



## Advanced Software Protection: Integration, Research, Exploitation



Bjorn De Sutter  
[bjorn.desutter@ugent.be](mailto:bjorn.desutter@ugent.be)

18 Oct 2016

Belgian OWASP Chapter Meeting

# Aspire in a nutshell

2

NAGRA

SafeNet

gemalto  
security to be free



Aspire : Advanced Software Protection: Integration, Research and Exploitation

# Man-At-The-End (MATE) Attacks

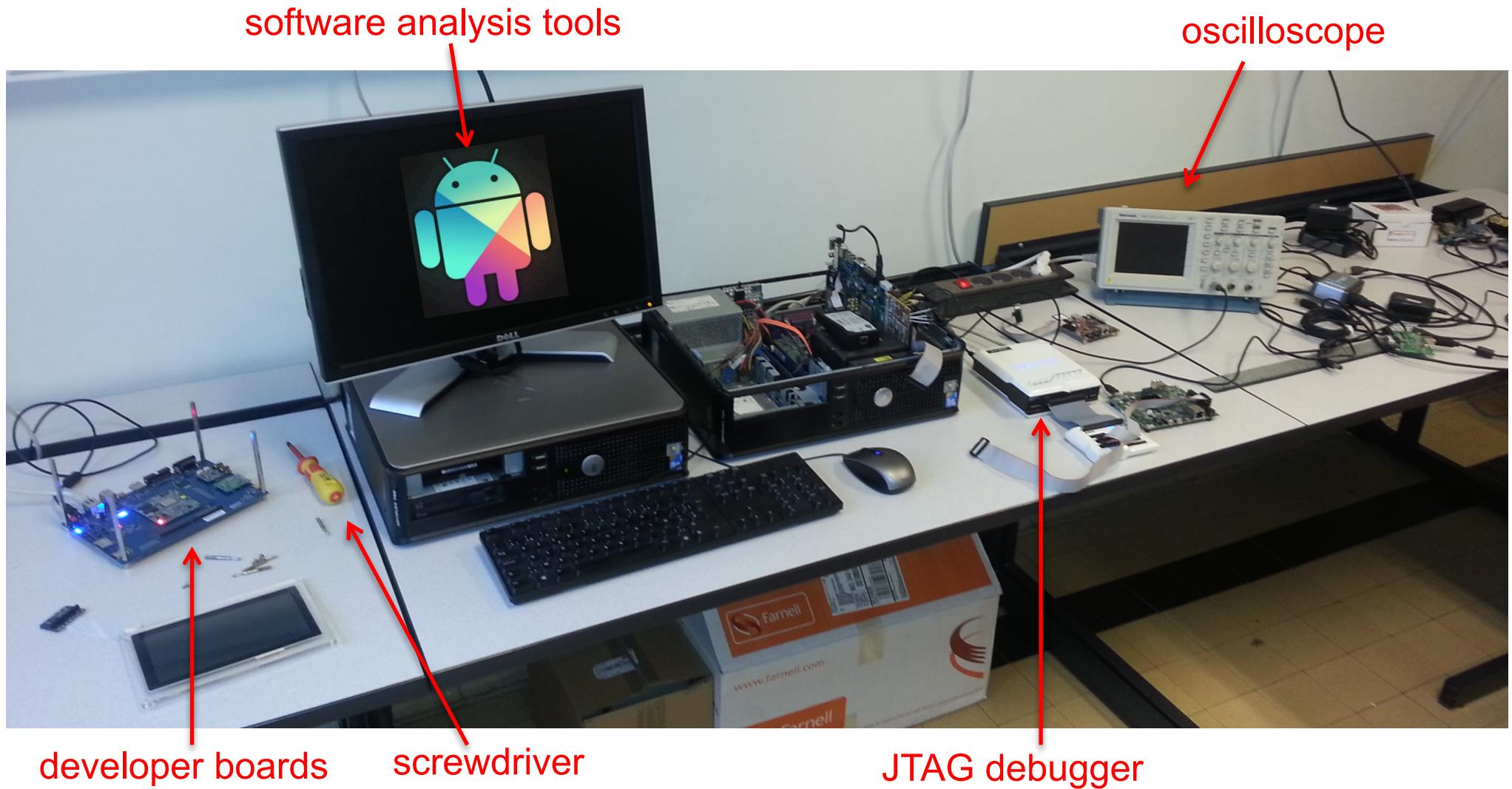
3



Aspire : Advanced Software Protection: Integration, Research and Exploitation

# Man-At-The-End (MATE) Attacks

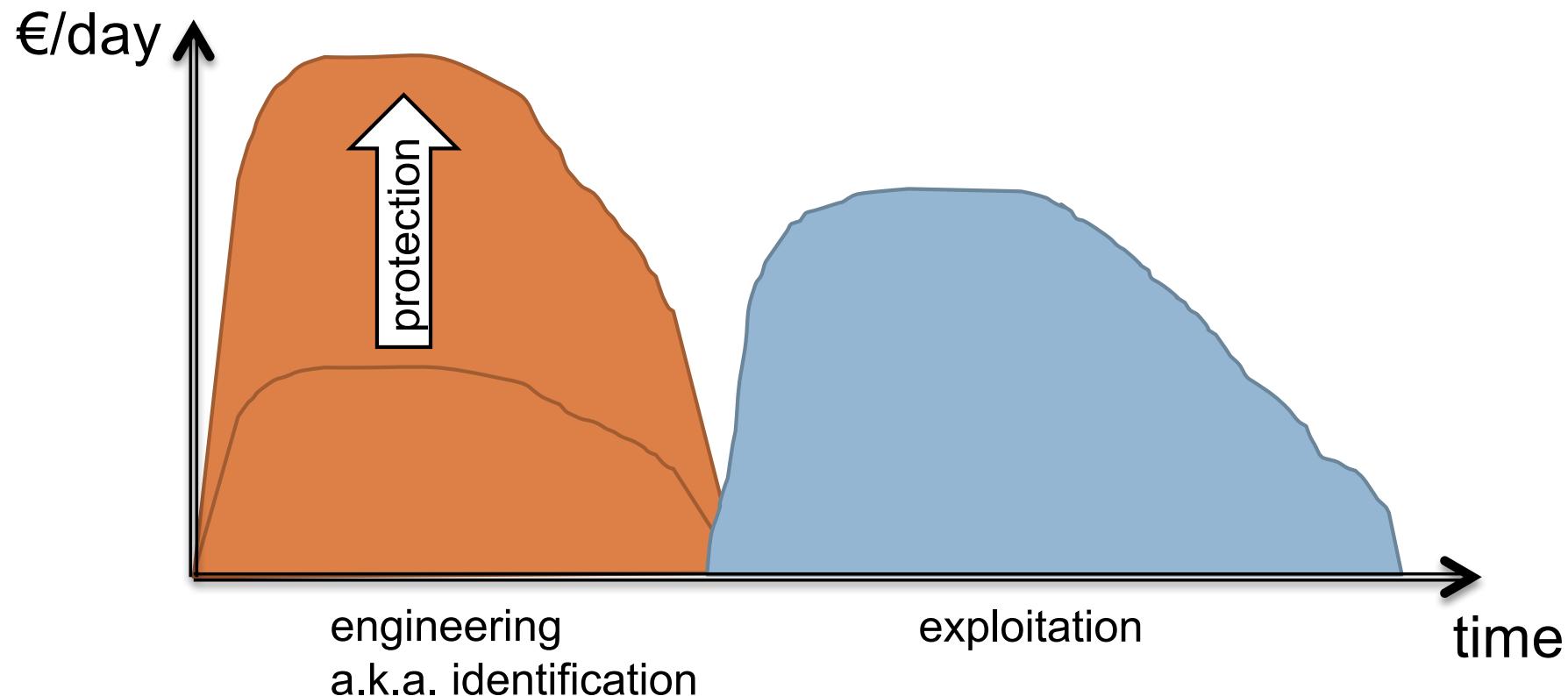
4



Aspire : Advanced Software Protection: Integration, Research and Exploitation

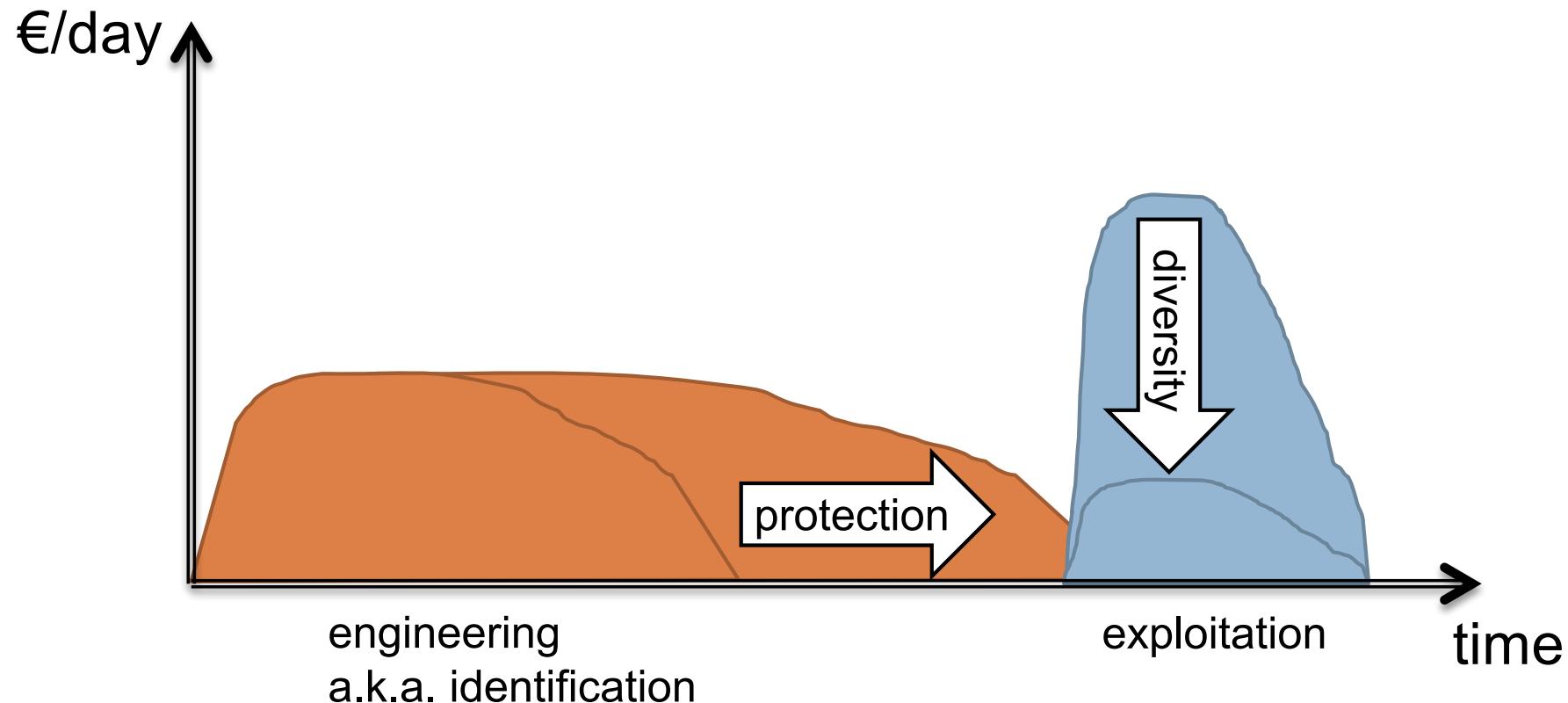
# Economics of MATE attacks

5



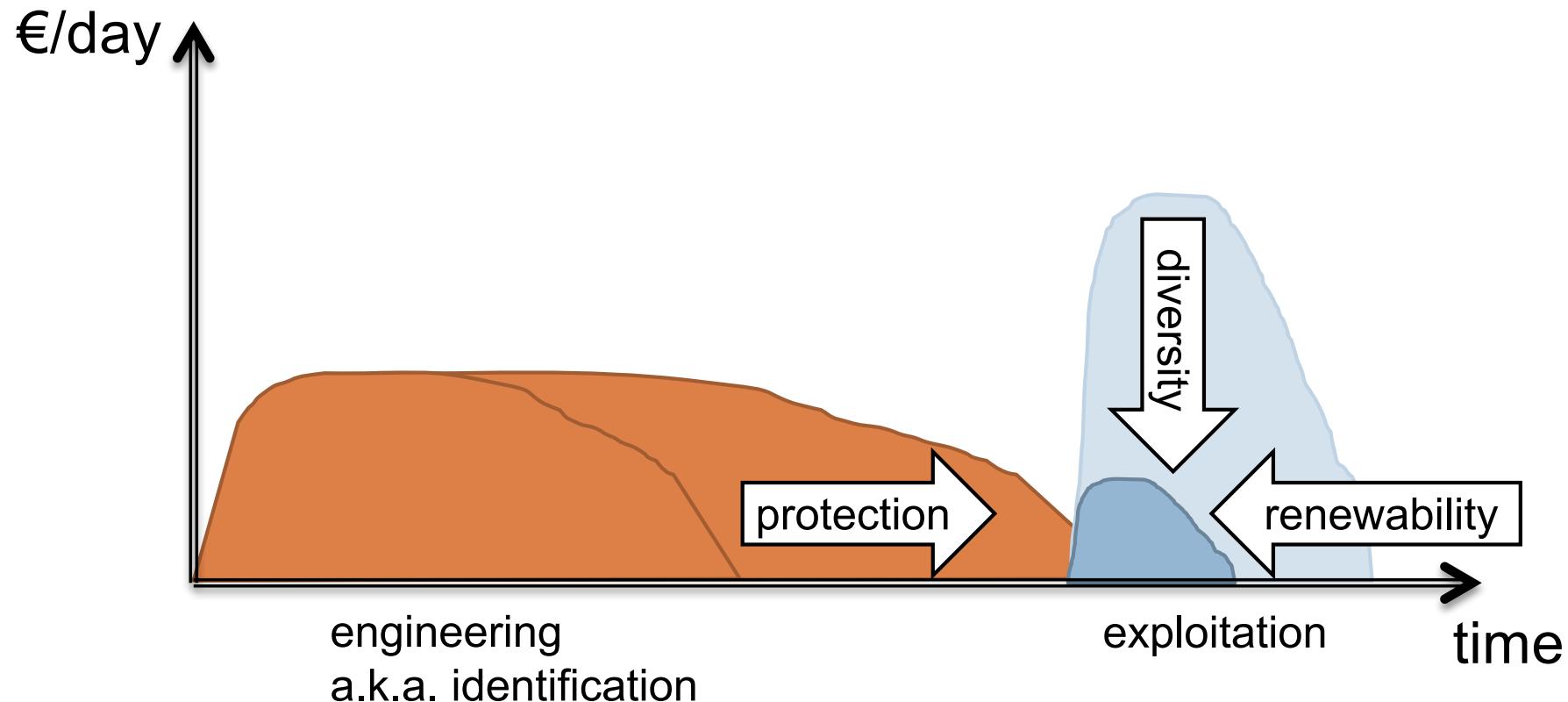
# Economics of MATE attacks

6



# Economics of MATE attacks

7



# Assets and security requirements

8

Asset category	Security Requirements	Examples of threats
<b>Private data</b> (keys, credentials, tokens, private info)	Confidentiality Privacy Integrity	Impersonation, illegitimate authorization Leaking sensitive data Forging licenses
<b>Public data</b> (keys, service info)	Integrity	Forging licenses
<b>Unique data</b> (tokens, keys, used IDs)	Confidentiality Integrity	Impersonation Service disruption, illegitimate access
<b>Global data</b> (crypto & app bootstrap keys)	Confidentiality Integrity	Build emulators Circumvent authentication verification
<b>Traceable data/code</b> (Watermarks, finger-prints, traceable keys)	Non-repudiation	Make identification impossible
<b>Code</b> (algorithms, protocols, security libs)	Confidentiality	Reverse engineering
<b>Application execution</b> (license checks & limitations, authentication & integrity verification, protocols)	Execution correctness Integrity	Circumvent security features (DRM) Out-of-context use, violating license terms

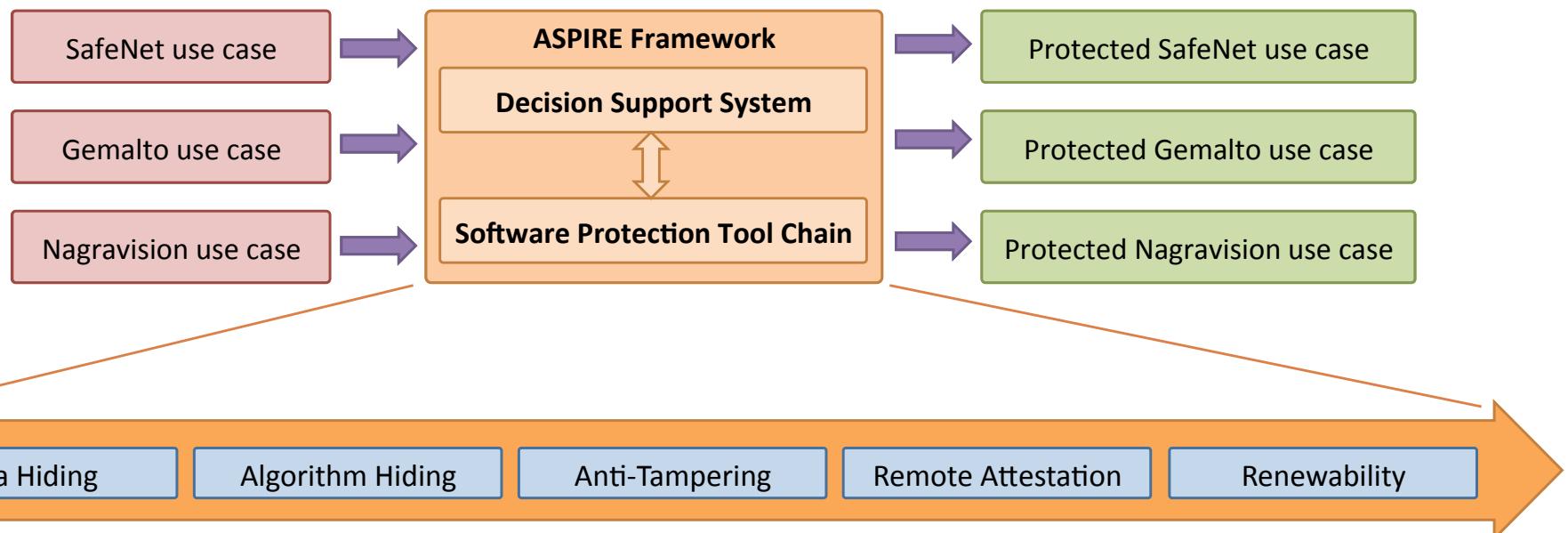
# Aspire in a nutshell

9

NAGRA

SafeNet

gemalto  
security to be free

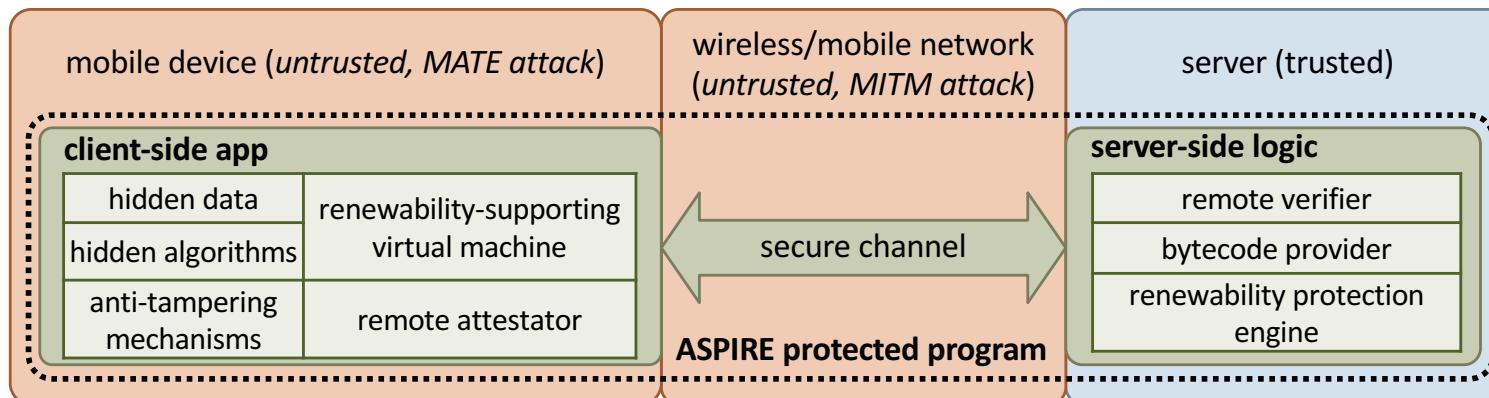


Aspire : Advanced Software Protection: Integration, Research and Exploitation

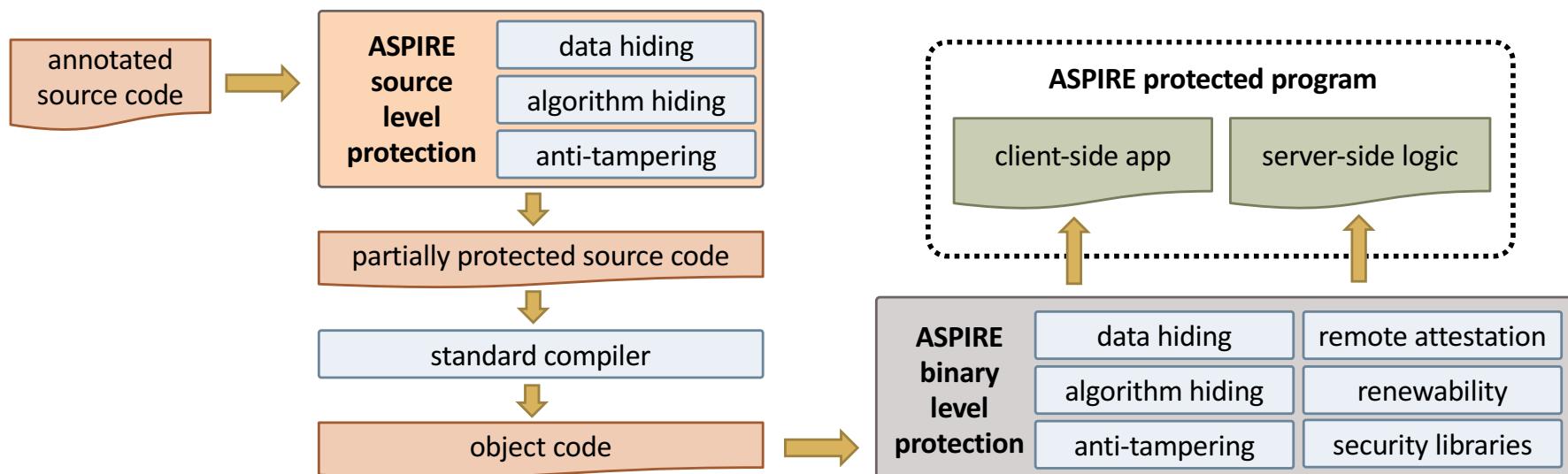
# Aspire Goals

10

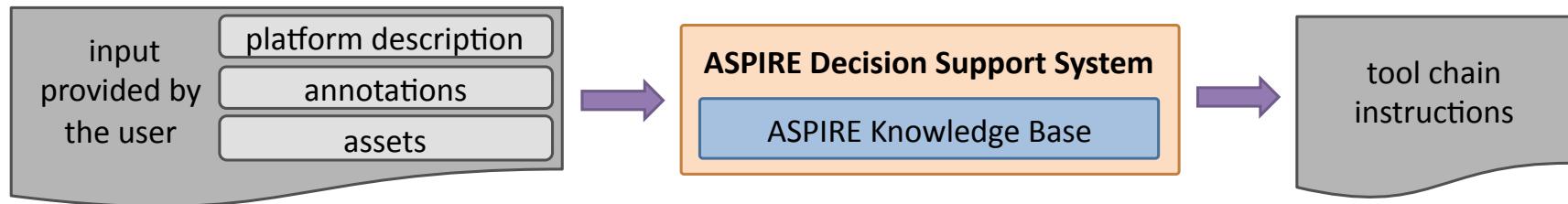
## 1. Reference architecture for protected mobile services



## 2. Software protection techniques and integrated plugin-based tool flow

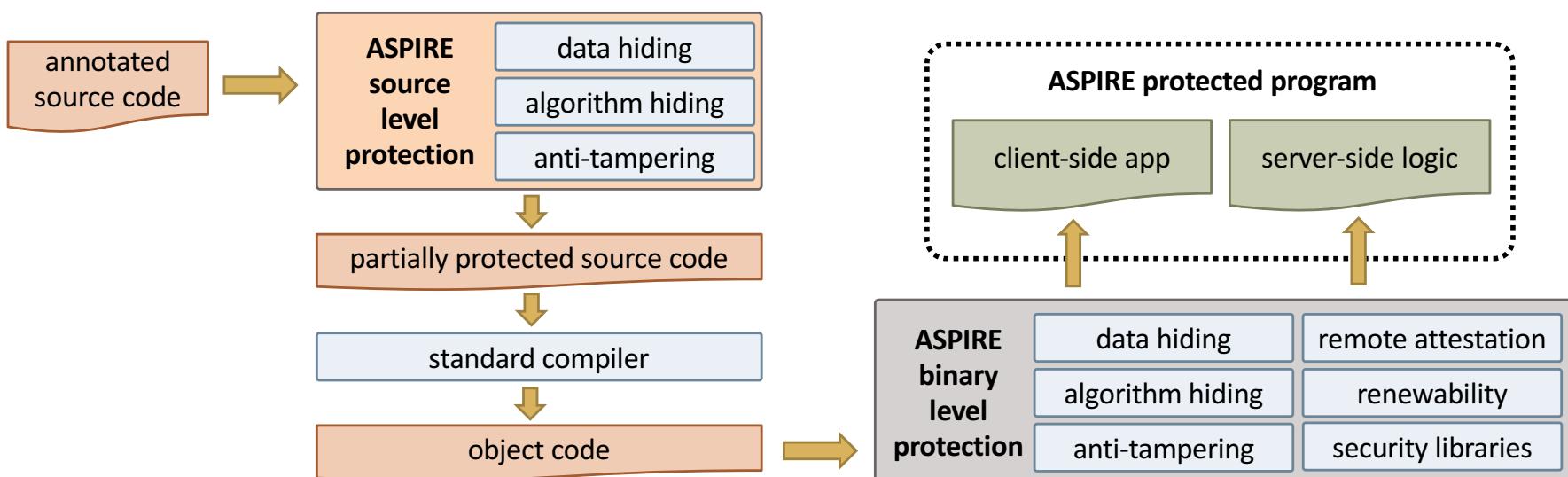


### 3. Decision Support System



- attack models & evaluation methodology
- security metrics
- experiments with human subjects
- public challenge

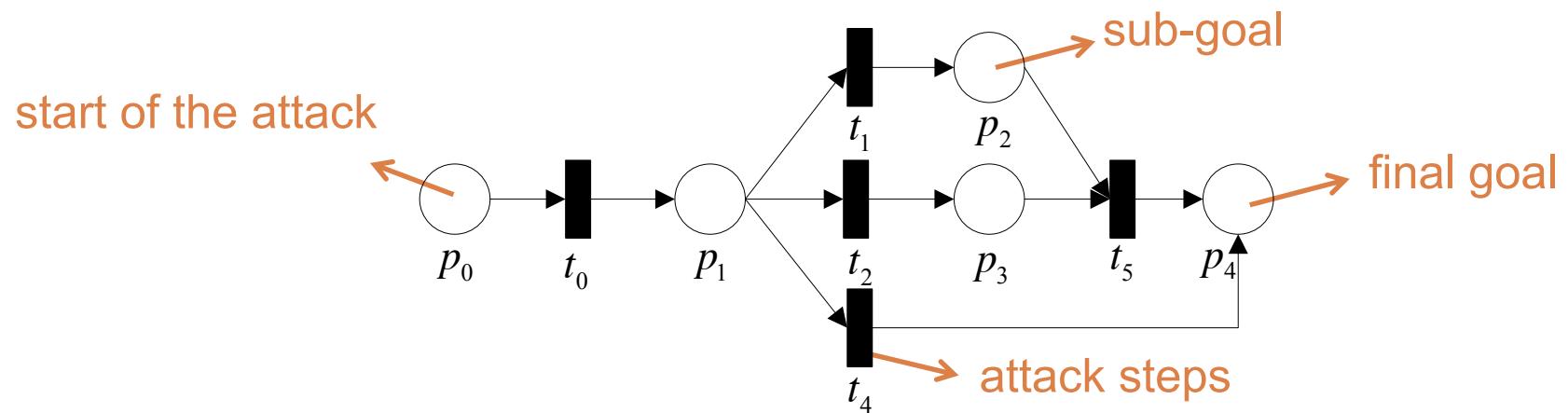
### 2. Software protection techniques and integrated plugin-based tool flow



# Part 1: Reference Architecture

12

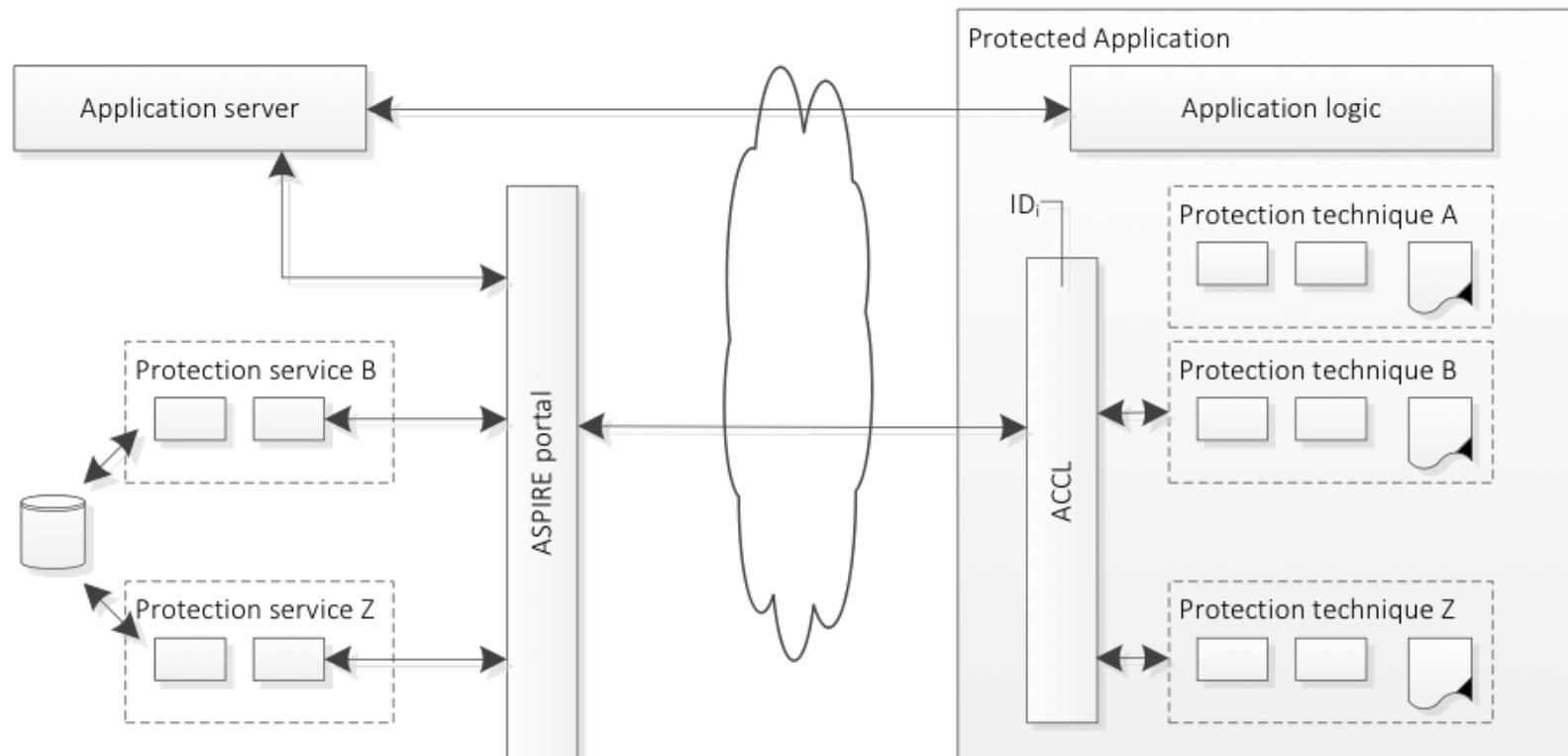
- Cookbook for combining protections
- Why?



# Part 1: Reference Architecture

13

- How to combine multiple protections?
  - How do the individual protections actually work?

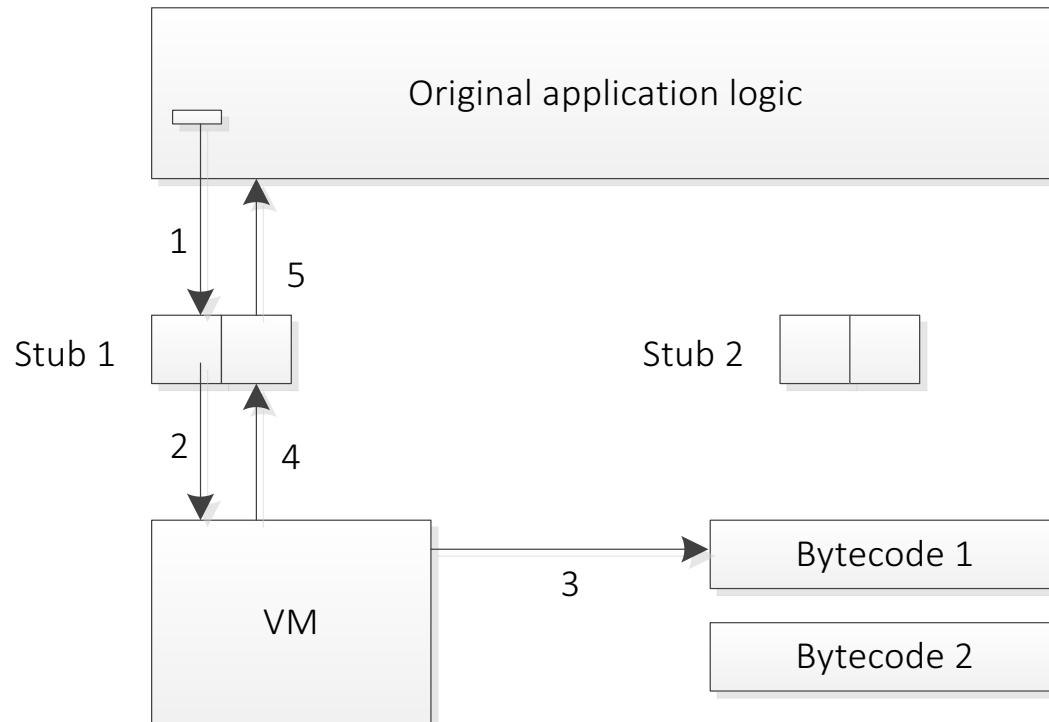


Aspire : Advanced Software Protection: Integration, Research and Exploitation

# Part 1: Reference Architecture

14

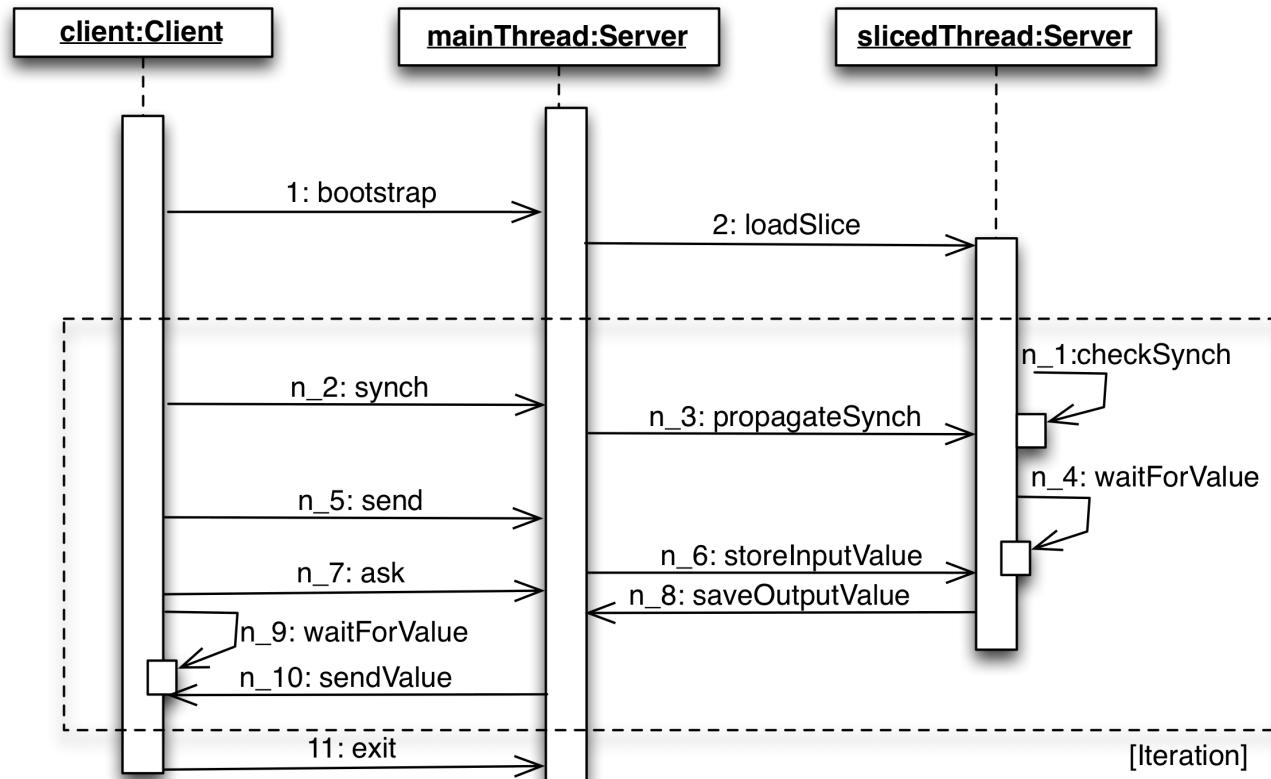
- How to combine multiple protections?
  - How do the individual protections actually work?



# Part 1: Reference Architecture

15

- How to combine multiple protections?
  - How do the individual protections actually work?



# Part 1: Reference Architecture

16

- How to combine multiple protections?
  - How do the individual protections actually work?



- data obfuscations
- white box cryptography (static keys, dynamic keys, time-limited)
- diversified crypto libraries



# Part 1: Reference Architecture

17

- How to combine multiple protections?
  - How do the individual protections actually work?



- control flow obfuscations
- multithreaded crypto
- instruction set virtualization
- code mobility
- self-debugging
- client-server code splitting



# Part 1: Reference Architecture

18

- How to combine multiple protections?
  - How do the individual protections actually work?



- code guards
- static and dynamic remote attestation
- reaction mechanisms
- client-server code splitting



# Part 1: Reference Architecture

19

- How to combine multiple protections?
  - How do the individual protections actually work?



- native code diversification □
- bytecode diversification □
- renewable white-box crypto □
- mobile code diversification □
- renewable remote attestation □



# Part 1: Reference Architecture

20

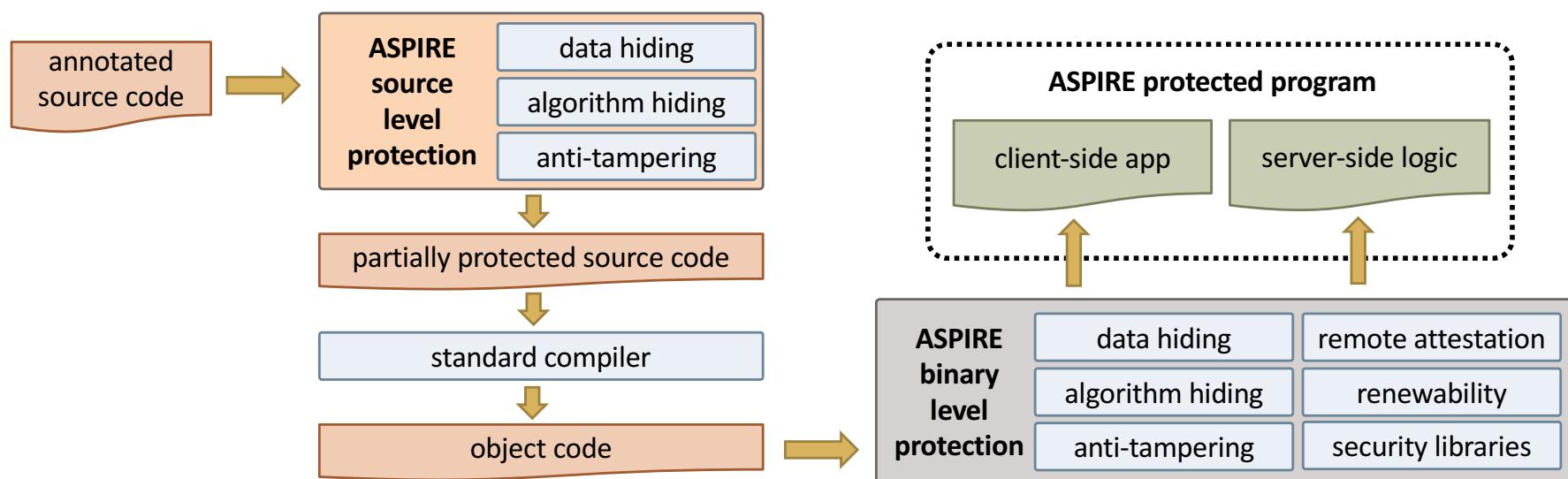
- How to combine multiple protections?
  - How do the individual protections actually work?
  - How do the protections compose?
  - Do the protections share components?
  - If protections compose, are there phase-ordering issues?
  - Which protections/components need to be combined and how?
  - Where is  $1 + 1 > 2$  in terms of protection strength?
  - What is the combined impact on software development life cycle?



# Part 2: ASPIRE Compiler Tool Chain

21

## 2. Software protection techniques and integrated **plugin-based** tool flow



- Python – Dolt compiler flow
- JSON configuration scripts
- invokes chain of +/- independent tools
- TXL source code rewriting
- Diablo link-time binary rewriting



Aspire : Advanced Software Protection: Integration, Research and Exploitation

# Source code annotations

22

```
static const char ciphertext[] __attribute__  
((ASPIRE("protection(wbc,label(ExampleFixed),role(input),size(16))")))  
= { 0x00, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07,  
    0x08, 0x09, 0x0a, 0x0b, 0x0c, 0x0d, 0x0e, 0x0f };
```

```
static const char key[] __attribute__  
((ASPIRE("protection(wbc,label(ExampleFixed),role(key),size(16))")))  
= { 0x00, 0x11, 0x22, 0x33, 0x44, 0x55, 0x66, 0x77,  
    0x88, 0x99, 0xaa, 0xbb, 0xcc, 0xdd, 0xee, 0xff };
```

```
char plaintext[16] __attribute__  
((ASPIRE("protection(wbc,label(ExampleFixed),role(output),size(16))")))  
;
```

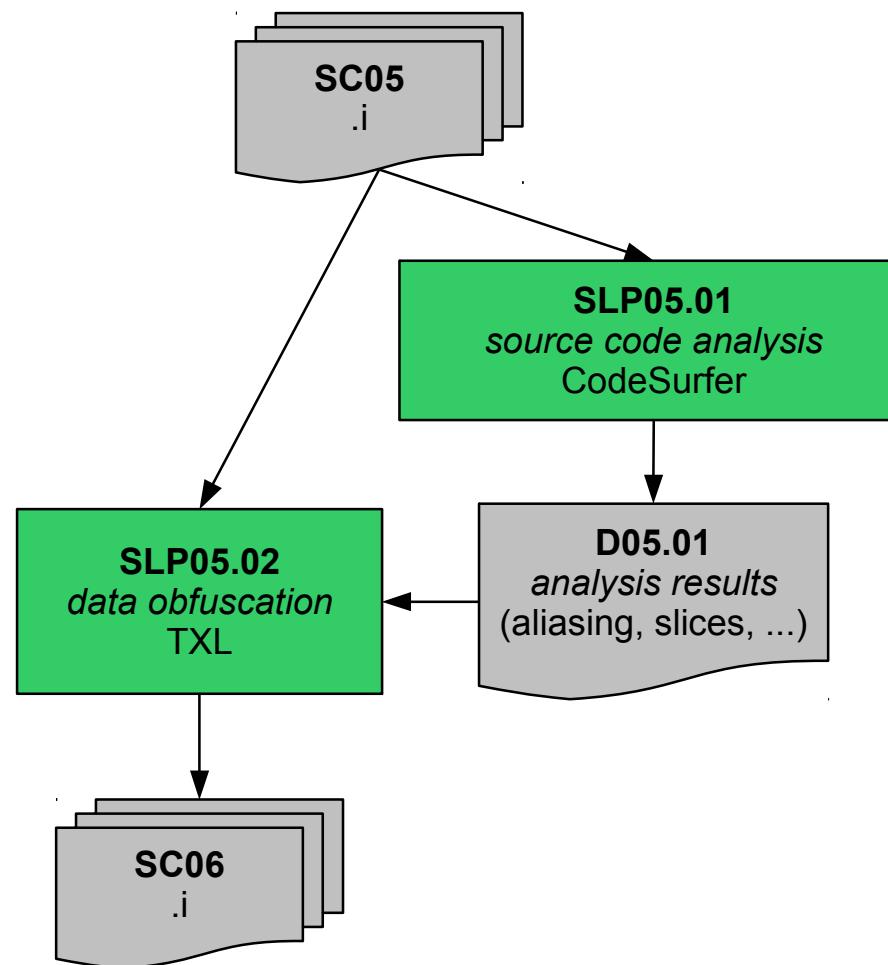
```
_Pragma ("ASPIRE begin protection(wbc,label(ExampleFixed),algorithm(aes),mode(ECB),operation(decrypt))")  
decrypt_aes_128(ciphertext, plaintext, key);  
_Pragma("ASPIRE end");
```



Aspire : Advanced Software Protection: Integration, Research and Exploitation

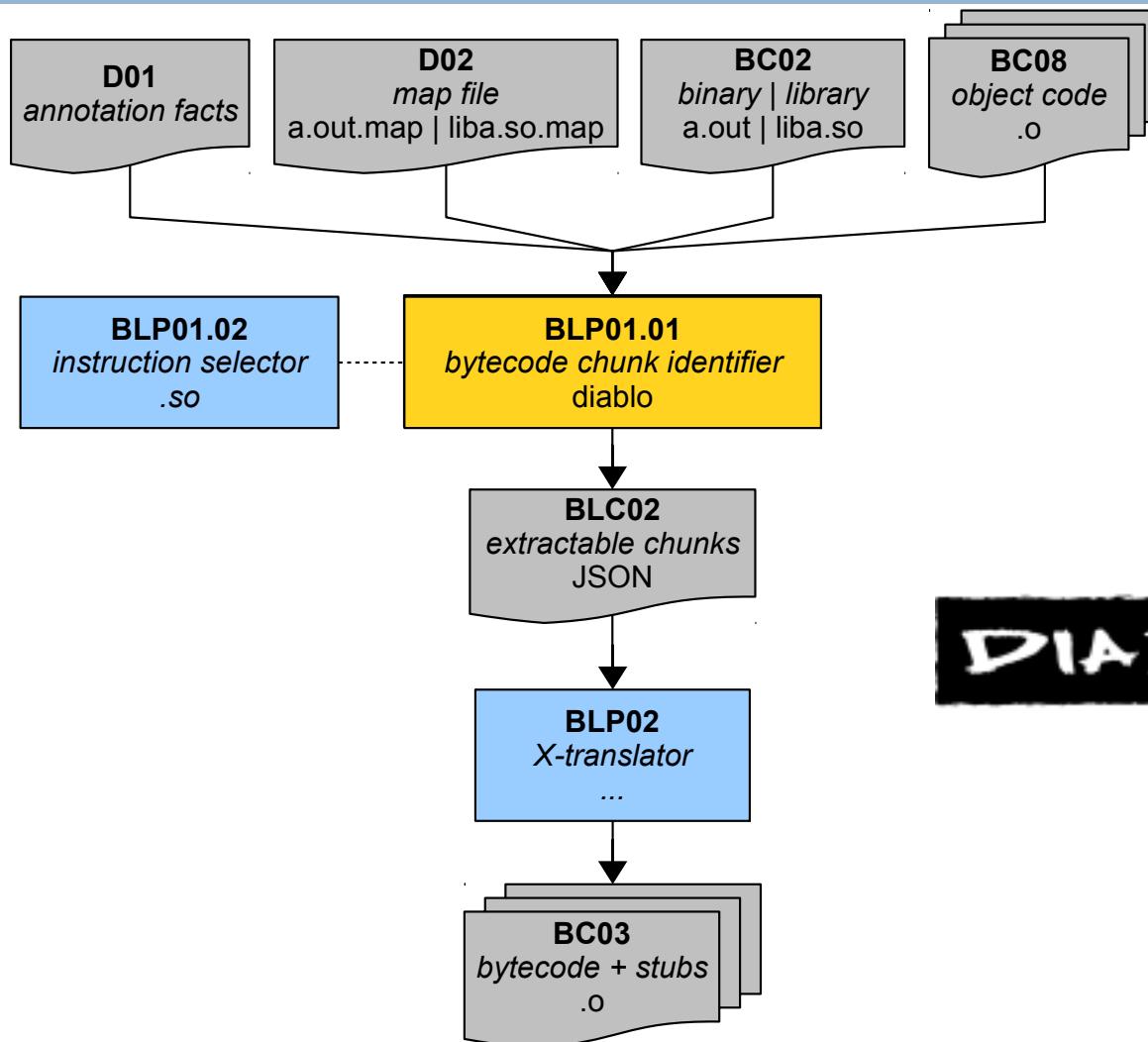
# Source Code rewriting

23



# Binary Code Rewriting

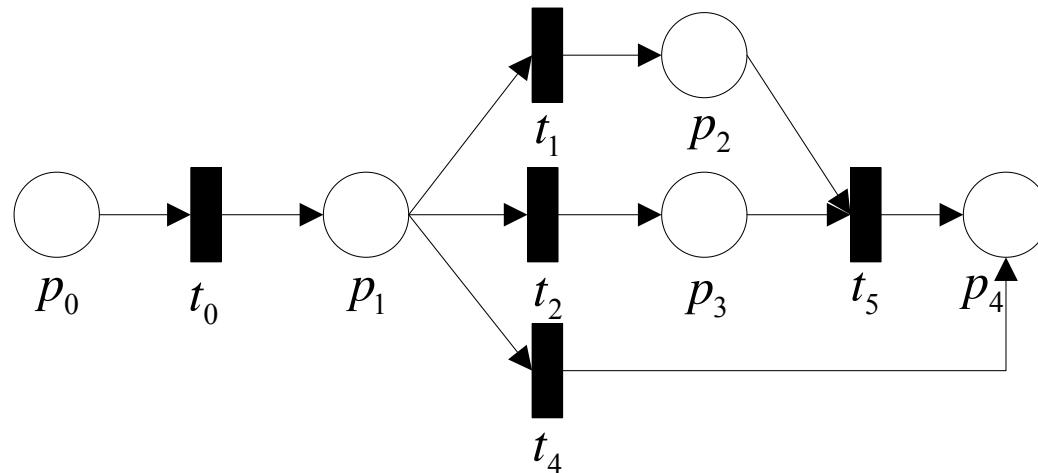
24



Aspire : Advanced Software Protection: Integration, Research and Exploitation

# Part 3: Decision Support

25



- Knowledge Base
- Complexity & Resilience Metrics
- Protection Strength Evaluation Methodology
- Optimization strategies



# Validation & Demonstration

26

- three real-world use cases
  - software license manager
  - one-time password generator
  - DRM protection
- security requirements from industry
  - functional requirements
  - non-functional requirements
  - assurance requirements
- dynamically linked Android 4.4 – ARMv7 libraries
- penetration tests by professional pen testers



Aspire

: Advanced Software Protection: Integration, Research and Exploitation

# Validation & Demonstration

27

- controlled experiments with academic hackers
- public challenge and bounties

The screenshot shows a web browser window for the ASPIRE Public Challenge. The address bar displays the URL <https://bounty.aspire-fp7.eu>. The page header includes links for Home, Terms, Rankings, and Contact. On the right side, there is a logo for "Aspire" with a blue speech bubble icon. The main content area features a heading "ASPIRE Public Challenge". Below it, a paragraph describes the challenge: "The ASPIRE Public Challenge invites everyone to break some of the software protections developed and implemented in the [ASPIRE Project](#). There are in total eight different challenges for the ARM platform (both GNU/Linux and Android), each of which contain a key check. It is your goal to extract this key. For every one of these challenges, the first successful attack is eligible for a prize of €200 (see the [terms and conditions](#) for more information).". To the right of this text are two forms: "Join the challenge!" and "Already have an account?". The "Join the challenge!" form contains fields for Username\*, Email\*, and First name\*. The "Already have an account?" form contains a Log in button.

Home Terms Rankings Contact

## ASPIRE Public Challenge

The ASPIRE Public Challenge invites everyone to break some of the software protections developed and implemented in the [ASPIRE Project](#). There are in total eight different challenges for the ARM platform (both GNU/Linux and Android), each of which contain a key check. It is your goal to extract this key. For every one of these challenges, the first successful attack is eligible for a prize of €200 (see the [terms and conditions](#) for more information).

**Join the challenge!**

**Username\***  
Required. 30 characters or fewer. Letters, digits and @/./+/-/\_ only.

**Email\***

**First name\***

**Already have an account?**

Log in

# More resources

28

- <https://www.aspire-fp7.eu>
  - papers
  - public reports
  - contact info
- <https://github.com/aspire-fp7>
- <https://github.com/diablo-rewriter>
- Youtube channel: ASPIRE-FP7 Software Protection Demonstration





# Aspire Grant Agreement No 609734

29

The Aspire project has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 609734.

If you need further information, please contact the coordinator:

Bjorn De Sutter, Ghent University

Technologiepark-Zwijnaarde 15, B-9052 Gent, Belgium

Tel: +32 9 264 33 67 Fax: +32 9 264 35 94

Email: [coordinator@aspire-fp7.eu](mailto:coordinator@aspire-fp7.eu)

Website: <https://www.aspire-fp7.eu>



The information in this document is provided "as is", and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.



Aspire : Advanced Software Protection: Integration, Research and Exploitation