Software Engineering and Security









April 11th 2017
Dipartimento di Informatica

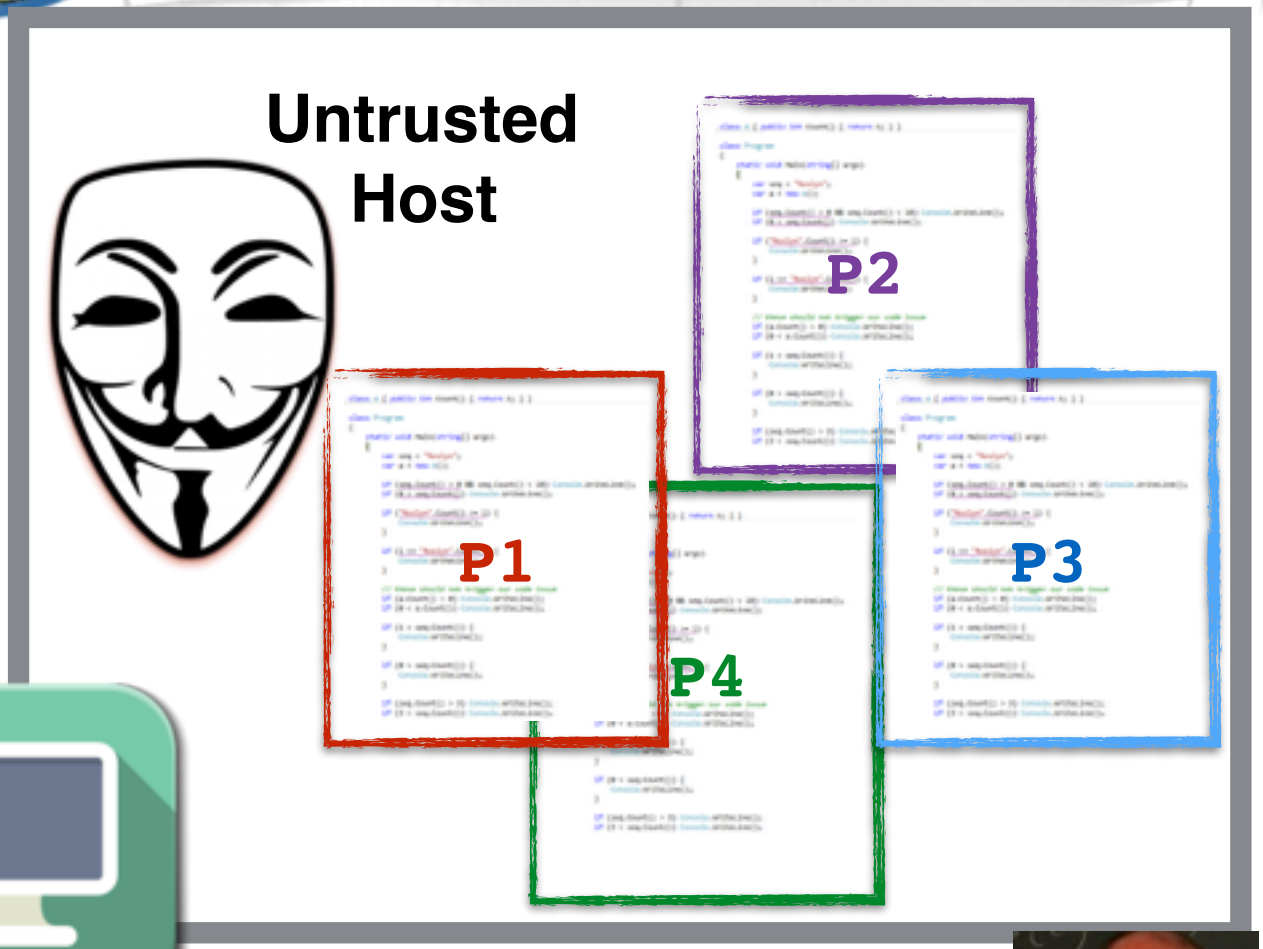
- SOFTWARE ENGINEERING and ANALYSIS

  Study and design of methodologies that support the development and maintenance of complex software systems

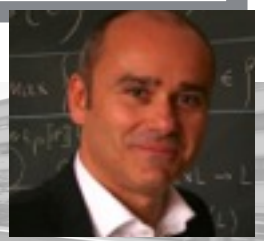
- SECURITY AND PRIVACY
  - malware (e.g., virus, trojans, backdoors…)
  - disclosure of sensitive data in the internet
  - network security (protocols)
  - web security (code injection, cross site scripting)
  - intellectual property protection (software piracy)
  - software integrity
  - …

Security
SW Protection and Malware

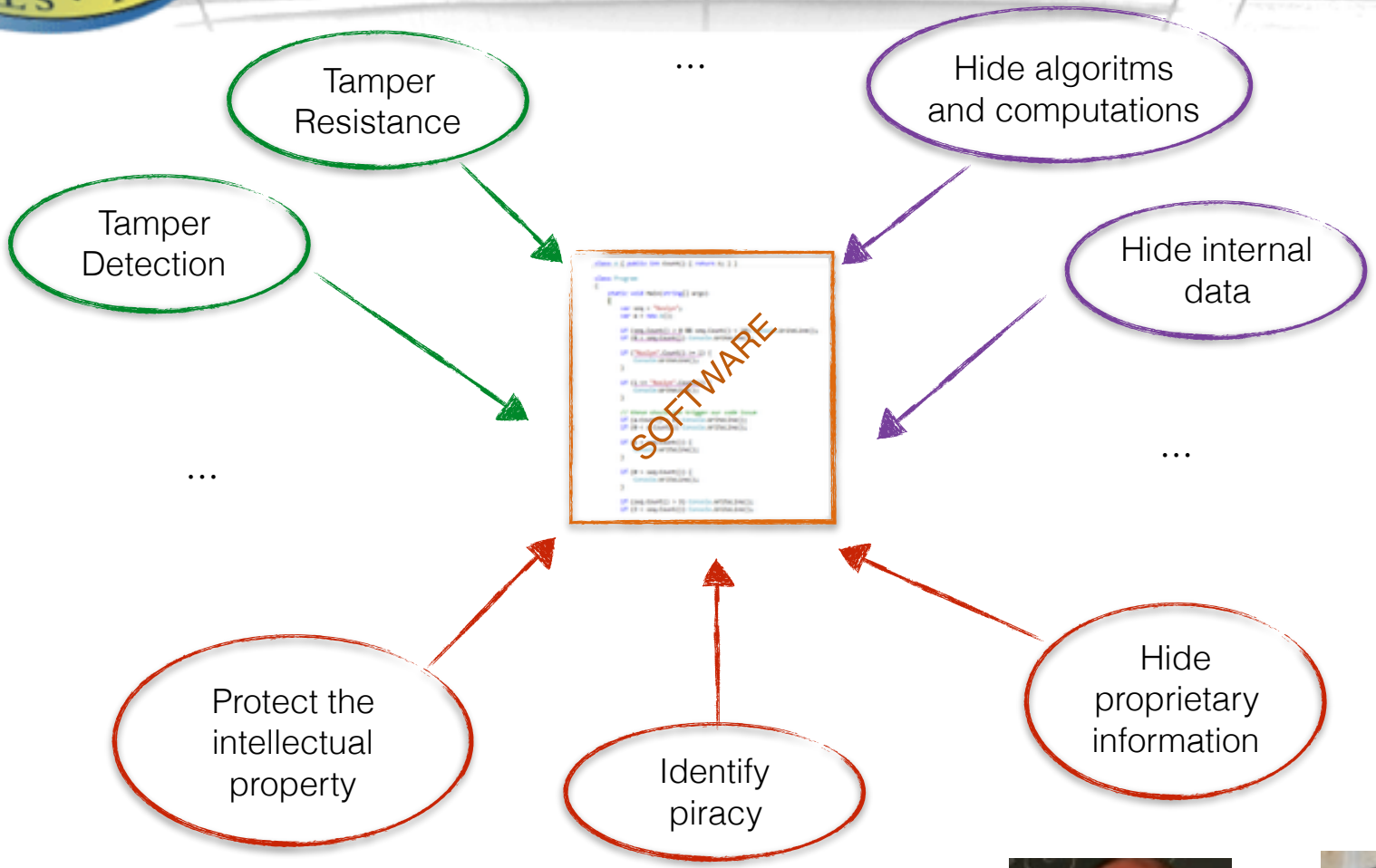
SW Protection



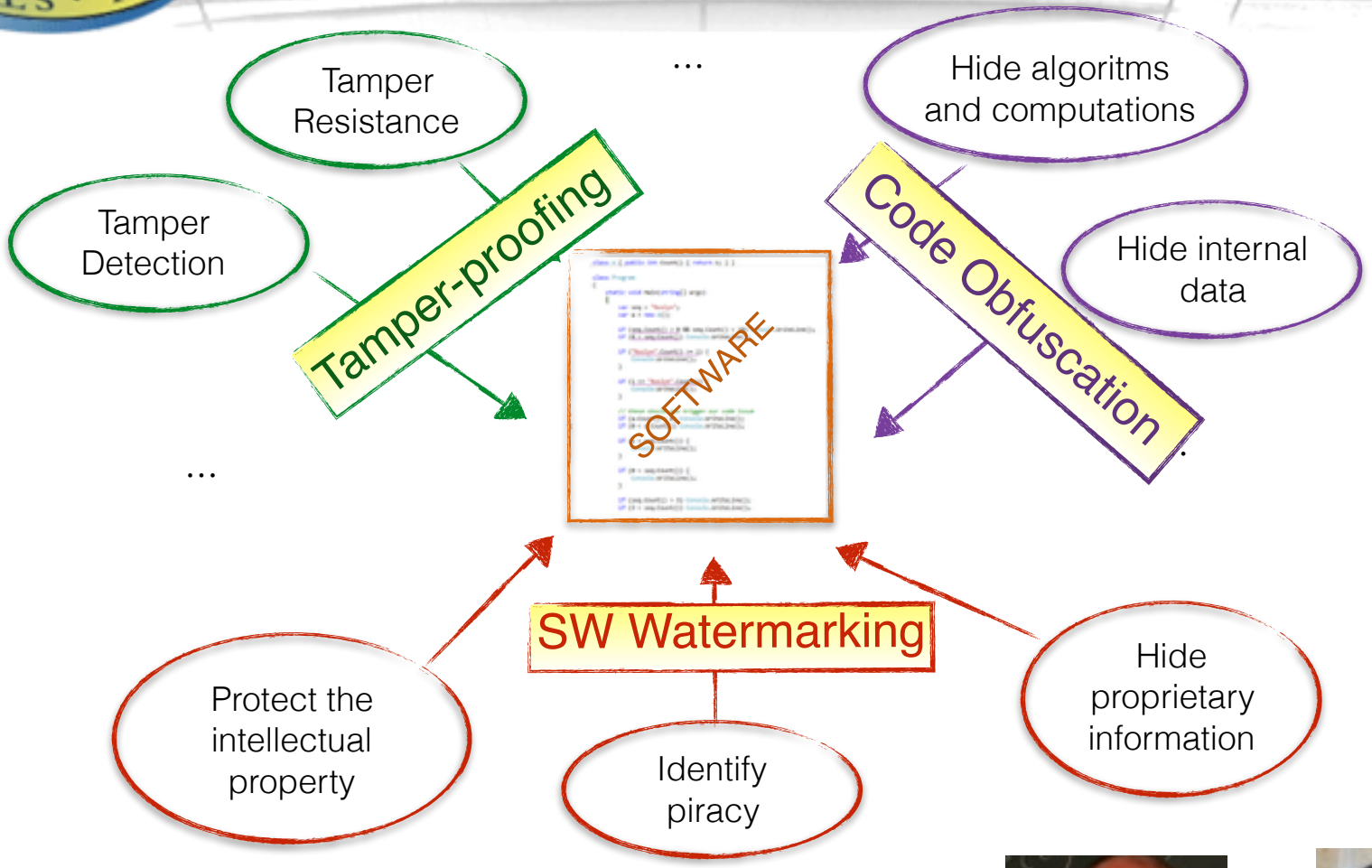
MATE attack

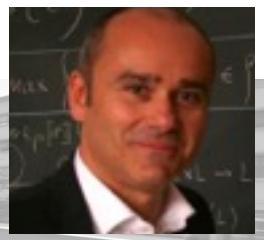


The Value of SW Protection

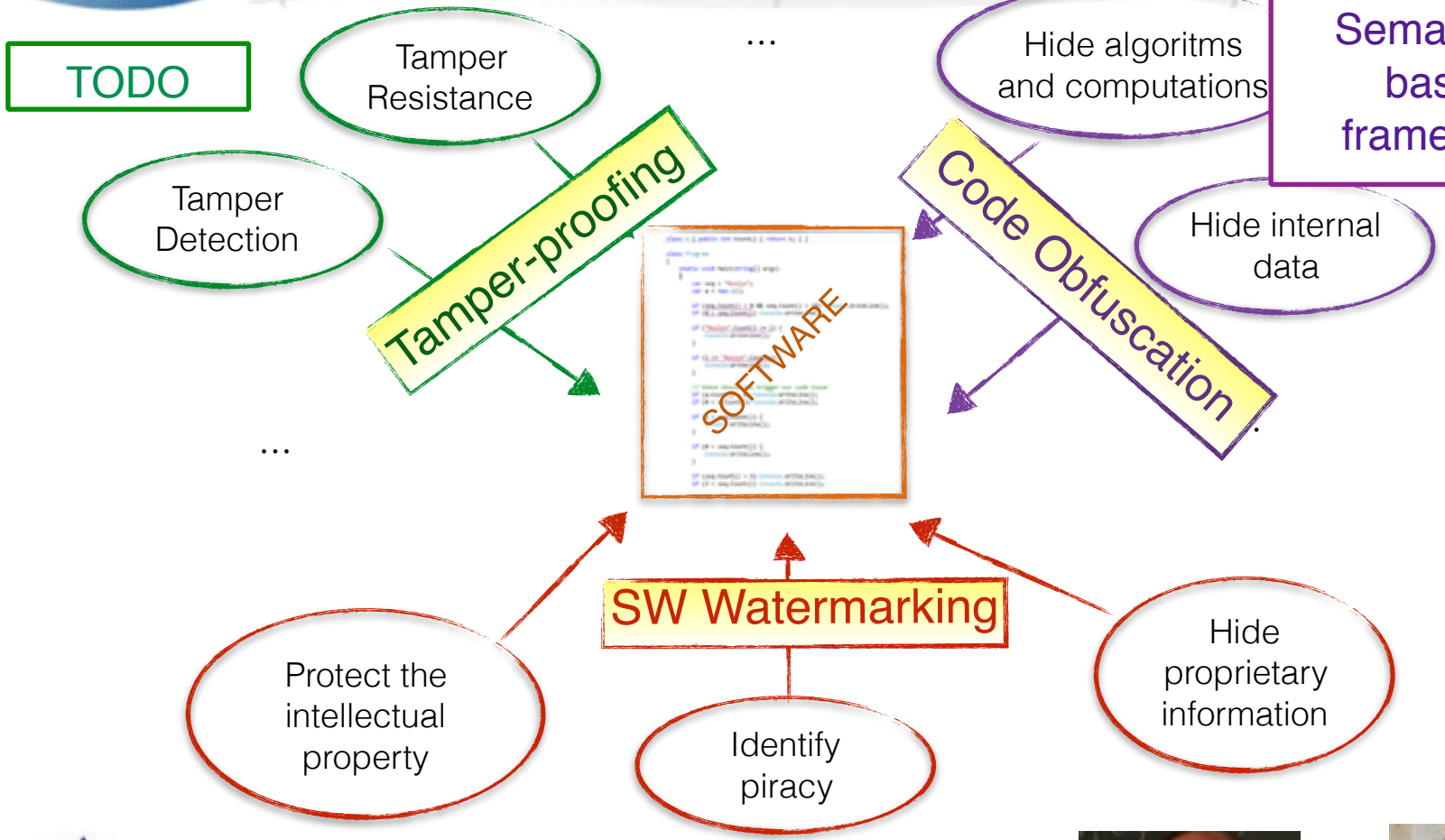


SW Protection Techniques



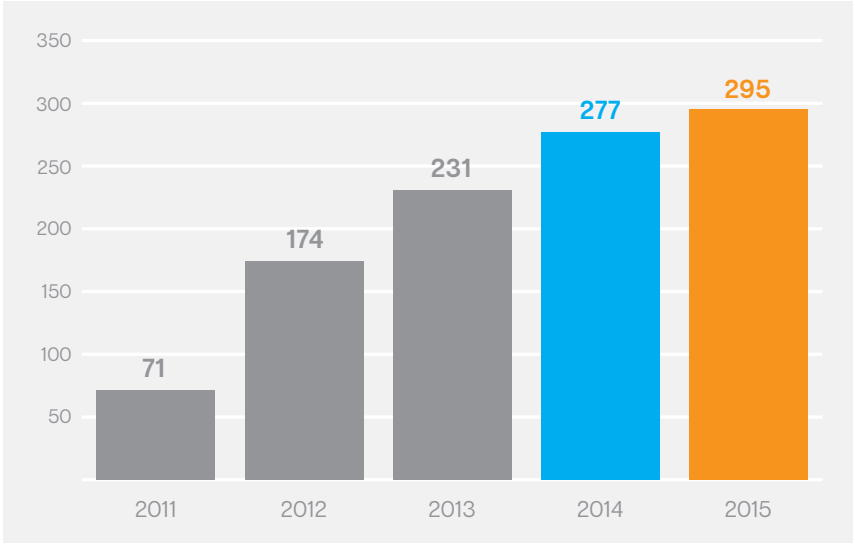


SW Protection Techniques



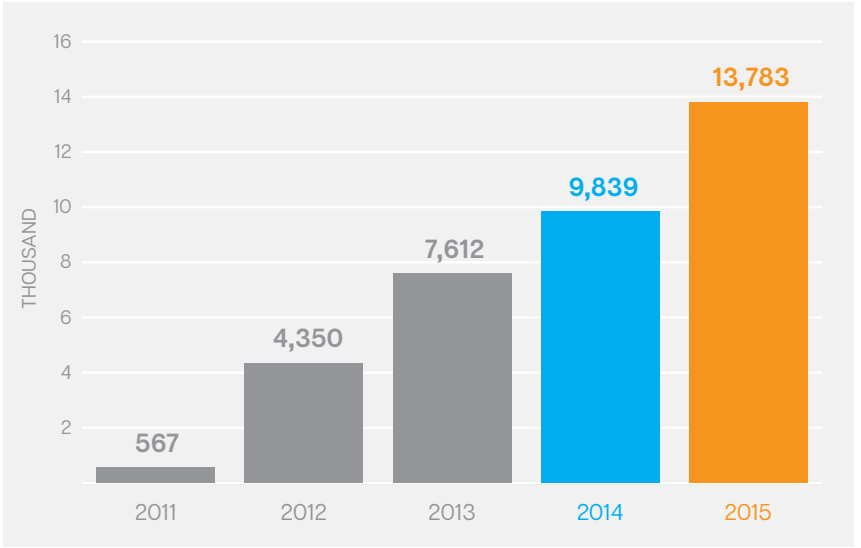
Malware & Malware Variants

Cumulative Android Mobile Malware



the number of Android **malware families** added in 2015 grew by **6%**, compared with the 20% growth in 2014

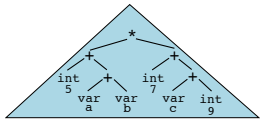
Cumulative Android Mobile Variants

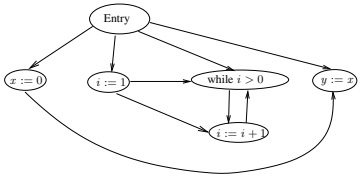


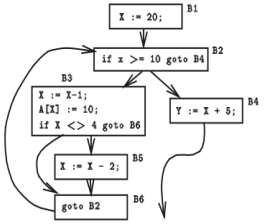
the number of Android **malware variants** added in 2015 grew by **40%,** compared with the 29% growth in 2014

[Symantec 2016]

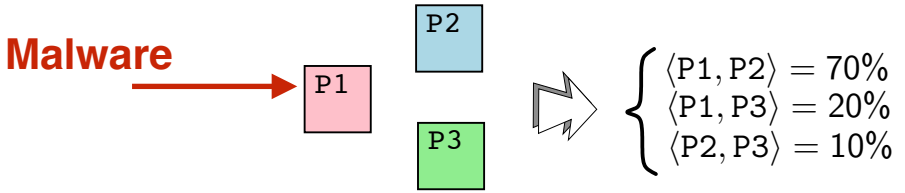
Similarity Analysis



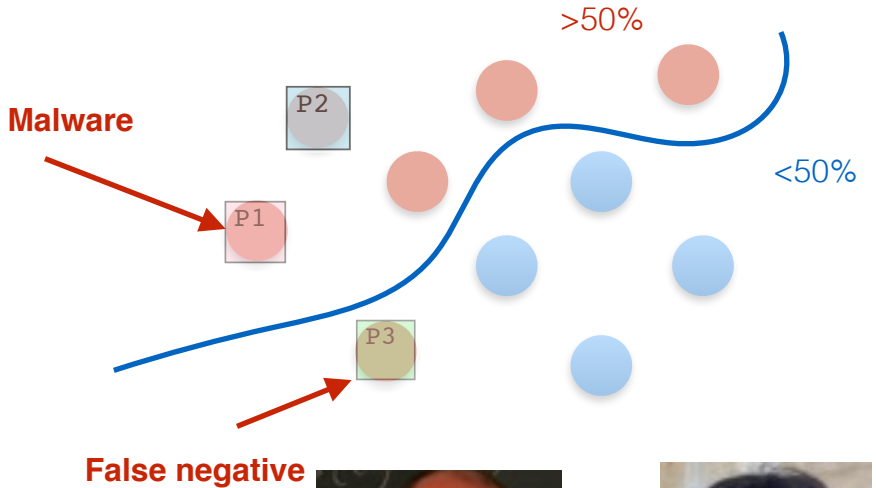




Mostly syntactic in nature (AST, PDF and CFG)



- clone detection
- software forensics
- plagiarism detection
- tamper detection
- software birth-marking
- malware detection
- vulnerability detection

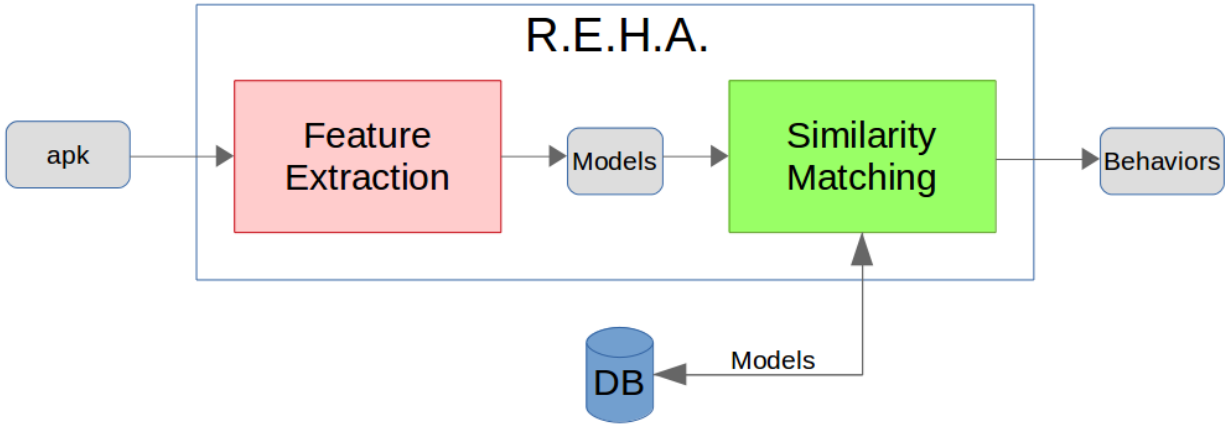


Semantic Similarity

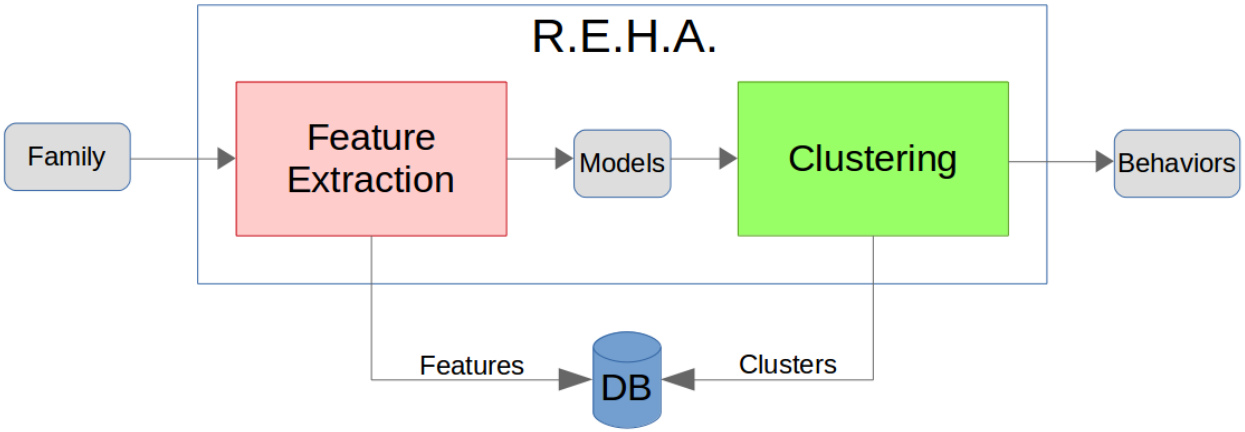
In order to identify malware variants, plagiarized code, tampered code and vulnerabilities in different application, we need to extract semantic models!

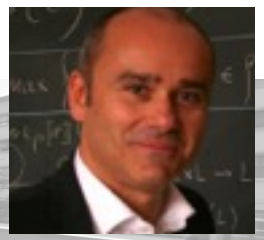
Develop a semantics-based similarity analysis!

Similarity Analysis



Similarity Analysis





Analysis

Analysis of Concurrent SW

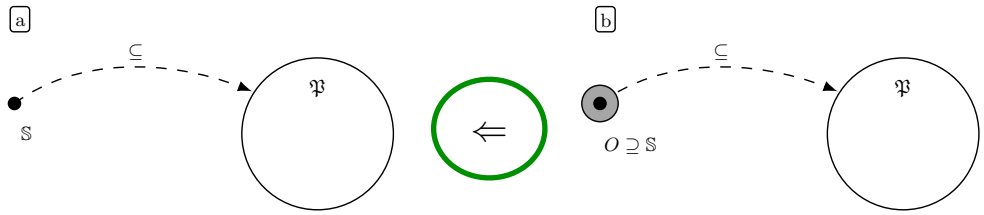
Race Condition

- Static analysis for the detection of race condition on Concurrent Java programs.
- Verification and inference of locking policies (@GuardedBy and @Holding annotations) and relative formal semantics
- Verification and inference of thread-confinement properties (@UiThreda and @WorkerThread annotations)

Hyperproperty Verification

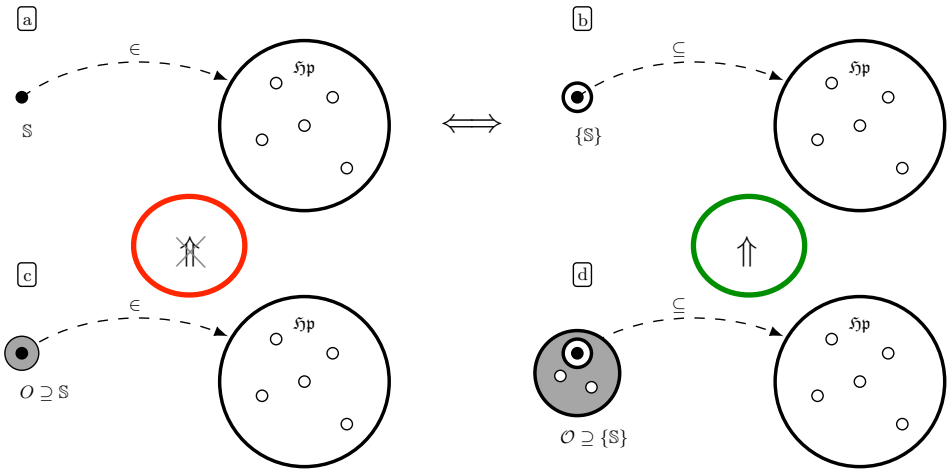
Hyperproperty:

Property that can be verified on sets of execution instead of on single executions.



verifying properties

verifying ***hyperproperties***

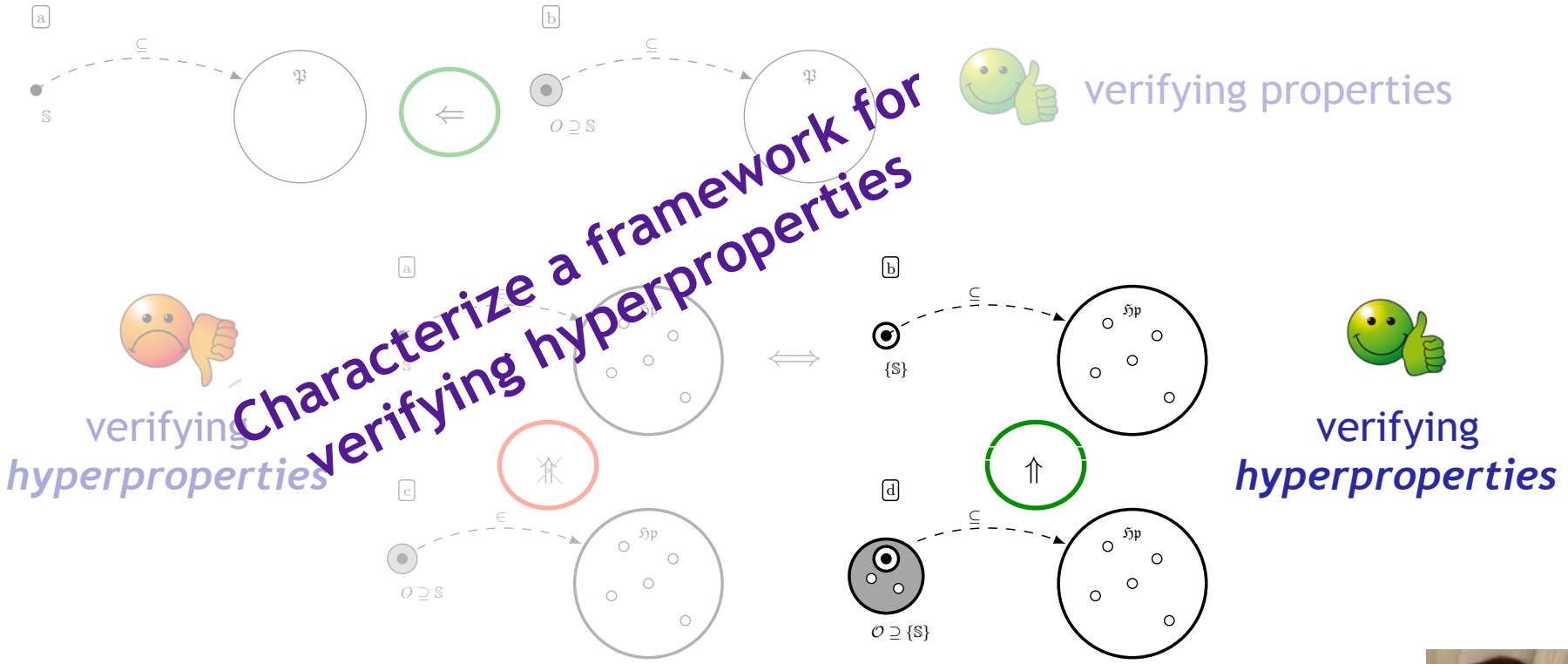


verifying ***hyperproperties***

Hyperproperty Verification

Hyperproperty:

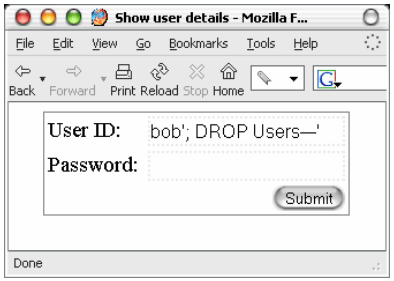
Property that can be verified on sets of execution instead of on single executions.



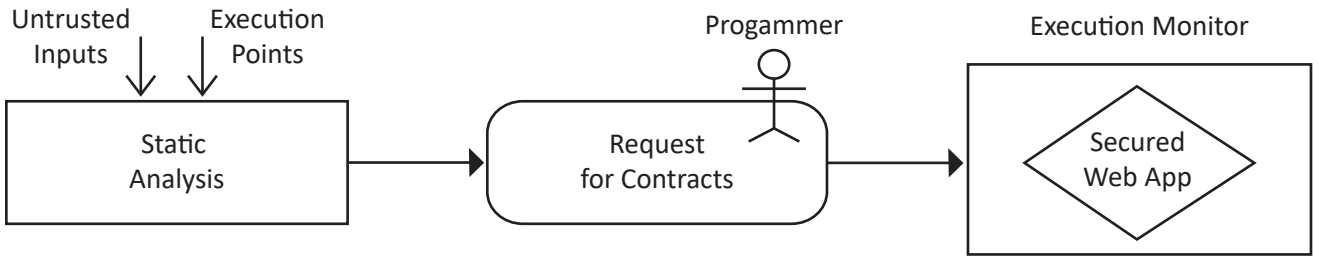
SQL injection

SQL injection example

```
query = "SELECT Username, UserID, Password
         FROM Users WHERE
         Username = 'bob'; DROP Users--
         'AND Password = ' '"
```



Our idea for releasing *secured* applications

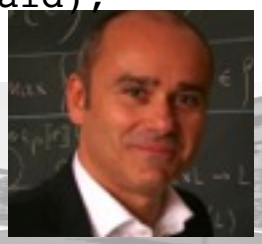


Static Analysis of Dynamic Code

JavaScript

```
// Retrive th ID for a camera
int cameraId = …;

// Create an obfuscated string
//containing the method call
String obfuscated = “AoApBeBnBA”;
String deobfuscated = obfuscated.replaceAll(“[AB]”,””);

Class<?> klass = Class.forName(
  “android.hardware.Camera”
);

// Retrive and invoke the method
Method method = klass.getMethod(
  deobfuscated,
  Integer.class
);

Camera camera = (Camera) method.invoke(cameraId);
```

```
x:= …
…
manipulation of x
…
Eval (x)
```



Thanks