# SecureChange

## Security Engineering for Lifelong Evolvable Systems

## A Future and Emerging Technologies Research Project funded by the European Union

## Final Publishable Summary

# Executive Summary

SecureChange is a Future and Emerging Technology Research Project co-funded by the European Union. The target of the project is to provide a high level of assurance for a highly evolving system. Figure 1 visualizes the drive of the project: today's system have usually a high level of security but little flexibility or, as a stark alternative, a quick and flexible evolution but little or no assurance.

The aim of the project is to support evolution while maintaining security at all levels of the software development process from requirements engineering down to deployment and configuration

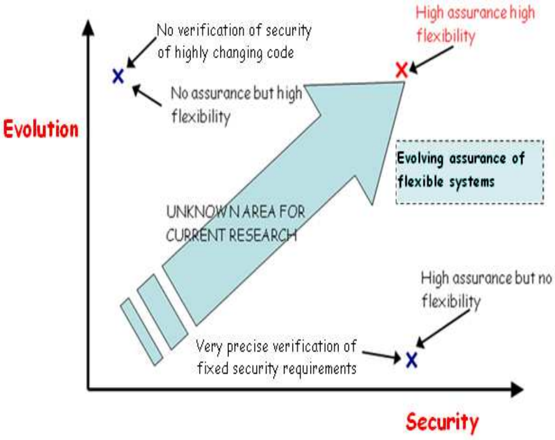

No verification of security of highly changing code
High assurance high flexibility
Evolution
No assurance but high flexibility
Evolving assurance of flexible systems
UNKNOWN AREA FOR CURRENT RESEARCH
High assurance but no flexibility
Very precise verification of fixed security requirements
Security

**Figure 1 - SecureChange's Target**

The key result of SecureChange is to solve these challenge by two ideas that allows us to successfully dominate evolution and assurance:

- Our technical solutions "***focus on the delta***", the difference between the old and the new release of the software, to show what need to be tested, what verified properties are preserved and which are not. Engineers can concentrate efforts where really is needed and engineers can beat the fast pace of evolution.
- Our engineering processes "***don't integrate rather orchestrate***" in order to beat rigidity and allow smooth adoption of technical results for some steps while keeping the benefit of standardization or customization of single proprietary steps in the process.

In the course of the first year the project has developed new models, methodologies and processes to guarantee security during software evolution. During the second year the SecureChange partners have consolidated these results into a conceptually integrated process and sharpened the project focus to address specific challenges from the industrial case studies of the project. The third and final year of the project focused on the industrial validation of the project results on the basis of real industrial scenarios in the domains of Air Traffic Management, Smart Cards Software Evolution, Home Appliances

The project partners delivered roughly 50 additional presentations and published more than 100 papers  addressing different parts of the project (26 journals, 87 conferences publications; 4 book chapters and more reports), delivered 11 tutorials and 11 invited talks. Project partners have developed 8 courses and additionally, 8 lectures where SecureChange results were integrated. In addition, there are 21 PhD theses which have been completed or close to completion – all of which are centred around research topic of SecureChange.

Project partners have been very active in developing research prototype tools to provide feasibility study and practical validation of the scientific results. SecureChange proudly announces that as many as 8 tools have been developed completely within the scope of the project, while an additional pre-existing 9 tools have been continued to be developed.  Most project tools (the Move Tool, the SecMer tool (and the underlying engines EMF-IncQuery and OpenArgue), the CARISMA tool, etc. have been made available on the web and there is a significant interest in their usage. The Rinforzando Tool developed by Thales is now in the process of de-risking for direct adoption in production environment. The results on the EvoTest tool by SmartTesting have been ported to the production environment.

The promising results of the SecureChange integrated process have contributed to the foundation of a spin-off company: QE LaB Business Services GmbH (http://www.qe-lab.com/).

# The Challenges of SecureChange

It is taken for granted that the future will be characterized by a quick pace of evolution: it should be possible to quickly design new services, or swiftly integrate new devices providing new and interesting contents to end users. It is also assumed that the break-neck pace of evolution of software products should be supported by maintaining security and trust properties. The design process should be agile and changes in threat model, security requirements, or just functionality should quickly percolate down the software engineering lifecycle.

These two assumptions are somehow at odds with the strong forces that are currently shaping the engineering process in industry. In order to cope with complexity and quality control the system, software, and service engineering process in industry has been (is, and will likely be) subject to strong push towards rigidity, especially when strong security requirements are at stake. The need to show compliance with standards e.g. ISO 15288 and ISO 12207, respectively for system and software engineering makes the engineering process rigid.

On one hand stakeholders demand flexibility to accommodate changes; on the other hand they demand quality assurance and compliance to standards in process and product. Process rigidity is further increased when security aspects standards are further taken into account. The use of ISO 2700x, EBIOS, CRAMM, BSIMM or SDLC might be mandated by customers or regulations and the design process must also be compliant with those standards.

For complex systems the engineering process is often supported by artifacts (UML models of the system to be, DOORS format for requirements, etc), and companies tend to customize these artefacts to fit their needs and application domains (e.g. by using Eclipse GMF), in order to decompose, compartmentalize and possibly subcontract the work. Some parts of the processes might also be outsourced so that a shared artefact may no longer exist.

In this scenario, integrating security and trust concerns which address simultaneously the calls for fast changes and hard compliance is difficult. While it is widely recognized that security considerations must be considered from the start, most research proposals have focused on new fully integrated security-system engineering processes (starting from the classic Van Lamsweerde ICSE'04 paper). This is also the “default solution” in many European Projects: yet another integrated process for security-[service or content or things] engineering.

Yet, all integrated processes have significant difficulties in adoption. The main reason behind these difficulties is that security-related activities (e.g. assessment, engineering and assurance) must comply with the constraints and pace of the existing mainstream engineering processes, methods and tools (e.g. [18,17]). The rigidity factors that we have mentioned above have shaped and customized each step of the engineering process and de facto unchangeable, as the switching cost would be too high. So there is no chance to adopt an entirely new security engineering process that can cope with the dynamicity and evolution challenges of the Future Internet.

In order to solve this apparently unsolvable problem we need to understand better the process of change and how it interplays with security. What actually changes? How security is actually affected by the change?

Showing data of proprietary product would be difficult and therefore we exemplify here the results of our empirical study on Mozilla Firefox, one of the most popular browsers with millions of users over the world. Browser are complex systems, essentially akin to operating system, and with a strong drive towards evolution to support new features. Our analysis spanned more than 5 years of development and 6 major versions. In Figure 2**Errore. L'origine riferimento non è stata trovata.** we show the lifetimes of the different versions of the browser.

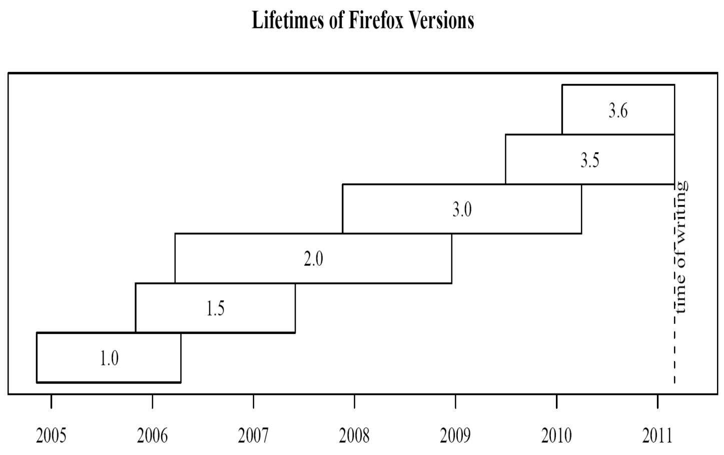


**Figure 2 - Version Evolution in Firefox**

Each version is represented by a box which is born and dies in a very short time span. Version replacement is even faster for Firefox's open source competitor: Google Chrome.

If software is replaced quickly, and the security engineering process is rigid, then the SecureChange idea of providing a novel security engineering process would have no chances of impact: vendors could develop their software with good old security engineering techniques and users could just switch to the new product.

In Figure 3 we see the real truth if one digs down in the code: each version is not really new. Rather it is old software that evolves to a new version. Evolution is neither major nor minor but still significant (only 30% is really new). In this setting it is clear that it is important to support requirements engineers in understating how their requirements have changed, to help test engineer to identify which tests are obsolete, which test are new and which tests are untouched, so that you don't need to re-test millions of code lines that have not changed, but rather spend effort on what is really new (or even better, more risky).

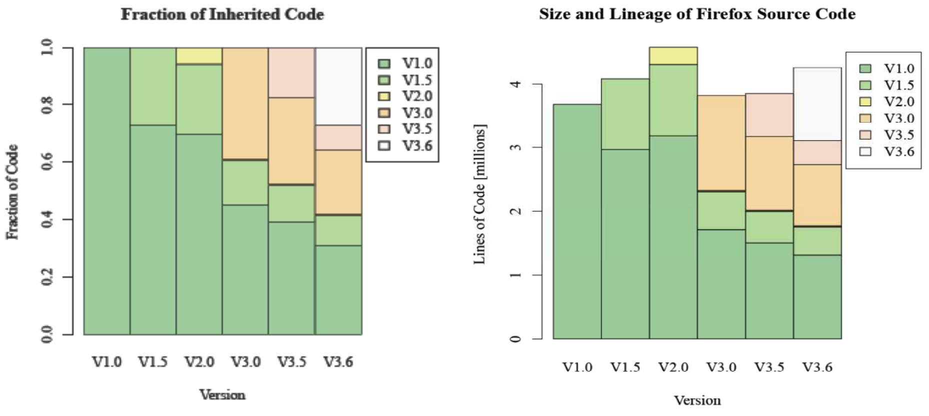


**Figure 3 - Code Evolution in Firefox**

In the same study on Firefox, Chrome and Explorer it emerged that a significant number of software vulnerabilities were inherited from one version to the other and this shows the importance of managing changes and improving the quality of the software. SecureChange challenge is more relevant than ever.

## SecureChange in a Nutshell

The key result of SecureChange is to solve these challenge by two ideas that allows us to successfully dominate evolution and rigidity:

- Our technical solutions "***focus on the delta***", the difference between the old and the new release of the software, to show what need to be tested, what verified properties are preserved and which are not. Engineers can concentrate efforts where really is needed and engineers can beat the fast pace of evolution.

- Our engineering processes "***don't integrate rather orchestrate***" in order to beat rigidity and allow smooth adoption of technical results for some steps while keeping the benefit of standardization or customization of single proprietary steps in the process.

The orchestrated process is based on the ***separation of concern principle***. An important advantage of separation of concern is that in-depth expertise in the respective domains is not a prerequisite. The orchestrated process allows the separate domains to leverage on each other without the need of full integration. Focussing on the deltas allows engineers to understand where the consistency of of concerns must still ensured. For example security risk managers, requirement managers, and system designers share a minimal set of concepts which is the interface between their own processes: each process is conducted separately and only when a change affects a concept of the interface, the change is propagated to the other domain following the ideas behind conceptual mappings and relations

SecureChange light engineering process will guide you in the ongoing process of managing software evolution, while the requirements engineering methodology and the risk assessment method will provide you with a path to identify the key risks and requirements. The testing tools will help you in managing the test engineering process in a tight loop with requirements, risks and design models. Verification technologies will provide an alternative to testing both at deployment time and on device.

## SecureChange's Results in the Big Picture

In order to understand how SecureChange results can be used in the big picture of mainstream system engineering let's consider the typical system lifecycle. as illustrated in Figure 4: (i) architecting, (ii) specification, (iii) design, (iv) realisation or acquisition, (v) integration and verification, (vi) validation and deployment, (vii) operation and maintenance, and (viii) disposal. During the evolution process, a system may occupy several of these phases at the same time: earlier specs might be going already through security testing while new requirements might still be at the architectural phase. Security risk management activities can be conducted regardless of the system lifecycle phase although the pursued goals may differ.

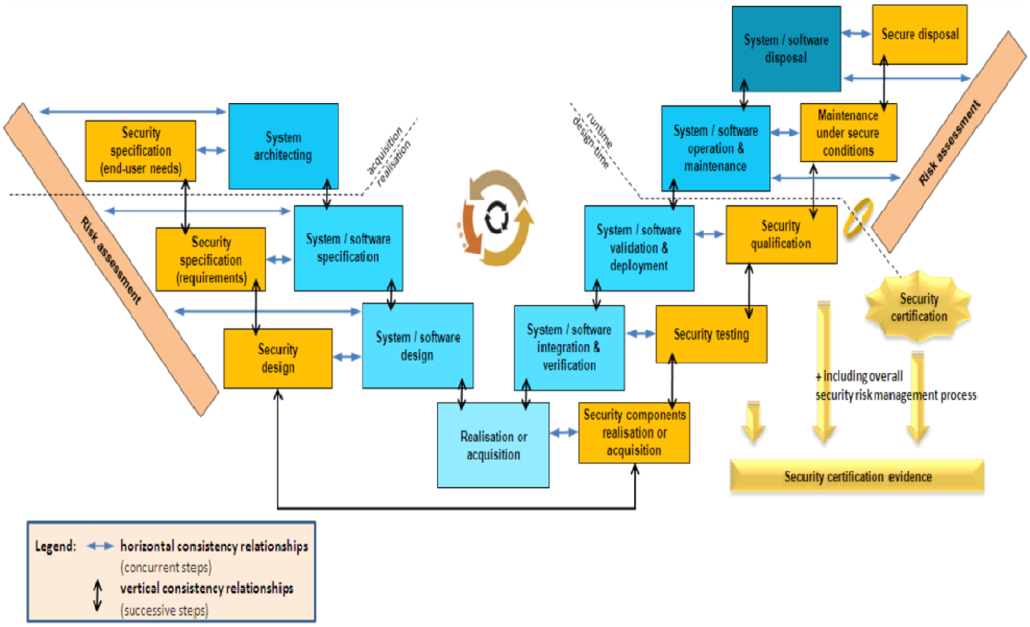

System / software disposal
Secure disposal
Security specification (end-user needs)
System architecting
acquisition / realisation
runtime / design time
System / software operation & maintenance
Maintenance under secure conditions
Risk assessment
Risk assessment
Security specification (requirements)
System / software specification
System / software validation & deployment
Security qualification
Security certification
Security design
System / software design
System / software integration & verification
Security testing
+ Including overall security risk management process
Realisation or acquisition
Security components realisation or acquisition
Security certification evidence
Legend:
horizontal consistency relationships
(concurrent steps)
vertical consistency relationships
(successive steps)

**Figure 4 - Mainstream and Security System/Software Engineering Processes**

The first phase is classically performed by the customer, using architectural frameworks (AF) such the NATO AF. It produces end-user requirements as documented in a call-for-tender. During this phase, the main goal of the security activities is to elicit security needs, possibly consolidated by a threat assessment. The remaining phases, except operation, are mainly performed by the system vendor/provider.

During the specification phase, the main goal of the security activities is to define the system requirements, and thus gain early assurance that the proposed architectural solution is sound with respect to security concerns. This step encompasses a high-level risk assessment backing-up the specification of security requirements. At this point in time, it is important to be able to quickly update models and bring them in synch. The end customer might be involved in the loop and must be able to do some form of what-if scenarios. For example it must be possible to identify possible evolutions and discuss possible tactical solutions in the choice of the components (e.g. the residual risk that a particular component might become useless depending on the outcome of a standardization body).

In contrast, during the design phase, the system degrees of freedom slowly freeze. As time goes by, any major change in design has increasingly significant costs, may require going back to the customer and could lead to unacceptable delays. Changes must be managed differently. The main goal is to make sure that the security properties are preserved across evolution. We accept the change because we know that security properties won't change. Technically this is an obvious observation. We could just re-verify the design after the change and see if the properties still hold. The challenge is to just specify the "delta" and use patterns or stereotype to capture only the "delta" of the change and specify the conditions on the delta that preserve security properties. Proven security design patterns may be used. Security risk assessment is performed in parallel, re-defining security objectives until residual risks are acceptable. Some early validation techniques may be applied in order to gain early assurance that the system design is sound.

The main goal of the security activities during the realisation or acquisition phase is to implement or acquire the countermeasures. In some cases, when the proposed security controls are elementary or available off-the-shelf, this activity may be carried out as part of the mainstream engineering process. When SOA technology is the targeted platform, security-as-a-service might be the right solution.

During the integration & verification phase, the main goal of the security-related activities is to integrate and test the countermeasures. As for realisation or acquisition, the integration of the security countermeasures may be carried out as part of the mainstream engineering process; however testing represents a security-specific task, aiming at proving that the information system protects data and maintains functionality as intended.

During the validation/quality check phase, the main goal of the security-related activities is the security qualification of the system, which will potentially lead to certification. The qualification of a product gives evidence of the robustness of the security services of the product. It is based on: (i) the verification of the conformity of the product with the security characteristics specified in the target, on the basis of an evaluation realized by a laboratory approved by a certification authority, e.g. ANSSI in France; (ii) the approval, by the certification authority, of the relevance of the security target with respect to the planned use and the requested level of qualification. This qualification allows: a) to separate the purely technical assessment of the system from a wider assessment of its ability to protect sensitive information in given conditions; b) to recognize that the same system can allow for the protection of information of different levels, and thus can obtain various levels of approval, according to the conditions of use.

During the operation & maintenance phase, the main goal of security risk management is to monitor the effectiveness of the countermeasures to determine the extent to which the controls are implemented correctly, operating as intended, and producing the desired outcome with respect to meeting the security requirements for the system or enterprise. In case security is found to be flawed, the previous activities may be performed anew to ensure an acceptable level of risk.

Figure 5 shows how the different workpackages of the project address each issue.

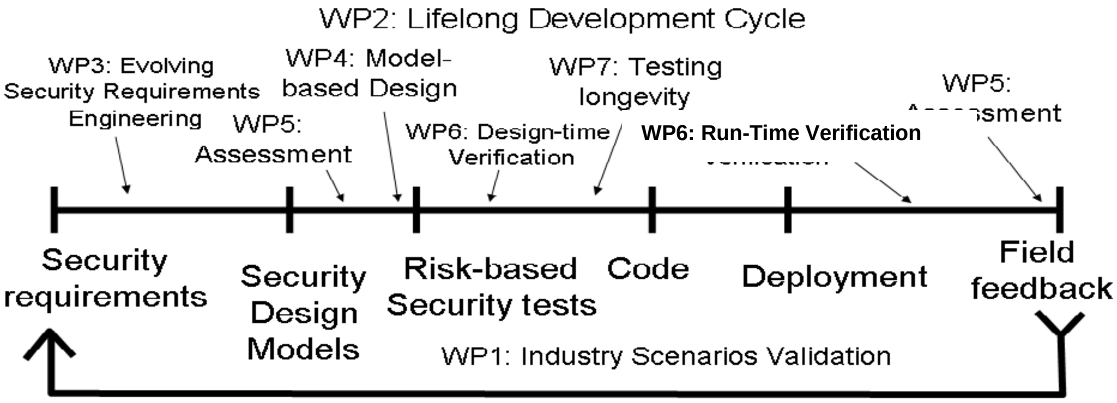

WP2: Lifelong Development Cycle
WP3: Evolving Security Requirements Engineering
WP4: Model-based Design
WP7: Testing longevity
WP5: Assessment
WP5: Assessment
WP6: Design-time Verification
WP6: Run-Time Verification
Security requirements
Security Design Models
Risk-based Security tests
Code
Deployment
Field feedback
WP1: Industry Scenarios Validation

**Figure 5 - SecureChange Contribution to Security Engineering**

In the course of the first year the project has developed new models, methodologies and processes to guarantee security during software evolution. A large number of academic publications in prestigious journal and magazines (e.g. IEEE Computer) have resulted from this effort. During the second year the SecureChange partners have consolidated these results into a conceptually integrated process and sharpened the project focus to address specific challenges from the industrial case studies of the project. The third and final year of the project focused on validation of the project results on the basis of real industrial scenarios namely three case studies from different domains. The validation has testified the applicability and feasibility of the SecureChange approach and further led to the improvement of the tools and prototypes which have initially been developed in the first and second year.

Figure 6 shows the temporal organization of the project.

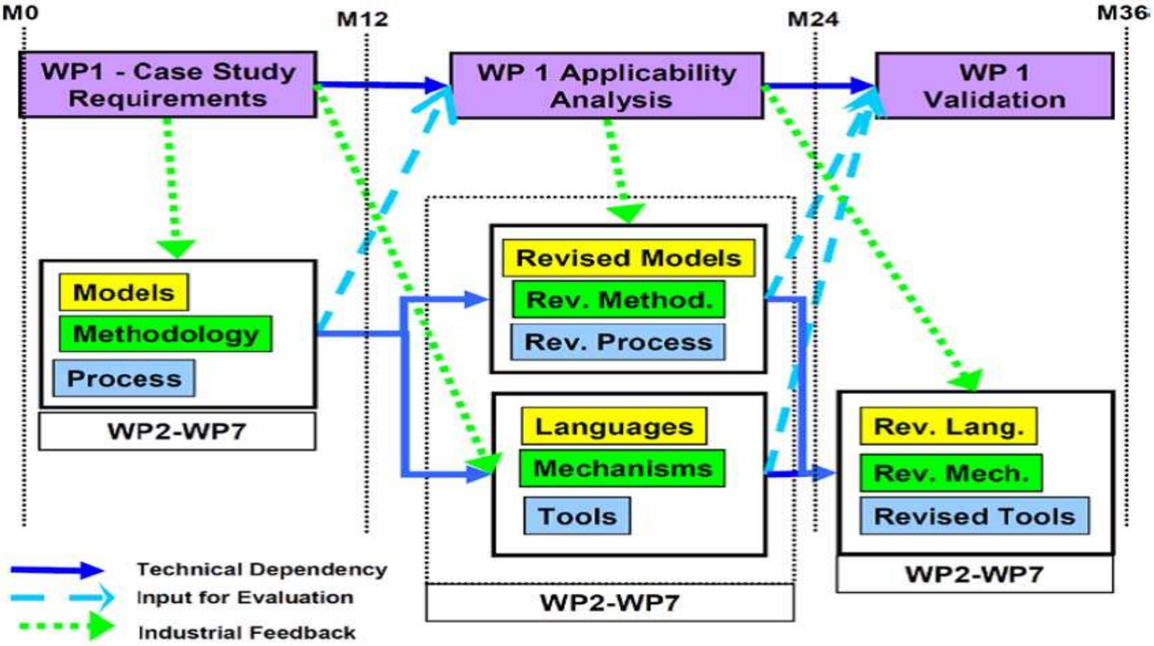


**Figure 6 - SecureChange Temporal Organization**

## SecureChange Industrial Case Studies

The case studies are drawn from largely different domains and highlight different change requirements and security properties with the aim of showing that the SecureChange research results do actually work for the widest range possible. They are representative of relevant but not exclusive application domains of SecureChange output:

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

Each case study comes with his specificities of change and impact. In particular, the requirements changes characterising the three case studies fall into three different types: ***Changes in Process***, ***Changes in Configuration***, and ***Changes in Software***.

The ATM case study involves various change requirements due to the introduction of new tools such as the Arrival Manager in the major restructuring of the ATM that will take place with the SESAR initiative. The ATM case study is concerned with how such new tools affect organisational as well as operational aspects. We call this change the ***Changes in Process***.

The HOMES case study is focused on change requirements on policies and critical on software modules providing critical security features of the Home Gateway that is the unique network access point for a wide range of the devices in the home of end customers. HOMES deals mainly with ***Changes in Configuration***.

The POPS case study focuses on an UICC card made of integrated circuit (hardware) and an operating system base on JavaCard and GlobalPlatform specification. This object must be security certified before its issuing, but its life-cycle includes change that could be done in the field. These changes result from adding a new application while preserving the implemented security. Therefore here we are mostly interested in ***Changes in Software***.

Therefore, the solutions provided to solve their change-related security problems are quite different. For example, the *consistency* of the risk analysis modelling artefact for the ATM case study is critical while the performance criteria is more suitable for the embedded running code for POPS or the *availability* criteria for the services for HOMES. In order to achieve few, streamlined, and orchestrated research strands, the technical integration among the WPs have been focussed and driven by the case studies.

# SecureChange Three Years At a Glance

The 1st year was devoted to represent the evolution of the different artefacts produced during the software engineering process and to define an initial methodology for change management:

- Description of the SecureChange scenarios and the related requirements (WP1)
- An Architecture and a Software Development Process for security-critical Lifelong Systems (WP2)
- A conceptual model and a methodology for characterizing and transformation of evolving requirements (WP3)
- Methodologies for modelling adaptative security designs and requirements (WP4)
- A language for documenting forecasts of future evolvement of a system (WP5)
- A conceptual model characterizing a new programming model and a notation supporting the programming model
- Integration of the evolution into a model-based testing approach
- Project website published, internal training workshop organised, dissemination activities started, plan for industrial exploitation (WP8)

The 2nd year of the project saw the first proof-of-concept tools for supporting processes and advanced methods and tools for designing, testing and verifying the security of evolving systems. A feasibility study of the results was carried out. This boiled down to the:

- Evaluation of the feasibility of the approaches and methods and their applicability to the industrial scenarios (WP1)
- Definition of an appropriate validation strategy for the solutions (WP1)
- A configuration management process for lifelong adaptable systems (WP2)
- Improvement of the conceptual model for the characterization of evolving requirements and definition of advanced algorithmic features that make it possible to propagate and explore the impact of changes in requirements to design models (WP3)
- Formal foundations for modelling adaptive security designs and requirements and a first prototype tool for analyzing these models (WP4)
- Development of methods for the assessment of systems with respect to future evolvement and a framework for documentation of the system and assessment results (WP5)
- Prototype implementation of a verifier that supports the programming model and a methodology to handle verification of adaptive security on new code and impact analysis of new security requirements on loaded code (WP6)
- Development of a methodology and a prototype for model-based testing of evolutions (WP7)
- Project website published, Internal training workshop organized, dissemination activities started, plan for industrial exploitation (WP8)

The objective of Year 3 was to refine methodologies, algorithms and tools that address the concerns from the industrial evaluation that was carried during the year.

- The validation strategy for the SecureChange solutions and their full evaluation of the usability of the SecureChange solutions in a realistic industrial context. (WP1)
- Development of a fully integrated tool-supported software development and security analysis process for lifelong systems, including the artefact-centric management. (WP2)
- Improvement of the algorithms for the incremental evaluation and transformation of requirements models, and proof-of-concept CASE tool implementation respectively. (WP3)
- Improvement of the prototype of the security design modelling solution and extending it by an approach for monitor generation to supervise the security of evolving systems. (WP4)
- Proof-of-concept integration of relevant elements of the security design modelling solution into the industrial MDE context of the Thales modelling environment. (WP4)
- Development of techniques and tools for semi-automatic revalidation of existing risk assessments with respect to changes. (WP5)
- Tools to re-verify security on device for software modules affected by changes, including a strategy for the interplay of development-time and on-device verification. (WP6)
- Generation of tests for the case studies. (WP7)
- Dissemination of results and experience gained from the project to scientific and industrial communities. Exploitation of results into the business domain of the industrial partners. (WP8)

## Validation of SecureChange Solutions

During the final year of the project, the industrial case studies supported a validation of the SecureChange artefacts. Deliverable 1.3 (Report on the Industrial Validation of SecureChange Solutions) describes how the technologies developed within the project are applied to specific problems (in terms of changes requirements and security properties) identified for each case study.

Each case study comes with his specificities of changes and impacts. Therefore the validation of the developed technologies concerned different validation criteria. Figure 7 Figure 7shows how the WP1's deliverables supported an industry-driven validation of the SecureChange artefacts.

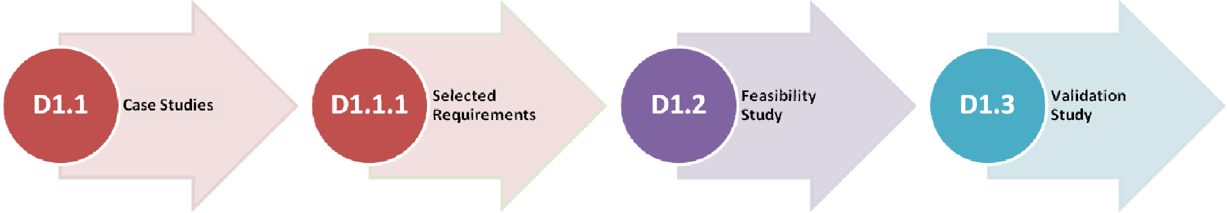


**Figure 7 - WP1 deliverables highlighting an industry-driven validation of SecureChange artefacts**

Deliverable D1.1 identified relevant change requirements and security properties drawn from the three different industry domains. Deliverable D1.1 intentionally identified a wide range of changes requirements and security properties in order to support the discussion of relevant evolutionary concepts underlying the SecureChange project. Deliverable D1.2 focused on the selected requirements and narrowed the scope of analysis according to industrial feasibility criteria. The point was to ease the adoption of SecureChange artefacts by aligning them with current industrial practices and expectations. Deliverable D1.2 was useful in order to link the industry problems (D1.1 and D1.1.1) to the validation of the SecureChange artefacts (D1.3) by narrowing the scope according to industrial feasibility criteria. Deliverable D1.3 assessed SecureChange artefacts according to a wider range of validation criteria.

The validation strategy involved the definition of specific validation scenarios and exercises (in terms of change requirements and security properties as identified in deliverable D1.1.1 Selected Change Requirements and Security Properties) for all validation objectives, that is, SecureChange results delivered by the other WPs (WP2-WP7). WP1 case studies (i.e. ATM, HOMES and POPS) defined such validation scenarios and exercises in collaboration with the other technical WPs. Deliverable D1.3 reports the validation scenarios and exercises for each case study and relevant validation objectives. The coverage of the case study with respect to the technical WPs has been clearly identified in the deliverable D1.1.1.

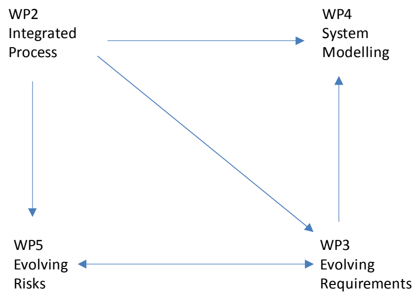
The identified validation scenarios and exercises stressed how SecureChange artefacts deal with change requirements while guaranteeing critical security properties. Specific validation criteria identified in the deliverable D1.2 (Report on the applicability of SecureChange technologies to the scenarios) and reported in the deliverable D1.3 highlight critical aspects of SecureChange artefacts with respect to change requirements and security properties.

The SecureChange results have been evaluated according to the identified validation criteria. The validation scenarios and exercises involved domain experts and end-users for the specific case studies (i.e. ATM, HOMES and POPS). This allowed us to collect feedback drawn from relevant industrial experiences. It also allowed us to assess how SecureChange artefacts would fit current industrial practices. Deliverable D1.3 (Report on the Industrial Validation of SecureChange Solutions) reports and discusses the validation results.

D1.3 consists of the results of the validation including the evaluation of the applicability in realistic industrial contexts, together with recommendations for the future improvements and refinements of the project results. It identifies and defines the validation strategy for the SecureChange final results, with the identification of the validation objectives for each WP and of the validation exercises and analyses of the related outcomes. The validation analyses the final and consolidated project results. It demonstrates that SecureChange artefacts can work efficiently in real life environments, while addressing the problem for which they have been developed.

The validation has taken place in the final year of SecureChange and it has delivered and influenced the research work in the last phase of the project lifecycle. The validation has provided insights for future further improvements and refinements of the SecureChange results. Particular attention has been given to the usability, of the project results, in real industrial contexts captured by the three different case studies: ATM, HOMES and POPS. Moreover, the validation criteria also include the applicability in real life and specific validation exercises designed to provide industrial feedback about essential aspects of the project results. The validation organisation has been tailored to capture specific validation objectives with respect to the SecureChange artefacts and industrial domain features. The overall validation organisation, activities and objectives build over the previous WP1 deliverables (D1.1, D1.1.1, D1.2), which have defined the scope and feasibility of SecureChange artefacts.

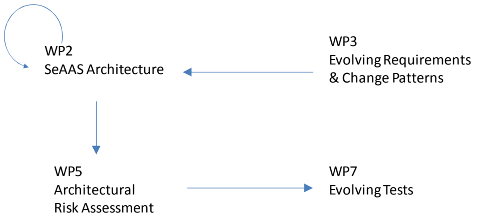
- **ATM** – The ATM case study focused on four work packages (i.e. WP2 Architecture and Design Process, WP3 Requirements, WP4 Model Design and WP5 Risk Assessment) and their artefacts. Due to the nature of the ATM case study (mainly concerning with technological changes from an organisational viewpoint) the WPs focusing on requirements, design and assessment aspects, the ATM case study has contributed towards the validation of relevant artefacts supporting specific design and assessment activities while preserving critical security features.



Each WP has produced different artefacts (e.g. methodologies, tools). Hence, it has been necessary to tailor the validation activities to the different peculiarities of the artefacts and their developmental stages. This required WP-tailored validation activities.The validation objectives of the ATM case study have concerned the relevance of SecureChange artefacts and their assessments by ATM domain experts (e.g. Air Traffic Controllers) and potential end-users (e.g. IT and operational experts within an Air Traffic Control Service provider). The validation activities have been tailored for each WP and related artefacts. This is to take into account the different nature of the artefacts (e.g. methodologies, modelling languages, tools). Moreover, it has been necessary to support different developmental paths of the artefacts. All SecureChange artefacts delivered by the ATM-related WPs have been validated by subsequent activities in order to support their developments through subsequent refinements (i.e. adjustments due to feedback). The main validation activities involved: Methodology Evaluation (modelling), Walkthrough and Tool Live Demo with ATM Experts. Each validation activity involved ATM experts in order to assess SecureChange artefacts from a practitioner viewpoint and to identify opportunities for exploitation of project results within the ATM domain. The ATM case study identifies specific user needs and expectations for the ATM industrial domain. In particular, the ATM validation highlights how SecureChange solutions can be used in the application domain and expected improvements to comply with industry practices.

The case study draws a scenario from the ATM domain where the compliance with and tool support for the security tailored V-model can be demonstrated. It includes the acquisition phase in which an air navigation service provider (ANSP) conducts a requirements specification and analysis, including threat source and feared event assessment, followed by the realization phase with service modelling and risk assessment at the specification and design levels as conducted by the system provider. Two iterations over these activities where the security engineering activities before and after the changes are demonstrated.

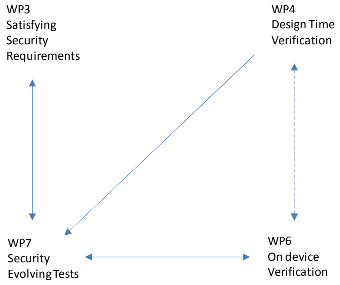
- **HOMES** – The validation objectives for the HOMES case study have been focused on the effective usage of the artefacts (including their applicability and degree human effort involved), as well as specific industrial criteria such as perceived value, performance, or usability. These high-level objectives are decomposed into measurable indicators.



The main validation activities were concerned with three major categories: Methodology Evaluation (modelling), Walkthrough and Tool Live Demo with HOMES Experts. Methodology evaluation consisted of modelling exercises focusing on specific changes and security requirements in order to refine and consolidate the underlying modelling languages and their methodologies, respectively. Walkthrough activities involved step-by-step evaluation of the SecureChange methodologies with HOMES experts. This allowed to assess the proposed methodologies with domain experts and to identify alternative usages (with respect to current practices within the HOMES domain). Finally, tool live demo activities and exercises allowed the validation (in terms of usability and acceptance by HOMES experts) of the tools supporting the SecureChange methodologies.

The integrated process and the orchestration of different tools are illustrated by a change scenario based on the initial change event driven by a gateway operator. The complaints of customers and third-party providers might pose a threat to the business model of the gateway operator and thus cause the risk analysis to detect weaknesses and vulnerabilities of the system.

- **POPS** – The POPS case study involved the software of an UICC card made of an OS, a Java card platform that executes applications (applets), a set of applets and a Globalplatform layer responsible of the card content management. The global scenario of this case study concerns a change of the software embedded on the card that results from adding a new application on the card or updating the platform layer due to an evolution of the corresponding specification. We are then in the context of software update as change requirement. The overall objective was then to provide means that will facilitate the assessment of those changes with respect to specific security properties that were the subject of study of WP4, WP6, WP7 and with WP3 only in order to close the orchestration loop between testing and requirements.



The life cycle of updating an UICC card involves several actors:  the developer of the software platform (the card manufacturer), the developer of the application to be added, the end-user (the card holder) and the card issuer that provides the card to the end user. The validation activities consisted in playing the role of a subset of these actors for using these artifacts and evaluate them in realistic industrial contexts. For each artifact, generally a specific tool, the security engineer takes the role of the application and the platform developer, and figures out a wide usage in the R&D. The people we have involved in the validation activities have several kinds of background and expertise in order to have the most representative sample of a generic R&D population. This validation intended to confirm the feasibility studies described in D1.2 and possibly provide further recommendations.

The running example in the POPS case study is the life-cycle of the Global Platform smartcard specification. Beside other aspects, the evolution of the life-cycle specification from version 2.1.1 to version 2.2 is considered. The different activities that performed while changing the specification are demonstrated, including changing and verifying the specification models, adapting the respective test cases, and (re-) verifying the applets to be executed on the changed platform.

The SecureChange validation identifies the validation objectives with respect to the project outcomes (for each WP) and the way the validation activities have been organised and carried out (in the final part of the project) in order to address these objectives.  Due to the complexity of validating diverse project outcomes, the validation strategy has taken into account changes and subsequent contributions. As natural consequence of the complexity of the SecureChange approach, tools and solutions that will be the outcomes of each SecureChange work package can be significantly different. Therefore, each work package has contributed to this document by designing, planning and performing different validation activities, compliant with the characteristics and scopes of the work package itself. The validation involved subsequent validation activities that have been planned for each case study and for each WP. The validation activities combined together highlight validation strategies and processes tailored to the specific validation objectives and case studies.

The validation activities highlight that SecureChange results address to a certain extent the lack of support in engineering evolving systems and guaranteeing security properties. The three case studies highlighted how WP artifacts support industrial practices. Moreover, the validation activities allowed us to identify alternative usages for SecureChange solutions. Overall, SecureChange artifacts provide suitable support to specific engineering activities that concern the modeling and verification of security features with respect to changes. The case studies and the conducted validation activities highlighted how the different artifacts support SecureChange objectives.

## SecureChange Integrated Process

The SecureChange process can be applied on various levels of granularity. In the ATM case study we have demonstrated how change scenarios can be applied on the level of single model/data elements, whereas in the HOMES case study changes scenarios concerned the coarse-grain level of model types. This flexibility in the level of granularity makes the SecureChange process suitable to seek a bootstrapping strategy for legacy systems with a transition from coarse-grain change management to fine-grain change management as soon as more detailed model information is available.

The SecureChange process is accompanied by tool support through the MoVE (Model Evolution Engine) tool. MoVE is a model repository with a generic meta model supporting the integration of models from various information sources through an adapter concept. The models may be created and maintained not only in modelling tools but in any tool from which structured information can be extracted (like data bases or spreadsheets). The models are committed to a central model repository assisting with conflict detection and the maintenance of links between model elements. The change-driven process is materialized through state machines attached to model elements. The state machines initiate chains of change propagation and change handling actions across models and tools. MoVE has been evaluated within the ATM case study.

Several Change Patterns have been formally specified and collected in a catalog. The catalog, together with the corresponding methodology and tool, provide support for the principled evolution of an architectural design in light of changes taking place in the requirements specification. For illustrative purposes, trust requirements have been addressed in the catalog, albeit the approach is applicable to the full spectrum of security requirements. The change Patterns methodology has been validated by means of both a controlled experiment (12 students) and a case study involving two engineers of Telefonica.

To close the gap with software development SecureChange has delivered a sustainable security-as-a-service architecture (SeAAS). SeAAS applies best practices from functional architecture design to security architecture design in order to provide as much flexibility and modularity in an evolving context. The architectural platform has been realised both with web service technology and with OSGi technology relevant in the HOMES case study. On top of SeAAS we developed a model-based configuration framework. In this framework models are used both to generate (XML level) policies configuring the SeAAS architecture and to generate interfaces for the user to provide missing parameters for the policy generation (e.g. concerning the selection of cryptographic protocols).

## Evolving Security Requirements

The objective of work package 3 is to develop the concepts and basic building blocks for the management of evolving requirements. During the first and second year of the project, the activity of work package 3 has produced the following artefacts

- the SeCMER conceptual model for evolving requirements,
- the SeCMER methodology for evolving requirements
- a quantitative reasoning for selecting design solutions resilient to evolving requirements
- a first version of the SeCMER tool which supports the methodology steps

During the third year of the project, the work package activities have focused on the integratability of the work package results into industry practice. In order to show how these results can be integrated into existing industrial security engineering processes, we have focused on the integration of the analysis of Security Requirements with Risk Assessment and Security Testing from the perspective of existing processes. We have shown that requirements evolution modelling and argumentation analysis can be orchestrated with risk assessment activities which are part of industrial security engineering processes.

Moreover, the quantitative reasoning technique on requirements evolution and change-driven transformation based on security patterns have been revised to address the need to have automatic decision support tools for change management in the air traffic management domain.

The results of Work Package 3 provide decision support process and tool to handle changes in requirements models and assess their impact on the security of the system. In particular, the SeCMER methodology helps the requirement analysts in managing changes in requirements model by means of decision support artefacts such as change driven transformation and argumentation analysis. Change driven transformations provide automatic detection of requirement changes and violation of security properties, while argumentation analysis helps to check whether security properties are preserved by evolution and to identify new security properties that should be taken into account. Compared with other academic and industrial tools for requirements management, the SeCMER tool provides decision support to the requirement analyst for handling security-related changes. The tool supports automatic detection of requirement changes that lead to violation of security properties using change-driven transformations and suggests possible corrective actions. The tool also supports argumentation analysis to check security properties are preserved by evolution and to identify new security properties that should be taken into account.

The applicability and usefulness of SecMER methodology, the tool and the quantitative reasoning on evolving requirements have been evaluated in a real industrial setting, which is the air traffic management domain. Three validation workshops have been organized with ATM experts at DeepBlue premises.

The SeCMER tool has been improved based on the feedbacks collected during the validation activities with ATM experts. The tool interface has been made user-friendly and the ability to detect new types of security violations such as need to know principle violation has been implemented. The enhanced version of the tool is presented in deliverable D3.4 delivered at M36.

## Managing Evolving Risks

The SecureChange Integrated Process incorporates security risk assessment as part of the overall approach to manage potentially security critical system changes. To this end, SecureChange offers a set of artefacts that combine into a model-based approach to security risk analysis of evolving risks. These artefacts are a method for systematically identifying and assessing changing and evolving risks, language support for modelling and documenting changing risks, techniques for tracing changes from system to risk, and tool to support for all the risk assessment and risk modelling tasks of the method. Tool support is moreover provided for automatically identifying changes to risks and for maintaining consistency of risk models under change.

In addressing the methodological needs of assessing changing systems and how to adequately model the changing risks, the artefacts have been developed independent of specific state-of-the-art methods and modelling techniques. Instead, the aim was to provide adequate methods and techniques at a generic level that can be instantiated in several existing approaches. In order to demonstrate the approach and validating the artefacts in the SecureChange case studies, the method and language was instantiated in the CORAS model-driven approach to risk analysis. Moreover, all tool support is Eclipse-based and has been developed to support the CORAS instantiation of the SecureChange risk assessment method.

Figure 8 illustrates how the risk assessment artefacts combine in supporting the risk assessment method, and how the artefacts are managed with and orchestrated by the tool. Building the system model is a part of the method, and the system model is used for describing (relevant aspects of) the target of the risk analysis. The risk identification is conduced based on the system model, and the results are documented in the risk model. The trace model is a specification of mapping rules between the system model and the risk model, thus establishing traceability between system and risks; when a change requirement results in an updated system model, the change is propagated to the relevant part of the risk model. Year 3 of the project has focused on automated tool support for this picture. In particular, the tool supports the importing of the system model, the specification of the mapping rules, the automatic identification of system changes after importing an updated system model after change, the automatic identification of risks that may be affected by system change, as well as the automatic detection of inconsistencies in risk models with respect to risk changes.

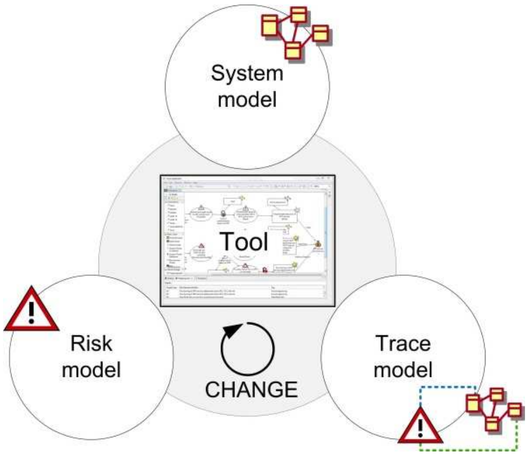


**Figure 8 - Artefacts to support the method for assessing evolving risks**

The validation activities of the risk assessment artefacts during Year 3 have involved the ATM and the HOMES case studies. The ATM validation addressed organizational level change requirements and was conducted during workshops involving external ATM experts. The experts were presented the various artefacts, including demonstrations and brief tutorials, and were actively using the tools in independence to conduct risk assessment exercises. The risk assessment method, language and tool were moreover validated in an ATM case study focusing on the orchestration of SecureChange tools across work packages, covering several steps in a security tailored development process. In particular, these validation activities combined artefacts from WP3, WP4 and WP5 to cover requirements capturing, risk assessment at operational level, system specification and design, and risk assessment at system specification level. The CORAS instantiation of the SecureChange risk assessment framework from WP5 were applied for the operational level risk assessment. In this setting we have moreover demonstrated tool support for automatically tracing changes from the requirement models to the risk models when using the former models as part of the input to the risk assessment. In the HOMES case study, the SecureChange risk assessment artefacts were used in WP2 to provide the risk models and to support the risk assessment activities of the SecureChange integrated process.

## Evolving and Formally Secure Design

During the first year we proposed a notation that allows one to specify multiple possible evolution paths for UML models. The notation is called UMLseCh and is a further extension of the UMLsec profile. During the second year we have specified a formal foundation for this notation that aims at automatic (re)-verification of security annotated diagrams after evolution (see our deliverable D4.2). To achieve this, we give a more precise definition of the UMLseCh semantics itself, which allows us to pin down what we mean by ‘evolution’ from a model M to an evolved M0. As a result of this, given an UMLseCh diagram we can extract one or more deltas Δi containing the model elements to be added, substituted or deleted from/to the original diagram. These modifications to an original diagram M have two main consequences: they may alter the consistency of the diagram from the purely UML syntactical point of view, but more importantly they may alter the security properties of M. We discuss the first problem to some degree, but we focus on the latter. For this, we present sound decision procedures for different security properties that allow us to establish whether a given Δ preserves them or not.

In Year 3 of the SecureChange project we have further developed the UMLseCh notation to the UMLchange profile. It focuses on the evolution aspects and separates them from the security aspects. This separation allows us to decouple the security checks from the origin of evolution on the one hand and more precise evolution descriptions on the other hand.

Beside the original supported change operations – addition, substitution, and deletion – the evolution notation now enables the specification of copying, moving, and editing model elements. These new options allow users a more precise definition of possible evolution paths that can then be checked by the analysis tool.

The decoupling from the evolution notation further enable the security checks to analyze evolution of different sources. The evolution can not only be described using the UMLchange notation, but it can also be derived from the versions of a model by difference computation. A first approach into that direction has be started in the third year of the SecureChange project. In Addition the prototype implementation of the security analysis tool has been further developed into the new analysis tool CARiSMA. It is based on Eclipse and has an open plug-in architecture that allows users to integrate their own security checks. The migration to Eclipse and the Eclipse Modeling Framework (EMF) furthermore provides the basis for integration with other modeling tools, especially those used in industrial contexts. Moreover, we have provided a proof-of-concept implementation of these algorithms as plugins for the existing UMLsec Tool Suite. This allows us an automatic verification of UMLseCh annotated models drawn with the ArgoUML tool. Metrics of the efficiency gain of this implementation as opposed to trivial re-verification are presented.

As a proof-of-concept, we model some fragments of the Global Platform (POPS case study) and verify the preservation of selected security properties under evolution. Some of these fragments are used to illustrate how our formally-based design method can be used to leverage and integrate the approach with other WPs.

Connection between the modeling and verification techniques developed by WP4 with WP3 (Requirements) based on the ATM case study has been shown as part of deliverable D4.2. A risk analysis done with the Thales Security DSML gives high-level security requirements, which are reflected in the System Design and analyzed by means of the UMLseCh approach. The general requirement considered is 'Organizational Level Change' and the properties considered are 'Information Access' and 'Information Protection'. D4.2 further describes how the result of the verification process at the model level can be used to push constraints to the verification at the code level, based on the POPS case study for a GP specific property and secure information flow. The general requirement considered is 'Software update' and the common property is 'Information protection'.

Also using the Global Platform life-cycle (POPS), we illustrates how model-based testing for evolving systems can benefit from formal design techniques. The general requirement considered is 'Specification Evolution' and the common property is 'Life-cycle consistency'. The integration has even been demonstrated on tool level based on our prototypes. It is described in the deliverables D4.2 and D4.3.

In Year 3 of the project it was furthermore researched how security properties that have been checked at design time can be enforced at runtime. One result is the generation of Java monitors from UML state charts in order to supervise if method calls conform to the specification. Another result is a log-based monitoring approach that checks the conformance between the runtime log of a program and the process specification given as UML activity diagram.

## An Industrial Reality Check for the Early Design Steps

In order to evaluate the feasibility of integrating the results of SecureChange in an existing industrial process, Thales has implemented a proof of concept solution. The focus was put on the development of an integrated prototype covering both system design and security risk assessment. In year two, a prototype was delivered which demonstrated this approach using a standard open source UML 2 editor, Papyrus UML, for the system design part. In year three, Thales made tremendous progress towards the industrialization of its solution, through both collaboration within the project and exploitation with business units within Thales. The new prototype features Thales' own solution for system design, SMS (SOA Modelling Suite) integrated with a much improved version of Thales' risk assessment tool Rinforzando.

As part of the collaboration with the partners in the project, Thales showed the applicability of its integrated solution through a large effort on the ATM use case. The tools used in this evaluation included SI*, CARiSMA, Rinforzando, CORAS, and SMS. The study covered a significant part of the security engineering chain in the presence of change. Thales played an active role in coordinating this validation exercise with the other partners in the project. The scope of this exercise included business modelling, security need modelling, risk assessment, system / software architecting and design, and security design. The feedback thus collected allowed Thales to evaluate and refine SecureChange's integrated process from an industrial point of view. Beyond the evaluation of the Thales tools (i.e. Rinforzando integrated to SMS), the evaluation activities were also focused on some of the partner tools, notably CARiSMA, of which Thales conducted an extensive experimentation. A study of the integration of the EMF-IncQuery framework with Thales' internal modeling tools was also performed, resulting in an ongoing collaboration.

Thales also built a successful internal exploitation plan for the Rinforzando tool. A series of meetings was organized with various business units in Thales with a strong acceleration in year 3 as the improved prototype was presented and refined, culminating with the shortly awaited decision by a central management body to proceed with the “derisking” of Rinforzando under the responsibility of at least two business units.

## Security Verification of Evolving Code

An important objective of SecureChange is the development of verification techniques for evolving systems, with a strong focus on the development time and run--time phases of the software lifecycle. For development-time verification, , we have developed in the first year of the project a theory of how to extend a separation-logic based verifier so that it can verify soundly absence of several classes of bugs, even in the presence of unchecked exceptions and dynamic code loading and unloading.

In the second year we have implemented a prototype for these reasoning techniques.

In order to support validation of the verifier on the HOMES and POPS case studies, we have implemented both Java and C front ends for the verifier, and we provide a full implementation of the techniques for dynamic code loading and unloading developed in Task 6.1.

In the third year, the prototype implementation was further extended to support a sufficiently large subset of JavaCard and C to support realistic experiments in the POPS and HOMES case studies

The tool is sufficiently mature that it can verify real Java Card code taken from the POPS case study. For this case study, one of the concerns is robustness (absence of denial-of-service issues) when software updates happen on the card. The prototype tool is being used to verify absence of runtime exceptions and infinite loops in Java Card applets that are loaded on a multi-application smartcard.

Extensive experiments have been performed in the third year on the feasibility and effectiveness of development-time verification in the POPS case study. Several medium-sized JavaCard applets were verified, including both existing applets as well as newly developed applets. These experiments show that the tool is ready to be used in practive, even though further improvements in usability (for instance reducing annotation overhead) remain a point of attention.

For the HOMES case study, the tool was used in the third year to verify the *secure extensibility* property for core security module updates of the home gateway. More specifically, an extensive experiment verifying memory safety and thread safety of a Policy-Enforcement-Point module from the POPS case study was performed, leading to similar conclusions as the POPS experiment.

The second stream of our verification activities focuses on the run-time aspects. Our deliverable D6.3 describes compositional techniques to verify evolving security at load-ing time on small embedded devices (multi-application smart cards). Such small devices have restricted memory and usual run-time monitoring techniques cannot be applied on them.

In order to preserve the security of  information exchange in a dynamic environment, where the applications from different stakeholders can evolve and talk to each other, the device should be able to verify the updates autonomously and in a very lightweight fashion. The proposed techniques cover such parts of the information protection requirement as control flow and special type of non-interference.

Our control flow verification provides assurance, that there is no illegal invocation of application services. Figure 9 below shows the basic idea

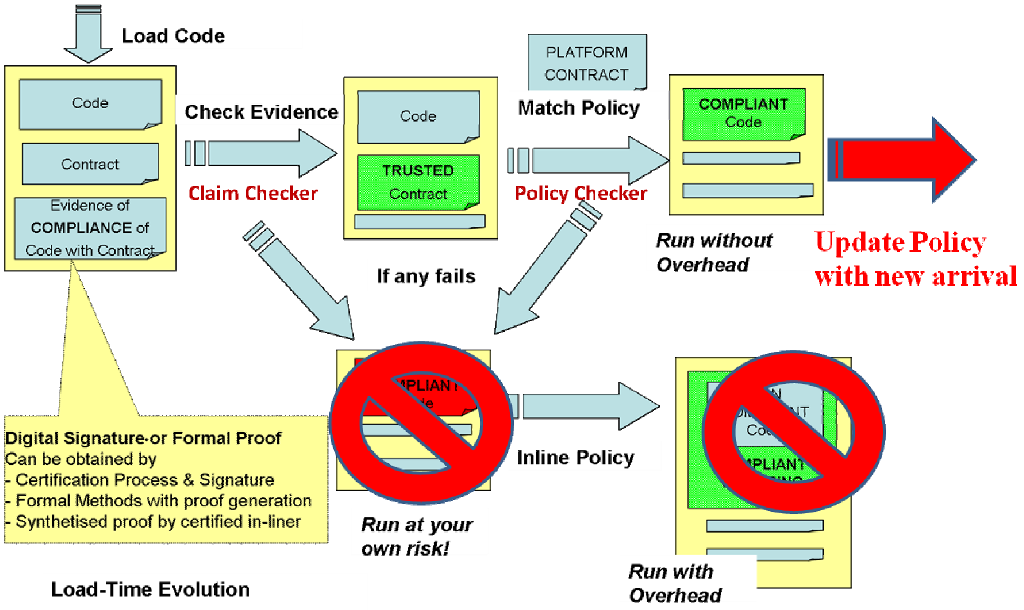


**Figure 9 - Security by Contract**

Our deliverable 6.4 shows the PolicyChecker component for incremental types of updates and discussed how the framework can be integrated with the Java Card system.  In the third year of the project, the feasibility of the Security-by-Contract approach was validated in the HOMES case study.Transitive control flow verification technique extend this framework with the aims of capturing illicit invocations of application methods in case of applications collusion with several solutions for dealing with decremental updates, each solution having a different trade-off between computation overhead and additional system memory required.

Global policy verification technique aims to detect forbidden sequences of methods calls at the system level, i.e. not necessarily only within one or two applications. This approach is inspired from proof-carrying-code (PCC) paradigm: static bytecode analysis conducted off-device generates proofs annotations embedded in the bytecode for easier on-device verification. We show_how to deal efficiently with decremental changes on-device with this model, mainly application removal because updates of the security policy (sets of forbidden sequences of method calls) have an impact on already loaded code stronger than expected and thus requires additional off-device computations but also on-device verifications.

Non-interference verification technique is also a PCC-like approach but whose goal is to detect illicit flows of data between applications clustered in domains. Even if the domain abstraction is strongly inspired from the GlobalPlatform environment, it is generic enough to be applied to any Java-based system.

All the aforementioned techniques support security policy updates. If a system security policy is updated the incremental on-device verification procedures will ensure that all the applications are compliant with new policy. Two approaches are sketched in case some of installed applications are not compliant with new policy. Either the policy update is rejected or the applications conflicting with new policy are made non-selectable.

Depending on the system requirements and stakeholders' needs it is possible to choose the most suitable verification technique. In the last year of the project, the transitive control flow verification technique was implemented and validated on the POPS case study. The two other techniques (the global policy model and the non-interference model) were not implemented, but their feasibility was validated by a detailed on-paper study for the POPS case study, as reported in deliverable D6.6.

We have discussed already the interplay between design and verification in D4.2. The main idea is to verify the same properties at the model level using WP4 techniques and at the code level using WP6 techniques to establish a coherency between (high-level) modeling of applications and their (low-level) implementations. We choose to focus on information protection related properties, and more precisely on the two control flow models and the non-interference model. For each of these models integration has been achieved by the establishment of new specific UMLsec stereotypes. For each WP6 model/WP4 stereotype, we rely on the same input, that is the security policy to be enforced. Furthermore, modifications on the model and the code are both dealt in incremental/decremental way to avoid full re-verification of the model and/or the code.

In addition to verify the same properties at different levels, a coherency report is established between UML models and the code analyzed. Actually, upon successful verification at the model level, some information is extracted from UML to permit additional verifications on the code and thus detect potential incoherencies between application(s) design and implementation.

In this setting testing and verification play a dual role on the information protection property of the POPS case--study. This requirement demands that assets (application data and specific services of security domains) of each stakeholder should be protected from unauthorized access. WP6 (on-device verification) provides techniques to ensure absence of illegal access to information data. WP7 is interested in the access to security domains services. It checks by testing the absence of possibility to misuse application installation and re-association processes, which grant direct access to security domains services.

In terms of integration we discussed threats for the information protection property and demonstrated that collaboration of two WPs provides protection against these threats. One of the main advantages of the collaboration is possibility for verification to rely on some assumptions about the installation process, because is testing guarantees the confidence in these assumptions. Another benefit is possibility to ensure absence of illegal transitive access to the security domains services, which can be verified by the techniques of WP6.

## Testing Evolving Systems

During the project, the objective of our work is to define a method and to produce a proof-of-concept implementation of our method of model-based testing techniques for evolving systems in regards of the state of the art (described in deliverable D7.1 section 2). This demonstrator provides a tool-set to ensure the preservation of security properties for long-life evolving systems using software testing. The main results are twofold:

- an approach for testing security properties, based on the use of test schema to which that formalize testing needs. Security properties are covered by a test generation process using a behavioural model of the SUT and associated test schemas.
- an approach for change management by means of model comparison. Our objective is to ensure the important criteria defined in the first year: test repository stability, traceability of changes, impact analysis and ability to automatically structure the test repository into evolution, regression and stagnation test suites.

This last year, we use the feedback of the evaluation realized by Gemalto on the POPS case study to improve demonstrator and the results obtained on case studies. This elements are resumed into deliverable D7.4  section 3.

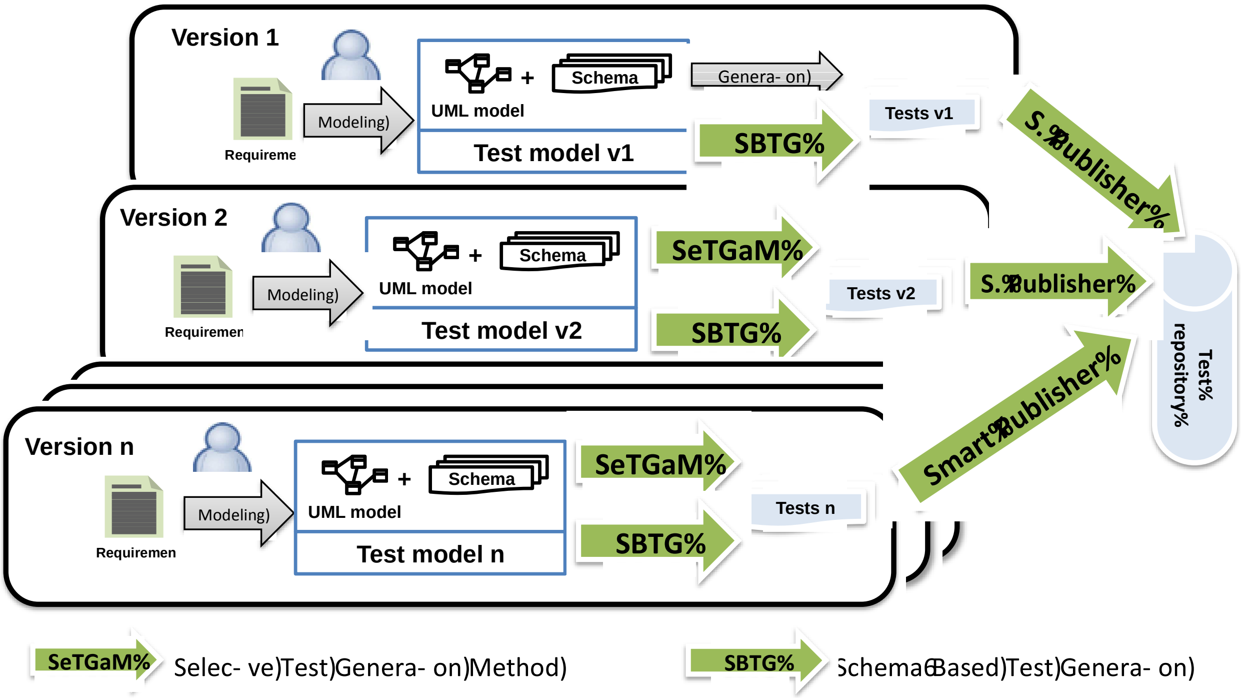


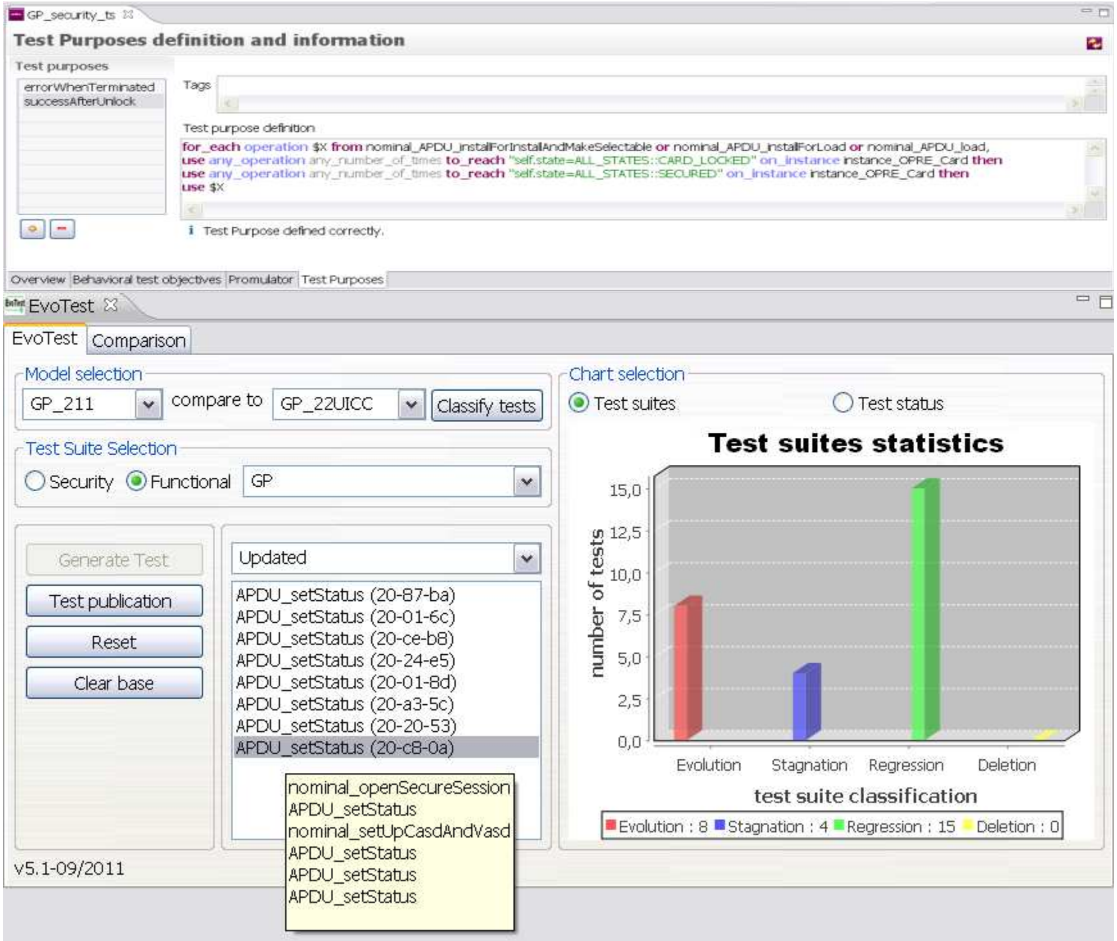
**Figure 10 - MBT Process for SecureChange project**

The demonstrator integrated the two aspects of the project: security and evolution. In Figure 10, we have in the green arrows the components developed during the SecureChange project:

- SBTG - Schema-Based Test generation: this component is composed by an editor with coloration enhanced and dedicated test generation module to take into account the test needs associated to security properties (see D7.3 section 3 and D7.4 section 3 for the last evolution). A strong adaptation of Smartesting test generation techniques has been developed to allow enough expressiveness for the test schema language in order to capture security test objectives.
- SeTGaM - Selective Test Generation Method: this component computes the dependencies of control and data (see D7.2 section 6) to select the impacted test suite by evolution. The dedicated algorithms based on dependencies analysis classified test suites (see D7.3 section 4). The last version integrated also model without statechars which more complex to compute dependencies (see D7.4 section 3).
- Smart Publisher: this component allows publishing the test suite into a test repository and take into account the history of the test suite between each version of the model as described into D7.3 section). It keeps tracks of previous tests status and minimizes

repository changes. For the project, we choice testlink (http://www.teamst.org/) as test repository.

We integrated all components into one eclipse plugin called EvoTest (see D7.3 section 5). In Figure 11, we have an example of all features proposed by plugin: schema editor, comparison between two model versions, test classification and publication.



errorWhenTerminated
successAfterUnlock

```
for_each operation $X from nominal_APDU_installForInstallAndMakeSelectable or nominal_APDU_installForLoad or nominal_APDU_load,
use any_operation any_number_of_times to_reach "self.state=ALL_STATES::CARD_LOCKED" on_instance instance_OPRE_Card then
use any_operation any_number_of_times to_reach "self.state=ALL_STATES::SECURED" on_instance instance_OPRE_Card then
use $X
```

Test Purpose defined correctly.

GP_211 compare to GP_22UICC

● Test suites ○ Test status

○ Security ● Functional GP

Updated

APDU_setStatus (20-87-ba)
APDU_setStatus (20-01-6c)
APDU_setStatus (20-ce-b8)
APDU_setStatus (20-24-e5)
APDU_setStatus (20-01-8d)
APDU_setStatus (20-a3-5c)
APDU_setStatus (20-20-53)
APDU_setStatus (20-c8-0a)

nominal_openSecureSession
APDU_setStatus
nominal_setUpCasdAndVasd
APDU_setStatus
APDU_setStatus
APDU_setStatus

🟥 Evolution : 8 🟦 Stagnation : 4 🟩 Regression : 15 🟨 Deletion : 0

**Figure 11 - EvoTest plugin**

## Scientific impacts Highlights

The Dissemination activities of the SecureChange project included a presentation of the project at the Project Track of the MODELS 2010 conference in Oslo, Norway to disseminate research questions and first research results to the scientific community. Additional presentation has been done at a number of EU events such as ICT, ServiceWave 2011. The project partners delivered roughly 50 additional presentations and published more than 100 papers addressing different parts of the project.

- 26 journals publications;
- 87 conferences publications;
- 4 book chapters and 2 reports.
- 12 collaborative papers co-authored by SecureChange researchers from at least two different partners
- 11 tutorials and 11 invited talks

As an example of the scientific highlights, the change patterns methodology and the results of the experimentation have been published to a special issue on model evolution of the Journal of System and Modeling (ISI impact factor 1.533), a IEEE Computer issue devoted to evolving critical systems has two articles by SecureChange researchers, the OpenArgue tool validation by SecureChange industry partners was invited to a special issue of the Requirements Engineering Journal.

Most project tools (the Move Tool, the SecMer tool (and the underlying engines EMF-IncQuery and OpenArgue), the CARISMA tool, etc. have been made available on the web and there is a significant interest in their usage. The Rinforzando Tool developed by Thales is now in the process of de-risking for direct adoption in production environment. The results on the EvoTest tool by SmartTesting have been ported to the production environment.

The SecureChange project collaborated with other FET projects by contributing to dedicated workshops and meetings coordinated by the EternalS coordination action including the organization of a workshop co-located with ICT 2011 in Budapest. Furthermore, the project organized several internal workshops and meetings to strengthen integration within the project. Industrial partners have identified promising and potentially usable results for exploitation.

Transferring knowledge to educational courses and academic research is an important channel to exploit the results of a research project. Project partners have developed 8 courses and additionally, 8 lectures where SecureChange results were integrated. In addition, there are 21 PhD theses which have been completed or close to completion – all of which are centred around research topic of SecureChange. For example the lectures " "Methodical Foundations of Software Engineering" (Methodische Grundlagen des Software Engineerings, 4+2 SWS) at TUD and the lectures "Security Engineering" (6ECTS) at UNITN are partly built upon SecureChange research results. The lectures will be continued/repeated in the next years.

Project partners have been very active in developing research prototype tools to provide feasibility study and practical validation of the scientific results. SecureChange proudly announces that as many as 8 tools have been developed completely within the scope of the project, while an additional pre-existing 9 tools have been continued to be developed. Several tools attracted significant interest from industrial partners and considered for future industrial exploitation.

## Potential Business Impacts

The SecureChange project will have a significant impact along two dimensions:

- In term of technology: the project has provided significant progress beyond current practices to a large panel of business domains and actors;
- In term of business impact: providing advanced engineering capabilities can have a great impact in several business domains, participating to their transformation. Their transformation will enable a more secure, eco-efficient society which in turn will contribute to a more economy efficient Europe.

In order to understand how the SecureChange project has the potential to reach its strategic impact it is useful to consider several numbers of the software production market:

- Software testing accounts for 50% of pre-release costs, and 70% of post-release costs.
- $200 billion are spent per year addressing software disruption.
- Non functional errors account for 50% of all software errors.
- 84% of software development projects are not completed on time.
- 58% of completed software development projects do not achieve the desired functionality.

These numbers are particularly daunting for the development of large scale critical distributed systems such as those where the industry partners of SecureChange are involved: air traffic management, airborne systems, naval systems, simulation, communications, C4I systems, security systems, e-passports, etc. In these domains the concept of system of systems (SoS) is spreading both in the military and civil domains. A SoS is a large scale global system with multiple, heterogeneous, distributed systems that interact and collaborate through networks at multiple levels and across multiple domains. Systems of systems can be briefly characterized by operational independence of the elements, managerial independence of the elements, evolutionary development, and high dynamicity of architectures, emergent behaviour, and geographic distribution. Here interoperability, security and dependability are key concerns.

Lifecycles are more and more incremental. Sometimes the architecture is not even fully defined at design time, but can evolve across design and operation stages (possibly during mission), depending on the operational context or, the available resources. Further, SoS have very long life cycles (20 years is a frequent duration). Large multi-disciplinary teams are involved over the lifecycle of these systems, in the definition, development or acquisition, verification, integration and validation, deployment, operations, support, disposal, maintenance and evolution, management, user training, etc.

In the SecureChange project we apply the proposed design flow to multiple use cases from several domains. A specific task for analyzing the impact was planned in order to evaluate and quantify the added value of the SecureChange approach during the implementation of the use cases. Based on the evaluation results, the following benefits are expected from the project:

| Benefit | Expected Result |
|---|---|
| Reduction in life cycle cost and cost of elimination of errors | The expected result is that more errors will be detected either already during design time or latest during integration. The number of errors occurring during operation will tend to be close to zero. |
| Ease SW development | SW design for combined safety and security issues 10% faster. |
| Consistent tool flow | Reduce tool flow breaks by 30%. |
| Reduction of development time | Reduction of development time by 20% through early error detection. |

An increasing number of domains require evolutionary systems. Building and updating these critical systems requires that the mainstream system architects, engineers and developers work hand-in-hand with security experts. Therefore the following industrial impacts are targeted:

- To efficiently manage the security of the overall system;
- To reduce the security flaws and their correction by a factor of two compared to current processes and systems.

The SecureChange project is addressing key industrial concerns of developing and managing complex systems. It is likely that the results will influence future research and development topics. Such kind of market interests large companies like Thales, EADS or Siemens.

Not being fully exhaustive, here after is a highlight of the business impacts that may be reached by SecureChange technology.

**Aeronautics**

The avionics market is expected to reach almost $83 billion for both forward and retrofit market segments to 2015 and $105 billion to 2020. Avionics solutions are facing the opening of the system to the open world (e.g. through internet connection to support maintenance tasks) creating de facto security flaws in the system. We assume SecureChange to have a direct impact on this market supporting in a secure way the necessary evolution of these systems.

In the aeronautic business, SESAR (Single European Sky ATM Research) is one of the most ambitious research and development projects launched by the European Community. SESAR has a strong concern with ATM security (ATM Target Concept, D3 WP1.1.3), which states:

> *"Security must be embedded in the SESAR ATM design process. It must become part of the ATM culture in a similar way to that which currently exists for safety. The concept of a closely integrated partnership of service users and providers is dependent on a level of trust between all the parties involved in the face of an aggressive evolving threat; the trust to be able to share information, to couple networks together, to protect airspace, to share staff and to implement joint security policies to protect the system from those who would disrupt it."*

With air transport being a facilitator of economic growth, SESAR is estimated to boost GDP by more € 400 billion over the period 2013-2030 (0.16% of combined EU GDP over that period), with 40% of the benefit being generated by SESAR directly and 60% from impact on suppliers and third parties. The economic benefits of implementing SESAR will be shared among European Union Member States and will contribute as much as two percentage points over and above yearly projected economic growth.

**Impact on automotive market**

The Automotive domain is another one that is faced with the new problematic of solving security aspects. Until recently, security aspects were not managed by IT technologies. On the contrary safety has always been of major importance in this domain and the increasing contribution of IT in the new generation of vehicles has lead for example to the elaboration of the new safety automotive standard ISO 26262. Automotive engineers are now conscious that IT could also lead to security violations (e.g. illegal modifications of calibrations that lead to regulation non conformities) and have the problematic to tackle conjunctly safety and security aspects.

The number of cars produced annually is estimated reaching over 90 Mio in 2017, with an average semiconductor content per car of more than USD 300 (Strategy Analytics). The most pronounced growth segments in automotive are powertrains and electronics. Key drivers are emission reductions, rising safety standards and greater comfort expectations. This will lead to increased electronics contents in cars, and in particular in the powertrain with alternative technologies such as for hybrid and electrical cars.

The automotive semiconductor market is expected to grow at an average rate of 9% over the next 5 years, with a more intimate interconnection of vehicle with the open world creating obvious security issues.

### Impact on Energy market

Cyber attacks have been an increasing source of concern in recent years but the threat was highlighted in 2010 by the first discovery of malicious code, called a worm, specifically formulated to target the systems that direct the inner operations of industrial plants. Since, in the US, the economic damage from a single wave of cyber attacks on critical infrastructure could exceed $700bn equivalent to the cumulative toll of 50 major hurricanes ripping into the nation simultaneously. SecureChange technology could be a major technology to reduce the risks of security threat on that market.

## Direct Business Impacts by Business Partners

### Deep Blue

Deep Blue will apply the SecureChange solutions proposed by technical WPs to its consultancy work as Human Factor, Safety and Validation expert in the Single European Sky ATM Research (SESAR) framework. In particular, the modelling approaches and tools that support requirements gathering (i.e. SeCMER) and risk assessment (i.e. CORAS) would be useful for future activities within the SESAR programme. Currently, Deep Blue is involved in a support activity for EURCONTROL that concerns the assessment of System Performances under Automation Degradation (SPAD). Within such activity, models are useful to assess alternative conditions of automation degradation. Deep Blue is also leading a Consortium, INNOVATE (INNOvation through Validation for Air Transportation in Europe), that will support the SESAR JU as associate partners in the area of *Modelling Support to Validation*. The Consortium will support the SESAR JU by providing the necessary support concerning modelling tailored for system validation. Within such activities, it is foreseen the use of modelling support in order to investigate and assess potential hazardous situations arising due to the introduction of new systems. The validation activities conducted within SecureChange have highlighted the potential of requirements modelling to support brainstorming activities about how new systems might affect critical features like security. Another outcome of the validation activities concerns the use of model-driven risk analysis (i.e. CORAS) to investigate the emergence of risk due to changes. Deep Blue will present the work carried out in collaboration with WP2 and WP5 to a wider audience of ATM stakeholders interested in Risk Assessment techniques coping with evolution and changes through dedicated meetings and ‘ad hoc’ presentations. Deep Blue would actively support the deployment of SecureChange's artefacts within the ATM domain.

### Gemalto

For Gemalto (and Trusted Labs) 2012 will be the year for the deployment of the solutions for the NFC market, i.e secure elements ( UICC is one them), trusted services manager (TSM) platform that will manages securely secure elements, services providers and Mobile operator and Validation authority (VA). This later, as Trusted Labs, is in charge of the validation of the services (application) before its loading on the UICC, on behalf the mobile operator or the service provider in case of payment service. Two options are available: off-card verification (with heavy environment constraints and lighter devices) or on-card verification (with light environment constraints and more code on the device). Although it is currently not clear with option will be preferred by the market (and at the end both could be available for different contexts) it is of strategic importance for Gemalto to have all the technologies ready to be delivered to its customers. For that, Gemalto intends to continue the investigation using the Securechange results and mainly the SxC technology and the off-card verification based on static analysis. So far, Gemalto and Trusted Labs is involved into several technical groups that are working on facilitating the deployment of services on the open product. This rely on the definition of a set of (security) rules that the application (service) has to ensure before being loaded on the open product. The SxC technology and the static analysis tools will be investigated for that context: Some of the rules could be implemented in the static analyser tool to check the application (if the source code is available). If the validation is successful, the applet is “stamped” and stored as a candidate in a database. The investigation for SxC technology will consist in checking if the rules that may depend on the target product, could be implemented as security policies, a part brought by the applet and a part stored on the product.

The socio-economic impact of the project results relies on the maturity of the solutions developped by the project. For example, the maturity of the on-card verification techniques are crucial for the end-user (card holder). Those techniques will provide the UICC with the ability to validate the application on the field and then will give the end- user the freedom to load any service he needs instantly.

For the off-card verification technique, the static analyser tool will give the opportunity to the service provider to check its application (service) against the rules imposed by the mobile operators. This will accelerate the business relationship between the services providers and the mobile operators. Finally, the testing techniques provided by the project will allow the card manufacturer to shorten the validation phase with respect to the changes of the card platforms. This will improve the time to market of this kind of products.

**Thales**

Thales views dissemination as an enabler for exploitation. The focus was put on internal dissemination, by using existing communication vectors but also by proactively seeking interest from businesses not traditionally concerned by security, especially since there is a strong trend to extend safety considerations to security ones. Examples of the communication vectors that were used are the “Journée de Palaiseau”, with 171 registered persons from the Thales group (France, UK, Canada, The Nederlands and Germany), and Networks of Excellence (NoE).

External dissemination was performed in opportunistic fashion. In 2011, Thales co-authored two publications with partners from the project: “Managing Changes with Legacy Security Engineering Processes” at ISI, and “Security and Change Engineering throughout the whole System Engineering Process” at Service Wave. Thales also contributed to the project brochure.

Thales' exploitation strategy can be summarised in three steps. The first step, at the start of the project, consisted in identifying the key partners and the promising technologies. As the project progressed, Thales moved to the second step, in which the previously identified technologies were assessed, internal "pilot projects" were identified, and an "Initial Gate" was prepared. Within Thales, an Initial Gate denotes the formal transfer of a technology from a research lab to operational units for a derisking phase. The third step, taking place after the end project, precisely consists of this derisking activity.

The most promising partner technologies that were identified for Thales' security engineering when SecureChange started are Si* for security requirement engineering, EMF-IncQuery for the incremental verification of structural constraints, UMLsec for early validation, and CORAS for risk assessment. That list evolved during the project.

During SecureChange, Thales' risk assessment prototype Rinforzando evolved from a limited standalone tool (TRL 1) to a finalised tool (TRL 3) closely integrated with a system engineering tool called SOA Modelling Suite. A large number of meetings (tens) with operational units within Thales were organised, with a strong acceleration in year 3. In early 2012, a decision was taken to proceed with an Initial Gate.

Thales plans its post-project exploitation through two roads:

- Internal exploitation;
- Exploitation through future research and development projects.

The key “selling technologies” for the “internal exploitation” are those project technologies that have been assessed by Thales has having the highest Technology Readiness Level (TRL) for industrial exploitation, namely Rinforzando, both in its standalone and integrated configurations, and the related security engineering process. The key “selling technologies” for the “exploitation through future research and development projects” are those project technologies that have been assessed by Thales has having low TRLs, but which seem promising, namely EMF-

IncQuery, and possibly CARiSMA and SecMER.

Related to the "internal exploitation" of Rinforzando and the related security engineering process, the action plan has two axes:

- Extend from the current "pilot-project" scope to a "domain-wide" scope;
- Extend from the "domain-wide" scope to a corporate scope.

The current exploitation pilot-project is the Galileo programme. The person responsible for the risk assessment on this Galileo programme (namely Raphael Vignon Davillier) is also responsible for the definition of the methods and tools for the "domain". Thus, our first goal beyond the effective application on the Galileo programme is the adoption of our technology to the "domain". The next step, in case of successful adoption, will be to propose the methods and tools at the corporate level for adoption within the entire Thales group, through the Engineering Process and Methods (EPM) corporate service.

For the exploitation of the other technologies through future research and development projects, no proposal has yet been submitted, but Thales will remain mindful of the technology evolutions in order to step in when industrial involvement is most useful. Considering EMF-IncQuery, this could be effective within a very short time.

**Smartesting**

The new concepts emerging from SecureChange are very promising to face the challenge of change management and testing security properties in the Model-based testing process. Smartesting, as a model-based testing solution provider, will concentrate its exploitation efforts mainly on WP7 results:

- In a short term timeline (1 year), we plan to integrate the results on change analysis between two versions of the test generation model in our core product; This will help to accelerate test generation time (reducing regeneration) and also to reinforce the stability of the generated test repository (which is often asked by QA team);
- In a midterm timeline (2 years) , we plan to develop a new offer targeting model-based dynamic application security testing both for security functions testing and for vulnerability testing. The results of the project regarding security test generation make up a first step that open for us a new direction of investigation. We want to continue to investigate several directions before defining a model-based testing generation solution to be pushed on the market:
    - Investigating the used of the Test Purpose language for vulnerability testing (for the moment, in the POPS case study, we addressed mostly testing of security function)
    - Continue to investigate the relation between risk analysis (using SINTEF CORAS method for example) and the design of the security test purposes.
    - Investigating the evolution of the functional model in order to capture stimuli of possible attacks and dedicated observation to generate accurate vulnerability tests.

## Spin-Off Creation

The promising results of the SecureChange integrated process have contributed to the foundation of the spin-off company QE LaB Business Services GmbH (http://www.qe-lab.com/)

in January 2012. QE LaB Business Services GmbH is supported by the Center for Academic Spin-Offs Tyrol (http://www.cast-tyrol.com/) , and the University of Innsbruck is shareholder of this company through its holding.

## Potential Impacts at Large

Although several international standards and state-of-the-art security and risk assessment frameworks stress change management as an important part of risk management, little or no specific support exist in terms of guidelines, modeling techniques, tool support, etc.

The results from SecureChange advance state-of-the art by delivering a set of artifacts that combine into a model-based approach to risk assessment that offers support for systematically handling change throughout the whole risk management process. For businesses, enterprises and organizations relying of security-critical software-based systems in the information society of today, which is highly heterogeneous and rapidly evolving, the SecureChange assessment framework offers an approach to updating the security and risk picture when triggered by change requirements without conducting a full analysis from scratch every time.

The potential socio-economic impact includes the potential for industry and organizations to reduce costs by more quickly adapting to changing environments, and by increasing their effectiveness and efficiency in managing security risks.

Citizens, businesses and governments will also benefit from trustworthy security critical ICT systems and applications in which evolving and emerging security risks are continuously managed and mitigated.

# APPENDIX CASE STUDIES

## ATM Case Study: Change Requirements

The ATM case study is concerned with **changes in the operational processes of managing air traffic in Terminal Areas**. Arrival management is a very complex process, involving different actors. Airport actors are private organizations and public authorities with different roles, responsibilities and needs. The subsequent introduction of new tools, i.e., the Queue Managers, and the introduction of a new ATM network for the sharing and management of information, affects the ATM system as a whole at a **process** and **organizational** level.

### Process Level Change

ATM procedures need to be updated in order to accommodate the introduction of the AMAN (**A**rrival **MAN**ager). The AMAN is an aircraft arrival sequencing tool helping to manage and better organise the air traffic flow in the approach phase. It 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, catering services, airlines, security and health authorities, etc.) in order to organise the ground services efficiently.

The introduction of the AMAN requires new operational procedures and functions (as described in the deliverable D1.1). Such new procedures and functions are supported by a new information management system for the whole ATM, an IP based data transport network that will replace the current point to point communication systems with a ground/ground data sharing network which connects all the principal actors involved in the Airports Management and the Area Control Centers.

**Goal:** The resulting ATM system (with the AMAN and the communication network introduction) needs to comply with suitable security properties, which prevent from corruption, accidental or intentional loss of data and guarantee the integrity and confidentiality of the aircraft sensible data against malicious attacks or intrusions.

### Organizational Level Change

The introduction of the AMAN affects Controller Working Positions (CWPs) as well as the Area Control Center (ACC) environment as a whole. The main foreseen changes (as described in the deliverable D1.1) in the ACC from an operational and organizational point of view are the automation of tasks (i.e. the usage of the AMAN for the computation of the Arrival Sequence) that in advance were carried out by Air Traffic Controllers (ATCOs), a major involvement of the ATCOs of the upstream Sectors in the management of the inbound traffic.

These changes will also require the redefinition of the Coordinator of the Arrival Sequence Role, who will be responsible for monitoring and modifying the sequences generated by the AMAN, and providing information and updates to the Sectors.

**Goal:** The AMAN's interfaces that provide access to different roles and authorizations need to make information available only to authorized personnel or trusted systems.

### Security Properties

The following security properties need to be guaranteed at the process and organizational level and will be the focus of the technical WPs.

**Information Access**. Authorized actors (or systems) must have access to confidential information regarding queue management in the terminal area. Access to information needs to comply with specific role-based access control rules drawn from the operational requirements. **Information Protection.** Unauthorized actors (or systems) are not allowed to access confidential queue management information. **Information Provision.** The provisioning of information regarding queue management sensitive data by specific actors (or systems) must be guaranteed 24 hours a day, 7 days a week, taking into account the kind of data shared, their confidentiality level and the different actors involved. **Information Need.** Confidential queue management information can be accessed by authorized actors (or systems) only when the information is necessary for operational purposes, which may vary even in real time, due to particular conditions (bad weather, emergency status, etc.)

## HOME Case Study: Change Requirements

HOMES is focused on digital home networks where some sensible changes take place from the point of view of the security. We consider some changes, from the large set of changes that anyone may identify in this context, very related to configuration and deployment. Our target is the home gateway as a critical point in the home network.

### Core Security Module Update

Home Gateway has some security modules implementing NAC functional components like the PEP. NAC technology and its functional elements are properly described in the deliverable D1.1.. During the lifecycle of the whole system some component updates shall be required for various reasons (better performance, bug fixes, etc.). Updating one of these core security modules in the home gateway is a critical operation and a relevant change. Any attack or failure in this process may be extremely harmful. A possible update on the core security modules could be the extension of information for the security assessment (more information in deliverable D1.1). In this case, the home gateway needs to be updated so that the new security status information is understood and assessed correctly.

**Goal:** Show that the security properties detailed below are still preserved after an update of a security module.

### Bundle Lifecycle operations

A Home Gateway is also a service platform for the home. Customers can install new home services, upgrade or delete existing ones. This type of change is similar to the previous one but here services do not usually implement security functionality. The bundles installed on the home gateway are used for higher level applications. The services may come from third parties and therefore some similar control over this software must exist. Trust relationships among the customer, the service provider, and the third parties may evolve over time. However in some cases security bundles could be deployed (provided by the operator)

**Goal:** Bundles have to be managed (update, addition, removal) in compliance with the trust relationships and assuring system consistency, i.e. the security properties need to be preserved despite these changes.

### Security Properties

**Secure extensibility**. The home gateway can be extended at run time with additional general software (e.g. bundles) coming from third parties in many cases. Such extensions should be verified to be secure in the sense that they do not introduce unauthorized information leaks or the possibility of denial of service **Policy enforcement.** The Policy Decision Point (PDP) is located in the security domain of the operator. The Policy Enforcement Point (PEP) is a core security module installed on the home gateway. The PEP always enforces policy decisions forwarded by the PDP so that only allowed actions can be carried out.

**Resilience to trust changes**. The system shall be able to accommodate a change in the trust relationships (among service provider, customers, 3rd parties) with a minimal impact on the software architecture.

**Security expandability**. System security can be enhanced by taking advantage of the home gateway extension ability (mentioned in the Secure Extensibility property) through the deployment of new security services (e.g., deployment of a non-repudiation service bundle to ensure that neither service provider nor customer can later deny having sent/received a purchased service). The infrastructure shall be able to efficiently enforce such new requirements with a minimal impact on it.

## POPS Case Study: Change Requirements

An USIM card has been certified w.r.t. Common Criteria security certification V3.1. This means that the embedded software on this device ensures a set of properties related to (at least) **confidentiality**, **integrity** and **availability** of its assets (but also non-repudiation, authentication, non-by-passability, etc).

But this “system” during its life cycle will evaluate. The Common Criteria impose that any change that occurs will lead to a re-certification of the card. As the evaluation process is expensive in term of cost and delay, we investigate means, that might be provided by the project, to speed up the re-certification of the card. The means are any kind of artefact that could be used for the evaluation: model, proof, test suites, etc. The objective is then to demonstrate this UICC card ensures those security properties after two realistic scenarios of changes, detailed below.

### Specification evolution

An UICC card embeds a component called the card manager, implemented according to GlobalPlatform specifications v2.1. This card component has been extensively verified and tested. The GlobalPlatform specification have been enhanced and extended and v2.2 has been issued. The card manager software component has been updated and extended against this new version. For simplicity reason, we restrict the 2.2 scope to the UICC configuration.

**Goal:** prove/demonstrate/test that the security properties are still preserved. For that we will concentrate on specific properties detailed below.

### Software update

The certified UICC card is deployed in the field. The mobile operator, owner of the card, has a new partner, a bank. He loads a new security *domain (a Java Card application)* on the UICC (card) using an OTA mechanism. This bank will have the delegated management privilege from the Mobile Network Operator to manage its applications in a **confidential** way. In particular, the bank will use its security domain to load an e-purse on the card.

**Goal**: prove/demonstrate/test that the new application preserves (do not break) the consistency of the existing and implemented security policies. Again the specific properties are detailed below.

### Security properties

**Denial of service**: Any application on the card do not generate a denial—of--service. This means that some robustness properties must be verified by the applets, such as no runtime exception, no infinite loop. Also the memory consumption must be bounded for the durability of the EEPROM and the Flash. For example, bounding the call-stack or detecting loop that updates the persistent memory.

**Life-cycle consistency**: Any command received by the card must respect the card and applet lifecycle. Its means that any command received in a state s leads to a state s’ and the resulting transition from s to s’ is correct w.r.t. the specifications.

**Information protection**: The applications on the card must be “isolated” (segregation), that means no illegal access to the data from one application to another. For that several security policies are described and assumed to be implemented on the card, like the JavaCard firewall (access control implemented by the virtual machine) or the security domains of GP. Therefore, some properties must be verified, when an applet is added on the card, like the consistency of the security domain hierarchy, the non-violation of the information flow policy implemented on the card, etc.

**Secure communication:** The Secure Channel protocol provides a secure communication between a card and the off-card entity during an application session. It means that the protocol must ensure entity authentication, an entity is an off-card one as the issuer (terminal) or an on-card entity. Each entity proves its authenticity to the other entity. The protocol must ensure also integrity and confidentiality of the transmitted data

## Glossary

| Acronyms | Definition |
|---|---|
| ACC | Area Control Center |
| AID | Application identifier |
| AMAN | Arrival MANager |
| APDU | Application Protocol Data Unit |
| ATC | Air Traffic Control |
| ATCO | Air Traffic COntroller |
| ATM | Air Traffic Management |
| CWP | Controller Working Position |
| DHCP | Dynamic Host Client Protocol |
| DMAN | Departure MANager |
| EMV | Europa MasterCard Visa |
| FTTP | Fiber To The Premises |
| ISD | Issuer Security Domain |
| 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 |
| SCP | Secure Channel Protocol |
| SIM | Subscriber Identity Module |
| TMA | TerMinal Area |
| USIM | Universal Subscriber Identity Module |
| VPN | Virtual Private Network |
| WIMAX | Worldwide Interoperability for Microwave Access |

## The Project Consortium

The consortium is formed by an ideal blend of research institutions, industry and small, research-oriented enterprises:

- Università degli Studi di Trento (UNITN), IT
- Budapest University of Technology and Economics (BME), HU
- Gemalto (GTO) FR
- Institut national de Recherche en Informatique et en Automatique (INR), FR
- Katholieke Universiteit Leuven (KUL), BE
- Smartesting (SMA), FR
- Open University (OU), UK
- Stiftelsen for industiell og teknisk forskning ved Norges Tekniske Hogskole – SINTEF (SIN), NO
- Thales (THA), FR
- Telefonica Investigacion y Desarrollo s.a.u. (TID), ES
- University of Innsbruck (UIB), AT
- Deep Blue s.r.l. (DBL), IT
- Technische Universitat Dortmund (TUD), DE

## Further Information

Further information can be obtained by contacting the project coordinator

Prof. Dr-Eng. Fabio Massacci
DISI - Universita' di Trento
via Sommarive 14 - 38123 Trento – Italy
email: Fabio.Massacci@unitn.it
Tel: +39-0461-282086 -- Fax: +39-0461-283987

Or by visiting the project web site: **www.securechange.eu**