

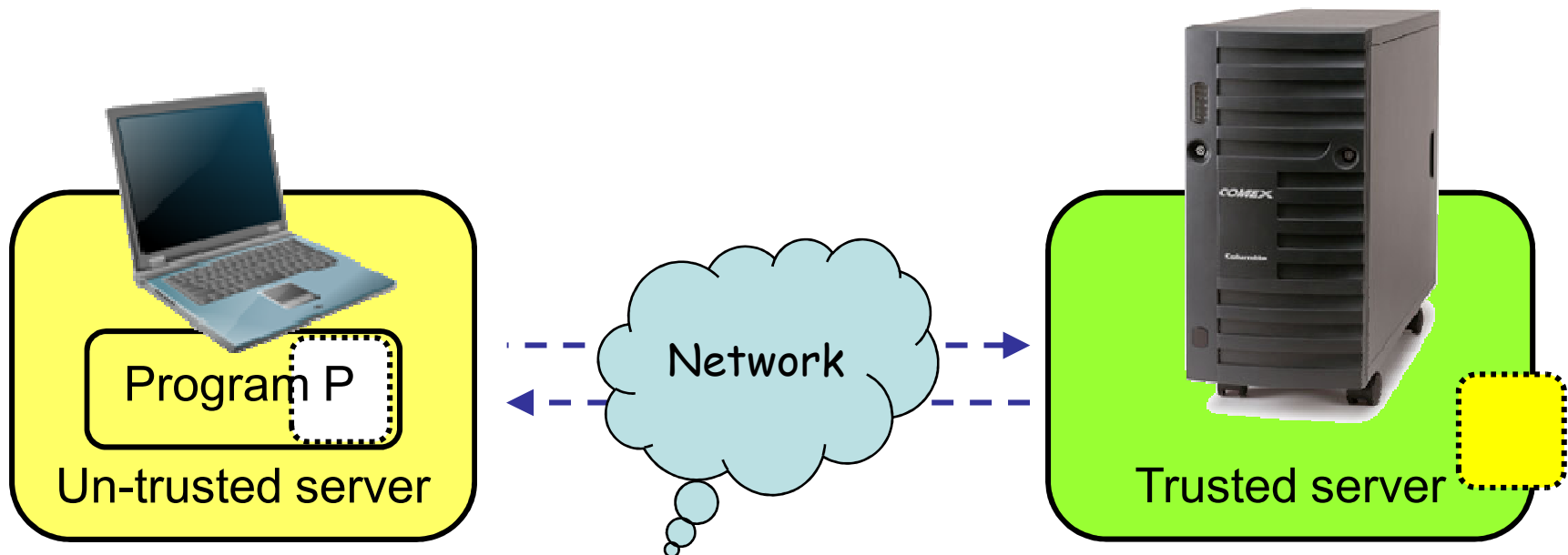
Empirical study on software obfuscation

Ceccato Mariano

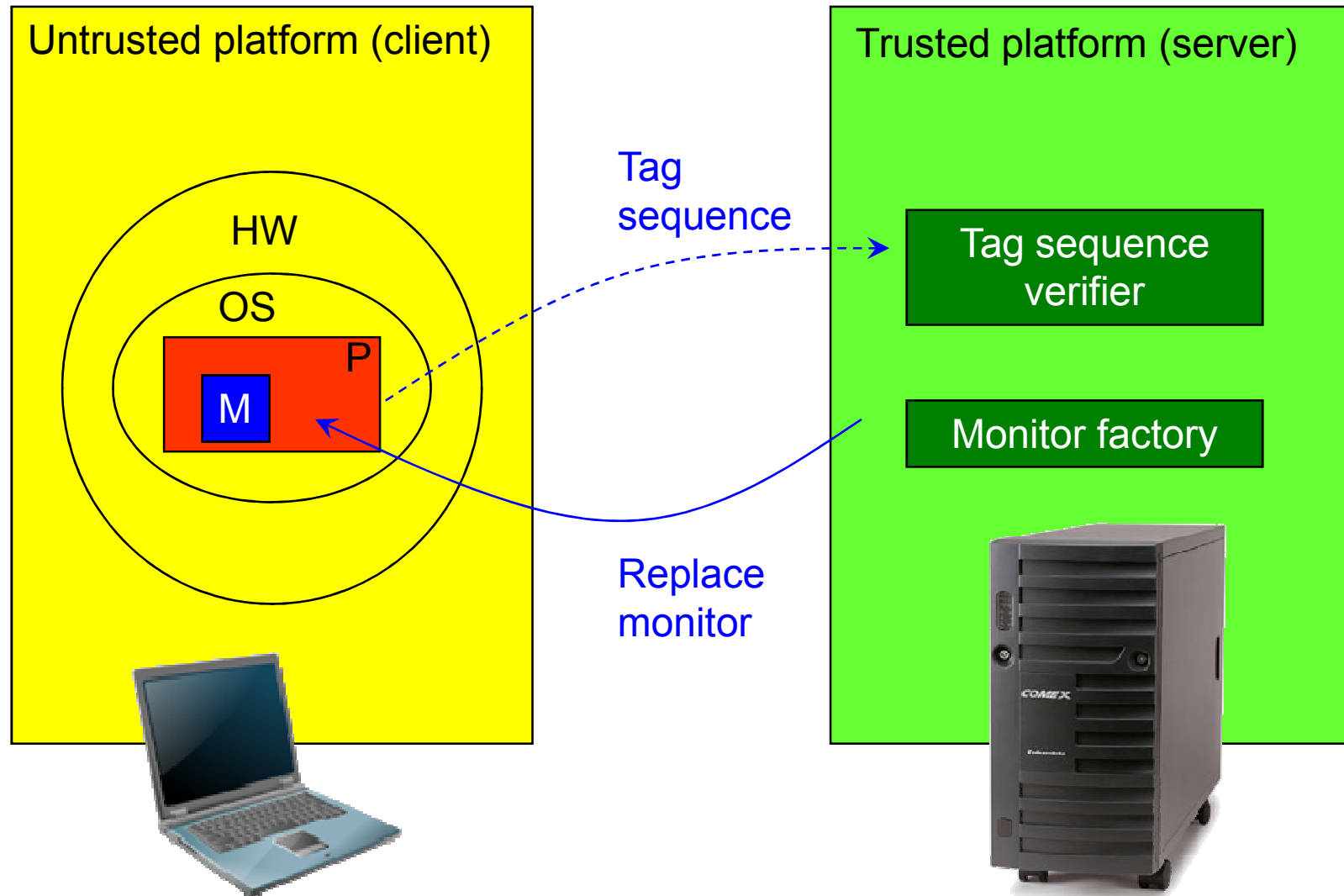
Fondazione Bruno Kessler-IRST, Trento, Italy

Barrier slicing approach

- A portion of the program is moved and it runs on the server, thus it is protect against tampering.



Reference architecture



Considerations

Obfuscation offer a limited level of protection: it can be broken with enough effort.

- Need to measure this level of protection.

There is a general acceptance that obfuscation is useful in protecting intellectual property, but...

- Obfuscation has not been **validated**
- Different obfuscation methods have never been **compared**.
- No study on which method to adopt to protect a given application
 - It may depend on **what** I want to protect.

Research questions

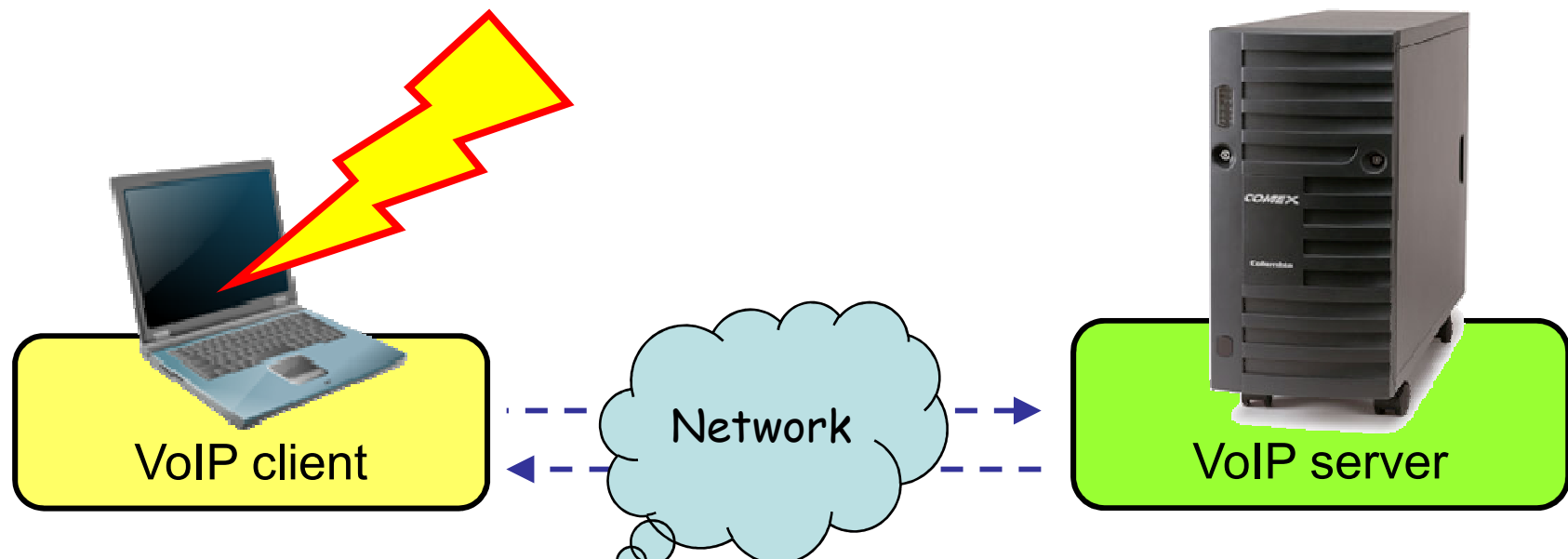
- What is the level of protection offered by obfuscation?
 - How much obfuscation make the understanding **effort increase**?
 - How obfuscation affects the **correctness** of maintenance/attack tasks?
 - How do different obfuscation techniques **compare**?

```
1  Student guy = new Student();  
2  String name = "Mathematics";  
3  Course course = new Course(name);  
4  guy.apply(course);  
5  course.commitChanges();
```

```
1  y1 x1 = new y1();  
2  String x2 = "Mathematics";  
3  y2 x3 = new y2(x2);  
4  x1.z1(x3);  
5  x3.z2();
```

The experiment

- We ask subjects to apply some selected attacks to a set of objects (network) applications.
- We study how obfuscation impacts on
 - time required to perform the attacks
 - correctness of the attacks

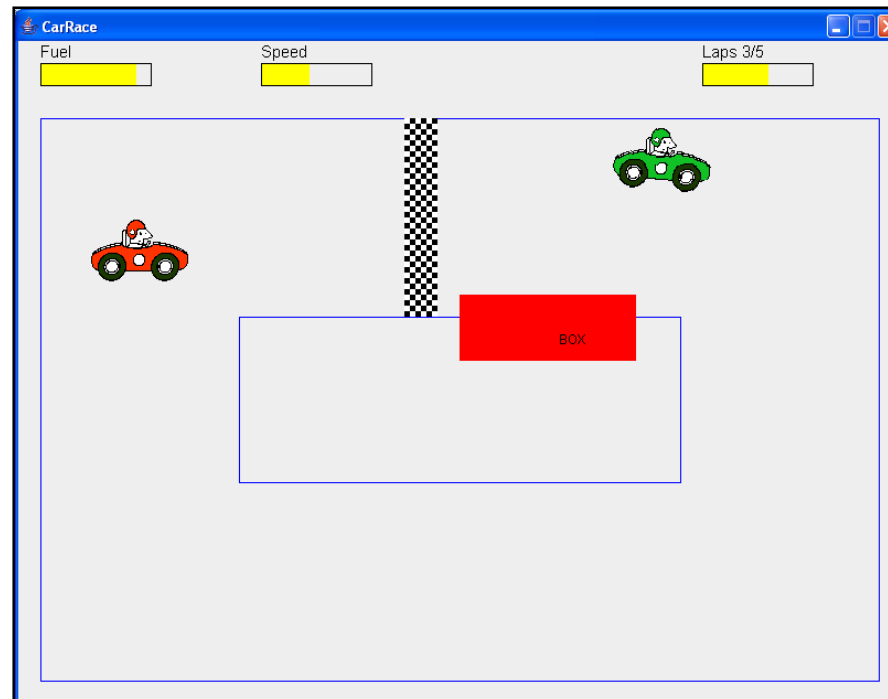


Experiment requirements

- Network application
- Decompiled code
- Code browsing tools
- Debuggers
- Possibility to browse the network
 - API specifications

Kinds of attacks

- Spotting specific functionalities
 - Observable features
- Tampering with the application
 - Make the application do something that is not available in the original code



Experimental design

- 4 groups
 - 2 applications
 - 2 sessions

- 2 hours per session
 - 2 spotting tasks
 - 2 tampering tasks

1 st session	Clear	Obfuscated
App1	G1	G2
App2	G4	G3

2 nd session	Clear	Obfuscated
App1	G3	G4
App2	G2	G1

Preliminary lecture

- Preliminary lecture to make the subjects aware of the experimental environment
 - IDE
 - Obfuscation
 - Debugging facilities
 - Pre questionnaire
 - Informed consent
 - Exercise on an application
 - To practice with the environment and mitigate the learning effect.

Experimental sessions

- 2 experimental sessions
 - Description of the application
 - Either clear or obfuscated source code
 - Possibility to run the (modified) code
 - Four paper sheets (each one contains a task)
 - A post questionnaire

Controlled environment

- All the variables in the experiment are controlled
- By replicating the experiment by modifying just one variable we can study how that variable affect the result
 - Subject experience
 - Obfuscation technique
 - Tool used in the attack
 - Programming language
 - ...



Observations

- Different subjects used different strategies
 - Comments are added to obfuscated methods
 - Identifiers are changed into meaningful names
 - The debugger is used
 - Code is just inspected and never executed
 - Comprehension start from libraries usage (not obfuscated)

Replications

- Trento: master students
 - Software analysis and testing
 - Laboratory of software analysis (TBD)
- Torino: PhD students
 - Software development advanced techniques
- Other?