D1.1 DESCRIPTION OF THE SCENARIOS AND THEIR REQUIREMENTS

A. Armenteros (TID), B. Chetali (GTO), M. Felici (DBL), V. Meduri (DBL), Q-H. Nguyen (GTO), A. Tedeschi (DBL), F. Paci and E. Chiarani (UNITN)

Document information

Document Number	D1.1
Document Title	Description of the Scenarios and their requirements
Version	1.4
Status	Final
Work Package	WP 1
Deliverable Type	Report
Contractual Date of Delivery	31/01/2010
Actual Date of Delivery	31/01/2010
Responsible Unit	TID
Contributors	DBL,GTO, UNITN
Keyword List	Security, Evolution, Case Studies, Requirements
Dissemination level	PU

Document change record

Version	Date	Status	Author (Unit)	Description
0.1	05/06/2009	Draft	A. Armenteros (TID)	Initial document structure
0.2	15/06/2009	Draft	A. Armenteros (TID)	New section added
0.3	06/08/2009	Draft	V. Meduri, M. Felici, A. Tedeschi (DBL)	ATM inputs added, Structure of sections Modified
0.4	21/08/2009	Draft	A. Armenteros (TID)	HOMES inputs added.
0.5	02/11/2009	Draft	B. Chetali, Q-H. Nguyen (GTO)	POPS inputs added.
0.6	03/11/2009	Draft	A. Armenteros (TID)	Integration of latest contributions in ATM, POPS and HOMES.
0.7	03/11/2009	Draft	V. Meduri, M. Felici, A. Tedeschi (DBL).	General review of ATM Scenario.
0.8	05/11/2009	Draft	V. Meduri, M. Felici, A. Tedeschi (DBL).	Review of the ATM Scenario after TID comments.
0.9	19/11/2009	Draft	A. Armenteros (TID)	Review of ATM, POPS and HOMES after comments. Introduction and executive summary added. Editing work.
0.10	20/11/2009	Draft	B. Chetali (GTO)	Review and update POPS chapter after comments.
0.11	10/12/2009	Draft	F. Paci (UNITN)	UNITN Feedback
0.12	10/12/2009	Draft	A. Armenteros (TID)	Some corrections

				after UNITN review.
0.13	21/12/2009	Draft	A.Armenteros (TID)	Small modifications within HOMES case study.
0.14	21/12/2009	Draft	V. Meduri (DBL)	Alignment of the version number
0.15	21/12/2009	Draft	B.Chetali (GTO)	Review and update POPS parts
1.0	22/12/2009	Draft	A.Armenteros (TID)	Ready for Quality Check
1.1	08/01/2010	Draft	E.Chiarani (UNITN)	Quality Check completed. Minor remarks
1.2	11/01/2010	Draft	A.Armenteros (TID)	Changes to fix all the issues detected in Quality Check first pass.
1.3	18/01/2010	Draft	E.Chiarani (UNITN)	Final Quality Check
1.4	19/01/2010	Final	A.Armenteros (TID)	Consolidated version ready to be submitted

Executive summary

This document provides the description of each Case Study considered within the Secure Change project, including details on each Case Study context (motivation scenarios involved technologies, etc.); a first view on changes and evolution related issues and finally a list of attached requirements is provided. Case Studies in WP1 are HOMES (led by TID), POPS (led by GTO) and ATM (led by DBL).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

This document present the three relevant case studies included in Secure Change as vehicles for demonstrating the advantages and benefits coming from the research in this project. The three use cases are led by industrial partners and provide adequate scenarios where changes and evolution related issues arise. Secure Change's technologies come to cope with such issues.

The three case studies lie on different complementary domains, in the aim of showing the Secure Change approach in the widest range possible. They are representative of relevant but not exclusive application domains of Secure Change output. The case studies, named according to their domains, are these ones:

- Air Traffic Management case study (ATM)
- Home Network case study (HOMES)
- Smart Card case study (POPS)

Each case study follows the same schema for presenting its contents: a full description of the application domain, motivation scenarios, and involved technologies is provided first, followed by a section stressing the change and evolution related issues to the case study and finished with a compilation of requirements for the scenarios.

2 Common requirement collection template

2.1 Introduction

As agreed by all the participants in the WP1, the Case Study Requirements should be expressed in natural languages, without a pre-defined, structured modelling, as the ones proposed by WP3 partners. This will be object of future work during the second year of Secure Change project. At this stage, a more general description is required, as an input for the other WPs. This input shall be the basis for other technical groups that will work on formalized descriptions using specific modelling techniques suitable for their purposes.

Nevertheless, to provide a homogeneous description for each Case Study, a common requirement template has been developed with the aim of providing general format, valid for all the Case Studies, but also enough descriptive to work as a basis for any formalization process.

Each Case Study presents its requirements following this template.

2.2 Description of the template

2.2.1 Requirements Format

Requirements are expressed in natural language (English). To avoid misunderstandings it is important to specify that convention on the meanings complies with the following criteria:

- “shall” is used to indicate mandatory requirement
- “should” is used to indicate a recommended requirement (but not mandatory),
- “may” is used to indicate an option (it depends form environment/procedures).

“The System” depends on the context and refers to the actual component/tool under analysis.

3 ATM Case Study

3.1 General Description

The ATM (Air Traffic Management) is an interesting case of complex socio-technical domain in which people must cooperate with each other and with technologies in order to achieve their goals.

The ATM domain involves an aggregation of Services provided by ground-based Air Traffic Controllers (ATCOs), see Figure 1. One of the main critical responsibilities of ATCOs is to maintain horizontal and vertical separation among aircraft. They must ensure an orderly and expeditious air traffic flow, by issuing orders and directions to aircraft and by providing flight context information to pilots, such as routes to waypoints and weather conditions.

Figure 1 - Tower Air Traffic Controllers at work.

3.1.1 ATM S&D Requirements

Air Transportation Systems are parts of critical infrastructures in modern societies. The main security aspects of Air Transportation Systems concern preventative security measures that take place during the airport operations and within the airlines operations. The Air Traffic Management system is also one of the key components of Air Transportation Systems, with a major role in “the safety, regularity and efficiency of air navigation” and also in its security.

Main aspects of security in ATM relate to self protection of facilities and resources of the ATM system as well as coordination with Air Defence authorities for exchange of information and coordination in case of aviation security incidents. The ATM is above all a cooperative system, based on mutual trust primarily between airspace users and ATM staff. Traffic surveillance relies currently on sensors that can bring additional confidence to the integrity of information received (i.e.: processing of surveillance information from multiple sensors). Surveillance of traffic and monitoring of information may be used to detect civil aircraft operating in such a manner as to raise suspicion of seizure by terrorists or hijackers.

Critical Security & Dependability (S&D) aspects are present in the ATM domain:

(a) The ACC socio-technical system must be resilient with respect to unexpected and unplanned situations in order to give a prompt and effective response;

(b) The decisions must be taken in a short amount of time guaranteeing absolute safety of flights under any possible circumstance;

(c) The services provided by the ACC socio-technical system must be reliable, accessible and available 24 hours per day, 7 days / 7;

(d) Only duly authorized personnel can access the Control rooms and communication between ACCs/ATS Units must be secured (nowadays dedicated communication links are adopted). In facts, the security of data about aircraft position and the robustness against unintentional or malicious data corruption are key elements to guarantee efficient, safe and reliable ATM Services and to keep the confidence of the general public on air transportation.

Other important characteristics of the ATM domain as it is today are that:

- There is a limited interaction with external world, thus limited ‘classical’ security problems;
- Human are at the centre of the decision process, with limited role of automated system;
- Current safety problems are mainly due to human errors, air-ground communication problems and degradation of technical and human services combined with adverse atmospheric conditions could raise safety troubles.

However, the introduction of new systems and the reorganization of ATM services are facing security issues. Both *ATM security and safety* are concerned with securing the ATM assets and services and on the *Airspace Security*, that seeks to safeguard the overall airspace from unauthorised use, intrusion or other violations are raising new challenges. In fact, the widespread deployment of innovative information system technologies at every stage of the Air Transport value chain (from ticket purchase to management of flight) raises major security concerns with regard to the vulnerability of these new information technologies. Up to now, the security aspects have not been fully taken into account in the development of the components of the ATM, but in few years this will became a central problem to be solved. However, security aspects are now taken into account alongside other ATM key performance measures. EUROCONTROL has recently issued several guidelines highlighting security as a critical factor for future ATM developments and identifying relevant security methodologies [12],[13],[14],[15],[16].

3.1.2 The Controller Working Position

The current working positions (Controller Working Position in the ATM jargon - see Figure 2 below) are based on a large monitor, where aircraft are represented with smaller label indicating the aircrafts position and all related information (callsign, altitude, speed, etc.) and another large monitor with more than one window containing detailed information of all aircraft data (electronic progress strips) necessary to the Planning Controller (for a detailed description of the Controller Roles and of the Sector Team duties, see ANNEX I - The ATM Operational Environment).

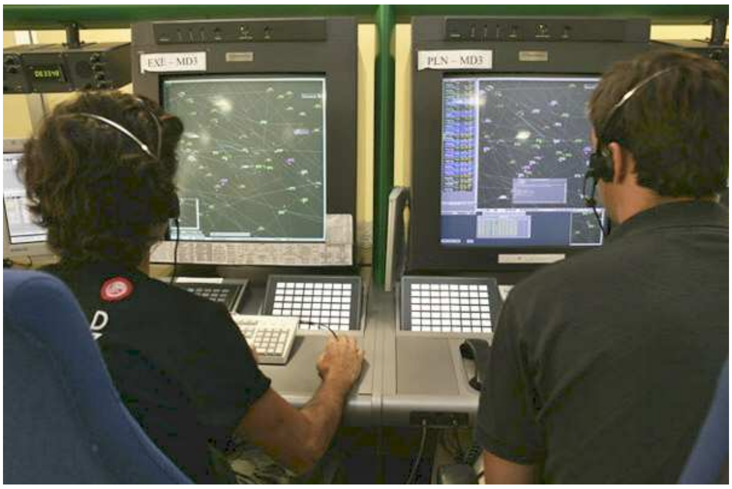


Figure 2 - The Controller Working Position.

On the Flight Progress Strip we can find displayed various details about a scheduled plane journey, including, flight level, destination, radio code and planned flight path, flight number and airline. ATCOs during en-route support activity can record on the strips possible changes to the flight details. In this way Flight Progress Strips maintain a record of what has been changed. On a larger scale the arrangement of the series of flight progress strips represents the current status of the control Sector (for a further description of Sectors and related concepts, see ANNEX I - The ATM Operational Environment), with the progress of a flight strip across the board being indicative of a plane's progress across the airspace.

The availability of the same radar data between the different control teams permits a strict coordination among members of the entire Area Control Center (ACC) staff (for a description of the ACC, refer to ANNEX I - The ATM Operational Environment). It is replicated on every working position from where controllers can monitor the own and others' activity by zooming in or out on the large displays. This promotes collaboration of various kinds allowing ATCOs to visualise how their work fits into an overall perspective, to assist and monitor in one another's work, providing some redundancy in the System.

As shown in the picture above the large monitor on the left (monitor of the Tactical Controller) displays radar data and provides a at-a-glance 'here and now' representation of what is actually happening in the skies, how busy they are, in what areas and whether two planes meant to pass 1000 feet apart are actually going to do so. Other typical tools available at CWP are: monitor displaying weather condition and forecast, monitor displaying inbound/outbound traffic planned for the sector, telephone switchboards to access direct (point-to-point) phone lines, radio receiver on the frequency assigned to the sector, a clock displaying the so call "Zulu time" (i.e. Greenwich time) used all around the world to indicate the flights time.

3.2 Changes and Evolution

In Air Traffic Management (ATM) the increase of air traffic is pushing the human performances to the limit, and the level of automation is growing dramatically to deal with the need for fast decisions and higher traffic load. In addition, there is an increase in data exchange between aircraft and ground and between ACCs due to new systems, equipments and ATM strategies. Therefore, there is a growing relevance for dependability, security and privacy aspects.

Software and devices must adapt to evolution of processes, introduction of new services, and modification of the control procedures. This adaptation shall preserve safety, security and dependability and be able to face new and unexpected security problems arising from evolution.

The *ATM 2000+ Strategic Agenda* [2] and the *Single European Sky ATM Research* [3] (SESAR) Initiative, involve a structural revision of ATM processes, a new ATM concept and a system approach for the ATM Network. This requires ATM services to go through significant structural, operational and cultural changes that will contribute towards SESAR. Moreover, from a technology viewpoint, future ATM services will employ new systems forming the emergent ATM architecture underlying and supporting the European Commission's SESAR Joint Undertaking. Introducing Safety and Security relevant methodologies in the ATM context requires us to understand the risk involved in order to mitigate the impact of possible future threats and failures.

The SESAR Operational Concept is a trajectory based system, having as major features:

1. A System Wide Information Management (SWIM) network to support all major processes.
2. Collaborative Decision Making to define a rolling Network Operations Plan, and to negotiate trajectory changes.
3. A Trajectory Managed environment rather than one that is based on Airspace Management.
4. Extensive use of automation support to reduce controller task load, but in which controllers remain in control as managers.
5. New separation modes to take advantage of advanced aircraft navigation capabilities and to allow tasks to be delegated to pilots so as to further reduce controller task load.

The essence of the system is to use precise trajectory data, combined with cockpit displays of surrounding traffic: (a) to improve predictability throughout the whole ATM system; (b) to increase capacity, productivity and safety; (c) to reduce environmental noise and emissions; and (d) to share tasks and increase the situational awareness of pilots and controllers.

In particular, the new SESAR ATM Target Concept follows a service-oriented approach based on a performance partnership amongst stakeholders. The stakeholders agree that, in order to strengthen the air transport value chain, the Airspace Users' requirements need to be better accommodated. To this end, each single flight shall be executed as close as possible to the intention of its owner.

This is the main driving principle for the ATM Target Concept, which is centred around the characteristic of the Business Trajectory (see Figure 3) representing an Airspace User's intention with respect to a given flight.

Air Traffic Management services necessary to execute this trajectory will ensure that it is carried out safely and cost efficiently within the infrastructural and environmental constraints.

Changes to the business trajectory must be kept to a minimum, altering it only for reasons of separation and/or safety or in case the Airspace Users' and ATM network goals (relating to capacity, environment and economic performance) are best met through maintenance of capacity and throughput rather than optimisation of an individual flight. Obviously, in the case of unplanned disruptions the overall ATM network goals will take precedence over individual flight trajectories.

Changes will ideally be performed through a Collaborative Decision Making mechanism but without interfering with the pilots' and controllers' tactical decision processes required for separation provision, for safety or for improvement of the air traffic flow, thanks to the new Tools that will be introduced in the CWP.

Business trajectories will be expressed in all 4 dimensions (position and time) and flown with much higher precision than today. Sharing access to accurately predicted, unique 4D trajectory information will reduce uncertainty and give all stakeholders a common reference, permitting collaboration across all organisational boundaries. Fundamental to the entire ATM Target Concept is a 'net-centric' operation based on: (1) A powerful information handling network for sharing data; (2) New air-air, ground-ground and air-ground data communications systems, and; (3) An increased reliance of airborne and ground based automated support tools.

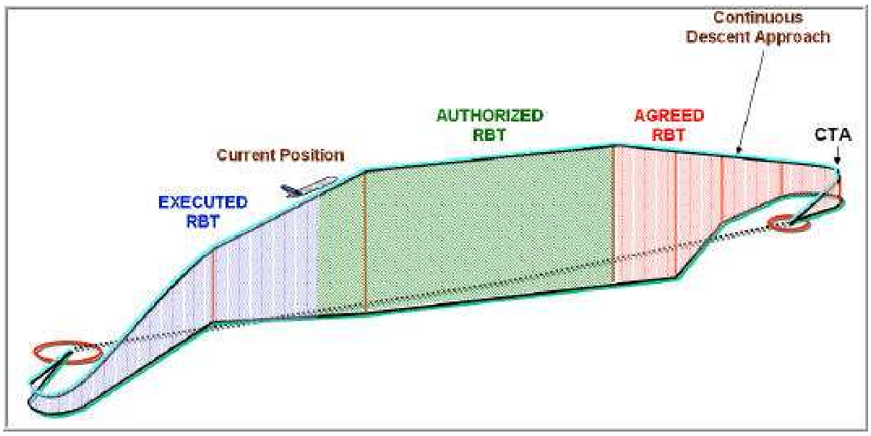


Figure 3 - The Business Trajectory

3.2.1 The SecureChange ATM Scenario

The new SESAR Operational Concept, based on Trajectory Management, that involves the deployment of new IT systems (e.g., new ground based and onboard decision making supporting tools, a new networks connecting all the ATM actors and providing them real time info, etc.) would enable an extraordinary evolution in order to deploy and support future ATM services. The deployment of new IT systems and their architecture are changing the nature of ATM services itself. From 'closed' and dedicated systems, ATM services are relying more and more on 'open', ubiquitous and ‘smart’ systems. Hence, ATM systems are becoming vulnerable to new types of hazards due to different factors. For instance, the openness of the information systems makes them vulnerable not only to malicious exploitations but also to integrity and availability hazards. Trust problems arise between ATM stakeholders, who have to rely on mediated information.

How to properly design the architecture of the future ATM System and how to manage the transition phase is an outstanding problem.

We will focus on the Control Work Position (CWP) for Air Traffic Management (ATM) and on how the CWP is fed by data and information for a safe management of air traffic.

In particular, we will highlight how Queue Managers will impact and support the definition, negotiation and implementation of the whole Lifecycle (see Figure 4) of Business Trajectories.

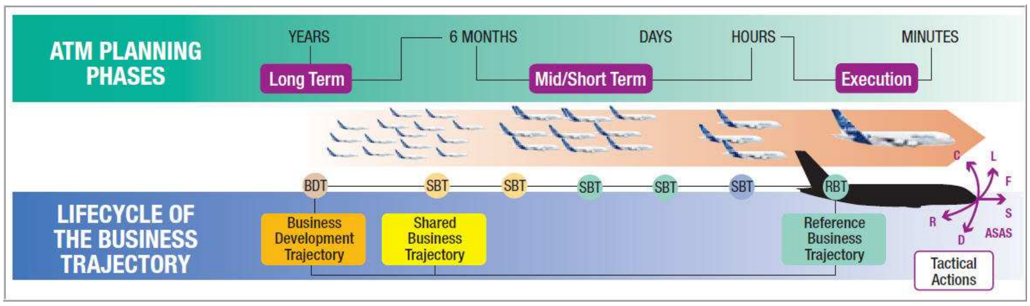


Figure 4 – Business Trajectory Lifecycle.

3.2.1.1 Architecture Description

The overall High Level Architecture is presented in next sections in terms of breakdown into logical components and modelling layers and views; it concerns the Sub-system level of the architecture and describes the system decomposition from System down to the Logical Components constituting the different Subsystems. For this purpose, the System Architecture view contains a first layer decomposition of the System into the Logical Components and the interactions between them. At this level the Logical Components might be also logically grouped into subsystems (e.g. ATC Tools including AMAN, MTCD etc.).

A System is a coherent set of integrated logical components aimed at providing a defined set of services on a defined set of interfaces towards external clients.

From a static point of view, each system is thought as decomposed into logical components. The model describes the interfaces of the Components and how some of these interfaces realize the System Interfaces (see Figure 5). The Components may be grouped into Sub-Systems. The grouping into Sub-System is purely logical and is not part of the layered refinement process. Components under analysis are described in order to highlight allocated responsibilities and its context view, indicating the main interfaces with other components or with external actors (human or system).

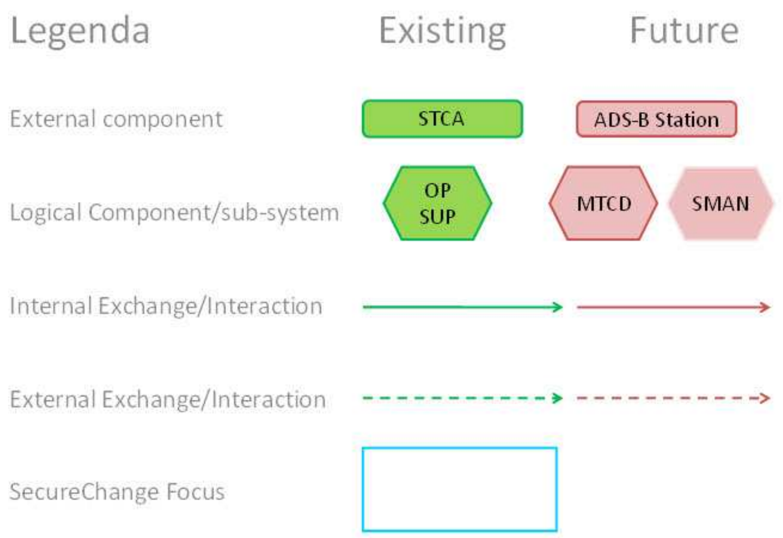


Figure 5 - Modelling Approach

3.2.1.2 The CWP of the Future

In the SESAR framework, CWPs will be integrated with information from decision support tools such as the Arrival and Departure Manager, from Safety nets such as the Short Term Conflict Alert and with other innovative tools (MONA). On the basis of this information the ATCOs take decisions to ensure a smooth, safe and efficient air traffic flow. The CWP can be directly connected with the data acquisition devices (today mainly radars) or with a unit that centralise and filter the information. CWP can be specialised for specific control purposes and several CWP are usually connected together in a network to support the co-operative work of the controllers.

The need for increased efficiency and for managing a growing number of aircraft is modifying the typical configuration of the ATM Control Centres. Aircraft will identify their exact location through satellites (GPS) and broadcast their position as well as other flight information. Remote data acquisition devices will collect this information and transmit them, using a mix of proprietary and public communication channels, to a network of CWPs located in different control centres. CWPs will co-operate to manage the air traffic according to predefined rules and following well specified procedures, with a logical architecture similar to the one shown in. These changes are already taking place and in some cases we have now a mix of the old (radar based) and new (GPS based) architectures.

The CWPs will operate in a quickly evolving environment and must exhibit a strong ability to adapt for possible changes. These may happen at different levels affecting:

- The controlled process, e.g. improved aircraft performances, increasing air traffic, new Trajectory-based environment;
- The system architecture, e.g. introduction of new controller supporting tools such as the Medium Term Conflict Detection facilities, the Arrival and Departure Managers (AMAN and DMAN), new Data-link services;

- The control procedures, e.g. introduction of new procedures using reduced separation minima between aircraft, partial delegation of responsibilities between ground and airborne.

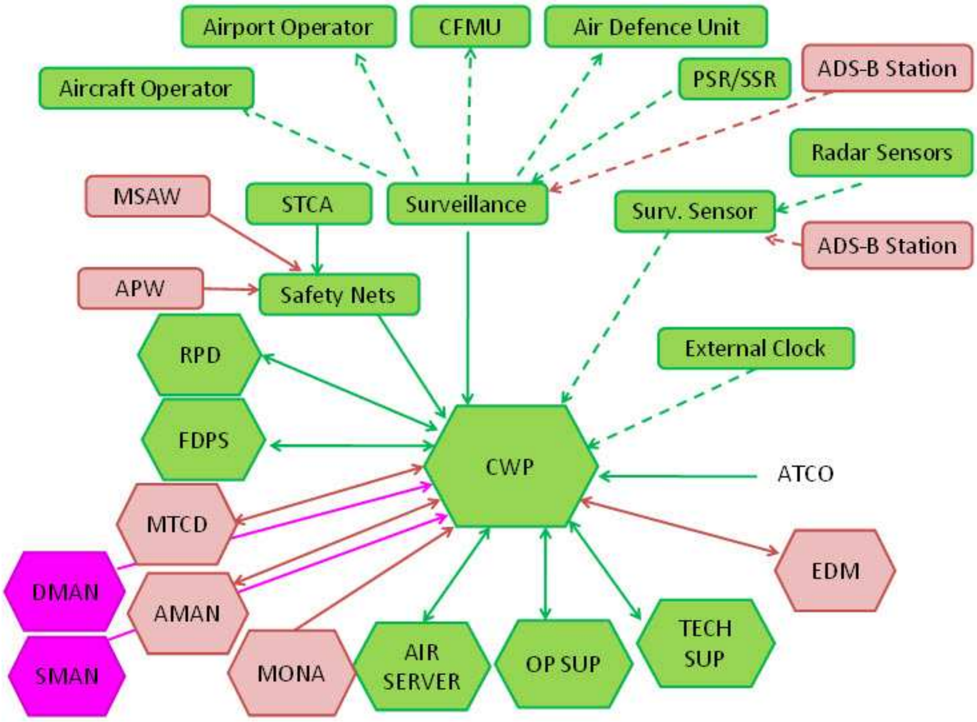


Figure 6 - CWP Future Architectural Description.

In spite of this adaptation, CWPs (Figure 6) will have to preserve the current security performances and in addition be able to face new and unexpected possible security threats that may arise from the evolution of the operational environment. For example, new operational procedures or new tools may facilitate the malicious identification of aircraft positions.

The ATM Scenario will consider several adaptations where Security performances have to be preserved, and where the CWPs shall be able to face these new and unexpected Security problems.

3.2.1.3 The Queue Management Tools

Terminal areas require specific attention not only because of the complexity of the traffic but also because of the environmental constraints. One of the major challenges in these very high sensitive areas is to take benefit of new aircraft capabilities to optimize flow management and to become more efficient while decreasing the environmental impact.

Queue Management Tools, i.e. Arrival Manager (AMAN), Departure Manager (DMAN) and Surface Manager (SMAN), are ATCO's decision support tools based on planning algorithms that will increase punctuality, predictability, and efficiency both with respect to the airport resources and to the overall network capacity. In the following a brief description of these tools, that will be introduced in the timeframe 2008-2020 (see

Table 1) for the management of queues and sequences in the approach, departure and taxing phases of flight.

The Arrival Management Process as it is Today

Arrival Management is a very complex process, involving different actors. An high level description of the Arrival Management process could be:

- Setting Goals (e.g. maximum usage of runway capacity, minimizing noise or fuel consumption).
- Creating a plan to meet the goals.
- Monitoring the conformance to the plan.
- Adjusting/updating the plan if necessary.

In the current Arrival Management Procedure, the STandard Arrival Routes (STAR), are defined from the entry fix to the metering fix on the boundary of the ACC en-route Sectors and TerMinal Areas (TMA). In some particular situations there are additional metering fixes located along the STAR (e.g. to merge the traffic from different STARS). These metering fixes are used as holding points were aircraft may hold until they can leave for the approach. Starting the approach, the flight is cleared for a standard approach or ‘vectored’ (i.e., heading, speed and altitude instructions) to the ILS (instrument landing system) localizer of the assigned runway.

3.2.1.4 AMAN

The AMAN (**A**rrival **MAN**ager), Figure 7 and Figure 8, is an aircraft arrival sequencing tool helping to manage and better organise the air traffic flow in the approach phase. The AMAN is directly linked to the airport organisation and the turnaround process because arrival sequencing/metering is important for airline operational control and airport operations (e.g. ground handlers) in order to organise the ground flow efficiently. AMAN calculates sequences on the basis of predicted times of arrival at a sequencing point, typically the initial approach fix, which is a navigation point usually 5-10 minutes before landing.

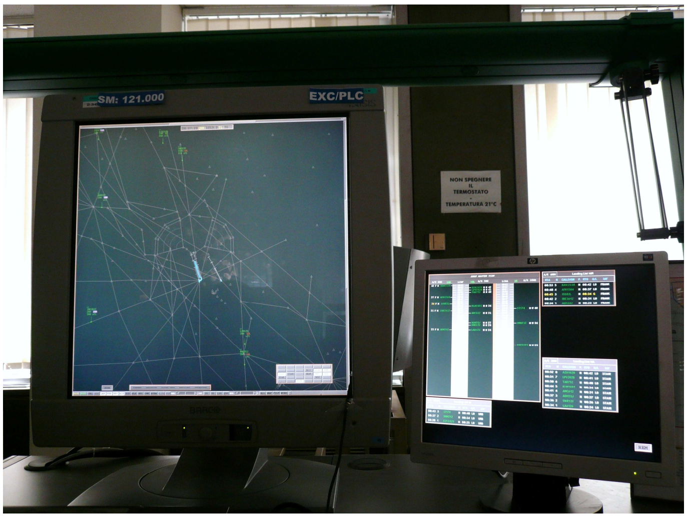


Figure 7 – An example of AMAN HMI integrated with the CWP

The aim of the AMAN is to achieve a more precisely defined flight profile and traffic flow management, in principle from off-block to arrival at the destination airport, in order to minimize the airport delay leading top better efficiency in terms of flight management, fuel consumption, time and runway capacity utilization.

Architectural Description

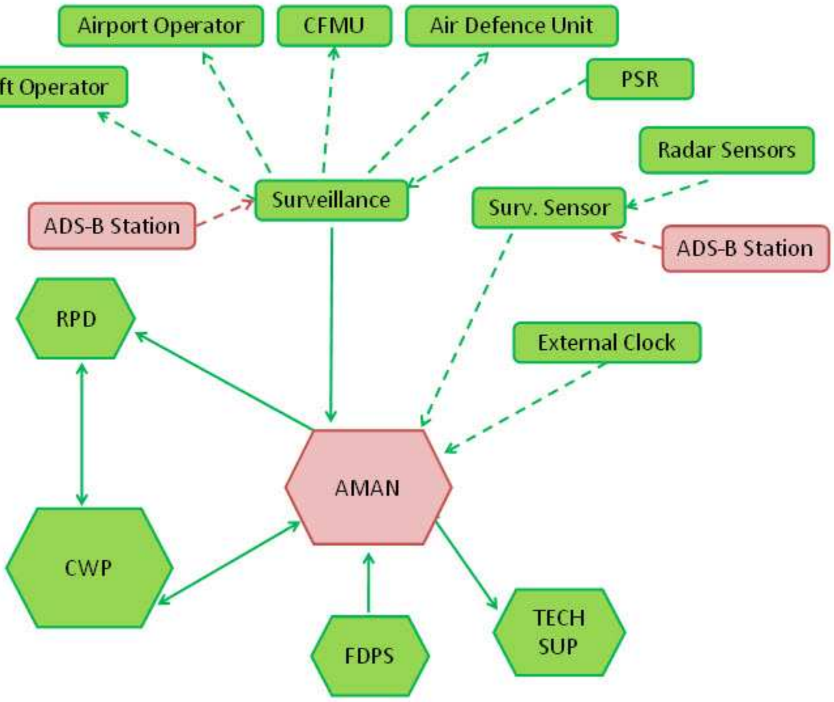


Figure 8 - AMAN Architectural Description.

AMAN Components and Functionalities

The AMAN system should ensure:

- The sequencing and metering capabilities for a runway, airport or constraint point.
- The creation of an arrival sequence using ‘ad hoc’ criteria.
- The management and modification of the proposed sequence.
- The provision of data to the HMI (in Figure 7) to allow controllers to implement the proposed sequence.
- The support of runway allocation at airports with multiple runway configurations.
- The exchange of data (in and out) with DMAN and SMAN in the same airport.
- The exchange of data (in and out) with DMAN at other airports within the AMAN operational horizon.
- The generation of advisories on: (1) Time to lose or gain, (2) Speed, (3) Top-of-descent, (4) Track extension, holding.
- The provision of AMAN what-if-probing and on-line simulations.

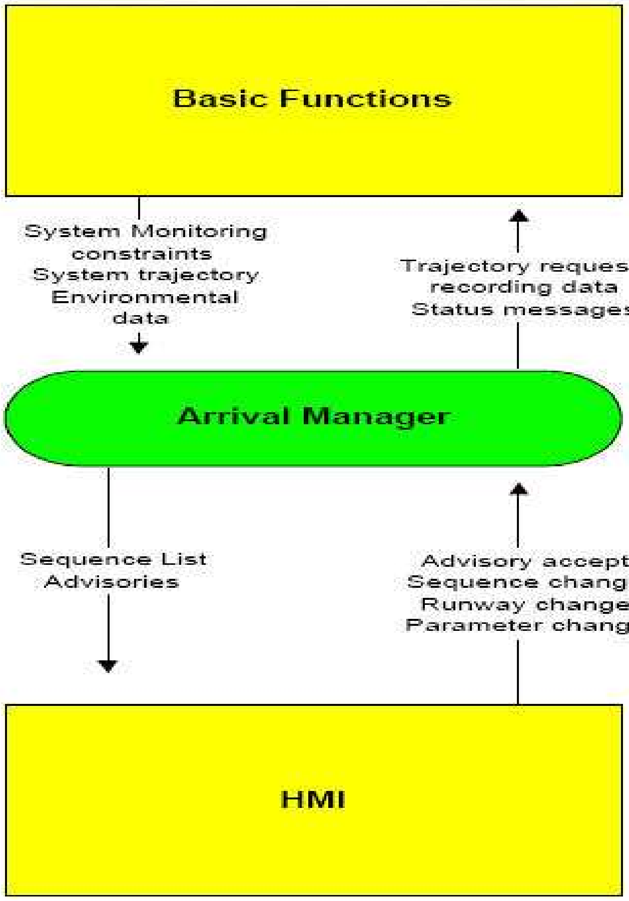


Figure 9 - AMAN Components.

All these operational requirements and services, performed by the different components in Figure 9, can be classified within the following system functionalities:

- Trajectory Prediction: Expected Time of Arrival generation.
- Scheduling: Sequence generation.
- Sequence Probe: what-if-analysis.
- Advisory Generation: heading, speed instructions.
- HMI: displaying all the above data, interaction with the AMAN system.

AMAN Process

The AMAN Process is a three step process. During the first step the Flight becomes eligible for AMAN Processing and the AMAN finds a slot within the natural sequence, created by using ‘primary criteria’. Then the Flight becomes eligible for the sequence optimization and thus the AMAN finds a slot within the optimized sequence, using other criteria. After generating an optimized sequence, the AMAN starts the Advisory generation. The Natural Sequencing within the AMAN eligibility horizon, will be based on the following criteria: (1) First-Come-First-Served; (2) Acceptance Rate and Separation Minima; (3) Meteo Conditions; (4) Individual Airport Constraints; (5) ATC

Priorities; (6) Aircraft Priorities; (7) Airport Operators Priorities (different actors: handling, security, health authority, cleaning, etc.).

The Optimized Sequence within AMAN active advisory and common path horizon is created by following the ‘second order’ criteria, such as: (1) Wake turbulence categories; (2) Minimum total delay; (3) Maximum slot delay; (4) Runway balancing; (5) Minimum cost/fuel consumption/noise.

AMAN Operational and Organizational Impact

There are a lot of expected and unexpected impacts with the introduction of the AMAN in the Controller Working Position, see Figure 10. First of all, from a technical viewpoint, the integration of the AMAN tool with the overall ACC system is not an trivial issue. Main problems are the in /out of sensible CNS data and the representation of information coherently with the current state. The AMAN needs Different User Interface and different functionalities for different roles.

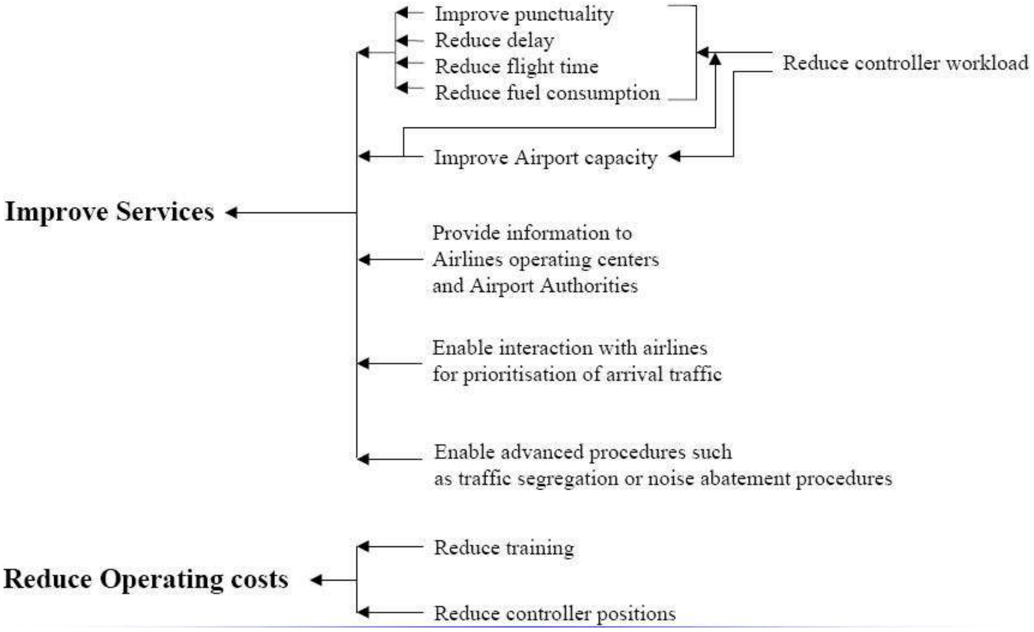

Improve punctuality
Reduce delay
Reduce flight time
Reduce fuel consumption
Reduce controller workload
Improve Airport capacity
Improve Services
Provide information to Airlines operating centers and Airport Authorities
Enable interaction with airlines for prioritisation of arrival traffic
Enable advanced procedures such as traffic segregation or noise abatement procedures
Reduce training
Reduce Operating costs
Reduce controller positions

Figure 10 - AMAN Aims and Scopes.

From and operational and organizational point of view:

- The ATCOs of the upstream Sectors will be directly and more proactively involved in the management of the inbound traffic. Possible unsafe situation in the Terminal Area (TMA) will be solved in advance by the ATCOs with the support of the AMAN.
- Creation of a new Role: the Coordinator of the Arrival Sequence, that will monitor and modify the sequences generated by the AMAN and will provide information and updates to the Sectors.
- The temporal horizon for the definition of the sequence will be larger than now.

Also the AMAN itself will evolve, as also reported in Figure 14 and in Table 1. Sequencing criteria and priorities will change,

- AMAN should take into account information provided by different airport entities,

- AMAN will be integrated with the other Queue Managers,
- New and more accurate, but less secure data inputs could be introduced, such as the Aircraft Derived Data
- AMAN should be interoperable and optimized at an European level

Current AMAN validation process has shown to increase airport acceptance rate and to reduce ATCOs workload. The coordination between ACC and APP has reduced significantly, allowing more traffic to be handled. In addition, the more structured stream of traffic into the TMA enabled the ATCOs to accept more traffic.

Apart from the capacity and delay reduction objectives for the AMAN system, two additional benefits have been identified: prioritization of traffic, with respect to different criteria and different needs, and training aspects.

Main open issues are the integration with the other Queue Managers and the management of an integrated network of airports, Figure 11. How generate the sequence for the last 30/40 minutes of a flight? How optimize various AMAN systems in Europe?

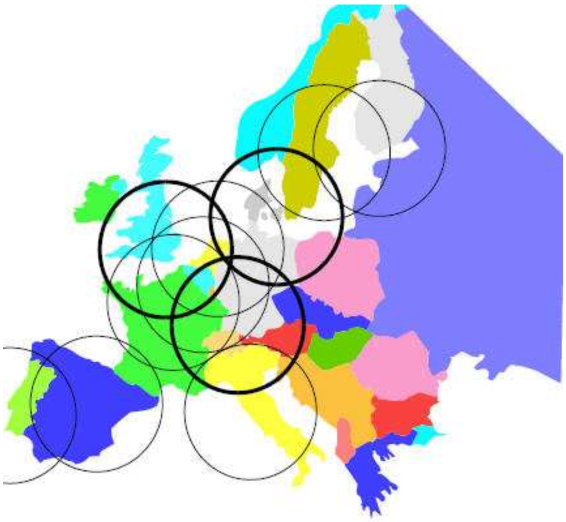


Figure 11 – Future integrated network of Queue Managers at European level.

Another important aspect is the introduction of the Aircraft Derived Data as inputs for the queue Management Tools.

3.2.1.5 DMAN

The DMAN (**D**eparture **MAN**ager), Figure 12, is a ground based planning tool. It assisted ATCOs in managing departure traffic, by providing take-off schedules as well as optimised and conflict-free climbing trajectories, in order to achieve optimal use of runway capacity and TMA airspace. As soon as the proportion of departing flights compared to the whole traffic is significant, managing departure traffic before take-off is mandatory. For each departure, as soon as the flight plan is available to the ground system, the DMAN allocated a runway and computed a scheduled takeoff time. The departure sequence is regularly updated to cope with the current traffic situation.

To build an optimised sequence, the DMAN takes into account many factors that encompass surface movement constraints, usage of runways, traffic organisation in the TMA and transfer conditions to the Extended TMA. The DMAN provides facilities for controllers to modify the computed sequences, and includes a “what-if” mode. It plans trajectories within the TMA and assists the ATCOs in performing this task by searching for optimised and conflict-free climbing trajectories in respect with operational rules.

The DMAN is adaptable to any airport configuration, i.e. runways used in single or mixed mode (Arrival or Departure, Arrival + Departure). It is able to support a safe and optimised handling of the share of runway usage between incoming and outgoing flows, in co-operation with an Arrival Manager.

Architectural Description

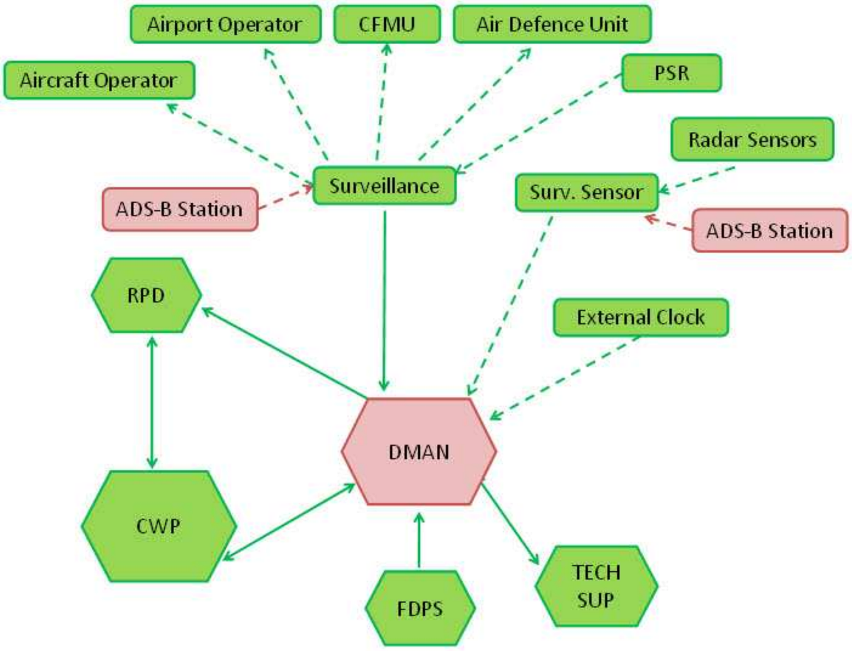


Figure 12 - DMAN Architectural Description.

3.2.1.6 SMAN

The SMAN (**S**urface **MAN**ager) is a planning and optimisation tool for airport surface traffic, closing the gap between AMAN and DMAN, with which it has to be coordinated and integrated. It is responsible for calculating the taxing time and managing the flight's progression on its trajectory during its routing between the apron and the runway. SMAN (Figure 13) also detects push-backs, line-ups, take-offs or special events such as passages made to the de-icing units.

Architectural Description

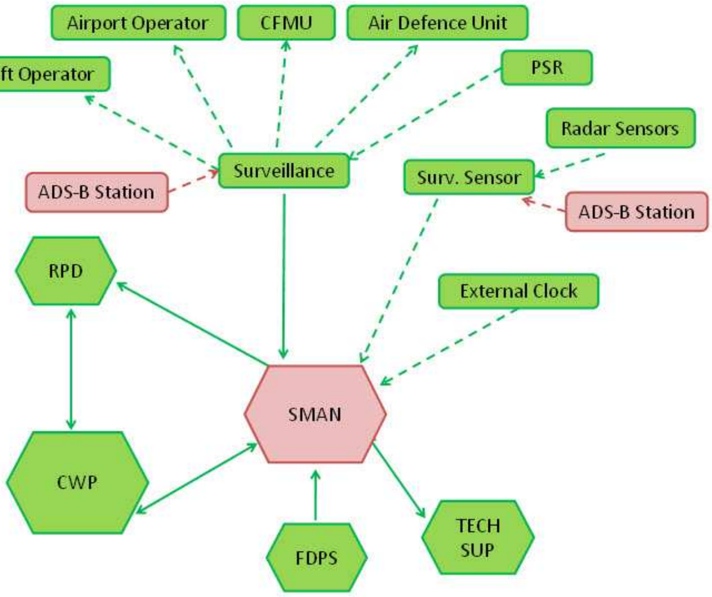


Figure 13 - SMAN Architectural Description.

3.2.1.7 Evolution Roadmap for Queue management tools

Arrival Management (AMAN) tools continue to be implemented and they integrate the En-Route part of the flight (AMAN extended in En-route by 2012).

DMAN tools are implemented in airports (2010) and are then synchronised with the pre-departure sequence (DMAN and Pre-departure) and with AMAN (if it has been implemented on the airport) to manage mixed mode runway operations, and identify and resolve complex interacting traffic flows (AMAN/DMAN integration). With the future A-SMGCS (Advanced Surface Movement Guidance and Control System) concepts, the SMAN of major airports will have to be substantially improved by the use of advanced surface radars coupled with ADS-B.

Then all the three tools will be integrated locally (2013) and for the 2020 a networked distributed environment will be implemented.

All the main actors involved in the Airports Management (handling, catering services, airlines, security and health authorities etc.) should be progressively become part of the local and then of the global network, as shown in Figure 14. The guarantee of availability, integrity and confidentiality properties in the management and sharing of this amount of sensible data is still an open problem, relevant for the SecureChange project research.

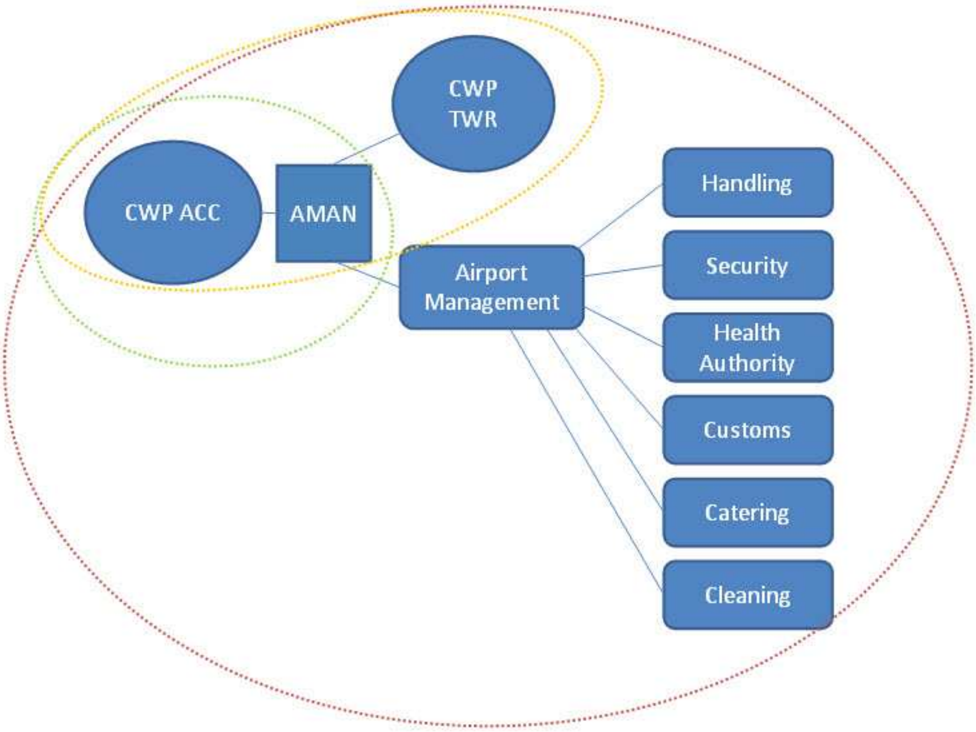


Figure 14 - Airport Management Architectural Description.

In 2016 will start the usage of Aircraft Derived Data (ADD) as inputs for Queue Management Tools. Aircraft Derived Data are avionics data transmitted from the aircraft to the ground for surveillance scopes. The supplied data may be displayed to the Air Traffic Controller and/or be used in ground processing functions and decision support tools.

All the evolution process is summarized in Table 1

Queue Management tools evolution	
2009	AMAN Introduction
2010	Provision of SMAN/DMAN
2011	
2012	AMAN/DMAN Integration
2013	Local Integration of AMAN/DMAN/SMAN
2014	
2015	

2016	Use Aircraft Derived Data for Arrival, Departure and Surface Management
2017	
2018	
2019	
2020	Networking of AMAN/DMAN/SMAN

Table 1 - Queue Management Evolution in SESAR.

3.2.1.8 Benefits for Network Performance and Business Trajectories Optimization

It is very important to notice that the usage of this Queue Management Tools will enhance not only the overall safety and efficiency of the ATM System, but also the cost-effectiveness of the overall Airport Management, in its different phases.

Airport organisation would benefit greatly from more accurate and up-to-date ETA (Estimated Time of Arrival) and even more so from better conformance to an RTA (Required Time of Arrival). ETAs need to be updated regularly and made available to the ground system, so that the airport organisation and airline operational control can optimize their resources in a cost efficient way. Efficient real time fleet management is a key to successful airline operations. The critical requirement is to maximise the usage of the aircraft in combination with minimum time on the ground (Turnaround), and a proper usage of the Queue Management Tools can greatly help in reaching the optimization of the overall Network Performance (Figure 15).

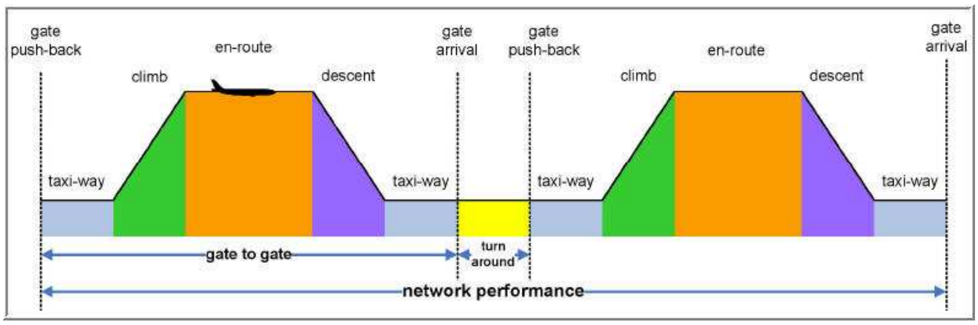


Figure 15 - Network Performance.

At airports, AMAN, DMAN and SMAN components are considered as combined entity which allows improved optimization extended to multiple airports configuration. Moreover they are integrated into advanced CDM (Collaborative Decision Making) processes, influencing the definition and negotiation phases of Business Trajectories

The lifecycle of Business Trajectories goes from the phase in which trajectories are negotiated between airlines, destination and arrival airports to the phase in which flights are operative and the Queue Managers of the departure airports and of the arrival airports respectively exchange information and support the ATCOs in taking decisions to maximize the capacity of airports runaways and to reduce flights delays.

This scenario is characterized by several security problems.

Confidentiality. Airlines may do not want to disclose all the information about their flights with the other carriers but only those information necessary to successfully complete a negotiation.

Access control. The ATCOs have different privileges according to the their role. For example, the Sequence Manager can modify the sequence of arrivals provided by the AMAN, while the Planner can only view it. The AMAN and DMAN may receive only the data about their geographically close airspace.

Integrity. The integrity of the information given as input to the DMAN and AMAN is fundamental to guarantee the correctness of the information these tools provide to the ATCOs.

Accountability. Data about flights must be available to the airlines.

3.3 List of Requirements

3.3.1 Functional Requirements

In this Section a first Draft List of Functional Requirements for the Tools under Analysis in the ATM Case Study.

3.3.1.1 Communication, Navigation and Surveillance

ATM_CNS_1 The system shall generate a track when it detects that the associated aircraft is located inside the ATSU domain of interest.

ATM_CNS_2 The system shall take into account data coming from different kind of sources: primary radars, secondary radars, mode S radars, ADS/B, …

ATM_CNS_3 The system shall not take into account a radar which is not working properly (for example, end of sector sequencing error, plot radar deconding error, antenna speed rotation instability, UTC time alignment problem etc), or too disturbed or old radar data (noise, garbling).

ATM_CNS_4 The system shall notify to the controller a precise information about surveillance sources contributing to the track update (mono/multi radars, primary/secondary radars, list of radars , Aircraft derived dataetc).

ATM_CNS_5 The system should alert the controllers when the primary radar coverage is lost or degraded.

3.3.1.2 Controller Working Position

ATM_CWP_1 The system shall be able to manage operations in ACC, APP, TWR and GND operational contexts

*ATM_CWP_*2 The system shall be adaptable to different hardware configurations (screens etc).

ATM_CWP_3 The system shall provide an interactive windows based environment

ATM_CWP_4 The system shall allow, if necessary, the visualization of different windows on more than one monitor.The windows shall be accessible with the same input devices.

ATM_CWP_5 The system shall be organised into different windows: a main radar window (General radar Picture) on which it is possible to display traffic and air space data and additional windows (Local Data Area, Main Menu Area, Strip Bay, Additional windows related to different tools).

ATM_CWP_6 The system shall be able to present and update, in a clear and efficient way, at least the following data as soon as they are available:
- Surveillance data
- Safety nets data
- Flight plan data
- Environment data
- CPDLC data
- AMAN/DMAN data
- Advanced decision assistance tools data.

ATM_CWP_7 The system shall provide for the controller the capability to personalize the presentation of information relating to a flight

ATM_CWP_8 The system shall always provide the controller with two ways to access a function or an information.

ATM_CWP_9 The system shall display the following window:
- AMAN sequence list (sequences to the IAF and to the Runway)

3.3.1.3 Queue Management Systems

AMAN

ATM_AMAN_1 The system shall be able to provide arrival sequences for more than one airport in the same area

ATM_AMAN_2 The system shall generate and provide advisories, proposed arrival runway, sequencing for configurable Multi-points (COPs, IAF, …)

ATM_AMAN_3 The system shall compute an optimized arrival sequence according to one of predefined optimisation strategies

ATM_AMAN_4 The system shall take into account local separation rules (aircraft separation standards, Wake turbulence separation standard, specific runway separation standards, weather conditions).

ATM_AMAN_5 The system shall be able to cater for all runway configurations to define aircraft separation

ATM_AMAN_6 The system shall, upon detection of a Late Appearing Flight, automatically insert the flight into the sequence (Even in the frozen part if required).

ATM_AMAN_7 The system shall generate a departure time for aircraft not airborne yet and that needs to be delayed.

ATM_AMAN_8 The system shall allow users to choose a predefined runway allocation strategy, based on the following criteria at least:

- approach fix
- departure airport
- parking stand
- airline preference
- wake turbulence separation
- noise restrictions
- taxi configuration
- aircraft performance
- runway length
- airport operation preference

ATM_AMAN_9 The system shall allow users to simulate AMAN what-if orders on a duplicate sequence.

ATM_AMAN_10 The system shall allow users to replace the operational sequence with the AMAN what-if sequence when ready

ATM_AMAN_11 The system shall allow users to manually manage the runway or point sequences. Manual sequence managment includes :

- changing planned runway for a flight
- moving a flight in the sequence
- swapping flights
- inserting a flight
- removing of a flight from sequence
- changing planned IAF
- delay a flight (without changing the sequence order)
- priority setting

ATM_AMAN_12 The system shall allow to reinsert, manually or automatically, a missed approach flight into the arrival sequence

ATM_AMAN_13 The system shall take into account priority of aircraft in arrival sequence updates

DMAN

ATM_DMAN_1 The system shall take into account the following input data to compute an ordered sequence of departing flights:

- Stand Position
- CTOT
- SID
- Flight Plan data loaded in the FDP
- Aircraft Type
- Airport Configuration
- Runway in use
- Foreseen traffic load for arrival flights (e.g. AMAN)
- Calculated taxi time to RHP

ATM_DMAN_2 The system shall provide a sequence of departing flights ordered according to their respective Target Start-up Approval Time (TSAT)

ATM_DMAN_3 The system shall allow eligible DLY controllers to manually modify the order of the proposed sequence

SMAN

ATM_SMAN_1 The system shall manage a taxi path for each aircraft, which is a path from a defined starting point to a defined ending point on the movement area.

ATM_SMAN_2 The system shall compute the most suitable taxi path knowing in advance and taking into account the following data:
- Departing/arrival stand/gate;
- Runway exit and runway entry point;
- Local standard routes;
- Local taxi restrictions with LVO/LVP;
- Type of aircraft;
- Closed Taxiways;
- Restricted areas;
- Obstacles;
- Temporary hazards;
- Intermediate waypoints (e.g. de-icing; temporary parking positions);
- Time constraints.

ATM_SMAN_3 The system, as stated by ICAO, should assure different levels of routing capability:
- manual;
- semi-automatic;
- automatic.

ATM_SMAN_4 The system shall suggest to ATCOs the most suitable predicted taxi-route taking into account the shortest taxi distance and current constraints

ATM_SMAN_5 The system shall update predicted most suitable route at any change of data taken in input

3.3.1.4 Queue Management Tool Coordination

ATM_QMT_COO_1 The system shall synchronise and optimise arrival and departure sequences.

ATM_QMT_COO_2 The system shall detect and forecast hazardous wake vortex conditions.

3.3.2 Non - functional Requirements

This Section reports High Levels Requirements for the overall future ATM System Architecture, concerning Security and Dependability issues relevant for the SecureChange Project.

From a theoretical viewpoint, note that “security”, per se, is a complex feature. The literature on Dependability, for instance, specifies security in terms of integrity, availability and confidentiality [17]. The SESAR project clearly stresses the criticality of dependability requirements [11]. Hence, we can expect that some security requirements arise from the interactions of other types of requirements. In the case of

ATM, it would be interesting to analyse the interaction between different ATM performance indicators (e.g., Safety, Capacity, Efficiency). Other important aspects, we are taking into account, concern mostly how to protect ATM assets and to 'secure the skies', according to EUROCONTROL definition of two complementary areas of aviation security:

Air Traffic Management (ATM) security is concerned with securing the ATM assets and services, to prevent threats and limit their effects on the overall aviation network. ATM Security is concerned with mechanisms and procedures, which improve the ATM capabilities to prevent or react to security threats and events, which affect flights (aircraft and passengers) or the ATM system.

Airspace security on the other hand seeks to safeguard the airspace from unauthorised use, intrusion, illegal activities or any other violation."

ATM Surveillance. The observation of an area or airspace for the purpose of determining the position and movements of IFR flights and/or VFR flights, in order to provide them air traffic control or information services in accordance to the airspace classification.

The following definitions, drawn from [11], make a distinction between aviation safety and aviation security.

Aviation safety: Aviation safety refers to the measures undertaken to guarantee all participants, users and third-parties of civil aviation from unacceptable risk of harm. Air safety reaches all domains of civil aviation: from airborne operations to air traffic control services; from airports to manufacturers and maintenance firms, not forgetting aviation staff training. It also covers every aeronautical project, product or procedure, from the conception to the certification or approbation.

Aviation Security: A combination of measures and human and material resources intended to safeguard civil aviation against acts of unlawful interference (ICAO Annex17). Aim of Aviation security is to prevent and protect against all acts of unlawful interference, including acts of terrorism and hijacking.

Acts of Unlawful Interference: Acts or attempted acts which would jeopardise the safety of civil aviation and air transport i.e.

- unlawful seizure of aircraft in flight;
- unlawful seizure of aircraft on the ground;
- hostage-taking on board aircraft or on aerodromes;
- forcible intrusion on board an aircraft, at an airport or on the premises of an aeronautical facility;
- introduction on board an aircraft or at an airport of a weapon or hazardous devices or materials intended for criminal purposes;
- communication of false information such as to jeopardise the safety of an aircraft in flight or on the ground, of passengers, crew, ground personnel or the general public, at an airport or on the premises of a civil aviation facility.

Availability (A): Concerns the ability of a system to fulfil its function and to do so within the required time or other performance criteria.

Integrity (I): of a system and the information it withholds guarantee that information is only modified by voluntary and legal (authorised?) action. For transmission, integrity includes guarantee of origin and destination of exchanged information.

Confidentiality (C): Protection of information against unauthorised disclosure.

3.3.2.1 Reliability

- *ATM_R_1* The system shall be designed to ensure reliability and portability as defined by [ISO/IEC 9126].
- *ATM_R_2* The system shall provide accurate information (tracks).

3.3.2.2 Safety

- *ATM_S_1* The system shall ensure adequate isolation of its components/functions, i.e., failure of one component/sub-function shall not to cause failure of another one
- *ATM_S_2* The system shall be designed in such a way that a failure in a non critical function does not affect the correct execution of a critical function.
- *ATM_S_3 The system architecture shall implement dissimilar channels as required to meet safety objectives for both software and hardware components*
- *ATM_S_4* The system architecture shall implement failure detection and automatic recovery with continuity of service as required to meet safety objectives
- *ATM_S_5* The design of system component has to comply with ESARR4 and national safety requirements.

3.3.2.3 Security

- *ATM_SEC_1* The system shall maintain consistency between data and presented informations.
- *ATM_SEC_2* The system shall be equipped with suitable security mechanisms to prevent from corruption of data.
- *ATM_SEC_3* The system shall be equipped with suitable security mechanisms to prevent from accidental loss of data.
- *ATM_SEC_4* The system shall be equipped with suitable security mechanisms to prevent from intentional loss of data.
- *ATM_SEC_5* The system shall guarantee the integrity and confidentiality of data against illicit attempt to obtain access to such data.

Security Requirements for CNS

- *ATM_SEC_CNS_1* The system shall provide updated tracks 24h/24, 7d/7.
- *ATM_SEC_CNS_2* The system shall prevent from the generation of false tracks (not corresponding to real aircraft).
- *ATM_SEC_CNS_3* The system shall prevent from the download/access to flight data/aircraft tracks by unauthorized people.

Security Requirements for CWP

- *ATM_SEC_CWP_1* The system shall provide a personalised "Login" process (with userID and password, or with fingerprint, or with smartcards) for each individual controller which allows him to memorize and recall automatically its personal HMI configurations
- *ATM_SEC_CWP_2* The system shall maintain consistency between data available on the same logical position (reliability)
- *ATM_SEC_CWP_3* The system shall always present on the CWP the critical data in a visible way (reliability/availability)
- *ATM_SEC_CWP_4* The System shall communicate with other components on dedicated secure channels.

Security Requirements for Queue Management Systems

- *ATM_SEC_QMT_1* The systems shall be accessible just to authorized personnel.
- *ATM_SEC_QMT_2* The systems shall provide different roles and access authorizations.
- *ATM_SEC_QMT_3* The systems shall communicate and exchange data on dedicate secured channels.
- *ATM_SEC_QMT_4* In case of Aircraft Derived Data (ADD), intrinsically insecure, the system shall validate and check them.

3.3.2.4 Flexibility

- *ATM_FLEX_1* The system should implement internal and external generic interfaces if no mandatory / regulatory interface is required.
- *ATM_FLEX_2* The system architecture shall be able to adapt to middleware implementation changes
- *ATM_FLEX_3* The system should be designed and implemented in such a way that can be modified/extended easily in the light of future European requirements.

4 HOME Case Study

4.1 General Description

HOMES Prototype's final purpose is focused on home networking, this plays an important role in recent years because an increasing number of devices are connected to the home network.

This heterogeneity means that we should pay more attention to security on these networks

4.1.1 Digital Home Networks

The **Home Networks** are those which allow all the devices in the home to communicate to each other and to connect to the outside broadband networks through the **Home Gateway**.

Home Gateway is where telecom operators are deeply involved in home networking and where the advantages of their networks and services are best represented. Interconnecting the home network and the network of the telecom operator, Home Gateway plays an essential role in digital home networking. On the one hand, Home Gateway provides network connectivity to the various network terminals at home, interconnects the public network and the subnets (e.g., PCs, telephones, electrical appliances) of the home network, and implements the remote management and control of these subnets. On the other hand, Home Gateway enables family users to enjoy the value-added services provisioned on the telecommunication network and Internet and provides access authentication and service security functions.

Home Gateway should be able to interconnect various intelligent peripherals, electrical appliances and network devices at home through advanced wireless or fixed line connection technologies and support multiple home networking standards to make it easy to build a home network and allow flexibility in extension, getting rid of the trouble of cable wiring and interface standard selection.

In Home Network it is possible to distinguish three types of sub-networks (Table 2 - Home Sub-NetworksTable 2): **Data Network**, **Multimedia Network** and **Control or Domotic Network**.

Data Network	Multimedia Network	Domotic Network
For personal computers, laptops and similar devices	For satellite receivers, multimedia servers, etc.	For sensors and actuators, control units and so on.

Table 2 - Home Sub-Networks

The Figure 16 is the networking diagram of a home network that consists of such broadband network terminals.

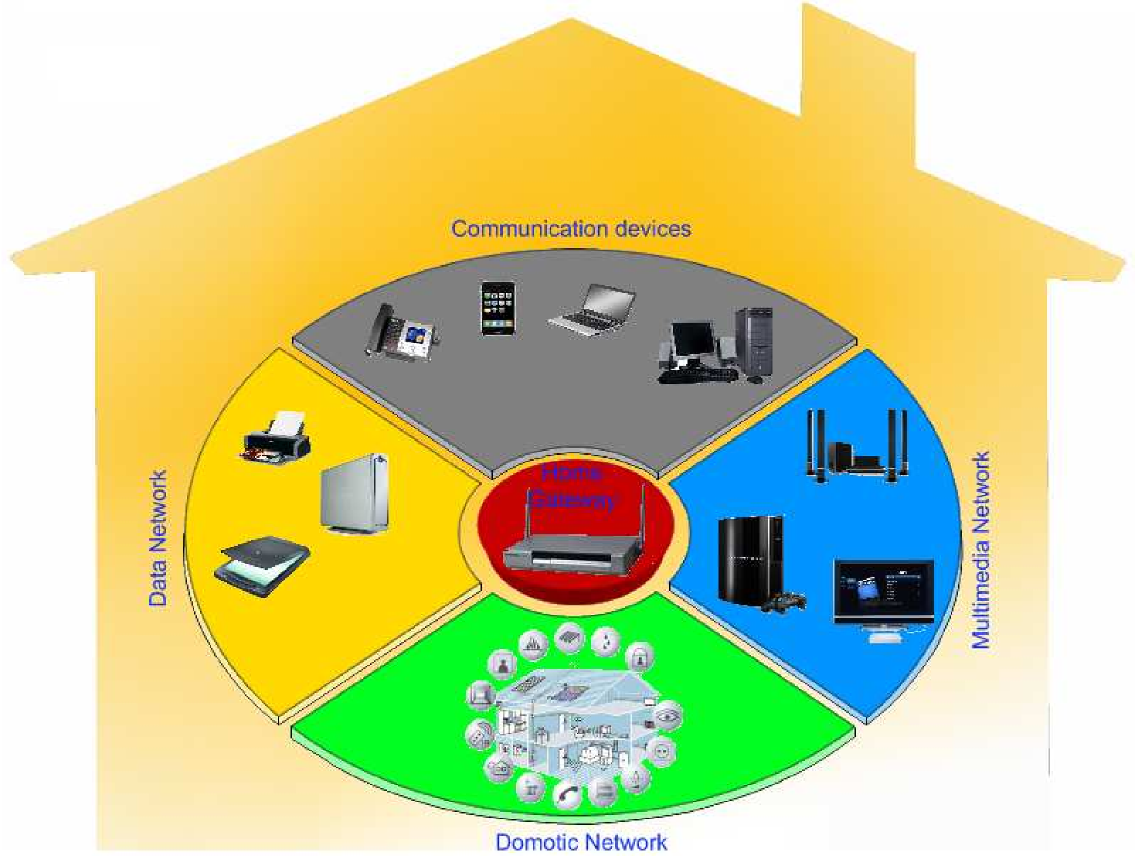


Figure 16- Home Domain Networks

4.1.1.1 Home Gateway

The Home Gateway plays an essential role in home networking, as it connects the LANs at home with the external telecommunication network. Home Gateway plays two major roles: 1) a physical interface between the external network and the home network; 2) a platform that brings various broadband network services (including services currently available and those will be available in the future) to family users.

The Home Gateway provides the network access, user interface, service extension and system architecture features and these functions:

- Home network control & information center
- A wide range of network interfaces (e.g., Ethernet, xDSL, FTTP, WLAN, WIMAX)
- Interconnecting interfaces inside the home network (e.g., Ethernet, HPNA, WLAN, PLC, BT)

- Supporting a great range of access authentication methods, ending the authentication flow (like PPPOE and DHCP)
- Routing functions (e.g., NAT, DHCP PROXY, QOS policies, VPN)
- Maintenance functions like POST, fault detection and diagnosis, upgrade, performance monitoring, and alarm report
- Security features (e.g., firewall)

We want to concentrate on Home Gateway software. Normally, home gateways use OSGi (**O**pen **S**ervices **G**ateway **i**nitiative) framework as a standard platform.

You can see the layered architecture of the Home Gateway at Figure 17.

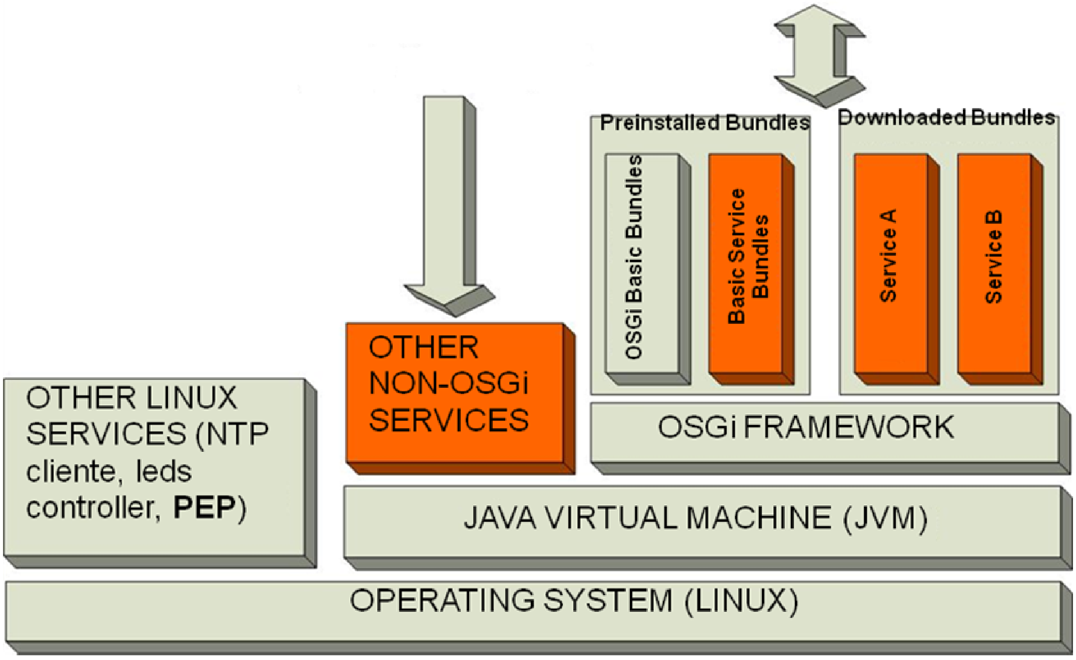


Figure 17 – Home Gateway layered architecture

4.1.2 OSGi Overview

The OSGi is a specification that defines and communicates the modularity of a Java application in a more dynamic way. Traditionally, a Java application is modularized as a JAR bundle. But working with JAR files has some limitations:

- JAR bundles are resolved through a class path environment variable, which doesn't provide a robust framework to manage JAR dependencies.
- JAR bundles can't be versioned; therefore, you can't track the history of created or modified JAR bundles.
- There's no framework to update the JAR files dynamically at run time whenever there's a code change.

To address the above issues, you can use the OSGi framework because it redefines the modular system of Java. An OSGi-based system has the following advantages over the traditional JAR modules:

- OSGi provides a robust integrated environment where bundles can be published as services and exported for other bundles to use.
- OSGi provides versioning of bundles for every new deployment. You can therefore track the history of bundle creation and changes.
- With OSGi, you can update the bundles dynamically at run time whenever there's a code change.

4.1.2.1 OSGi Lifecycle support

Most of the functionalities provided in HOMES case study shall be implemented as OSGi bundles. Just like with any other application, we can foresee common bundle lifecycle activities such as removals, updates, etc. Currently OSGi framework provides some mechanisms to help managing the set of bundles available in the environment: OSGi framework provides a component called Bundle Manager. It uses several bundles defined by the OSGi specification. These bundles support downloading bundles from the Global Service Management Platform, storing, updating and removal of bundles. The manager also handles configuration, permission, priority and dependency management of bundles.

A Bundle object is the access point to define the lifecycle of an installed bundle. Each bundle installed in the OSGi environment must have an associated Bundle object.

A bundle must have a unique identity, a long, chosen by the Framework. This identity must not change during the lifecycle of a bundle, even when the bundle is updated. Uninstalling and then reinstalling the bundle must create a new unique identity.

A bundle can be in one of six states:

- UNINSTALLED
- INSTALLED
- RESOLVED
- STARTING
- STOPPING
- ACTIVE

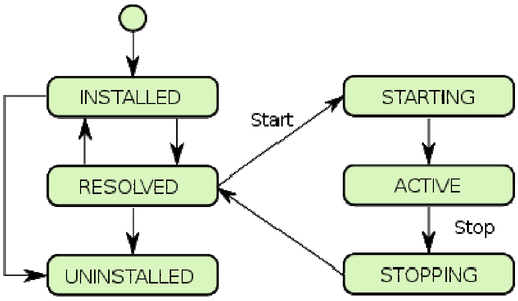


Figure 18 OSGi bundle lifecycle

Values assigned to these states have no specified ordering; they represent bit values that may be ORed together to determine if a bundle is in one of the valid states. A bundle should only execute code when its state is one of STARTING, ACTIVE, or STOPPING. An UNINSTALLED bundle cannot be set to another state; it is a zombie and can only be reached because references are kept somewhere.

The Framework is the only entity that is allowed to create Bundle objects, and these objects are only valid within the Framework that created them.

We are going to talk about the states of the bundles.

- **UNINSTALLED**: the bundle is uninstalled and may not be used. This state is only visible after a bundle is uninstalled; the bundle is in an unusable state but references to the Bundle object may still be available and used for introspection. The value of UNINSTALLED is 0x00000001.
- **INSTALLED:** the bundle is installed but not yet resolved. A bundle is in this state when it has been installed in the Framework but is not or cannot be resolved. This state is visible if the bundle's code dependencies are not resolved. The Framework may attempt to resolve an INSTALLED bundle's code dependencies and move the bundle to the RESOLVED state. The value of INSTALLED is 0x00000002.
- **RESOLVED:** The bundle is resolved and is able to be started. A bundle is in this state when the Framework has successfully resolved the bundle's code dependencies. A bundle must be put in the RESOLVED state before it can be started. The value of RESOLVED is 0x00000004.
- **STARTING**: The bundle is in the process of starting. A bundle is in this state when its start method is active. The value of STARTING is 0x00000008.
- **STOPPING**: The bundle is in the process of stopping. A bundle is in this state when its stop method is active. The value of STOPPING is 0x00000010.
- **ACTIVE**: The bundle is now running. A bundle is in this state when it has been successfully started and activated. The value of ACTIVE is 0x00000020. A bundle must pass through three states before it could reach the ACTIVE state.

4.1.2.2 OSGi Security Layer

Security is based on Java and the Java 2 security model. Provides the infrastructure to deploy and manage applications that must run in fine-grained controlled environments. The Security layer is optional in OSGi specifications to allow implementations where security is not a requirement. It is activated at JVM level.

OSGi allows access control to bundles based on permissions. These permissions are assigned according to some parameters such as software download location and downloaded software digital signature.

The OSGi Framework uses Java 2 permissions for securing bundles. Each bundle is associated with a set of permissions. During runtime, the permissions are queried when a permission is requested through the Security Manager. The management of the bundle’s permissions is handled through OSGi specific module called Conditional Permission Admin or Permission Admin (older version).

A key aspect of this security management is the use of policies. Policies contain a set of permissions that are applicable when the related conditions are met. A policy can both allow (the Java 2 model) as well as deny access when the permissions are implied. Deny permissions can significantly simplify security management. The real time management of Conditional Permission Admin enables management applications to control the permissions of other applications with immediate effect; no restart is required.

At higher layers there are defined those management services that can manage the permissions that are associated with the authenticated unit of code. These services are:

- Permission Admin service – Manages the permission based on full location strings. Conditional Permission
- Admin service – Manages the permissions based on a comprehensive conditional model, where the conditions can test for location or signer.

Further details shall be found in [19].

4.1.2.3 OSGi distributions

Here we mention the main characteristics of some distributions of OSGi Frameworks.

- Knopflerfish

The open source Knopflerfish OSGi framework is easy to install and provides a great desktop GUI. Before have been installed, the Knopflerfish framework is really easy to use and its desktop is intuitive and visual.

- Felix

Felix is a community effort to implement the OSGi R4 Service Platform, which includes the OSGi framework and standard services. Felix currently implements a large portion of the OSGi release 4 specifications, but not all the specification. Despite this fact, the OSGi framework functionality provided by Felix is very stable.

The installation is too easy like the Knopflerfish but we do not want to use this distribution because is not finished yet, and it could be some problems around the framework.

- Concierge

Concierge is optimized OSGi R3 framework implementations with a file footprint of about 80 kBytes. The design of Concierge has been developed with respect to such platforms. Concierge uses resources in a very careful way and is able to provide significantly better performance in resource-constrained environments.

4.1.3 The NAC Scenarios

HOMES case study heavily works on NAC technology. In fact, **the goal is to deploy a NAC system in a Home Network environment**. The NAC is a very broad term to describe an access control system for LANs. A NAC system has to assign a proper access to the network depending on some parameters

NAC technology allows analyzing the security status of any device connecting to the network and restricting its access in case it does not fulfil the defined security policy. Devices not passing the security check can be connected to a remediation LAN or even denied to connect.

We use TCG TNC specifications [18] as a general reference to describe the NAC components (Figure 19):

- **Access Requestor** (AR): the element requesting the access. It includes the necessary software to request the access to a network, like a 802.1X supplicant.
- **Policy Decision Point** (PDP): the element that validates the credentials, checks security data obtained from the requestor and determines the policy to apply.
- **Policy Enforcement** (PEP): the element which finally apply the policy.

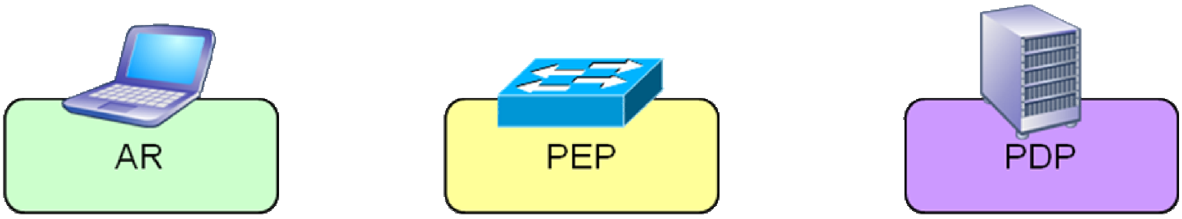


Figure 19-NAC actors

In summary, NAC technology allows to analyze the security status of any device connecting to the network and restrict its access in case it does not fulfil the defined security policy (at the PDP). Devices not passing the security check can be connected to a remediation LAN (where you may have antivirus services, device checking software or any other method to move the device to a valid state to pass the security check and therefore access the network) or even denied to connect.

For the specific case of NAC deployment over Home Networks, we can observe the proposed general architecture:

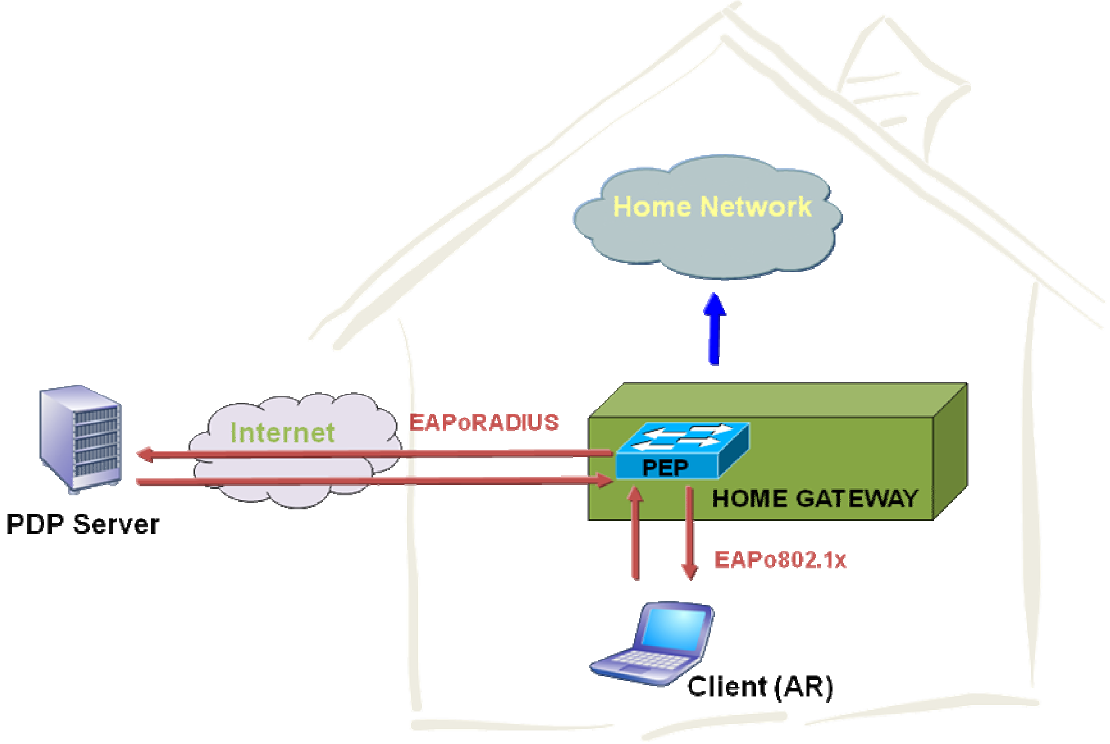


Figure 20- General NAC architecture

In Figure 20 you may notice the actual mapping of NAC components into Home Network elements.

The very general process of the NAC functionality is:

1. The AR request access, sending credentials and device's security status info obtained through a security assessment.
2. The PDP selects the proper policy to be applied.
3. The PEP enforces the policy (executes some actions).

We can distinguish two kinds of security assessments:

- Agent-based: performed by an agent installed in the device. It is a synchronous check.
- Without agent or "agentless": scanning the device when it is not possible to have an agent in the device. It is asynchronous.
- Hybrid options based in soluble agents (one-time-use agents).

This is the intended deployment in HOMES case study. Specific details on final implementation and deployment may vary depending on Secure Change work. We want to focus ideally on first two cases, but for the time being we cannot assure whether we will be able to deploy both cases in the prototype. Nevertheless, we provide details for both of them:

4.1.3.1 Agent-based assessment

We are explaining here what happens when an AR tries to connect to the Home Gateway.

1. The AR tries to connect to the Home Gateway.
2. The Home Gateway requests identification once the device link between the network and the device is active.
3. The AR sends the identity response. The Home Gateway forwards it to the Authentication Server (RADIUS).
4. The Authentication Server initiates an Authentication algorithm the AR through the Home Gateway. There are several authentication methods available. Also End User's device is assessed (from the point of view of security)
5. The credentials and the assessment data (Statement of Health) are sent to the Home Gateway
6. The Home Gateway forwards the data to the Authentication Server/PDP. It validates the credentials, and according to the validation result, the assessment data and the defined policies, it generates a response for the access request in the form of policy actions, which is sent to the Home Gateway.
7. The Home Gateway informs the client about the result of the process; send the required remediation actions if necessary, and -acting as PEP- applies the received policy actions.

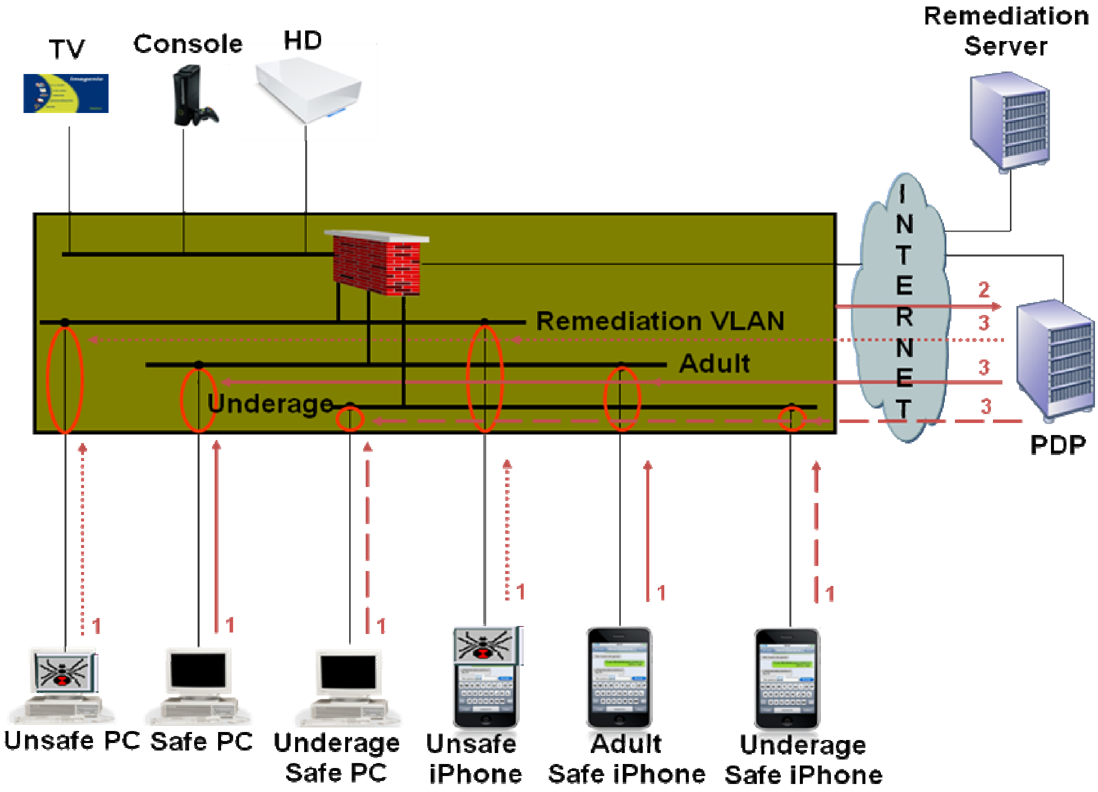


Figure 21 - Agent-based scenarios

We can see in Figure 21 how the NAC reacts from every device that tries to connect.

The protocols used in Agent-based scenarios are:

- Authentication follows **802.1X protocol schema** [20]
- The Authentication Server implements the **RADIUS** protocol [21].
- The Authentication Type used is **PEAP** [22]. Messages convey both identity and security status info.
- Status info is described via TNC's **Statement of Health (SoH)** [23].
- Security Policies do not follow any standard (ad-hoc implementation). More details on policies are provided in section 4.1.3.3

The "Statement of Health" provides some details on the security status of the device, such as:

- Antivirus status: is the AV activated?
- Operating System patch level: does the device have latest security OS patched installed?

Statement of Heatlh may be complemented by other techniques to provide extra security info, like TPM-based assessment (remote attestation).

4.1.3.2 Agentless assessment

In this case, for the security measurement, the approach is performing some kinds of scans over the device. This case is complementary to the previous one to cover the s situations with devices that do not implement an agent. The process is different but the final result shall be the same: applying enforcement actions defined in the selected policy. There is a scanning process is running continuously at the Home Gateway, looking for incoming connections. The overall process is this:

1. The scanner detect a new device connecting to the network
2. A remote assessment process is started to determine the security status of the device.
3. The assessment is sent to the PDP
4. PDP select the proper policy accordingly to the assessment output.
5. PDP sends back the obligations to be performed by the PEP
6. PEP applies the actions.

The remote assessment is a process that analyzes the traffic incoming from the device and may includes Intrusion Detection System techniques¸ virus remote scanning, etc. The output shall include related to threats like: suspicious traffic, vulnerability detection, viruses detection, etc. PDP examines this info and shall change the policy to protect the system from those threats.

As potential clientless devices we may consider:

- Hard Drive/Media Center: antivirus scan checking contents
- PC/Smart Phone: network applies an Intrusion Detection System and a vulnerabilities scanner over the device.

Figure 22 depicts agentless scenarios.

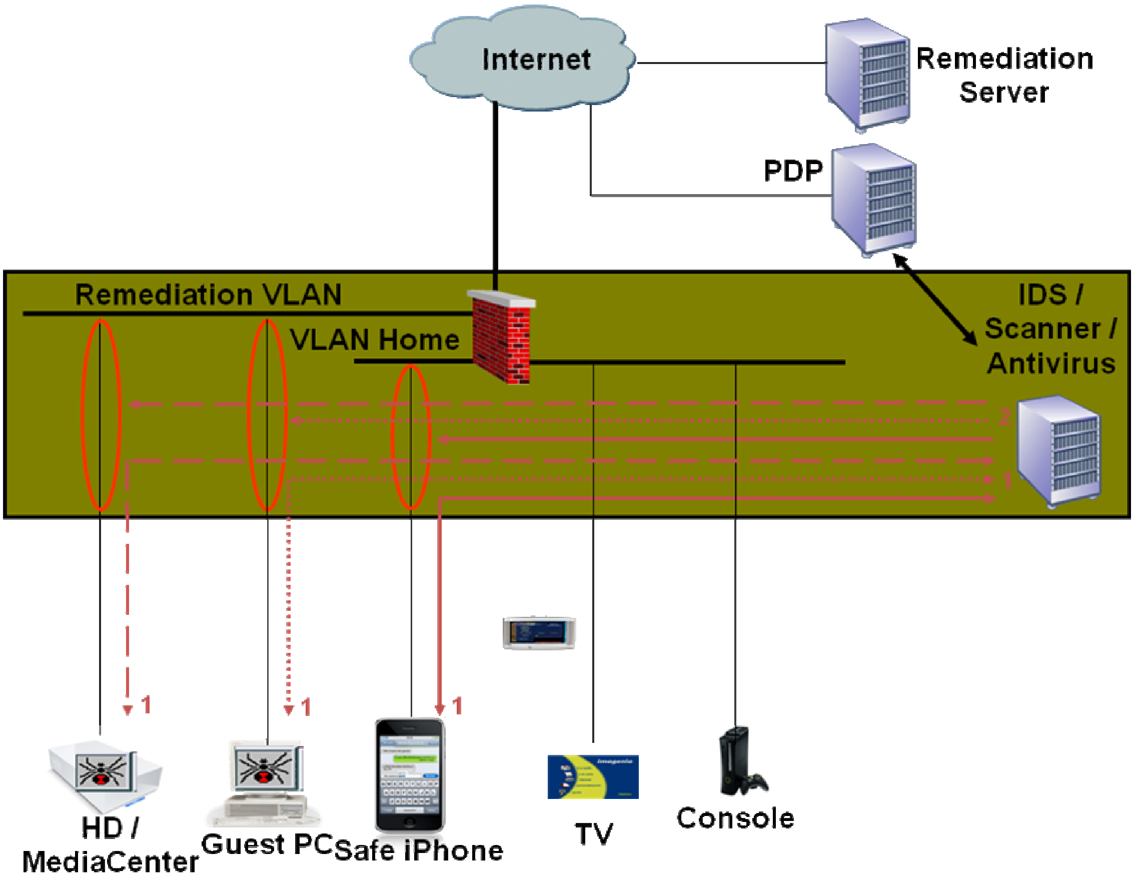


Figure 22 - Agentless scenarios

In this case the protocols vary from the previous case:

- There is no authentication in this case. It is supposed that devices are authenticated previously in some manner.
- PEP and PDP communicates each other using ssh connections
- Obligations are normally actions related to firewall operation and VLAN configuration.

4.1.3.3 Policies and Enforcement

PDP is responsible of selecting the proper policy rule taking in count the result of the assessment and user credentials. Many other parameters may be considered in more enhanced scenario (physical location, behaviour analysis, etc.) but we will not consider more for the sake of simplicity. Upon a successful validation of credentials, PDP checks the received security status of the device (the Statement of Health) and select the right policy rule.

In HOMES context we observe this terminology:

— Policy: a set of rules, and a set of obligations.

— Obligation: an operation specified in a policy or policy set that should be performed by the PEP and/or the device.

Thus, a policy rule is a set of actions to be performed under specific circumstances. These rules take into account different possible situations when considering the security assessment result and set the reply providing info about:

- Access level: access granted, access denied, access level limited to restricted network. This is done specifying obligations to be executed by the PEP (typically dealing with firewall and network reconfiguration)
- Remediation actions at the device: obligations for the device to get the access granted. Typically performing actions like activating the antivirus, applying OS security patches, etc.

Here we present some examples of rules in natural language:

“If device has the antivirus deactivated then denies the connection and tell the device to activate it”

“If device does not have latest patches deny the connection and tell the device to install them”

PDP does not communicate directly with the device but with the PEP, which is the intermediary. This component receives the reply from PDP containing the list of obligations to be applied. PEP applies the actions related to the network access by modifying firewall rules and reconfiguring VLANs. It also sends to the device the actions related to remediation in case of negative security analysis.

4.2 Changes and Evolution

In the context of the Home Network domain several kinds of changes may take place but we want to have a special focus on Home Gateway since it is a critical component within the NAC architecture. Nevertheless, other changes may be considered. We have identified some changes/evolutions where Secure Change can provide means of improvement. Work developed in some technical workpackages will help in ending up with a better system from the point of view of the security

We have identified the following sort of changes/evolutions:

4.2.1 Architectural changes

NAC architectures comprises of some functional elements mapped into actual physical components in the deployment. We are proposing a specific architecture deployment, but it may happen that a different deployment is considered interesting for some reasons in the future but not realizable because of an inflexible architecture. We may consider then a local in-home PDP instead of operator-side general PDP, for instance.

Such kind of change might be unapproachable due to severe impact from the business point of view (high costs coming from possible client's hardware upgrades, lateral effects in other home services, etc.) Secure Change may help in two ways:

- Testing new architecture: we may perform some tests simulating the new deployment to help in the assessment. Secure Change is expected to provide testing tools able to accept changes.
- Design time methodologies and/or tools to come up with a "change-aware" flexible architecture what is able to minimize the impact of architecture-level changes

4.2.2 Software lifecycle Changes

Although current OSGi approach helps in managing bundles, we observe some lack of security features to measure and mitigate the impact of new bundles added to the system, old ones being removed or even existing ones being updated. For instance**,** new domotic services may cause troubles leading to inconsistent system behaviour as it may interfere with existing services or deployed policies.

Secure Change is intended to study this sort of situations and address all the related issues coming from different levels. We may consider:

- Requirement level: actual check of requirements after any lifecycle operation.
- Functional level: new functionalities don't interfere with older ones.
- Code level: actual verification of bundle code correctness

Secure Change will **complement OSGi** features (lifecycle support, security layer) with additional security processes and tools that will assure the optimal management of bundles that will assess and mitigate the impact of addition, removal and update of bundles.

4.3 List of Requirements

4.3.1 Functional requirements

- *HOMES_FUN_1* End User Clients may implement an assessment agent able to check the security status of the device. The security status includes at least information about the antivirus and may include other details such OS patch level or TPM-based security analysis.
- *HOMES_FUN_2* Home Gateway client services shall be implemented as OSGi bundles.
- *HOMES_FUN_3* There shall be some available client services on the system. At least one bundle will be available for testing, implementing a home service. It may be a remote domotic controller, a multimedia manager or something similar.

- *HOMES_FUN_4* Bundles shall be managed (update, addition, removal) assuring system consistency

- *HOMES_FUN_5* The Home Gateway is the only entry point to the network: the AR interacts only with the Home Gateway to achieve the access to the network.

4.3.2 Non-Functional requirements

4.3.2.1 Reliability

- *HOMES_REL_1* The system may be designed to be able to accept architectural changes in the future with the minimum impact possible.

4.3.2.2 Security

- *HOMES_SEC_1* There shall be a PDP located in the operator network which takes authorization decisions based on access control policies

- *HOMES_SEC_2* The client shall implement an Access Requestor in the form of a 802.1X supplicant

- *HOMES_SEC_3* The Home Gateway shall implement an 802.1X gateway, forwarding EAPoL requests to PDP and PDP responses to the AR.

- *HOMES_SEC_4* The Home Gateway shall implement a agentless-assessment application to perform remote security analysis. It will provide the security posture for agentless clients.

- *HOMES_SEC_5* The Home Gateway shall implement a PEP module what implement the logic necessary to enforce the policy rules sent by the PDP.

- *HOMES_SEC_6* Credential based authentication shall be supported by the system

- *HOMES_SEC_7* Interaction between Home Gateway and Operator’s Authentication Server shall be done using a secure channel.

- *HOMES_SEC_8* Interaction between AR and Home Gateway shall be done using a secure channel.

- *HOMES_SEC_9* The authenticity *of* bundles shall be verified before its addition to the system.

5 POPS Case Study

5.1 General Description

5.1.1 Introduction

The advent of Java (Java Card^{TM}) as a programming language for smart cards has completely changed the ecosystem of the mobile object. Indeed, before the 2000s, applications for smart cards were developed on proprietary basis: the software platform and the microprocessor. So they were dependent from the microprocessor chosen to develop the application and from the Operating System implemented on the top of this piece of hardware. The main drawback was clearly the lack of portability/interoperability of the software and a very long time spent to develop new applications. Multiplication aspects were rather considered as new data to add to the card during its life than new applications to be downloaded. Progressively attempts were made to introduce the concept of Open Platform. The Open Platforms include the hardware and software necessary to give a single visibility of them to the application. They are built to support de facto multiplication and to receive downloaded applications.

But this new paradigm comes also with new security concerns as: who is responsible of the card w.r.t; a major issue, is it the issuer of the card or the applications providers?, are the applications protected from each others, is the operating system protected from the applications running on it (as the developer of each piece is no more the same), etc

The POPS Case Study involves a smart card, and in particular an UICC card, i.e., an Universal Integrated Circuit Card, intended to be plugged in a mobile phone (or in another mobile devices) to provide services to an end user (card holder). We will consider a multi-application and a cross-sector card because this USIM card will be used for mobile payment.

This smart portable object is composed of:

- An embedded Integrated Circuit (IC),
- An embedded Operating System (OS) which provides classical OS features (memory access, etc.) using OS functionalities,
- A Java Card System (JCS) according to [24] built on top of the OS which manages and executes applications called applets. It also provides APIs to develop applets on top of it, in accordance with Java Card specifications.
- A GlobalPlatform (GP), a set of card management services like the loading of applications. This software component provides an interface to communicate in a secure way with the external world, in accordance with [27] specifications.

- (U)SIM APIs, which provide a means to specifically interact with (U)SIM[1] applications, according to [ETSI-TS131.130] specifications.

An example of architecture of this object is presented in Figure 23, where on top of the operating system and of the Java Card virtual machine, some components are dedicated to this specific use case like an UICC APIs, a smart card web server (SCWS), a component specific to the Over the Air communication (OTA), the bearer protocol (BIP) that enable the USIM card to download data through the UMTS network. For the purpose of the project, we will focus on the GP and JCS part of this architecture. We assume that in post issuance (after the issuing of the card to the final user, when it is in the mobile phone of the user), the card is managed over the air (OTA) e.g., using SMS.

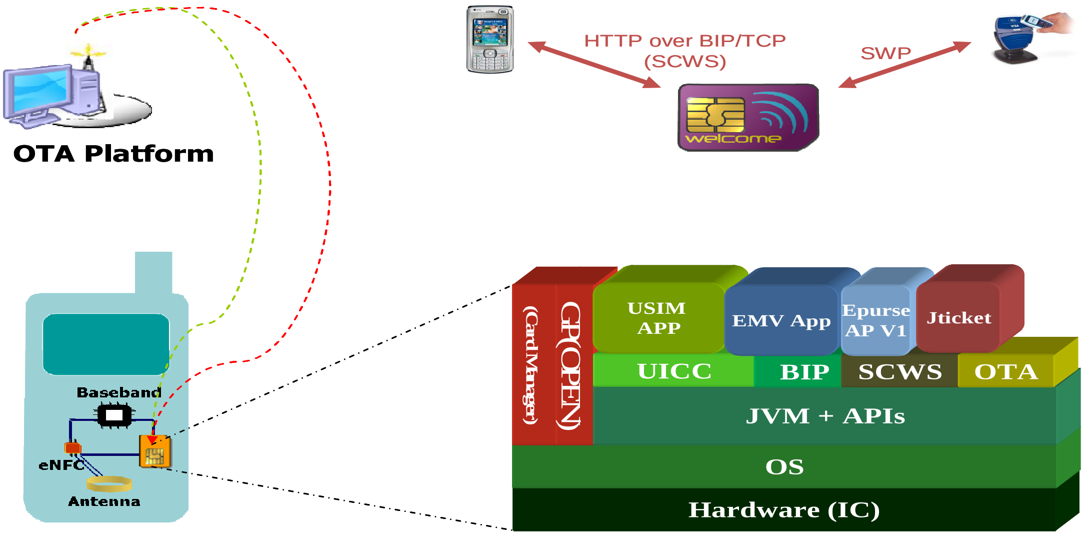


Figure 23 - Architecture of POPS

5.1.2 Java Card System

The Java Card System is defined by a runtime environment (JCRE) specified in [24] that includes:

- A virtual machine (JCVM) specified in[25]: the JCVM is responsible for interpreting the byte-codes of the applets.
- A set of APIs specified in [26] that provides commonly used services to the applets. Some support services for managing securely and efficiently the applets.

[1] A (U) SIM application is generally a Java card application that stores and manages data of the subscriber in a GSM network (or UMTS one) and its communications.

In contrast to a normal Java runtime environment, the JCRE is always in a running state on the card.

Java Card Firewall Since smart cards are mainly used in fields where security is very much an issue, a special security concept was designed for Java Card, to ensure the applications “isolation” (to protect them from theft of their sensitive assets like keys and PINs). This additional security is ensured in smart cards via a context firewall system. The basic concept is that only one applet can be selected at one time. Each applet belongs to a specific context. One or more applets may belong to the same context. In current Java Card technology, all applets sharing the same package are in the same context (package context). Only the objects belonging to the context of the selected applet can be accessed. Whenever an applet is deselected and an applet belonging to another context is selected, the context is also deselected and the other context becomes active (selected). The JCRE ensures that references to objects do not cross over context borders. The only objects that may be referenced over a context border are special objects owned by the JCRE (JCRE Entry Point Objects and global arrays). For example, these are instances of exceptions that the applet might want to return. Consequently, applets in a context run in isolation to those in another context and to the JCRE, and they only have access to objects and methods within this context.

Shareable interfaces Sometimes, it is necessary to communicate between applets (i.e. getting through the firewall). In Java Card, the interactions between two applets are done using the shareable interfaces. The server applet (which provides some services) defines a shareable interface including these services. The client applet connects to the interface to get the services. The JCRE ensures that only the services defined in the interface are available to the client. The server applet may also implement an access control on the client applets asking for its services. This ensures that only the allowed clients can use its services. The access control is mainly based on the identifier of the applets (AID – Applet identifier).

5.1.3 GlobalPlatform

The GlobalPlatform [27] consists in: ICI

- A **runtime Environment**, running on top of the Java Card runtime environment, responsible for providing a secure storage space for applications to ensure that each application's code and data can remain separate and secure from other applications on the card. Fixed memory addresses can be allocated to each application on the card, preventing each of them from accessing the memory space assigned to another application.
- **GlobalPlatform API**. While the Runtime Environment provides generic services needed by a basic smart card application, the GlobalPlatform environment, that essentially defines services of a Card Manager, provides additional services relating to card and application management and a mechanism for securing communication between a card and an off-card entity.
- The **Card Manager**. The Card Manager represents the Issuer’s interest on the card by preventing unauthorized use of the card. The Card Manager is what

enables the Card Issuer to maintain ultimate control of the card and its contents. The Card Manager supports the following four functions; Command Dispatch, Content Management, Security Management and Security Domain.

- **Card Content Management**. The applications and data on the card represent the differentiated and customisable services that can be offered to cardholders. There will be one or more applications on the card, and each of these applications will need to connect with a terminal, containing the complementary terminal component of the application before it can be used. These applications and data can be loaded or removed during Pre-issuance and Post-Issuance (e.g. a load file is a CAP file for Java Card™).
- **Security Domains**. In addition to the Issuer Security Domain, separate Security Domains can be established on the card to protect application providers or groups of applications. Security Domains enable the applications of various providers to share space on a card without compromising the security of any particular provider or application. Security Domains also allow the application owner to control its applications without the Issuer having to share its keys with the provider. The use of Security Domains is ideal when the Issuer is dealing with a trusted provider who is capable of maintaining its own applications. This prevents the Issuer from incurring the administrative overhead associated with monitoring and controlling applications, which are not part of its core business. Multiple Security Domains can coexist on the card.

  Using the extradition operations, the Security Domains may be structured in one or several hierarchies in the card. In a hierarchy, a Security Domain may use the services provided by the Security Domain that is extradited to it.

5.1.3.1 GlobalPlatform API

The Java card Runtime API provides the basic services of a smart card application. The GlobalPlatform environment provides another layer of service. These services are accessed through a separate GlobalPlatform API that handles such things as secure off-card communication, card or applet lockdown during security threats, and enables secure applet personalization such as key loading. The GlobalPlatform API interacts between the applications and the Card Manager or the Security Domains. It is important to know that the GlobalPlatform API acts as the link to the services offered by Card Manager and security domains, while the Runtime API acts as a link to the services offered by the underlying operating system.

The GlobalPlatform API provides another level of interoperability for application developers. The application providers can create a single version of the application that works through the GlobalPlatform API to leverage the unique services of the Card Manager and Security Domains. This avoids costly application development costs across multiple Issuer systems.

Secure Channel protocol: GlobalPlatform API provides a secure service for communication with the devices or server through Secure Channel Protocol. The secure communication through the GlobalPlatform API is available for use by the applications through services supported by the Card Manager and the Security Domains. The services are authentication, confidentiality, and integrity of the messages. This secure communication is critical to off-card communication with devices. The GlobalPlatform API provides the method for this secure communication by opening the channel and securing messages exchanged between the on-card application and the off-card application of the device/server. Depending on the security needs of the business model, different Secure Chanel Protocols are available. The Secure Channel Protocol SCP01 and SCP02 offer secure communication based on symmetric keys while SCP10 uses the asymmetric keys.

GlobalPlatform Personalization Support: The GlobalPlatform API provides services for personalization on the card. This is another service available to the applications. The GlobalPlatform API provides the services to accomplish this task using standardized tools that application providers can tap into during the development of their particular application.

GlobalPlatform Card Global Services: The GlobalPlatform API provides mechanisms that allow applications to offer on-card services to other applications. An application who decides to propose a Global Service can register this service. A second application can request this service without having to know who is actually providing the service. The client applet uses an authentication method defined by the server applet. This authentication method restricts access to the service and manages a trusted list of services. An example of this method is the Java Card shareable interfaces or the cardholder verification method services.

Cardholder Verification Methods: Another use of the GlobalPlatform Global Services is to manage access to the cardholder verification methods (CVM) present on the card. For example, a cardholder may have a PIN or password that is held on the card in a CVM space. GlobalPlatform API provides applications access to this cardholder verification service. This allows the applications to verify the cardholder’s authenticity through a centralized, shared space. Cardholders can then change their PIN access once and have all applications on the card use the same revised access code.

Using a GlobalPlatform Secure Channel (e.g. during personalization) allows an application to minimize its own code size by leveraging the card’s platform security mechanisms and minimizes the impact on systems using the standardized secure communication protocols.

Using the GlobalPlatform Life Cycle State management gives an application the advantage of making its Life Cycle State available to off-card management systems in a standardized manner through Issuer Security Domain commands. It also provides useful functionality for supporting application specific risk management policies and implementing associated enforcement mechanisms.

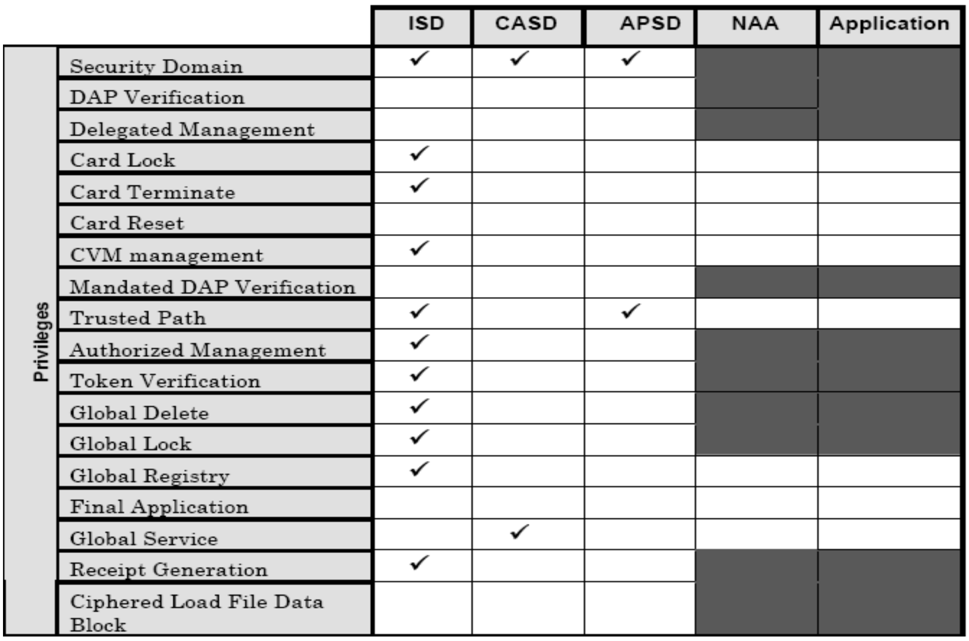
Using the GlobalPlatform CVM (a.k.a Global PIN) in applications that employ a user PIN can simplify the application itself, and can increase card usability by having a common PIN for multiple applications.

5.1.4 UICC configuration

The UICC configuration [28] is a specific configuration of GP that is dedicated to USIM cards. These cards implement ETSI-related specifications. The UICC configuration requires an issuer security domain (ISD), AP (application provider) security domains, and optionally a trusted CA (controlling authority) security domain.

Table 3 describes the privileges (see §6.6 of [27]) of the security domains in the UICC configuration. This means for example that the AP security domain have at least the *trusted path* privilege while the CA has at least the global service privilege.

The UICC configuration requires that the ISD supports the SCP80 secure channel protocol specified in [ETSI-TS-102.225]. It is also required that in any Security Domain hierarchy, there is at least one “Authorized Management” security domain. Moreover, there is not more than one “Authorized Management” security domain in each path from the root to the leaves.



		ISD	CASD	APSD	NAA	Application
Privileges	Security Domain	✓	✓	✓	▨	▨
	DAP Verification				▨	▨
	Delegated Management				▨	▨
	Card Lock	✓				
	Card Terminate	✓				
	Card Reset					
	CVM management	✓				
	Mandated DAP Verification				▨	▨
	Trusted Path	✓		✓		
	Authorized Management	✓			▨	▨
	Token Verification	✓			▨	▨
	Global Delete	✓			▨	▨
	Global Lock	✓			▨	▨
	Global Registry	✓				
	Final Application					
	Global Service		✓			
	Receipt Generation	✓			▨	▨
	Ciphered Load File Data Block				▨	▨

Table 3 Privileges in the UICC configuration

5.1.4.1 Card life cycle

The main actors are:

- The IC manufacturer who is responsible for designing and manufacturing the IC
- The smart card manufacturer who is responsible for designing the OS, JCS, GP and for manufacturing the card
- The application providers who are responsible for developing the applications
- The card issuer who is the owner of the card
- The verification authority that is responsible for verifying the applications before allowing them to be loaded.

The card has the following phases:

Pre- issuance: The IC is developed and manufactured by the IC manufacturer in parallel with the smart card software (OS+JC+GP) by the card manufacturer. Then in a production and personalization phase, the software is “embedded”, the JC and GP environment are installed and initialized on the card. It is a **pre-personalization**

phase. In a **personalization phase**, the card being compliant to GP and JC, the application can be installed and initialized. And the whole is personalized by the card manufacturer.

Post-issuance: The normal usage of the card when it is in the hand of the final user.

The specificities of the "open" card imply that software (applications) could be "loaded" on the card that is already on the field. This loading is done contactless over the air (OTA) or with a reader over the Internet (OTI).

5.1.4.2 Application development life cycle

A Java Card application is developed and compiled on a host using standard compiler. Those standard Java class files are then *converted* into a Java Card Converted Applet file (CAP File). Export files representing the imported tokens are input for the converter too. The converter resolves in fact external references and adapts bytecode accordingly; an off-card byte-code verifier may be used to statically check the CAP Files. After thorough testing on a software-simulation environment, such as the JCWDE environment, and hardware emulation (optional), the CAP File can be loaded by a loader and installed on the card. The installation could be performed within a secure environment, in a pre-issuance phase or in a post issuance for open cards. The life-cycle of an application is described in Figure 24.

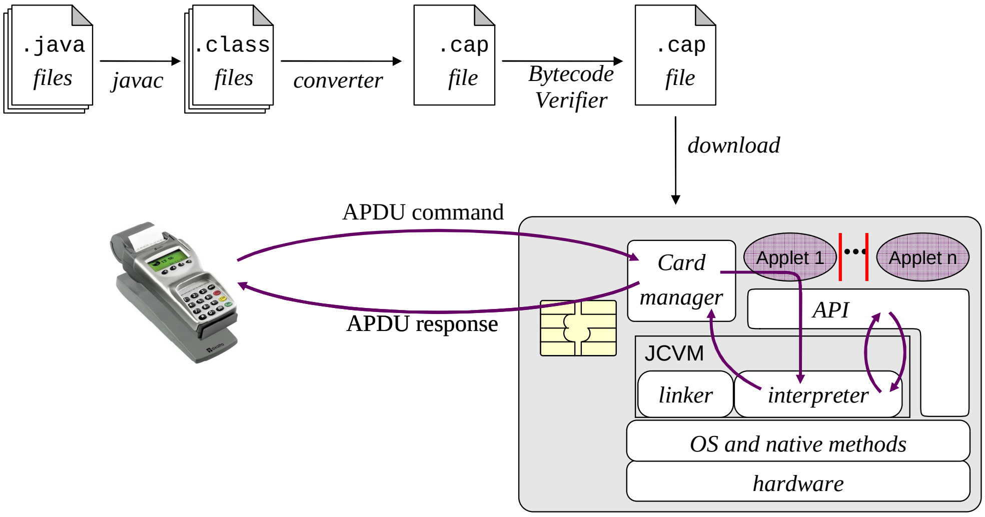


Figure 24 - Application life-cycle

5.2 The scenario

In this section, we describe a scenario of using an UICC card that includes the development of applications, their loading on the card and contents management using the GlobalPlatform commands, their execution and some changes on the card.

An overview of the scenario is given in Figure 25. A mobile network operator proposes a SIM card to its customers that will also be used for payment as a contactless credit/debit card and for mass-transport as contactless ticket card.

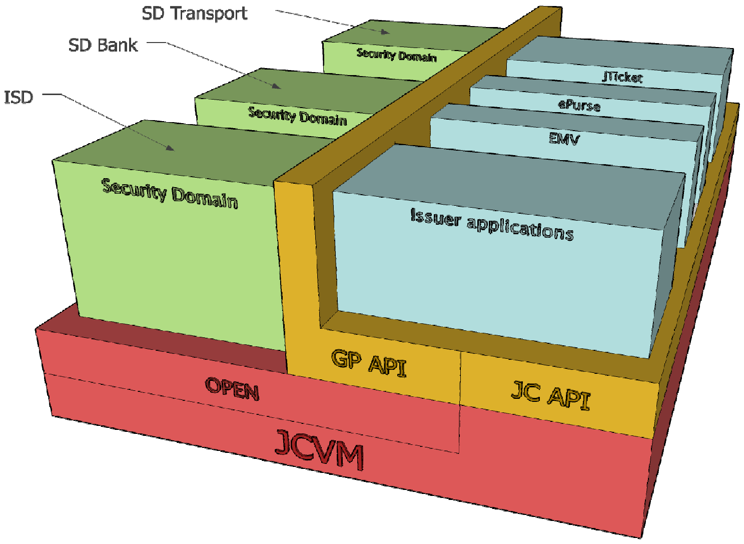


Figure 25 - Overview of the scenario

5.2.1 Actors

The scenario involves several off-card actors that will be represented by on-card entities:

- The mobile operator is the card issuer (He bought the card from the card manufacturer)
- A bank that develops and provides some banking applications
- A transport operator that develops and provides some transport applications

5.2.2 Samples of applications

The applications involved in the scenario are from different markets, banking and transport field. Those applications will be the on-card representative of the actors and of the services they provide.

- An e-ticketing application (JTicket) that will be used as a set of tickets :
  - Each access decrement the counter (N :number of tickets) by 1
  - When the counter is 0, the access is denied (JTicket needs to be refilled)
  - Jticket refill :
    - Use the epurse to buy tickets

        - Two cases :
            - Transaction accepted : N is re-initialized
            - Transaction refused : the requested amount is higher than JTicket_limit (the limit amount that epurse grants to JTicket)
- EMV: a classical credit-debit application
- Epurse (an electronic purse): for any epurse transaction, the purchasing amount must be lower than a pre-defined limit (epurse_limit)

5.2.3 Flow

We describe here the main steps to load the application on the card, illustrating the use of GP commands (blue typesetting in the figures).

First in the pre-personalisation phase, that is proprietary because GlobalPlatform is not yet available on the card:

- we create the issuer security domain, ISD
- we initialize the issuer security domain by loading its keys

Then in the personalization phase we use the GP specification:

- Figure 26: we create the bank security domain (SD_Bank) using mainly the command SELECT and the command INSTALL with parameters like the identifier AID of the SD being created and its privileges. The resulting state of the SD is SELECTABLE
- Figure 27: we personalize the keys of the bank, where the main command is the PUT_KEY with key data. The resulting state of the SD is PERSONALIZED.
- Figure 28: we load the EMV application, which has been provided by the bank to the issuer, into the SD_Bank. The loading of the applet is done using first and INSTALL command followed by several LOAD commands; each one allows loading “a piece” of the applet. When the loading is achieved, the applet is made “SELECTABLE”.

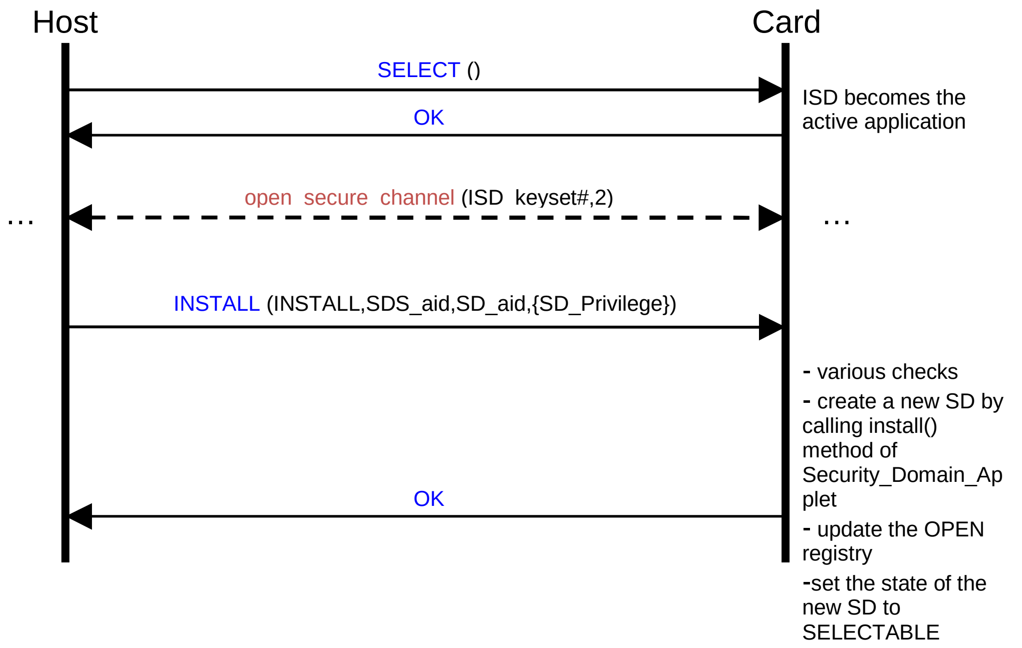

Host
Card
SELECT ()
OK
ISD becomes the active application
open secure channel (ISD keyset#,2)
…
…
INSTALL (INSTALL,SDS_aid,SD_aid,{SD_Privilege})
- various checks
- create a new SD by calling install() method of Security_Domain_Applet
OK
- update the OPEN registry
-set the state of the new SD to SELECTABLE

Figure 26 - Create a new security domain

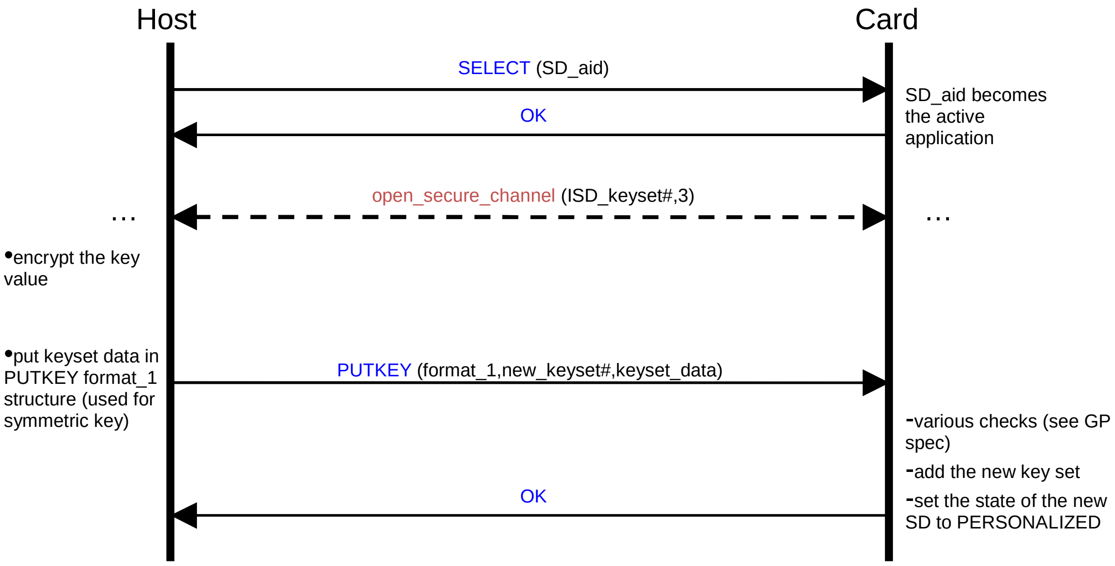

Host
Card
SELECT (SD_aid)
SD_aid becomes the active application
OK
open_secure_channel (ISD_keyset#,3)
…
…
•encrypt the key value
•put keyset data in PUTKEY format_1 structure (used for symmetric key)
PUTKEY (format_1,new_keyset#,keyset_data)
-various checks (see GP spec)
-add the new key set
OK
-set the state of the new SD to PERSONALIZED

Figure 27 - Personalize a security domain

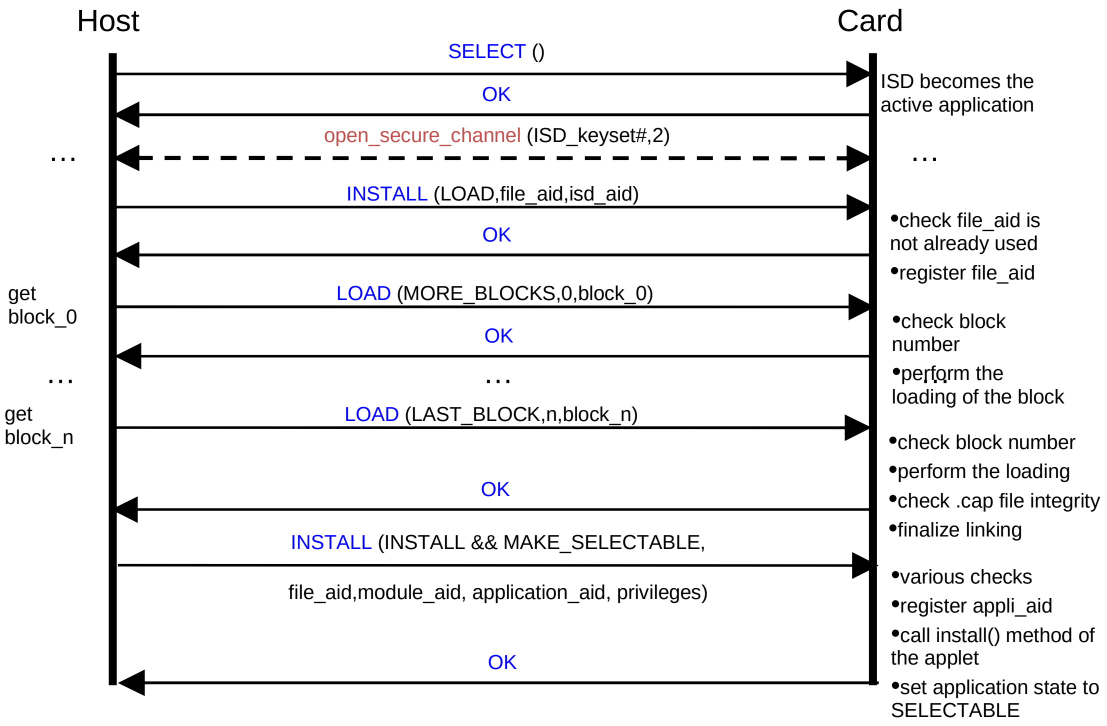

Host
Card
SELECT ()
OK
ISD becomes the active application
open_secure_channel (ISD_keyset#,2)
…
…
INSTALL (LOAD,file_aid,isd_aid)
OK
•check file_aid is not already used
•register file_aid
get block_0
LOAD (MORE_BLOCKS,0,block_0)
OK
•check block number
•perform the loading of the block
…
…
get block_n
LOAD (LAST_BLOCK,n,block_n)
OK
•check block number
•perform the loading
•check .cap file integrity
•finalize linking
INSTALL (INSTALL && MAKE_SELECTABLE,
file_aid,module_aid, application_aid, privileges)
OK
•various checks
•register appli_aid
•call install() method of the applet
•set application state to SELECTABLE

Figure 28 - Load a new application (application_aid)

Secure communication :

The command `open_secure_channel(keyset#,security_level)` is an abstraction of the process to establish a secure way to communicate: it opens a secure channel between the card and an external entity using a specific keyset and providing a specific security level (integrity, integrity and confidentiality, or none). Figure 29 shows how it is done in SCP02 using the GP commands.

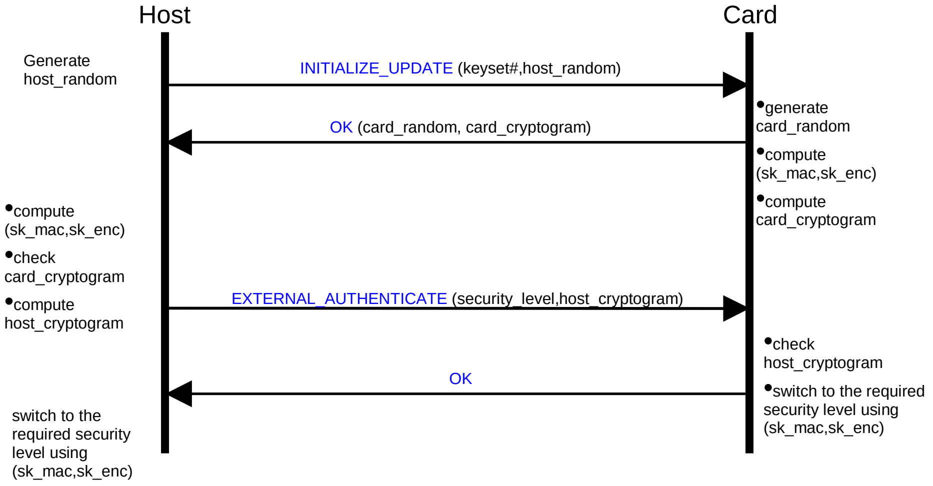

Host
Card
Generate host_random
INITIALIZE_UPDATE (keyset#,host_random)
•generate card_random
OK (card_random, card_cryptogram)
•compute (sk_mac,sk_enc)
•compute card_cryptogram
•compute (sk_mac,sk_enc)
•check card_cryptogram
•compute host_cryptogram
EXTERNAL_AUTHENTICATE (security_level,host_cryptogram)
•check host_cryptogram
•switch to the required security level using (sk_mac,sk_enc)
OK
switch to the required security level using (sk_mac,sk_enc)

Figure 29 - Open a secure channel following SCP02

5.2.4 Implementation details

The EMV, Jticket and e-purse applications are implemented as Java Card applets. Each of them has an unique identifier (AID).

The security domains are implemented by a proprietary applet.

- each security domain is an instance of this applet (created with “SD_Privilege”)
- each security domain has a unique identifier (AID)

5.3 Changes and Evolution

We focus on the changes during the usage phase of the UICC card, because in the other phases, the changes are managed in a secure environment (in card manufacturer or card issuer highly-protected sites). In order to illustrate the security properties, two categories are considered: administrative and usage changes.

5.3.1 Administrative changes

- The first administrative change is the creation of the security domain associated with the transport operator (SD_transport). This step is necessary in order to download the ticketing application (JTicket). Then, JTicket is loaded into SD_transport.
- The bank offers an epurse service: the epurse application is loaded into SD_bank

5.3.2 Application data changes

The changes on the application data includes:

- Change of the limit representing the total number of tickets (Jticket_limit). This occurs when the user consumes all its tickets.
- Change of the maximum amount of “money” of the epurse (epurse_limit). This occurs when the “users” decide that the max amount set by the bank is too low and is “consumable” rapidly. So the user needs to fulfil his epurse frequently. So he can decide with its bank to increase this value.

5.3.3 Security policy changes

EMV and epurse are two applications on the card. The communication between them obeys to the security policy defined by their application provider that is the bank. We assume in the scenario that initially, both applications do not communicate and runs standalone. At a given moment, the bank decides that it will be easier and more flexible if the user can change by him. For that, we need to change the code of epurse as it was not developed with that “functionality”. Also we suppose that the EMV application manage a list of its client ID.

Changing epurse could be done using two methods:

- Patching all the code and overwriting it
- Updating the application by adding a new method “refill_by_emv()”.

5.4 List of Security Requirements

In the following, only security requirements are provided. Functional requirements are generally collapsed into a single one “correctness” with respect to the specifications. The fulfilment of these requirements is the basis for all the security requirements.

5.4.1 GP requirements

- *POPS_GP_1* All entities on the card must be uniquely identified.
- *POPS_GP_2* The loading of applications in post issuance must respect the policy of the card.
- *POPS_GP_3* The application installation must be safe, e.g. to ensure that all external references of the application are valid.
- *POPS_GP_4* The application deletion must be secure, i.e., it shall not leak previous information to the new application to be allocated in the same memory space.
- *POPS_GP_5* Card content management must be done by authorized actors and ensure:
  - The consistency of card and application life-cycle
  - The enforcement of the card issuer policy

5.4.2 Application development requirements

- *POPS_APP_1* All the applications loaded on the card must follow the guidelines to develop secure Java Card applications
- *POPS_APP_2* All the applications must be bytecode verified (off-card or on-card). If the verification is done off-card then:
  - the adequacy between the export files used in the verification and those used for installing the verified file must be ensured
  - no modification of the file is performed between the verification and the signing operation (by the verification authority)

5.4.3 Changes-related requirements

- *POPS_CHG_1* A new security domain must not be able to access the applications from other security domain.
- *POPS_CHG_2* If an actor is added on the card, the consistency of the existing security policy must be preserved, e.g. the privileges policy.

- *POPS_CHG_3* Adding an application or updating an application must not generate
  - Illegal interaction with existing applications
  - illegal modification of GP system data (e.g. card or application's life-cycles, privileges, AID)
  - illegal modification or leak of GP user data (e.g. ISD/SD keys, global PIN)
- *POPS_CHG_4* Updating an application code or data must preserve its consistency/correctness (with respect to its specification).

5.4.4 UICC specific requirements

- *POPS_UICC_1* The security domain keys of the application providers are generated and stored in a secure way.
- *POPS_UICC_2* The transmission of keys to the application provider must be trusted and performed in a secure way.
- *POPS_UICC_3* Isolation between the security domain hierarchies: a security domain or application in a hierarchy is not allowed to access to the data/services of the other hierarchies.
- *POPS_UICC_4* Isolation between branches (a branch is a path from the root to a leaf) in a hierarchy: the data and services of a security domain are only accessible to the security domains and applications that are under it in the hierarchy.

6 Glossary

6.1 ATM Case Study

Acronyms	**Definition**
ACARS	Aircraft Communications Addressing and Reporting System
ACC	Area Control Center
ADD	ADD Aircraft Derived Data
ADS-B	Automatic Dependent Surveillance Broadcast
ADS-C	Automatic Dependent Surveillance Contract
AMAN	Arrival MANager
ANS	Air Navigation Services
ANSP	Air Navigation Services Provider
ATC	Air Traffic Control
ATCO	Air Traffic COntroller
ATM	Air Traffic Management
BT	Business Trajectory
CNS	Communication Navigation Surveillance
CORA	COnflict Resolution Assistant
CWP	Controller Working Position
DMAN	Departure MANager
ETA	Estimated Time of Arrival
EUROCONTROL	The European Organization for the Safety of Air Navigation
FAA	Federal Aviation Administration
FDP	Flight Data Processing
FMS	Flight Management System
HMI	Human Machine Interface
ICAO	International Civil Aviation Organization
MSAW	Minimum Safe Altitude Warnings
MSP	Multi Sector Planner
MTCD	Medium -Term Conflict Detection
RTA	Required Time of Arrival
SESAR	Single European Sky ATM Research

SMAN	Surface MANager
SSR	Secondary Surveillance Radar
STCA	Short-Term Conflict Alert
SWIM	System Wide Information Management
TMA	TerMinal Area

6.2 HOMES Case Study

Acronyms	**Definition**
DHCP	Dynamic Host Client Protocol
FTTP	Fiber To The Premises
NAC	Network Access Control
NAT	Network Address Translation
OSGi	Open Service Gateway Initiative.
PDP	Policy Decision Point
PEP	Policy Enforcement Point
PLC	Power Line Communication
PPPOE	Point-to-Point Protocol over Ethernet
QOS	Quality of Service
VPN	Virtual Private Network
WIMAX	Worldwide Interoperability for Microwave Access

6.3 POPS Case Study

Acronyms	**Definition**
AID	Application identifier
APDU	Application Protocol Data Unit
SCP	Secure Channel Protocol
EMV	Europay MasterCard Visa
ISD	Issuer Security Domain
SIM	Subscriber Identity Module
USIM	Universal Subscriber Identity Module

7 ANNEX I - The ATM Operational Environment

In Annex I we wish to describe more in details the ATM Operational Environment and its main features. Figure 30 shows the different phases forming a flight profile.

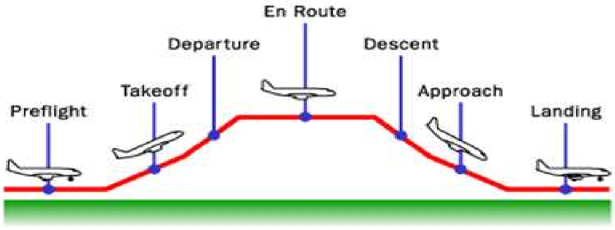


Figure 30 - Profile of a typical commercial flight.

It consists of the following phases:

- Pre-flight / Taxi – this phase is performed on ground and includes flight check, push-back from the gate and taxi to the runway
- Takeoff – the pilot powers up the aircraft and speeds down the runway
- Departure/Climb – the aircraft lifts off the ground and climbs to a cruising altitude
- En-route – the aircraft travels according to its flight plan towards the destination
- Descent – the pilot descends towards the destination airport
- Approach – the pilot aligns the aircraft with the landing runway
- Landing / Taxi – the aircraft lands on the runway, taxis to the gate and parks at the terminal.

During these phases, the Air Traffic Management (ATM) Service[2] [2] is provided by Air Traffic Controllers operating at Airports for the arrival and departure flight phases and in Air Traffic Control Centres (ACC) for the en-route flight phase. With respect to the flight profile represented in the above figure:

- Pre-flight and Take-off phases are managed by Tower ATCOs operating at departure airport,
- Climb, En-route and Approach phases are managed by En-route ATCOs operating in the various ACCs the aircraft passes through,

[2] *Air Navigation Services (ANS), according ICAO definition, include Air Traffic Services (ATS), Aeronautical telecommunications Service (COM), Meteorological Services for air navigation (MET), Search and Rescue (SAR) and Aeronautical Information Services (AIS). These services are provided to air traffic during all phases of operations (approach, aerodrome control and en route). With the implementation of CNS/ATM systems, ATS and COM will be replaced by **ATM** and **CNS** which are broader in scope.*

- Approach and Landing phases are managed by Tower ATCOs operating at destination airport.

All the operations of the Air Traffic Controllers are defined in very elaborate Procedures (called ‘Internal Permanent Procedures’ or I.P.I.) driven by imperative safety requirements.

7.1.1.1 The En-route Sector Team

The airspace managed by each ACC is organised into adjacent volumes, so-called Sectors. Each Sector is operated by a team of two ATCOs, consisting of a Planning Controller and a Tactical (former Executive) Controller. The Planning and the Tactical Controllers work together as a Team and share the responsibility for the safe operation of the sector they control. The Tactical Controller is in charge of all air/ground communication. He monitors the aircraft in the sector and provides pilots with instructions/clearances on aspects such as speed, altitude and routing to maintain a safe separation with other aircraft flying in the sector. He also gives pilots weather and air traffic information. When the aircraft approaches the sector boundary, he passes it off to the Tactical Controller of the adjacent sector (not always belonging to the same ACC). The Planning Controller assists the Tactical Controller, coordinating entry and exit flight level and entry and exit flight point with adjacent sectors in order to ensure a smooth air traffic flow. He also monitors the traffic within the sectors and in most of cases updates the ATC system with the instructions given by the Tactical Controller. The Team can also rely on an en-route information binder reporting the up-to-date information about the sector they manage. Typical information that can be found in the binder are: general description of ACC Organization and Air Space Boundaries, Equipment description and user manual, detailed description of possible ACC’ layouts (Sectors) and Sectors description/information, adjacent and subordinate ATS units coordination procedures, restricted zones description, flow management procedures, contingency procedures and so forth.

7.1.1.2 The Supervisor role

Groups of neighbouring Sectors are coordinated by a Supervisor, who is in charge of managing the sectors configuration under his responsibility according overall traffic forecast. Supervisor is also responsible for the formation of the Sector Teams. Good cooperation between Planning and Tactical Controllers is a mandatory requirement kept in serious consideration when the Supervisors set up Teams. Composition of Teams and the assigned sector is planned in advance by the ACC Operation Office. Supervisors could change Teams and Sectors layout in order to answer to the traffic modifications and a record is kept in an official log. Another aspect taken in consideration by Supervisor is the level of experience gained by a Controller in a specific sector: some packed sector like “arrivals” or “departures” related to some great Airports, are accessible only to Controllers with deep knowledge of the critical aspects regarding these specific sectors.

The number of Sectors operating at the same moment is directly dependent on the number of flights under control and in a singular ACC there can be more than one Supervisor managing the air traffic of different geographical zones (e.g. islands). Each Sector, and consequently each ACC, has a predefined Capacity, which is the maximum number of flights that can be safely managed by the ATCOs operating in the

Sector/ACC. The Supervisor uses to walk around the ACC just to see what staff are doing and to understand what is going on. This usual practice can be useful and help in promoting shared knowledge by facilitating self-reporting of hazards. The Supervisor facilitate self-reporting by explaining that reports are the basis for the constructive improvement of safety and not the basis for disciplinary actions. In case of traffic increase, the Supervisor may decide to modify the sector configuration, thus splitting one or more sectors into two/three sectors in order to increase the number of sectors and consequently the ACC capacity. The new sector configuration, whenever necessary, is not defined by the Supervisor, but follows predefined procedures of sector configuration, which imply the availability of the necessary resources, in terms of both Air traffic Controllers to manage the new sectors and Control Working Positions to allow them to safely manage the air traffic. The Supervisors have thoroughgoing knowledge and experience of positions below their present standing and therefore, how these different 'sub-roles' work to compose the overall activity. As a result a Supervisor has the ability to monitor, supervise, assess and explain to his ATCOs, as well as being able to fluidly assist and take over the roles of controllers as required. The Supervisor has also the responsibility to promptly avoid delays in crucial information transmissions due to partial failures of automatic information systems. He accomplishes this task directly communicating with ATCOs. A typical example is a sudden meteorological worsening that might lead to closure of a destination airport.

7.1.1.3 The Area Control Center Environment

From a technical point of view each ACC is a very complex System, consisting of a large number of automated equipments dedicated to the presentation of the air traffic, obtained through Surveillance sensor (e.g. Radar), to the traffic forecast, obtained through the connection at specific data bases, and to the presentation of all the other information helpful for the management of the flights (e.g. weather information). Control Team interact with the System through keyboards, mouse and touch-screen.

Moreover each ACC is linked to one or more Communication Centre where, through dedicated and secured radio frequencies, each ACC Sector is in touch with all aircrafts flying in its airspace and with dedicated point to point (thus secure ‘by default’) telephone line or radio link with adjacent ATS Units. All radio frequency and telephone line are recorded in the eventuality of an investigation following an accident or a nearly accident.

The Control room is hosted in large open space environments designed guaranteeing adequate lights positioning, low ambient noise and absence of architectonic obstacles. The physical arrangement of the ACC supports collaboration within the ACC by favouring knowledge sharing activities such as communication between ATCOs of adjacent sectors and monitoring activity of Supervisors.

The CWP are grouped in dedicated operative zones (e.g. islands). Each zone refers to interrelated portions of air space controlled by the ACC (see Figure 31).

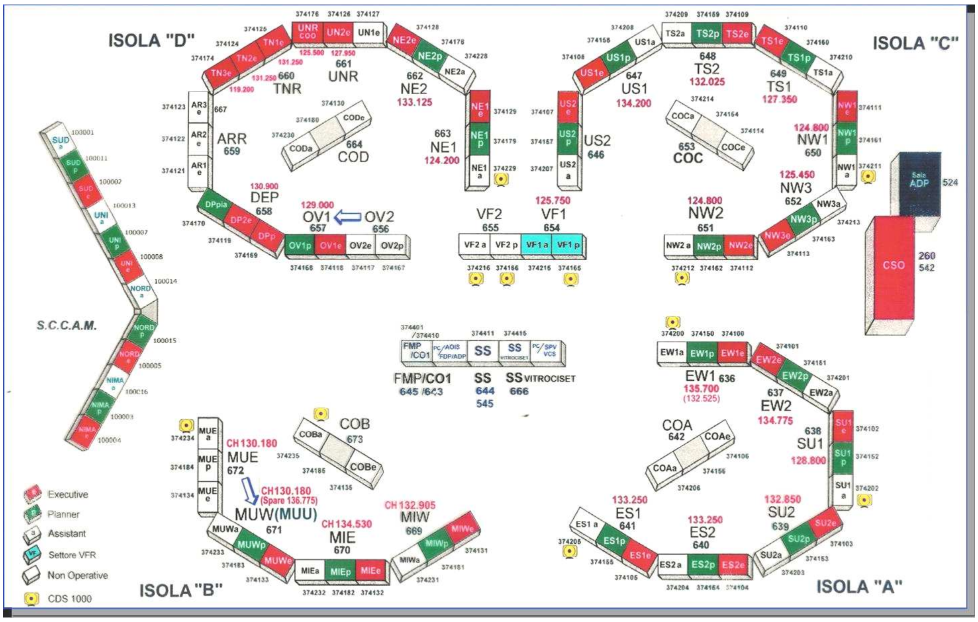


Figure 31 - Typical layout of an ACC.

References

[1] ICAO Global Air Traffic Management Operational Concept Document, Doc. 9854, International Civil Aviation Organization.

[2] ATM Strategy for the Years 2000+, vol. I and vol. II, Eurocontrol, 2003.

[3] SESAR D3 – The ATM Target Concept, SESAR Consortium, 2008.

[4] http://www.eurocontrol.int/corporate/public/standard_page/biz_security.html

[5] http://www.flightsafety.org/fsd/fsd_sept_oct99.pdf

[6] Matthews S. Future developments and challenges in aviation safety. Flight Saf Dig 2002;21(11):1–12

[7] SESAR D5 – The SESAR MasterPlan ATM Target Concept, SESAR Consortium, 2008.

[8] Tamvaclis, C., McFarlane, N., Josefsson, B., Use of aircraft derived data for more efficient ATM operations; Digital Avionics Systems Conference (DASC 04), October 2004.

[9] ARINC, ATS datalink applications over ACARS air ground network, Specification 622-4, USA, AEEC, 2001.

[10] ARINC, Aircraft Communications Addressing and Reporting System, Characteristic 724B-5, USA, AEEC, 2003.

[11] SESAR Definition Phase – WP 1.1.3 – Security.

[12] EUROCONTROL, ATM Security Risk Assessment methodology, Edition 1.0, May 2008.

[13] EUROCONTROL, ICT Security Guidance, Edition 1.0, May 2008.

[14] EUROCONTROL, Security Management Handbook – A framework, Edition 1.0, May 2008.

[15] EUROCONROL, ATM Threat Model – Part A, Edition 0.4, Draft, May 2008.

[16] EUROCONTROL, Critical Asset Identification for ATM, Edition 0.4, Draft, May 2008.

[17] Algirdas Avižienis, Jean-Claude Laprie, Brian Randell, Carl Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing", IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 1, pp. 11-33, Jan.-Mar. 2004

[18] Trusted Computing Group - Trusted Network Connect Specifications in public review. http://www.trustedcomputinggroup.org/resources/trusted_network_connect_specifications_in_public_review

[19] OSGi Specification Release 4. http://www.osgi.org/Download/Release4V42

[20] IEEE 802.1X Specification. http://standards.ieee.org/getieee802/download/802.1X-2001.pdf

[21] RFC 2865, "Remote Authentication Dial In User Service (RADIUS)", by C. Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.

[22] [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP) Specification. http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5BMS-PEAP%5D.pdf

[23] [MS-SOH]: Statement of Health for Network Access Protection (NAP) Protocol Specification. http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-SOH%5D.pdf

[24] Runtime Environment Specification Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 2003

[25] Virtual Machine Specification Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 2003

[26] Application Programming Interface Java Card™ Platform, version 2.2.1 Sun Microsystems, Inc., June 23, 2003

[27] Global Platform Specification 2.2, March 2006

[28] Global Platform UICC Configuration Version 1.0 October 2008, GPC_GUI_010