ORIGINATOR:	**ISSUE DATE:**	**VERSION:**
ATOS ORIGIN	05-May-2006	EN12.00

TAXATION AND CUSTOMS UNION DG

CCN/CSI Project

SUBJECT:

CCN/CSI SYSTEM OVERVIEW

CCN-COVW-GEN

FRAMEWORK CONTRACT TAXUD/00/C063
SPECIFIC CONTRACT 18

DOCUMENT HISTORY

Edi.	Rev.	Date	Description	Action(*)	Paragraphs
0	00	16-Sep-1996	Creation	I	All
1	00	15-Oct-1996	Quality Check	R	All
2	00	02-Jan-1997	Typographic + additional information	R, I	§6 Glossary
			Typographic + administration functional description		§4.3.1
			New paragraph about particular architectures functional description		§4.3.4
			Administration architectural description, script entity definition		§4.4.3
			(Corrections related to JP037V03.DOC, OR 39 and CCN/CSI-FR-001-DGXXI)		§5.2.1
3	00	14-Jan-1997	Corrections related to QCR011L2.DOC	R	All
4	00	30-Jun-1997	Corrections related to Change Request 15, 9, 19 , 22	R	All
5	00	31-Aug-1997	Corrections related to QAM026L2	R	All
6	00	28-Apr-1998	Upgrade for Project Phase 2	R	All
7	00	07-Aug-1998	Taking into account QAM64	R	All

Edi.	Rev.	Date	Description	Action(*)	Paragraphs
8	00	14-Jul-2000	Modifications related to CCN/TC SC 2564: - DG XXI replaced by DG TAXUD; - MSA replaced by NA; - suppression of some obsolete particular architectures (Low Speed Link, P1 Tunneling, Queue Replication and Complementary Platform); - Phase 1 and Phase 2 of the specification and implementation steps are completed, CCN/CSI is now operational	R	All
9	00	02-Aug-2000	Official version	/	None
10	00	29-Aug-2000	Modifications related to WR30/WA30: Suppression of the Transfer Agents	R	§ 4.4.3.7, § 4.4.3.9
11	00	24-Jan-2001	Official version	/	None
11	01	06-Feb-2006	RFA147: Technical Documentation Update	R	§1.1, §2, §4.2.2.1, §4.3.1, §4.3.2, §4.3.3, §4.4.3, §5.2.1, §old 5.2.1.5 suppressed, §5.2.2, §6
11	02	07-feb-2006	Internal Quality Review		All
11	10	07-feb-2006	Official Version Delivered for Review		None
11	11	22-Feb-2006	Taken into account DRF_RFA147_CCN-COVW-GEN-EN11.10_CCO		§1.1, §4.2.2.1, §4.3.1.1.4, §4.3.1.1.5, §4.4.1, §4.4.3.1, §4.4.3.2.3, §4.4.3.4, §4.4.3.6, §4.4.3.7.1, §4.4.3.7.2, §0, §5.1, §5.1.3, §5.2.2, §5.2.2.3, §6
11	20	03-Mar-2006	Official Version Delivered for Review		None

Edi.	Rev.	Date	Description	Action(*)	Paragraphs
11	21	07-Apr-2006	Taken into account DRF_RFA147_CCN-COVW-GEN-EN11.20_CCO		§4.2.2.1, §4.4.3.1, §6
11	30	10-Apr-2006	Official Version Delivered for Review		None
12	00	05-May-2006	Official Version Delivered for Acceptance		None

(*) ACTION: I=INSERT R=REPLACE

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 PROJECT BACKGROUND

The Common Communication Network and Common System Interface (CCN/CSI) project was decided in late 1992 by the Joint meeting of General Directors of Customs and Indirect Taxation. One of the objectives is to assist in the resolution of problems of support and maintenance in several heterogeneous systems with multiple types of hardware, software and communications equipment. Another objective is to provide a coherent approach toward the development of new systems.

The CCN/CSI provides an infrastructure for the development of applications for DG TAXUD and National Administrations (NA) in the sector of customs and indirect taxation. Those National Administrations can be located inside the European Union (i.e. the Member States Administrations (MSA)) or outside the European Union. This infrastructure allows a harmonised approach to system development, in conformity with the regulatory constraints of the Community in the field of data transmission.

The Common Communication Network and Common System Interface are designated by two terms: CCN and CSI.

The Common Communication Network (CCN) is made up of a series of physical gateways (GW) located either in the National Administrations or on the DG TAXUD premises. These gateways are interconnected through their own communication services, and communicate with the Application Platforms (AP).

The Common System Interface (CSI) is a set of protocols and application programming interfaces allowing the applications to exchange information through the CCN. It ensures the interoperability between the relevant application platforms that belong to any NA and a gateway provided by DG TAXUD.

In addition to the CSI, the CCN/CSI infrastructure also supports Intranet exchanges via the HyperText Transfer Protocol (HTTP) as well as mail exchanges via the Simple Mail Transfer Protocol (SMTP), the Internet Mail Access Protocol (IMAP) and the Post Office Protocol version 3 (POP3). Mail exchanges are supported thanks to dedicated Local CCN Mail Servers (LCMS).

1.2 PURPOSE OF THE DOCUMENT

This document gives an overview of the CCN/CSI system to be provided by the software installed on the Application Platforms and the Gateways.

Chapter 4: presents the general requirements and objectives, the Functional Subsystems composing the CCN/CSI system and the CCN/CSI architecture.

Chapter 5: contains the functional entity relationship model underlying the CCN/CSI system.

Chapter 6: provides a definition of technical terms, product names, standards, abbreviations and acronyms used by the CCN/CSI project. This is the project glossary.

For further details or information concerning CCN/CSI, please refer to the applicable and reference documents listed in chapter 2.

1.3 FIELD OF APPLICATION

The business objective of the CCN/CSI project is to:

- Specify, implement, deploy and operate CCN in all the gateways of the infrastructure.
- Specify, implement and deploy CSI in all the application platforms connected to CCN.
- Specify and integrate the LCMS within the National Administrations.

The CCN/CSI system overview applies on the specification and the implementation of the CCN/CSI system.

2. DOCUMENTS

2.1 APPLICABLE DOCUMENTS

Id	*Reference*	*Title*	*Version*
[AD1]	DG TAXUD/A3/PHT/hc-D(2004) 62615	Request for Action (RFA) n°147	

2.2 REFERENCE DOCUMENTS

Id	*Reference*	*Title*	*Version*
[RD1]	CCN-*-PH4 (set of documents)	Feasibility Study	/
[RD2]	CCN-CFRS-GEN	CCN/CSI Requirement Specifications	EN4.00
[RD3]	CCN-CFSS-GEN	CCN/CSI Functional System Specifications	EN8.00
[RD4]	CCN-CAD-GEN	CCN/CSI Architecture Design	EN8.00

3. TERMINOLOGY

See Annex: Glossary in order to have the definitions of technical terms, abbreviations and acronyms used in the framework of the CCN/CSI project.

4. CCN/CSI PRESENTATION

4.1 INTRODUCTION

This chapter is divided into three parts:

- The first part describes the CCN/CSI general objectives.
- The second part describes the CCN/CSI system in a functional way. It starts by a presentation of the CCN/CSI system from the point of view of the CCN/CSI applications. Then, the Functional Subsystems composing the CCN/CSI system are presented.
- The third part presents the CCN/CSI system in an architectural way. The general architecture and the various Software Products composing the CCN/CSI system are presented.

4.2 GENERAL OBJECTIVES

4.2.1 INTRODUCTION

The general objectives of the CCN/CSI system can be summarised as follows:

- To offer all National Administrations a coherent method of access to all DG TAXUD applications.
- To offer all National Administrations a coherent method of access to other National Administration applications.
- To provide a high level of service on an equal basis to all National Administrations.
- To allow for the integration of other Commission entities hence extending the coherence of access to non-DG TAXUD applications.
- To provide a solution that is not dedicated to a particular application (or even several applications) but rather a general purpose solution which will be valid for a long period of time.

The technical objectives are:

- To provide a robust and standardised backbone.
- The backbone provides to the applications CSI synchronous and asynchronous services, web services and mail facilities in an integrated manner.
- To have a high quality administration system so as to offer a high Quality of Service. This robust administration of the community domain adheres to the subsidiarity principle, allowing the National Administrations to benefit from a local administration for their national domain.
- To provide a consistent API on the Application Platforms of all the National Administrations.
- To provide state of the art technology services and widely accepted API.

4.2.2 SERVICES OFFERED

The services are divided into two parts:

- Services offered to the applications of the National Administrations.
- Services offered for the management of the CCN system.

4.2.2.1 SERVICES OFFERED TO THE APPLICATIONS

The proposed services allow the development of applications using synchronous, asynchronous or web interactions on the different hosts of the National Administrations.

The CSI API offered to the application developers are based on a XATMI-like interface for the synchronous exchanges[1], on a MQI-like interface for the asynchronous exchanges[2], on HTTP for web exchanges and on SMTP, IMAP and POP3 for mail exchanges.

The CSI API is implemented through remote interface technique based on a RPC mechanism and includes security features. Integration of web exchanges and mail exchanges on CCN/CSI relies on standard protocols and also provide security features.

The services offered can be summarised as follows:

- Synchronous service: an application can invoke another application in a request/response mode; with the restriction that CCN/CSI does not offer distributed transaction support.
- Asynchronous service: an application can exchange messages with other applications in an asynchronous manner (store and forward technique). In this case, the applications see each other via message queues.
- Web service: an application can exchange messages with other applications in a synchronous way using the HTTP protocol.
- Mail service: a user (or an application) can exchange mail messages with other users (or applications) in an asynchronous way using the STMP protocol.
- Interoperability: National Administrations applications run on different Application Platforms, each one with its own data format. The infrastructure handles all the necessary conversions on behalf of applications, which deal only with their own native format.

---

[1] For example, the XATMI *tpcall* verb is mapped on to the *HL_call* verb of the High Level CSI API (cf. Figure 19)

[2] For example, the MQI *MQPUT* verb is mapped on to the *HL_mq_put* verb of the High Level CSI API (cf. Figure 21).

- The exchanges between National Administrations Application Platforms are secured. Two levels of security are provided:
  - At the Community level (security devices are present between each CCN/CSI Gateway and the CCN/CSI Backbone, so that the CCN/CSI Gateways are in a DMZ).
  - At the CCN/CSI access point (links between the Application Platforms and the Gateways) where authentication, confidentiality and integrity mechanisms are provided.

4.2.2.2 SERVICES OFFERED TO THE SYSTEM MANAGEMENT

The management services provided for CCN/CSI are applied according to Service Level Agreement (SLA) procedures defined between DG TAXUD and the CCN Technical Centre (CCN/TC), which is in charge of the central administration. These procedures define the actions and the responsibilities of the National Administrations and the CCN Technical Centre.

4.3 FUNCTIONAL OVERVIEW

4.3.1 CCN/CSI SYSTEM OFFERED TO APPLICATIONS

The applications that are to be implemented over the CCN/CSI infrastructure can be grouped in a domain called Application Domain (see the Figure 1 below).

The services offered by the CCN/CSI infrastructure to the applications can be seen as an Inter-Application Bus entity (IAB).

In the IAB model, subject to the appropriate access controls, any application can call any other application accessible through CCN/CSI, just by knowing a logical name (queue, synchronous service, web service or functional mailbox). The CCN/CSI infrastructure takes all required operations to carry the messages, in particular the routing and security functions.

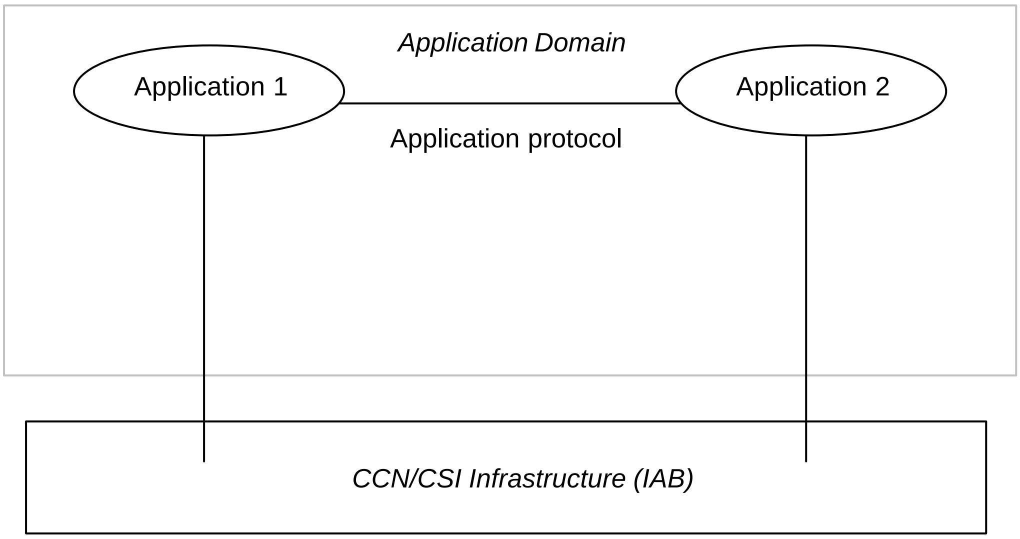


FIGURE 1: APPLICATIONS AND CCN/CSI INFRASTRUCTURE.

The applications communicate by means of an application protocol. This dialogue is made possible by a set of services (residing in the Community domain) provided by the CCN/CSI infrastructure that can be accessed in various National Administrations sites.

The applications call services provided by the CCN/CSI infrastructure or by the remote applications. The applications can issue requests/responses associated with the called services including Quality of Service requirements (QoS).

The services offered by CCN/CSI concern the telecommunication aspect of applications that means everything necessary to communicate with a remote site only known through a service identifier. This includes:

- Interaction functions, with all possible inflections, organised around several paradigms.
- Security features, namely authentication, integrity and confidentiality.
- Quality management of interactions, summarising other features provided by CCN/CSI.

4.3.1.1 TYPOLOGY OF INTERACTIONS

CCN/CSI provides five usable types of interactions between applications:

- CSI blocking Request/Response
- CSI non-blocking Request/Response
- HTTP interaction
- CSI message queuing
- Mail exchange

The first three types of interactions correspond to synchronous transmission modes. In synchronous transmission modes, all the entities participating in the transmission of a message are physically connected between the time the message is submitted by the requester application to the CCN/CSI infrastructure and the time the message is read by the responder application.

The last types of interaction correspond to an asynchronous transmission mode. In asynchronous transmission mode, applications need not be active at the same time. There is no direct connection between the application processes.

4.3.1.1.1 CSI BLOCKING REQUEST/RESPONSE TYPE

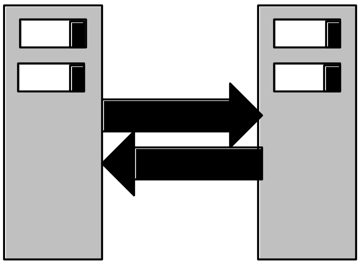


FIGURE 2: CSI BLOCKING REQUEST/RESPONSE TYPE OF INTERACTION.

The CSI blocking Request/Response type of interaction corresponds to a request/response synchronous mode of transmission. In this mode, there is only one send-receive exchange. This type is blocking due to the fact that the requester application waits for the reply to a request before continuing its processing.

The main characteristics of this type of interaction are:

- One request, one response.
- No delay between request and reply.
- Strong synchronisation.
- The context does not have to be maintained by the application explicitly.

4.3.1.1.2 CSI NON-BLOCKING REQUEST/RESPONSE TYPE

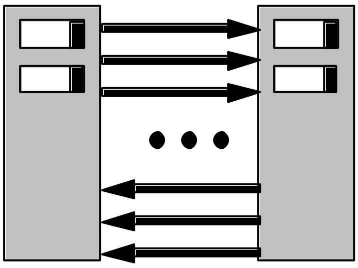


FIGURE 3: CSI NON-BLOCKING REQUEST/RESPONSE TYPE OF INTERACTION.

The CSI non-blocking Request/Response type of interaction also corresponds to a request/response synchronous mode of transmission. In this mode, there is only also one send-receive exchange. This type is non-blocking due to the fact that the requester application does not wait for the reply to a request before continuing its processing.

The main characteristics of this type of interaction are:

- Several requests pending at the same time
- Short delay between request and reply
- Relieved synchronisation, however, a connection still necessary
- Application maintains a context for each request/reply

4.3.1.1.3 HTTP INTERACTION TYPE

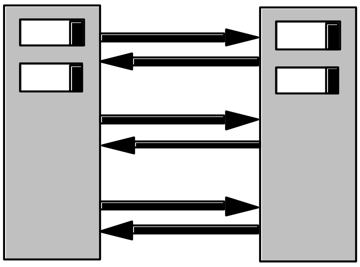


FIGURE 4: HTTP TYPE OF INTERACTION.

The HTTP type of interaction corresponds to an interaction based on the HyperText Transfer Protocol. This type is blocking due to the fact that the requester application waits for the reply to a request before continuing its processing.

The main characteristics of this type of interaction are:

- One request, one response, but one request can lead to other HTTP interactions (to include all sub-elements of the original request).
- No delay between request and reply.
- Strong synchronisation.
- Stateless interaction.

- The context does not have to be maintained by the application explicitly.

4.3.1.1.4 CSI MESSAGE QUEUING TYPE

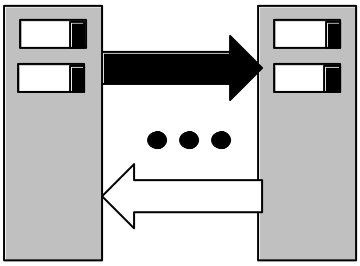


FIGURE 5: CSI MESSAGE QUEUING TYPE OF INTERACTION

The CSI message queuing type of interaction between applications is an asynchronous mode of transmission. The application works on queues where it can put or get messages.

The main characteristics of this type of interaction are:

- No answer waited-on, an answer may come later.
- No synchronisation.
- No context to maintain with regard to the execution of the application.

4.3.1.1.5 MAIL EXCHANGE TYPE

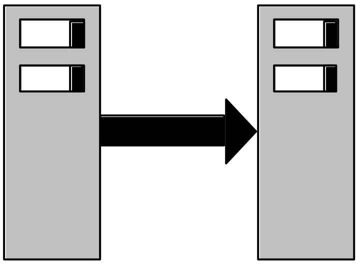


FIGURE 6: MAIL EXCHANGE TYPE OF INTERACTION

The mail exchange type of interaction between applications is an asynchronous mode of transmission. The application works on mailboxes where it can put or get messages.

The main characteristics of this type of interaction are:

- No answer waited-on, an answer may come later.
- No synchronisation.
- No context to maintain with regard to the execution of the application.

4.3.1.2 SECURITY

The security of CCN/CSI is organised according to the underlying IDA architecture, in particular to the EuroDomain and Local Domain principles. So, there are two kinds of security responsibility domains:

The Community security domain includes only systematic, non-negotiable services: mutual authentication of gateways, access control lists (user / user profile / resource) recorded on the gateways and a hardware-based encryption of all trans-European communications.

The National security domains include an authentication of the user and application to the gateway in two possible contexts:

- Secure links, meaning well protected, communication lines from an application platform having its own security subsystem may be considered as secure enough to have a direct access to the gateway.
- In other cases, the link is said to be non-secure, and additional authentication features such as three-way authentication are offered.

4.3.1.3 ADMINISTRATION

The Administration Software of the CCN/CSI Gateway provides a set of tools allowing the administrators to configure and manage the CCN Gateway software. It provides the administration functions covering the following functional areas:

- Configuration and Name Management, which allows the administrators to manage the configuration and naming of the various software products and components of the CCN Gateway.
- Fault Management, which provides error detection and reporting mechanisms to facilitate troubleshooting.
- Accounting and Performance Management, which provides functions to collect statistics information for further analysis by external applications (e.g. generation of reports regarding CCN service availability, message transit delay, backbone usage, Gateway management).
- Security Management, which provides functions to the administrators to configure and manage security information, such as user authentication keys, user profiles and Access Control Lists (ACL). Security management also provides security logging mechanism.
- Control and Monitoring, which allows the administrators to control and monitor the operation of the CCN Gateway software. Control and monitoring functions include start-up, stop and restart of the CCN Gateway software, and supervision of the running CCN/CSI software processes to detect possible unexpected fallen processes.

Most of the common configuration information is stored in the CCN Directory, which allows the Administration Software to provide auto-configuration facilities, to make use of the replication mechanism provided by the CCN Directory to synchronise configuration information and to simplify configuration management.

The Administration Software provides the technical means to allow the administrators of the National Administrations to locally administer configure and manage the CCN Gateway software. It provides also facilities to allow the Central Administrator from the CCN Technical Centre to perform centralised remote administration.

Although the administrative functions described in this specification are technically available both to the local and to the central administrators, in practice, they are assigned to either the local, the central or both types of administrators. This division and sharing of administrative responsibilities is based on the Subsidiarity Principle and the IDA rules, and are determined through common agreements between the NA and the CCN/CSI service owner (DG TAXUD), then enforced by the CCN/TC, in accordance with the SLA.

Due to technical constraints, some administrative functions are only available to the local NA administrators, e.g. backup configuration information on physical media. This is pointed out in the relevant functions.

Note that when an administrator performs a task, another administrator must not doubly perform it.

4.3.1.4 QUALITY OF SERVICE

Quality of Service is a set of indicators that specify a set of services requested from the CCN service; it applies to a message.

4.3.2 CCN/CSI SUBSYSTEMS

The applications communicate with each other through the IAB. The accesses to the QoS management are carried out by the IAB. The IAB constitutes the CCN/CSI system. It is split into several subsystems as presented in the figure below.

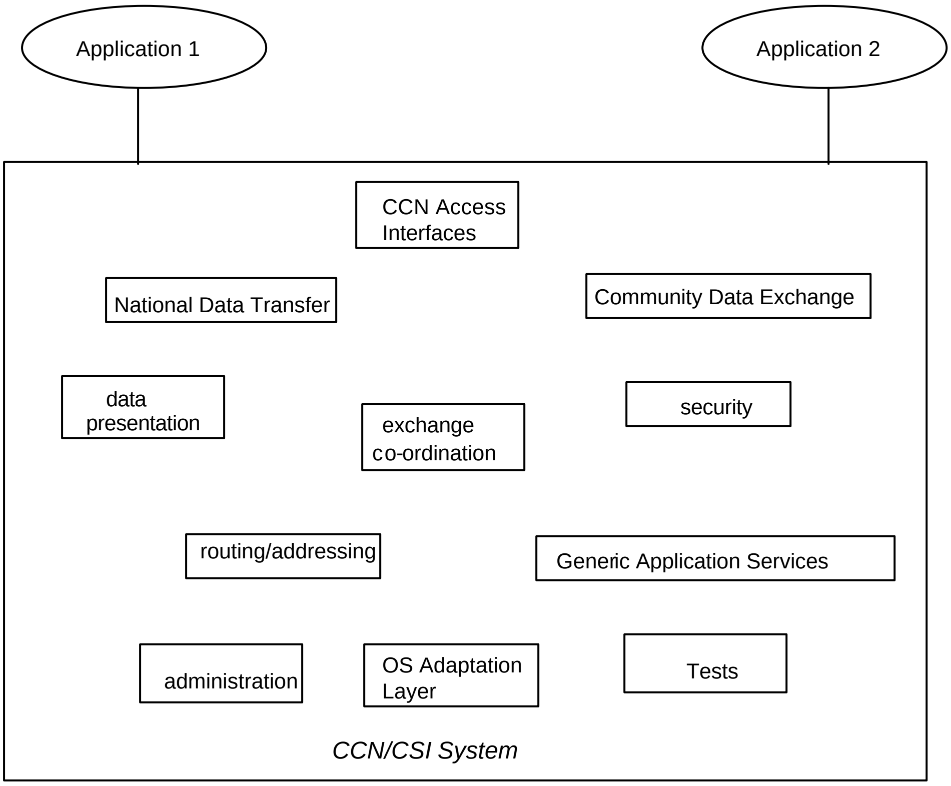


FIGURE 7: CCN/CSI FUNCTIONAL SUBSYSTEMS.

The Functional Subsystems composing the CCN/CSI system are as follows:

- CCN Access Interfaces, this subsystem is the entry point offered to the applications in order to access and to use the CCN/CSI system. In the generic architecture described in Chapter 4.4, this subsystem is the access point for all information entering or leaving the CCN/CSI system. It is distributed on the Application Platform and the CCN/CSI Gateway. It includes the HL_API, the CSI_API and the Function layer described in the next paragraph "Functional Architecture Model". The SPI_API is included in this subsystem as an internal entry point in the gateway though it cannot be used by the applications. Concerning the HTTP interactions, the API provided to the National Administrations relies directly on specific handlers run by the HTTP server. For mail exchanges, the CCN Access Interfaces are in fact SMTP, IMAP and POP3 [3].

---

[3] Concerning the SMTP, IMAP and POP3 protocols, a webmail interface is also offered for convenient purpose.

- National Data Transfer, this subsystem provides the transfer means used to exchange information between the Application platforms and the related CCN/CSI gateway. It is distributed on the Application Platform and the CCN/CSI Gateway. It includes the T_API, the GT layer, the CT_API and the CT layer described in the next paragraph "Functional Architecture Model".
  For HTTP and mail exchanges, TCP/IP takes the transfer between the Application Platforms and the CCN backbone in charge.

- Community Data Exchange, this subsystem is responsible for the data transfer between two CCN/CSI gateways. It includes the Function layer in the Community Domain described in the next paragraph "Functional Architecture Model". The Function and Transmission Layer role is to convey the CCN/CSI messages. The Communication layer of the Community Domain is based on the following underlying products: Tuxedo, MQSeries (accessed through the GT_API), Apache (HTTP protocol) and Postfix (SMTP protocol).

- Exchange Co-ordination, this subsystem allows the scheduling of the services offered by other subsystems (Data Presentation, Security, and Routing/Addressing) in order to ensure their activation in a consistent way. It takes charge of the Quality of Service required by the Applications. This subsystem is a mandatory access point for all information to be handled within the system.

- Security, this subsystem provides the security services, including access control, authentication, integrity and confidentiality. It provides the GSS_API to the Applications in order to access to the security services. It is distributed on the Application Platform and on the CCN/CSI Gateway.
  For HTTP exchanges it makes use of SSL, especially during the authentication phase.

- Generic Application Services, this subsystem includes all the CCN/CSI services running directly on the CCN gateways.
  For synchronous and asynchronous services, these applications are relying on the same API as applications running on Applications Platforms. The only difference is located in the authentication mechanism, which is replaced by an identification mechanism.
  For web services, these applications are implemented via a specific content-handler integrated in the Apache HTTP server running on the gateway.

- Routing/Addressing, this subsystem is responsible for the routing/addressing resolution of the messages sent by the applications to their destination. Its goal is to associate a resource logical name to a final destination, assuring the coherence regarding the mode of the partners and the gateways they are running on.

- Data Presentation, this subsystem provides the means to convert the data exchanged between heterogeneous application platforms, to and from a common format used by the IAB.

- Administration, this subsystem allows the administration of the subsystems that make the CCN/CSI system.

- Tests, this subsystem provide the means to perform application environment tests with the CCN/CSI system.

- OS Adaptation Layer, this subsystem provides, for portability purpose, services to the Application Platforms subsystems permitting to mask the dependencies of system dependent operations.

Note: The Directory is not defined as a subsystem. It is related to the architecture and is a way to implement the configuration information storage on the CCN/CSI gateway. Nevertheless, the requirements to be met regarding the Directory are part of the Functional Requirement Specification ([RD2]).

4.3.3 FUNCTIONAL ARCHITECTURE MODEL

This section presents the functional architecture model that is used, in the Functional System Specification document ([RD3]) as a reference to locate the described functions. It takes into account the IDA recommendations and the approach defined in the CCN/CSI feasibility study ([RD1]).

The functional architecture is based, for each National Administration, on the distribution of the CCN/CSI subsystems on the CCN/CSI gateway (Eurogate) and on the National Administrations Application Platforms.

4.3.3.1 FUNCTIONAL ARCHITECTURE MODEL – APPLICATION PLATFORMS

For synchronous Request-Response and asynchronous modes, the Application Platforms where the CCN/CSI Functional Subsystems are distributed make use of several layers as follows:

- National Communication layer (NA communication), it contains all the communication functions providing the adequate standard protocols for a given platform. This functional entity dialogues with a peer entity present on the gateway. It provides services by means of the Communication API (C_API).

- National Transmission layer (NA transmission), it provides a consistent functional interface whatever the communication means capabilities. This functional entity dialogues with a peer entity present on the gateway. It is divided into two parts. The first part is the Communication dependent Transmission layer (CT), which isolates the upper part of the Transmission layer of the system dependencies regarding the Communications layer interface. The CT layer offers the CT_API to the Generic Transmission layer (GT), which is the second part of the Transmission layer. The GT layer offers mechanisms to implement remote procedure calls. The GT layer provides the T_API to the Function layer.

- National Function layer (NA function), it participates in the identified functions within the framework of the dialogue with the gateway. For example, the encoding of a QoS requested by an application or the encoding of data in case this latter has to be interpreted at the gateway level. The QoS shall be conveyed through the FTC layers in order to make all pertinent entities aware of its value. The Function layer provides the HL_API to the NA applications as well as the CSI_API. The HL_API provides a high level API, which masks to the application the use of the GSS_API (provided by the security subsystem), the use of the PRES_API (provided by the Data Presentation subsystem) and the use of the CSI_API (provided by the CCN Access Interface subsystem).

For the HTTP interactions, the CCN/CSI Functional Subsystems on the Application Platforms can be described as follows:

- National Communication layer (NA communication), no specific CCN/CSI software is needed to fulfil this role. The TCP/IP layer directly provides the services.

- National Transmission layer (NA transmission) & National Function layer (NA function), these two roles are filled by the use of a HTTP client implementing the HyperText Transfer Protocol (HTTP). This protocol is generic, stateless and implements the feature of typing and data presentation, allowing messages exchanges independently of the platform issuing the request.

For the mail exchanges, the CCN/CSI Functional Subsystems on the Application Platforms can be described as follows:

- National Communication layer (NA communication), no specific CCN/CSI software is needed to fulfil this role. The TCP/IP layer directly provides the services.
- National Transmission layer (NA transmission) & National Function layer (NA function), these two roles are filled by the use of a client compliant with the SMTP, IMAP and POP3 protocols.

4.3.3.2 FUNCTIONAL ARCHITECTURE MODEL – CCN/CSI GATEWAYS

For synchronous Request-Response and asynchronous exchanges, the CCN/CSI gateway where the CCN/CSI subsystems or functions are distributed is divided into several layers as follows:

- National Communication layer (NA communication).
- National Transmission layer (NA transmission).
- National Function layer (NA function).
  The three above layers are the peer entities of the NA Communication, Transmission and Function layers present on the Application Platforms. The peer entity of the CSI_API is the SPI_API whose role is to implement the remote verbs.
- Community Communication layer (CO communication).
- Community Transmission layer (CO transmission).
- Community Function layer (CO function).
  The three above layers provide the necessary functions permitting the exchange of information between the CCN gateways.
- Exchange coordination takes charge of the activation in a consistent way of the different functional model parts.
- Administration functions, allow the administration of the previously listed parts.

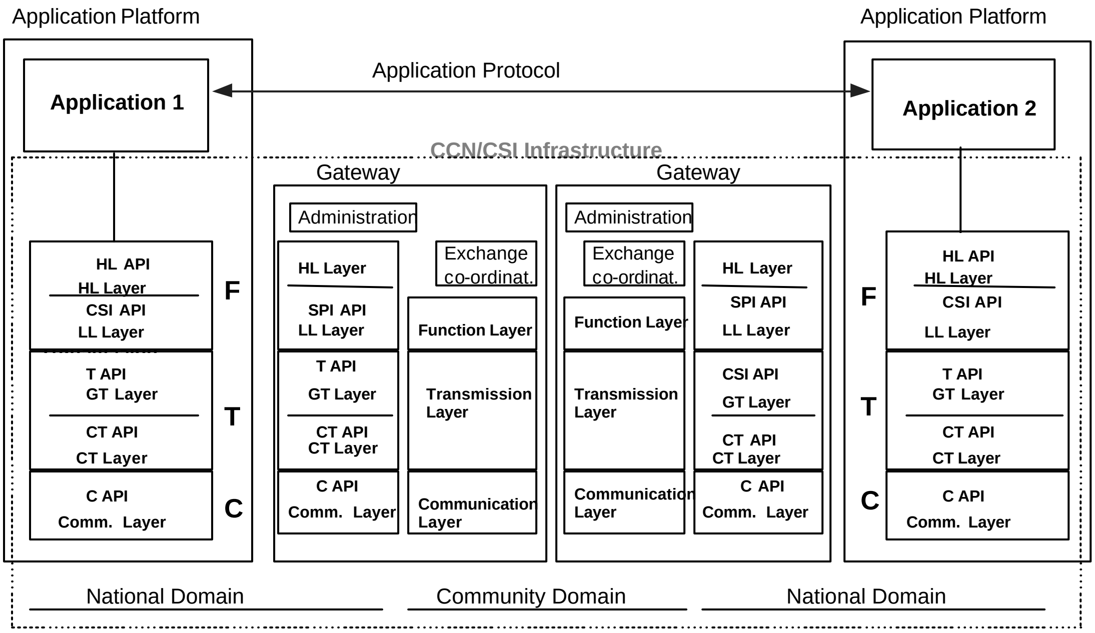


FIGURE 8: FUNCTIONAL ARCHITECTURE MODEL (1).

For the HTTP interactions, the CCN/CSI gateway where the CCN/CSI subsystems or functions are distributed is divided into several layers as follows:

- The National Communication layer (NA communication) and the Community Communication layer (CO communication) are peer entities relying directly on TCP/IP.
- The National Transmission layer (NA transmission) and the National Function layer (NA function), the Community Transmission layer (CO transmission) and the Community Function layer (CO function) are corresponding entities which functionalities are encapsulated in the HTTP protocol.
- Exchange coordination takes charge of the activation in a consistent way of the different functional model parts.
- Administration functions, allow the administration of the previously listed parts.

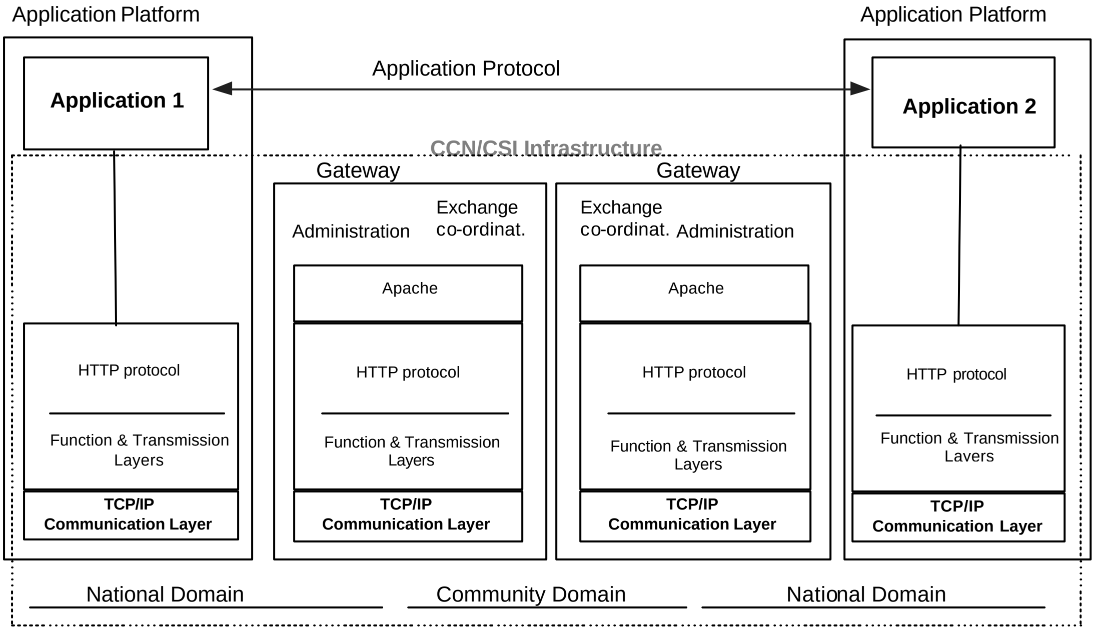


FIGURE 9: FUNCTIONAL ARCHITECTURE MODEL (2).

For mail exchanges, the CCN/CSI gateway[4] where the CCN/CSI subsystems or functions are distributed is divided into several layers as follows:

- The National Communication layer (NA communication) and the Community Communication layer (CO communication) are peer entities relying directly on TCP/IP.
- The National Transmission layer (NA transmission) and the National Function layer (NA function) are based on the SMTP, IMAP and POP3 protocols).

---

[4] Mail exchanges are supported thanks to dedicated gateways called “Local CCN Mail Servers” (LCMS).

- The Community Transmission layer (CO transmission) and the Community Function layer (CO function) are corresponding entities which functionalities are encapsulated in the SMTP protocol.
- Exchange co-ordination, takes charge of the activation in a consistent way of the different functional model parts.
- Administration functions, allow the administration of the previously listed parts.

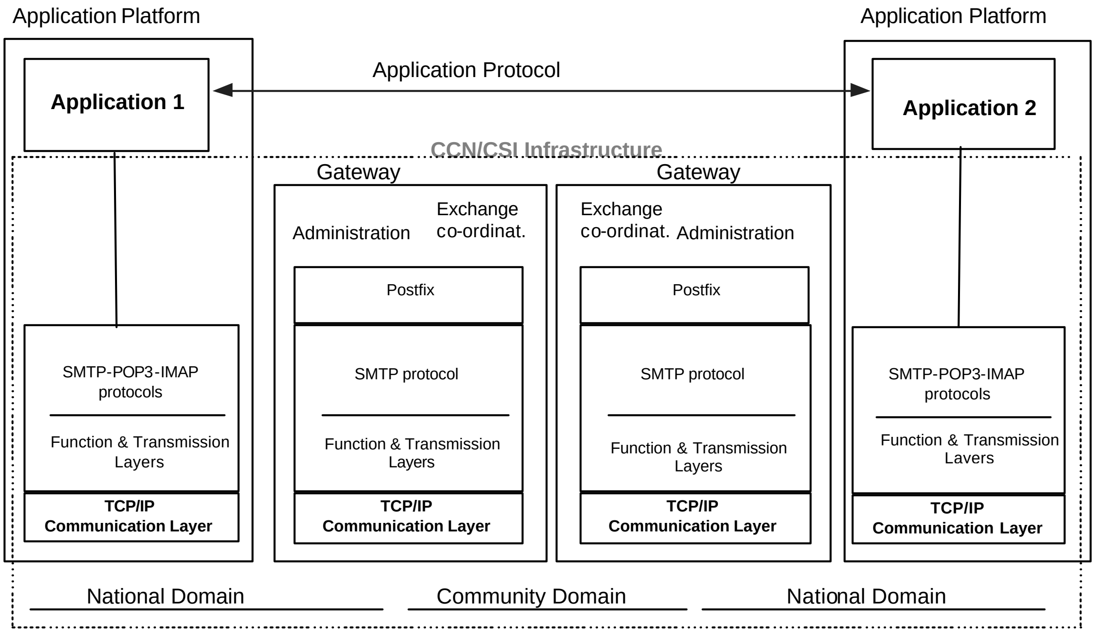


FIGURE 10: FUNCTIONAL ARCHITECTURE MODEL (3).

4.4 ARCHITECTURE OVERVIEW

4.4.1 INTRODUCTION

The following Figure 11 gives an overview of the CCN in terms of networks and platforms.

Note: In the present document, the term "platform" is used to reference the trio made up of computer hardware, the operating system it runs and possibly the transactional monitor. Thus, the computer hardware running two different operating systems (and vice versa) are considered as different platforms.

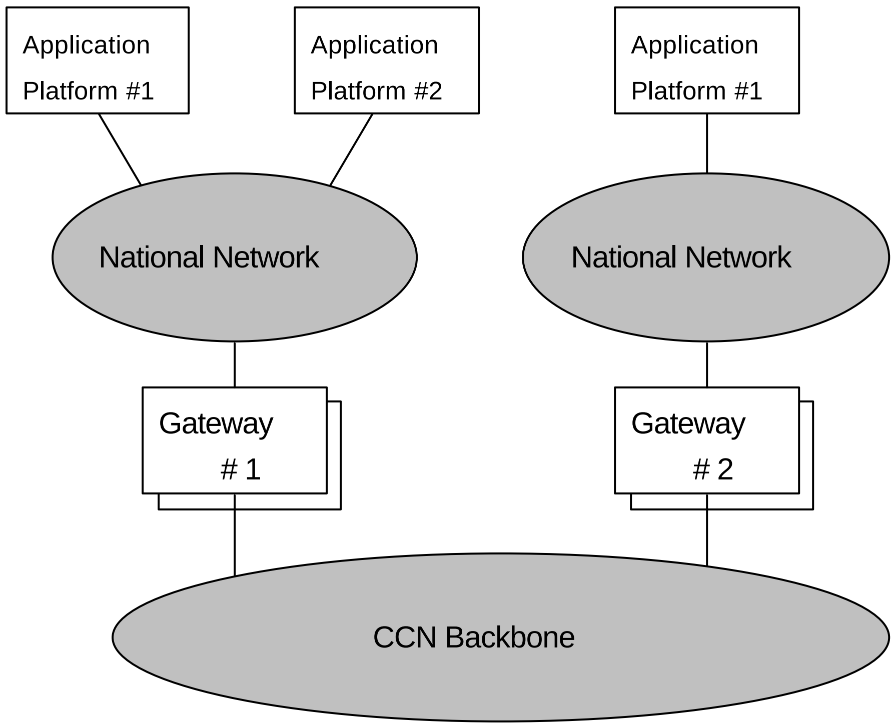


FIGURE 11: CCN OVERVIEW.

The national networks are owned and administrated by the National Administrations. The distributed trans-European applications run on the Application Platforms and dialogue through the CCN backbone.

The CCN Gateways are the only access points to the CCN. They offer the CSI for both the remote (Application Platform located) and local (Gateway located) applications. As explained in §5.2.1.1, the local (Gateway located) applications are generic applications services (GAS). As depicted on the figure, to provide a better service availability, each CCN Gateway is backed up.

An Application Platform (AP) is a platform (see above) running CCN/CSI applications.

The following figures, Figure 12 and Figure 13, zoom in on the CCN/CSI application and Gateway. These two components are detailed hereinafter.

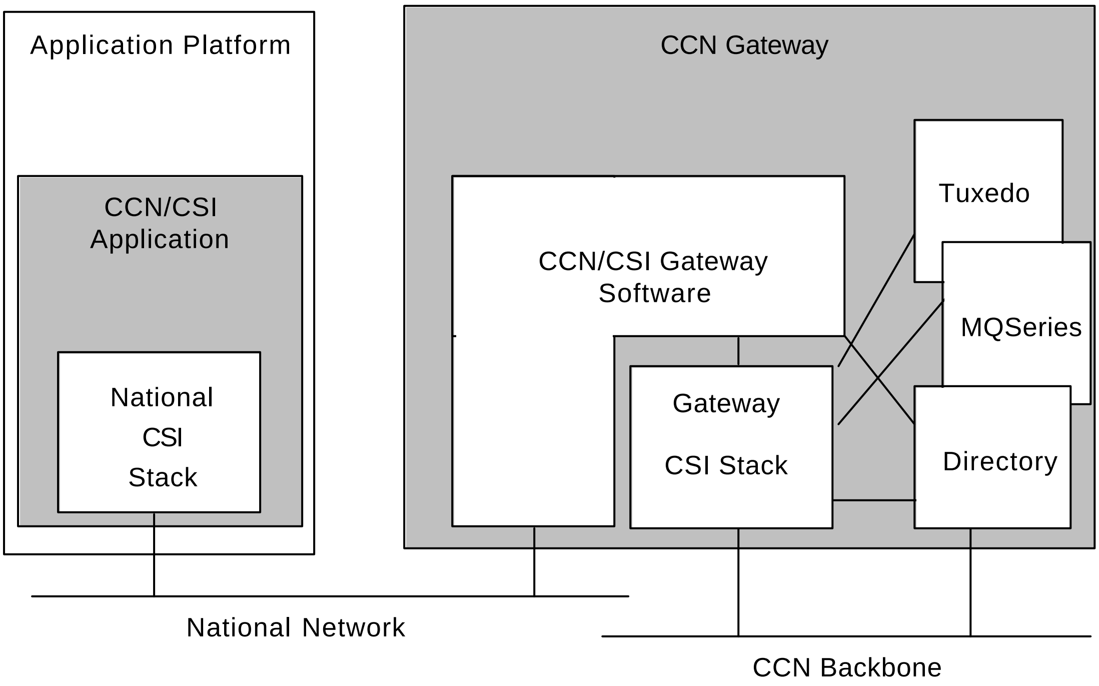


FIGURE 12: CCN/CSI APPLICATIONS AND GATEWAYS ARCHITECTURES FOR REQUEST-RESPONSE AND ASYNCHRONOUS INTERACTIONS.

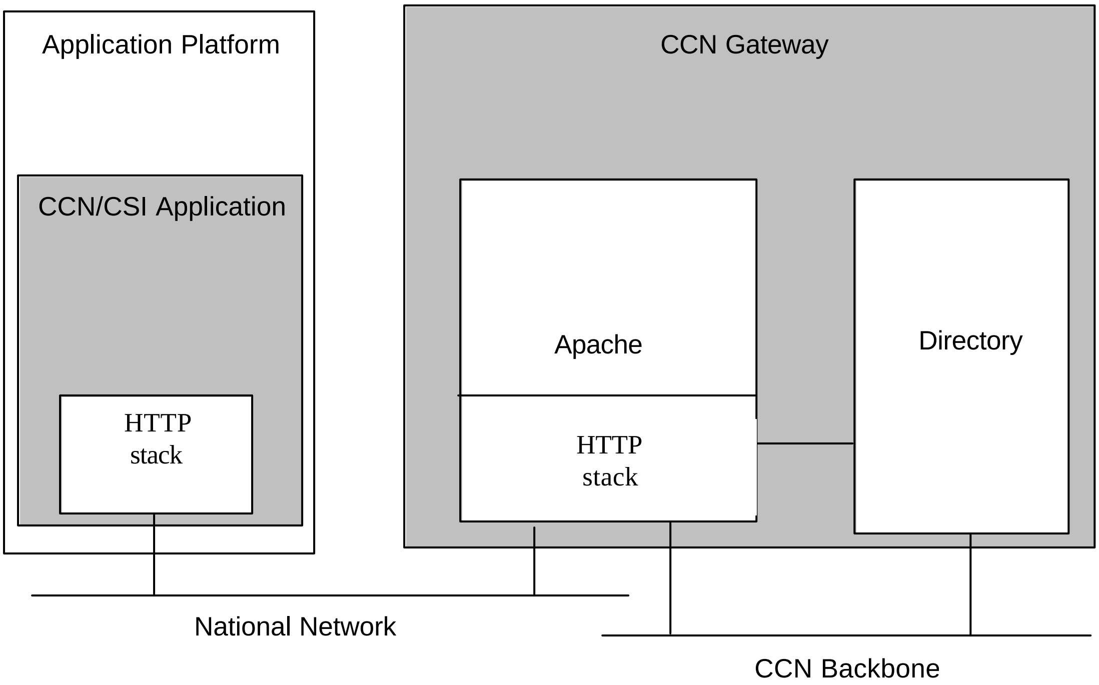


FIGURE 13: CCN/CSI APPLICATIONS AND GATEWAYS ARCHITECTURES FOR HTTP INTERACTIONS.

Note: The two above figures do not show the possibility of having CCN/CSI Generic Application Services (GAS) running directly on the CCN Gateway as the architecture of such applications does not differ from Application Platform-located applications: only the CSI stack is changed as the access to the CCN/CSI Gateway Software is then local and does not require any networked access.

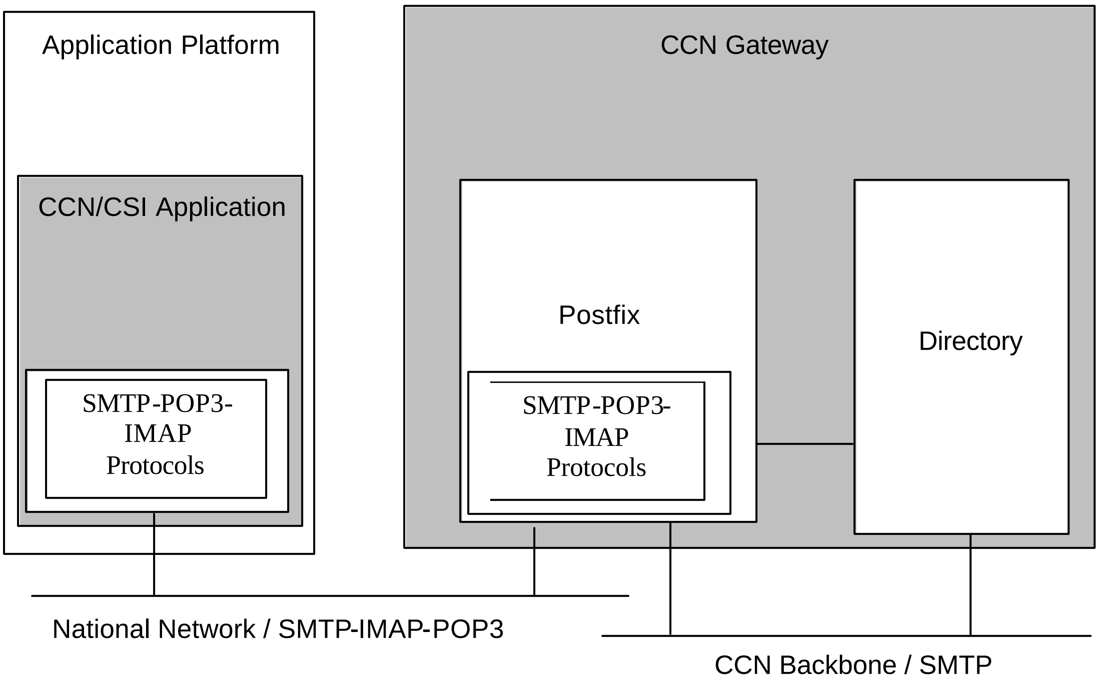


FIGURE 14: CCN/CSI APPLICATIONS AND GATEWAYS ARCHITECTURES FOR MAIL INTERACTIONS.

4.4.2 CCN/CSI APPLICATIONS

4.4.2.1 APPLICATION ARCHITECTURE

The architecture model for the CCN/CSI applications is the client/server model. Thus, a CCN/CSI application can behave either as a client and/or as a server. A client application is an application that invokes (initiates) services (operations) to be performed by a server application.

A CCN/CSI application can be client, server or both at the same time.

The CCN/CSI applications rely on the CSI stack to connect to the CCN Gateway through the National network. The CSI stack, by offering a consistent set of Application Programming Interfaces (API), makes the applications independent of the underlying data transfer protocols.

The CSI API offer the application two sets of verbs to allow the application to function in synchronous (connection oriented) or asynchronous (message oriented) modes.

When functioning in a synchronous manner, both the client and server applications are active during the interaction. Two types of synchronous verbs are available to the applications: blocking and non-blocking synchronous verbs. When using a blocking synchronous verb (a request-response or a HTTP interaction), the client application is blocked until the server application has completed the processing of the requested operation or has reported failure whereas non-blocking synchronous verbs allow the application to invoke other operations without waiting for the previous operation(s) to complete, other verbs being available for the application to retrieve the operation results/outcome.

When working in an asynchronous manner, only one entity (i.e. the client or the server) is active at a time during the interaction. The asynchronous or message oriented mode relies on message-queuing and store-and-forward mechanisms that do not require the applications to be active at the same time.

4.4.2.2 NATIONAL CSI STACK

Due to the variety of the application platforms, the CCN/CSI applications, regardless of their role and the exchange mode they rely on, can be:

- Controlled by a Transaction Processing (TP) monitor.
- Run in a batch environment.
- Run by an operator or a user.

The National Network between the Application Platform and the CCN gateway relies on the TCP/IP protocol for all interactions.

If the whole CSI stack functionality was to be implemented on every Application Platform, a dedicated CSI stack would have to be designed and implemented for each application platform supported by the CCN/CSI software.

The chosen implementation is to remove as many CSI stack features as possible from the application platform and to locate it on the CCN Gateway in order to make the application platform dependent part (i.e. the part to be rewritten for each AP) as small as possible.

This can be achieved by executing remotely (i.e. on the Gateway) each of the CSI verbs. The objective being to restrict the CSI interface on the application platform to the role of a data transmission protocol.

Some of the CCN/CSI features involve the presence of evolved data processing mechanisms on the Application Platforms. An example of such functionality is the security protocol that requires the application (via the CSI stack) to compute the security tokens for the identification, authentication, confidentiality and integrity features. Another example is the data presentation.

As all the features requiring some processing to be performed on the Application Platform are part of the Function layer, this layer cannot be placed remotely. Similarly, the Communication layer consisting mostly of the operating system provided communication protocol software required in order to communicate with the Gateway has to remain on the Application Platform.

On the other hand, the services offered by the Transmission layer (e.g. execution of a call to a service, access to message queues, etc.) can be executed remotely on the CCN Gateway. The Transmission layer on each Application Platform can then be restricted to a function of data encapsulation and forwarding; its peer on the gateway being in charge of both data extraction and execution of services.

4.4.3 THE CCN GATEWAY

4.4.3.1 GATEWAY SOFTWARE PRODUCTS

The CCN Gateway relies on the following software products:

- The CCN/CSI Gateway Software.
- The Tuxedo TP monitor.
- The MQSeries queue manager.
- The Apache HTTP server.
- SuSE Open Exchange Server, for the LCMS usage only.
- The Directory System.
- The AIX and the Linux Operating Systems.

The Tuxedo Transaction Processing monitor, the MQSeries queue manager, the Apache HTTP server and SuSE Open Exchange Server are the pieces of software on which the CCN/CSI Gateway and LCMS Software is built. They are also used to implement the communication protocol used to exchange data on the CCN backbone.

The Directory System is based on a X.500 client-server architecture: the client queries and receives responses from one or more servers using the Lightweight Directory Access Protocol (LDAP). The Lightweight Directory Access Protocol (LDAP) is a simpler version based on TCP/IP of the Directory Access Protocol (DAP). The Directory client, called the Directory User Agent (DUA), provides the standardized functionality that supports searching or browsing through one or more directory databases. The Directory System Agent (DSA) is the database in which the CCN/CSI configuration information (e.g.: definitions of the users, user profiles, applications, services, Application Platforms, Gateways) information is stored. This database is hierarchical in form, designed to provide fast and efficient search and retrieval. The Directory System Protocol (DSP) controls the interaction between two or more Directory System Agents. The Directory System also manages the storage and replication (using the Directory Information Shadowing Protocol (DISP)) of the Directory Information Base (DIB) to synchronise the Directory content on all the CCN Gateways and improve the information access time by maintaining a copy of the directory content on every CCN Gateway and backup Gateway.

The AIX and the Linux Operating Systems offer services, such as file management, basic networking software, etc., on which the CCN/CSI Gateway Software relies heavily.

The mail exchange functionality is running on dedicated hardware running SuSE Open Exchange Server. These machines, called Local CCN Mail Servers (LCMS) are totally independent from the CCN/CSI gateways.

The CCN/CSI Gateway Software implements the CCN/CSI functions. The following figures show the breakdown of the CCN/CSI Gateway Software into software products.

4.4.3.2 SOFTWARE BREAKDOWN FOR CSI SYNCHRONOUS & CSI ASYNCHRONOUS INTERACTIONS

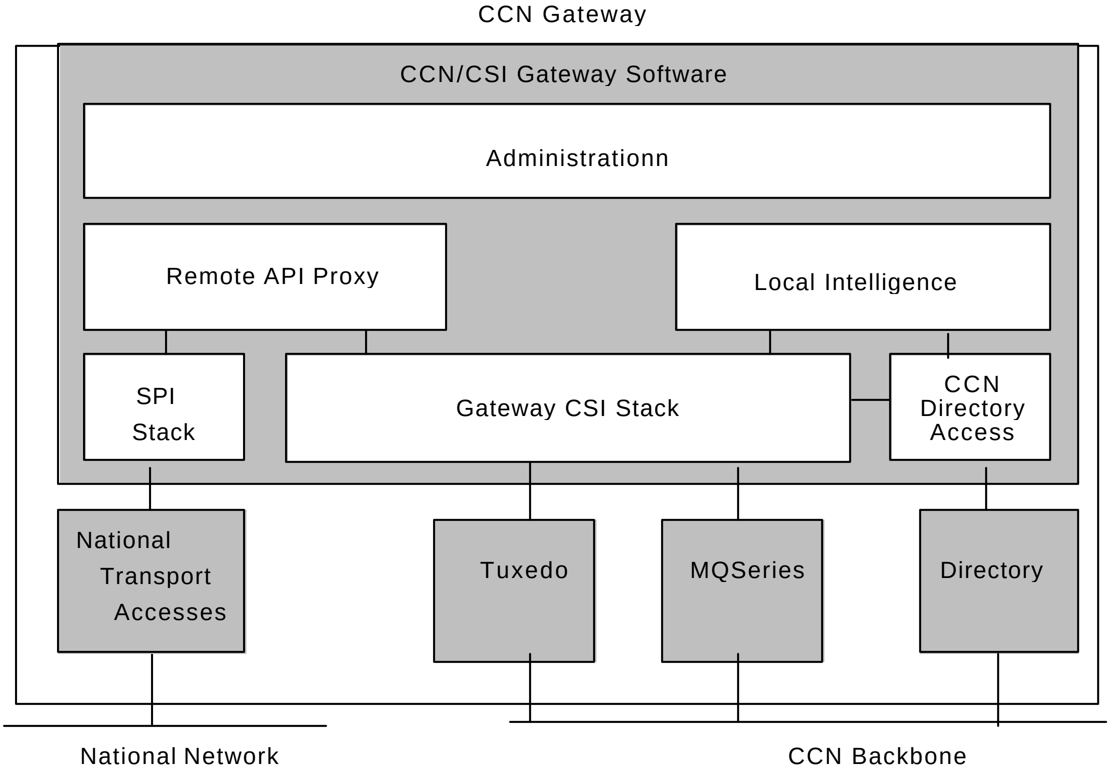


FIGURE 15: BREAKDOWN OF THE CCN/CSI GATEWAY SOFTWARE INTO SOFTWARE PRODUCTS (1).

The following sections describe each of the above depicted software products of the CCN/CSI Gateway Software.

The hatched software products have already been described above except for the National Transport Accesses. The National Transport Accesses consist of the networking software provided by the Gateway operating system and providing the above listed data transfer protocols required to connect with the Application Platforms.

4.4.3.2.1 SPI STACK

The Service Provider Interface (SPI) stack is the peer of the CSI stack running on the application platforms. It does not include the High Level Function layer as this layer is included in the Remote API Proxy (RAP) software product.

The SPI stack offers the Service Provider Interface to the RAP. This interface, also similar to the CSI interface in terms of functionality, is very different in terms of verbs. As the CSI stack functions are remotely executed by the RAP, the SPI looks like the CSI Interface as seen from the Low Level Function layer of the National CSI stack. The following figure summarises this design.

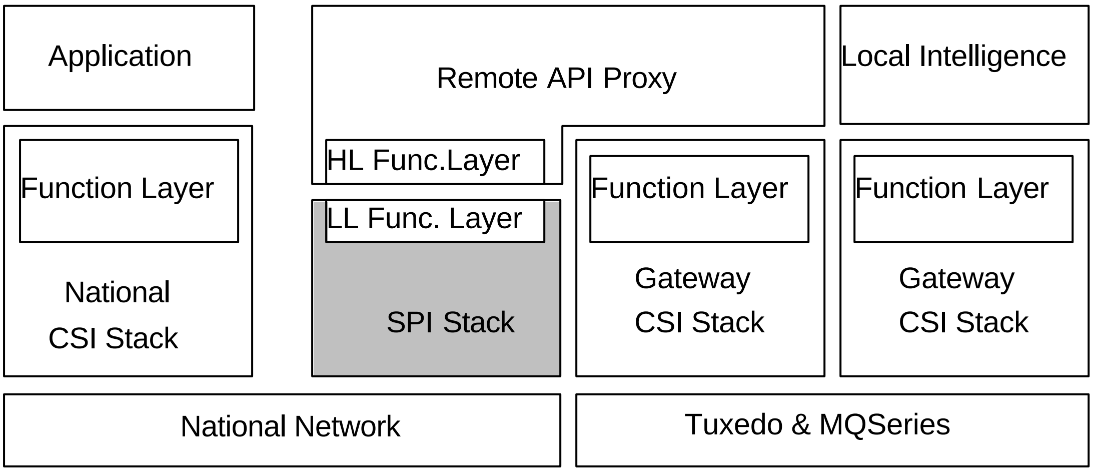


FIGURE 16: SERVICE PROVIDER INTERFACE AND REMOTE API PROXY DESIGN.

To perform the remote execution of the CSI interface verbs, the SPI and the National CSI stack need to exchange data. To make the design as simple as possible, the exchanged data and the associated protocol shall not be dependent on the verbs invoked by the application.

In other words, once the data is encapsulated (except the QoS), the semantics of the verb disappears until reception by the RAP for which the semantics shall be restored. Thus, whether the application uses synchronous or asynchronous transmission mode, the dialogue between the National CSI stack and the SPI stack shall be the same.

A connection-oriented dialogue mode is then required in order to support both the connection-oriented and the message-oriented data exchanges. Moreover, maintaining a connection between the application and the CCN Gateway enables both peers to detect the disappearance of the remote and eases the error recovery.

As a client application is the initiator of the requests, it is also in charge of establishing the connection to the CCN Gateway that shall exist prior any operation invocation. On the other hand, the CCN Gateway is responsible for opening the connections to the server applications prior forwarding operation requests (if none already exists).

Thus, whereas the client applications are always the connection initiators and the server applications are always the connection responders, the applications exchanging messages in asynchronous mode can be either connection initiators or responders or both.

4.4.3.2.2 GATEWAY CSI STACK

The role of the Gateway CSI stack is threefold:

1. To offer the standard CSI API set to applications running locally on the Gateway, such as Generic Application Services, test and administration tools.
2. To allow the CCN/CSI Gateway Software Products to dialogue between each other using the same set of API. Such software products are the Remote API Server, the Local Intelligence (LI).
3. To offer the CCN/CSI Gateway Software an access to the Community Data Exchange functions.

The Gateway CSI stack relies on Tuxedo and provides additional features such as data given by file reference.

In step #1, the full set of CSI API is available to the CCN/CSI Application, including the HL, CSI, Presentation and GSS (Generic Security Service) API. But, as opposed to the CSI stack running on the application platform, the Gateway CSI stack does not implement any security service since the fact of running on the Gateway itself is made secure. Thus, no security module is accessible through the offered GSS API: no authentication is performed and the data is returned unchanged to the application for transmission (no sealing, no ciphering).

In step #2, the Gateway CSI stack translates CSI calls into XATMI (synchronous mode) and MQI (asynchronous mode) calls.

In step #3, the XATMI-like verbs of the CSI API are translated into XATMI (synchronous mode) and MQI (asynchronous mode) calls.

4.4.3.2.3 REMOTE API PROXY

The Remote API Proxy is the CCN/CSI Gateway Software component that is responsible for the remote execution of the CSI API verbs (remote as seen from the application, i.e. not located on the application platform).

It interfaces to the SPI to dialogue with the applications and relies on the Gateway CSI stack to execute the CSI verbs locally.

The RAP is also in charge of the activation of the security function on behalf of the CCN/CSI Gateway Software. Thus, the RAP software includes security modules, accessed through the GSS API, which, as opposed to those available in the Gateway CSI stack, implement the full set of security services and protocols required for CCN/CSI (authentication, confidentiality and integrity).

To be able to activate these security services, especially authentication, the RAP has to behave as a Function layer peer, implementing the protocol machine activated by the applications when using the HL API.

The remote execution of the CSI API verbs is achieved by establishing a connection between the CSI stack running on the Application Platform and the SPI stack. The RAP is in charge of controlling, accepting (case of a client or server-initiator application) or initiating (case of a server-responder application) this connection.

4.4.3.2.4 LOCAL INTELLIGENCE

The Local Intelligence (LI) is the core of the CCN/CSI Gateway Software. It is in charge of routing the CCN messages, performing the data format translation and activating external services in order to fulfil the quality of service requested by the application.

The Local Intelligence activates the following services:

- Exchange Co-ordination, to schedule the other LI services.
- Security, to authorise/deny the use of the message by the user/application according to the Access Control Lists (ACL).

- Data Presentation, to encode data from the Application Platform format into the CCN/CSI pivot format and vice versa.
- Routing, to route a message to the target CCN Gateway or application platform according to the Quality of Service (QoS) requested by the client application and the mode of the resource.

The LI relies on the Gateway CSI stack running over Tuxedo to interface with the Application Platforms through the Remote API Proxy and with the remote CCN Gateway through the CCN Backbone. It uses the CCN Directory Access to retrieve the configuration, security and routing information (e.g.: Application service/queue name, default QoS, remote Gateway addresses, ACL). The Gateway CSI stack is a layer implemented over XATMI and MQI. It provides to the Gateway Software Products the CSI API to interface with Tuxedo and MQSeries.

4.4.3.3 SOFTWARE BREAKDOWN FOR HTTP INTERACTIONS

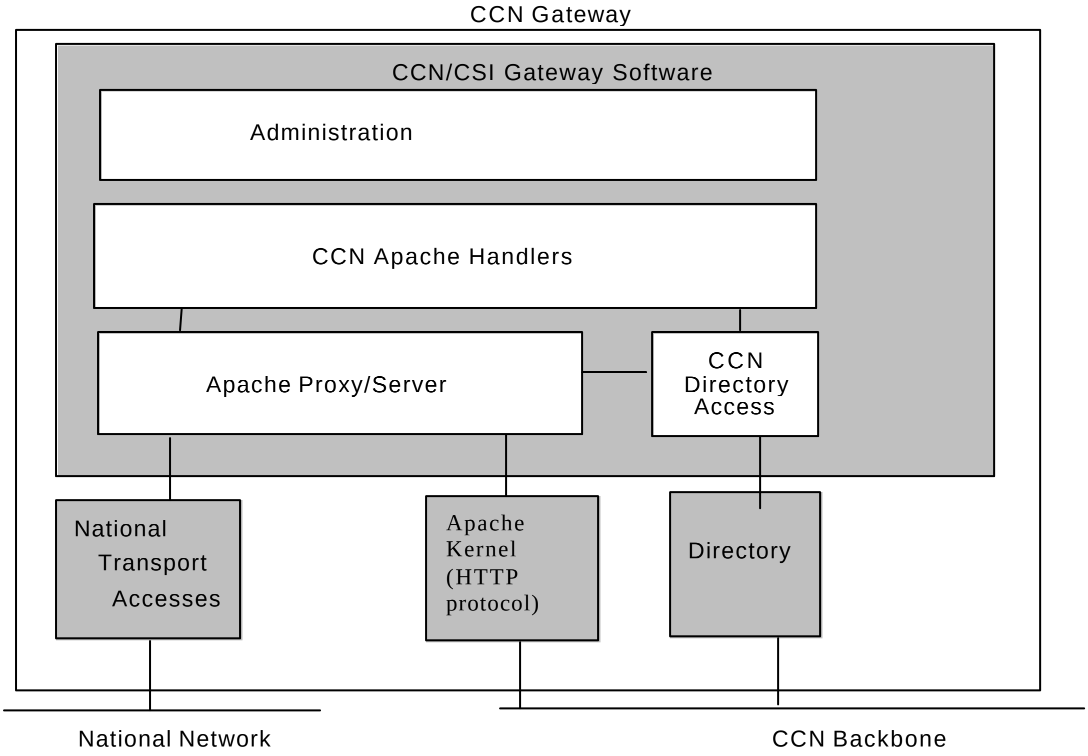


FIGURE 17: BREAKDOWN OF THE CCN/CSI GATEWAY SOFTWARE INTO SOFTWARE PRODUCTS (2).

The HTTP server Apache is used for HTTP interactions. HTTP clients running on Applications Platforms performs requests respecting the HTTP protocol towards this HTTP server. This server acts like a proxy that will forward the initial request or a real server that will answer the request in the case it is hosting the asked resource. This server is also responsible for the forwarding of the request on the Community Domain if needed.

The CCN/CSI software relying on the Apache HTTP server is encapsulated in handlers. These handlers are executed one after the other in conformity with the request lifecycle implemented in Apache. More details about this life cycle will be given in a following chapter.

These handlers and Apache itself fetch in the CCN Directory the needed configuration information.

4.4.3.3.1 HTTP PROXY

The HTTP Proxy is the CCN/CSI Gateway Software component responsible for the caching of HTTP requests made by CSI HTTP clients. More precisely the HTTP Proxy is a `forward` proxy, an intermediate server sitting between the client and the target server: the client sends the request naming the target server to the proxy which in turn acts as a client with this server. The main goal of such an implementation is to reduce network load.

The HTTP Proxy is also in charge of the activation of the security function on behalf of the CCN/CSI Gateway Software. It also authorises or denies the use of the HTTP service against the Access Control Lists (ACL). Thus, the HTTP Proxy software includes security modules, accessed through the GSS API.

The HTTP Proxy is in charge of the routing of the request towards the HTTP server, taking into consideration the mode of the resource and the gateways.

4.4.3.3.2 HTTP SERVER

When the HTTP Apache server running on the gateway hosts the requested resource, Apache will activate the appropriate handler to provide the correct response. In this case, it acts like a GAS. All the CCN/CSI handlers are written using the C Apache module API.

4.4.3.4 SOFTWARE BREAKDOWN FOR MAIL INTERACTIONS

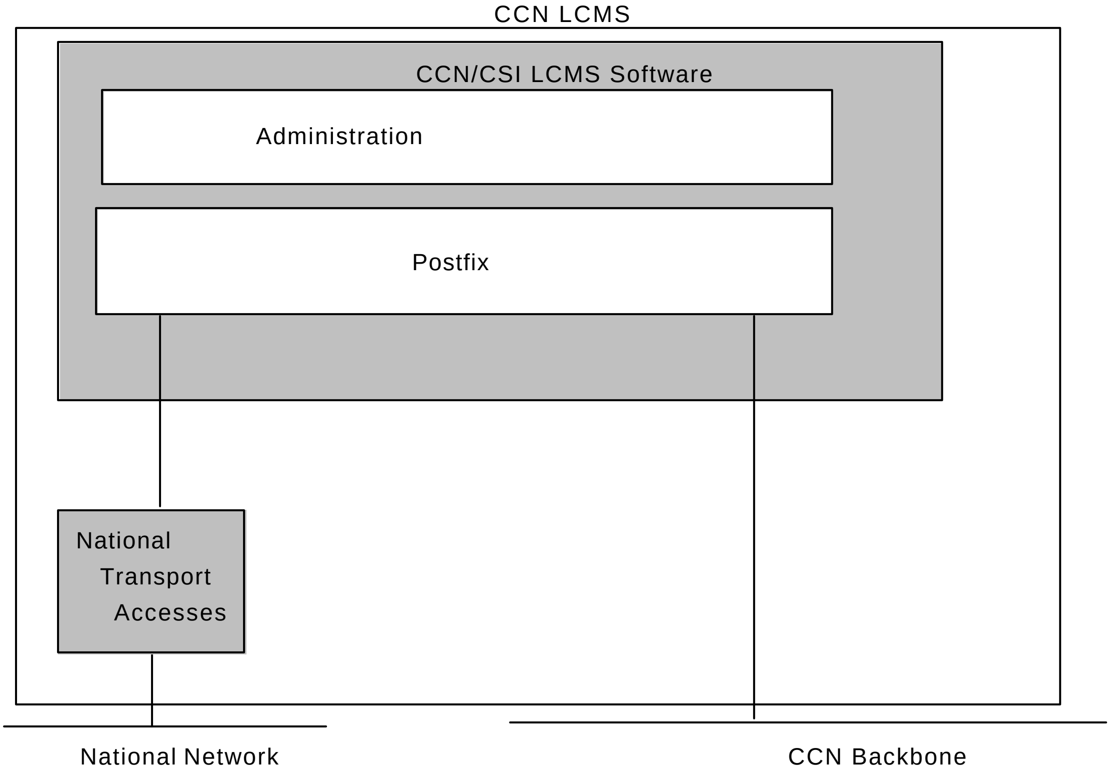


FIGURE 18: BREAKDOWN OF THE CCN/CSI GATEWAY SOFTWARE INTO SOFTWARE PRODUCTS (3).

The SuSE Open Exchange software is installed on LCMS machines. This suite installs the Postfix software that takes the mail exchanges in charge. Mail clients running on Applications Platforms performs requests respecting the SMTP, IMAP and POP3 protocols towards this mail server. This mail server is responsible for the routing and the delivery of the message to its final destination using the SMTP protocol. At the other end, another mail client will read the message using the IMAP and POP3 protocols.

4.4.3.5 CCN DIRECTORY ACCESS

The CCN Directory Access (CDA) provides a value-added access to the Directory in which are stored the CCN/CSI configuration information.

It formats the Directory Access Protocol (DAP) requests using the LDAP API and forwards them to the Directory User Agent (DUA).

The CCN Directory Access is implemented as a library. The CDA is, besides the Administration tools, the sole software product of the CCN/CSI Gateway Software that is aware of the structure of the CCN Directory, i.e. the Directory Information Tree (DIT).

To enhance the Gateway performances, the CDA implements a CCN/CSI oriented caching mechanism. In the processing by the CCN/CSI Gateway Software of a CSI verb invocation, several types of information records are queried from the CCN Directory (e.g.: user, user profile, application, service, and gateway). During a CCN Directory query, a record is stored in the cache if it is not yet in the cache. This allows faster accesses to the CCN Directory during the subsequent queries for the same records.

4.4.3.6 ADMINISTRATION

The Administration Software product provides administration functions to the administrators to configure and manage the CCN Gateway. This set of administration functions has been described in the functional overview chapter.

The operators activate the administration functions. The operators can run and stop the administration functions as individual tasks, and independently of the other software running on the CCN Gateway. Thus, in terms of architecture, the administration software is independent from the other CCN Gateway software.

The Administration Software Product runs on the CCN Gateway installed at each National Administration site. It allows the local administrator to configure and manage the CCN Gateway locally. It allows also the central administrator from the CCN/TC to perform remote administration on the CCN Gateway of the National Administration site, through the remote logging mechanism or through an internet/intranet access to the CCN Gateway HTTP server. The Administration Software Product runs also on a CCN Gateway installed at the CCN/TC, to allow the central administrator to perform administration tasks relevant to the whole CCN system, such as configuration and management of the CCN configuration information stored in the CCN Directory.

The administration software is actually the first part of the CCN Gateway software that is started on each CCN Gateway. The administrator then activates the other CCN Gateway software via the administration software.

When the CCN Gateway software start-up is completed, the various administration functions don't need to be present permanently, except two function groups:

- Process Supervision, which supervises the processes running on CCN Gateway.
- Error, Trace and Statistics Data Collection & Reporting, which collect the error and trace messages and various statistics data reported by the other CCN Gateway software.

Besides the operator commands, the main types of interaction between the administration software and the other CCN Gateway software are the collection of the error and trace messages and various statistics data reported by the other CCN Gateway software.

The administration software provides a common and consistent reporting service for this purpose, which is used by other CCN Gateway software components to generate error messages, trace messages, security records, performance-related statistics data and accounting-related statistics data.

In general, all the CCN Gateway specifically developed software products make use of the common reporting interface to generate error messages and trace messages. On the other hand, only some software products make use of this common reporting interface to generate security records, performance-related and accounting-related statistics data. The relevant software products are identified.

With few exceptions, the administration software relies directly on the native administration tools/interfaces provided by the CCN Gateway software products, such as Gateway Operating System, Tuxedo, MQSeries, Apache and CCN Directory. Some administration functions are implemented as high-level scripts based on the native commands provided by administration tools of these software products.

Two distinct interfaces are provided to the administrators for accessing the administration software:

- A text-oriented interface consisting in an authenticated access to the Gateway Operating System (either local or remote).
- A graphical Web-oriented interface relying on HTML browsers accessing the HTTP server running on the concerned CCN Gateway.

4.4.3.7 DETAILS ON EXCHANGES

4.4.3.7.1 CSI SYNCHRONOUS REQUEST-RESPONSE EXCHANGE

A CSI synchronous request-response call of a service by an application can be summarised as follows:

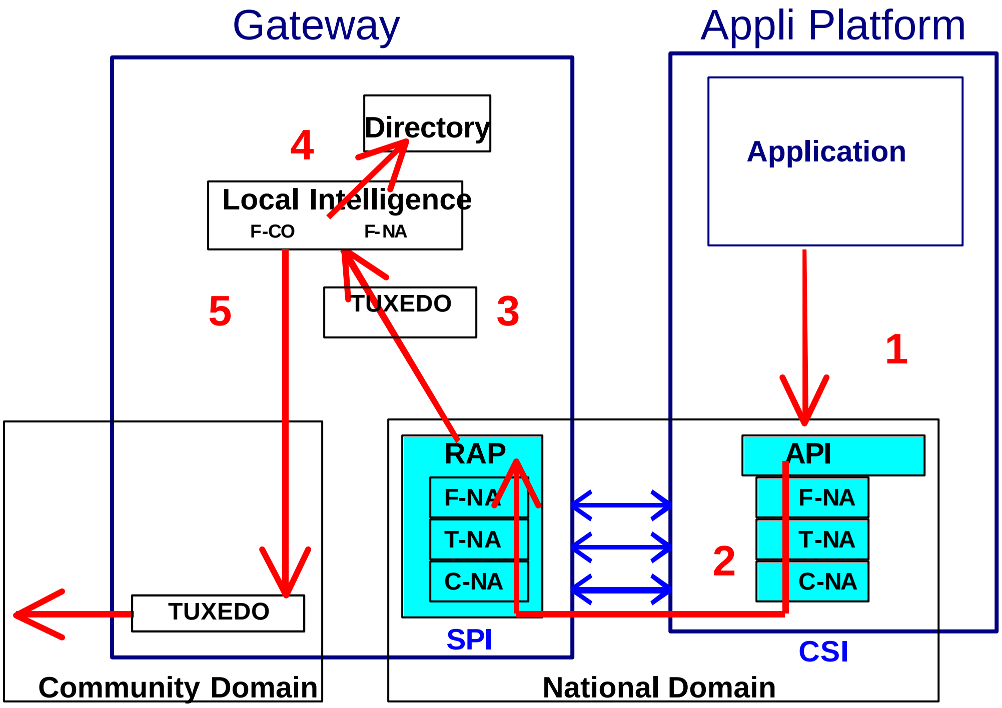


FIGURE 19: CSI SYNCHRONOUS REQUEST-RESPONSE EXCHANGE.

1. The application uses the HL_call() API verb.
2. The Application Platform CSI stack packages the parameters of the invoked verb and sends them to the Gateway.
3. The Gateway SPI stack unpacks the parameters, recognises the HL_call() verb and invokes the appropriate function within the Remote API Proxy, which is in charge to map this verb to the Tuxedo tpcall() verb to the Local Intelligence.
4. The LI looks up information in the Directory for ACL, routing and presentation. It checks the ACL and performs presentation conversions.
5. The LI forwards the HL_call() to the target Gateway using the CSI API and sends the request through Tuxedo following the Request/Response synchronous mode.

4.4.3.7.2 SYNCHRONOUS HTTP EXCHANGE

A synchronous HTTP exchange can be described as follows:

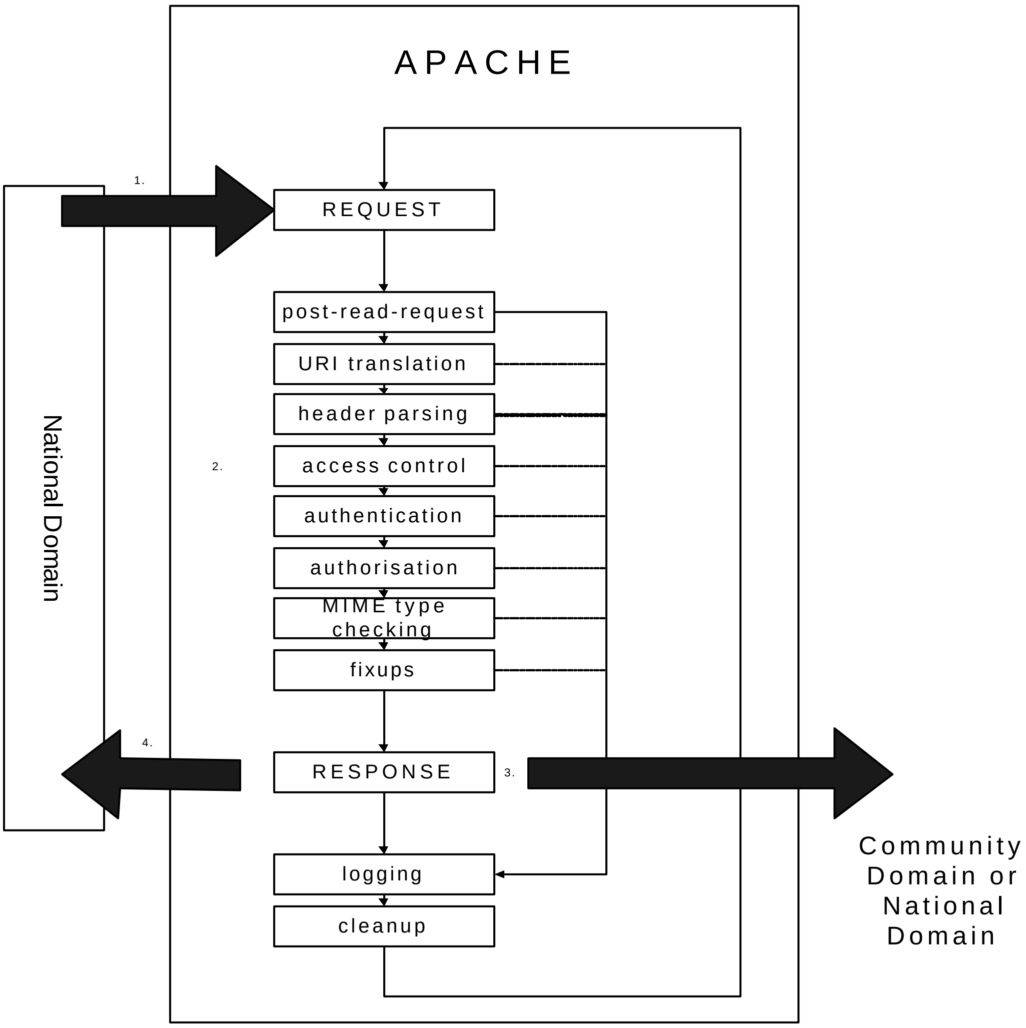


FIGURE 20: SYNCHRONOUS HTTP EXCHANGE.

1. A request is made by a CCN/CSI HTTP client running on a Application Platform.
2. Each handler part of the request lifecycle is executed in turn. These handlers can be the default ones defined in Apache or customs ones developed to suit a particular task. Note that each handler can stop the normal flow of execution to jump to the logging handler. A brief description of each handler is given hereafter.
3. The response is built in a so-called content-handler. This handler can behave like a proxy (normal behaviour if the invoked resource is hosted on a server in the National Domain) or a server (GAS).

4. The response is sent back to the client.

Here is a brief description of each Apache handler depicted in Figure 20:

- The post-read-request handler

  This handler is called each time an Apache process receives an incoming request. It is executed before the server has translated the URI to a filename. This handler must be used to do processing that must occur once for each interaction.

- The URI-to-filename translation handler

  This handler is invoked after the Apache process has parsed out the request. It takes the request and transforms the URI into a filename. This handler is often used to recognise and handle proxy requests.

- The header parser handler

  In this handler, Apache gives a second chance to examine (and even modify) the headers and to take a specific action. The difference with the post-read-request handler is that at this point the physical pathname for the URI is known.

- The access handler

  This handler is the first of the three handlers involved in authentication and authorisation. This is here that a cookie-based access control can be implemented.

- The authentication handler

  The authentication handler is called whenever the requested file or resource is password-protected.

- The authorisation handler

  After the authentication phase comes the authorisation phase of the interaction, in which the invoked handler can determine whether a specific user can fetch a specific URI.

- The MIME type checker handler

  Following the successful completion of the access control and the authentication steps (if any), Apache tries to determine the MIME and encoding types of the requested document. They are usually determined by the filename extensions.

- The fixup handler

  This handler gives a last chance to add information to the environment or to modify the request before the content handler is invoked.

- The content handler

  This handler is the step of the content generation or the response phase. It takes all the information generated by the previous phases to build a single document and serve the result to the client that made the request.

- The logging handler

  The very last phase of the interaction is the logging phase. At this point, the request record contains everything there is to know about the interaction, including the final status code and the number of bytes transferred to the client.

4.4.3.7.3 CSI ASYNCHRONOUS EXCHANGE

The following scenario summarises the operations performed when an application puts a message in a queue in asynchronous mode:

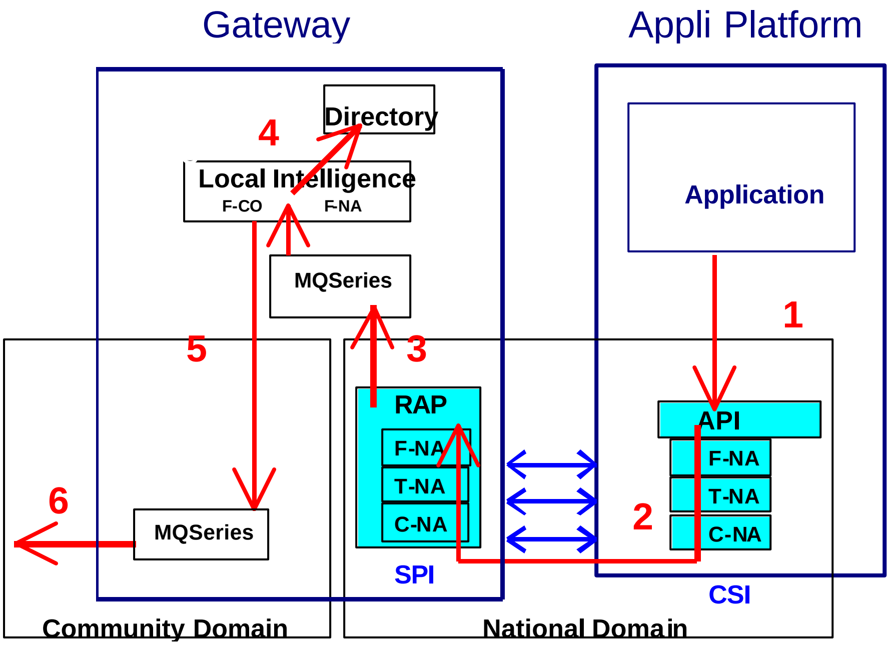


FIGURE 21: CSI ASYNCHRONOUS EXCHANGE.

1. The application uses the HL_mq_put() verb.
2. The Application Platform CSI stack packages the parameters of the invoked verb and sends them to the Gateway.
3. The Gateway SPI stack unpacks the parameters, recognises the HL_mq_put() verb and invokes the appropriate function within the Remote API Proxy, which is in charge to map this function to MQSeries MQPUT verb to the Local Intelligence.
4. MQSeries writes the message to a queue and wakes up the LI. The LI looks up information in the Directory for ACL, routing and data presentation. It checks the ACL and performs presentation conversions.
5. The LI forwards the HL_mq_put() to the target Gateway using the CSI API and sends the message through MQSeries.

4.4.3.7.4 MAIL EXCHANGE

The following scenario summarises the operations performed when an application sends a mail on the CCN backbone:

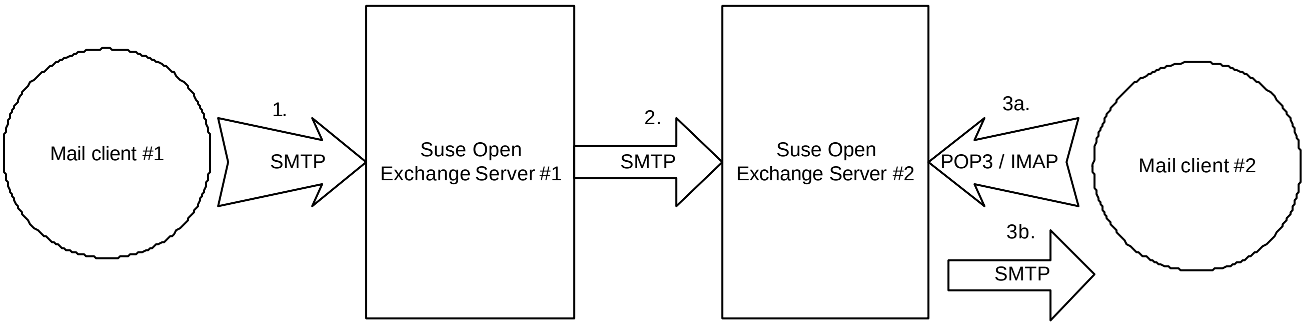


FIGURE 22: MAIL EXCHANGE.

1. One mail client sends a mail to its Local CCN Mail Server running SuSE Open Exchanger Server. This operation relies on the SMTP.
2. The transfer of this mail is done using the SMTP from the originator LCMS to the recipient LCMS.

3a. One mail client reads the mail stored on its Local CCN Mail Server running SuSE Open Exchanger Server. This operation relies on the IMAP and POP3 protocols.

3b. As an alternative of the *read* operation performed by client #2 described at step 3a above, the recipient LCMS can forward the mail to the National Administration mail server using the SMTP.

5. ENTITY RELATIONSHIP DIAGRAM

5.1 CCN/CSI MACRO-MODEL

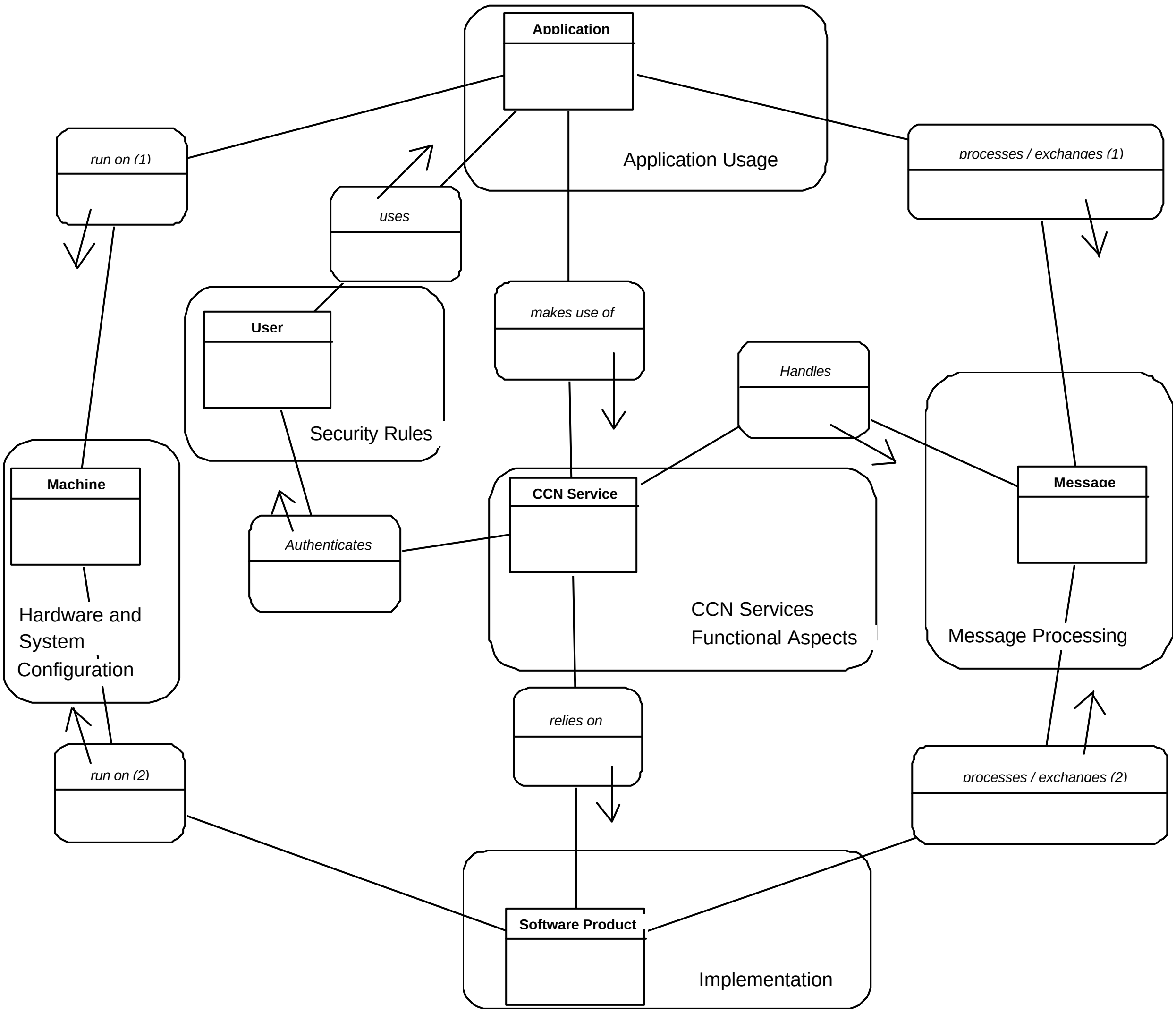


FIGURE 23: CCN/CSI MACRO-MODEL.

This figure is another global view of the CCN/CSI infrastructure. It is composed of six macro-entities that group the main entities of CCN/CSI according to six major aspects of CCN/CSI. A given entity can be included in several macro-entities. The macro-entities are briefly described below.

Please note that the following conventions are used to draw Figure 23 and Figure 24:

- The big rounded boxes represent the six macro entities of CCN/CSI.
- The square boxes included within the big rounded boxes represent the main entities of CCN/CSI.
- The small rounded boxes (with arrows) illustrate the relationships between entities.

5.1.1 APPLICATION USAGE

The *Application Usage* macro-entity covers:

- The characteristics of the *application* itself

  An *application* can be a client application or a server application or both.
  It offers and makes use of *application services*.
  In *synchronous mode*, application services are offered by server applications as functions that are remotely called by the client applications.
  In *asynchronous mode*, the applications exchange *messages* through *queues*.

- The interactions with the CCN/CSI infrastructure

  The application communicates through the *CCN Access Interfaces* with the *CCN Services*.

Application Usage macro-entity includes the following entities: *Application Instance*, *User*, *Application Service*, *Queue*, *CCN Access Interface*.

<u>*Remark*</u>: The interfaces offered by the client applications to the users can be graphical user interfaces or text oriented interfaces. The design of user interfaces is under the responsibility of application developers and is not within the scope of the CCN/CSI project.

5.1.2 CCN SERVICES FUNCTIONAL ASPECTS

The *CCN Services Functional Aspects* cover all the services offered by the CCN/CSI infrastructure and the interfaces to those services.

CCN/CSI Services Functional Aspects macro-entity includes the following entities: *CCN Access Interface* and all the *CCN Service* entities (*Security Service*, *Directory Service*, *Communication Service*, *Presentation Service*).

5.1.3 MESSAGE PROCESSING

The *Message Processing* macro-entity covers all the operations executed by the applications and by the software components implementing the CCN services. In particular, the satisfaction of the QoS that accompanies the messages is one of the most important operations (please refer to § 5.2.1.9).

Message Processing macro-entity includes the following entities: *Message*, *QoS*, *Transmission Mode*, all the *CCN Service* entities, *Application Service* and *Queue*.

5.1.4 SECURITY RULES

The *Security Rules* are executed by the *Security Service* to protect the CCN/CSI infrastructure, the applications and the users against unauthorised usage of the resources and to guarantee the confidentiality and the integrity of the exchanged messages.

Security Rules macro-entity includes the following entities: *Security Service*, *User*, *User Profile*, *Application Instance and Access Control List.*

5.1.5 HARDWARE AND SYSTEM CONFIGURATION

The Hardware and System Configuration macro-entity covers:

- The *platforms* of the National Administration.
- The platforms of the Community Domain implied in the CCN/CSI infrastructure (the *Gateways*).
- The links between those platforms (*National Networks* and *CCN Backbone*).

The various platforms are Gateways, Backup Gateways, Application Platforms, Workstations, and Administration Stations.

The links can be "Secure" or "Non-Secure".

Hardware and System Configuration macro-entity includes the following entities: *CCN Gateways*, *NA Platforms*, *National Network and CCN Backbone*.

5.1.6 IMPLEMENTATION

The Implementation macro-entity covers all the *Software Products* of the CCN/CSI infrastructure installed on the platforms enumerated above.

These software products are:

1. Off the shelf products like Tuxedo, MQSeries, Apache HTTP server, Sun ONE Directory, communication stacks.
2. New specific components developed for the CCN/CSI project (e.g.: CSI stacks, Local Intelligences, Remote API Proxies).

5.2 CCN/CSI ENTITY-RELATIONSHIP DIAGRAM

The CCN/CSI entity-relationship diagram is a more detailed view of CCN/CSI and presents the main entities concerned by or implied in the CCN/CSI infrastructure and the relationships that can be established between them.

Those entities and relationships are described below in Figure 24.

Please note that the relationships and their associated entities are also highlighted in each sub-paragraph of § 5.2.2 in order to localise them within the picture and to illustrate each part of the relationship descriptions.

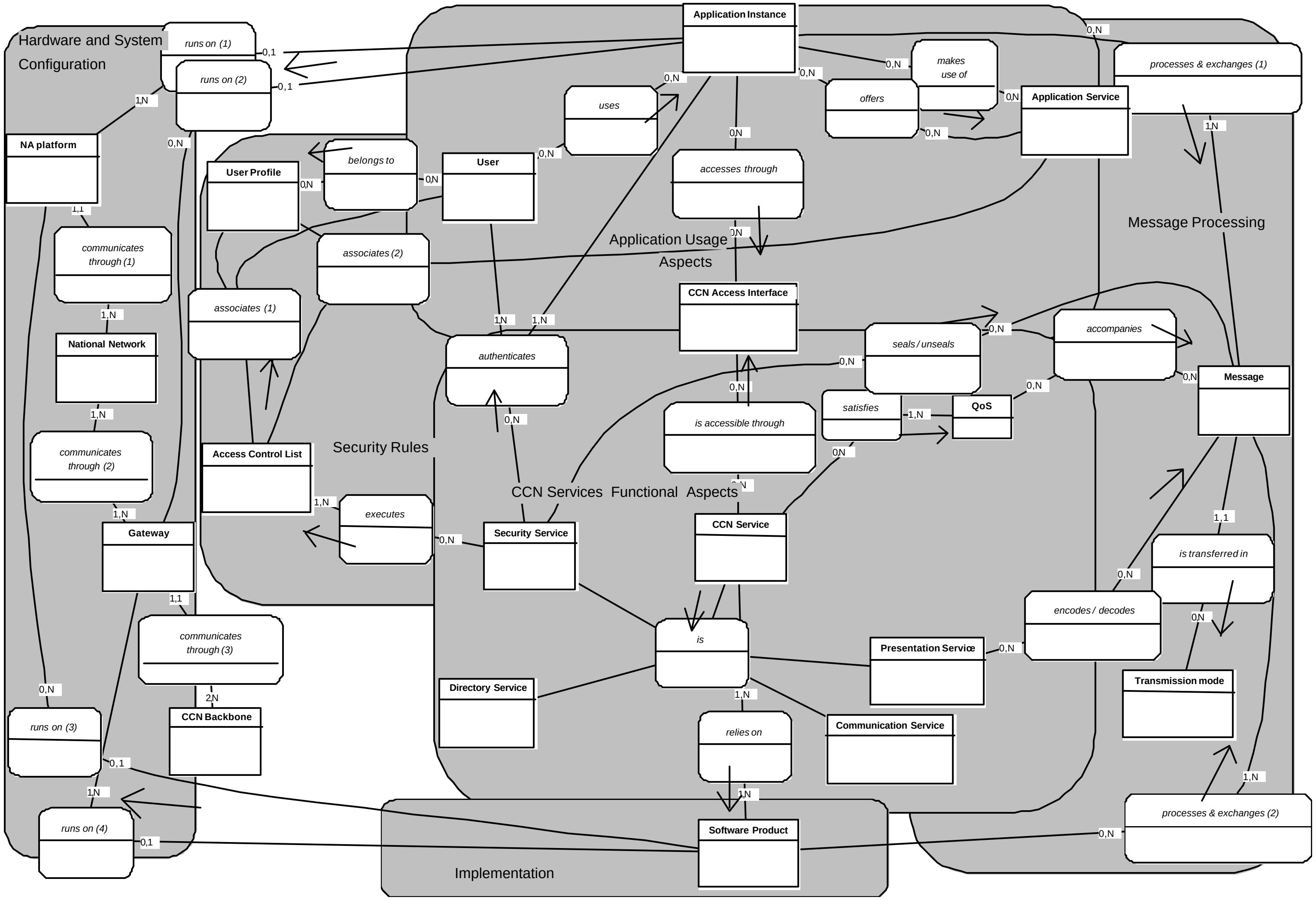


FIGURE 24: CCN/CSI ENTITY-RELATIONSHIP DIAGRAM.

5.2.1 ENTITIES

5.2.1.1 APPLICATION INSTANCE

In the frame of CCN/CSI, an *application* is a program that runs on an *NA Platform* (or *CCN Gateway*) and that exchanges messages with other applications running on the CCN/CSI infrastructure. The instance of a program is called an application process.

On the Entity-Relationship Diagram, the *application instance* entity represents the instance of a CCN/CSI application running on one given platform.

The first kind of application is the user application whose role is to allow the National Administrations to execute their core business. The user applications run on Application Platforms on the National Domain.

The other type of application concerns the applications running on the Gateways on the Community Domain. Whether they implement Generic Application Services (GAS) accessible by any user application or they are tools or programs for test and administration purpose insuring a high level of availability and performance of the CCN/CSI infrastructure, they never offer a specific service to a particular user application, respecting in this way principle 4 of the IDA recommendations: "The EuroDomain and the EuroGates remain pure from end-user applications".

5.2.1.2 APPLICATION SERVICE

The applications make use of application services or offer them to other applications. An application service is a program or a portion of a program that performs, upon receipt of a specific message, an application function to process the message. An application service is invoked by a requester application that sends to it a message, using the CCN/CSI infrastructure.

In the classical client/server sense, the requester application is a "client" and the responder application is a "server", where a client sends requests and a server sends responses to those requests. In this terminology, client and server exchange their messages in a connection oriented transmission mode and there is a "one-to-one" relationship between the application service defined here and its implementation in a synchronous service provided by one or several application processes available via asymmetrical API (e.g.: tpcall, tpacall, tpgetgrply, tpservice, tpreturn for synchronous request-response services relying on Tuxedo, web services relying on the Apache HTTP server and primitives implemented in the SMTP, IMAP and POP3 protocols for mail exchanges).

The application service can also be offered by applications that exchange message in a "peer-to-peer" relationship using message queuing transmission mode. In this case, both applications can be the requester as they can also be responders. The service implementation is provided by one or several processes available via symmetrical API(e.g.: mqopen, mqclose, mqput, mqget).

5.2.1.3 TRANSMISSION MODE

CCN/CSI defines two transmission modes:

Connection oriented mode	In this transmission mode, all the entities participating to the transmission of a message are physically connected between the time the message is submitted by the requester application to the CCN/CSI infrastructure and the time the message is read by the responder application. This transmission mode is also called synchronous transmission mode. A call to an application service in synchronous request-reponse mode can be blocking or non-blocking according to the fact the requester application waits for the reply to a request or not, before continuing its processing.
message queuing mode	In this transmission mode, there are no physical connections between application processes exchanging messages. Communication occurs by one process putting messages in a queue and another process extracting the messages from the queue. This transmission mode is also called asynchronous transmission mode.

TABLE 1: TRANSMISSION MODES.

5.2.1.4 MESSAGE

A message is a collection of data sent by an application process or by a component of the CCN/CSI infrastructure and intended for another application process or CCN/CSI component.

CCN/CSI defines two types of message:

Application message	A message generated by an application or by a component of the CCN/CSI infrastructure.
System message	A message generated by a component which is provided by the CCN/CSI infrastructure.

TABLE 2: APPLICATION MESSAGE TYPES.

In connection-oriented mode, CCN/CSI defines two sub-types of message:

Request	A message for which a reply is expected
Response	A reply to a request message

TABLE 3: APPLICATION MESSAGE SUB-TYPES (ORIENTED-MODE).

In message queuing mode, CCN/CSI defines four sub-types of message:

Datagram	A message for which no reply is expected
Response	A reply to a request message
Request	A message for which a reply is expected
Report	A message that describes an event such as the occurrence of an error

TABLE 4: APPLICATION MESSAGE SUB-TYPES (QUEUING-MODE).

A message consists of two parts: control information and message body.

The control information contains such items of information as the type and sub-type of the message, an identifier for the message and which quality of service is applied or required for this message.

The structure and content of the message body vary with the message type.

The structure and content of the application message body are determined by the participating application processes, not by CCN/CSI which is only responsible for delivering the messages to an appropriate user application process in the appropriate data presentation format.

The structure and content of the system message body are determined by CCN/CSI. Such a message contains some diagnostic information about the event or the error to report.

5.2.1.5 USER

The "user" represents the entity using an application. This entity is involved in the authentication with the CCN/CSI security services.

A user can be a human user or a logical user. Generally, human users use client applications on workstations and server applications are used by logical users. However, it is the responsibility of the National Domain administrators and application administrators to choose what type of users to apply.

The resource usage rights are not defined for each user individually. They are defined according to the user profile, as explained hereinafter.

The users are created and administrated locally by the National administrators.

5.2.1.6 USER PROFILE

The user profile contains common security information for a group of users. A user profile is valid during a given period.

A user profile is created and administrated by the National administrators. A user profile is common for the whole CCN/CSI infrastructure and its definition is available for all the National Domains.

5.2.1.7 ACCESS CONTROL LIST

CCN/CSI defines two types of Access Control List.

The first list contains the authorised profiles for each user. A user belongs to one or more profiles. The National administrators grant the membership of a user profile.

The second list contains the authorised resources for each profile. More precisely, we mean by authorised resources, the application services and queues. The membership of a service in a profile authorises any user belonging to the given profile to call the service. The membership of a queue in a profile authorises any user belonging to this profile to put messages into the queue and to extract messages from the queue.

5.2.1.8 CCN ACCESS INTERFACE

The CCN Access Interface entity represents the way by which applications and gateway components interface with CCN. This covers all the API offered by CSI: HL_API, PRES_API, GSS_API, CSI_API, SPI_API, T_API, CT_API and the API to the web services provided by HTTP servers hosted on CCN Gateways or Application platforms.

- The HL_API, CSI_API, SPI_API, PRES_API and GSS_API are the API to the Function Layer implementing the Function Services (e.g.: encode/decode data, compress/uncompress data, seal/unseal data, generate tokens for security context, transfer security context, 'acall', 'getrply', activation of service, 'return', 'mqput', 'mqget', ..).
  The T_API is the API to the Transmission Layer implementing the Transmission Services (i.e.: remoting call of the CSI_API, local call of the CSI_API.).
  The CT_API is the generic API to the Communication dependent Transmission Services implemented on locally available communication systems (TCP/IP).
- The Apache HTTP servers running on the CCN gateways offer entry points to CCN backbone. The CCN Access Interface offered by these servers relies on the content-handler mod_proxy (the built-in Apache module responsible for the caching of the requests).
- The LCMS machines offer entry points via the SMTP, IMAP and POP3 protocols.

5.2.1.9 QOS

When an application uses a CSI synchronous request-response service or puts a message into a queue, it requires the satisfaction of a QoS (quality of service).

The QoS is conveyed through the whole CCN/CSI infrastructure in order to make all pertinent entities aware of its value.

If no per-message QoS is required by the application, default QoS is applied.

The QoS conveyed with a message contains information about confidentiality and integrity of the message, urgency of the message, notifications to send to the application, compression to apply to the message body.

The QoS conveyed also with the message information concerns the Class of Traffic (CoT). This information, defined and used by the administrator in order to provide a particular level of service, is used on the gateway to route the message according to the required level of service. Using this information, for example, the CCN/CSI can store badly formatted messages to the appropriate dead-letter-queue.

5.2.1.10 CCN SERVICE

The CCN service is introduced here to make the entity-relationship diagram easier to read. In reality it groups all the services that constitute the CCN functions offered to the applications or used for its internal purposes.

Some services are implemented in software components on Gateways and Application Platforms (e.g.: the National FTC-stack). Other services are implemented in software components (e.g.: the Local Intelligence) on Gateways only.

These services are listed hereinafter.

5.2.1.11 SECURITY SERVICE

The Security Service covers three functions: authentication, confidentiality and integrity.

It is part of the Function Layer.

On National Domain, the GSS_API is used to call the security services and to compute security tokens. For confidentiality and integrity, data are sealed and unsealed (in the GSS_API terminology, “sealing/unsealing” means sealing/unsealing **and** enciphering/deciphering). For the authentication, the Security Service relies on CSI_API to transfer the security context.

On Community Domain, the Security Service relies on the Gateway cryptographic devices plugged on the CCN Backbone.

5.2.1.12 PRESENTATION SERVICE

The Presentation Service offers the data format conversion and data compression facilities described in the Functional System Specifications.

5.2.1.13 CCN DIRECTORY SERVICE

The CCN Directory Service contains all the CCN/CSI configuration information (e.g.: definition of the users, user profiles, applications, application services, queues, Application Platforms, Gateways, message formats).

The CCN Directory Service relies on the Sun ONE Directory. It manages storage and replication in order to synchronise the Directory content on all the Gateways and to improve the information access time by maintaining a copy of the Directory content on every Gateway.

5.2.1.14 COMMUNICATION SERVICE

On National Domain, Communication Services are relying on the TCP/IP layers offered by the Application Platforms.

On Community Domain, Communication Services are based on the Tuxedo, Apache HTTP server, MQSeries and SuSE Open Exchange products to exchange data on the CCN Backbone.

5.2.1.15 SOFTWARE PRODUCT

All the CCN Services rely on Software Products implemented or not by the Project. The Software Products are the following packages installed on the NA platforms and/or on the CCN Gateways (cf. Architecture Specifications document for a description of those Software Products).

- The HL_API, PRES_API, GSS_API, CSI_API, T_API, CT_API and OS_API libraries implemented on the NA platforms.
- The API to the communication systems available on the NA platforms (sockets, Windows sockets), those API on the Gateways.
- The SPI_API on the Gateways, which corresponds to the CSI_API on the NA platforms.
- The Remote API Proxy on the Gateways.

- The HL_API, PRES_API, GSS_API, CSI_API, T_API, CT_API libraries implemented on the Gateways.
- The Local Intelligence on the Gateways.
- The Generic Application Services (GAS) implementation on the Gateways.
- The Administration tools on the Gateways.
- The Tests tools on the Gateways.
- The Tuxedo TP monitor on the Gateways.
- The Apache HTTP server on the Gateways.
- The MQSeries queue manager on the Gateways.
- The Sun ONE Directory System on the Gateways.
- The SuSE Open Exchange Server on the LCMS machines.

5.2.1.16 NA PLATFORM

The term "platform" is used to refer to the trio made up of computer hardware, the operating system its runs, and possibly the transactional monitor. Thus, the computer hardware running two different operating systems (and vice versa) are considered as different platforms.

A NA Platform (or Application Platform) is a platform running an application used by a National Administration for its core business.

5.2.1.17 CCN GATEWAY

For convenience purpose on the schema, the CCN Gateway entity includes also the LCMS machines.

The CCN Gateways and LCMS machines are the only access points to the CCN.

In the IDA terminology, they are the EuroGates to the EuroDomain.

5.2.1.18 NATIONAL NETWORK

The National Network is the link by which a NA Platform communicates with its CCN Gateway.

The link can be a secure link or a non-secure link, according to the type of NA Platform and according to the communication media between the NA Platform and the Gateway.

A secure link is a connection between an NA Platform supported by a 'mainframe' type environment, which has an intrinsic security structure, and a CCN Gateway protected by a high physical security. In this type of environment, the host system has the means that enable it to identify the connected terminals and to authenticate the users. The link between the host and the CCN Gateway is a direct link, more often within the same room. The risks of intrusion are thus very low.

A non-secure link is a connection of a NA Platform supported by an environment of the 'autonomous system' type (PC, LAN), which has no intrinsic security structure and a CCN Gateway. In this case, the identification/authentication of the entity (person or process) requires a higher level of security in order to avoid a masquerade.

Therefore, the link is considered as non-secure if the NA Platform is "non-secure" or if the communication media between the NA Platform and the Gateway is “non-secure”.

5.2.1.19 CCN BACKBONE

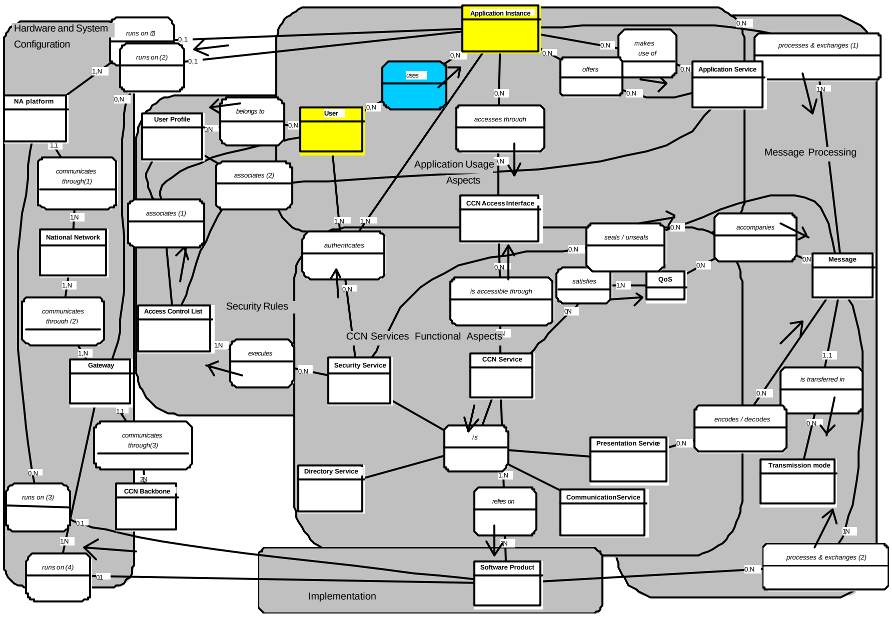
The CCN Backbone is the link by which the CCN Gateways communicate.

5.2.2 RELATIONSHIPS

5.2.2.1 USES

A given user *uses* zero to many application instances (0,N connectivity for the link "*User - uses*").

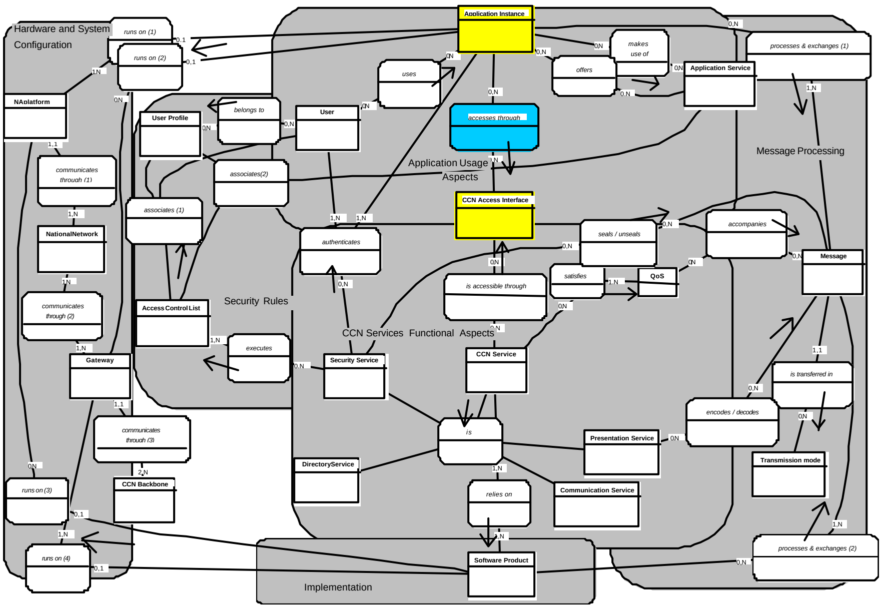
A given application instance is used by zero to many users (0,N connectivity for the link "*Application instance - uses*").



5.2.2.2 ACCESSES THROUGH

A given application instance *accesses through* zero to many CCN access interfaces (0,N connectivity for the link "*Application instance - accesses through*").

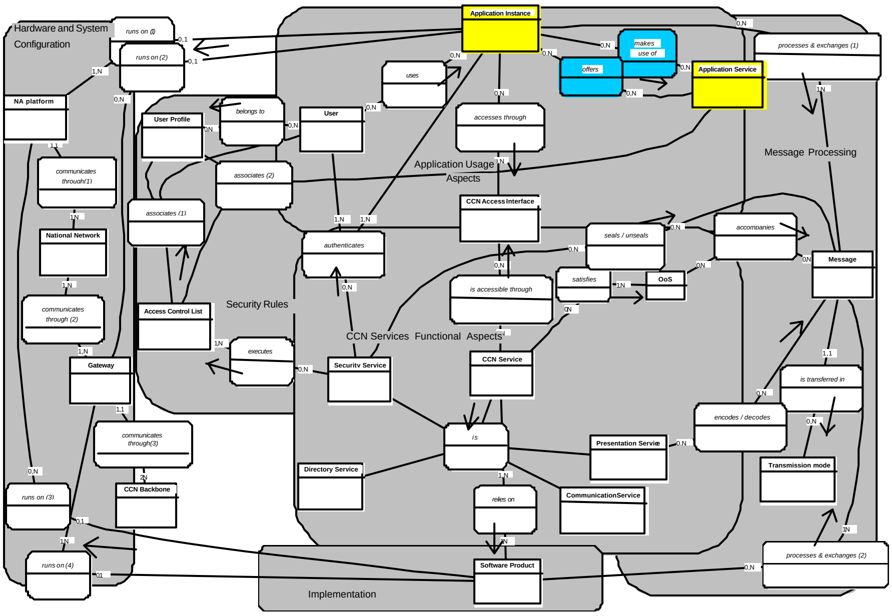
A given CCN access interface is used by zero to many application instances (0,N connectivity for the link "*CCN Access Interface - accesses through*").



5.2.2.3 OFFERS / MAKES US OF

A given application instance *is in relation with* zero to many application services (0,N connectivity for the link "*Application instance – offers*").

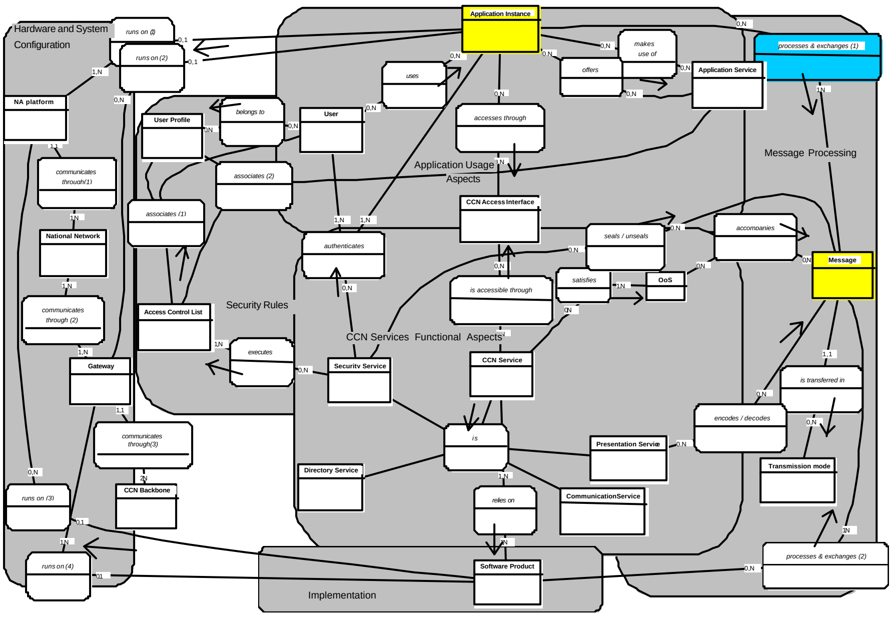
A given application service *is in relation with* zero to many application instances (0,N connectivity for the link "*Application Service – makes use of*").



5.2.2.4 PROCESSES / EXCHANGES (1)

A given application instance *processes* zero to many messages *and exchanges* them with other application instances (0,N connectivity for the link "*Application instance - processes & exchanges (1)*").

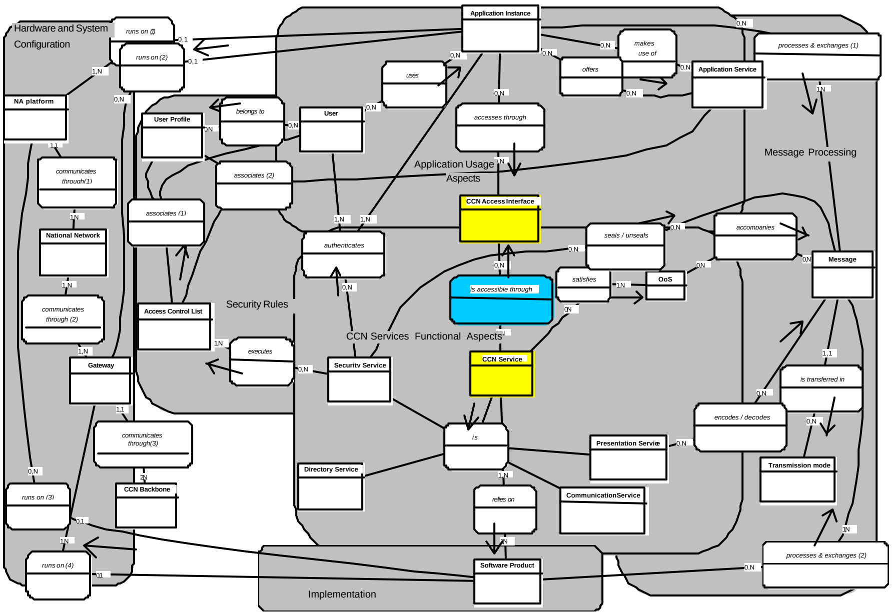
A given message is processed and exchanged by at least one application instance (1,N connectivity for the link "*Message - processes & exchanges (1)*").



5.2.2.5 IS ACCESSIBLE THROUGH

A given CCN Service *is accessible through* zero to many CCN Access Interfaces (0,N connectivity for the link "*CCN Service - is accessible through*").

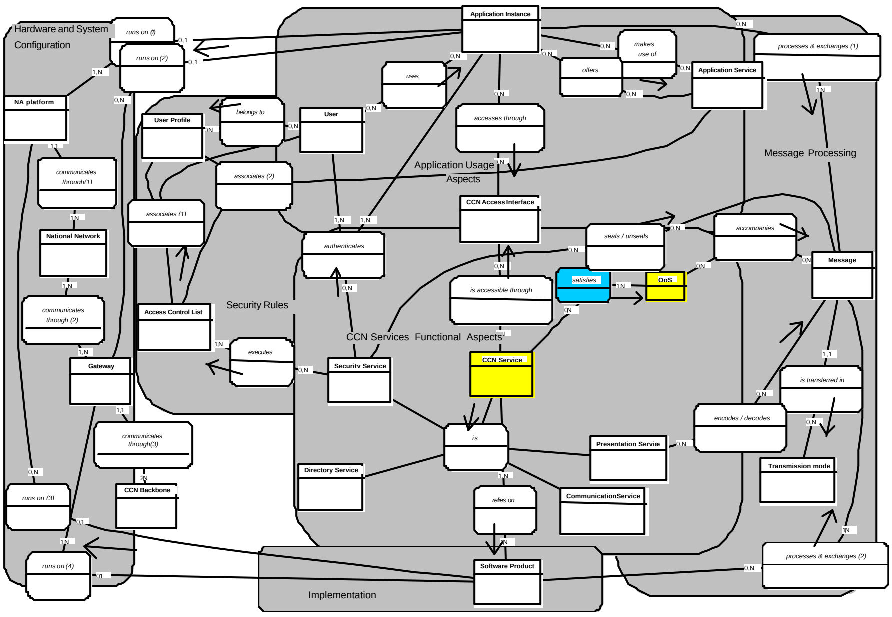
A given CCN Access Interface is used for access zero to many CCN Services (0,N connectivity for the link "*CCN Access Interface - is accessible through*").



5.2.2.6 SATISFIES

A given CCN Service *satisfies* zero to many Quality of Service (0,N connectivity for the link "*CCN Service - satisfies*").

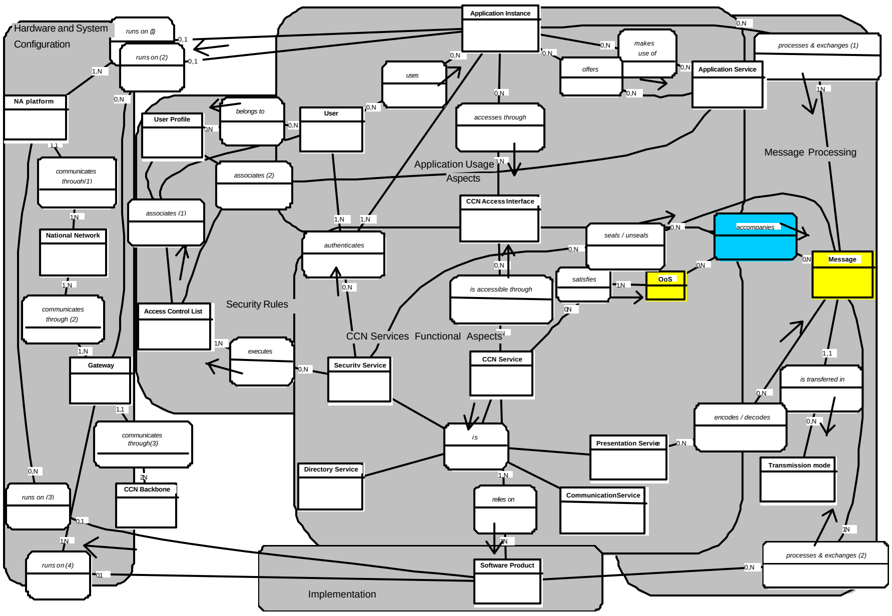
A given QoS is satisfied by one to many CCN Services (1,N connectivity for the link "*QoS - satisfies*").



5.2.2.7 ACCOMPANIES

A given Quality of Service *accompanies* zero to many messages (0,N connectivity for the link "*QoS - accompanies*").

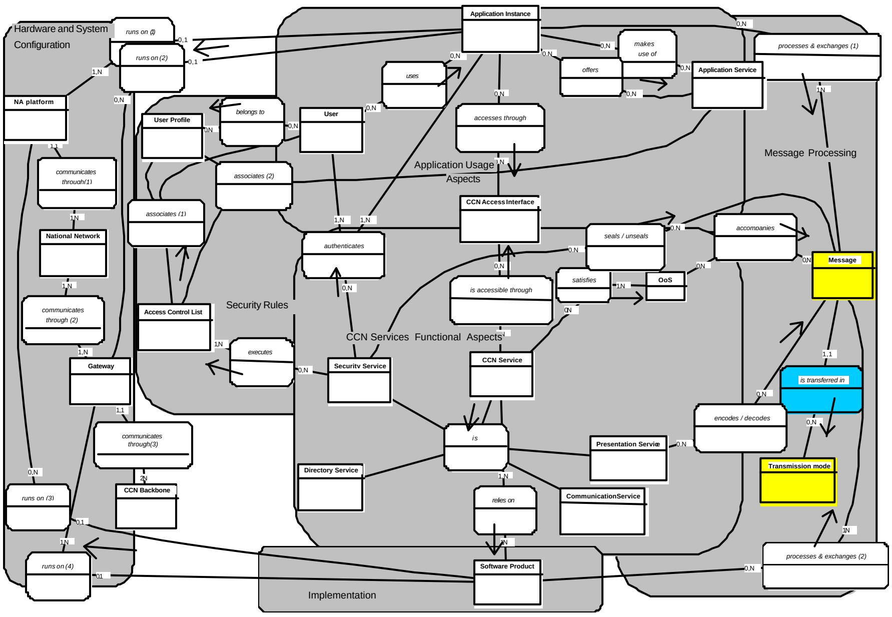
A given message is conveyed with one to many Quality of Service (1,N connectivity for the link "*Message - accompanies*").



5.2.2.8 IS TRANSFERRED IN

A given message *is transferred in* one and only one transmission mode: synchronous mode or asynchronous mode (1,1 connectivity for the link "*Message - is transferred in*").

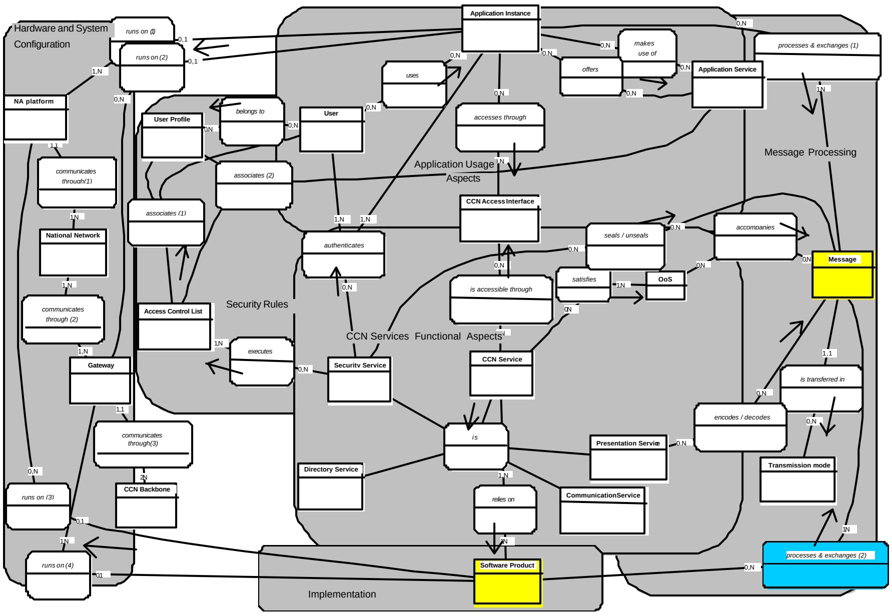
A given transmission mode is used for transfer zero to many messages (0,N connectivity for the link "*Transmission mode - is transferred in*").



5.2.2.9 PROCESSES / EXCHANGES (2)

A given software product *processes* zero to many messages *and exchanges* them with other software products (0,N connectivity for the link "*Software Product - processes & exchanges (2)*").

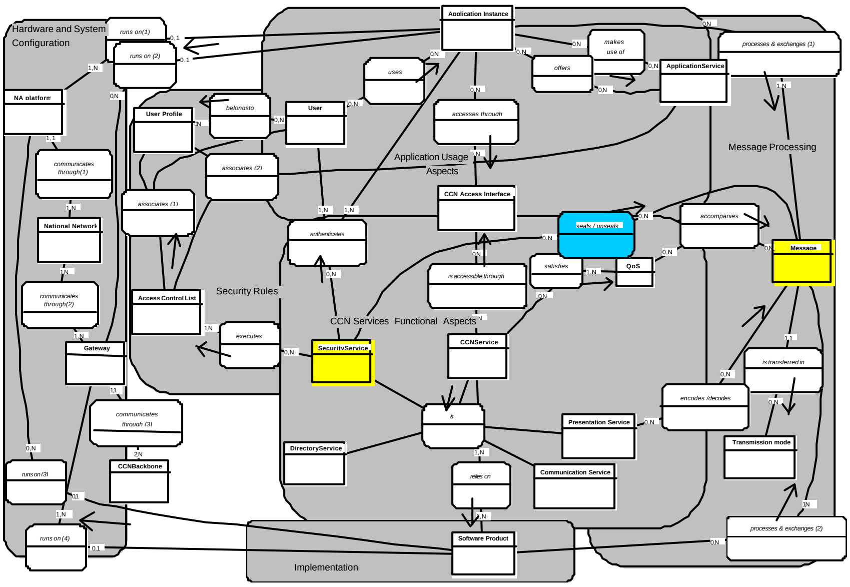
A given message is processed and exchanged by at least one software product (1,N connectivity for the link "*Message - processes & exchanges (2)*").



5.2.2.10 SEALS / UNSEALS

A given security service *seals or unseals* zero to many messages to insure integrity and confidentiality of the exchanged message (0,N connectivity for the link "*Security service - seals / unseals*").

A given message is sealed (unsealed) by zero to many security services (0,N connectivity for the link "*Message - seals / unseals*").

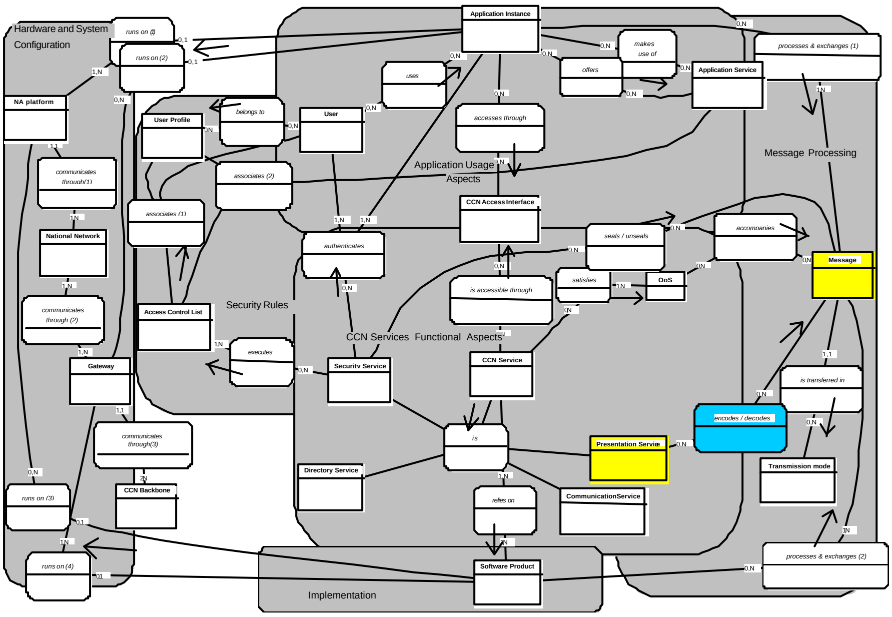


5.2.2.11 ENCODES / DECODES

A given presentation service *encodes or decodes* zero to many messages (0,N connectivity for the link "*Presentation service - encodes / decodes*").

A given message is encoded (decoded) by zero to many security services (0,N connectivity for the link "*Message - encodes / decodes*").

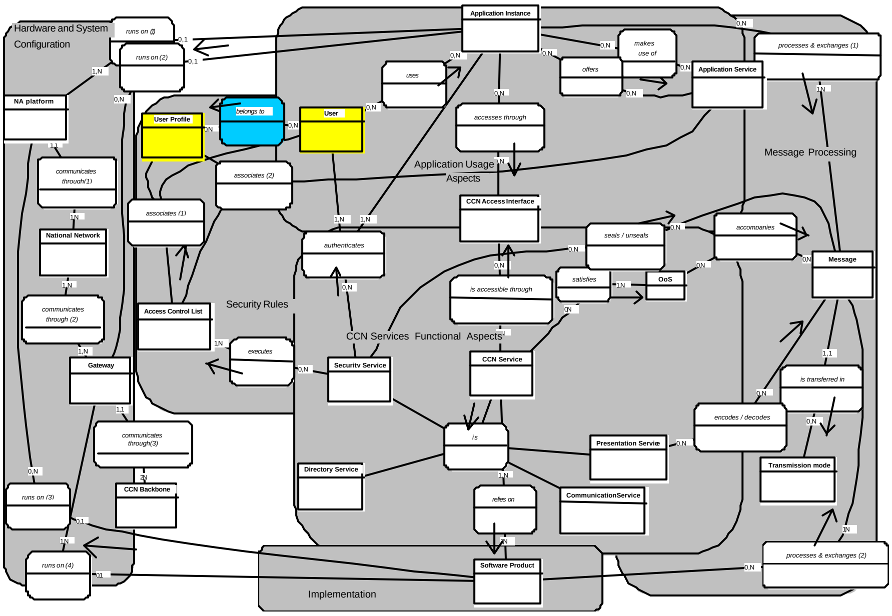
Remark: The relationship "*encodes / decodes*" involves also the relationship "*compress / uncompress*" which is not represented in the Entity-Relationship Diagram.



5.2.2.12 BELONGS TO

A given user *belongs to* zero to many user profiles (0,N connectivity for the link "*User - belongs*").

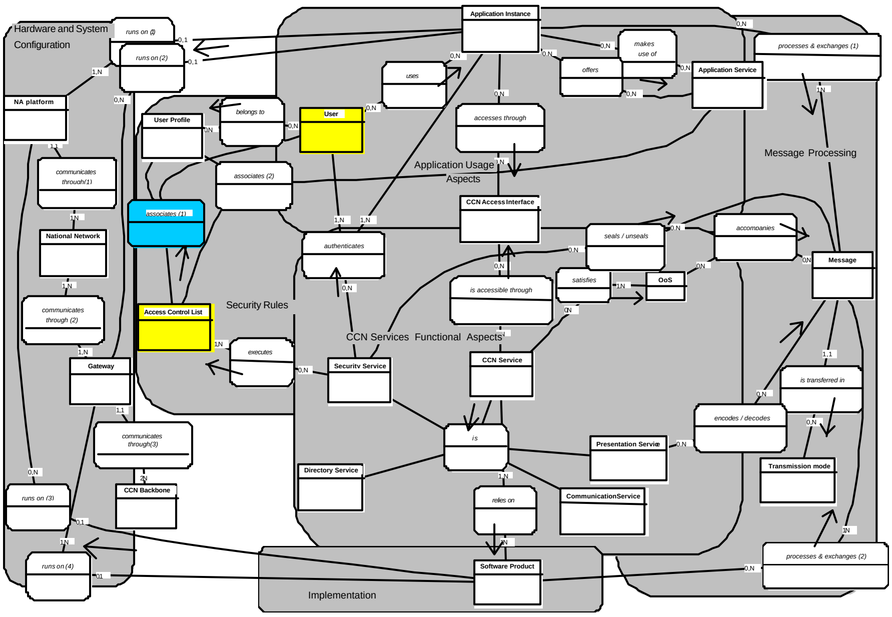
A given user profile owns zero to many users (0,N connectivity for the link "*User Profile - belongs*").



5.2.2.13 ASSOCIATES (1)

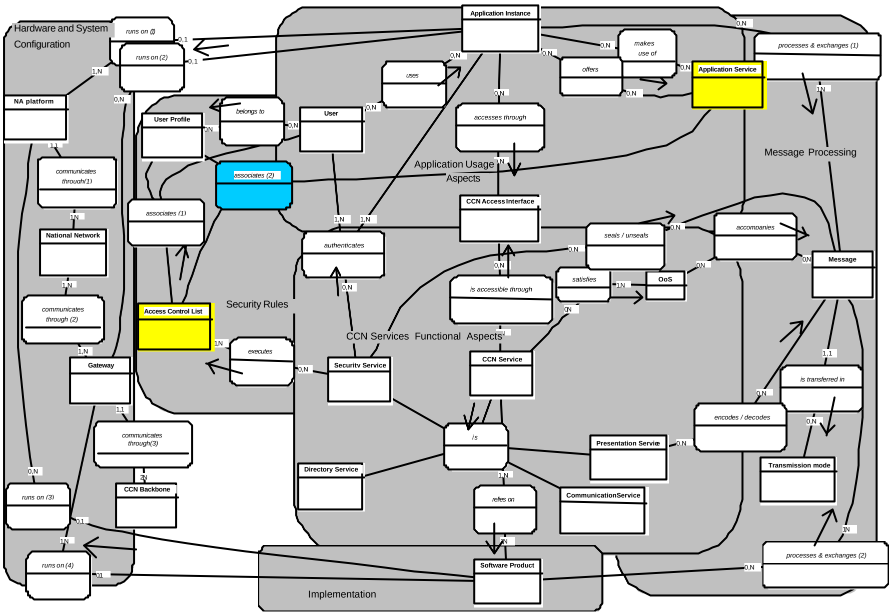
There are two types of ACLs.

The first type of ACLs *associates* each User Profile with zero to many Users. This is described by the above relationship "*belongs to*".



5.2.2.14 ASSOCIATES (2)

The second type of ACLs *associates* each User Profile with zero to many Application Services.

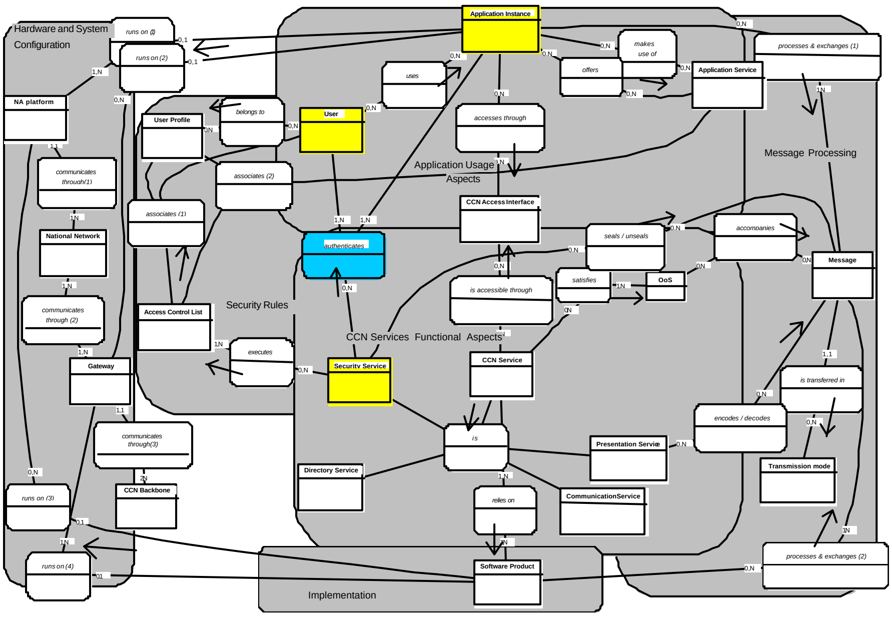


5.2.2.15 AUTHENTICATES

A given Security Service *authenticates* zero to many Users and zero to many Applications (0,N connectivity for the link "*Security Service - authenticates*").

A given User may be authenticated by one to many Security Service (1,N connectivity for the link "*User - authenticates*").

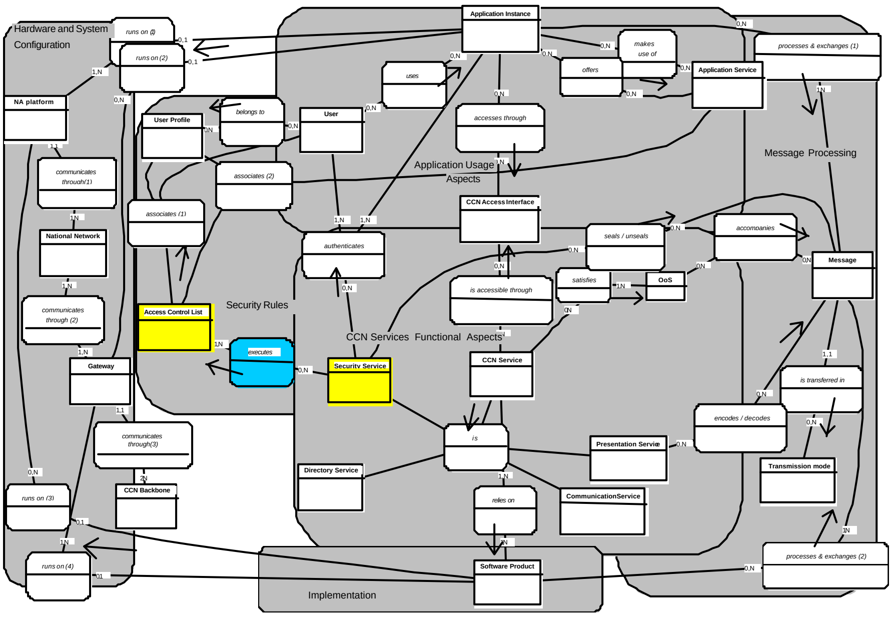
A given Application may be authenticated by one to many Security Service (1,N connectivity for the link "*Application - authenticates*").



5.2.2.16 EXECUTES

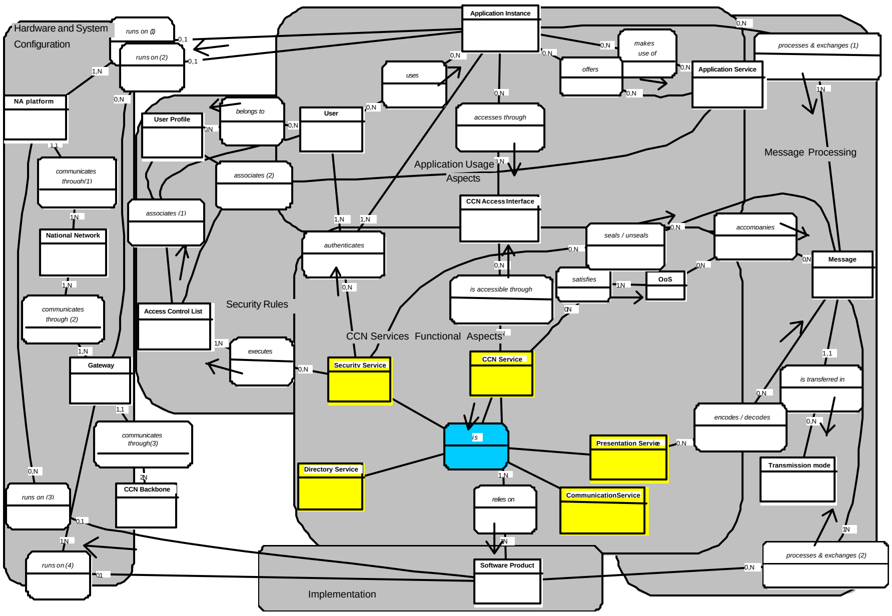
A given Security Service executes zero to many Access Control Lists (0,N connectivity for the link "*Security Service - executes*").

A given Access Control Lists is executed by one to many Security Services (1,N connectivity for the link "*Access Control List - executes*").



5.2.2.17 Is

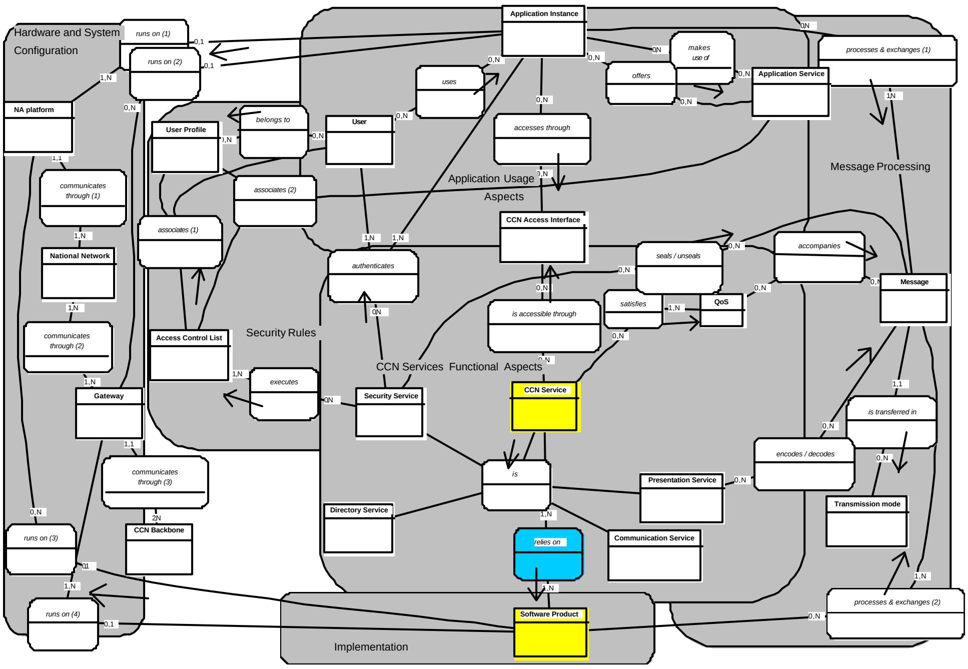
A CCN Service *is* one of the services presented on the Entity-Relationship Diagram: Security Service, Directory Service, Presentation Service and Communication Service.



5.2.2.18 RELIES ON

A given CCN Service *relies* on one to many Software Products (1,N connectivity for the link "*CCN Service - relies on*").

A given Software Product involves one to many CCN Services (1,N connectivity for the link "*Software Product - relies on*").



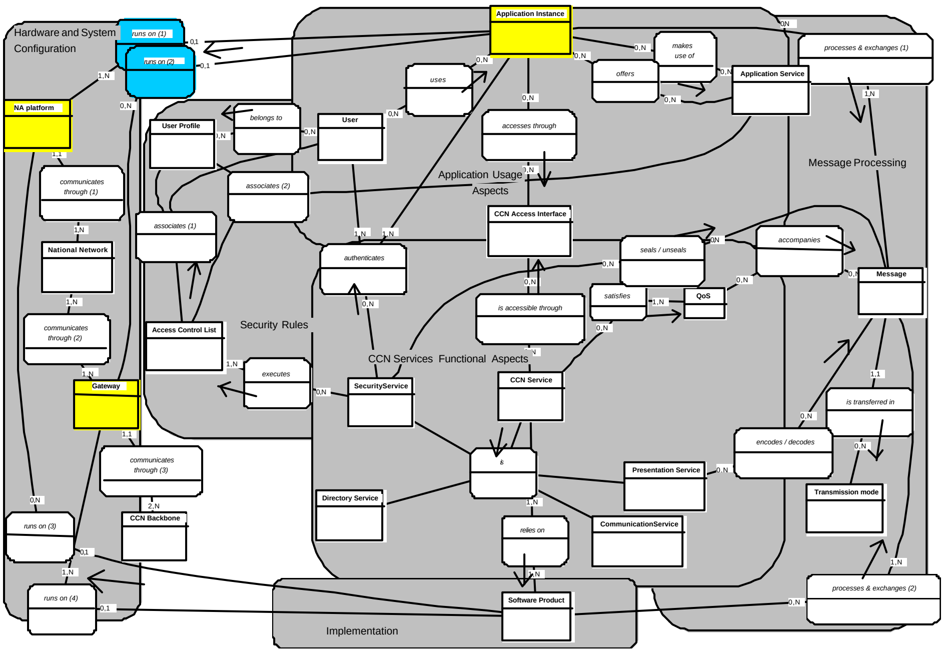
5.2.2.19 RUNS ON (1) - RUNS ON (2)

A given application instance *runs on* zero or one NA Platform (0,1 connectivity for the link "*Application instance - runs on (1)*") or it runs on zero or one Gateway (0,1 connectivity for the link "*Application instance - runs on (2)*"). This latter is called the Generic Application Services (GAS).

A given NA Platform supports one to many applications instance (1,N connectivity for the link "*NA platform - runs on (1)*").

A given Gateway supports zero to many application instances (0,N connectivity for the link "*Gateway - runs on (2)*").

<u>*Constraints*</u>: A given application instance runs on one NA Platform or it runs on one Gateway.



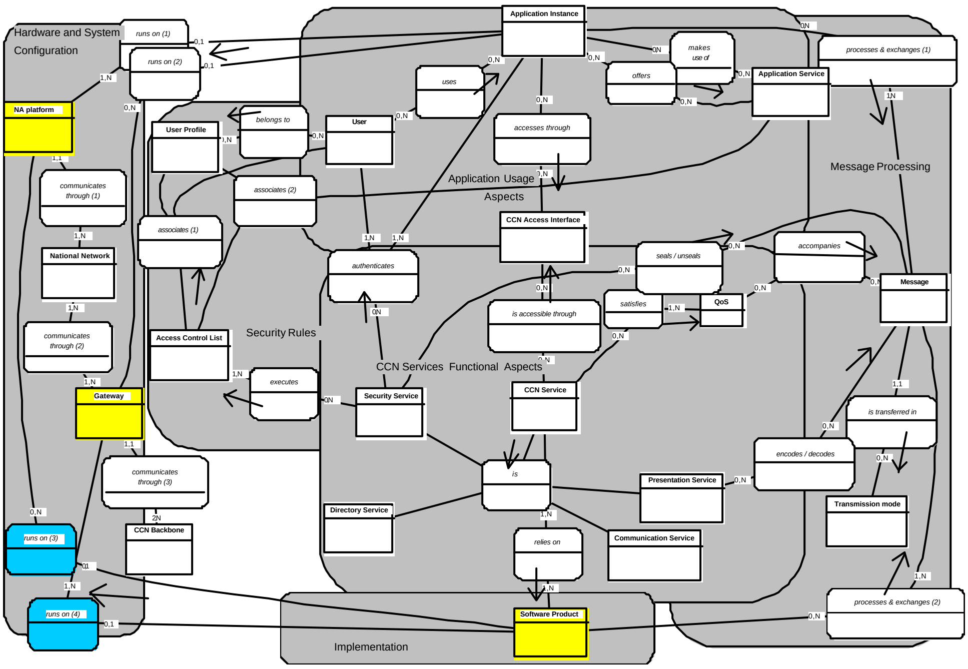
5.2.2.20 RUNS ON (3) - RUNS ON (4)

A given software product *runs on* zero or one NA Platform (0,1 connectivity for the link "*Software Product - runs on (3)*") or it runs on zero or one Gateway (0,1 connectivity for the link "*Software Product - runs on (4)*").

A given NA Platform supports zero to many software products (0,N connectivity for the link "*NA platform - runs on (3)*").

A given Gateway supports one to many software products (1,N connectivity for the link "*Gateway - runs on (4)*").

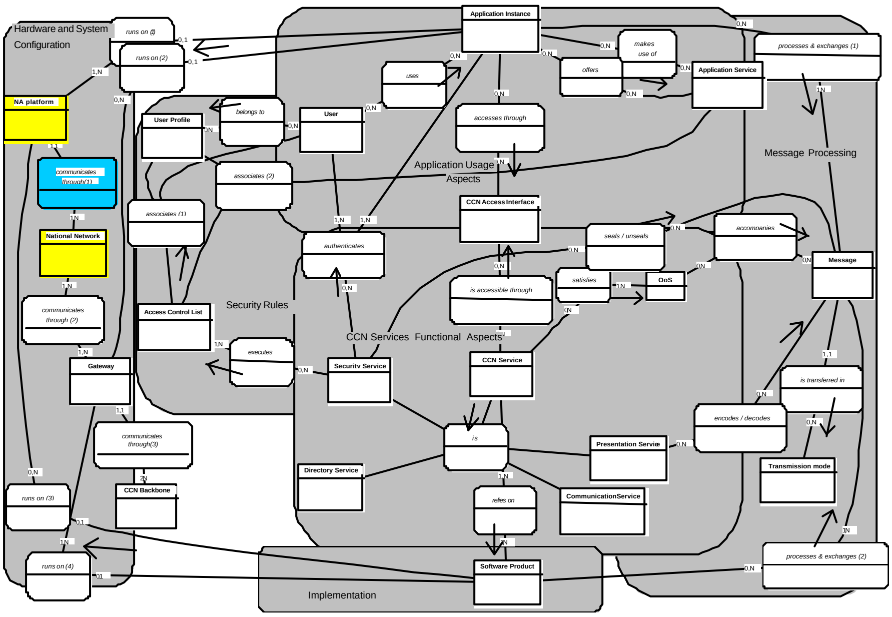
Constraints: A given software product runs on one NA Platform or it runs on one Gateway.



5.2.2.21 COMMUNICATES THROUGH (1)

A given NA platform *communicates through* one National Network to access the CCN Gateways (1,1 connectivity for the link "*NA Platform - communicates through (1)*").

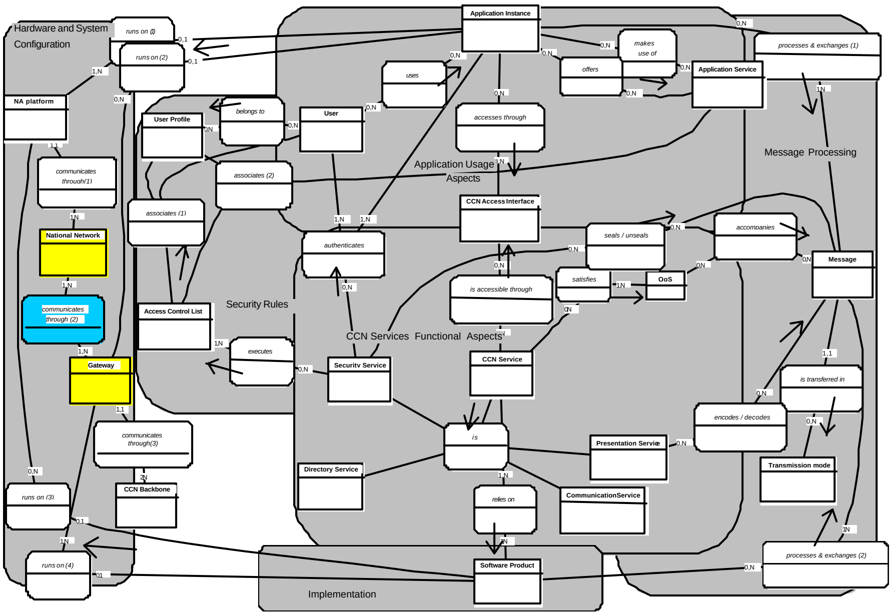
A given National Network is used by one to many NA platforms to communicate with Gateways (1,N connectivity for the link "*National Network - communicates through (1)*").



5.2.2.22 COMMUNICATES THROUGH (2)

A given Gateway *communicates through* one to many National Network to offer access point to the NA platforms (1,N connectivity for the link "*Gateway - communicates through (2)*").

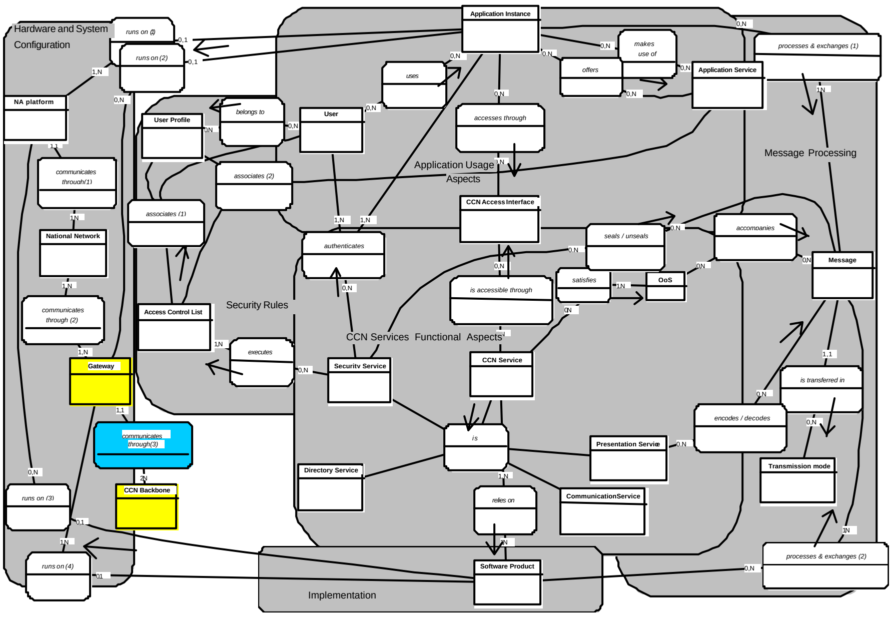
A given National Network may be used by one to many Gateways to communicate with NA Platforms (1,N connectivity for the link "*National Network - communicates through (2)*").



5.2.2.23 COMMUNICATES THROUGH (3)

A given Gateway *communicates through* the CCN Backbone with the other Gateway (1,1 connectivity for the link "*Gateway - communicates through (3)*").

The CCN Backbone is used by at least two Gateways that communicate together (2,N connectivity for the link "*Gateway - communicates through (3)*").



6. ANNEX: GLOSSARY

The table hereinafter contains the definitions of technical terms, abbreviations and acronyms used in the framework of the CCN/CSI project.

Acceptance Test Certificate (ATC)	Acceptance document certifying successful completion for each test session.
Acceptance Test Description (ATD)	Acceptance document containing the detailed description of the tests and the procedures for each acceptance testing activity.
Acceptance Test Plan (ATP)	Acceptance document outlining, for each testing activity, the framework of the tests to be made.
Acceptance Test Report (ATR)	Acceptance document containing the test logs, the fault reports and the minutes of the review meetings produced for each test session.
Acceptance Test Specification (ATS)	Acceptance document containing the specification of the test cases to be performed for each acceptance testing activity.
Access Control List (ACL)	Access Control List is the means of preventing access by non-authorised applications and users.
Administration Domain	National and Community Administration Domains. The national administration domain covers all the entities administered by the member State administration personnel. This domain also covers the whole national domain and part of the Community domain. The community administration domain covers all the entities that are centrally administered by DG TAXUD, either directly or by sub-contracting to a third party.
Administration Subsystem	Subsystem allowing the administration of the other subsystems composing the CCN/CSI system.
Administration Test Tools	Set of tools intended to test the correct working of the national and community Networks for administration purposes.
Apache	Apache is an open-source HTTP server. It provides a secure and extensible server that follows the HTTP standards. Please refer to § 4.3.1.1.3, § 4.4.3.3 and § 4.4.3.7.2 for more details on the main product's features used by CCN/CSI, as well as to the official web site of Apache: http://httpd.apache.org.

Application	An application is a program that runs on an NA Platform (or CCN Gateway) and that exchanges messages with other applications using the CCN/CSI infrastructure. (syn.: CCN/CSI Application)
Application Domain	An application domain is a domain that includes all the CCN/CSI applications that strive towards the same aim as well as the data directly managed by those applications.
Application Instance	An application instance represents the instance of a CCN/CSI application running on one given platform.
Application Language Message Definition	Structure definition and message identifier (in C or COBOL) to be included in the application.
Application Message Format	Physical representation of a CCN/CSI message as it is used by the application. The application message format depends on the application platform type.
Application Platform (AP)	Machine where the applications run on using the CCN/CSI service through the provided interface (e.g.: CSI). This term is synonymous of "Host" or "NA platform".
Application Programming Interface (API)	Programmatic interface of a software module providing verbs in a given language (C, COBOL) in order to enable the applications to invoke the services of this software module.
Application Service	Is a program (or a part of it) that performs, upon receipt of a synchronous request, an application function within a given application domain.
Architecture Design (AD)	Document describing the CCN/CSI architecture by means of "hierarchy" and "scenarios" (see [RD4]).
Asynchronous Community Communication Subsystem	This subsystem covers all the facilities put at the NA disposal by the European Commission in order for them to communicate between themselves in asynchronous mode. This subsystem is based on the features of the product MQSeries Distributed Queue Management
Asynchronous Mode	In this mode, an application sends a message without having established a connection with its peer. Messages are exchanged by placing and extracting them in queues.
Authentication	Authentication is a mechanism ensuring that a given entity owns its proclaimed identity..
Backup Gateway	This is a Gateway aimed at replacing the operational Gateway when the latter is unavailable.

Blocking Mode	In this mode, an application waits for the reply to a request, before continuing its processing. The contrary is the non-blocking mode.
Caching	This is an X.500 function that allows better performance by providing a cache and to avoid file access for each request.
CCN	Common Communication Network. The Common Communication Network is made up of a series of physical gateways located either in the National Administrations or on the DG TAXUD premises. These gateways are interconnected through their own communication services, and communicate with the Application Platforms.
CCN Access Interface Subsystem	This subsystem is the entry point offered to the applications or to the application platform in order to access and to use the CCN/CSI system
CCN/CSI Application	Synonym of “Application”
CCN/CSI Services	All the technical services provided by the CCN/CSI system (or infrastructure) to the applications.
CCN Directory Access (CDA)	CCN library used by the CCN software modules to interface with the CCN Directory.
Client Application	An Application that issues requests to Application Services.
Class of Traffic (COT)	One of elements of the Quality of Service (QoS) used to ask for a given level of service. The COT is especially used to put badly formatted messages to the appropriate dead-letter-queue.
C_API	This API is offered by the Communication layer (C). For example, on UNIX systems, this is the “socket” API.
Communication Layer (C Layer)	This layer is covered by commercial products allowing communication within the National Domain and within the Community Domain (between Application Platform and Gateway).
CO Communication	Community Communication Layer

CO Transmission	Community Transmission Layer
CO Function	Community Function Layer
Community Data Exchange Subsystem	This subsystem is responsible for the data transfer between two CCN/CSI Gateways..
Community Domain	It covers all the facilities put at the NA disposal by the European Commission in order for them to communicate between themselves.
Component	A Component is a part of a Software Product (process or API) and is the smallest entity described at architecture specification level.
Component Functional Specification (CFS)	Document describing the functional details of a Component.
Confidentiality Mechanisms	Means to ascertain the non-disclosure of the sent/received information.
Confirm on Arrival	Issued to the sending application when a message is placed in the destination Queue Manager (destination GW)
Confirm on Delivery	Issued to the sending application when an application retrieves a message in a way that causes a message to be deleted from the queue.
Connection Oriented Mode	This mode requires the two applications to establish a logical connection with each other before communication can take place. It is also called Synchronous Transmission Mode (blocking or non-blocking). It supports both request/response mode and conversational mode.
Credential Information	Information like user ID, application ID, user Password, application secret key used by the application to build an authentication token.
CSI	Common System Interface used by the applications in order to exchange information through CCN.
CSI Services	All the Application services relying on the CSI.

CSI Stack	They are two types of stack: AP CSI, and Gateway CSI stack. The AP CSI is located on the Application Platform and the other on the Gateway.
CSI_API	Is the Application Programming Interface provided by the low level functions in order to access the CSI services.
CT_API	This API is offered by the Communication dependent Transmission layer (CT). It is the lowest communication protocol independent API.
CT layer	The Communication Specific Transmission Layer is part of the Data Transfer Functional Subsystem and corresponds to the lower part of the Transmission layer (the upper part being the GT layer).
Data Flow Diagram (DFD)	Standard model used to split CCN/CSI system into objects and to formalise the data flows.
Data Presentation Subsystem	This subsystem provides the means to convert the data exchanged between heterogeneous platforms.
Datagram	Asynchronous message for which no reply is expected.
Directory	This Software Product allows the storage of data of the configuration information used by the different Software Products. It is a central point in the CCN/CSI system where configuration information shared by CCN/CSI subsystems is stored.
Directory Access Protocol DAP	This protocol describes how a Directory is accessed.
Directory Information Base (DIB)	The database of configuration items in the X.500 system (typically the CCN Directory).
Directory Information Shadowing Protocol (DISP)	The protocol used to replicate and synchronise the CCN Directory content.
Directory Information Tree (DIT)	Structure of the Directory content, describing the nodes and associated attributes.
Directory System Agent (DSA)	The X.500 program that looks up the location and definition of a configuration item in a Directory Information Base (DIB). It accepts requests from the Directory User Agent (DUA).

Directory System Protocol (DSP)	The protocol used to control the interactions between two or more Directory System Agents.
Directory User Agent (DUA)	A X.500 function of software module that sends a request to the Directory Server Agent (DSA) to look up the location of a resource on the network.
DMZ	This acronym stands for DeMilitarized Zone. In a DMZ configuration, most computers on a network run behind a firewall connected to a public network like the Internet. One or more computers also run outside the firewall, in the DMZ. Those computers on the outside intercept traffic and broker requests for the rest of the LAN, adding an extra layer of protection for computers behind the firewall.
Enciphering/Deciphering	Security mechanisms activated by the applications via the QoS. They allow confidentiality of the exchanged data.
Entity Relationship Diagram	Formalism used in order to represent the CCN/CSI data model described in the FRS.
Exception	Occurs when a permanent condition prevents delivery of a message to its destination queue and the application cannot be notified synchronously of the problem/error.
Exception Notification	An Exception notification is a notification message that reports an exception (i.e. problems and errors.)
Exchange Co-ordination Subsystem	This subsystem allows the scheduling of the services offered by other subsystems (Data Presentation, Security, and Routing/Addressing) in order to ensure their activation in a consistent way.
Expiry Time	Expiry time is defined as the time when a message has to be discarded prior to its delivery to an application because its time has expired.
F,T,C	Abbreviations emanating from the CCN/CSI Feasibility Study for the Functional, Transmission, and Communication Layers.
Factory Acceptance Test (FAT)	Acceptance session whose purpose is to validate that the implemented CCN/CSI system meets the requirements.
Final System Acceptance (FSA)	Acceptance phase, checking that operational quality criteria are fulfilled.

Function (FU)	A function is a part of a Functional Module and fulfils a well-defined handling process.
Function Layer	Defined in the CCN/CSI feasibility Study as the upper layer of the F,T,C stack. It provides for example the HL_API and CSI_API to the NA applications.
Functional Module (FM)	A Functional Module is a part of a Functional Subsystem. (e.g.: Authentication in the security sub-system).
Functional Requirement Specification (FRS)	Functional Requirement Specification document containing the CCN/CSI requirements and the CCN/CSI functional description.
Functional Subsystem	A Functional Sub-System is part of the system. (e.g.: Administration).
Functional System Specification (FSS)	Functional System Specification document containing the CCN/CSI functional description.
Generic Application Services (GAS)	The GAS runs on the Gateways on the Community Domain. They use the CSI API. They implement Generic Application Services accessible by any user application.
Generic CSI	Part of the software relative to CSI that is not impacted by different platform environments.
Grouping	Function of the T layer offered in the Community Domain in order to group short messages.
GSS	Generic Security Service is a generic mechanism used to perform client/server authentication.
GSS_API	Security API making part of the function layer and provided by the Security subsystem.
GT_API	This API is offered by the generic Transmission layer (GT). It is the highest communication protocol independent API.
GT layer	Generic part of the Transmission layer in the National Domain offering the T services.
Hierarchy	Formalism used in the Architecture Specifications for the breakdown of the CCN/CSI system.

HL_API	HL_API is the High Level API that provides access to CSI on the Application Platform. The Function Layer provides the HL_API.
HTTP protocol	HTTP is a generic stateless object-oriented protocol. A feature of the HTTP protocol is the negotiation of data representation.
IAB	Inter Application Bus corresponding to the CCN/CSI system.
IDA	Interchange of Data between Administrations
IMAP	IMAP stands for **I**nternet **M**essage **A**ccess **P**rotocol. It is a method of accessing electronic mail or bulletin board messages that are kept on a (possibly shared) mail server. In other words, it permits a "client" email program to access remote message stores as if they were local.
Initiator/Responder Tools	Set of tools installed on the CCN gateway and used to test a remote CCN/CSI application, acting as the peer entity during the exchanges.
Integrity Mechanisms	Means to ascertain the non-alteration of the transmitted information.
Internal Message Definition Syntax	Definition syntax of a message resulting from the compilation of the "message formal definition" and which is used as input by the Encoding/Decoding tool.
Internal Message Syntax	Internal message syntax definition corresponding to formal definition of a message. This definition is stored in the directory.
LDAP	Lightweight Directory Access Protocol. A simpler version of the DAP protocol.
LCMS	Local CCN/CSI Mail Server: this is a mail server running SuSE Open Exchange Server.
LI	Local Intelligence is a piece of software located on the gateway. It is responsible for the co-ordination during the processing of the messages
LL API	Set of API called Low Level API including: GSS_API, PRES_API, CSI_API.

Local Queue	A local queue is a Message Queue located on the Gateway, used to store the exchanged messages when the Asynchronous Exchange Mode is used.
Message formal definition	Message structure definition in a specific formal language (such as: IDL, ASN.1, EDIFACT).
Message formal definition compiler	Software that analyses input message formal definitions and produces the corresponding output internal message syntax.
Message Queuing Mode	Asynchronous sending mode provided by the CCN/CSI services. No connection between application processes has to be established before sending the messages. Messages are exchanged using queues. This transmission mode is also called Asynchronous Mode.
Module	A module is a part of a component identified in the High Level Design of the CCN/CSI components (or software products).
MQI	Message Queuing Interface. Asynchronous API used to exchange messages in Message Queuing Mode.
MQSeries	MQSeries is a messaging middleware that allows programs to communicate with each other across a variety of platforms, including UNIX systems. It is used in the CCN/CSI project to provide the asynchronous services for the CSI applications. MQSeries is a product of IBM. Please refer to § 4.3.1.1.4 and § 4.4.3.7.3 for more details on the main product's features used by CCN/CSI, as well as to the official web site of MQSeries: http://www-306.ibm.com/software/integration/wmq/ .
MSA	Member State Administration.
Naming Rules	Set of rules that allow the handled entities to be named and uniquely identified by the CCN/CSI system, e.g.: Application, Script, and Service).
NA	National Administration.
NA Platform	Synonym of Application Platform

National Communication layer (NA Communication) C-NA	Identify the required software in order to provide the appropriate standard protocols for a given platform.
National Data Transfer	This subsystem provides the transfer means used to exchange information between the Application Platforms and the related CCN/CSI gateway. It is distributed on the Application Platform and the CCN/CSI Gateway.
National Function layer (NA Function) F-NA	Its the function layer relative to the Application platform and the GW to interface
National Transmission layer (NA Transmission) T-NA	The transmission layer used for communication between the Application Platform and the local Gateway.
Non-Secure Link	A connection on the CCN Gateway belonging to an Application that is supported by an "autonomous system" type PC/ Workstation (UNIX) environment.
Notification	Type of message sent to inform the originator of the result of its message.
OS Adaptation Layer	This subsystem provides, for portability purpose, services to the Application Platforms subsystems permitting to mask the dependencies of system dependent operations.
OS_API	Application Programming Interface for the operating system that allows CSI software to be independent of the environment.
Pivot Message Format	Physical representation of a CCN/CSI message as it is used in the Community domain between gateways and in the National domain between PC, UNIX platform and gateway. Pivot message format is seen as a particular typed message format.
POP3	Stands for Post Office Protocol version 3. It is intended to permit a workstation to dynamically access a maildrop on a server host.
Postfix	Postfix is a mail server written originally by Wietse Wenema. It is fast, easy to administrate, secure and most of the time compatible with Sendmail's functionalities.
Presentation API (PRES_API)	Presentation API is part of the function layer and provided by the Presentation sub-system.
Provisional Site Acceptance Tests (PSAT)	Acceptance session, whose purpose is to demonstrate that the system delivered, is operational on the site architecture. PSAT sessions are organised for a set of identified pilot sites and for all deployment sites.

Quality of Service (QoS)	The value of the Quality of Service leads to the determination of services required by the application. The QoS conveyed with a message contains information about confidentiality and integrity of the message, urgency of the message, notifications to send to the application, compression to apply to the message body.
Reference Value	The Reference value is used within the security subsystem. It is stored in the Directory and combined with the Security key of the gateway in order to obtain the private key of the user or applications.
Remote API Proxy (RAP)	Software on the Gateway that works in accordance with the AP CSI stack that provides the remote execution of the CSI verbs on the gateway. There is one RAP client (or server) process per one connected client (or server) on the Gateway.
Remote Queue	Message Queue on a distant Gateway in a different domain than the sending application domain (used mainly with the Asynchronous Exchange mode).
Replication	Replication is the duplication of part of the DIB on several DSA. In CCN/CSI only shadowing is used (no caching in the DSA).
Replication parameters	Configuration parameters used to perform the replication.
Requirement	Defines a particular expected behaviour of the system. The set of CCN/CSI requirements is defined in the FRS document.
Routing/Addressing	This subsystem is responsible for the routing/addressing resolution of the messages sent by the applications to their destination. For example, it associates a logical name of service to a destination
RPC	Remote Procedure Call. This mechanism is implemented in the CCN/CSI Project to transfer the execution of a verb from a NA Platform to a CCN Gateway.
Sample Applications	Programming samples, written in C, Java or in COBOL, provided to CCN/CSI programmers.
Scenario	A description of the sequence of the components involved in the processing of a particular input message. It is used in the Architecture Specifications.

Sealing/Unsealing	Security mechanism that can be activated according to the QoS and link security context, that is applied to the exchanged data in order to achieve the authentication of the data origin and data integrity. See also Enciphering/Deciphering, Authentication, ACL in this glossary that are other security mechanisms.
Secure Link	A secure link is a connection between an Application Platform supported by a "mainframe" environment with an intrinsic security structure and an environment that is protected by a high security level and a local CCN Gateway.
Security Subsystem	This subsystem provides the security services, including access control, authentication, integrity and confidentiality. It provides the means of ascertaining the identity of the peer application, the non-disclosure and the non-alteration of the sent/received information.
Sequencing	Sequencing is the property that allows the data to be received by the receiver in the same order as the sender issued it.
Server Application	A server application runs on an AP and aims to provide a service to the client applications.
Service	The service is the request processing implementation in an application instance that is invoked when a synchronous message arrives.
Service Level Agreement (SLA)	Agreement between the CCN/TC and the DG TAXUD for exact attribution of administrative responsibilities of the NA and the CCN/TC.
Service Provider Interface stack (SPI)	The Service Provider Interface stack is the peer stack software provided on the gateway that allows the Remote API proxy to dialogue with the CSI stack on the Application platform.
SMTP	SMTP stands for Simple Message Transfer Protocol. The objective of this protocol is to transfer mail reliably and efficiently. It is described in RFC 2821.
Software Product (SP)	A Software product is part of a Technical Sub-System. It is described at Architecture Specification Level.
Specific Connector	The main characteristic of the Specific Connector architecture is its protocol stack that is implemented for dialogue with the AP. It does not conform to CCN/CSI layered architecture. A specific connector on the Gateway is provided through a published interface.

Specific CSI	This is a part of CSI software and is dependent on a particular platform (i.e.: it differs for each platform).
SSL	Stands for Secure Socket Layer. It is a protocol developed for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data - a public key known to everyone and a private or secret key known only to the recipient of the message.
State Transition Diagram (STD)	State Transition Diagram is the formalism used in the CFS document in order to specify the Protocol mechanism related to a Component.
SuSE LINUX Open eXchange Server (SLOX)	SLOX is an inter-personal messaging system running on the SuSE LINUX operating system. It is SMTP compliant. SuSE LINUX Open Exchange Server is a product of Novell. Please refer to § 4.3.1.1.5, § 4.4.3.4 and § 4.4.3.7.4 for more details on the main product's features used by CCN/CSI, as well as to the official web site of SLOX: http://www.novell.com/documentation/suseslox/ .
Synchronous Community Communication Subsystem	This subsystem covers all the facilities put at the NA disposal by the European Commission in order for them to communicate between themselves in synchronous mode. This subsystem is based on the features of the product Tuxedo /DOMAIN.
Synchronous mode	In this mode an application has to establish a connection before sending a message (in a blocking or non-blocking mode). Another term is: Connection Oriented Mode.
Test Tools	Sample Applications, Initiator/responder Tools and Administration Reserved Tools
Tests Subsystem	This subsystem provides the necessary means to perform test of the applications and of the CCN/CSI system
Transaction	Any set of operations that must be completed as part of a unit.
TP monitor	Transaction Processing monitor. In a distributed client/server environment, a TP monitor provides integrity by ensuring that transactions do not get lost or damaged. TP monitor guarantees that all transactional resources are updated from a single transaction.
Transmission API (T_API)	T_API is the Transmission API that provides access to the Transmission layer.

Transmission Layer	Transmission Layer is part of the FTC stack defined in the Feasibility Study and includes: Generic Transmission layer (GT), Communication specific Transmission layer (CT)
Transmission Mode	Designates the Asynchronous Mode and/or the Synchronous Mode
TCP/IP	Transmission Control Protocol/Internet Protocol
Tuxedo	Tuxedo is one of the major TP monitor in the UNIX client/server environment. It is used in the CCN/CSI project to provide the synchronous services for the CSI applications. Tuxedo is a product of BEA Systems, originally developed by AT&T. Please refer to §4.3.1.1.1, §4.3.1.1.2 and §4.4.3.7.1 for more details on the main product's features used by CCN/CSI, as well as to the official web site of Tuxedo: http://edocs.bea.com/ .
Typed Message Format (TMF)	Is an application message format where an identifier of message type (also called Header) has been associated. The typed message format depends of the application platform type.
Typed message format converter	Program converting a typed message format to another typed message format, using the internal message syntax and knowing the physical representation conversion rules between the source and target formats.
Uniqueness	Uniqueness corresponds to the property that allows an application to receive a successful issue message just once.
Unit Process (UP)	A unit process is part of a function that has been defined in order to be part of a component.
User	The "user" represents the entity using an application. This entity is implied in the authentication with the CCN/CSI security services. A user can be a human user or a logical user. Generally, human users use client applications on workstations and server applications are used by logical users.
User Profile	A profile is associated to a user and an identity to the operation that can be performed.

Workprocess	Is the formalism used in the CFS in order to describe the sequence of UPs for a particular Component.
XATMI	X/Open Application Transaction Monitor Interface. This interface is used in the Project for synchronous exchanges of messages.

END OF DOCUMENT