1 Publishable Summary

1.1 Description of Project Context and Objectives

1.1.1 Context

Resource Constrained Embedded Systems (RCES) are today integrated in increasingly more sophisticated applications. They require more security and dependability and they require more trust. This complexity is due to integration and functional considerations. First RCES are integrated into systems involving complex supply chains with many different business stakeholders focusing on different subsystems (e.g. processors, ASIC, operating systems, middleware, application components) and different integration levels. Secondly, RCES are often used in multipurpose applications involving multiple and possibly independent applications. There is a switch from simpler value chains/ecosystems to fairly complex ones, where it is needed to define specific measures to protect individual computing assets.

Resource Constrained Embedded systems (RCES) are characterised as follows:

- they can be found everywhere, in different application sectors (automotive, aerospace, home, etc.), in different form factors (standalone systems, peripheral subsystems to a main computing system, etc.), in many different devices (sensors, automotive electronic control units, intelligent switches, home appliances e.g. washing machine drum control, meters, etc.)
- Computing resources, e.g. memory, tasks and buffers, are statically determined. For instance, the entities managed by the underlying operating systems are typically predetermined. Another example is the OSEK-VDX RTOS (www.osek-vdx.org) standard which defines tasks, resources, alarms entities. These entities are identified statically at design time (e.g. 3 tasks, 4 resources, 2 alarms would make up a given system).
- Most RCES are high integrity systems, or systems which must meet assurance requirements. Depending on application requirements, different levels of assurance can be involved from the most stringent which involve certification (e.g. DO178, IEC-61508 for safety-relevant embedded systems development), to lighter levels of assurance (e.g. industry practices). As a matter of fact, many RCES involve very significant software development costs and therefore use advanced engineering disciplines (automatic code generation, model-driven developments).

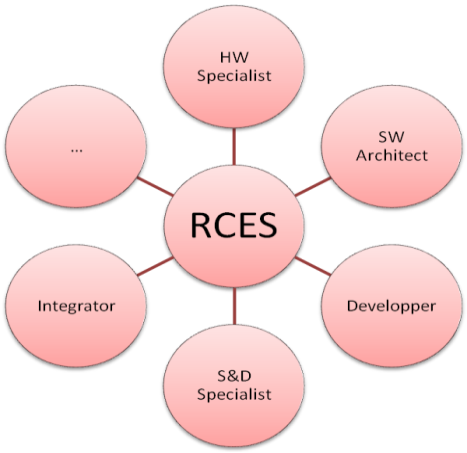

Figure 1 : RCES as a synergy of business.

1.1.2 Objectives

The goal of TERESA is to define, demonstrate and validate an engineering discipline for trust that is adapted to resource constrained embedded systems. Trust is defined as the degree with which security and dependability requirements are met.

The TERESA approach is to use a model-based repository of Security and Dependability (S&D) patterns:

- Application sector trust models are defined as profiles (e.g. UML, SysML profiles), based on a common trust meta-model.
- Security and dependability platform independent patterns are identified and defined for each application sector (some patterns could be used by several application sectors).

- Formal properties on security and dependability are defined and validated for patterns belonging to application sectors requiring that level of assurance.
- Platform dependent implementations of the patterns are guided with very precise requirements.

TERESA has the following objectives:

- Objective O1: Support for S&D pattern reuse in a railways sector use case.
- Objective O2: Support for S&D pattern reuse in a metering use case.
- Objective O3: Ensuring the genericity of approach by providing guidelines for the specification of sector specific RCES for trusted computing engineering. Software process engineers in a given sector, e.g. automotive, use the guidelines to define a trusted computing engineering process that is integrated to the software engineering process used in the RCES sector.

1.2 Results

This section explains the results achieved by the project. The results are presented as follows:

- The TERESA theoretical contribution to engineering (methodology, language, formal validation)
- Demonstration in specific application domains (case studies, demonstrator and genericity of approach)
- Supporting toolset (tools for the approach)
- Example patterns

1.2.1 A PBSE Theoretical Foundation

A Pattern Based System Engineering (PBSE) methodology based on a repository was specified. This engineering methodology fully takes into account the need for separation of roles by defining three distinct processes, the pattern modelling process, the repository specification process, and the pattern integration process. A set of languages was specified for the repository structure and content. The latter consists of specifications of patterns, S&D properties and processes.

The use of formal validation in the process was successfully carried out. The formal validation is based on the SeMF framework. A validation methodology was defined to ensure that S&D patterns provide the expected properties with respect to their interfaces. The methodology was applied on a number of example patterns. Guidelines were defined on how to help integrating any pattern, and a framework to help validating the instantiation of the pattern was defined.

1.2.2 Application Domain Examples

Requirements for the project (engineering viewpoint, process viewpoint and repository viewpoint) were defined by studying use cases in four application domains corresponding to a partner's expertise, and **S&D patterns** of interest were identified.

Two case studies were studied in depth: Safe4Rail and a Secure Gateway for smart metering:

- The railways domain demonstrator was developed to validate the TERESA approach from an already existing dependable system where security is taken into account. This domain involves strong constraints with well-defined and mature engineering processes.
- The metering domain demonstrator was developed to validate the TERESA approach in domains where new engineering approaches need to be established. This domain is increasingly prominent, but without the long history of the Railway sector.

Implementations were done in two versions: in one version a full traditional implementation was carried out or reused. In the other version the same implementation was carried out using S&D patterns in order to assess the repository, tools and process. These demonstrators showed that **TERESA O1 and O2 objectives were reached**.

A study was carried out in the automotive domain to assess the ease with which the TERESA approach could be adapted to the toolset provided by the automotive tool provider company ETAS. This study validated **the genericity of TERESA (O3 objective).**

An evaluation study was carried out, consisting of presentations to stakeholders and of analysis of Key Performance Indicators. They confirmed the potential of the TERESA approach, and the rising interest for MDE and pattern based engineering.

1.2.3 MDE Tool chain

TERESA has produced a model driven engineering (MDE) tool chain. The following tools targeted to the S&D developer, i.e. the engineer who creates S&D patterns, were developed:

- **Gaya**, a repository based on MDE technology was developed. This repository allows for the storage of engineering and process knowledge associated with S&D patterns.
- **Arabion,** a tool for the creation and edition of S&D patterns. Such patterns must be stored in such a way that they can reused later, enhanced and modified.
- **Tiqueo,** a tool for the creation and edition of S&D properties and constraints. Focus are on the non-functional requirements that are associated with S&D patterns

The following tools targeted to the S&D integrator, i.e. the engineer who uses S&D patterns, were developed:

- **Naravas,** a tool for the creation and edition of engineering processes. Engineering processes are often domain specific (i.e. they could be based on different standards) and stakeholder specific (i.e. specific corporate processes)
- **Access tools for Safe4Rail.** The tool transforms the Gaya representation of S&D patterns into a representation that is consistent with the Safe4Rail set of tools (mostly Rhapsody based) and the Safe4Rail process.
- **Access tools for the secure gateway for smart metering.** The tool transforms the Gaya representation of S&D patterns into a representation that is consistent with the associated set of tools (mostly Rhapsody based) and the need to support a process compatible with smart gateway common criteria protection profiles.

The following contributions on engineering processes were made:

- **an example railways process** based on the Safe4Rail use case
- **an example metering process** based on the Secure gateway protection profile
- **a study of feasibility in an automotive process** based on the ETAS toolset
- **guidelines on how to use common engineering metamodels.**

1.2.4 Examples Patterns

In order to demonstrate and validate the TERESA approach for S&D reuse-based on patterns, the project has contributed to the specification of pattern information (or artefacts) to be stored in the TERESA repository. The flexibility of the repository is based on different levels of representations which can be stored in the repository, depending on (1) the application domain supported, and (2) the engineering process phase supported. Overall, **59 S&D patterns were developed**, consisting of 20 system level patterns, 25 architecture level patterns and 14 design level patterns.

Forty-one patterns were developed and used while working on the railways and metrology use cases. The 18 remaining patterns were added to the repository for generic or training purposes.

The table below presents the 41 one used patterns. A pattern is based on the name of the solution and the phase in the process where it is available. Some have only a domain-specific version (mostly for metrology), but others are also described in a domain independent version.

Pattern (composed of a name and a development state)		Domain Supported	Security / Dependability
SafetyCommLayer	System Concept	Domain Independent	Dependability
	System Architecture		
	Software Architecture		
	Module Detailed Design		
	System Concept	Railway Domain	
	System Architecture		
	Software Architecture		
	Module Detailed Design		
Hypervisor	System Concept	Domain Independent	Dependability
	System Architecture	Railway Domain	
Majority Voter	System Concept	Domain Independent	Dependability
	System Architecture		

	Software Architecture	Railway Domain	
	Module Detailed Design		
Reciprocal Monitoring	Software Architecture	Domain Independent	Dependability
	Software Architecture	Railway Domain	
	Module Detailed Design		
TMR	System Concept	Domain Independent	Dependability
	System Architecture		
	System Concept	Railway Domain	
	System Architecture		
Security Comm Layer	System Concept	Domain Independent	Security
	System Architecture		
	System Architecture	Railway Domain	
	Software Architecture		
	Module Detailed Design		
Watchdog	System Architecture	Domain Independent	Dependability
	Software Architecture		
	System Architecture	Railway Domain	
Data Agreement	System Concept	Railway Domain	Dependability
	System Architecture		
	Software Architecture		
	Module Detailed Design		
Secure Remote Readout	Detailed Design	Metrology Domain	Security
	Implementation		
Wakeup Service	Detailed Design	Metrology Domain	Security
Secure Communication	Detailed Design	Metrology Domain	Security
Secure Logger	Detailed Design	Metrology Domain	Security
Key Manager	Detailed Design	Metrology Domain	Security
RNG Test	Unit Test	Metrology Domain	Security
Smart Meter Gateway Skeleton	Architecture Design	Metrology Domain	Security

1.3 For More Information

For more information, contact the project co-ordinator or the project deputy co-ordinator:

Project Co-ordinator :	**Deputy Co-ordinator :**
Antonio Kung, TRIALOG 25 rue du Général Foy 75008 Paris, France +33 (0) 1 44 70 61 00 antonio.kung@trialog.com	Cyril Grepet, TRIALOG 25 rue du Général Foy 75008 Paris, France +33 (0) 1 44 70 61 00 cyril.grepet@trialog.com

You can also visit the TERESA website: www.teresa-project.com