUMBER CA-16: PUBLICATION 68
3.30 MODULE NUMBER CA-17: COMMUNICATION TO KGS 68
3.31 MODULE NUMBER CA-18: CRL FACTORY 69
3.32 MODULE NUMBER CA-20: CERTIFICATE UPDATE AGENT 69
3.33 MODULE NUMBER CSP-1: CSP-API 70
3.34 MODULE NUMBER CSP-2: CERTIFICATE SIGNER 70
3.35 MODULE NUMBER CSP-3: CRL SIGNER 71
3.36 MODULE NUMBER KC-2: CERTIFICATE FACTORY 71
3.37 MODULE NUMBER KC-5: ADMIN INTERFACE 72
3.38 MODULE NUMBER KC-6: ADMIN API 72
3.39 MODULE NUMBER KC-15: ACCESS CONTROL 73
3.40 MODULE NUMBER KC-26: KEY PAIR FACTORY 74
3.41 MODULE NUMBER KC-30: SECRET EXPORT 74
3.42 MODULE NUMBER KC-31: SECRET PRINTING 75
3.43 MODULE NUMBER KC-32 SECRET IMPORT 75
3.44 MODULE NUMBER KGS-3: LOG 76
3.45 MODULE NUMBER KGS-5: ADMIN INTERFACE 76
3.46 MODULE NUMBER KGS-6: ADMIN-API 77
3.47 MODULE NUMBER KGS-8: AUDIT 78
3.48 MODULE NUMBER KGS-10: KGS API JOB SCHEDULER 78
3.49 MODULE NUMBER KGS-14: KEY STORE 80
3.50 MODULE NUMBER KGS-15: ACCESS CONTROL (CENTRAL KGS) 81
3.51 MODULE NUMBER KGS-22: CERTIFICATE STORAGE 82
3.52 MODULE NUMBER KGS-23: EXPORT CERTIFICATE REQUEST TO CA 82
3.53 MODULE NUMBER KGS-24: IMPORT CERTIFICATE REPLIES FROM CA 83
3.54 MODULE NUMBER KGS-25: USER INTERFACE 84

3.55 MODULE NUMBER KGS-26: KEY PAIR FACTORY 84
3.56 MODULE NUMBER KGS-27: USER KGS API 86
3.57 MODULE NUMBER KGS-28: ACCESS CONTROL (USER KGS) 87
3.58 MODULE NUMBER KGS-29: EXPORT PKCS#12 88
3.59 FUTURE ASPECTS (ENHANCEMENTS, EXTENSIONS, CAPACITY RESERVES) 89

4 APPENDIX 90

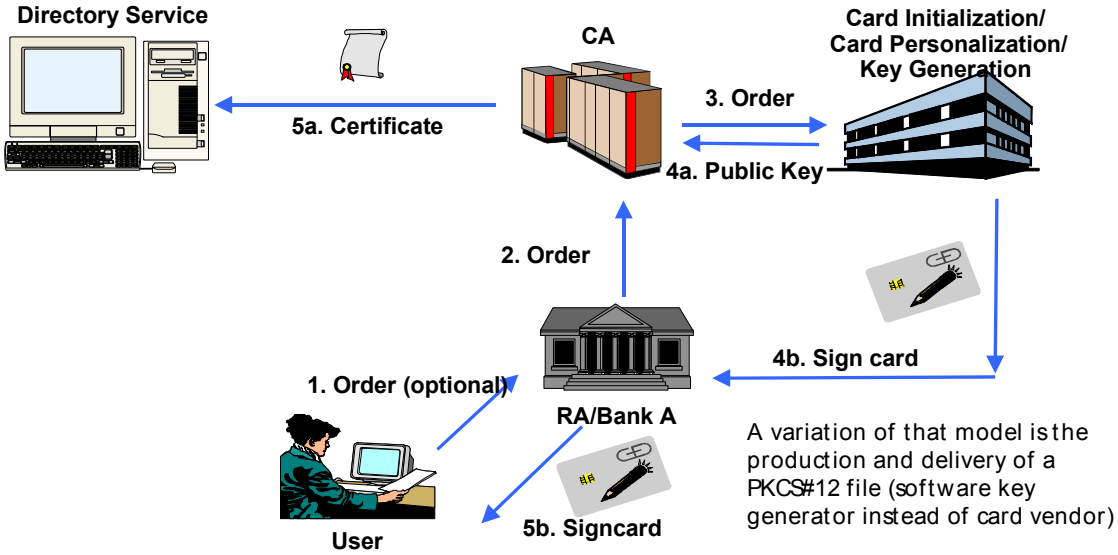
4.1 TERMS AND DEFINITIONS, GLOSSARY 90
4.2 REFERENCES 90

1 Introduction to *EU-PKI*

This chapter lists three typical scenarios of PKI implementations. Each scenario shows a different way to enroll (register) certificate holders and create and distribute keys and certificates. The scenarios vary by type of token, sequence of steps and location of key generation. They cover most of the alternatives in each category as has been listed in D3.7.

They have been a source of further considerations and were placed in this document for illustration purpose only. So these scenarios do not expand or replace the requirements from the deliverables of WP3.

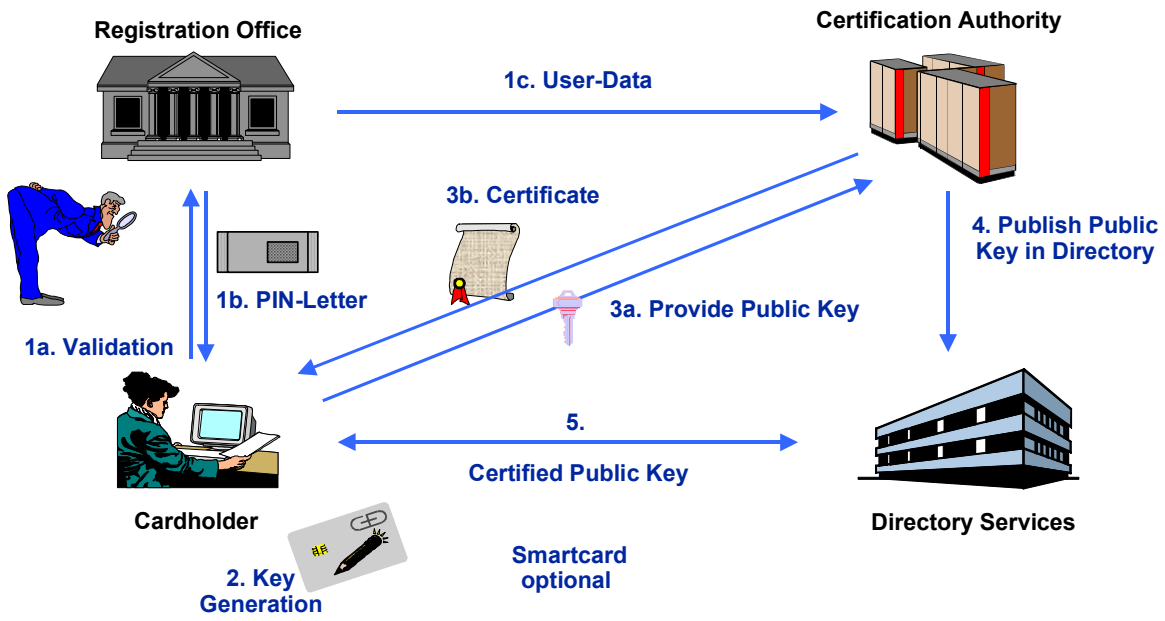
1.1 Workflow Model A: "Bank" Scenario



A banking card scenario is an attempt to combine ordinary and existing account creation procedures typically found in banks with the issuance of a smartcard containing certified key pairs. A user registers at a bank (creates an account) and applies for a smartcard. The data is forwarded to the CA operated by the bank which orders a smartcard from the card vendor. The card vendor does produce the card and the keys and delivers the public key to the CA, which certifies the key and distribute the certificate to the directory, and the card to the RA for delivery to the user.

If the certificate is to be stored on the card as well, two options are currently used: either the card production and key generation phase is done in advance, the CA has a list of public keys already available at the card vendor and delivers the certificate with the final card order to the card vendor, or the certificate is downloaded to the card at the RA or by the user.

1.2 Workflow Model B: "Online" Scenario



In this scenario the key generation happens at the user. The sequence of steps is:

1. Registration: the user applies for a card, delivers his information which gets verified and forwarded to the CA. The user gets an empty card and an initial one-time password in a security envelope from the RA.

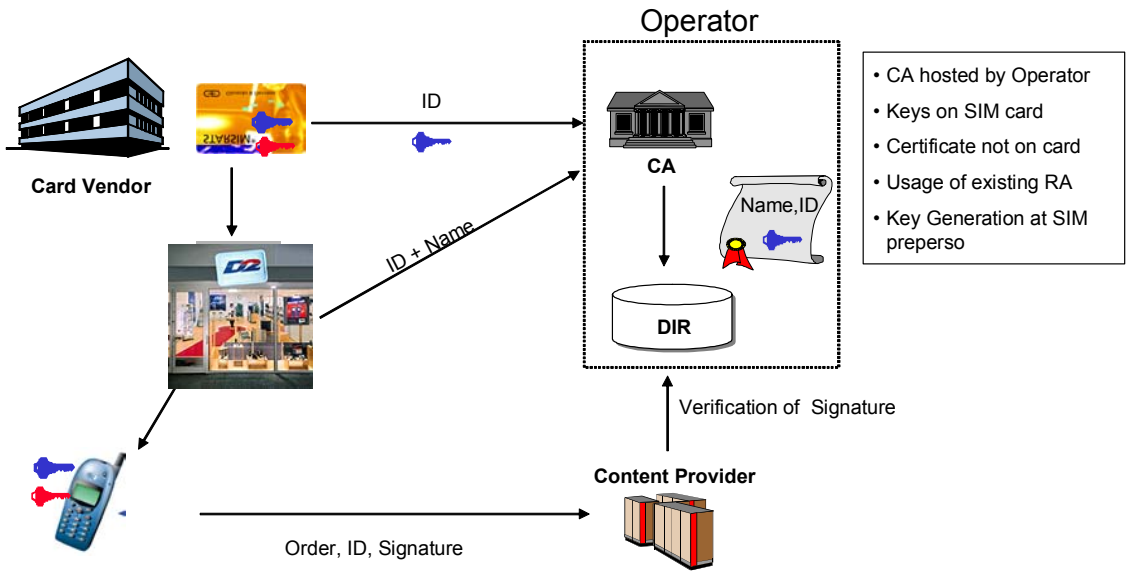
2. Key generation: the user generates keys and stores them on the card

3. Certification request: the user sends the public key along with the one-time password online to the CA. The password establishes a relationship between the key and the user information delivered by the RA to the CA. A certificate with this information is created and returned to the user and published in a directory.

This scenario is capable to support any type of key storage media, including keys stored in files (soft keys). It is the scenario typically deployed with internet applications like browsers and e-mail clients and is supported by their built-in functionality. It is the basic model for the PKI standardisation activities conducted by

the IETF and its PKIX working group. The initial one-time password is called the "IAK" in the PKIX model.

1.3 Workflow Model C: "Mobile" Scenario



The mobile scenario is close to the banking scenario. Again it is an attempt to combine ordinary user enrolment procedures, this time those found at mobile network operators, with delivery of keys and certificates to enable PKI-functionality. Key generation is done centrally during card production. The difference between this scenario and the banking scenario is the change in the sequence of steps "card distribution" and "certificate production". Whereas in the banking scenario the intended card holder is known during "card production", this is not true for the mobile scenario where cards are assigned to cardholders at the card distribution points. Thus "certificate production" has to be delayed until a card is given to a cardholder.

In this scenario a CA, probably a mobile phone operator, orders a batch of cards (SIMs) from a card vendor. The cards are equipped with key pairs by the card vendor and delivered to card distribution points (operator outlets where mobile phone contracts are sold). The CA receives information on cards and public keys delivered to the distribution points.

Once a new user shows up at a distribution point, he/she is assigned a card. The CA gets informed on the personal information of the cardholder and the card he/she did

receive, produces a certificate for the now assigned key(s) and publish them in its directory.

In contrary to other scenarios the certificate is not distributed to the cardholder. If the cardholder needs to see his own certificate he/she has to perform a directory lookup.

2 Bird's View of Global Architecture

2.1 The Three Main Module Blocks in EU-PKI

The EU-PKI system is modularised to support different deployment scenarios, which are mainly differentiated by the way keys are generated and obtained by end entities. In a first iteration the EU-PKI system is divided into three main building blocks:

CA Certification Authority

RA Registration Authority (Front Office and Back Office)

KGS Key Generation System

The **RA block** is the part of the EU-PKI system that deals with entity data. The task of this block is to maintain a database on entities with information records for each entity consolidated and verified according to the PKI policy and to order PKI-operations like certificate production, key production, or revocation for an entity. Entity data managed by the RA block is restricted to entity-supplied person-related information like name, organization, nationality etc. It does not associate PKI-specific information to the entity data – the relationship between entities, keys and certificates is managed by the CA block.

The **CA block** is responsible for managing the relationship between public key values and entities by issuing certificates and controlling their lifecycle. The entity information to be certified is obtained from the RA block, the key values from the KGS. The CA block publishes certificates and status information on certificates through an external supplied directory where relying parties do have access to.

A **CSP block** within the CA block isolates the technology used for keys and certificates from the management issues for certificates. Cryptographic functionality and certificate formatting is implemented and executed within the CSP.

A **KC block** within the CA block to initiate the cryptographic functionality. The Key Ceremony is used to create a CA key-pair and the certificate of the administrator.

The **KGS block** is responsible for providing public key values for entities to the CA block. Key generation is supported in two flavors: entity based key generation or central key generation. If the entity produces the key pairs on its own, the public key is delivered through the entity or by the help of an RA agent to the CA, otherwise the CA orders key pair generation and distribution at a central KGS.

2.2 Not Part of the EU-PKI

KA	Key Archive	BUT: Interface to it
OCSP	Web based Certificate testing	BUT: CRL (Certificate revocation list)
DIR	Directory	BUT: Interface to it
Delivery Agent for token delivery		BUT: might be the RA
Billing		

2.3 Global Interaction between modules

The basic process flow between the main building blocks is illustrated in **Erreur ! Source du renvoi introuvable.**. New entities are registered, existing entities apply for updates to their information records, and revocations are requested at the RA. The RA delivers corresponding action requests and entity information to the CA where they get executed, with support by the KGS for key generation and distribution. The figure shows the certificate and key generation process. For more detailed process flows see chapter 2.7.

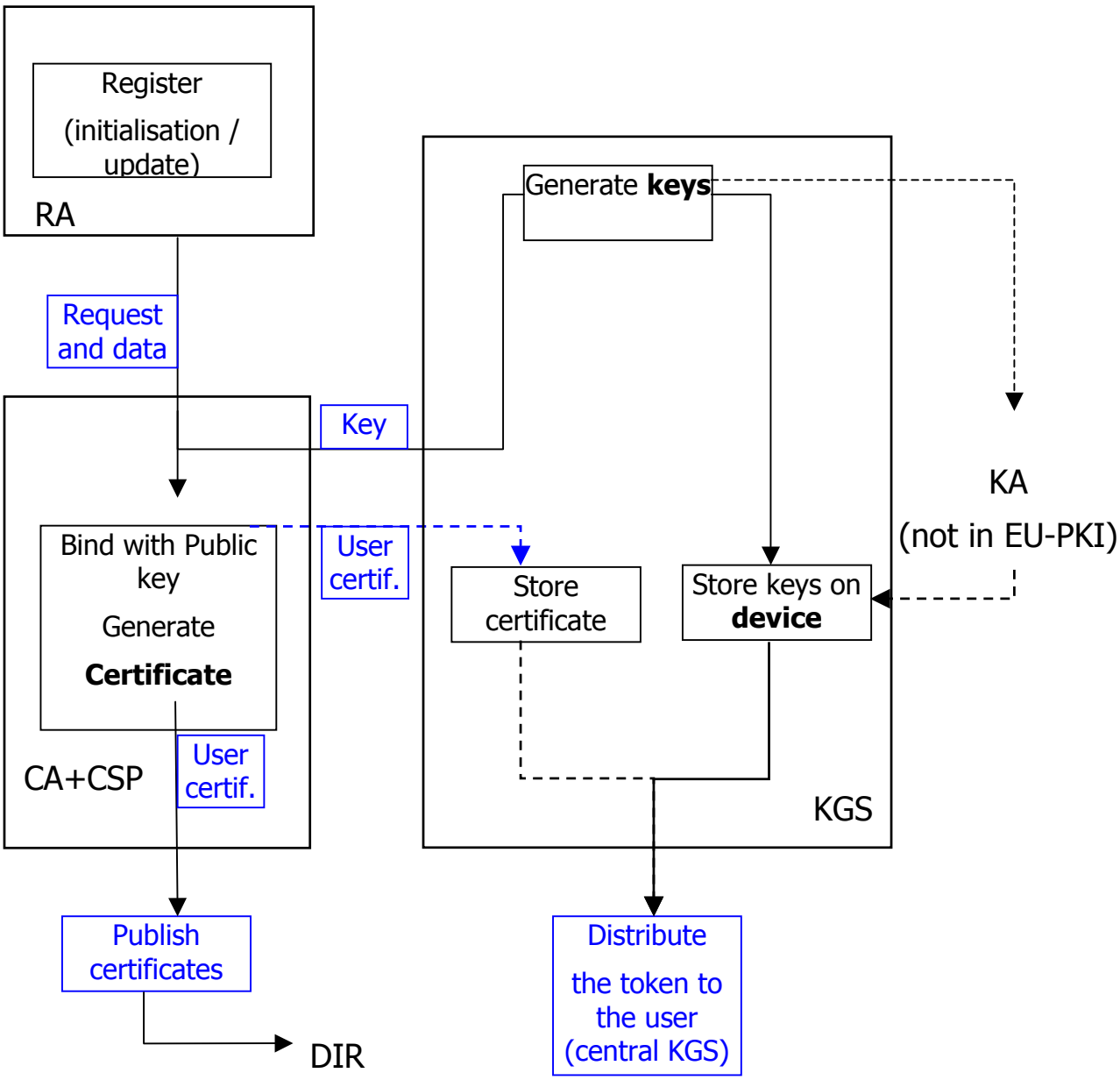


Figure 1: Global building blocks

The main blocks are modularised further. The modules and their basic connections are shown in the next few diagrams.

The block diagrams describe the EU-PKI system in terms of functional blocks, that are composed of modules and storage elements. Some modules are using common

functionality that is isolated as sub modules and probably realised as software libraries (see figure legend).

A module is defined by its functionality, and it's in- and output. Storage elements are treated as passive, with the active access control element as a separate connected module. Storage elements are meant to be logical databases that can be replicated or divided across locations as long as the access module is able to hide these variations from other accessing modules.

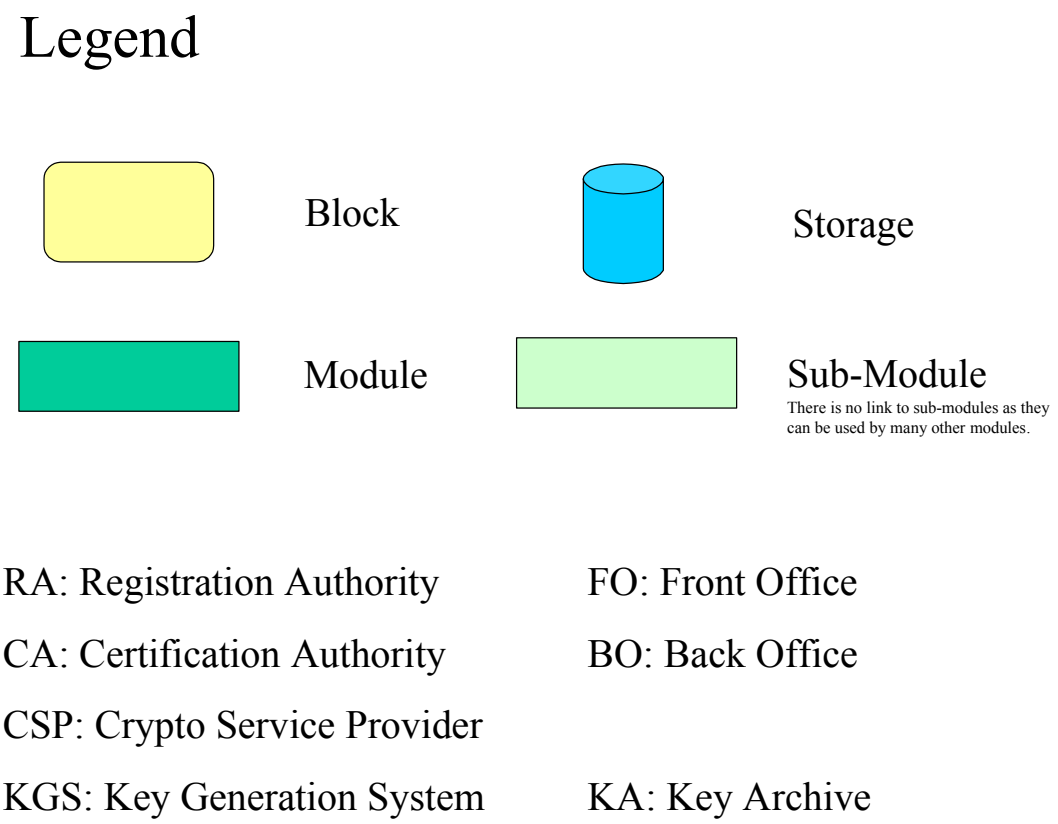


Figure 2 : Legend for block diagrams

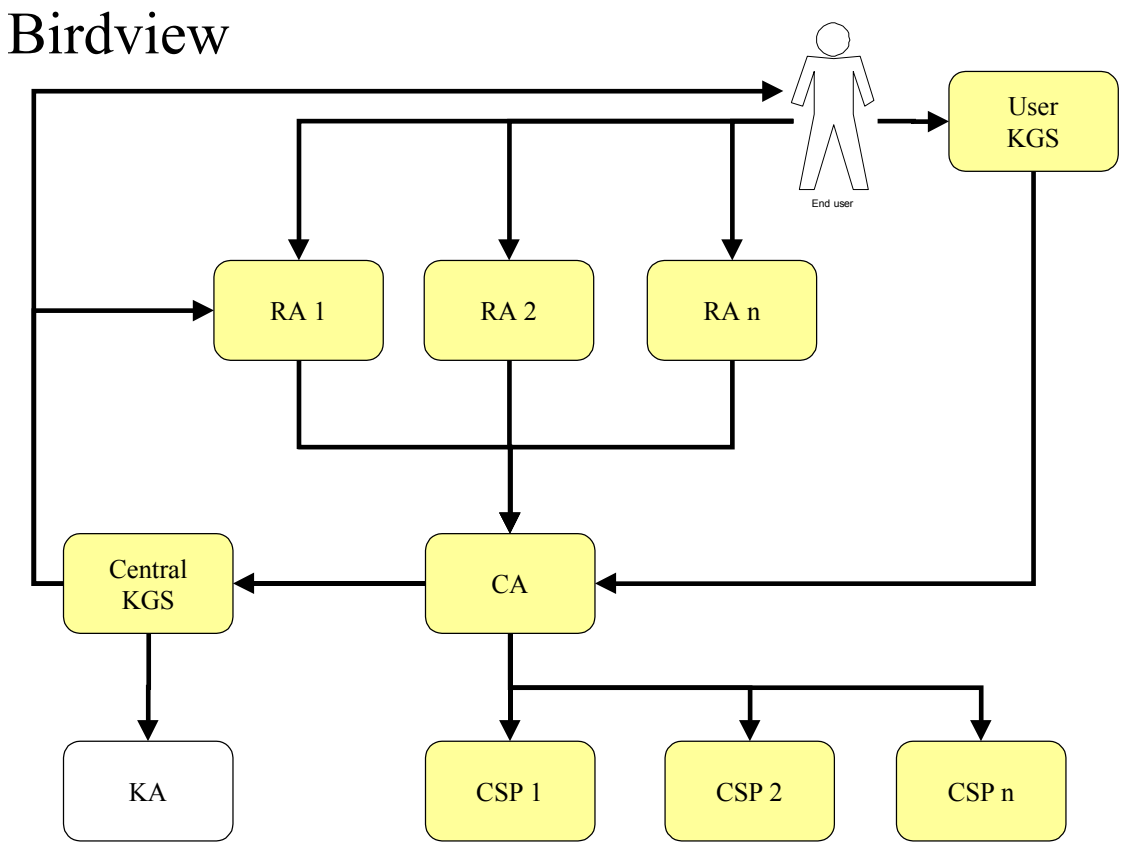


Figure 3: Bird's view on the EU-PKI system

The main view on the structure of the EU-PKI system is given in Figure 3. The EU-PKI system is able to operate a number of different Certifying Authorities. Each certifying authority has an associated CSP where the private signing key of this authority is operated, and certificates and revocation lists are created. The common management of several CSPs is controlled through the CA block. Several RA blocks may be connected to the EUPKI-system with each defining a distinct entity-space (namespace) and registration and certification policy. In its full sized version, EU-PKI supports the operation of several distinct PKIs for different purposes. The KGS is connected to the rest of the EU-PKI system either as a central key generation facility or as a piece of software operated by the end-entity. An end-entity-KGS is interacting either directly with the CA (if the online scenario is used) or indirectly via the RA. The KA (key archive) may be connected to the Central KGS to support encryption key recovery. The KA archive module itself and the key recovery procedure is not within the scope of the EU-PKI system.

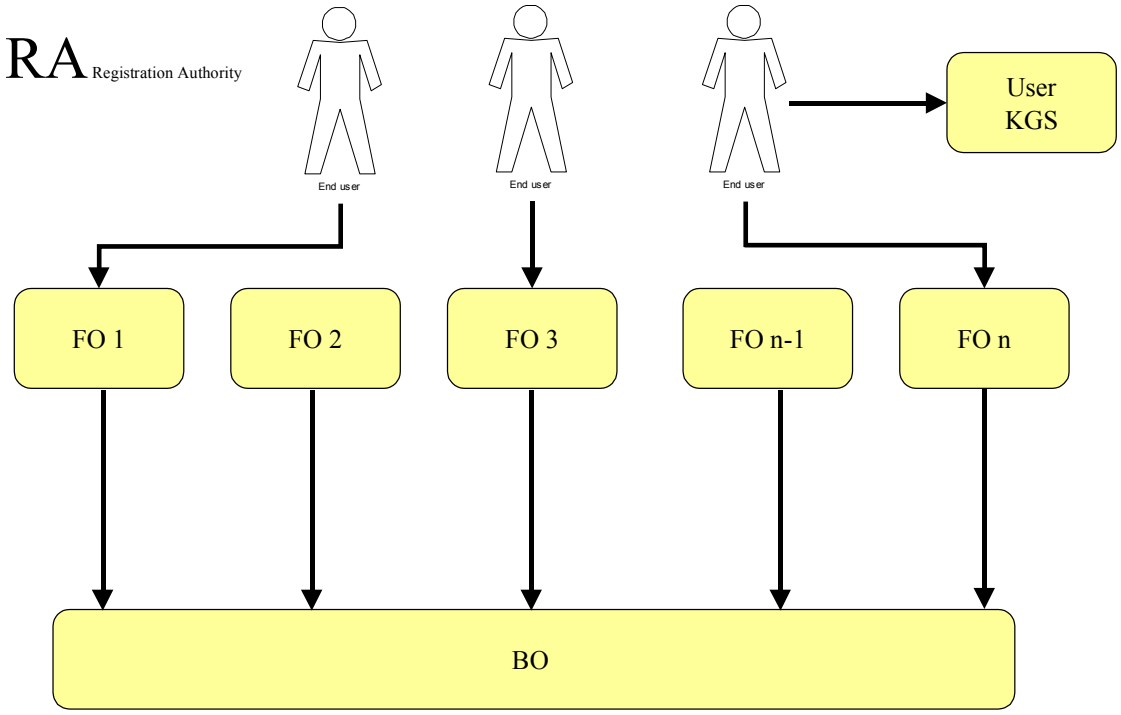


Figure 4: RA organization

The registration procedures are performed through distributed front offices which deliver entity data into back-office systems. Most of the operation of the front office is organisational, with a Registration front-end software that enables registration activities to delivered into the back-office system. The back office system may be operated in a distributed manner; however, the distribution must be hidden from a registration officer through database replication or other techniques.

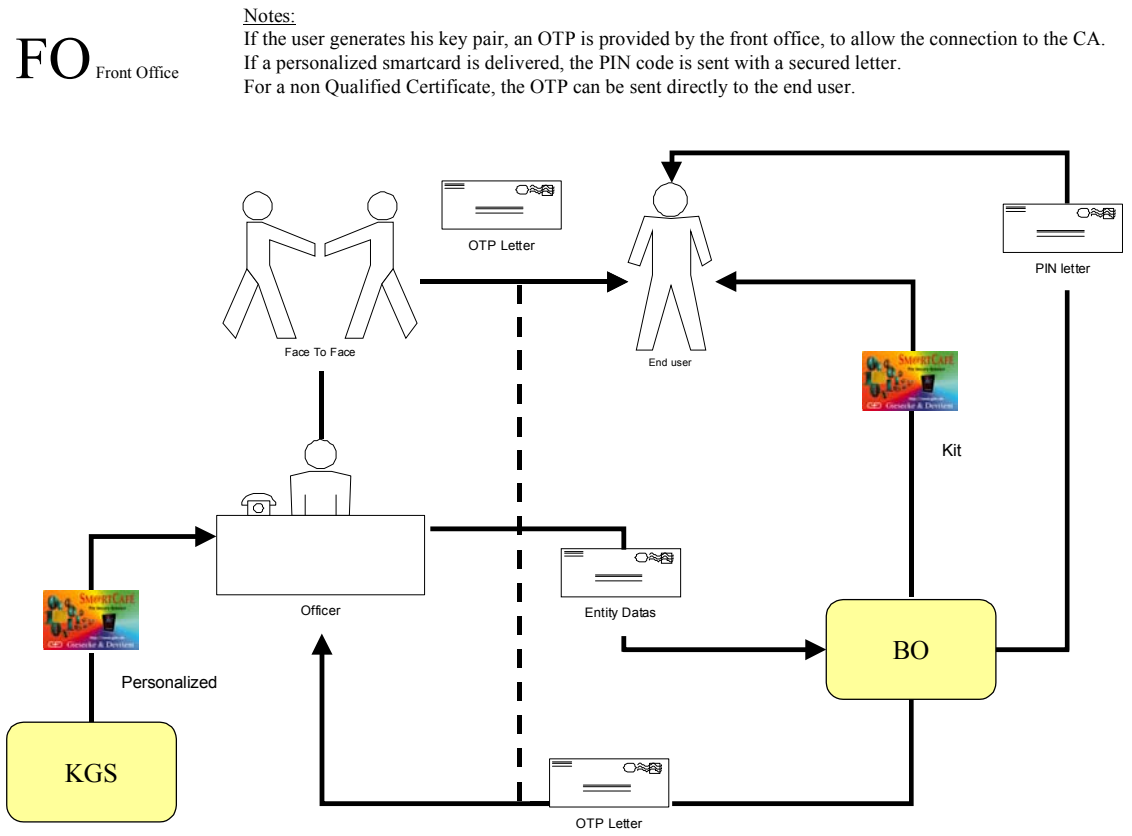


Figure 5: Front office operation

Some possibilities to organise the front office operation are presented in Figure 5. The basic step performed in each case is the acquisition of entity data by the RA officer. Validation of the acquired data is performed here according to the certificate policies in operation for the PKI. Depending on the way keys are generated and supplied to end entities several options are then possible to proceed, after the data registration and verification step. To support user based key generation the front office may deliver a one time password to the user to enable him to acquire a certificate through an online interaction at the CA, or a key pair is produced and delivered secured by a PIN. The interaction between the front office and the EU-PKI system is handled through the BO-system.

2.4 Detailed Module View

The main IT parts of the EU-PKI system are the FO-system, the BO-system, the CA-system, the CSP-system and both User- and central KGS. Their structure is given in the following figures.

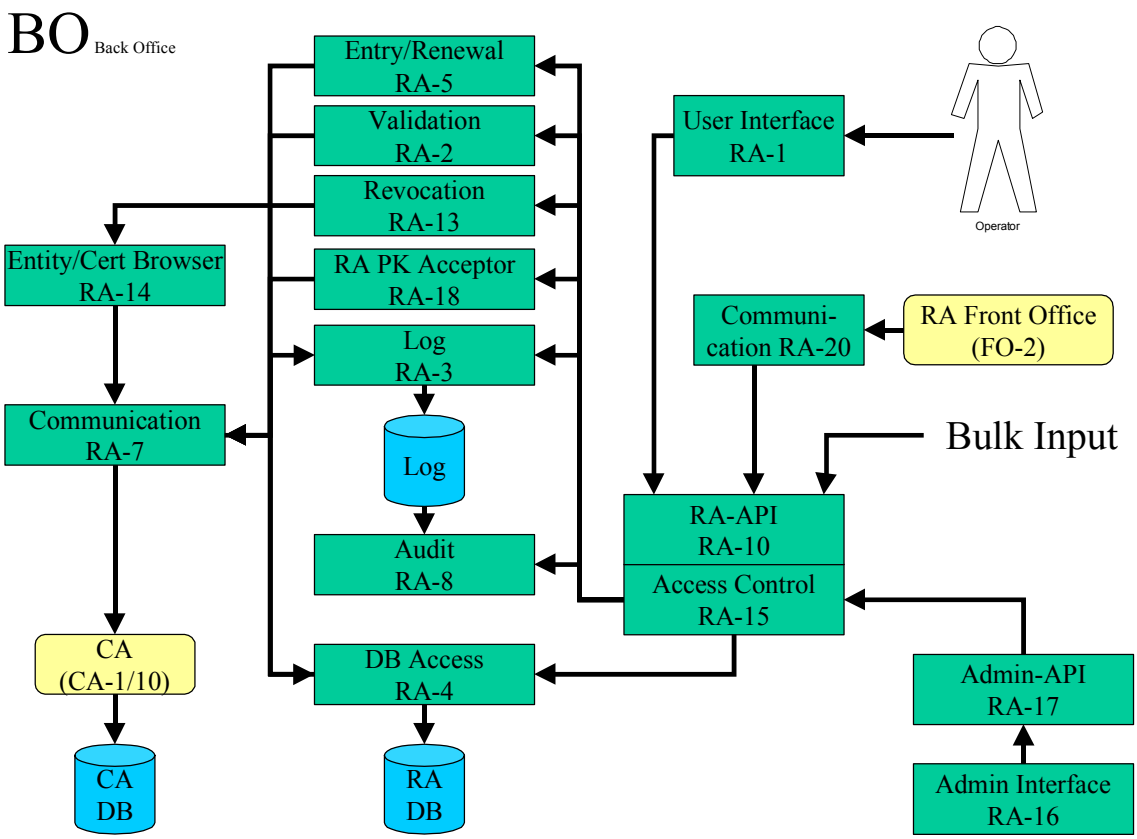


Figure 6: BO structure

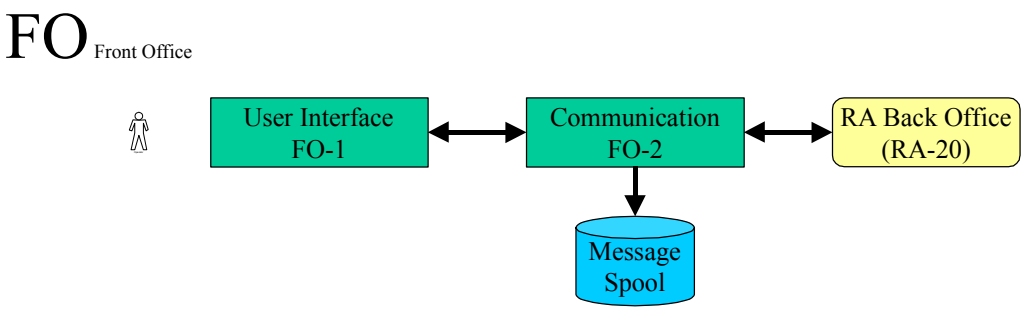


Figure 7: FO structure

The FO-system offers a user interface for controlling the BO-system from several locations. Communication of commands and results of actions executed by the BO-system are performed through a messaging system (FO-2) to support temporarily offline-conditions.

Within the BO-system (Figure 6) the entity related functionality's "Entry" (new registration) "Renewal" (information update) and "Revocation" are controlled by the RA officer through a – possibly remote – user interface. Entity information is located in the RA DB. The main data entry and control point of the BO system is the RA-API that offers all functions of the BO-system as a server. It can be used to insert bulk registration data obtained from external data sources. For obtaining information on issued certificates (e.g. to select the certificate for revocation) an "Entity/Cert Browser" module is used that queries the RA DB and the CA DB through the RA/CA communication module.

Once the needed entity information is acquired and confirmed, optionally through a central validation module for final decision on approval or rejection of a desired action, the BO-system issues a request to the CA-system through the communication module. The communication may be performed online or through batches of requests delivered by data media.

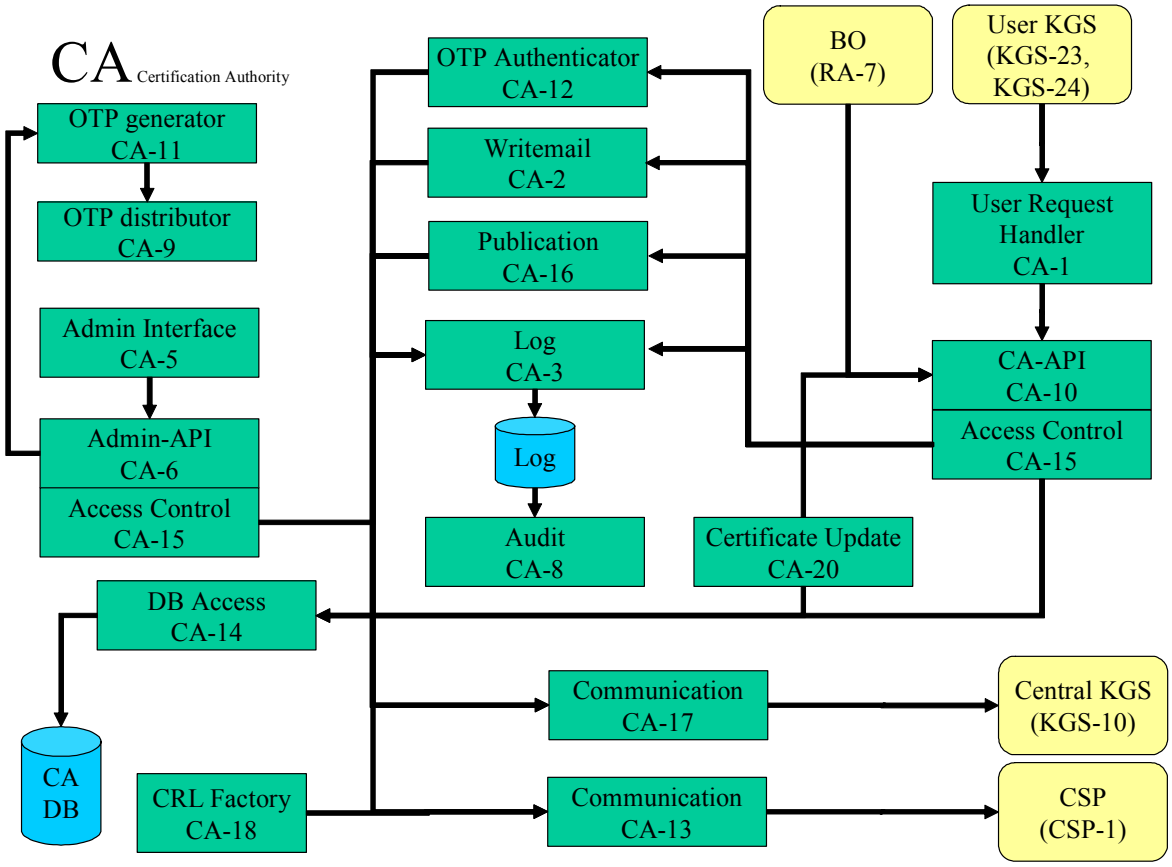


Figure 8: CA structure

The CA-system controls the common functionality of the certifying authorities which are themselves isolated into CSP-systems. The CA-system acquires public keys either from the entity or from a central key generation facility, obtains certificates and CRLs from the CSP and distributes the information towards distribution points, mostly

directories. The CA-system maintains a CA DB to store certificates, their status and related entity information (with a reference to the entry in the RA). Communication between CA and CSPs is online.

A subsystem of the CA-system is the OTP-system that is described more detailed in chapter 2.7. The purpose of the OTP-system is to establish an initial authentication key as a shared secret between the CA and an entity to support direct online certification requests by registered entities through the user request handler module.

Otherwise the operation of the CA-system is controlled through the CA-API module by received requests in batches or online from the BO-system. The CA-system may control a central KGS to obtain key pairs for fulfilling certification requests for entities. This control is done either online or by exchanging batches of requests and responses.

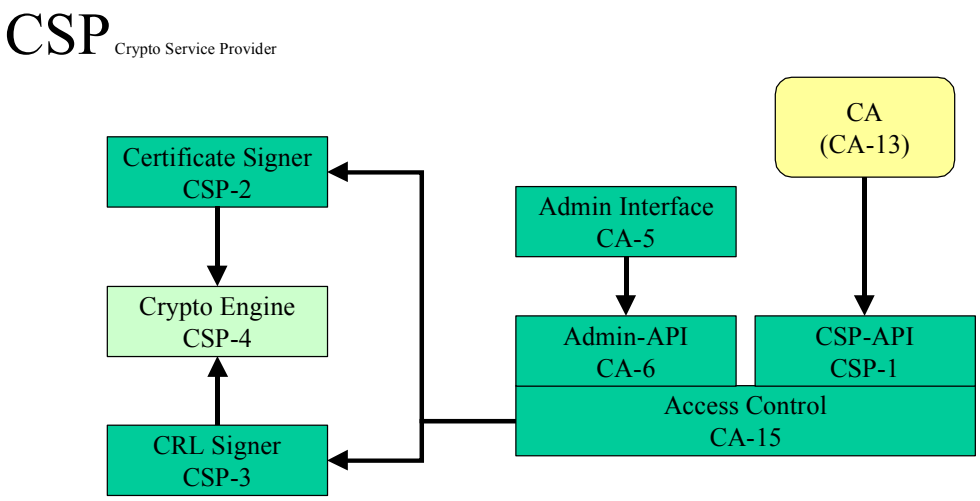


Figure 9: CSP structure

The CSP system is the set of modules that is producing certificates and CRLs using a CA key in a cryptographic engine. It offers its services through the CSP-API module. The crypto engine may be shared by several CSPs and host several keys for different certifying authorities. The CSP system reports its computed results (certificates and CRLs) to the CA system for database storage and distribution.

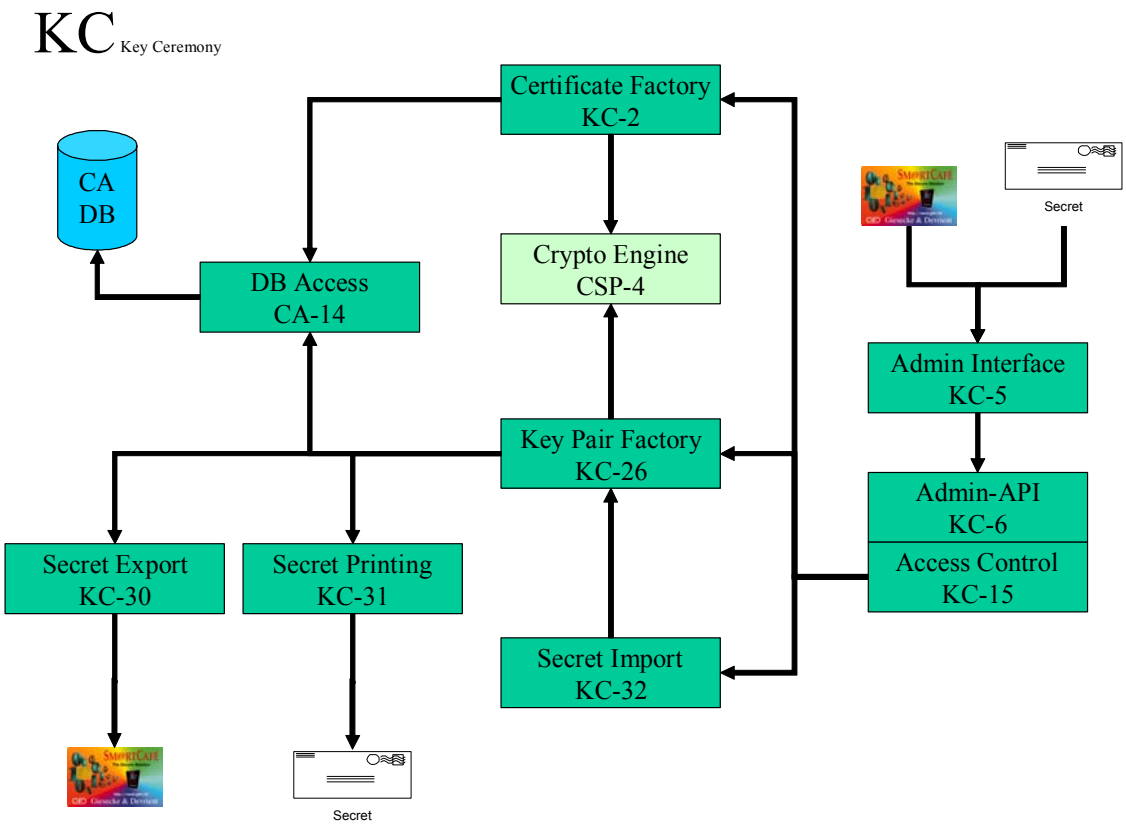


Figure 10: Key Ceremony

The KC system is the set of modules that is initiating the CA certificates using a cryptographic engine. It offers its services through the Admin Interface.

There are three modes:

- Generating the root CA with its administrator
- Generating a sub CA
- Changing the administrator

In order to generate a CA key pair, we will not use a random generator but some secrets entered via the Admin interface. Theses secrets can be printed on several secure letters and/or stored into smartcards.

To generate a sub CA, the administrator certificate must be used.

To change the administrator certificate, the secrets must be used.

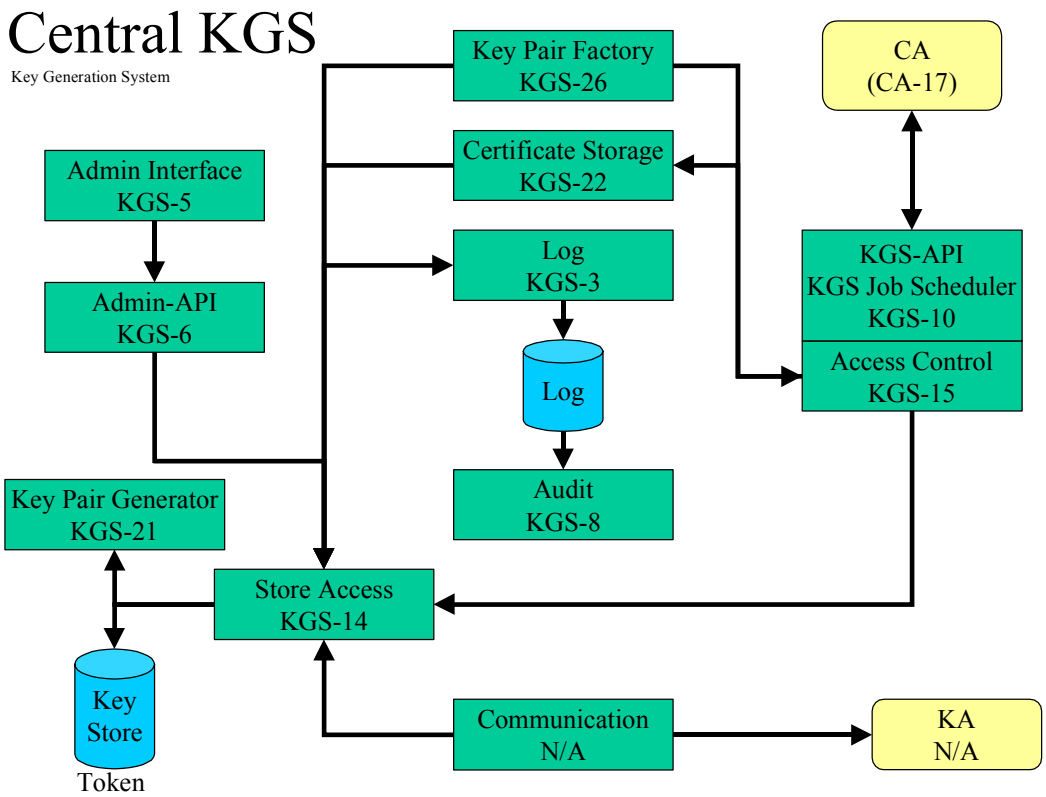


Figure 11: Central KGS

The central KGS (Figure 11) supports centralised key generation if requested by the CA (optionally a central KGS may generate keys by claiming requests from the BO – thus doing the initiating step on its own). Operation of the central KGS is controlled through the KGS-API. The key pair factory and optionally – if needed – the certificate storage module operate on batches of key generation requests or certification responses exchanged with the CA online or by data media. They are using the store access module to create and store keys and certificate. The store access module may use an external key pair generator to obtain key pairs to be stored on the token or use a built-in token functionality to create key pairs. Once the token is produced it is distributed toward the end entity. Distribution may be performed through file delivery if the token to be distributed is a soft key solution.

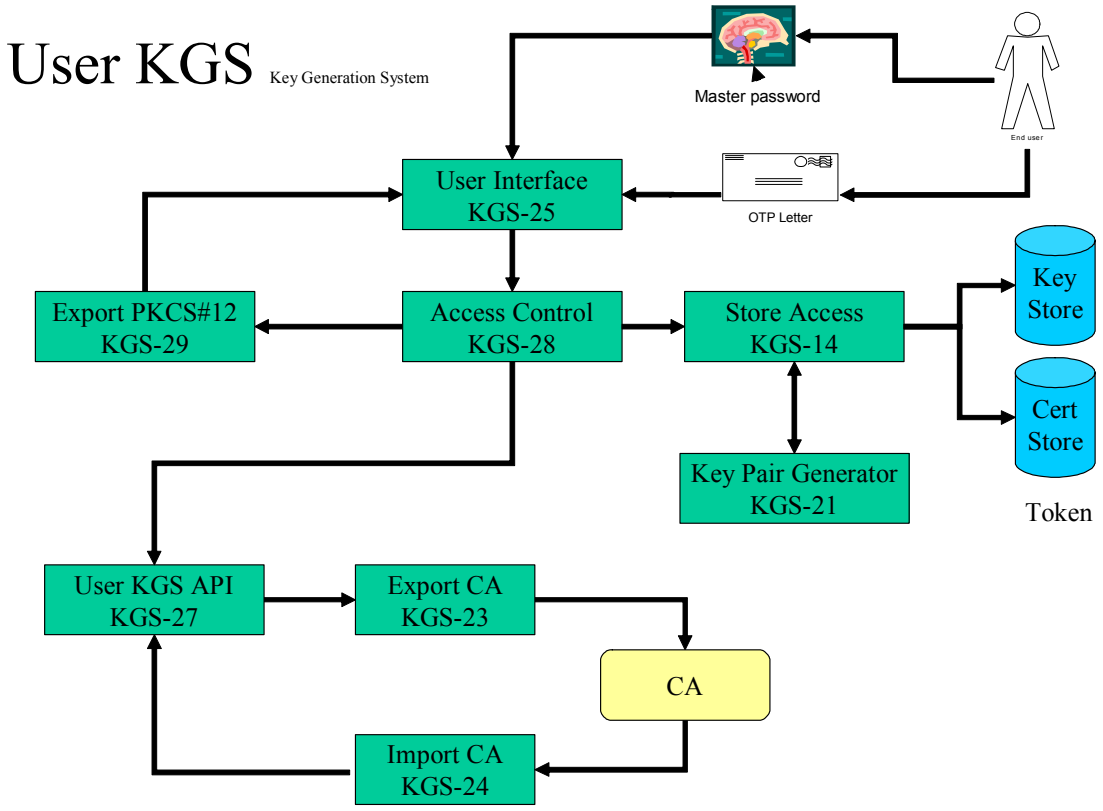


Figure 12: User KGS

The User KGS (Figure 12) is a set of modules provided to the user to enable him to generate key pairs locally and present public key values toward the CA (or RA) for certification. The user operates key generation and certificate requests through a user interface which may be presented through a web-interface. The store access module does manage the key and certificate storage at the user (which may be a soft- or hardware token), and optionally uses the key pair generator if none is supplied with the token. After key generation, the user creates and forwards a certification request containing his public keys either authenticated with an OTP obtained during registration in the front office directly toward the CA or presents it to a RA. Certificates will be returned as a response, either online or as a file and imported into the key store. The module structure resembles an architecture typically found in Internet browsers.

2.5 Module Table

Blocks:	Block number:	Modules:
FO	FO-1	User Interface
	FO-2	Communication to BO
BO	RA-1	User Interface
	RA-2	Validation
	RA-3	Log
	RA-4	RA database
	RA-5	Entry / Renewal
	RA-7	Communication to CA
	RA-8	Audit
	RA-10	RA-API
	RA-13	Revocation
	RA-14	Entity/Cert Browser
	RA-15	Access Control
	RA-16	Admin Interface
	RA-17	Admin API
	RA-18	RA PK Acceptor
	RA-20	Communication to FO
CA	CA-1	User Request Handler
	CA-2	Writemail
	CA-3	Log
	CA-5	Admin Interface
	CA-6	Admin API
	CA-8	Audit
	CA-9	OTP distributor
	CA-10	CA-API
	CA-11	OTP Generator
	CA-12	OTP Authenticator
	CA-13	Communication to CSP
	CA-14	CA database
	CA-15	Access Control
	CA-16	Publication
	CA-17	Communication to KGS

	CA-18	CRL Factory
	CA-20	Certificate Update Agent
CSP	CSP-1	CSP-API
	CSP-2	Certificate Signer
	CSP-3	CRL Signer
	(CSP-4)	(Crypto Engine, sub module)
KC	KC-2	Certificate Factory
	KC-5	Admin Interface
	KC-6	Admin API
	KC-15	Access Control
	KC-26	Key Pair Factory
	KC-30	Secret Export
	KC-31	Secret Printing
	KC-32	Secret Import
Central KGS	KGS-3	Log
	KGS-5	Admin Interface
	KGS-6	Admin API
	KGS-8	Audit
	KGS-10	KGS-API Job Scheduler
	KGS-15	Access Control (Central KGS)
	KGS-22	Certificate Storage
	KGS-26	Key Pair Factory
Both Central and User KGS	KGS-14	Key Store
	KGS-21	Key Pair Generator
User KGS	KGS-23	Export Certificate Request to CA
	KGS-24	Import Certificate Replies from CA
	KGS-25	User Interface
	KGS-27	User KGS API
	KGS-28	Access Control (User KGS)
	KGS-29	Export PKCS#12

2.6 Function Table

FUNCTIONS		MODULES	
Number	**Name**	**Number**	**Name**
F1	Generate keys and certificate	RA-1	User Interface
		RA-2	Validation
		RA-3	Log
		RA-4	RA database
		RA-5	Entry / Renewal
		RA-7	Communication to CA
		RA-10	RA-API
		RA-15	Access Control
		CA-10	CA-API
		CA-14	CA database
		CA-15	Access Control
		CA-13	Communication to CSP
		CA-16	Publication
		CA-17	Communication to KGS
F1.1	Generate keys	KGS-26	Key Pair Factory
		KGS-21	Key Pair Generator
F1.2	Generate certificate	CSP-1	CSP-API
		CSP-2	Certificate Signer
		CSP-4	Crypto Engine
F2	Recover private encryption key		Out of scope
F3	Revoke certificate	RA-1	User Interface
		RA-3	Log
		RA-4	RA database
		RA-7	Communication to CA
		RA-10	RA-API
		RA-13	Revocation
		RA-14	Entity/Cert Browser
		RA-15	Access Control
		CA-10	CA-API
		CA-13	Communication to CSP
		CA-14	CA database
		CA-15	Access Control
		CA-16	Publication
		CA-18	CRL Factory

		CSP-1	CSP-API
		CSP-3	CRL Signer
		CSP-4	Crypto Engine
F4	Repudiate keys	CA-10	CA-API
F5	Publish certificate in a directory	CA-16	Publication
F6	Publish CRL	CA-16	Publication
F7	Suspend certificate	RA-13	Revocation
		CA-10	CA-API
F8	Reactivate suspended certificate	RA-10	RA-API
		CA-10	CA-API
F9	update Data of certificate	FO-1	User Interface
		FO-2	Communication to BO
		RA-20	Communication to FO
		RA-10	RA-API
		RA-15	Access Control
		RA-5	Entry / Renewal
F9.1	certificate renewal	CA-14	CA database
		CA-20	Certificate Update Agent
		CA-2	Writemail
		CA-10	CA-API
F10	View event log	RA-8	Audit
		CA-8	Audit
		KGS-8	Audit
F11	Recover certificate		out of scope

2.7 Process Model

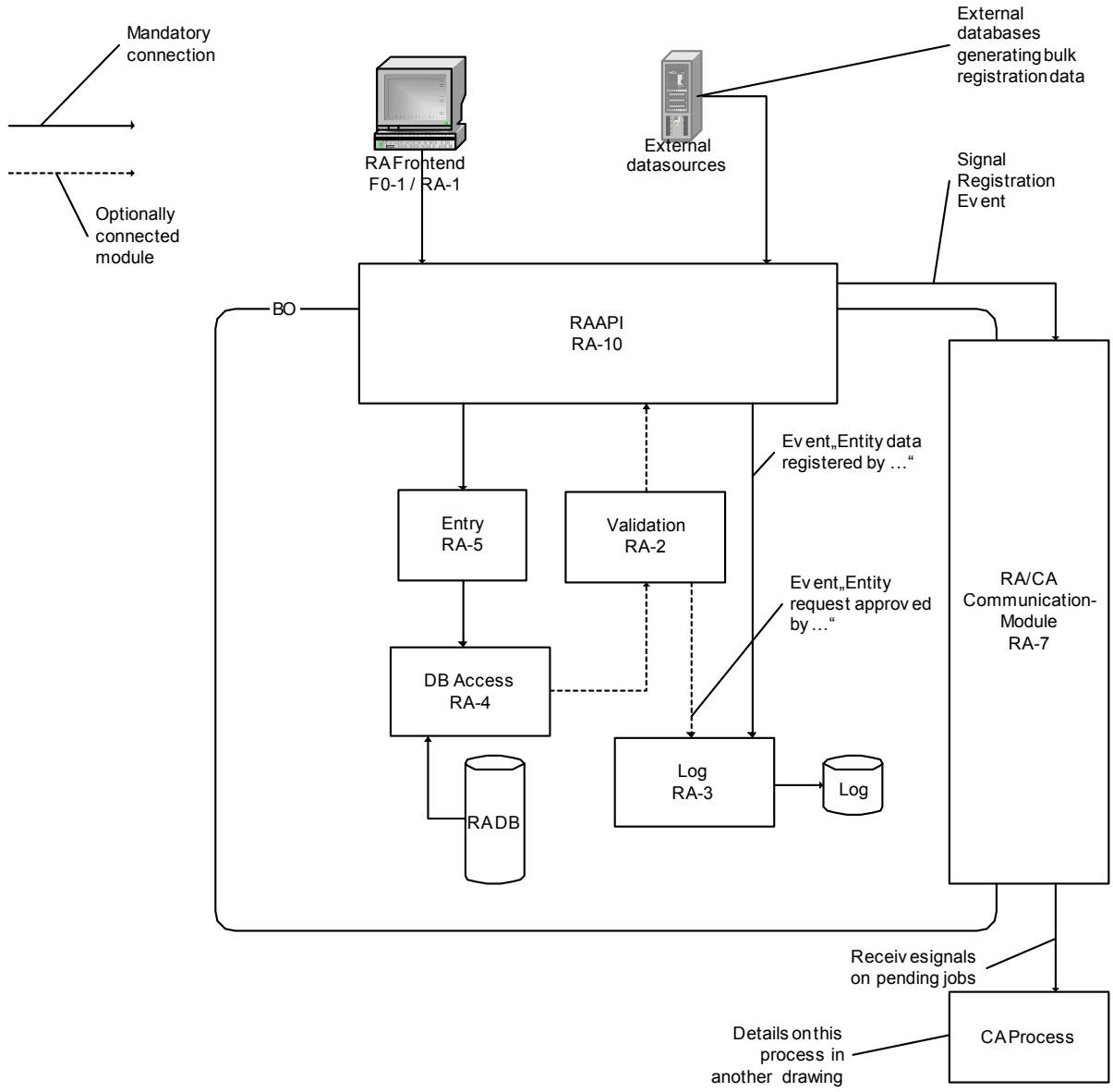


Figure 13 RA Process

The RA process organises the entity data management and the PKI actions caused by entities to this data:

- Initial Acquisition of data

- Entity data content update

Each of the above actions will cause approval and production or rejection of creation of new certificates for that entity.

Each event will cause (if approved) a certificate production job and an associated entity information record within the RA DB. The central module RA-API is the data

entry point for all registration events. It creates the database entries and links registration events and entity information record for identical entities together.

The RA DB entry contains all information for the production of certificates, beside public key values: Number of keys to be certified, CA to request certification from, Certificate profiles, Type of token, link to entity information record, method for public key creation and delivery: central or decentral. The RA does not deal with the keys itself.

The creation of each new entry in the RA DB is signalled to the CA through the communication bus module. The CA may react immediately to that signal and start certificate production, may collect the signals into batches or may ignore them and instead rely on a regular inspection of the RA DB.

If the RA DB is operated in a distributed and replicated manner, the communication bus module will offer services to replicate the DB with a central repository where all local RAs are connected to. Logically it is essential that there is only one logical RA DB, as entities may show up at different RA offices.

Note that all connections are authenticated, data records signed, and access to any system authenticated and controlled. Access Management, Signature Creation and Signature verification modules are not shown in this drawing.

The final output the RA process generates are verified entity data records with authenticated origin, logged registration events, and certificate production orders.

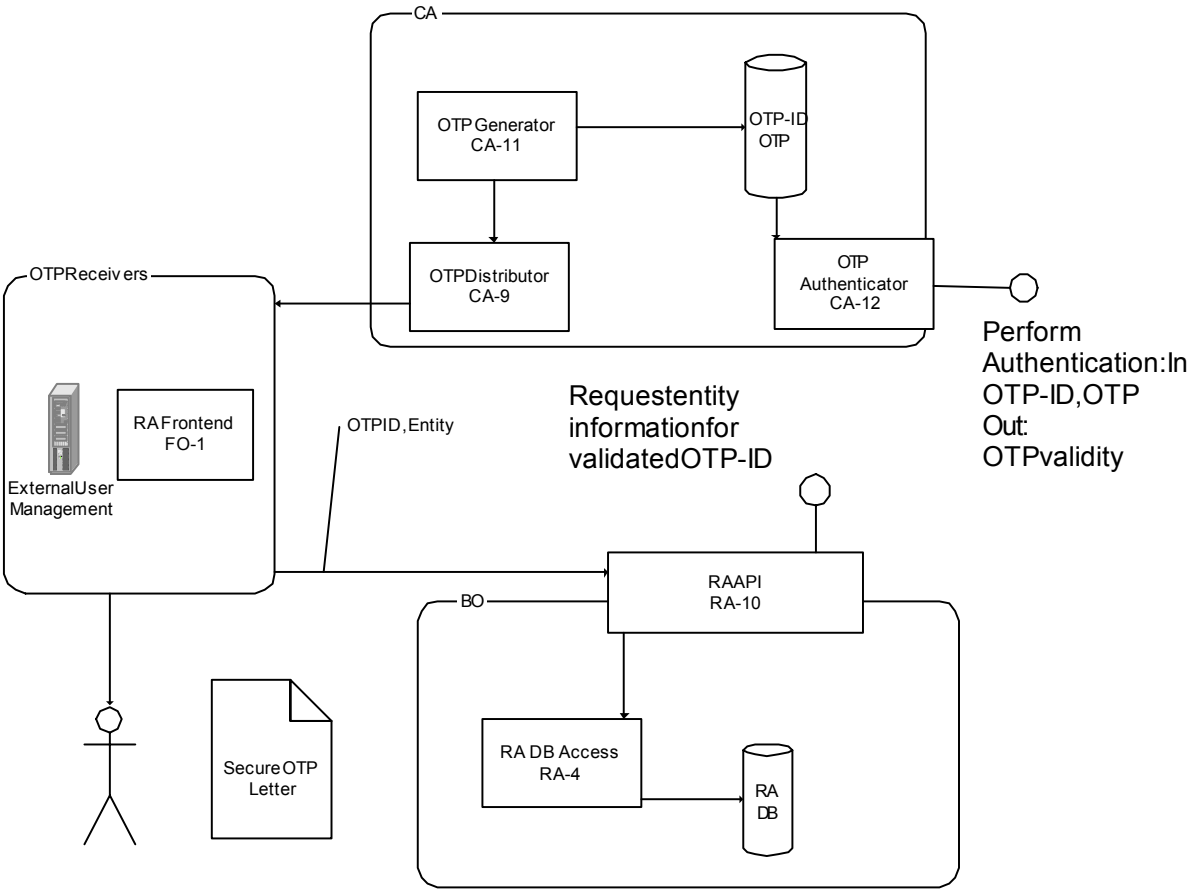


Figure 14 OTP Process

The OTP Process is an optional Element of a PKI. It creates and delivers one-time passwords (OTPs) to entities. Entities may use these OTPs to communicate with any PKI-Service if they have no other means to authenticate, and authentication of the communication is needed.

One use case scenario is the certification of public keys generated by users and presented to a certification service across a network (e.g. The internet). Using the OTP the user authenticates the origin of his/her key to the certification service, which uses entity data obtained from the user during a preceding registration step to issue certificates to the presented public key.

An OTP has a value, that is a shared secret known to the user and the OTP Authenticator module. The OTP is associated with an ID that is known to the RA or other institution that hand over sealed OTPs to entities. This ID is recorded within the entity information delivered by the RA or other institution. The ID links an OTP to an entity information record.

The OTP authenticator delivers validity to a service requesting authentication by presenting the OTP-ID and the authentication value (either the OTP itself or a authentication value derived from it by a known algorithm (e.g. a MAC)). The OTP authenticator validates the authentication value for the given OTP-ID and returns the validation result. The requesting service can then lookup entity information associated with a validate OTP ID through the RA-API.

A practical example for producing and distributing OTP values is the production of security envelopes (PIN mailers) with randomly generated OTPs inside and OTP-IDs printed on the outside. These letters are distributed toward RAs or other institutions in batches, where they get assigned to users in either a bulk or a case by case manner.

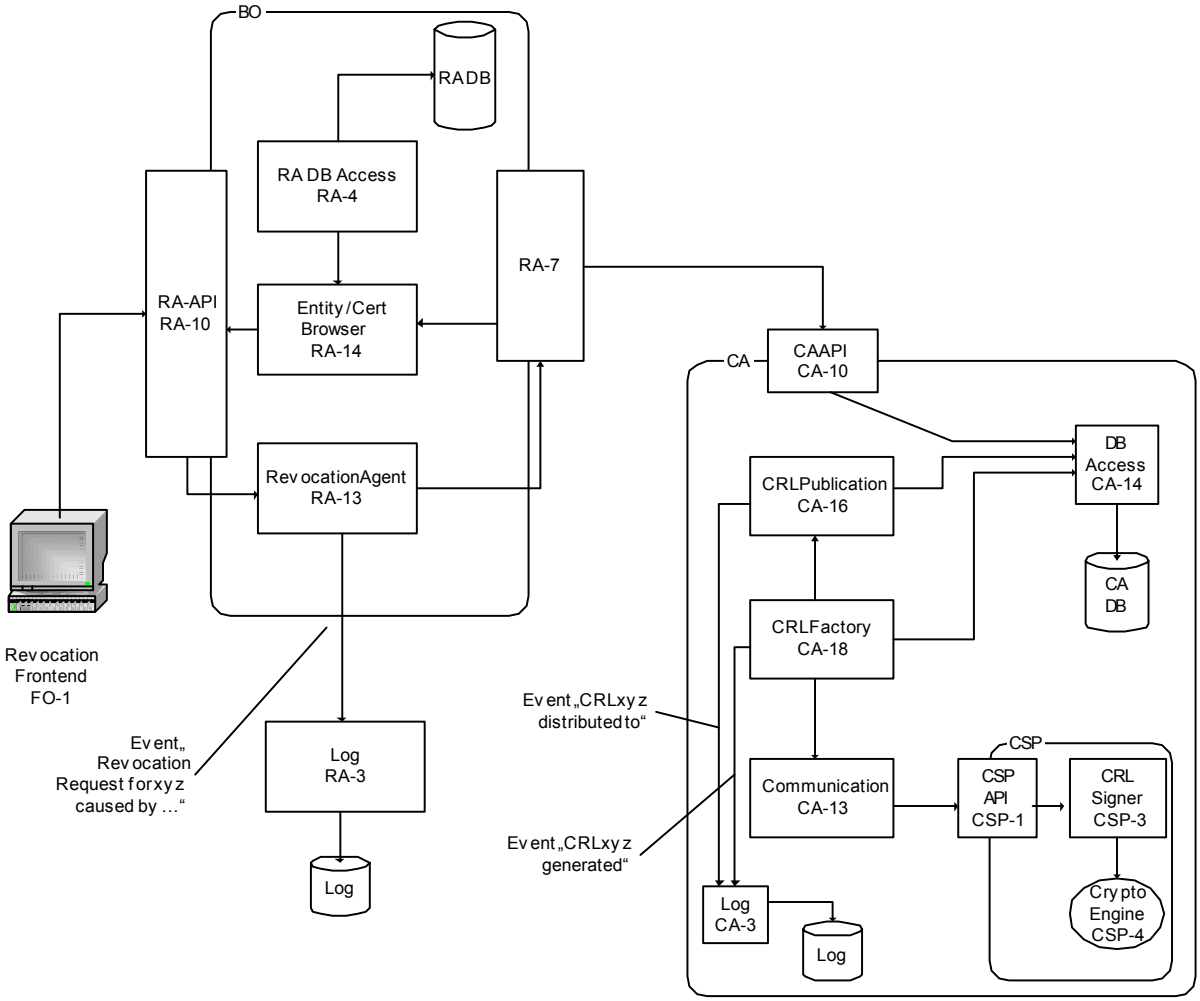


Figure 15: Revocation Process

The revocation process causes certificates to be revoked, suspended, and possible existing suspensions cancelled.

Revocation is operated through a front-end, that makes use of a database browser presenting entity and certificate information. It includes the ability to query for all certificates issued to an entity and shall support the lookup of the right certificates to be revoked, suspended or reactivated from a suspended state. The revocation front-end may be implemented in two flavours: one operated by a CA official, and one operated by the user itself. The latter one needs to impose restrictions on the data presented through the database browser to user related data.

Once the certificates that shall receive a status change are identified, the revocation agent is requested to update the state entry of those certificates within the CertDB.

The CRL Generator is a „daemon" process, that operates at regularly scheduled intervals, querying the database for certificates to be included into the next revocation list to be generated. The result is transformed into a X.509 CRL and forwarded to the CRL Publication which takes over the responsibility to publish the CRL to configured distribution points, which may include several LDAP directories as well as a database archive of CRLs.

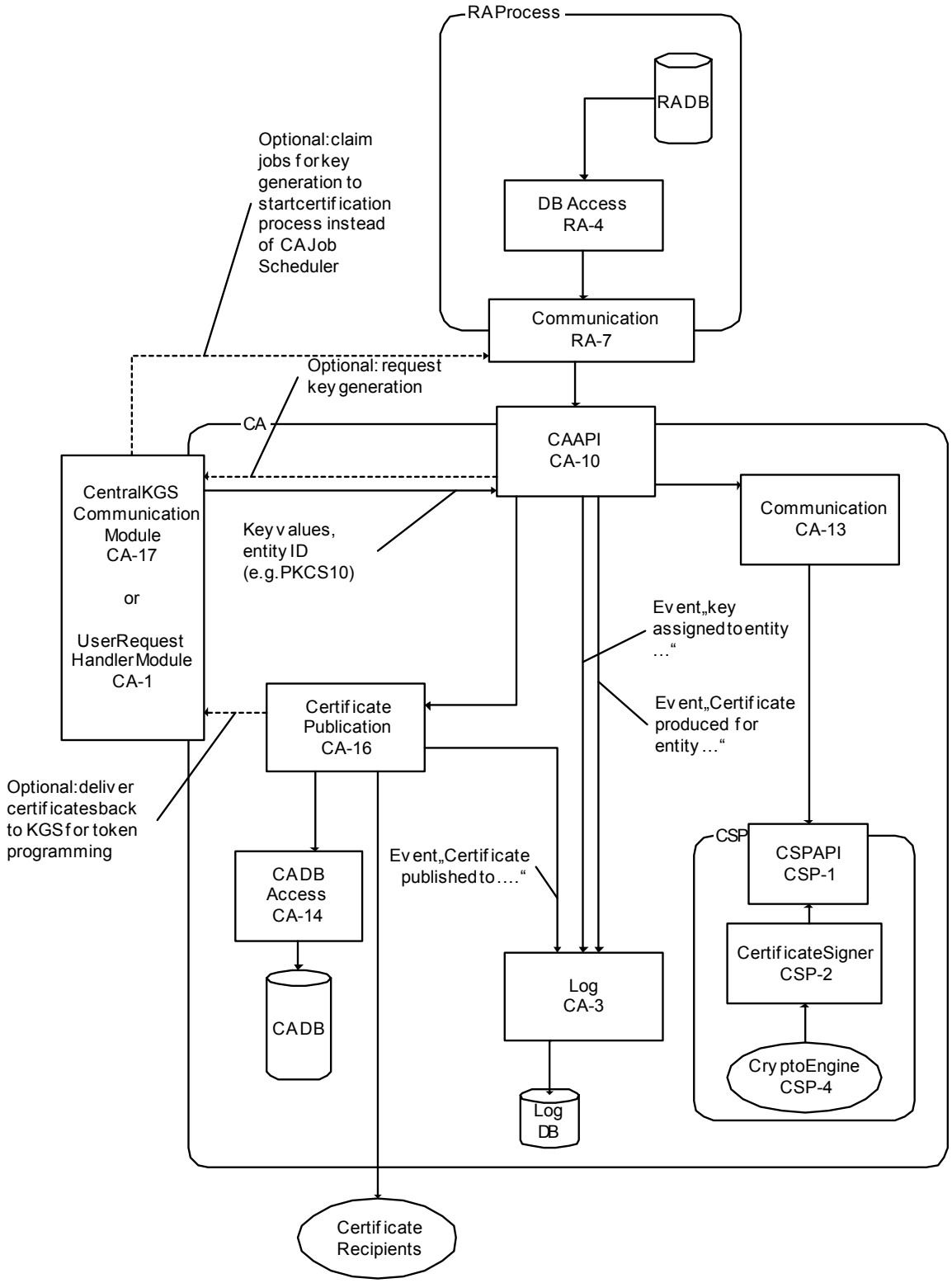


Figure 16: CA process

The CA process combines entity information with public key values obtained from the KGS and distributes them toward certificate recipients (Users, KGS, Directories etc.). There are x ways to start the production of certificates: a) the CA API acts as a job scheduler and claims pending jobs and queries the KGS to deliver public keys for

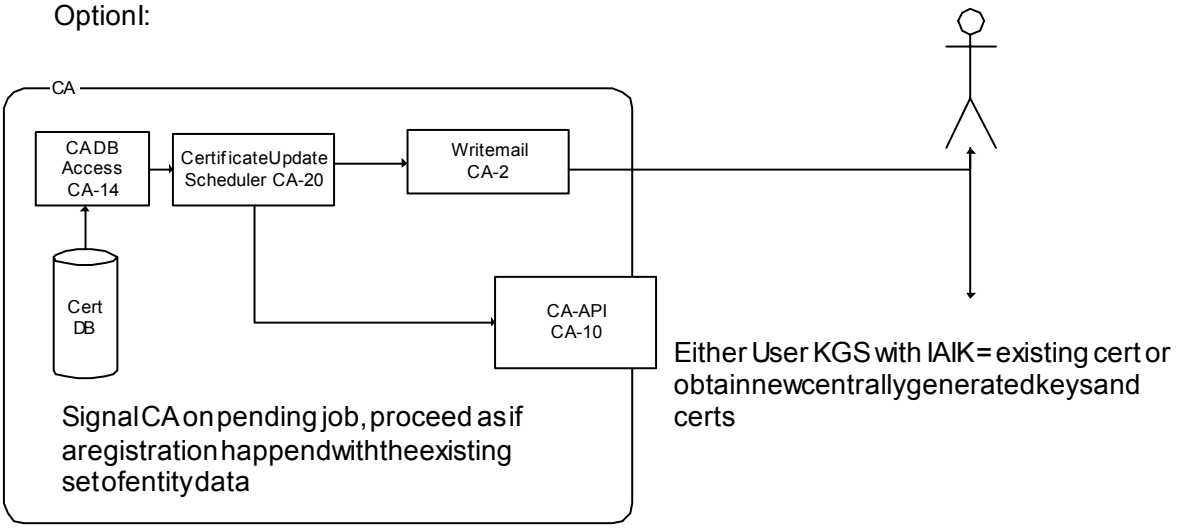
those jobs (and deliver private keys to entities - this is not used for entity based key generation) b) the KGS claims pending jobs, generate keys and deliver public key values to the CA API and c) the KGS is a user based key generator and delivers a public key/entity (or job) reference to the CA API for certificate production.

Any option allows the CA API to construct a complete certificate production job with keys and entity information and forward that to the certificate factory where certificates are produced. The factory delivers output to the certificate distributor that takes care of distributing certificates to directories, users or the KGS for combined delivery of private keys and certificates.

Produced certificates are stored in a CertDB for lifetime management. The crypto module containing the CA keys and the CertDB is shared with the CRL process.

Twooptionsforcertificaterenewal:
I:certificate(s)arerenewedwithnewkeygeneration
II:certificate(s)arerenewedwithoutnewkeygeneration

OptionI:



OptionII:

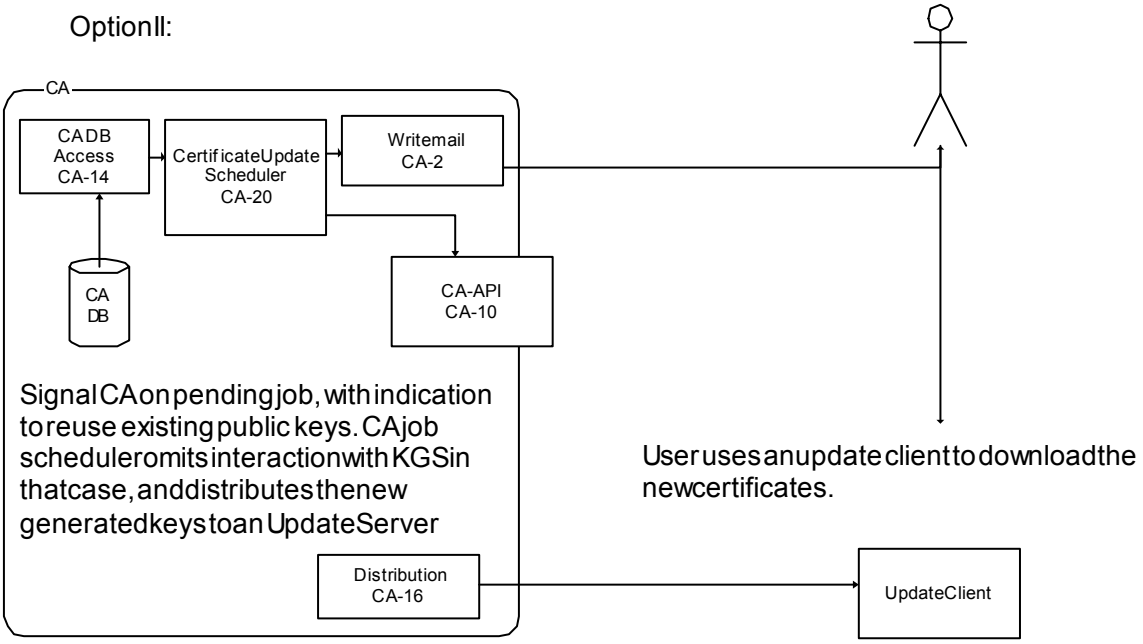


Figure 17: Certificate renewal process

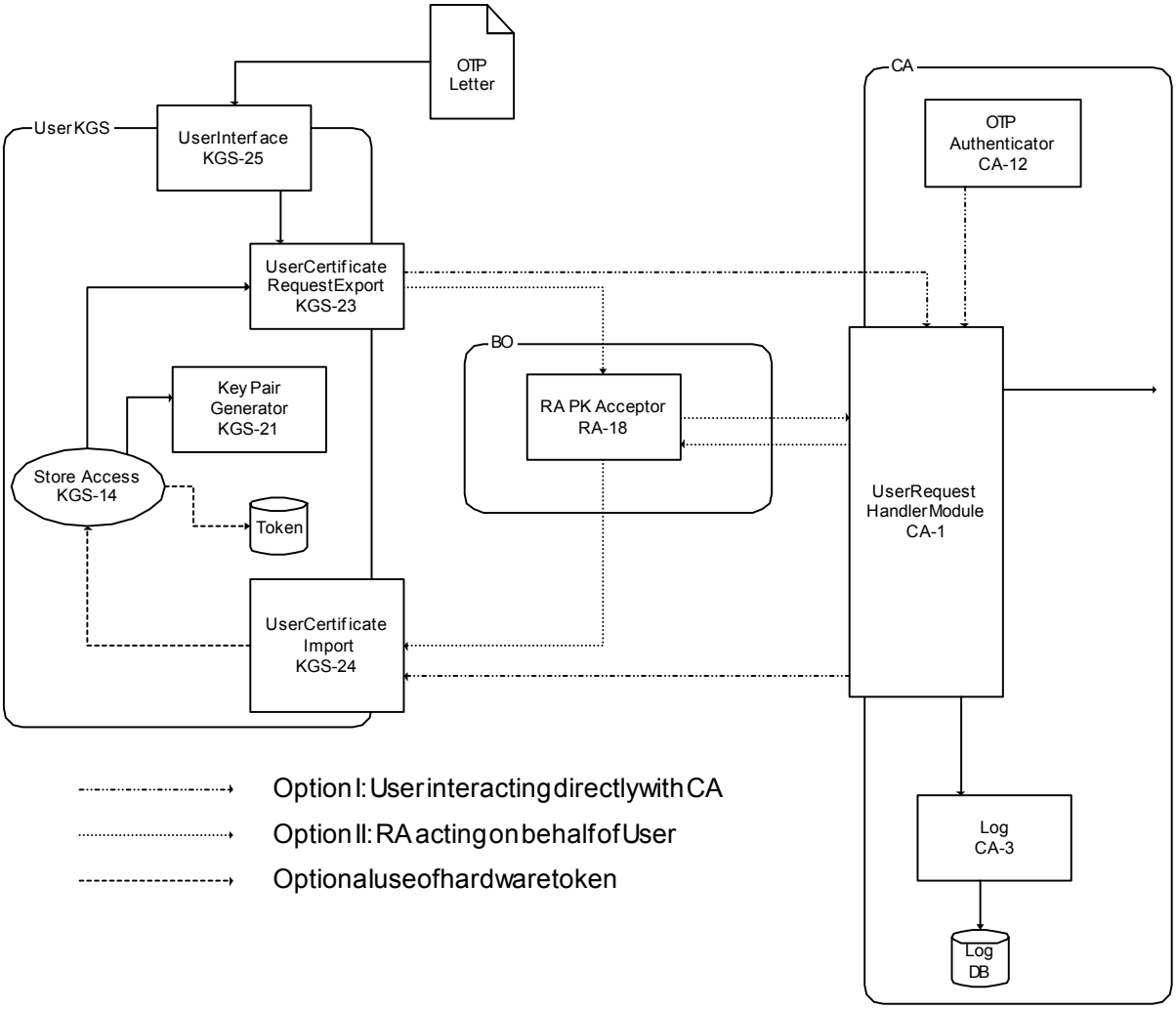


Figure 18 User Key Generation System

The user key generation system allows public keys generated by entities to be introduced into the certification process. Optionally the user key generation system acts as a receiving end of certificates after production.

Public key introduction by entities may be done in a direct online interaction between User (Entity) and CA or with assistance by an RA. The User Key Pair Generator module produces key pairs, stores the key values in a secure entity owned storage space (token) and allows to export the public key values to either the User Certificate Request Export agent or the RA. The User Certificate Request Export agent adds authentication information and a reference to a registered entity (OTP, OTP-ID) to the request and delivers it to the CA User Request Handler module where it gets authenticated and a certificate production job is issued to the CA API. If an RA requests certification on behalf of the user the RA adds the entity reference and authentication to the request. The transport of public keys from the entity to the RA is offline in that case.

If the certificate(s) are to be returned to the entity they get delivered through the CA User Request Handler module to either the RA where it is forwarded offline to the User certificate import agent, or directly to the import agent.

Note that standard software like Internet Explorer, Netscape Communicator and Internet Information Server are implementations of a User KGS, User Certificate request Agent, and User certificate import agent.

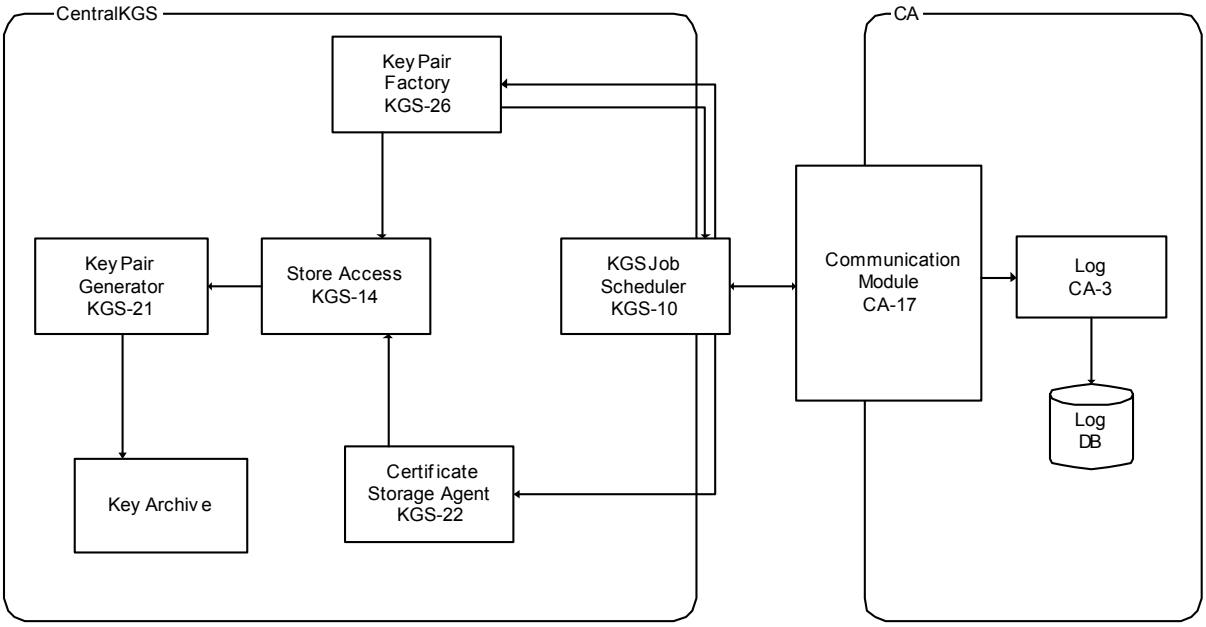


Figure 19: Centralized key generation system

Centralised key generation allows a PKI to produce a token with keys and optionally certificates stored on it and deliver it to an entity unable to produce the token on its own.

The KGS job scheduler either starts key generation on its own by claiming pending job entries in the Job DB of the RA or by receiving an order from the CA job scheduler. It issues key generation requests to the Central Key Pair Factory and storage module which produces key pairs, store them safely on the token and deliver the public key values to the CA job scheduler.

If the certificates are to be stored on the entity token as well, the certificate distributor forwards them to the certificate storage agent to store them on the token.

After the token is finalised, the organisational token distribution module takes care for token delivery.

2.8 Data Model

2.8.1 Database organisation

Specifically the PKI uses internal databases for the following:

- To store all certificate requests, certificate issued, and CRLs
- To store logs of all system and operator events

- To store details on registration policies
- To store the messaging communications between CA and RA
- To store eventually registration information not included in certificates (e.g. customer attributes, biometric data, one time password …)
- To store all audit data

In order to assure the security of the overall system each critical action/ insert in the database should be signed by the key of the process or operator storing the particular entry. Critical are: every action in the CSP, each entry and validation of entity data, each revocation done by an operator. In addition each certificate request should have a transaction identification number associated with it and should be indexed within the log in the database.

The basic data items stored in databases are entities and certificates. Entities are persons or systems that are controlling one or more public key pairs. Entities do have one or more names. The same name must not be assigned to two or more distinct entities. Certificates bind names of entities to public keys.

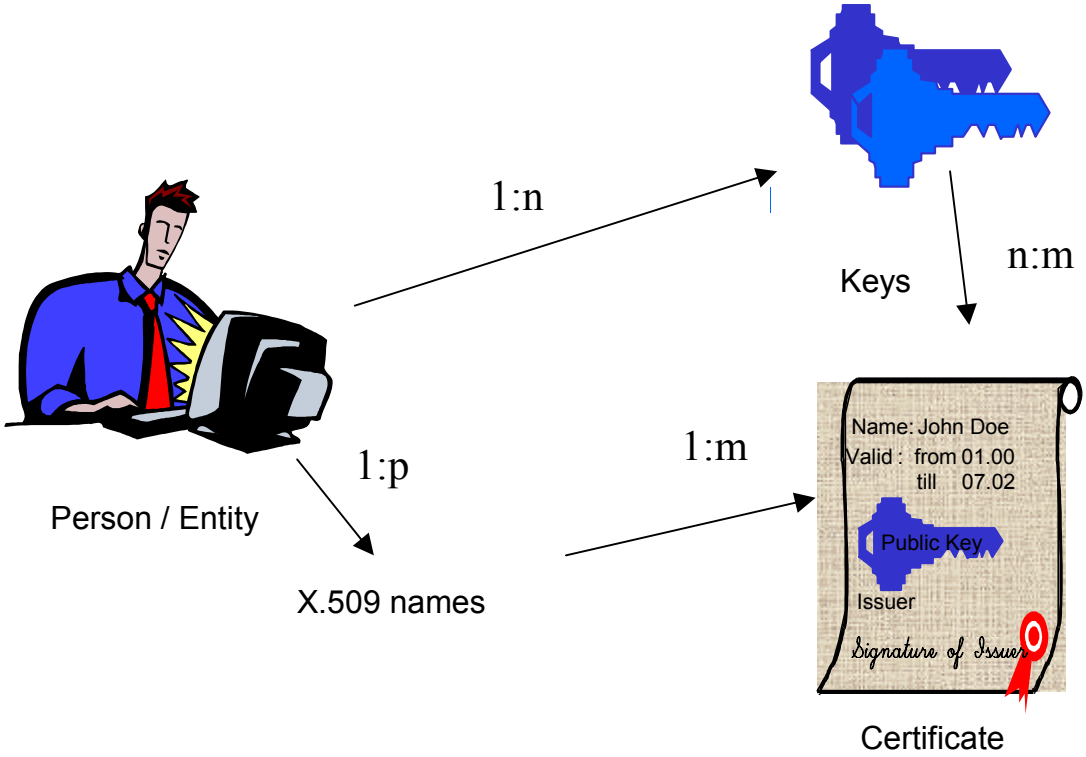


Figure 20: Data relations in EU-PKI system

2.8.2 Data Structure RA-DB

Entity information is all information that is acquired and validated from an entity during registration. The RA DB stores all entity information beside public key values and certificates issued to the entity.

Each entity is stored with a set of identifying information items, the *identifying information*. The content of this information item set is used to differentiate entities among each other and to recognise identical entities during each registration act. An

entity providing an identical information item set to an already registered entity is treated as the identical entity, an entity providing an information item set differing in one item at least is treated as a separate entity. The information item set must be chosen in such a way that it can be assured that identical entities provide the same information set on each registration act, that the information set can be easily verified, that the set is minimal for the purpose of differentiating entities, and that there are no optional elements. An example for an identifying information item set for natural persons are: surname, birth name, date of birth, sex, city of birth. Each entity is assigned a unique entity ID.

An entity must provide another information set, the *request information*, for the purpose of certificate production. The structure and content of the request information depends on the purpose of the certificates and the policy to be applied for certificate production and issuance. This information item set may contain mandatory and optional elements and may share elements with the identifying information set. An entity may provide one or more different request information sets, which may be identical with request information sets provided by other distinct entities.

The definition of a identifying information set and a request information set constitutes a "RA policy". RA policies are identified by an RA policy ID unique within the EU PKI system. The same entity ID may be assigned to different entities using different RA policies, and different entity IDs established by different RA policies may be assigned to the same entity. For the purposes of the EU PKI system, entities registered by different RA policies are treated as distinct entities.

The RA DB stores the following relations for each RA policy:

Primary key[1] Entity ID, Primary key Identifying information

Primary key Entity ID, Request Information

Definition of identifying information with a rule for determining equality

Definition of request information

Entity ID and request information are communicated to the CA for certificate production purposes. Identifying information remains within the RA.

2.8.3 Data Structure CA-DB

For the purpose of obtaining certificates the CA constructs an X.509 distinguished name from the request information for the entity and one or more certificate extensions. The rules how to construct the X.509 name and the extension set is called a "Certificate profile". The set of entity IDs and their X.509 distinguished names (without extensions) defines a "PKI name domain". As the input for the X.509 name construction (the request information) is provided by the RA policy, the RA policy is part of the definition of a PKI name domain. An RA policy may be reused within several PKI name domains. Within a PKI name domain, no two or more

[1] A "Primary key" designates an information item in the relation that is unique for each row.

distinct entities may share the same X.509 distinguished name. On or more CAs will sign certificates for entities within one PKI domain.

A specific CA is identified by its own X.509 subject name (without extensions) and must not sign certificates within more then one PKI name domain. A CA may have more then one key to sign certificates and may have more then one certificate signed by different other CAs (including a self-signed certificate). There may be more then one certificate issued to this CA by another CA (including self-signed certificates). Each certificate issued to this CA for the purpose of certificate signing, regardless of origin and issuer, must state the same subject name of this CA[2].

A CA may issue more then one certificate to any entity/key pair relationship within its PKI name domain.

Each CA is managing the certificates that carry the subject name of this CA in its issuer field. For managing the certificates a certificate "Status" is maintained which can be valid (default), revoked, or suspended. At any point in time there does exist a (possibly empty) Certificate revocation list containing all certificates that are not valid. This list is distributed at regular intervals with a limited validity lifetime carrying a signature of either the CA that did issue the certificates or of an associated entity entitled to sign CRLs for the CA. At least one key of the CA with a valid certificate is used for signing the CRL, maybe more then one, each with a valid certificate. In case an associated entity signs the CRL, it has to be listed in the certificate profile, and it may use any of its keys entitled for CRL signing in a valid certificate to sign the CRL.

This list is stored in its actual state as a relation within the CA DB. An archive of older states of the revocation lists is maintained as well.

A "complete PKI definition" consists of

PKI name domain:	an RA policy
	a certificate profile
a set of CAs serving the PKI name domain	

For each PKI definition the CA DB stores the relations:

Primary key Certificate, entity ID

Primary key Certificate, Status, Revocation date

Certificate profiles

RA policy ID

At the global scope the CA DB stores

Primary key CA ID, CA Certificate

[2] This is required by the concept of X.509 certificates. Each CA issuing a certificate to another CA is serving the same PKI name domain, the name domain of certifying authorities. This is a global namespace and uniqueness of names within this domain has to be assured by agreement and organizational methods. The EU PKI system cannot ensure that a CA name is not reused in another EU PKI or other PKI installation.

Primary key CA ID, CRL signing key Certificate

Primary key CRL signing key certificate, CRL, Creation Date

Primary key PKI Definition ID, CA ID

2.8.4 Data Structure Access-DB

Each RA-, CA- and KGS-system has an associated Access-DB that associates actors within the PKI-system with rights to operate certain functions. Actors are operators, administrators and entities. The access DB does organise actors in to groups and assigns rights to groups. Each member of the group has the right associated with the group. For every specific right there exists a standard group that has exactly this right exclusively granted. New groups can be defined that combine rights . An actor can be member of several groups and a group can be the member of another group thus inheriting its rights.

The access DB stores

Primary key Group ID, actor ID or group ID

Primary key Group ID, set of rights

2.8.5 Data Structure Log-DB

Each system maintains a record of actions in a log DB. Each row in the log relation records one event. The relations stores

Primary key event ID, time, event source, event description

The even description must allow queries for

All events related to a specific certificate

All events related to a specific entity

All events related to a specific CA

with or without restrictions on time and event source.

2.8.6 System Resilience

In designing the PKI architecture in our point view hardware resilience has been maximised where practical.

Normally the following steps have been taken to address data resilience:

1. The Master directory Server should be completely duplicated to another directory server. Entries should be replicated immediately they are added to the directory
2. In the case of Master Directory Server failure, the Replicated Directory server may be configured to take the place of the Master allowing the service to function as normal during repair work
3. The **CA and RA database server** should be duplicated to another database server for hot-backup reasons. The period of the replication is to agreed

4. In the case of **CA and RA database server failure** the replicated database server may be configured to take its place during repair

For these reasons PKI should be designed to recover from system and communication failures that may occur in the usage. Any component may go out of service without losing transactions.

The PKI needs in any case to be able to recovery from any disaster.

There are two mission critical parts of the PKI: the private keys and the databases.

Provided that these two elements are backed up , the PKI can be completely recovered or rolled backed to some previous state.

2.9 Common Technical Standards & Protocols

2.9.1 Assumptions

EUPKI users expressed their need for a high-quality libre software PKI.

As it has been mentioned wp3 documents, EUPKI should be:

- Interoperable
- Scalable
- Multi-platform
- Modular
- Based on open standards and protocols

The PKI market became a mature one. PKI leaders are working now on the adoption of new standards and protocols, allowing greater interoperability between products.

Web services can be defined as Business objects that can be deployed independent of technologies and protocols. These technologies are today the state of the art in distributed and multi-platform environment architectures. EUPKI should use web services standards and protocols whenever is possible.

2.9.2 Standards and standardisation bodies (for PKI)

2.9.2.1 IETF, IAB, Isoc

Internet engineering tasking force (IETF) and Internet Architecture board (IAB) are the technical related activities of the Internet society (Isoc). Most of the actual internet standards were specified by IETF.

http://www.ietf.org

http://www.iab.org

http://www.isoc.org

2.9.2.2 W3c

The World Wide Web Consortium was created in October 1994 to develop and promote protocols used by the web. Focused on http and html in the beginnings , they are now responsible for the new XML standards (signature, encryption, transport, ..)

http://www.w3c.org

2.9.2.3 Oasis

OASIS is an international standardisation body that produces standards for security, web services and XML.

http://www.oasis-open.org/

2.9.2.4 ETSI

ETSI are responsible for specifying European telecommunication standards. In particular, they are working on specifications related to electronic signature.

http://www.etsi.org

2.9.2.5 CEN

CEN is working for harmonisation between different national standardisation bodies in Europe.

http://www.cenorm.be/

2.9.3 Common protocols

2.9.3.1 PKI

IETF PKIX

PKIX protocols and related RFC's for PKI are widely discussed and known. Thus, they will not be treated in this document.

The entry point for more information is:

http://www.ietf.org/html.charters/pkix-charter.html

SPKI

There are no known (yet) IETF drafts for Simple Public Key Infrastructure.

Some requirements can be found on the IETF web site.

http://www.ietf.org/html.charters/spki-charter.html

2.9.3.2 Cryptographic operations

RSA PKCS standards

http://www.rsasecurity.com/rsalabs/pkcs/index.html

2.9.3.3 Web services related

Based on essential standards like *SOAP* (for remote exchange) , *WDSL* (web services description language) and *UDDI* (directory of web services) , different architectural frameworks like .Net , Java one and Rosetta Net allows the design, implementation and deployment of web services.

Soap

Transport:

RPC, smtp/pop3/imap, mom, http

Structure:

Header

non functional aspects : Session, transaction, security

Body

Functional & specific aspects:

Business specific methods

Open source links:

http://xml.apache.org/soap/

WDSL

Definition:

XML language that contains information about the interface & semantics of a call to a Web service.

Structure:

Works with different languages (Java , c++, c#,)

Open source links:

http://www-124.ibm.com/developerworks/projects/wsdl4j/

UDDI

Definition:

Repository for web services

Structure: four data structure types

businessEntity : Information about the party who publishes information about a service

businessService : Descriptive information

bindingTemplate : Technical information

tModel : Descriptions of specifications

Example:

http://demo.alphaworks.ibm.com/browser/

Open source links:

http://www-124.ibm.com/developerworks/oss/uddi4j/

XKMS (XML key management standard)

Definition :

Key management operations with XML. Aim to replace PKIX standards.

The XML Key Management Specification (XKMS) is composed by two parts : the XML Key Information Service Specification (X-KISS) and the XML Key Registration Service Specification (X-KRSS).

Transport:

Soap based

Functions :

request, renew, validate, revoke

Extensions :

x-bulk(smartcard management) , x-krss

Links:

http://www.xmltrustcenter.org/xkms

Xbulk

Xbulk is an extension to Xkms, to enable XML key management bulk operations, especially with smartcards. The specification is proposed by Baltimore, with the help of smartcards vendors.

http://www.baltimore.com/devzone/x-bulk/

SAML (Security Assertion markup language)

Definition:

Handle the formats and protocols for authentication and authorisation information

Web services security

Maintained by IBM.

http://www-106.ibm.com/developerworks/library/ws-secure/

2.10 Common Security Requirements

The EU-PKI system deals with secrets (CA keys, User private keys) and entity information to be treated according to data protection laws. All information stored persistently within the PKI system needs protection against alteration as audit ability and review ability of PKI transactions are a major concern. The system is operated by personal. To avoid security breaches it is necessary to design and manage the rights assigned to each operator to operate a certain function of the PKI. For certain functions the enforcement of an 4-eye principle must be realised.

See Deliverable D4.1a for a more formal description of security requirements to the EU-PKI system on the bird's view level.

The general rules to meet the security requirements basically are

1. All online connections of modules are protected for confidentiality and bilateral end-to-end authentication of parties taking part into transaction.
2. All operators are able to authenticate themselves. Each sub-system of the EU-PKI is managing an access control module that verifies authentication and determines the "right to do".
3. All modules offering online-available services should be able to authenticate themselves.
4. All data transmitted either offline or online is carrying a signature of its originator. If that is not possible, the data has to be treated as "untrusted" and special attention should be assigned to its handling, see next point.
5. All systems with online connections are must be protected by IT and network security elements like firewalls and intrusion detection systems when deployed. During implementation care must be taken when processing data received online from an untrusted source (this is mainly data received by end entities and front-end systems operated outside specially access-controlled areas).

2.11 Scalability

Each of the above-described modules should be able to deal with increasing amounts of requests. As a matter of fact, EUPKI may be used in several environments types, like small enterprises, large ones, government services…

As specified in the WP3 requirements, the following deployment requirements in terms of scalability are identified :

- Millions of users for:
    - E-Government ;
    - Social services ;
    - Home networking ;
    - Telco ;
    - Utilities
- Thousands of users for :
    - Financial services ;
    - Internal PKI ;
- Hundreds of users for :
    - Small companies ;
- Bulk processes

Going further, as a working hypothesis, the system should be able to manage 10% of the users simultaneously per hour.

In order to get a global idea of the induced peak of load, three scenarios will be considered : 100, 10 000 and 1 000 000 users.

Following the working hypothesis, this leads these respective peaks of load :

	100 users	**10 000 users**	**1 000 000 users**
Simultaneous users	**0.003 user /sec**	**0.3 users/sec**	**30 users/sec**

However, please note that this does not include bulk registration processes.

Going even more further, we could now consider the impact on each module. Indeed, some of them may be solicited several times during a single "session", whereas some others will only be requested once. This is indicated in the column called "Solicitation". As a result, the next three columns show the peak induced in each scenario. Please bear in mind that these estimates are for a single key-pair / certificate request – i.e. in case that two certificates were be requested, some operations may be done twice, de facto doubling the related peak.

Blocks:	Block number:	Modules:	Solicitations :	Load (requests/sec) :		
FO	FO-1 / RA-1	User Interface	N/A	N/A	N/A	N/A
	FO-2	Communication to BO	x1 or up to ? (Bulk)	0,003 ➔ ?	0,3 ➔ ?	30 ➔ ?
BO	RA-1 / FO-1	User Interface	N/A	N/A	N/A	N/A
	RA-2	Validation	x1 or up to ? (Bulk)	0,003 ➔ ?	0,3 ➔ ?	30 ➔ ?
	RA-3	Log	x5	0.015	1,5	150
	RA-4	RA Database	x1	0,003	0,3	30
	RA-5	Entry / Renewal	x1	0,003	0,3	30
	RA-7	Communication to CA	x2	0.006	0,6	60
	RA-8	Audit	x1	0,003	0,3	30
	RA-10	RA-API	x3	0.009	0,9	90
	RA-13	Revocation	x1	0,003	0,3	30
	RA-14	Entity/Cert Browser	x1	0,003	0,3	30
	RA-15	Access Control	N/A	N/A	N/A	N/A
	RA-16	Admin Interface	N/A	N/A	N/A	N/A
	RA-17	Admin API	x1	0,003	0,3	30
	RA-18	RA PK Acceptor	x1	0,003	0,3	30
	RA-20	Communication to FO	x1 or up to ? (Bulk)	0,003 ➔ ?	0,3 ➔ ?	30 ➔ ?
CA	CA-1	User Request Handler	?	N/A	N/A	N/A
	CA-2	Writemail	x2	0.006	0,6	60
	CA-3	Log	x8	0,024	2,4	240
	CA-5	Admin Interface	N/A	N/A	N/A	N/A
	CA-6	Admin API	x1	0,003	0,3	30
	CA-8	Audit	x1	0,003	0,3	30
	CA-9	OTP Distributor	x1	0,003	0,3	30
	CA-10	CA-API	x4	0.012	1,2	120
	CA-11	OTP Generator	x1	0,003	0,3	30
	CA-12	OTP Authenticator	x1	0,003	0,3	30
	CA-13	Communication to CSP	x2	0.006	0,6	60
	CA-14	CA database	x1	0,003	0,3	30
	CA-15	Access Control	x1	0,003	0,3	30
	CA-16	Publication	x2	0.006	0,6	60
	CA-17	Communication to KGS	x2	0.006	0,6	60
	CA-18	CRL Factory	N/A	N/A	N/A	N/A
	CA-20	Certificate Update Agent	x3	0.009	0,9	90

		Agent				
CSP	CSP-1	CSP-API	x4	0.012	1,2	120
	CSP-2	Certificate Signer	x1	0,003	0,3	30
	CSP-3	CRL Signer	N/A	N/A	N/A	N/A
	CSP-4	Crypto Engine	?	N/A	N/A	N/A
Central KGS	KGS-3	Log	x5	0.015	1,5	150
	KGS-5	Admin Interface	?	N/A	N/A	N/A
	KGS-6	Admin API	?	N/A	*N/A*	N/A
	KGS-8	Audit	x1	0,003	0,3	30
	KGS-10	KGS-API Job Scheduler	x4	0.012	1,2	120
	KGS-15	Access Control	x2	0.006	0,6	60
	KGS-26	Key Pair Factory	x1	0,003	0,3	30
	KGS-22	Certificate Storage	x1	0,003	0,3	30
Both Central and User KGS	KGS-14	Key Store	x2	0.006	0,6	60
	KGS-21	Key Pair Generator	x1	0,003	0,3	30
User KGS	KGS-23	Export Certificate Request	N/A	N/A	N/A	N/A
	KGS-24	Import Certificate Replies	N/A	N/A	N/A	N/A
	KGS-25	User Interface	N/A	N/A	N/A	N/A
	KGS-27	User KGS API	N/A	N/A	N/A	N/A
	KGS-28	Access Control	N/A	N/A	N/A	N/A

Legend :

- "?" indicates that this module is no longer part of the specifications
- "N/A" means that the estimate number of users does not apply to this module (most often because it is local or administrative-tasks oriented).
- The decreasing color gradation indicates the decreasing needs for scalability.

.

2.12 Concept for Administration and Configuration

Each module needs configuration information relevant for its own activities. Each functional block (RA, CA, CSP etc.) has an administration module that is used to maintain and present these configuration sections to the modules in a centralised way.

Basic configuration options applicable to each block are network information (description of communication channels with endpoints etc.), database location information, access control information. The RA system in particular will have

configuration options for the supported RA policy definitions and PKI name domains. The CA system will contain several PKI definitions and CA definitions listing all applicable options related to a specific CA (keys, CA certificates, optional assigned CRL signing keys and certificates, distribution points for certificates and CRLs).

Further module-specific implementation options will be defined in D4.3.

3 Module Descriptions

Remark: states of the objects (requests etc.) will be treated in the fine spec (D4.3).

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 Preconditions

Web-browser (decision for browser based implementation already taken if feasible).

3.1.3 Input, and from where

Action of the operator

3.1.4 Output, and whereto

The User Interface module calls only the R.A. API (RA-10) module, either directly or via a communication module (FO-2).

3.1.5 Relevant standards

SSL v3 between user and server if realised web-based.

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 polices.

3.2.2 Preconditions

The network must be down for the packages to get appended to the spool.

3.2.3 Input, and from where

The API provides individual packages to be appended by the module.

Data are from F.O. User Interface (FO-1).

3.2.4 Output, and whereto

The API requests the contents of the spool, and the check sum as one single entity.

Data are sent to R.A. API (RA-10).

3.2.5 Relevant standards

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 Preconditions

IE 5.x and more, or Communicator 4.7x and more.

3.3.3 Input, and from where

Action of the validation operator

RA-4: List of requests to be approved, maybe signed by RA officer.

3.3.4 Output, and whereto

Log of the action (signed)(RA-3).

Database access (RA-4): State change of requests

Communication to CA (RA-7): Event Transmission

3.3.5 Relevant standards

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 Preconditions

3.4.3 Input, and from where

All modules: Contents and nature of the logs

3.4.4 Output, and whereto

Database: Logs to storage

Log (RA-3): (Error) Time, Module’s name, Operator's Id, Error’s origin, Error’s Id, Comment.

3.4.5 Relevant standards

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.

3.5.2 Preconditions

3.5.3 Input, and from where

Functional module: Data to be backed up or to be read.

3.5.4 Output, and whereto

Log (RA-3): (Error) Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment.

A database: Data to be backed up or to be read

3.5.5 Relevant standards

SQL, (JBDC abstraction if java is the implementation).

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 Preconditions

The operator must have access to this functionality.

3.6.3 Input, and from where

From RA-10 (entity data)

3.6.4 Output, and whereto

Log RA-3: (Audit) Time, Module's name, Operator's Id, End user's DN.

Log RA-3: (Error) Time, Module's name, Operator's Id, End user's DN, Error's origin, Error's Id, Comment

DB Access RA-4: back up of the entry.

User Interface RA-1 via RA-10: Creation's Result.

3.6.5 Relevant standards

XML for data acquisition.

CMP or XKMS

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 Preconditions

3.7.3 Input, and from where

Event from a Functional module.

RA-4: Request to be transmitted.

3.7.4 Output, and whereto

CA-10: Request to be transmitted.

RA-4: Request state changed depending on Result.

3.7.5 Relevant standards

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 Preconditions

The operator must have access to this functionality.

3.8.3 Input, and from where

Operated through RA-1 and RA-16, receives requests from RA-10 and RA-17.

Log data source.

3.8.4 Output, and whereto

Response to queries to the requesting module for presentation and printout

3.8.5 Relevant standards

None

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.

RA-10 presents the services available to the operator and provided by the RA:

1. Recording of information of an entry
2. Bulk Input
3. Ask certificate for an entry
4. Consultation of the certificates and their state for an entry
5. Validation of a request or operation
6. Ask for revocation of a certificate
7. Ask for renewal of a certificate
8. Consultation of the statistics and the logs of RA

For the whole of these 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 Preconditions

The operator or server must be authenticated

Access Control module must have turned over the profile.

3.9.3 Input, and from where

Access Control RA-15: Approval and definition of the operator's profile.

All actions seized by the operator through the IHM module (RA-1 or FO-1).

Batch actions from F.O. Communication (FA-2) or Bulk Input.

3.9.4 Output, and whereto

Call of the functional module following the request of the operator with transmission of the list of necessary information.

3.9.5 Relevant standards

SOAP for bulk input interface.

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 Preconditions

The operator must have access to this functionality.

For the success of the operation the certificate must be valid.

3.10.3 Input, and from where

R.A. API (RA-10): CA distinguished name, serial number of the certificate.

3.10.4 Output, and whereto

Log (RA-3): (Audit) Time, Module's name, CA, Operator's Id, End user's DN, Serial number.

Log (RA-3): (Error) Time, Module's name, CA, Operator's Id, End user's DN, Error's origin, Error's Id, Comment.

DB Access (RA-4): Back up request.

Communication to CA (RA-7): Event Transmission

User Interface : The operator is informed that the request was transmitted to the CA.

3.10.5 Relevant standards

CRL, OCSP, CMP or XKMS

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 a 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 Preconditions

The operator must have access to this functionality.

3.11.3 Input, and from where

R.A. API (RA-10): DN of the End-User.

3.11.4 Output, and whereto

Log (RA-3): (Audit) Time, Module's name, Operator's Id, End user's DN.

Log (RA-3): (Error) Time, Module's name, Operator's Id, End user's DN, Error's origin, Error's Id, Comment.

User Interface (RA-1): The list and the state of the certificates of the end user by CA.

3.11.5 Relevant standards

X.509, PKCS7.

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 Preconditions

The operator/administrator must have a valid certificate.

The operator/administrator must be authenticated.

3.12.3 Input, and from where

R.A. API (RA-10)/Admin API (RA-17): The Certificate of the operator/administrator

R.A. API (RA-10)/Admin API (RA-17): DN of the operator/administrator.

3.12.4 Output, and whereto

R.A. API (RA-10), Admin API (RA-17): The access control list of the operator/administrator.

Log (RA-3): (Access) Time, Operator's Id, Authentication result, Option: The reason of the failure.

Log (RA-3): (Error) Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment

3.12.5 Relevant standards

SSL v3, X.509

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 Preconditions

IE 5.x and more, or Communicator 4.7x and more.

3.13.3 Input, and from where

Action of the administrator

3.13.4 Output, and whereto

The User Interface module calls only the Admin-API module.

3.13.5 Relevant standards

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.

RA-17 presents the services available to the administrators:

1. manage the administrators
2. manage the operators
3. Consultation of the statistics and the logs of RA
4. Configuration setup and changes to the BO modules

For the whole of these 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 Preconditions

The administrator must be authenticated.

Administrator's Access Control module must have turned over the profile.

3.14.3 Input, and from where

Administrator's Authentication: The administrator authenticates himself with his certificate.

Administrator's Access Control: Definition of the administrator's profile.

All actions seized by the administrator through the IHM module (RA-16).

3.14.4 Output, and whereto

Call of the functional module following the request of the administrator with transmission of the list of necessary information (RA-17).

3.14.5 Relevant standards

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 Preconditions

The RA officer must be authenticated

3.15.3 Input, and from where

Entity Browser (RA-14): Find entity entry.

R.A. API (RA-10): PK via the User Interface.

3.15.4 Output, and whereto

Database access (RA-4): Bindings between Entity and PK (signed).

Log (RA-3): the action (signed).

3.15.5 Relevant standards

PKCS10

3.16 Module Number CA-1: User Request Handler

3.16.1 Functionality

The CA-1 module is the only entry point for interaction between the end user and the CA. The objective of this module is to generate the GUI and display it at the end user. Thanks to it the end user can send a Certification Signing Request.

3.16.2 Preconditions

Web-browser (decision for browser based implementation already taken if feasible).

3.16.3 Input, and from where

Action of the end user via a web browser or the User KGS (KGS-23).

3.16.4 Output, and whereto

The User Interface module calls only the C.A. API (CA-10) module.

3.16.5 Relevant standards

SSL v3 between user and server if realised web-based.

3.17 Module Number CA-2: Writemail

3.17.1 Functionality

This module generates text for emails that will be sent to end-users, operators or administrators.

3.17.2 Preconditions

3.17.3 Input, and from where

CA Modules: The type of message to generate.

3.17.4 Output, and whereto

Log (CA-3): (Audit) Time, Module’s name, RA's Id, Mail’s Id, comment.

Log (CA-3): (Error) Time, Module’s name, RA's Id, Request, Error’s origin, Error’s Id, comment

CA Modules: The generated message.

3.17.5 Relevant standards

RFC822, MIME (S-MIME ?).

3.18 Module Number CA-3: Log

3.18.1 Functionality

This module is used as interface between the CA 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 emissions of events network like a SNMP trap.

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".

3.18.2 Preconditions

3.18.3 Input, and from where

All C.A. modules: Contents and nature of the logs

3.18.4 Output, and whereto

Database: Logs to storage

Log (CA-3): (Error) Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment.

3.18.5 Relevant standards

3.19 Module Number CA-5: Admin Interface

3.19.1 Functionality

The CA-5 module is the only entry point for interaction between the administrator and the CA. 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.

3.19.2 Preconditions

IE 5.x and more, or Communicator 4.7x and more.

3.19.3 Input, and from where

Action of the administrator

3.19.4 Output, and whereto

The User Interface module calls only the Admin-API module (CA-6).

3.19.5 Relevant standards

3.20 Module Number CA-6: Admin-API

3.20.1 Functionality

This module is dedicated to the administrators. It establishes the link between the actions seized by the administrators and the administration's functionalities of the CA.

CA-6 presents the services available to the administrators:

- manage the administrators
- Consultation of the statistics and the logs of CA
- Create Sub-CA

For the whole of these 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.

3.20.2 Preconditions

The administrator must be authenticated.

Administrator's Access Control module must have turned over the profile.

3.20.3 Input, and from where

Administrator's Authentication: The administrator authenticates himself with his certificate.

Administrator's Access Control: Definition of the administrator's profile.

All actions seized by the administrator through the IHM module (CA-5).

3.20.4 Output, and whereto

Call of the functional module following the request of the administrator with transmission of the list of necessary information.

3.20.5 Relevant standards

3.21 Module Number CA-8: Audit

3.21.1 Functionality

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

3.21.2 Preconditions

The operator must have access to this functionality.

3.21.3 Input, and from where

Operated through CA-5, receives requests from CA-6.

Log data source.

3.21.4 Output, and whereto

Log (CA-3): (Audit) Time, Module's name, Administrator's Id.

Log (CA-3): (Error) Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment.

Response to queries to the requesting module for presentation and printout.

3.21.5 Relevant standards

None

3.22 Module Number CA-9: OTP Distributor

3.22.1 Functionality

The objective of this module is to distribute the OTP for the user.

3.22.2 Preconditions

3.22.3 Input, and from where

OTP generator (CA-11): User's Id, OTP.

3.22.4 Output, and whereto

Log (CA-3): (Error) Time, Module's name, User's Id, Error's origin, Error's Id, Comment.

3.22.5 Relevant standards

3.23 Module Number CA-10: CA-API

3.23.1 Functionality

This module is the heart of the CA. It establishes the link between the actions seized by the end-users or the RAs (RA-7) and the functionalities of the CA.

CA-10 presents the services available to the operator and provided by the CA:

1. OTP Authenticator
2. SendMail
3. Publication
4. Consultation of the statistics and the logs of CA

CA works as a black box. The requests arriving of the RA are checked (Authentication of the origin of the data and integrity of the transmitted contents) and are carried out immediately.

This module ensures the management of the requests of generation, renewal and revocation of the certificates. It calls also the interface's modules with "Central KGS" and "CSP".

3.23.2 Preconditions

Check module (CA-15) must have checked the integrity and the origin of the data.

3.23.3 Input, and from where

All the requests transmitted by the RAs (RA-7).

Database Access (CA-14): Control Access List of the API invoker.

3.23.4 Output, and whereto

Log (CA-3): (Audit) Time, RA's Id, Request, comment.

Log (CA-3): (Error) Time, Module’s name, RA's Id, Request, Error’s origin, Error’s Id, comment

Communication to KGS (CA-17): Requests (Key Generation, Certificate storage)

Communication to CSP (CA-13): Requests (certificate generation)

3.23.5 Relevant standards

3.24 Module Number CA-11: OTP Generator

3.24.1 Functionality

The objective of this module is to create an OTP for a user.

3.24.2 Preconditions

The user should not already have active OTP. In this case, the previous OTP will be erased and an error log will be generated.

3.24.3 Input, and from where

Admin-API (CA-6): Entity's Id.

3.24.4 Output, and whereto

OTP Distributor (CA-9): Result (Entity's Id, OTP).

Log (CA-3): (Error) Time, Module's name, User's Id, Error's origin, Error's Id, Comment

3.24.5 Relevant standards

3.25 Module Number CA-12: OTP Authenticator

3.25.1 Functionality

The objective of this module is to check the validity of a user ID with OTP which was assigned to him.

Once the validation obtained, the module turns over the positive result and removes this OTP for this ID.

3.25.2 Preconditions

The end-user must have an OTP.

3.25.3 Input, and from where

C.A. API (CA-10 via CA-15): Entity's Id, OTP.

3.25.4 Output, and whereto

CA Access Control (CA-15): Answer for OTP Authentication.

Log (CA-3): (Access) Time, Entity's Id, result.

Log (CA-3): (Error) Time, Module's name, Entity's Id, Error's origin, Error's Id, Comment

3.25.5 Relevant standards

3.26 Module Number CA-13: Communication to CSP

3.26.1 Functionality

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

3.26.2 Preconditions

3.26.3 Input, and from where

C.A. API (CA-10): Request to be transmitted.

CRL Factory (CA-18): CRL to be signed.

3.26.4 Output, and whereto

C.S.P API (CSP-1): Request to be transmitted

3.26.5 Relevant standards

X.509

3.27 Module Number CA-14: CA Database

3.27.1 Functionality

This module is used as interface between the CA and the solution of backup of information.

3.27.2 Preconditions

3.27.3 Input, and from where

Functional module: Data to be stored or to be read.

3.27.4 Output, and whereto

Log (CA-3): (Error) Time, Module's name, RA's Id, Error's origin, Error's Id, Comment.

Database source: Data to be stored or to be read

3.27.5 Relevant standards

SQL, (JBDC abstraction if java is the implementation).

3.28 Module Number CA-15: Access Control

3.28.1 Functionality

This module checks the requests emitted by the RA. The request consists of three parts:

- The request itself.
- The name of transmitting RA.
- The signature of the request.

This module checks the signature starting from the certificate of RA. That brings a double insurance:

- The origin of the request (recognized RA).
- The integrity of the contents of the request.

3.28.2 Preconditions

The CA must have the RAs certificates.

3.28.3 Input, and from where

C.A. API (CA-10)/Admin API (CA-6): The Certificate of the RA/administrator

C.A. API (CA-10)/Admin API (CA-6): DN of the RA /administrator.

3.28.4 Output, and whereto

C.A. API (CA-10), Admin API (CA-6): The access control list of the RA/administrator.

Log (RA-3): (Access) Time, RA's Id, Authentication result, Option: The reason of the failure.

Log (RA-3): (Error) Time, Module's name, RA's Id, Error's origin, Error's Id, Comment

3.28.5 Relevant standards

SSL v3, X.509

3.29 Module Number CA-16: Publication

3.29.1 Functionality

This module publishes the certificates and the CRL. To ensure the coherence of the publication's service, this module erases also the certificates published following their revocation or their expiry.

3.29.2 Preconditions

3.29.3 Input, and from where

Database Access (CA-14): Certificates and CRL to publish.

3.29.4 Output, and whereto

Third Part Server (Directory, Web, Ftp …): Certificate, CRL.

Log (CA-3): (Audit) Time, Module's name, RA's Id, Mail's Id, comment.

Log (CA-3): (Error) Time, Module's name, RA's Id, Request, Error's origin, Error's Id, comment

3.29.5 Relevant standards

LDAP, CRL, X.509

3.30 Module Number CA-17: Communication to KGS

3.30.1 Functionality

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

3.30.2 Preconditions

3.30.3 Input, and from where

Event from a Functional module.

Database Access (CA-14): Request to be transmitted.

3.30.4 Output, and whereto

K.G.S API (KGS-10) : Request to be transmitted.

Database Access (CA-14): Request state changed depending on Result.

3.30.5 Relevant standards

3.31 Module Number CA-18: CRL Factory

3.31.1 Functionality

At regularity scheduled intervals the CRL Factory module builds a CRL from the certificates stored in the CA DB and marked as revoked or suspended. The CRL is stored in the CA DB.

The CRL Factory is supported by the CSP.

3.31.2 Preconditions

None.

3.31.3 Input, and from where

Database Access (CA-14): revoked/suspended certificate's serial numbers.

3.31.4 Output, and whereto

Publication (CA-16): event "CRL generated".

Communication to CSP (CA-13): List of certificates to be included in CRL.

Database Access (CA-14): the signed CRL.

Log (CA-3): (Audit) Time, Module's name, Operator's Id, event: CRL generated

Log (CA-3): (Error) Time, Module's name, Operator's Id, Error's origin, Error's Id, Comment.

3.31.5 Relevant standards

X.509

3.32 Module Number CA-20: Certificate update agent

3.32.1 Functionality

CA-20 does a check for certificates about to expire through Database Access (CA-14) and initiates appropriate actions to write mail (CA-2) and C.A. API (CA-10) for renewing certificates.

3.32.2 Preconditions

The certificate must be valid (not revoked).

3.32.3 Input, and from where

Database Access (CA-14): certificates about to expire.

3.32.4 Output, and whereto

Write mail (CA-2): mail to Entity.

3.32.5 Relevant standards

MIME (S-MIME), RFC822.

3.33 Module Number CSP-1: CSP-API

3.33.1 Functionality

The CSP-API module is the interface for all CSP services offered to the CA. It delegates the CA's requests to the various CSP modules.

3.33.2 Preconditions

none

3.33.3 Input, and from where

Communication to CSP (CA-13): any requests and related data.

3.33.4 Output, and whereto

Certificate Signer (CSP-2): Certificate to sign.

CRL Signer (CSP-3): CRL to sign.

Communication to CSP (CA-13): any responses and related data

3.33.5 Relevant standards

3.34 Module Number CSP-2: Certificate Signer

3.34.1 Functionality

The Certificate Signer module produces certificates by signing the combination of entity information and the entities' public keys. The Certificate Factory is supported by the Crypto Engine sub-module CSP-4.

3.34.2 Preconditions

3.34.3 Input, and from where

C.S.P. API (CSP-1): Certificate to sign.

3.34.4 Output, and whereto

Crypto Engine (CSP-4): TBS Certificate Digest, Signature Algorithm.

C.S.P. API (CSP-1): Signed Certificate.

3.34.5 Relevant standards

X.509

3.35 Module Number CSP-3: CRL Signer

3.35.1 Functionality

The CRL Signer module produces a valid CRL by signing a list of revoked and suspended certificates. The CRL Signer is supported by the Crypto Engine sub-module CSP-4.

3.35.2 Preconditions

none

3.35.3 Input, and from where

C.S.P. API (CSP-1): CRL to sign.

3.35.4 Output, and whereto

Crypto Engine (CSP-4): TBS CRL Digest, Signature Algorithm.

C.S.P. API (CSP-1): Signed CRL.

3.35.5 Relevant standards

X.509

3.36 Module Number KC-2: Certificate Factory

3.36.1 Functionality

The Certificate Signer module produces certificates for a CA or for the CA administrator. The Certificate Factory is supported by the Crypto Engine sub-module CSP-4.

3.36.2 Preconditions

none

3.36.3 Input, and from where

Admin. API (KC-5): Certificate to sign.

3.36.4 Output, and whereto

Crypto Engine (CSP-4): TBS Certificate Digest, Signature Algorithm.

DB Access (CA-14): Signed Certificate.

Admin API (KC-6): Signed Certificate.

3.36.5 Relevant standards

X.509

3.37 Module Number KC-5: Admin Interface

3.37.1 Functionality

The KC-5 module is the only entry point for interaction between the administrator and the KC. 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.

3.37.2 Preconditions

Specific application (not web based) to initialize the CA.

3.37.3 Input, and from where

Action of the administrator.

3.37.4 Output, and whereto

The User Interface module calls only the Admin-API module (KC-6).

3.37.5 Relevant standards

X.509

3.38 Module Number KC-6: Admin API

3.38.1 Functionality

This module is dedicated to the administrators. It establishes the link between the actions seized by the administrators and the administration's functionalities of the KC.

KC-6 presents the services available to the administrators:

- Create the root CA and its administrator
- Create Sub-CA

- Create a new administrator

For the whole of these 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.

3.38.2 Preconditions

The administrator must be authenticated.

Administrator's Access Control module must have turned over the profile.

3.38.3 Input, and from where

Administrator's Authentication: The administrator authenticates himself with his certificate.

Administrator's Access Control: Definition of the administrator’s profile.

All actions seized by the administrator through the IHM module (KC-5).

3.38.4 Output, and whereto

Call of the functional module following the request of the administrator with transmission of the list of necessary information.

3.38.5 Relevant standards

3.39 Module Number KC-15: Access Control

3.39.1 Functionality

This module ensures the service of authentication of the administrator. The adopted solution is the authentication of the administrator starting from the presentation to the Admin Interface of his certificate.

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

3.39.2 Preconditions

The administrator must have a valid certificate.

The administrator must be authenticated.

3.39.3 Input, and from where

Admin API (KC-6): The Certificate of the administrator

Admin API (KC-6): DN of the administrator.

3.39.4 Output, and whereto

Admin API (KC-6): The access control list of the administrator.

3.39.5 Relevant standards

SSL v3, X.509

3.40 Module Number KC-26: Key Pair Factory

3.40.1 Functionality

This module generates a key pair based on different secrets entered via the Admin Interface (KC-5).

The number of secrets to enter is variable.

The secrets can be printed on a secure letter.

The generated key can be exported (split) on different smartcard.

3.40.2 Preconditions

3.40.3 Input, and from where

Admin API (KC-6): The different secrets

3.40.4 Output, and whereto

DB Access (CA-14): The public key.

Crypto Engine (CSP-14): The key pair.

Secret Export (KC-30): The secrets.

Secret Printing (KC-31): The secrets.

3.40.5 Relevant standards

RSA

3.41 Module Number KC-30: Secret Export

3.41.1 Functionality

This module generates m secret parts that will be stored into smartcard. Only n parts are necessary (n<=m) to retrieve the originals secrets (Threshold scheme).

Those smartcard are distributed to trusted people that are different from those who entered the original secrets.

3.41.2 Preconditions

3.41.3 Input, and from where

Key Pair Factory (KC-26): The different secrets are computed into a master key.

3.41.4 Output, and whereto

SmartCard: the secret parts.

3.41.5 Relevant standards

3.42 Module Number KC-31: Secret Printing

3.42.1 Functionality

This module prints the different secrets on a printer.

3.42.2 Preconditions

The printer device must be installed. For test purpose the printer can be a file.

3.42.3 Input, and from where

Key Pair Factory (KC-26): The different secrets

3.42.4 Output, and whereto

Printer: The different secrets

3.42.5 Relevant standards

3.43 Module Number KC-32 Secret Import

3.43.1 Functionality

This module generates the master key from the secret parts that was stored into smartcards. Only n parts are necessary among the m parts (n<=m) to retrieve the original master key (Threshold scheme).

3.43.2 Preconditions

The secret parts are read form the smartcard reader connected to the Admin interface.

3.43.3 Input, and from where

Admin API (KC-6): The different secret parts

3.43.4 Output, and whereto

Key Pair Factory (KC-26): The master key.

3.43.5 Relevant standards

3.44 Module Number KGS-3: Log

3.44.1 Functionality

This module is used as interface between the Central KGS 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".

3.44.2 Preconditions

3.44.3 Input, and from where

All modules: Contents and nature of the logs

3.44.4 Output, and whereto

Database: Logs to storage

Log (KGS-3): (Error) Time, Module’s name, Operator's Id, Error’s origin, Error’s Id, Comment.

3.44.5 Relevant standards

3.45 Module Number KGS-5: Admin Interface

3.45.1 Functionality

The KGS-5 module is the only entry point for interaction between the administrator and the Central KGS. 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.

3.45.2 Preconditions

IE 5.x and more, or Communicator 4.7x and more.

3.45.3 Input, and from where

Action of the administrator

3.45.4 Output, and whereto

The User Interface module calls only the Admin-API (KGS-6) module.

3.45.5 Relevant standards

3.46 Module Number KGS-6: Admin-API

3.46.1 Functionality

This module is dedicated to the administrators. It establishes the link between the actions seized by the administrators and the administration's functionalities of the Central KGS.

KGS-6 presents the services available to the administrators:

1. manage the administrators
2. Consultation of the statistics and the logs of Central KGS

For the whole of these 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.

3.46.2 Preconditions

The administrator must be authenticated.

Administrator's Access Control module must have turned over the profile.

3.46.3 Input, and from where

Administrator's Authentication: The administrator authenticates himself with his certificate.

Administrator's Access Control (KGS-15): Definition of the administrator's profile.

All actions seized by the administrator through the IHM module (KGS-5).

3.46.4 Output, and whereto

Call of the functional module following the request of the administrator with transmission of the list of necessary information.

3.46.5 Relevant standards

3.47 Module Number KGS-8: Audit

3.47.1 Functionality

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

3.47.2 Preconditions

The administrator must have access to this functionality.

3.47.3 Input, and from where

Operated through KGS-5, receives requests from KGS-6.

3.47.4 Output, and whereto

Response to queries to the requesting module for presentation and printout

3.47.5 Relevant standards

None

3.48 Module Number KGS-10: KGS API Job Scheduler

3.48.1 Functionality

The KGS-10 module is the only entry point for interaction between the CA and the Central KGS. It performs the communication between CA and KGS, maintains the state of that coordination and coordinates the activities of the remaining KGS modules to deliver the overall functionality of the Central KGS.

KGS-10 presents the services available to the CA and provided by the Central KGS:

- process key pair generation requests issued by the CA
- receive and store certificates presented by the CA on tokens
- request certification for produced public keys

For function 1 and 2 the KGS-10 module is acting as a server reacting on requests generated by the CA. In function 3 the KGS-10 module is acting as a client using the CA as a certification server.

The KGS-10 module operates in a batch-oriented communication mode with the CA-17 module as the remote communication partner. Communication may performed

electronically or by file transfer. Acknowledgement of received messages is not necessary. Each communication partner cannot be sure that a specific request was received by the other end until the result of the message processing is returned. To match response batches to request batches each requesting party needs to assign a message ID to the request, which is repeated in the response to this message.

In function 1 the KGS-10 module will receive a request for generating key pairs for an indicated number of tokens. The request will include the reference to a key generation profile configured at the KGS-26 module, which contains information on token format, number, algorithms, size, and purpose of keys. The request will as well include an indication whether the token to be produced shall wait for certificates to be returned by the CA. The KGS-10 creates a job entry for each requested key pair generation in a job table shared with the KGS-26 module and optionally signals the incoming batch toward the KGS-26 module. Once the KGS-26 module has processed the batches and inserted the results into the job table the KGS-10 module produces and delivers a response batch to the CA. Each completed batch is logged by the KGS-10 module.

Jobs may fail at the KGS-26 module. Failed key generation jobs are not automatically rescheduled, but the admin console will offer a possibility to inspect the failure reason and reschedule the job manually.

If a key generation job does not contain a request by the CA to wait for certificates to be returned it is removed from the job-table.

In function 2 the CA delivers a batch of certificates to be stored on tokens, which store key pairs produced by function 1. It is an error to request certificate storage for a key pair not produced within function 1. The CA repeats the message ID of the key production request in the delivered certificate storage request. The KGS-10 module looks up the production job in the job DB, updates it with the received certificate information and optionally informs the Certificate Storage module KGS-22 on pending jobs. Completion of the job is task of the certificate storage module KGS-22.

In function 3 the KGS-10 module initiates the process of certificate generation by requesting certificates for public keys delivered in the request. [xxx details tbd.]

KGS-10 uses the access control module KGS-15 to verify and authorize the origin of incoming requests and to add authentication information to produced responses.

3.48.2 Preconditions

3.48.3 Input, and from where

Sender Module	Message	Content
CA-17	Key Generation Request	Message-ID M1, Number of key generation jobs, Profile Name
CA-17	Certificate Storage Request	Message-ID M2, Message-ID M1, Certificates

KGS-26	Key generation completed	Set of Job-IDs
KGS-22	Certificate storage completed	Set of Job-IDs

3.48.4 Output, and where to

Addressed Module	Message	Content
CA-17	Key Generation Completed	Message-ID M3, Message-ID M1, Public key values
KGS-26	Key generation requested	Set of Job-IDs
KGS-22	Certificate storage requested	Set of Job-IDs

3.48.5 Relevant standards

XML, SOAP

3.49 Module Number KGS-14: Key Store

3.49.1 Functionality

The store access allows 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, may it be realised as a software storage solution or a hardware device. An example of a Store Access Module in the sense of this specification is a PKCS#11-API implementation.

The operation of the KGS-14 module within the central KGS is controlled by the KGS-26 module and the KGS-22 module. For each key generation job pending at the KGS-26 module the KGS-14 module receives a command to generate a set of key pairs for one token, store it on the token and deliver the public key values as a result of the operation back to the KGS-22 module.

KGS-14 is a device-specific module with a device-independent interface toward KGS-26 and KGS-22. The KGS-14 module makes optional use of a key pair generator Crypto Engine (KGS-21) as a sub-module, i.e. that cryptographic operation to produce a key pair is delegated to that module. The separation of this functionality from the KGS-14 module and the overall control of that functionality through the KGS-14 module is mainly motivated by the fact that cryptographic hardware devices do exist, that implement key generation within their hardware. A KGS-14 module operating on this type of token would make use of the built-in functionality of the device instead of using a software implementation of the Crypto Engine.

3.49.2 Preconditions

3.49.3 Input, and from where

Sender Module	Message	Content
KGS-26	Key Generation Requested	Profile content
KGS-21	Keys generated	set of key pair values

3.49.4 Output, and whereto

Addressed Module	Message	Content
KGS-26	Key Generation Job completed	Result of Job, Set of Public Key Values for this job if no error
KGS-21	Key generation request	number of key pairs, algorithm, size

3.49.5 Relevant standards

None

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

3.50.1 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 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.

3.50.2 Preconditions

The Administrator must have a valid certificate.

The Administrator must be authenticated.

3.50.3 Input, and from where

K.G.S. API (KGS-10) / Admin API (KGS-6): The Certificate of the CA/administrator

K.G.S. API (KGS-10) / Admin API (KGS-6): DN of the CA/administrator.

3.50.4 Output, and whereto

K.G.S. API (KGS-10), Admin API (KGS-6): The access control list of the CA/administrator.

Log (KGS-3): (Access) Time, Administrator's Id, Authentication result, Option: The reason of the failure.

Log (KGS-3): (Error) Time, Module's name, Administrator's Id, Error's origin, Error's Id, Comment

3.50.5 Relevant standards

3.51 Module Number KGS-22: Certificate Storage

3.51.1 Functionality

The module KGS-22 stores certificates on tokens, if the certificates are to be delivered to the user (end entity) together with centrally generated key pairs.

3.51.2 Preconditions

The corresponding key pair must have been generated centrally.

3.51.3 Input, and from where

K.G.S. API (KGS-10) via (KGS-15): certificate, reference to corresponding key pair production.

3.51.4 Output, and whereto

Store Access (KGS-14): certificate.

3.51.5 Relevant standards

X.509

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

3.52.1 Functionality

Package the OTP and the public key in a single message, transmit accordingly and notify the API of the success or failure of the operation.

Data entities (OTP, key, target, etc.) are provided by the API which in turn acquires them from different sources (stores, interfaces, etc.).

3.52.2 Preconditions

3.52.3 Input, and from where

Sender Module	Message	Content
KGS-10	Send package	Target, Method, Data
Target	Response	Status

3.52.4 Output, and whereto

Status notification of the request to the API.

3.52.5 Relevant standards

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

3.53.1 Functionality

Data import abstraction represents the general way of imputing blocks of data into the system. Data blocks are identified as either final certificates or key pairs. Its main use is to acquire this data from the network but secondary scenarios could be constructed where the input is provided by the API (through the interface).

The main function of this module is the input of final certificates. The acquired certificates are stored in the certificate store, and the API module is notified of the success or failure of this operation.

The secondary function is the input of key pairs where the keys are generated centrally. The behaviour in this case is basically the same. The only difference is the store the keys are recorded in.

The request for data retrieval is initiated by the API, who also provides the necessary authentication tokens, if the request needs to be verified (OTP, PK, one-time URL, etc.).

3.53.2 Preconditions

3.53.3 Input, and from where

Sender Module	Message	Content
KGS-10	Get data	Method, Location, Auth tokens
Data provider (RA-10, central KGS-10, etc.)	Put data	Data block

3.53.4 Output, and whereto

The equivalents of the input messages.

3.53.5 Relevant standards

3.54 Module Number KGS-25: User Interface

3.54.1 Functionality

The interface of the user KGS system is meant to collect and present data from and to the user. The basic mode of operation is a "wizard" like approach, possibly having an extended and detailed mode for lower level users.

The integration with the API module is tighter than the one between the RA interface and the RA API, as the logic is less a client-server one.

The abstraction should be made at the level of the "public" methods of the API, in order to have the possibility to add or remove interface assistants, different organization of data, etc.

3.54.2 Preconditions

It is assumed that the user is authenticated, and thus has access to the key and certificate stores.

3.54.3 Input, and from where

The user provides the event based input.

3.54.4 Output, and whereto

All communication is conducted directly with the KGS API module, and reflects all its "public" methods.

3.54.5 Relevant standards

3.55 Module Number KGS-26: Key Pair Factory

3.55.1 Functionality

The Key Pair Factory KGS-26 produces upon request by the KGS-10 module key pairs. It utilizes the key generation Crypto Engine (KGS-21) module for generation of each key pair set related to one token and reports the created public key values back to KGS-21.

The KGS-26 module operation is initiated either manually through the admin console KGS-5 or automatically upon receiving a signal on pending jobs from the KGS-10 module.

The pending key generation jobs are presented to the KGS-26 module through a job database populated by the KGS-10 module. Each job entry indicates the parameters for the key generation through reference to a profile, and contains further information to be filled or modified by the KGS-26 module: public key values, job state.

The profile referenced for each job describes the number of key pairs to be produced, the purpose of each key pair, location on the token, key size and algorithm to be used. It is configured at the KGS-26 module and contains information on the abstraction level of the PKCS#11-API, i.e. it treats the token to be produced as a storage for certificates, key pairs, and further data objects[3].

KGS-26 processes each job in turn, forwarding a single key generation job to the Store Access Module KGS-14. The generated public key is obtained from a response from KGS-14 on each job and stored in the corresponding entry in the job database. The status of the job is updated accordingly[4].

After all pending jobs are processed, KGS-26 signals KGS-10 the completion of the batch.

3.55.2 Preconditions

3.55.3 Input, and from where

Sender Module	Message	Content
KGS-10	Key Generation Requested	Set Of Job-IDs
KGS-14	Key Generation Job completed	Result of Job, Set of Public Key Values for this job if no error

3.55.4 Output, and where to

Addressed Module	Message	Content
KGS-10	Key generation Completed	Set of Job-IDs
KGS-14	Key generation request	Profile content

[3] It may be as well a script that controls the operation of PKCS#11 module – scripting modules for PKCS#11 are available in open source.

[4] The job state will be something like "pending" (initial state), "keys generated", "certificates stored" (optional state), "error". The will be no finished state, as finished jobs are removed from the job DB.

3.55.5 Relevant standards

PKCS#11, PKCS#15

3.56 Module Number KGS-27: User KGS API

3.56.1 Functionality

The API module is the core that provides the functionality for the user KGS interface and renders public all the resources of underlying modules. Its purpose is much like the central KGS's API module but the functionality is somewhat different (no job control, optimized for user interaction, etc.).

The module is event based, with events coming in from the user interface. A multithreaded model should be imposed in order to obtain full asynchronous support for most operations. It is also passive in that no action is initiated without an explicit user request.

Functions:

1. Store access
2. Key generation
3. Access control
4. Key / OTP transmission
5. Certificate import

For function #1, access is provided to the key and certificate stores in order to give the possibility of exporting keys and certificates in different formats. One of the several possible export solutions is the PKCS#12 format that facilitates the transfer of certificates and keys between the user KGS and the browser. Several modules may need this export method and as such, a dedicated module for standard format enforcement should be considered.

Function #2 uses the key factory to generate key pairs and record them in the key store. The public key is sent to the CA together with the OTP in order to start the certificate creation process. Eventually, the private key may be bound to the final certificate in order to create a PKCS#12 ready to be imported into other consumers.

Function #3 is meant to restrict access to key and certificate stores. Basically, any operation that does not involve the stores should be granted, as the KGS application can be universally used. One such operation that does not require prior authentication is the "cold" generation of keys with immediate exportation.

Function #4 retrieves a previously acquired user OTP from the user interface, uses the public key generated as part of function #2 and transmits the final package containing the two to the CA in order to start the certificate creation process.

The ideal use of function #5 is in a user transparent manner, through "notification files", much like Windows Media link files (ASX, WMX) or WinAmp PLS files. The user receives such a file by mail or web, and the file that is associated with the user KGS application, launches the certificate retrieval procedure automatically, possibly asking

the user for a password or confirmation. The file should contain all data that is necessary to create the certificate retrieval request (location, method, etc.). Of course, this procedure should also be available through means of direct user interface input, where the user types in all coordinates required.

Certificate retrieval can be performed as part of the initial public key transmission, if the CA supports real-time certificate signing.

The base implementation of the API should provide a common mode of operation for all functions, with wizard-like features to be developed at a later time.

Common modes for function #2 are stand-alone key generation and key generation as part of function #4. For function #4, the basic use case is the transmission of an OTP/PK package, possibly followed immediately by a certificate retrieval procedure.

3.56.2 Preconditions

3.56.3 Input, and from where

The input is provided by the user according to each functionality. As the general model is event based, the data flow is basically a user request-response one. All input coming from other modules uses the API as a bridge towards either, the user interface, the network or a hard media.

The actual low level input is directly dependent on the interface and its navigation schema.

Internal inputs are keys coming from the stores or the key factory. Function #4 uses this input to construct the PK package sent to the CA.

Direct user inputs are the OTP and the parameters for key creation.

3.56.4 Output, and whereto

Output mainly follows the input data model with the main listeners being the user interface and module KGS-23.

3.56.5 Relevant standards

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

3.57.1 Functionality

The user KGS application may have two modes of operation, where certain components of the system are protected by a means of access control. The basic API functionality is universally available while the key and certificate stores are access restricted. Furthermore, individual keys and/or PKCS#12 entities may be pass phrase encrypted, with the access module providing the means to decrypt and use this type of data.

The module restricts access to the stores through a password system. Several authentication systems may be considered, if the module is shared with the CA and

the RA entities. Authentication may occur anywhere in the navigation schema, as required by the requests issued by the user.

The module generates a profile that is transported through the whole user session, providing one-time authentication for the whole system. The profile may even be passed along by the API to external compatible entities, thus transforming the profile into a general use "passport". This approach needs further analysis though.

3.57.2 Preconditions

3.57.3 Input, and from where

The API requests the authorisation to access a certain resource, given a required grant level. It passes the grant level and the resource, and asks for a valid profile.

3.57.4 Output, and whereto

The module returns either an error or a valid module containing the user data.

3.57.5 Relevant standards

3.58 Module Number KGS-29: Export PKCS#12

3.58.1 Functionality

The only function of this module is the export of final certificates.

PKCS#12 is a common format, accepted by Internet Explorer and Netscape families of browsers.

PKCS#12 contains both public and private keys, thus the user will be asked to choose a password to protect the private key. For security reasons the export password must be different from the master password.

3.58.2 Preconditions

Before exporting the certificate, the user has to be authenticated with the master password.

3.58.3 Input, and from where

Sender Module	**Message**	**Content**
KGS-28 Access control		
KGS-14 store access	Get keys	keys

3.58.4 Output, and whereto

Sender Module	**Message**	**Content**
KGS-25 User interface		

3.58.5 Relevant standards

PKCS#12

3.59 Future Aspects (Enhancements, Extensions, Capacity Reserves)

The WP4.2 and WP4.3 deliverable documents will translate the WP4.1 architecture specification into detailed prescriptions.

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.