Possible directions for a follow up of the project

Mariano Ceccato

ceccato@fbk.eu

Adolfo Villafiorita

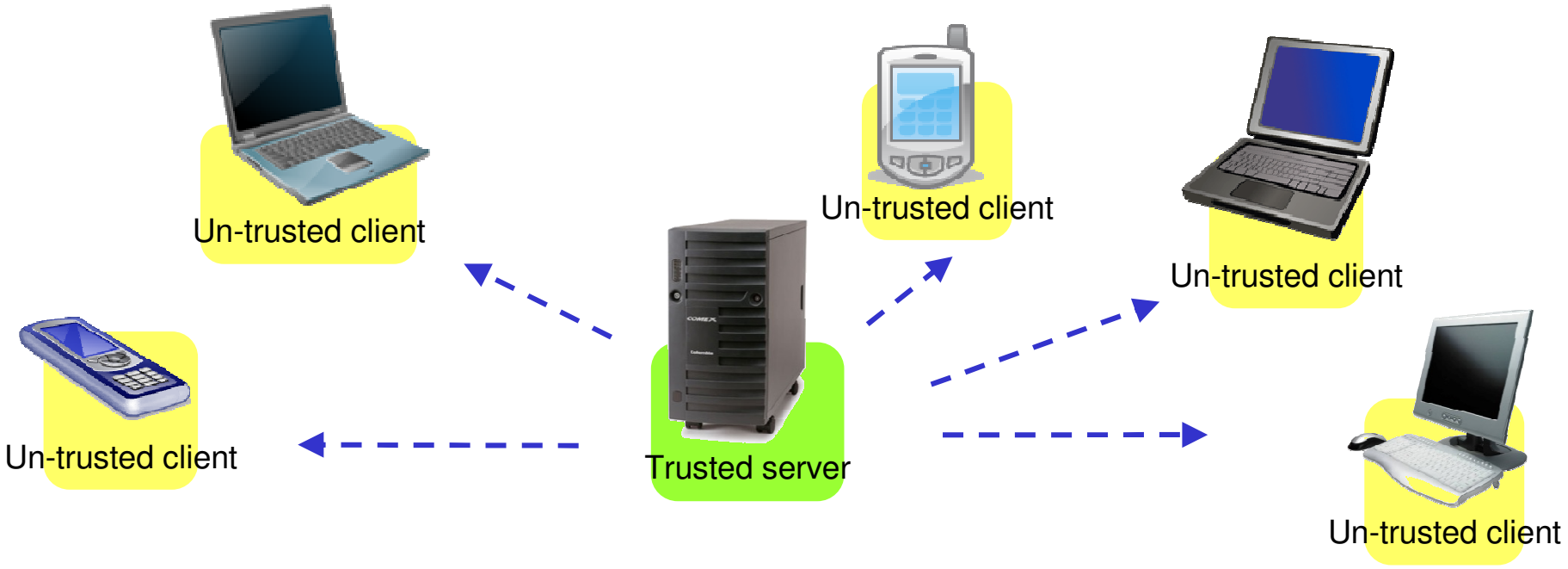
adolfo.villafiorita@fbk.eu

Outline of the Talk

- Summary of some techniques
- Information about call 5
- Some scenarios
- Next steps

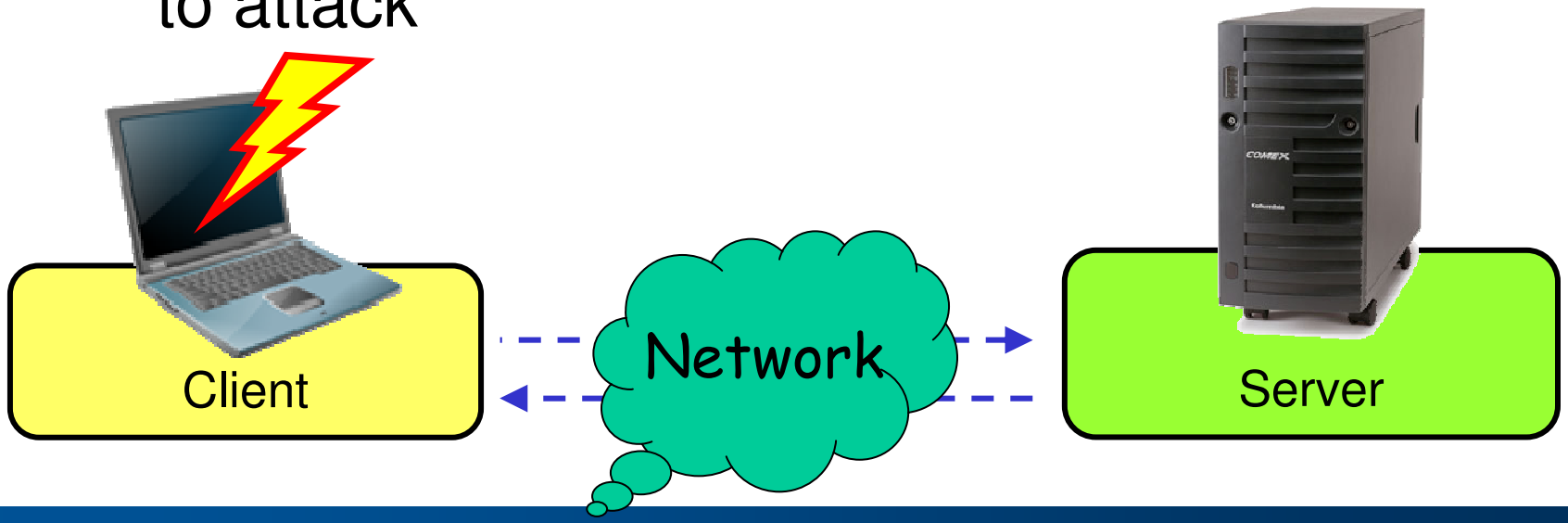
Remote software trusting

- *Remote software authentication*: ensuring a (server) that an un-trusted host (client) is running a “healthy” version of a program (code integrity)
- Before delivering any service the server wants to know that the client is executing according to its expectations

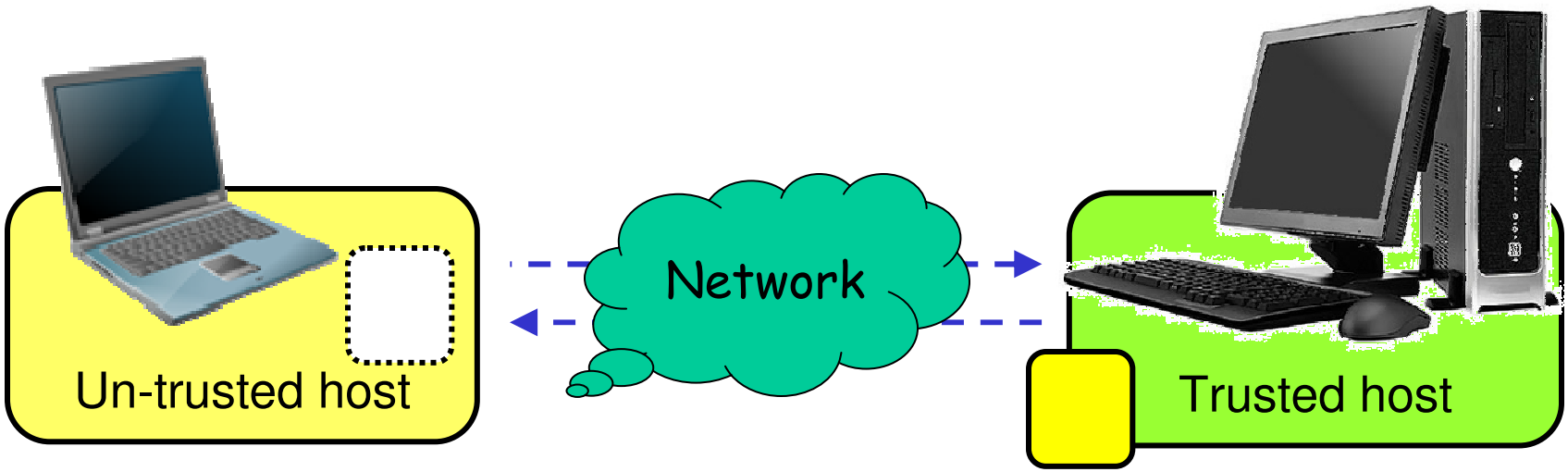


Attacker goal

- **Goal:** to tamper with the application code without being detected by the server
  - Substantial program understanding effort by a human to understand the inner logic to attack

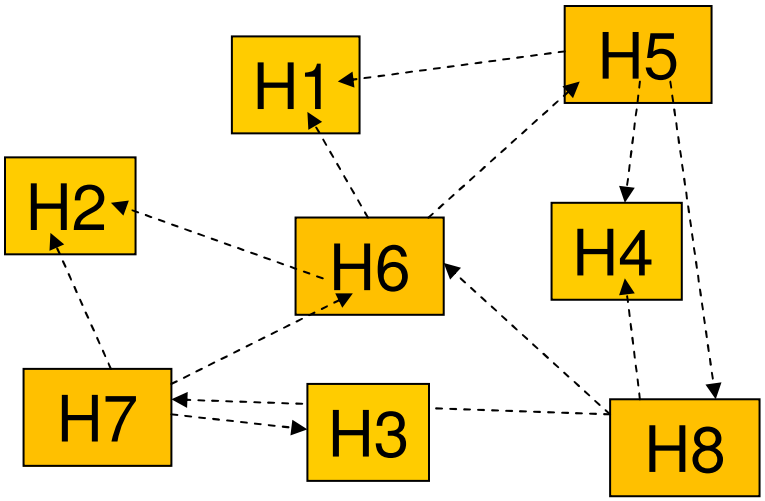


Barrier slicing

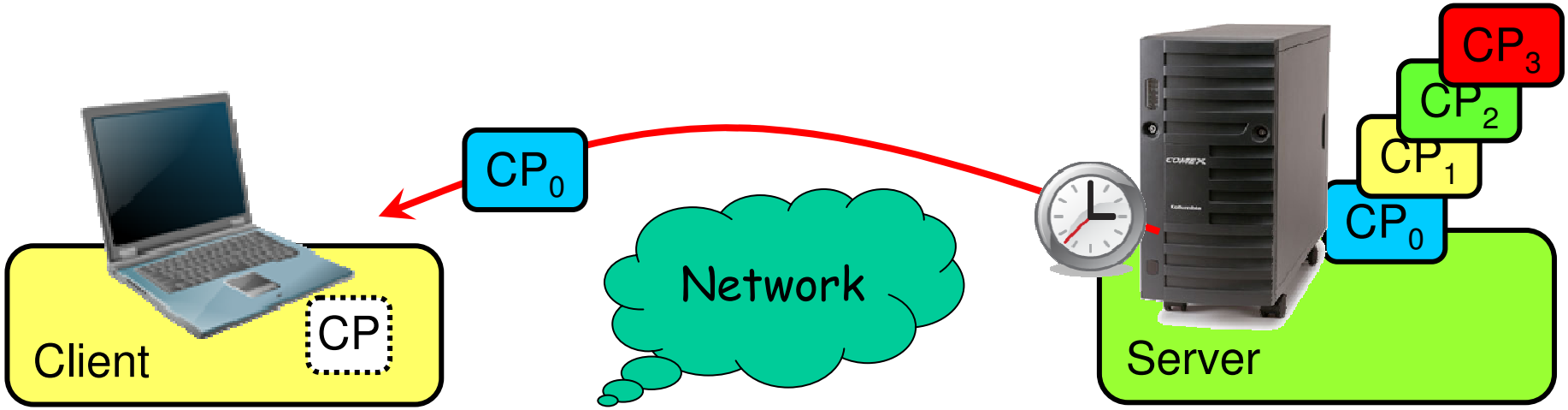


Open problems in barrier slicing

- It does not exactly fit the reference architecture
- Distributed network of trust based on code splitting
- An attack is successful in more than N hosts collude to mount an attack



Orthogonal replacement



repeat

CP_i = RendomTransform (CP)

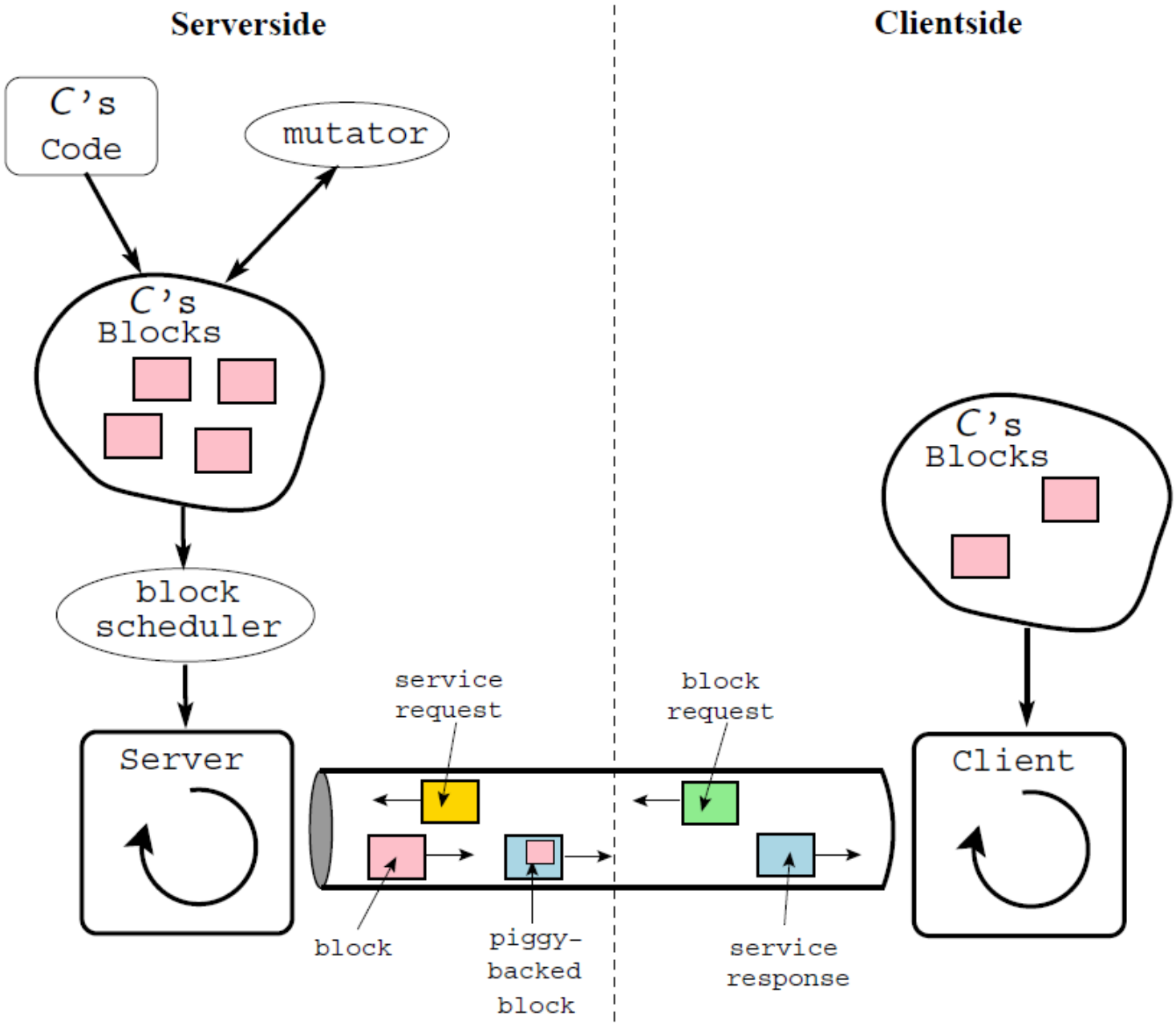
CP = CP_i

(C_i, S_i) = MoveCompToServer(CP_i, C_1,…,C_{i-1})

until ($C_i \perp C_1$) ∧ … ∧ ($C_i \perp C_{i-1}$)

Open problems in orthogonal replacement

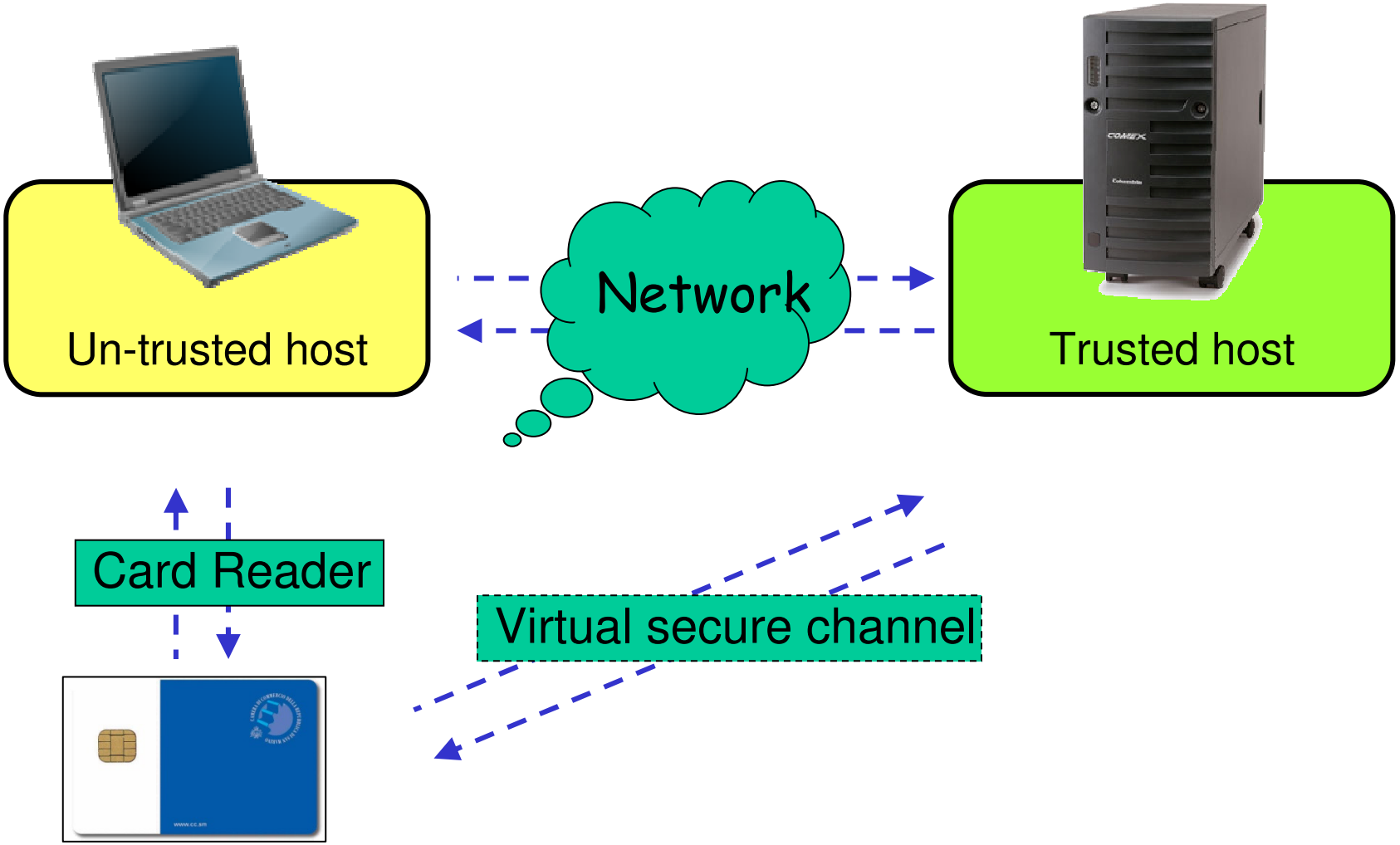
- Extending the notion of code orthogonality to
  - Internal data structures
  - Network messages
- More robust check for orthogonality (e.g., semantic check?)


Serverside
Clientside
C's Code
mutator
C's Blocks
block scheduler
Server
service request
block request
block
piggy-backed block
service response
C's Blocks
Client

Open problems on continuous replacement

- Measuring the level of tamper-proofing
- Engineering how to generate new blocks
- Clarify how attackable / protected are newly generated blocks

Hardware assistance

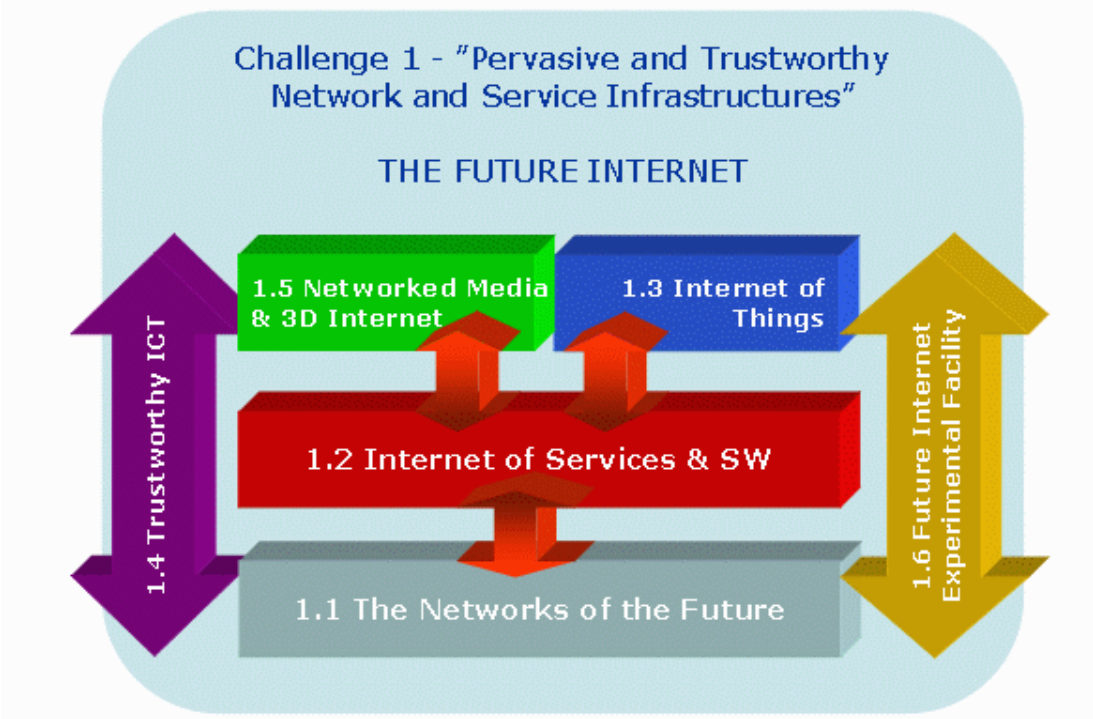


Other approaches

- White Box Remote Procedure Execution
- Crypto guards
- TPM tick stamping
- White box cryptography

Information about the Call

- ICT Challenge 1, Call 5, Objective 1.4
- Pervasive and Trustworthy Network and Service Infrastructure
- Trustworthy ICT



Information about the Call

- Publication Date: ~ 31/07/2009
- Closure Date: 03/11/2009
- Budget: 80M (IP, STREP) + 10M (NoE, CSA)
- Budget allocation IP >= 40M; STREP >= 26M
- About 15-20 STREP projects (?)
- **Event planned on the 18th of June 2009** in Brussels ("presentation of the call and opportunities to present ideas in two/three slides")

Call Details

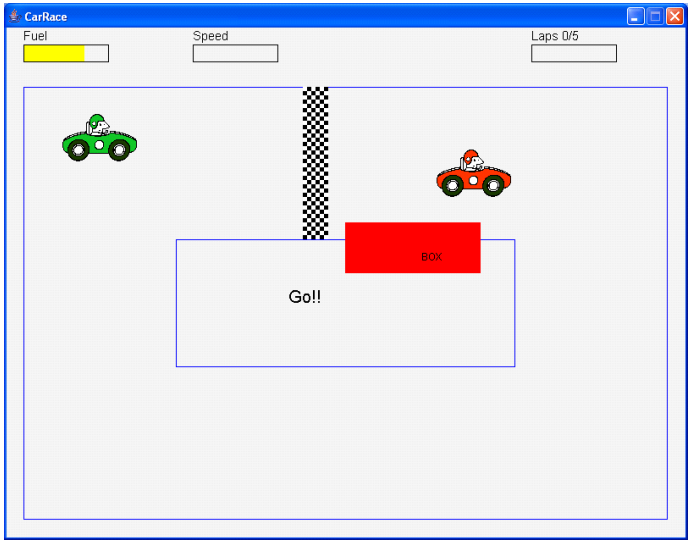
- Four main target outcomes:
  - Trustworthy Network Infrastructure (IP)
  - Trustworthy Service Infrastructure (IP)
  - Technology and Tools for Trustworthy ICT (call for small or medium-scale focused research actions STREP)
  - Networking Coordination and Support (Noe and CSA)

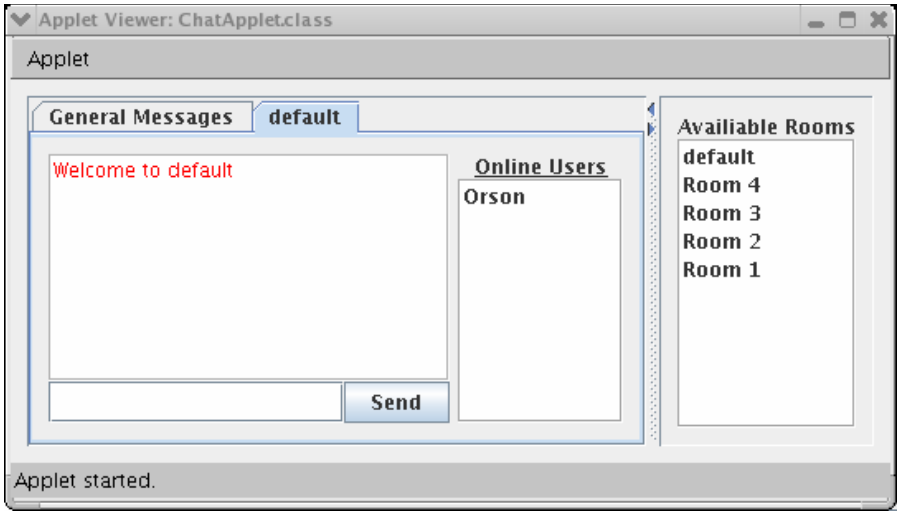
- Technology and Tools for Trustworthy ICT
  - In highly distributed networked process control systems and in networks of very high number of things. Understanding threat patterns for pro-active protection.
  - For user-centric and privacy preserving identity management, including for management of risks and policy compliance verification.
  - For management and assurance of security, integrity and availability, also at very long term, of data and knowledge in business processes and services.
  - **For assurance and assessment of the trustworthiness of complex and continuously evolving software systems and services.**
  - **In enabling technologies for trustworthy ICT.** This includes cryptography, biometrics; trustworthy communication; virtualisation; and certification methodologies.

- For all projects:
  - Improved European industrial competitiveness in markets of trustworthy ICT, by: facilitating economic conditions for wide take-up of results; offering clear business opportunities and consumer choice in usable innovative technologies; and increased awareness of the potential and relevance of trustworthy ICT.
  - Adequate support to users to make informed decisions on the trustworthiness of ICT. Increased trust in the use of ICT by EU citizens and businesses. Increased societal acceptance of ICT through understanding of legal and societal consequences.

Application scenarios: gaming

- Avoid players to cheat and gain unfair advantages
- Lot of software development in this field
- Maybe not strategic for EU



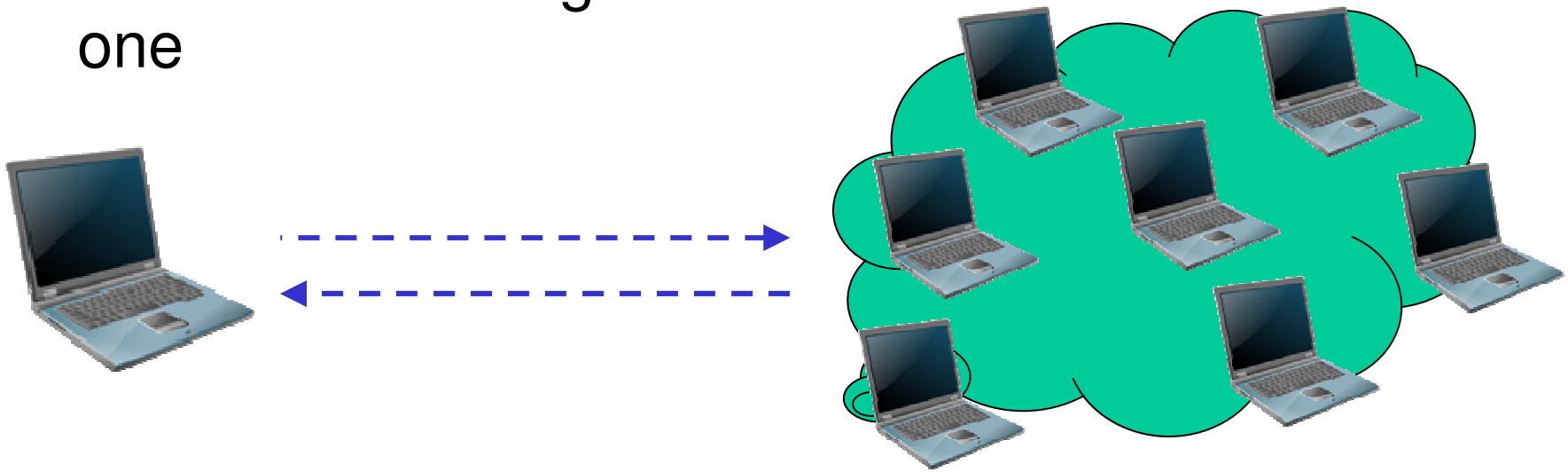


Application scenarios: Web 2.0

- Rich web applications are more and more common (e.g. Google office)
- Code is most of the time in clear (JavaScript/ajax)
- Vulnerable to phishing, spyware, …

Application scenarios: cloud computing

- A computation intensive problem (e.g. genomics) is delivered, no idea what host will run it
- Business model: pay-per-computation
- Problem: ensuring that the result is the correct one



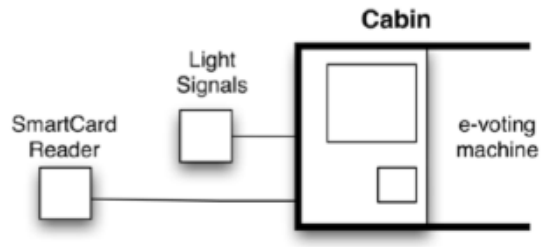
Application scenarios: pay-per-use licenses

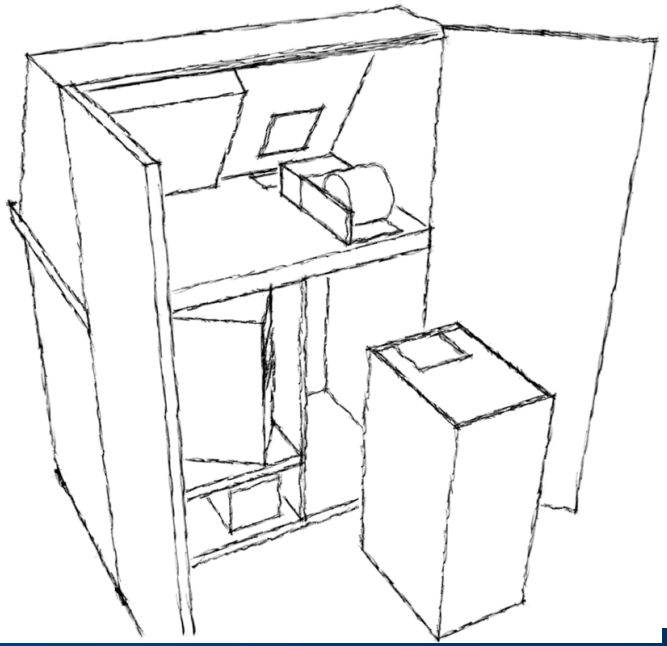
- Instead of buying software forever, pay it only when you need it
  - Under development countries
  - Small companies that requires very expensive tools
- Problem: enforcing that the software is executed no more times (or no longer) than allowed

Application scenarios: e-voting

eVoting

- DRE with VVPAT
- External signaling system
- Smart-card for operating the machine
- Java + (Custom) Linux
- Voting application about ~11K SLOC
- Core logic formally verified



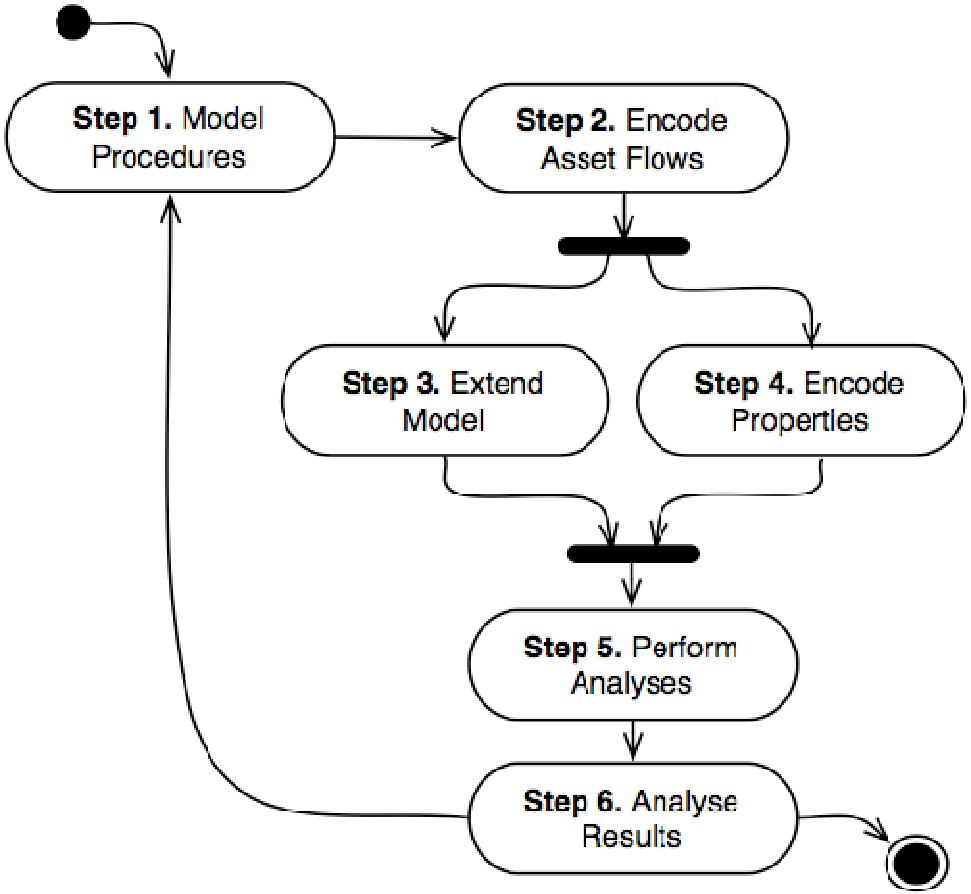


Formal Procedural Security Analysis

- eVoting: a lot more than just the machine:
  - digital and physical assets
  - asset mobility and evolution change the risks associated to breaches
  - security depends upon procedures performed by various actors over which there is limited or no control
- Approach: Methodology and analysis to verify critical procedures

The Methodology

- Based on the concept of threat-injection: capabilities attackers have in “modifying” behaviors
- Works well for both errors and malicious attacks
- Based on formal verification
- Complexity is an issue



Voting and Remote Entrusting

- In Internet Voting: obvious!
- Even in the scenario depicted above:
  - Trusting the machines: OS is critical as the software and is left in a potentially non-controlled environment (Mutual entrusting/Remote entrusting)
  - Trusting the server: when tabulated data is sent for polling stations to

Proposal

- Driven by the application scenario(s)
- Centered on the development of a tool (or procedures) to make remote entrusting applicable
- It includes activities related to consolidating the theoretical framework and, possibly, experimentations to validate the theoretical framework

Next Steps

- Consolidate idea and consortium
- Prepare for the June event with a clear vision
- Consolidate technical idea, plan and budget before the Summer
- Finalize (and possibly adjust) work after release of official proposal

Next Steps

Questions?

Considerations?!

Call for contribution!