IOT BZH

Security and privacy in your embedded systems

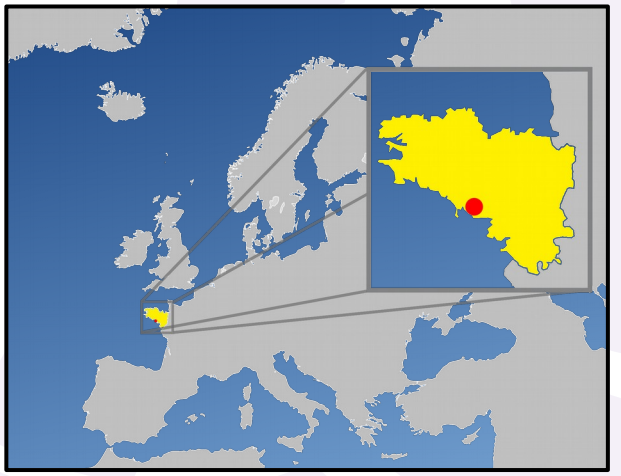
Strong isolation of applications using Smack and Cynara

José Bollo
security at IoT.bzh
jose.bollo@iot.bzh

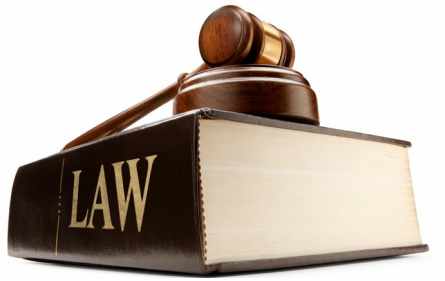
FOSDEM'16

IoT.bzh

- Specialized on Embedded & IoT
- Contributing to AGL Project for Renesas
- Expertise domains:
  - System architecture
  - Security
  - Application Framework
  - Graphics & Multimedia
  - Middleware
  - Linux Kernel
- Located in Brittany, France



Connected cars

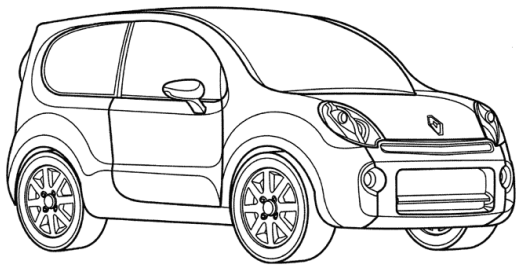


internet
+ cloud

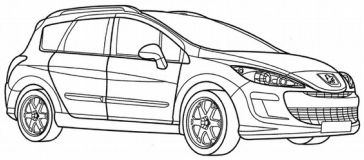
WiMAX

LTE Advanced

Wi-Fi HOT SPOT



NFC



HI-SPEED CERTIFIED USB

Wi-Fi

CAN

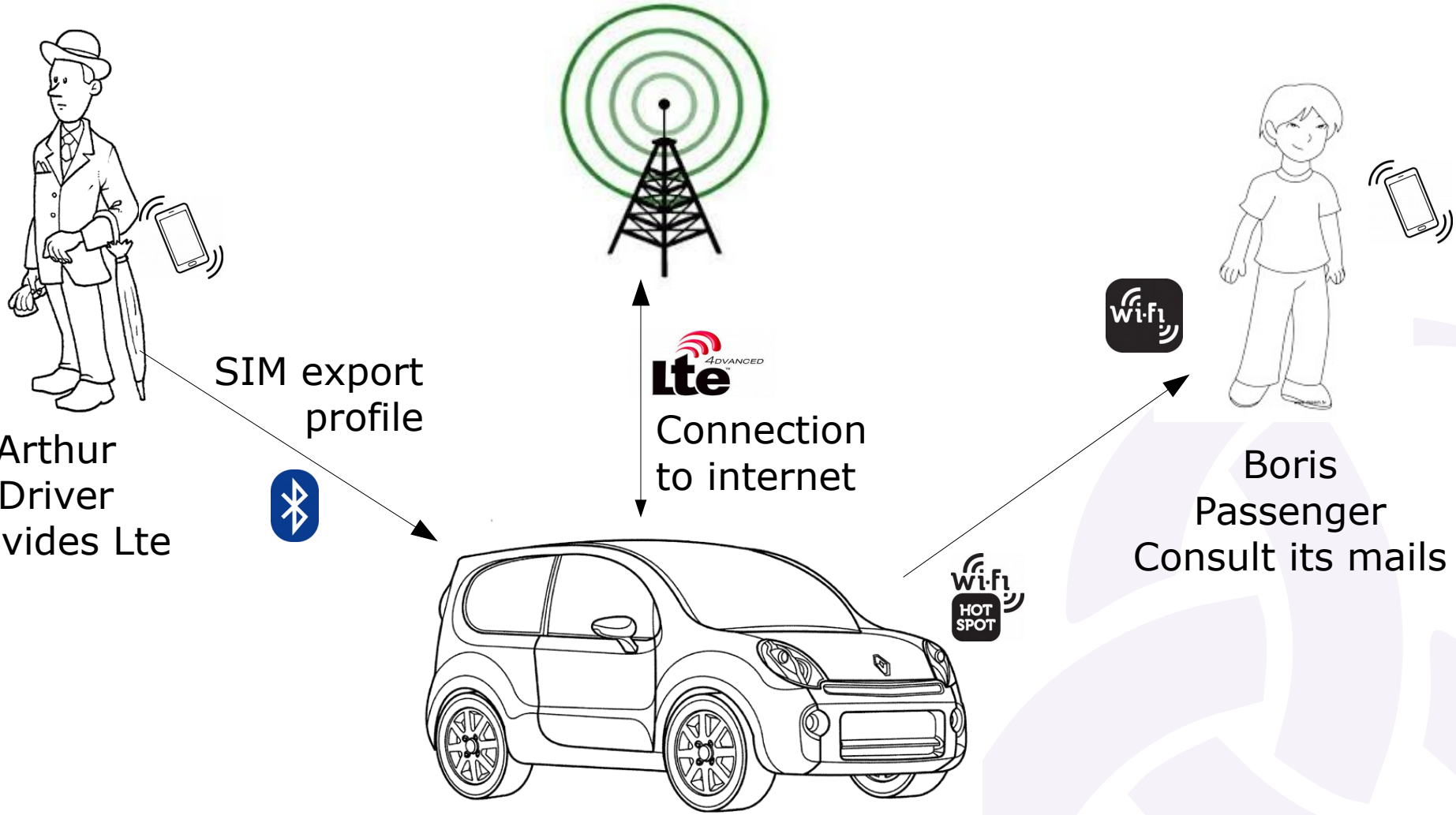
Attacks

- Some people have interest to attack systems:
  - States / Armies
  - Criminals
  - Family
- Attacks of the system can also be hazardous:
  - Bugs
  - Misuses + bugs
  - Wear
  - Accidents

Privacy

- No one wants to be spied or stolen
- Some people have interest to spy:
  - Merchandizers
  - Insurances
  - States / Armies
  - Criminals
  - Family

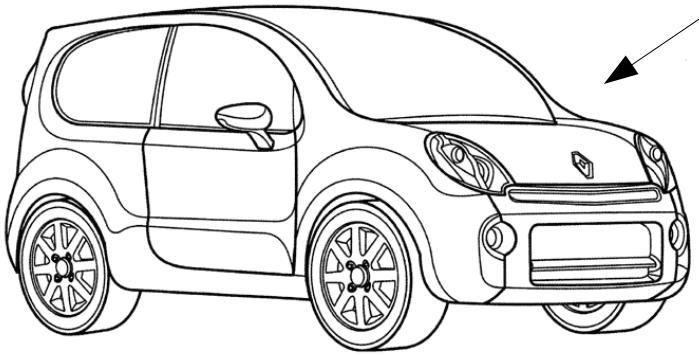
Example 1


SIM export
profile
Arthur
Driver
Provides Lte
Connection
to internet
Boris
Passenger
Consult its mails
wi-fi
HOT
SPOT

Example 2

Connection
to internet

Boris
Installs that very
cool application



Touch
screen

Some aspects of security

- Keep system integrity
  - System must not be changed
  - System must **update itself**
  - System must resist to brutal power off
- System should detect problems, intrusions, report
- **Applications must be isolated and their power must be restricted**

Isolation

- Isolate users
  - use DAC
- Isolate applications
  - Tizen: use MAC
  - Android: use DAC
- Use user aware services
  - Bluez should isolate as needed...

Restricting process's power

- Action of processes are restricted (sandboxed)
  - Each sensitive action should be filtered by the security sub-system
  - The security sub-system checks if the process has the permission to perform the sensitive action
  - The security sub-system prohibits the actions that aren't permitted
- Implementations details may differ from the above description

Permissions

- Any process has a set of permissions reflecting the sensitive actions that it can perform
- (user, application) → permissions
- Variants: does permissions change while process runs?
  - Static: NO
  - Dynamic: YES

DAC versus MAC

- With DAC, the **permissions can be changed object by object** by any possible writer
- With MAC, the **permissions are set by a fixed matrix** and changing the MAC tag of objects requires a linux capability
- Both operate on system objects

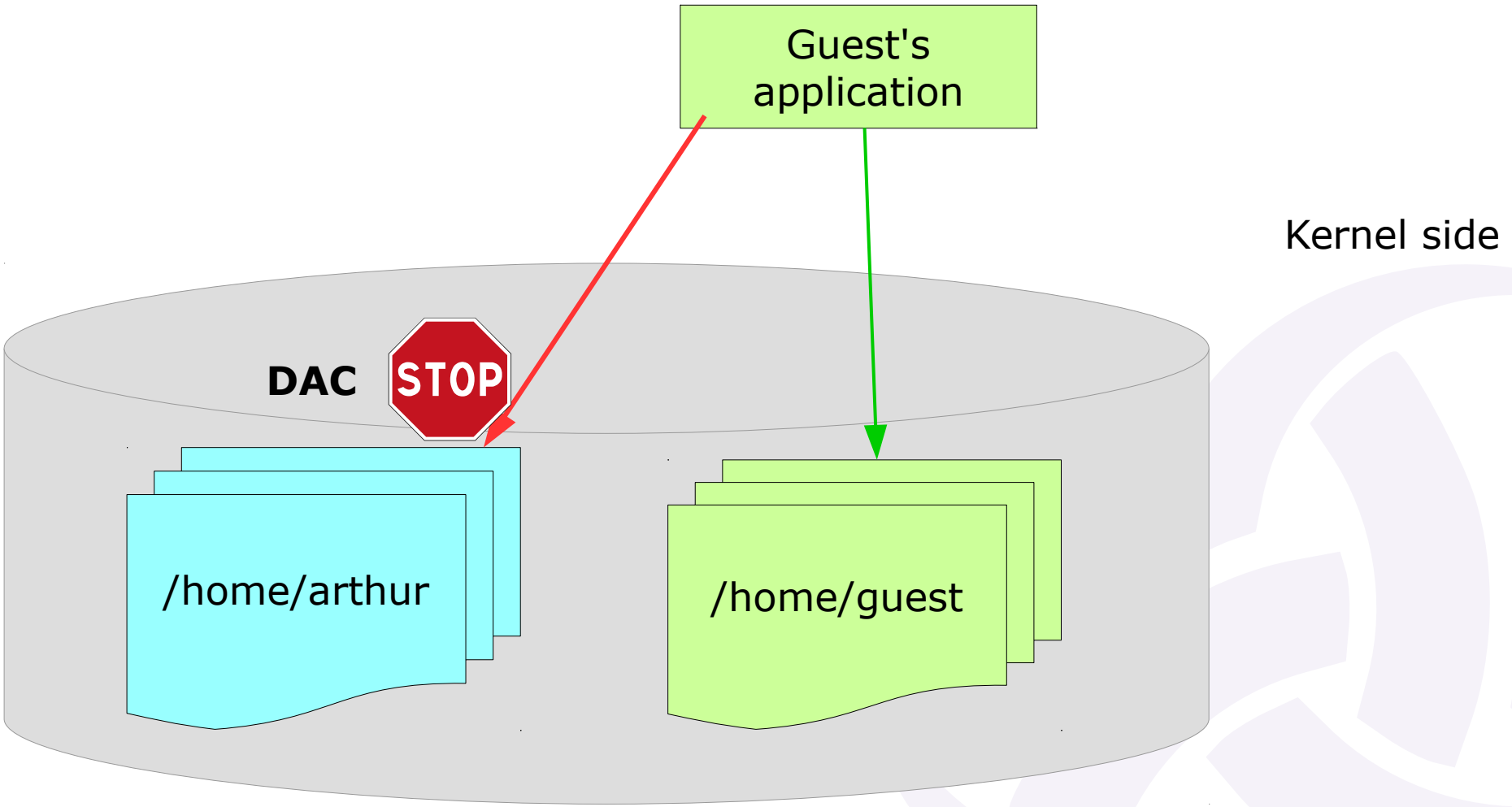
API permissions

- Some permissions can not be checked/filtered using system objects
- Examples:
  - Entering full screen
  - Acces to specific BlueTooth profile
-

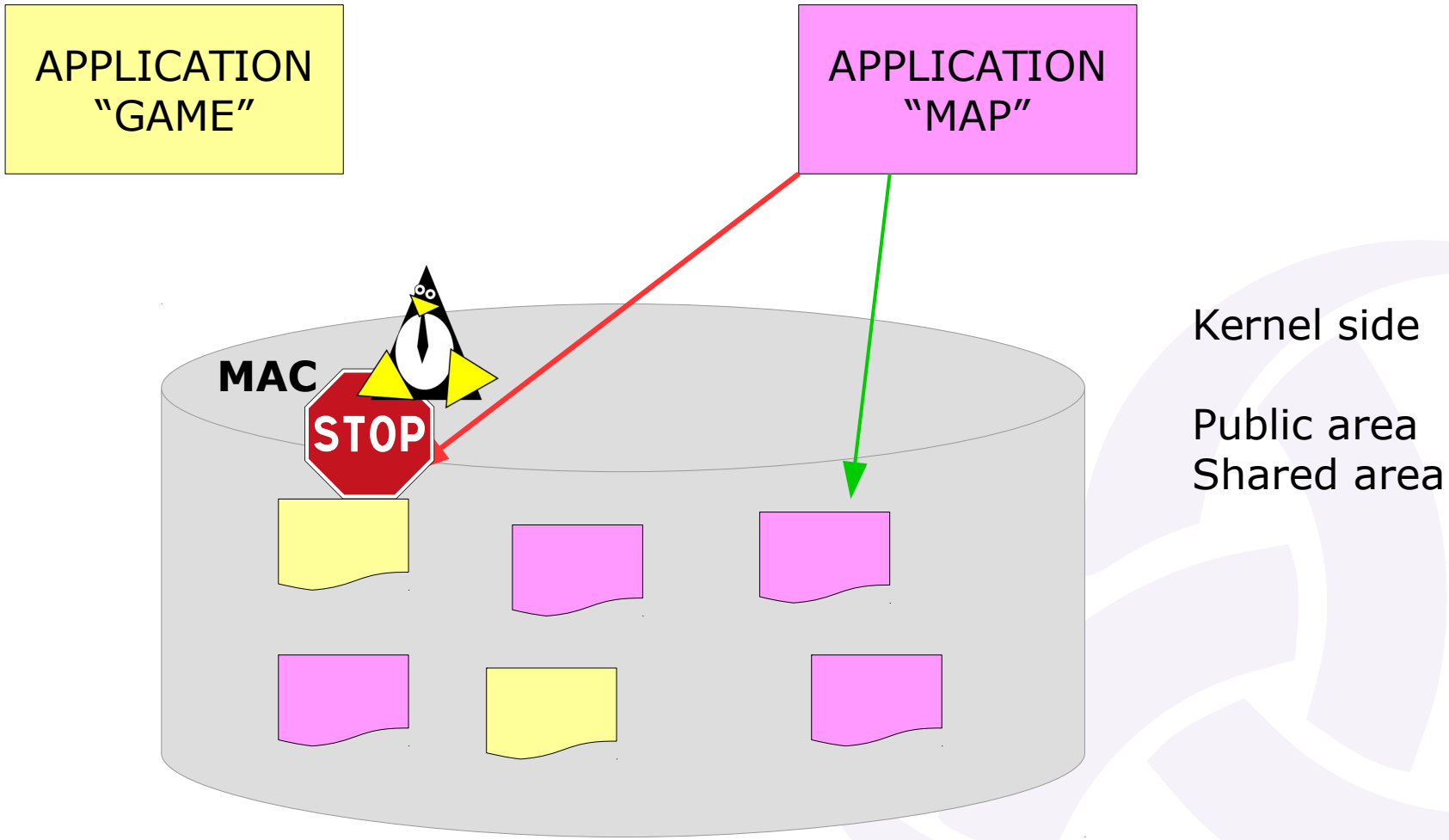
Implementations

- Virtualisation: specific environments are prepared for execution of processes
- Tizen: MAC (Smack) + DAC + Cynara
  - Allows native applications
- Android: MAC (SELinux) + DAC + Binder
  - Enforces use of binder the kernel module
- AGL: MAC (Smack) + DAC + Cynara + Binder
  - Allows native applications

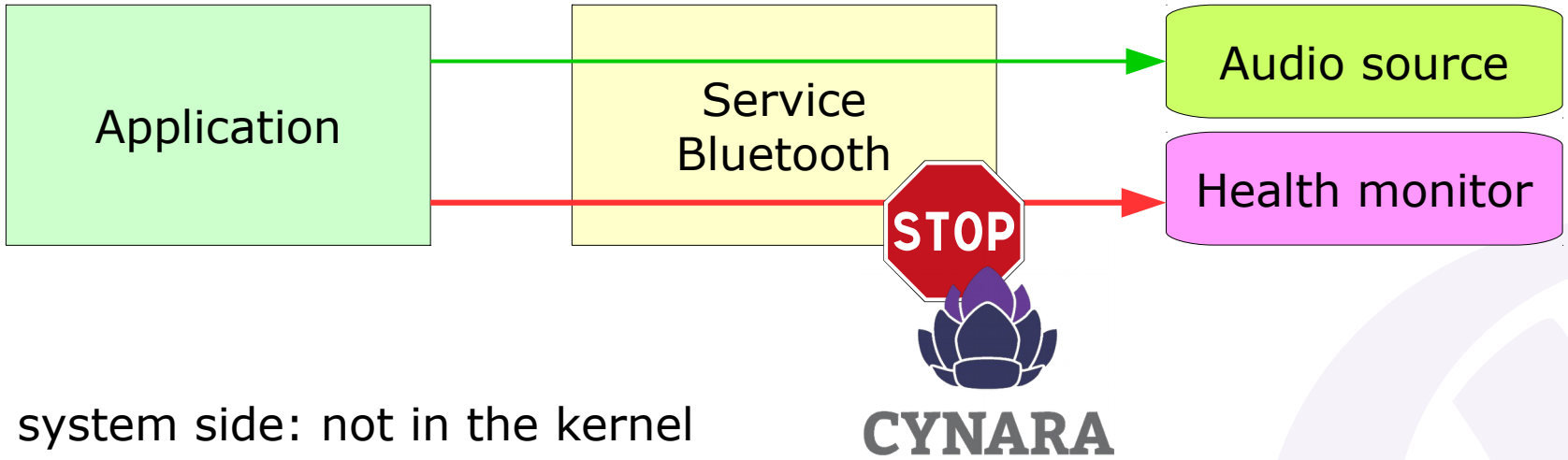
Isolation of users



Isolation of applications

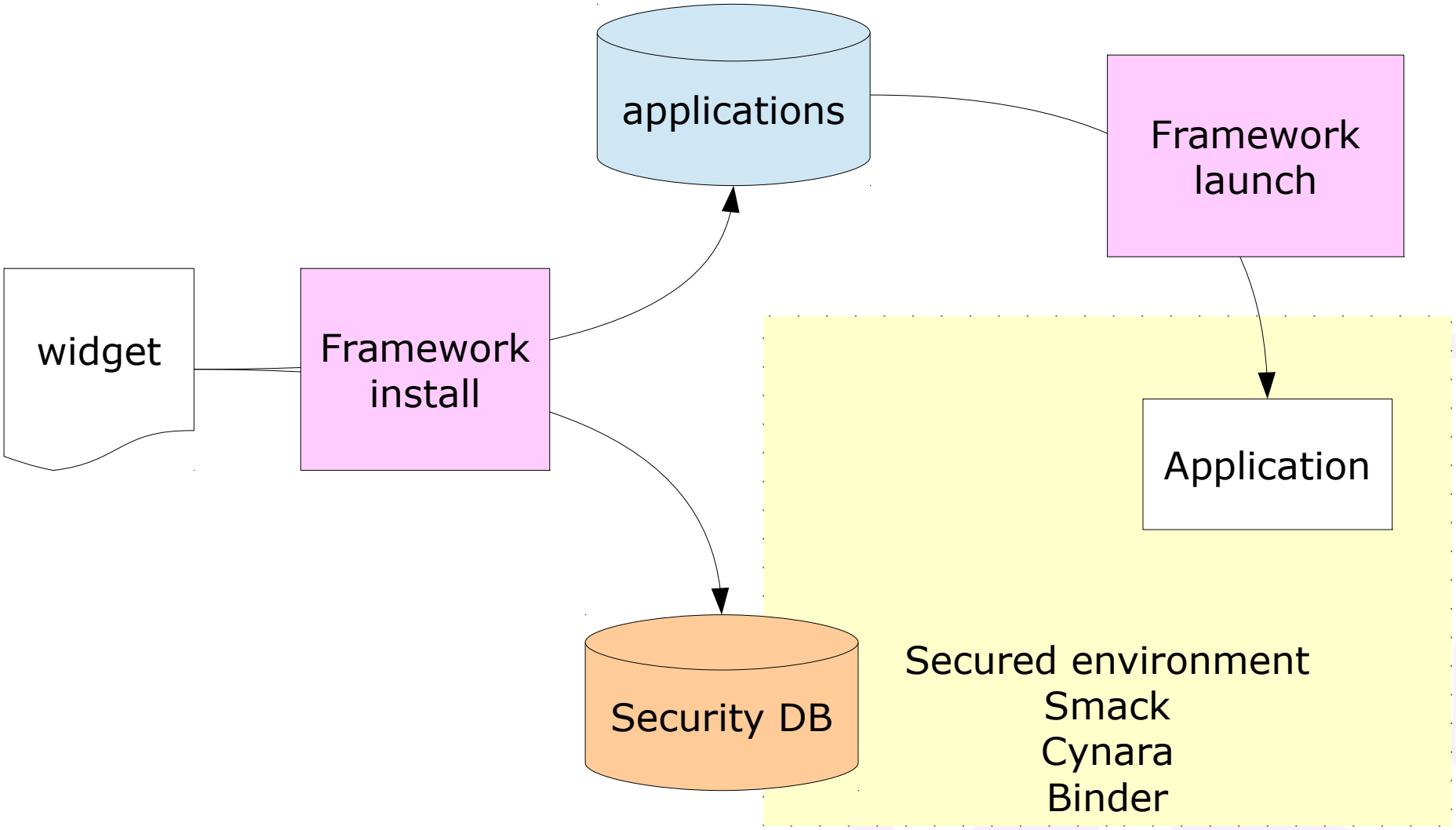


Restriction of services

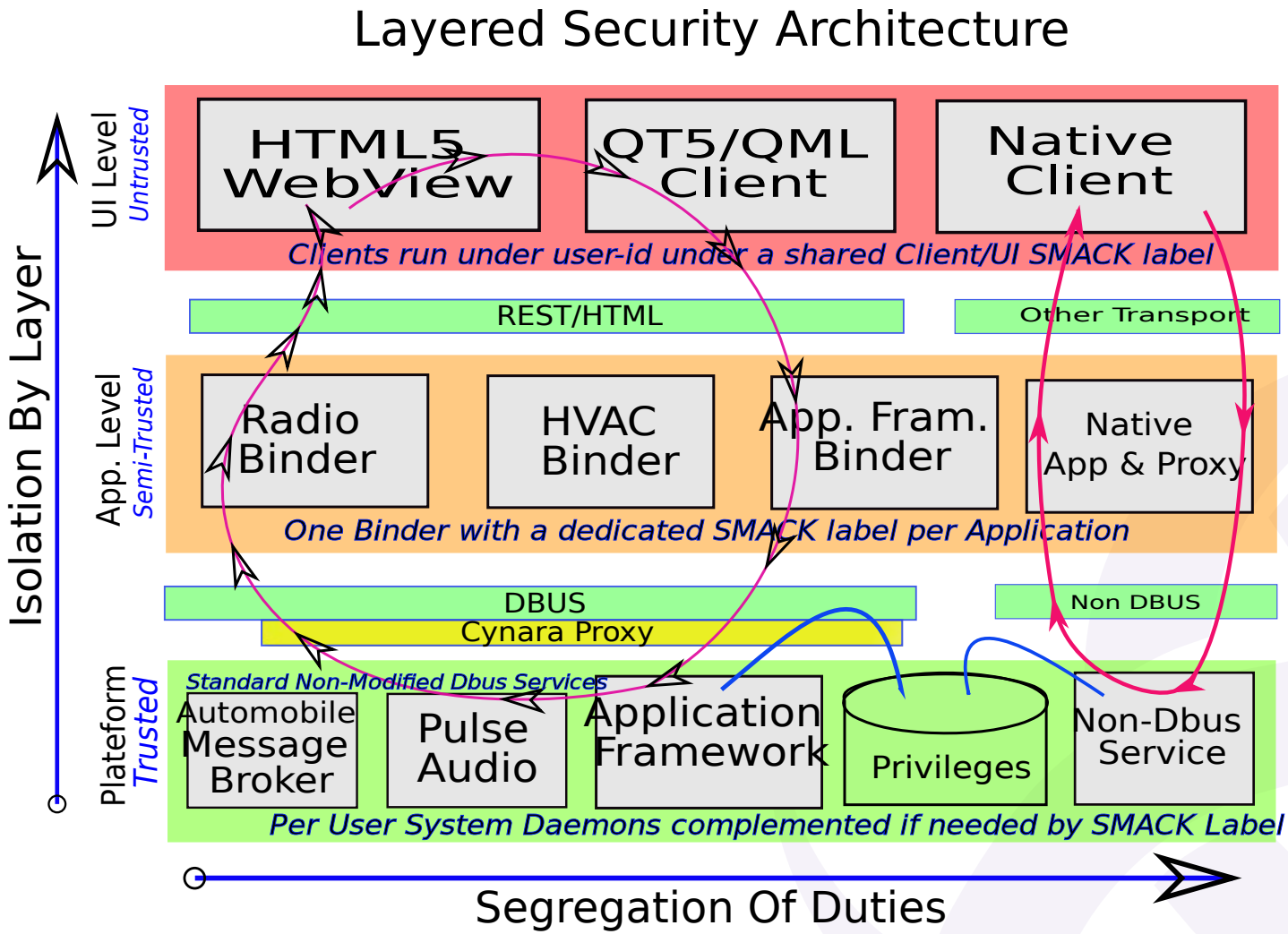


system side: not in the kernel

AGL framework



AGL framework



Who is interacting?

- A big problem: How handle the interaction, to wich user context to attach it?

Next?

- Reporting intrusion: nice-lad
- Secured Tagging: a proposal I made
- Kdbus? Binder?
-

QUESTIONS...

Too late for prevention

links

- http://www.iot.bzh
- https://wiki.tizen.org/wiki/Security
- http://schaufler-ca.com/
- https://archive.fosdem.org/2015/schedule/event/sec_enforcement/
- https://www.automotivelinux.org/
- http://iot.bzh/download/public/2015/tizen-security-lessons-learnt-initial.pdf