RE-TRUST "Kick-off" Workshop
on
"Run-time Software Integrity and Authenticity"

18-19 - September - 2006

Trento, Italy

Project Overview

Prof. Yoram Ofek

Computing/Networking Convergence

- Exponential growth in **computing/networking**
- Leading to unifying: **computing/networking**
- **All machines are interconnected**
- **Ensuring that all machines are TRUSTED is critical [operating as specified]**
- **Avoiding manipulation of programs/protocols**
  - **STEALING** content and information
  - **DENIAL** of service – TCP example
  - **FAIR** on-line bidding/trading/gaming
  - … … …

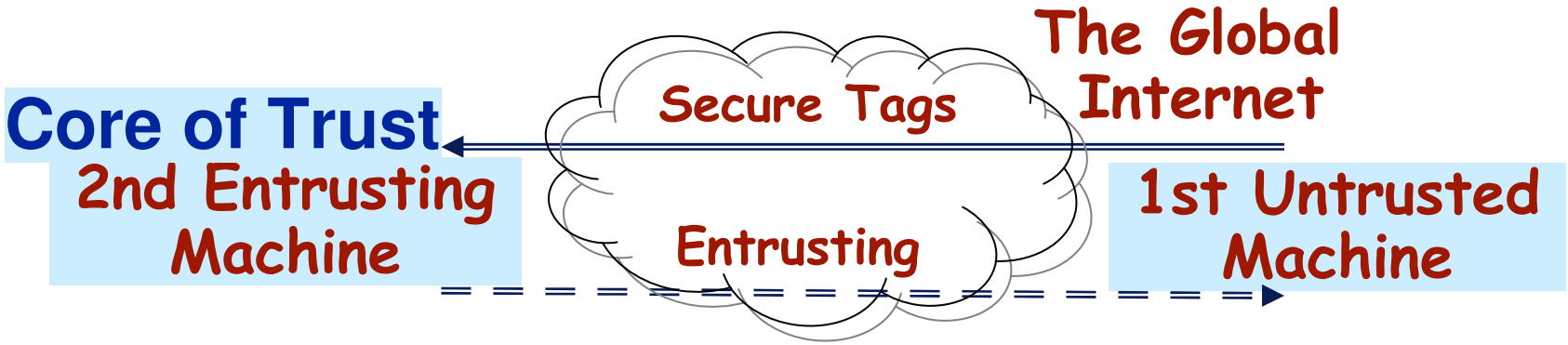
Remote Entrusting: Design Objective

How a remote code (SW application) on a
1st untrusted machine
can be entrusted by a
2nd entrusting machine?"
[albeit running inside an untrusted environment]

- I.e., Distribution of trust
  or entrusting

Functional Description: Remote Entrusting

- 1st Untrusted machine emanates Secure Tags from a code/software during execution

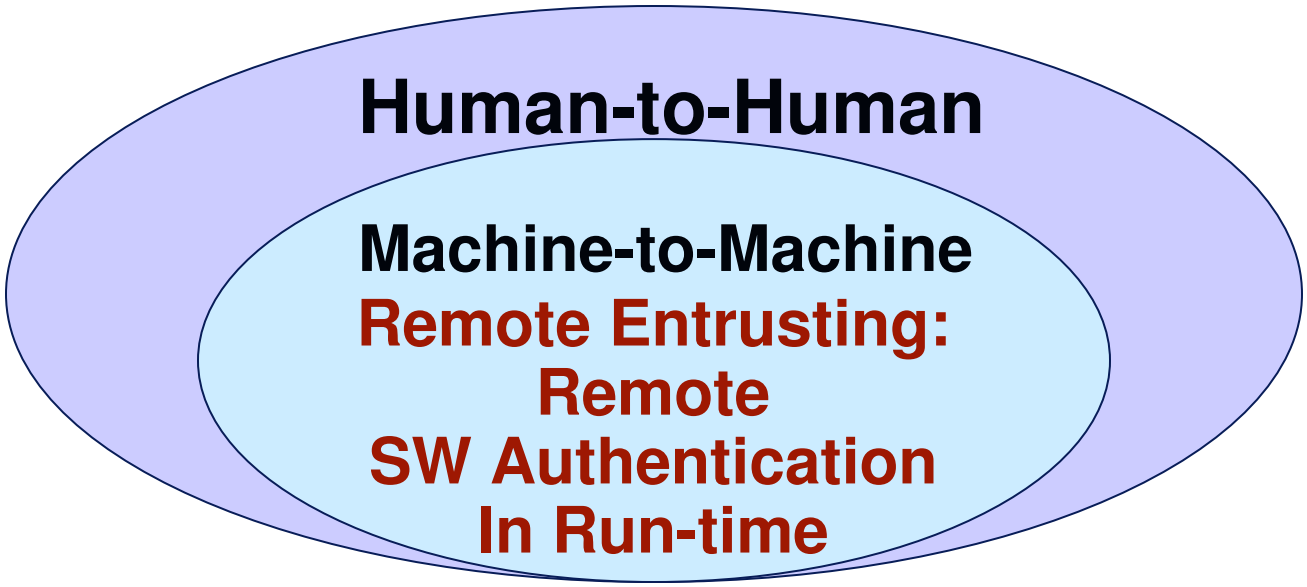


- 2nd Entrusting Machine is ENTRUSTING the 1st Untrusted machine by verifying the Secure Tags

Definition of Trust for Remote Entrusting

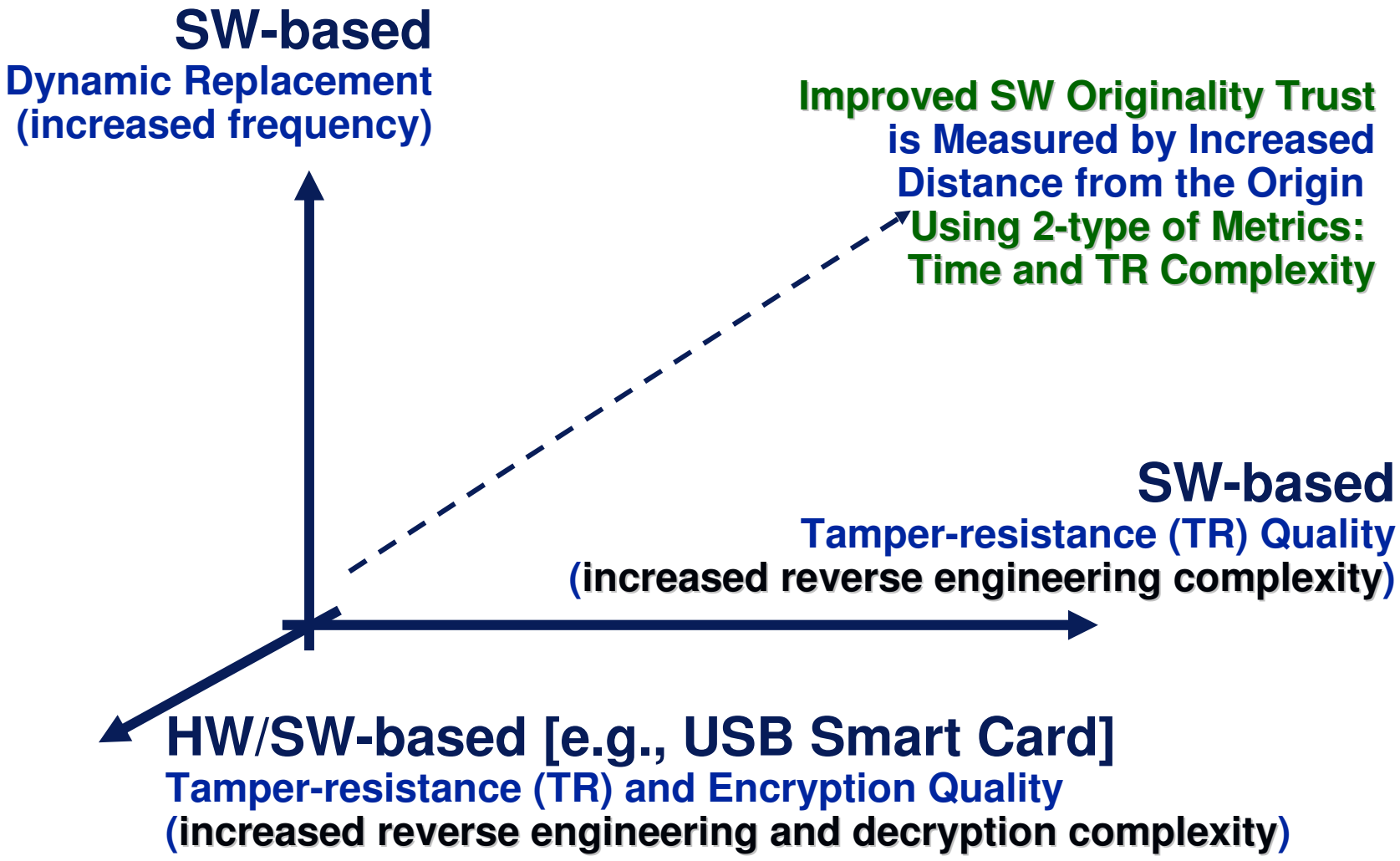
A software (code/protocol) is deemed authentic/trusted if and only if its functionality has not been altered/tampered by an untrusted/unauthorized entity prior to or during execution

Scope of Trust in RE-TRUST

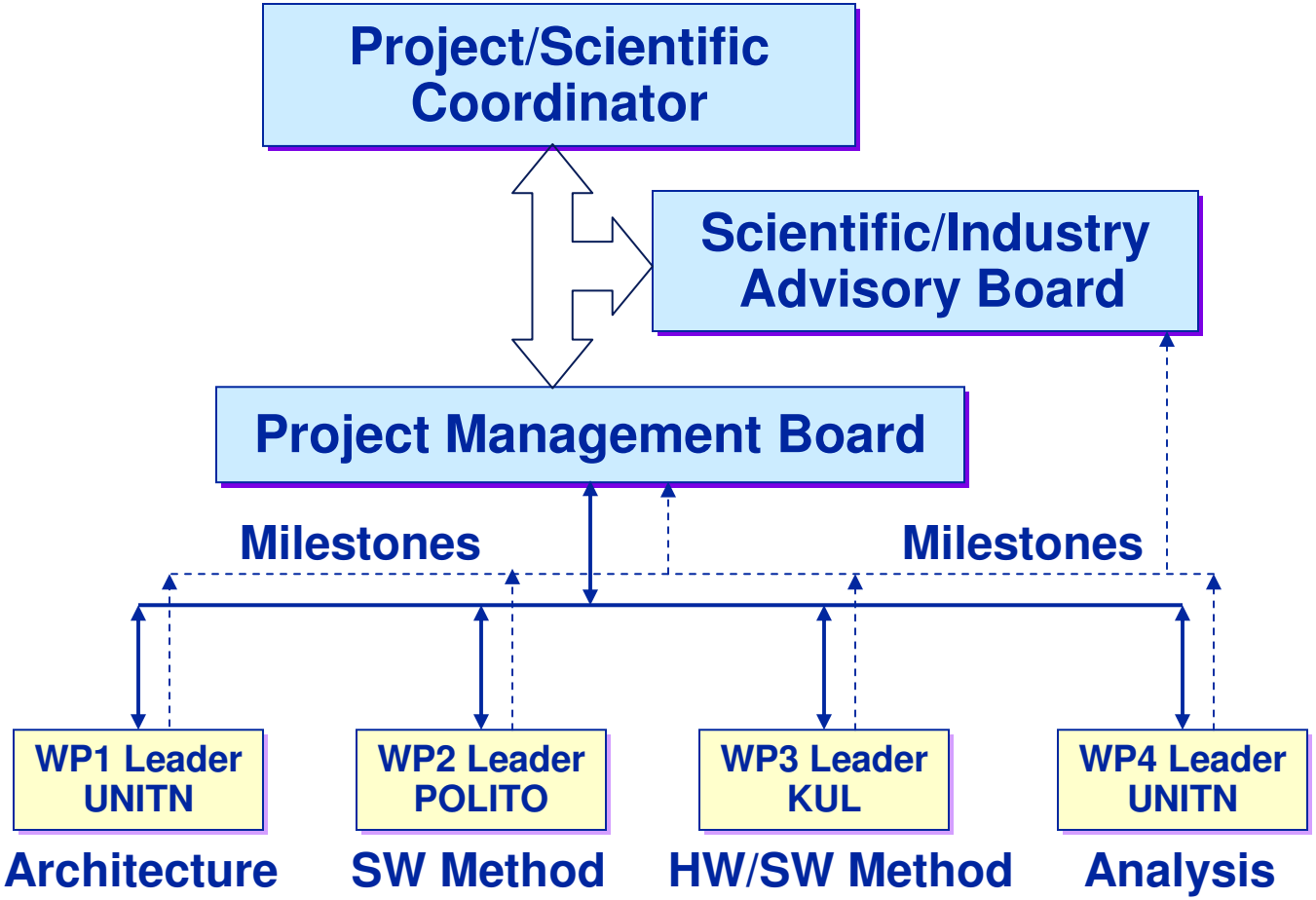


- Trust necessary condition: some sort of "identity"
  - **Signatures/attestations/Authentications of SW & HW in run-time**
  - { Avoidance of the **"man-at-the-end" attack [i.e., the untrusted user]** }

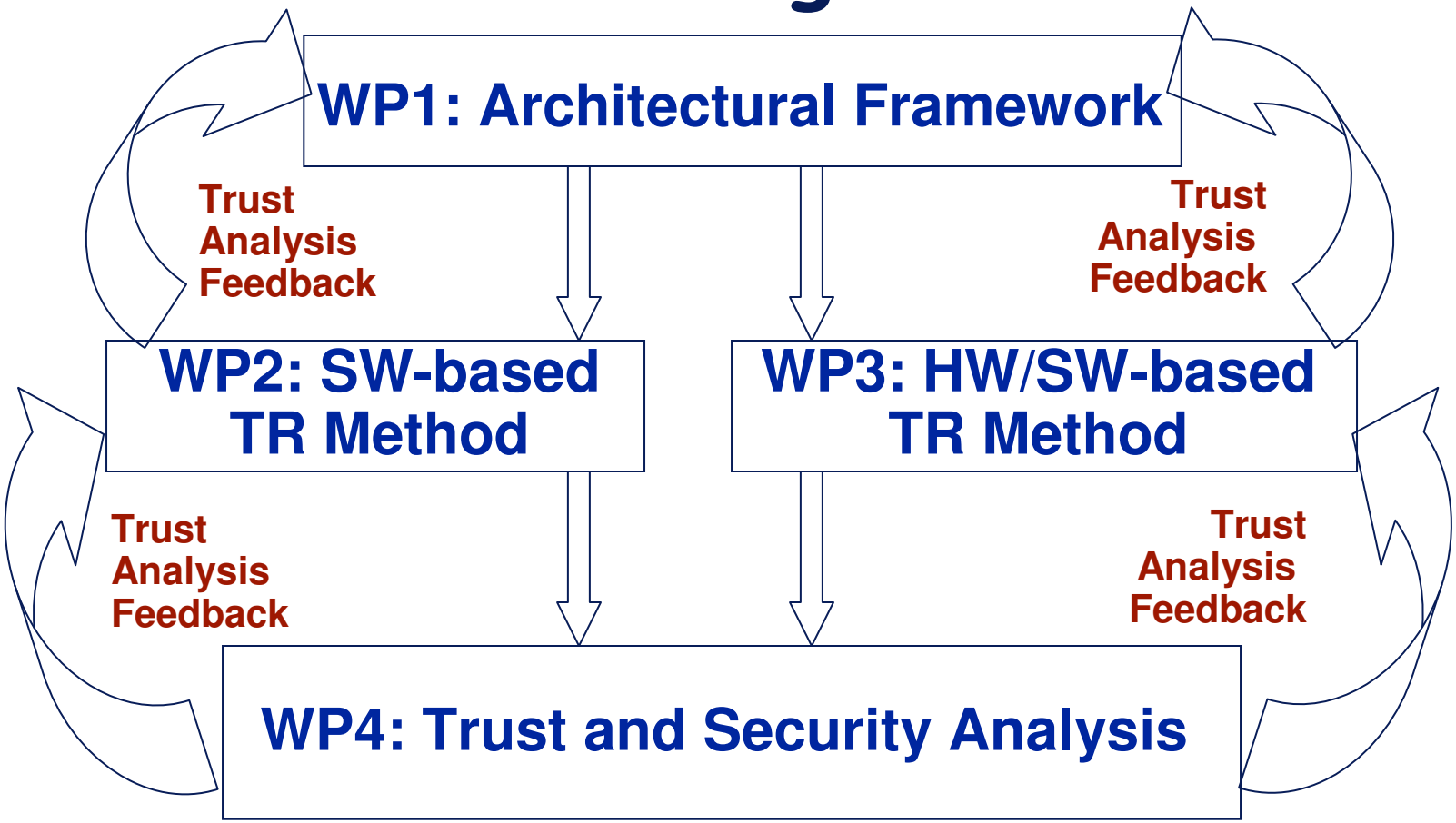
Quality of Remote Entrusting SW and HW/SW Methods



Project Structure



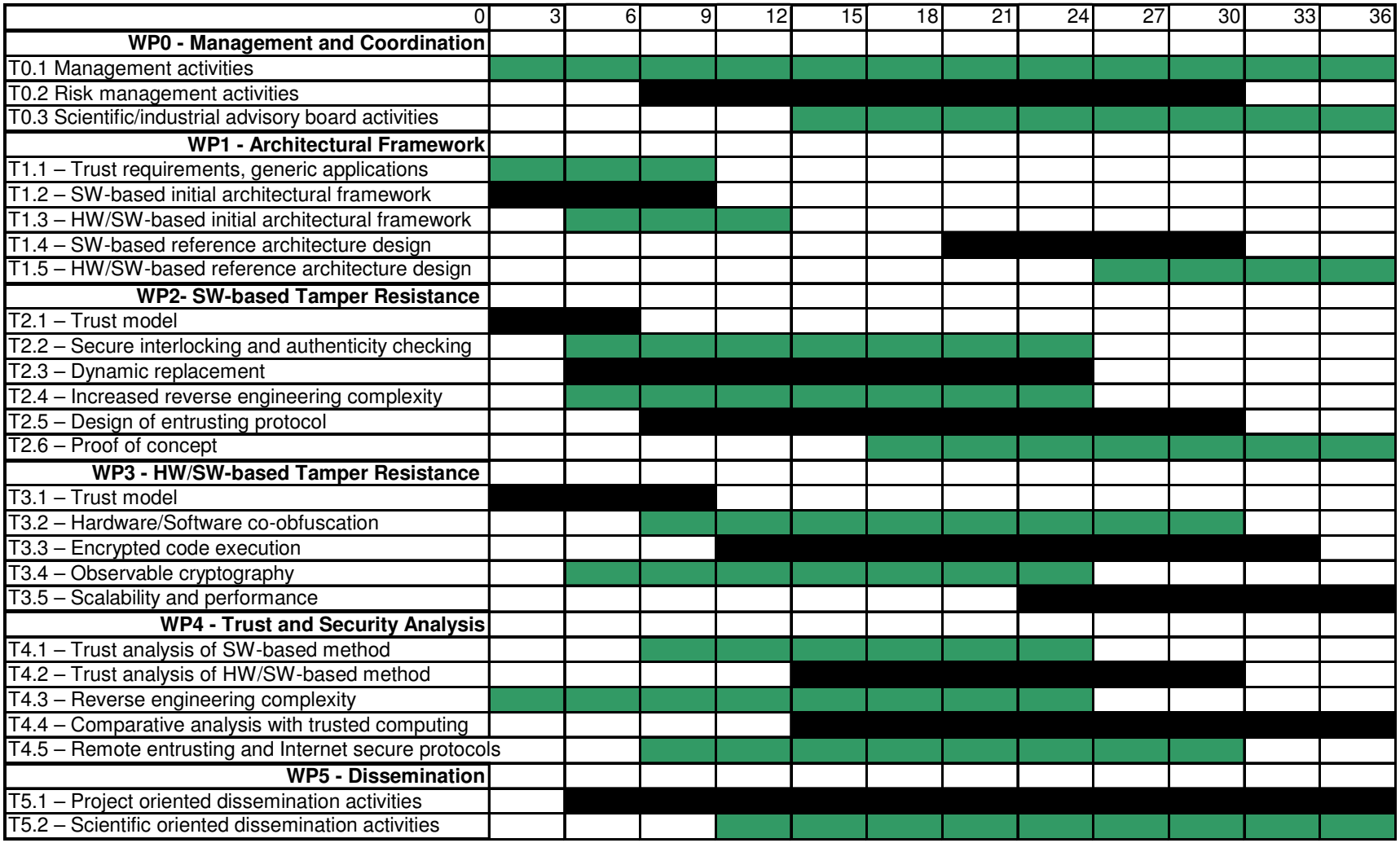
Work Plan Organization

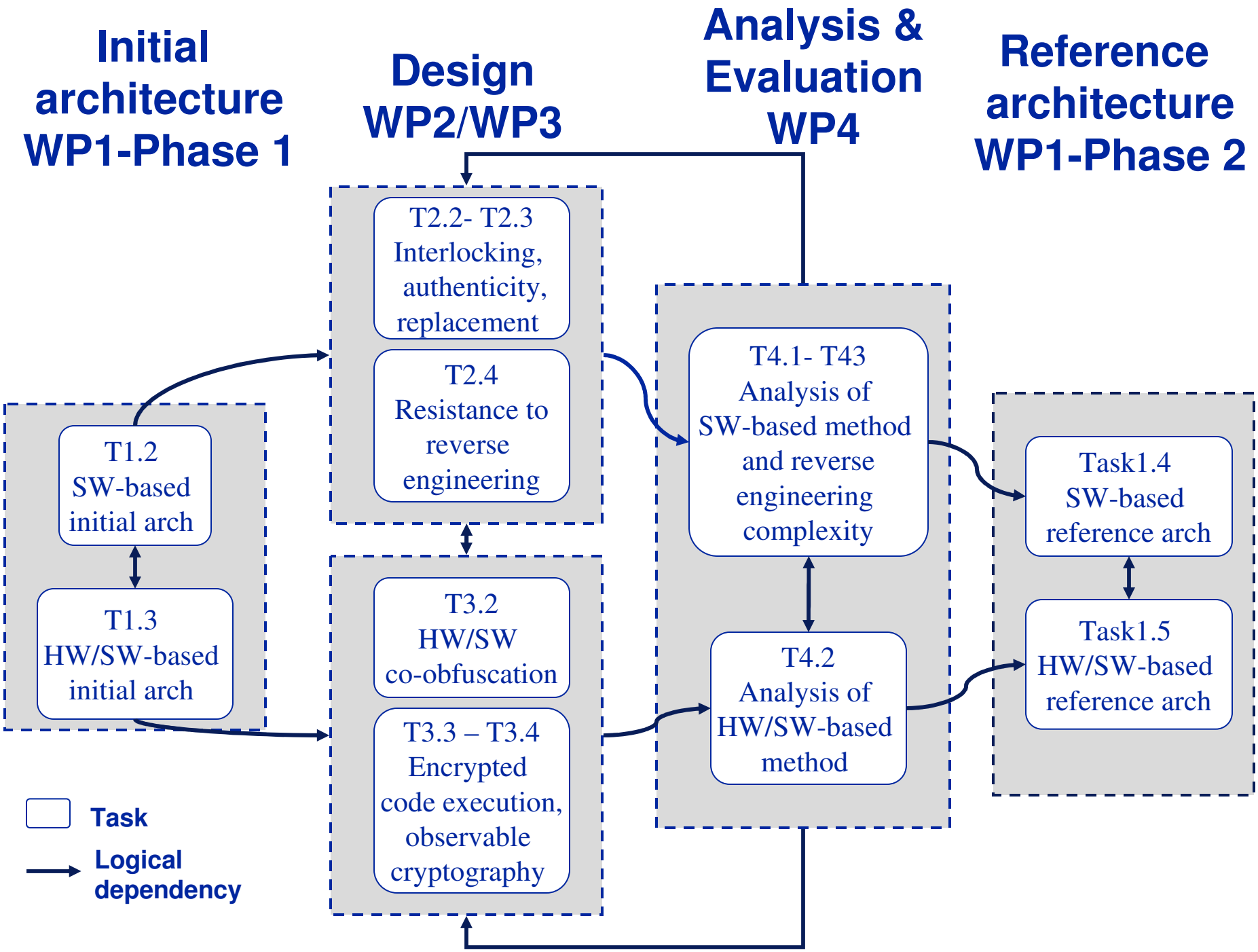


Primary Expected Results

- WP1: Reference architectures – SW & HW/SW
- WP2 & WP3: Remote entrusting methodologies
- WP2 & WP3: Proof-of-concepts/prototypes
- WP1/2/3: Standardizations
- WP1/2/3: Product solution definitions

Work-planning and timetable – Gantt




Initial architecture WP1-Phase 1
Design WP2/WP3
Analysis & Evaluation WP4
Reference architecture WP1-Phase 2
T2.2- T2.3 Interlocking, authenticity, replacement
T2.4 Resistance to reverse engineering
T1.2 SW-based initial arch
T1.3 HW/SW-based initial arch
T3.2 HW/SW co-obfuscation
T3.3 – T3.4 Encrypted code execution, observable cryptography
T4.1- T43 Analysis of SW-based method and reverse engineering complexity
T4.2 Analysis of HW/SW-based method
Task1.4 SW-based reference arch
Task1.5 HW/SW-based reference arch
Task
Logical dependency

Work package list

No.	Workpackage title	Lead contractor	Person-months	Start month	End month	Deliverable No.
WP0	Management and Coordination	P1	14	1	36	6
WP1	Architectural Framework	P1	36	1	36	5
WP 2	SW-based Tamper Resistance Methods	P2	95	1	36	7
WP 3	HW/SW-based Tamper Resistance Method	P4	72	1	36	5
WP 4	Trust and Security Analysis	P1	75	1	36	8
WP 5	Dissemination, Exploitation, Standardization	P1	18	1	36	5
	TOTAL		**310**			

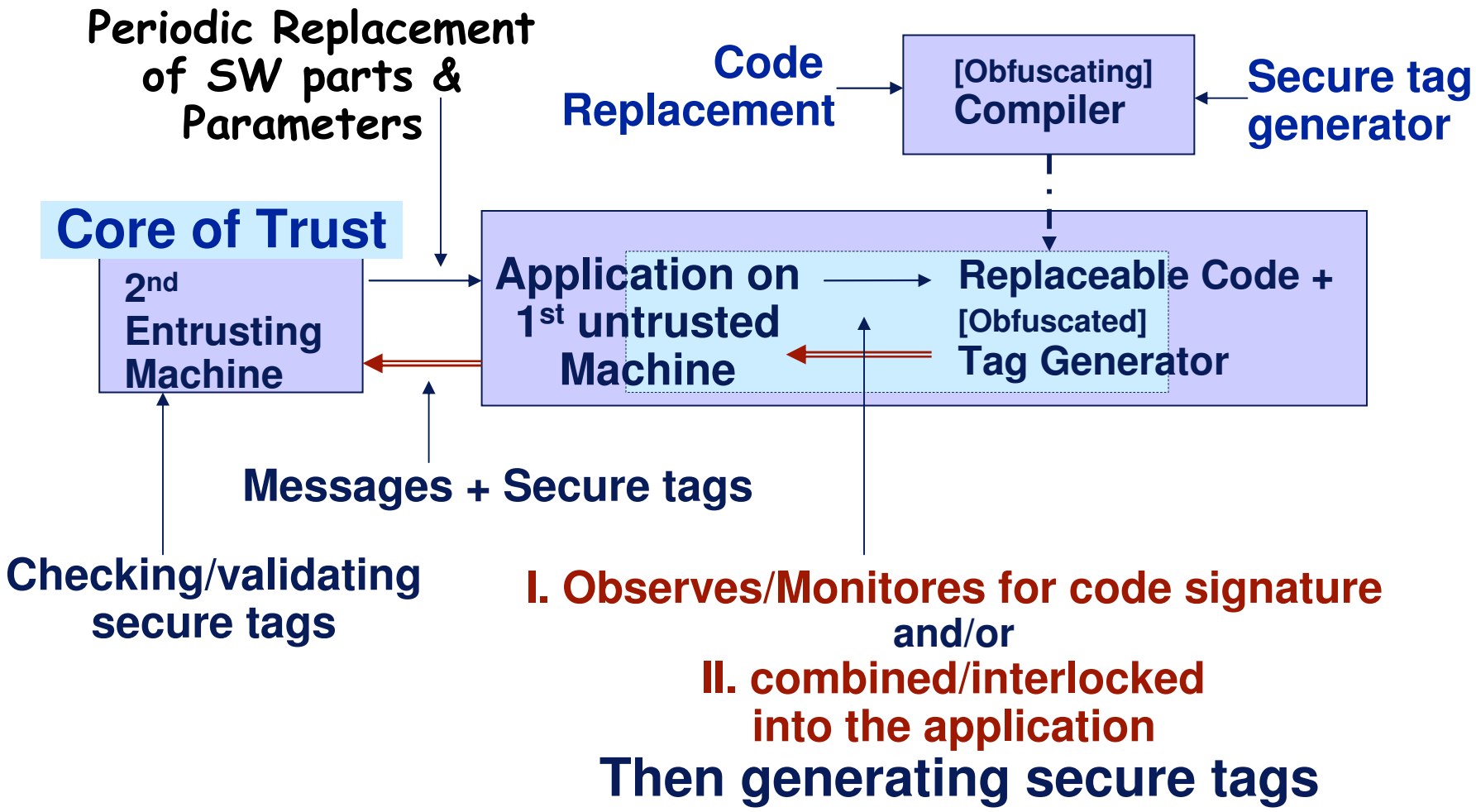
WP1 - Architectural Framework

- Objective
  - To define and analyse generic application classes and their corresponding trust requirements
  - To define unified framework and specific requirements for the two basic methods:
    - SW-based and
    - HW/SW-based
  - To analyse the research results towards possible proof of concepts

WP1 - Architectural Framework

- **Description of work**
  - **Task1.1 – Trust requirements of generic classes of applications (Dates: M0-9)**
    - **Participants: UNITN, POLITO, GEM+, KUL, SPIIRAS**
  - **Task1.2 – SW-based initial architectural framework (Phase 1) (Dates: M0-9) – see diagram:**
    - **Participants: POLITO, UNITN, KUL**
      - Subtask T1.2.1 Trust and security requirements
      - Subtask T1.2.2 Design alternatives: programming environments, operating systems, etc.
      - Subtask T1.2.3 General design analysis and initial trust attack models

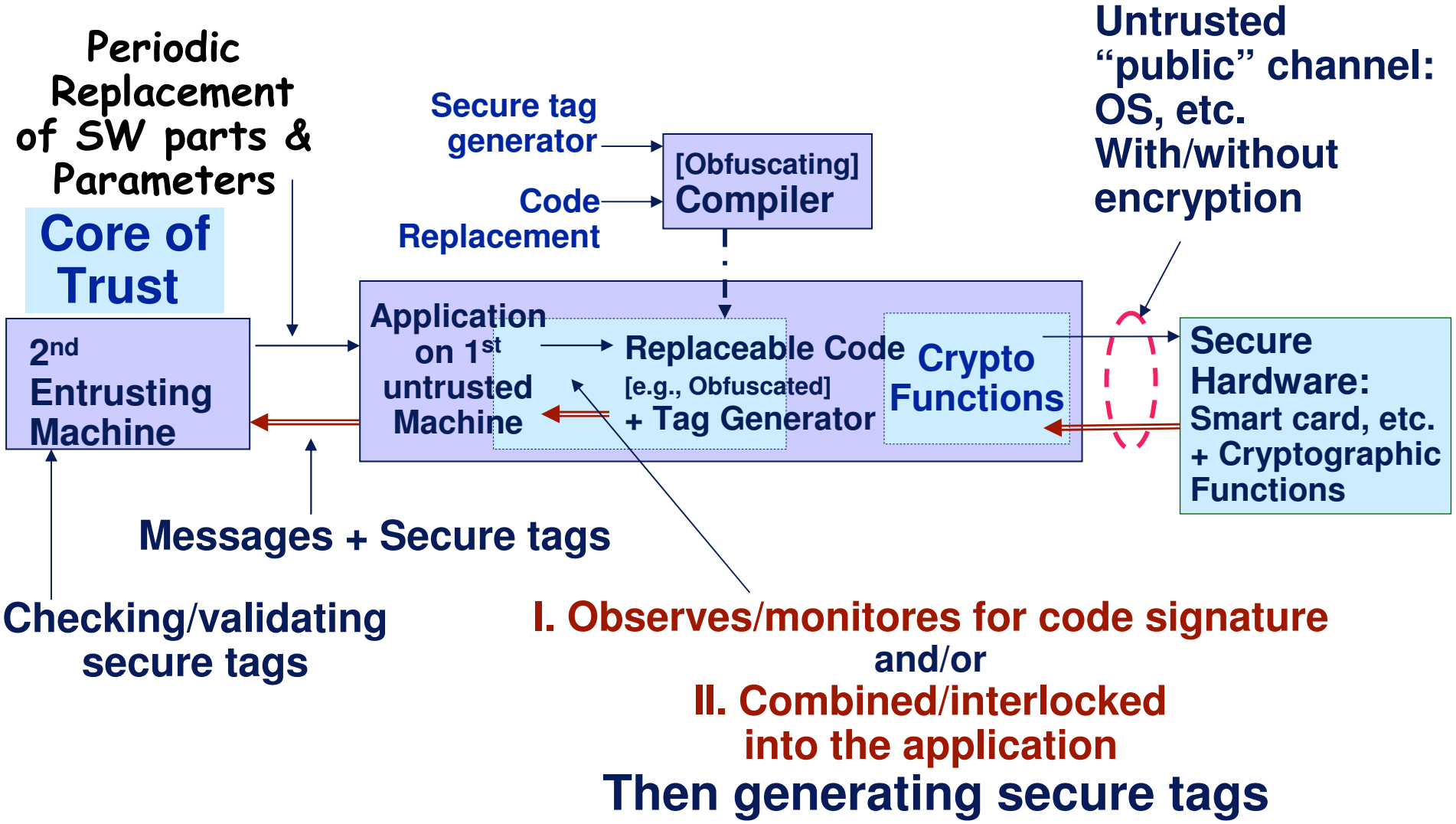
Initial Functional Description: SW-based Tamper Resistance - TR



WP1 - Architectural Framework

- Description of work
  - Task1.3 – HW/SW -based initial architectural framework (Phase 1) (Dates: M3-12)–see diagram:
    - Participants: KUL, GEM+, UNITN
      - Subtask T1.2.1 Trust and security requirements for both SW and HW
      - Subtask T1.2.2 Design alternatives: programming environments, operating systems, etc.
      - Subtask T1.2.3 General design analysis and initial trust attack models

Initial Functional Description: HW/SW-based - TR



WP1 - Architectural Framework

- Description of work
  - Task1.4 – SW-based reference architecture and product solution definitions (Dates: M18-30)
    - Participants: UNITN, POLITO, KUL
      - Subtask T1.4.1 Reference architecture design. Refinement of initial architectural framework
      - Subtask T1.4.2 Proof of concept design based on the reference architecture
      - Subtask T1.4.3 Product solution definitions and possible standardization while collaborating with selected members of the scientific/industrial advisory board
  - Task1.5 – HW/SW-based reference architecture and product solution definitions (Dates: M24-36)
    - Participants: UNITN, KUL, GEM+
      - Subtask T1.5.1 Reference architecture design. Refinement of initial architectural framework
      - Subtask T1.5.2 Proof of concept design based on the reference architecture
      - Subtask T1.5.3 Product solution definitions and possible standardization while collaborating with selected members of the scientific/industrial advisory board

WP1 - Architectural Framework

WP1 Deliverables List			
No.	**Title**	**Responsible**	**Delivery date**
D1.1	Analysis of generic classes of applications	UNITN	M6
D1.2	SW-based method initial architecture	POLITO	M6
D1.3	HW/SW-based method initial architecture	KUL	M12
D1.4	SW-based method final reference architecture design	UNITN	M30
D1.5	HW/SW-based method final reference architecture design	UNITN	M36
WP1 Milestones List			
No.	**Title**	**Responsible**	**Delivery date**
M1.1	Summary report of D1.2 and D1.3	POLITO/ KUL	M12
M1.2	Summary report of D1.4 and D1.5	UNITN	M36

WP2 - Software-based Tamper Resistance Methods for Remote Entrusting

- Objectives:
  - To design and compare various SW-based alternatives
  - **To investigate and apply solutions developed for software dependability to remote entrusting**
  - **To design and analyze software tamper resistance using two basic methods:**
    - Dynamically replacing software modules during run-time
    - Increasing the complexity of software reverse engineering
  - To design and analyze solution alternatives for continuous and secure signature generation

WP2 - Software-based Tamper Resistance Methods for Remote Entrusting

- **Description of work**
  - **Task2.1 – Trust model (Dates: M0-6)**
    - **Participants: UNITN, POLITO**
      - Subtask2.1.1. Analysis of untrusted environment weaknesses and elicitation of trust assumptions
      - Subtask2.1.2. Initial estimation of reverse engineering complexity and the tradeoff between replacement and resistance to reverse engineering
  - **Task2.2 – Secure interlocking and authenticity checking (Dates: M3-24)**
    - **Participants: POLITO**
      - Subtask2.2.1. Software dependability techniques to securely combine original application and secure software module
      - Subtask2.2.1. SW dependability techniques to protect authenticity of application code and data
  - **Task2.3 – Dynamic replacement for increased tamper resistance (Dates: M3-24)**
    - **Participants: UNITN, POLITO, SPIIRAS**
      - Subtask2.3.1. Replacement strategies for interpreted code
      - Subtask2.3.2. Replacement strategies for compiled code
      - Subtask2.3.3. Automated and non-predictable generation of secure software

WP2 - Software-based Tamper Resistance Methods for Remote Entrusting

- Description of work
  - Task2.4 – Increased reverse engineering complexity for software protection (Dates: M3-24)
    - Participants: KUL, GEM+
      - Subtask2.4.1. Source-to-source obfuscation.
      - Subtask2.4.2. Obfuscation of Java byte code.
      - Subtask2.4.3. Protection of embedded keys with white-box cryptography techniques.
  - Task2.5 – Entrusting protocol (Dates: M6-30)
    - Participants: POLITO, SPIIRAS
      - Subtask2.1.1. Protocol design
      - Subtask2.1.2. Protocol analysis
  - Task2.6 – Proof of concept (Dates: M15-36)
    - Participants: POLITO, KUL, GEM+, UNITN
      - Subtask2.6.1. Identification of a sample application domain to be prototyped and selection of application technology (OS, programming language, support tools)
      - Subtask2.6.2. Development of prototype infrastructure and integration of solutions proposed in previous tasks

WP2 - Software-based Tamper Resistance Methods for Remote Entrusting

WP2 Deliverables List			
No.	**Title**	**Responsible**	**Delivery**
D2.1	Trust model and assumption for software-based TR methods	UNITN	M6
D2.2	Methods to dynamically replace the secure software module and to securely interlock applications with secure SW module (interim)	POLITO	M12
D2.3	Methods to dynamically replace the secure software module and to securely interlock applications with secure SW module	UNITN	M24
D2.4	Protection methods for hardening the secure software module	KUL	M24
D2.5	Protocol design	POLITO	M36
D2.6	Proof of concept	POLITO	M36
WP2 Milestones List			
No.	**Title**	**Responsible**	**Date**
M2.1	Executive summary of initial version of SW-based approach	POLITO	M12
M2.2	Selection of a sample application domain to be prototyped, and agreement on technological aspects of test-bed development	POLITO	M12
M2.3	Selection of possible obfuscation transformations for proof of concept	GEMPLUS	M24
M2.4	Proof-of-concept and possible product solution analysis	POLITO	M36

WP3 - Hardware/Software-based Tamper Resistance Method for Remote Entrusting

- Objectives:
  - To investigate the combination of HW/SW-based protection
  - **To utilize HW that alone may not be able to provide enough functionality, e.g., smart cards**
  - To investigate:
    - Low trust protection mode, execution of the code must be split between HW and SW, in a way that maximizes protection and minimizes the performance penalty
    - Full trust protection, methods that allow an attacker to observe the entire communication between the computing engine (the secure hardware) and the memory (in the PC), without learning any useful information

WP3 - Hardware/Software-based Tamper Resistance Method for Remote Entrusting

- **Description of work**
  - Task3.1 – Trust model (Dates: M1-9)
    - Participants: UNITN, GEM+, KUL
  - Task3.2 – HW/SW co-obfuscation (Dates: M6-30)
    - Participants: KUL, POLITO
  - Task3.3 – Encrypted code execution (Dates: M9-33)
    - Participants: GEM+, KUL
  - Task3.4 – Observable cryptography (Dates: M4-24)
    - Participants: KUL
  - Task3.5 – Scalability & performance (Dates: M21-36)
    - Participants: SPIIRAS , UNITN, GEM+, KUL

WP3 - Hardware/Software-based Tamper Resistance Method for Remote Entrusting

WP3 Deliverables List			
No.	**Title**	**Responsible**	**Delivery**
D3.1	Trust Model for the combined Hardware-Software approach	UNITN	M12
D3.2	First Analysis Encrypted Code and HW assisted SW Protection	KUL	M24
D3.3	Encrypted code final report	GEM+	M30
D3.4	Hardware assisted Software Protection for entrusting module	KUL	M33
D3.5	Report on combination of the different approaches	SPIIRAS	M36
WP3 Milestones List			
No.	**Title**	**Responsible**	**Date**
M3.1	Trust model for HW/SW method	UNITN	M12
M3.2	Application of computation with encrypted data to whitebox crypto	GEM+	M24
M3.3	Plausibility analysis and exact goals for Task 3.5	KUL	M36

WP4 - Trust and Security Analysis

- Objectives:
  - To provide trust and security analysis of the tamper resistance techniques introduced in WP2 and WP3
  - To provide feedback to the overall solution architecture in WP1
  - To provide tools to evaluate the results delivered within the project
  - To develop methodology for evaluating reverse engineering complexity

WP4 - Trust and Security Analysis

- Description of work:
  - **Task T4.1 – Trust analysis of SW-based method (Dates: M6-24)**
    - **Participants: SPIIRAS, POLITO**
      - Subtask T4.1.1. Analysis of possible attacks
      - Subtask T4.1.2. Trust analysis of interlocking, authenticity checking, and dynamic replacement
  - **Task T4.2 – Trust analysis of HW/SW-based method (Dates: M12-30)**
    - **Participants: KUL, GEM+, UNITN**
      - Subtask T4.2.1: Trust analysis of HW/SW Co-obfuscation
      - Subtask T4.2.2: Trust analysis of encrypted Code Execution
      - Subtask T4.2.3: Trust analysis of white-box and physical observable cryptograph
      - Subtask T4.2.2: Implementability of the trust assumptions must be evaluated
      - Subtask T4.2.3: Analysis of pitfalls of HW/SW- based remote entrusting mechanisms and possible attacks on remote entrusting

WP4 - Trust and Security Analysis

- Description of work:
  - **Task T4.3 – Reverse engineering complexity (Dates: M0-24)**
    - **Participants: UNITN, POLITO**
      - Subtask T4.3.1: Evaluation metrics for code replacement
      - Subtask T4.3.2: Evaluation metrics for code obfuscation
  - **Task T4.4 – Comparative analysis with TC from OS perspective (Dates: M12-36)**
    - **Participants: UNITN, KUL**
      - Subtask T4.4.1. Comparative analysis (and synergies) of RE-TRUST solutions and alternative solutions using trusted hardware (i.e. TC).
      - Subtask T4.4.2. Analysis on which trust assumptions are required at the OS level to assure that RE-TRUST cannot be circumvented. We may investigate different levels of assurance
  - **Task T4.5 – Remote entrusting and Internet secure protocols (Dates: M6-30)**
    - **Participants: SPIIRAS, UNITN**
      - Subtask T4.5.1 Analysis of integration of remote entrusting with existing Internet security protocols
      - Subtask T4.5.2 Integration and analysis of secure protocols to support remote entrusting methods. For instance, how to integrate public key infrastructure for server authentication

WP4 - Trust and Security Analysis

WP4 Deliverables List			
No.	**Title**	**Responsible**	**Delivery**
D4.1	Initial trust analysis of SW-based and HW/SW-based methods	POLITO	12
D4.2	Trust analysis of SW-based method	POLITO	24
D4.3	Analysis of the Reverse Engineering Complexity	UNITN	24
D4.4	Trust analysis of HW/SW-based method	KUL	30
D4.5	Comparative analysis of RE-TRUST with TC	UNITN	36
D4.6	Analysis of OS issues within RE-TRUST	UNITN	36
D4.7	Analysis of interaction of RE-TRUST with security protocols	SPIIRAS	36
WP4 Milestones List			
No.	**Title**	**Responsible**	**Date**
M4.1	Outcome of the reverse engineering investigation. Definition of the feasibility and of the parameter to consider by replacement and obfuscation	UNITN	24
M4.2	Results of trust analysis related to SW-based methods	POLITO	24
M4.3	Results of the two trust analysis related to SW/HW-based methods	KUL	36