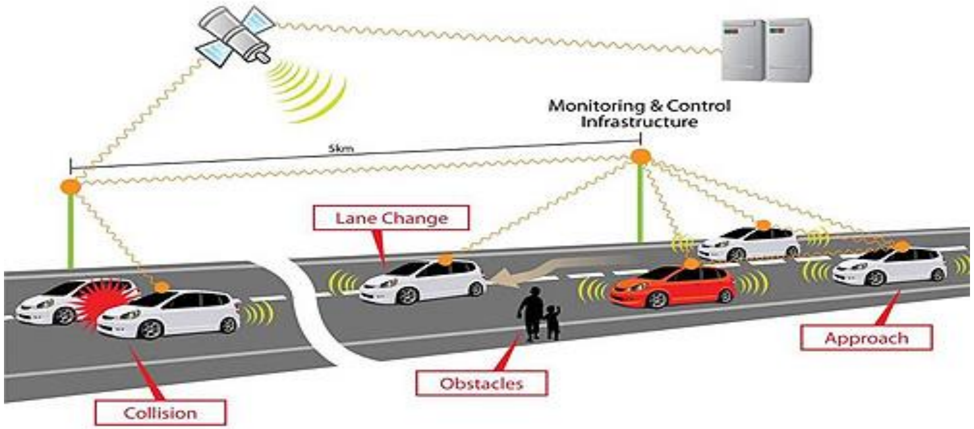
The Intelligent Vehicle as a Hostile Environment
智能汽车是一个可攻击的的环境

Yuan Xiang Gu (顾元祥)

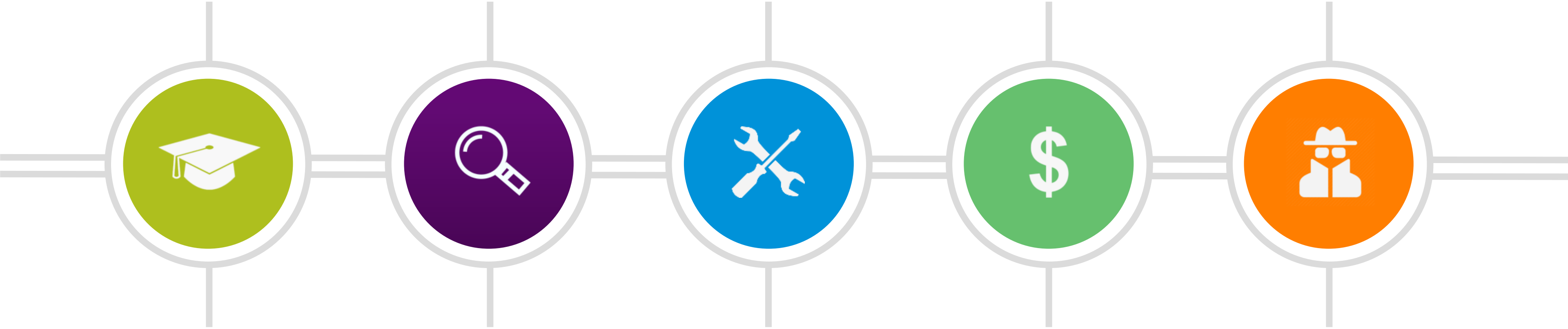
Co-funder of Cloakware, Chief Architect, Guest Professor of Northwest University

The Intelligent Vehicle is Vulnerable for Attacks ?



Security Attacks to Computer Systems Never Stop

Cybersecurity Trajectory



00

Educating

A “new” industry’s cybersecurity cycle begins with security researchers raising industry awareness of vulnerabilities

01

Probing

Once academics expose vulnerabilities, elite hackers become attracted, searching for new technical challenges

02

Specializing

After elite hackers create working hacks, they develop highly specialized toolkits that enable unsophisticated users

03

Commercializing

In about three years, black-hat hackers determine how to monetize security flaws for eventual sale

04

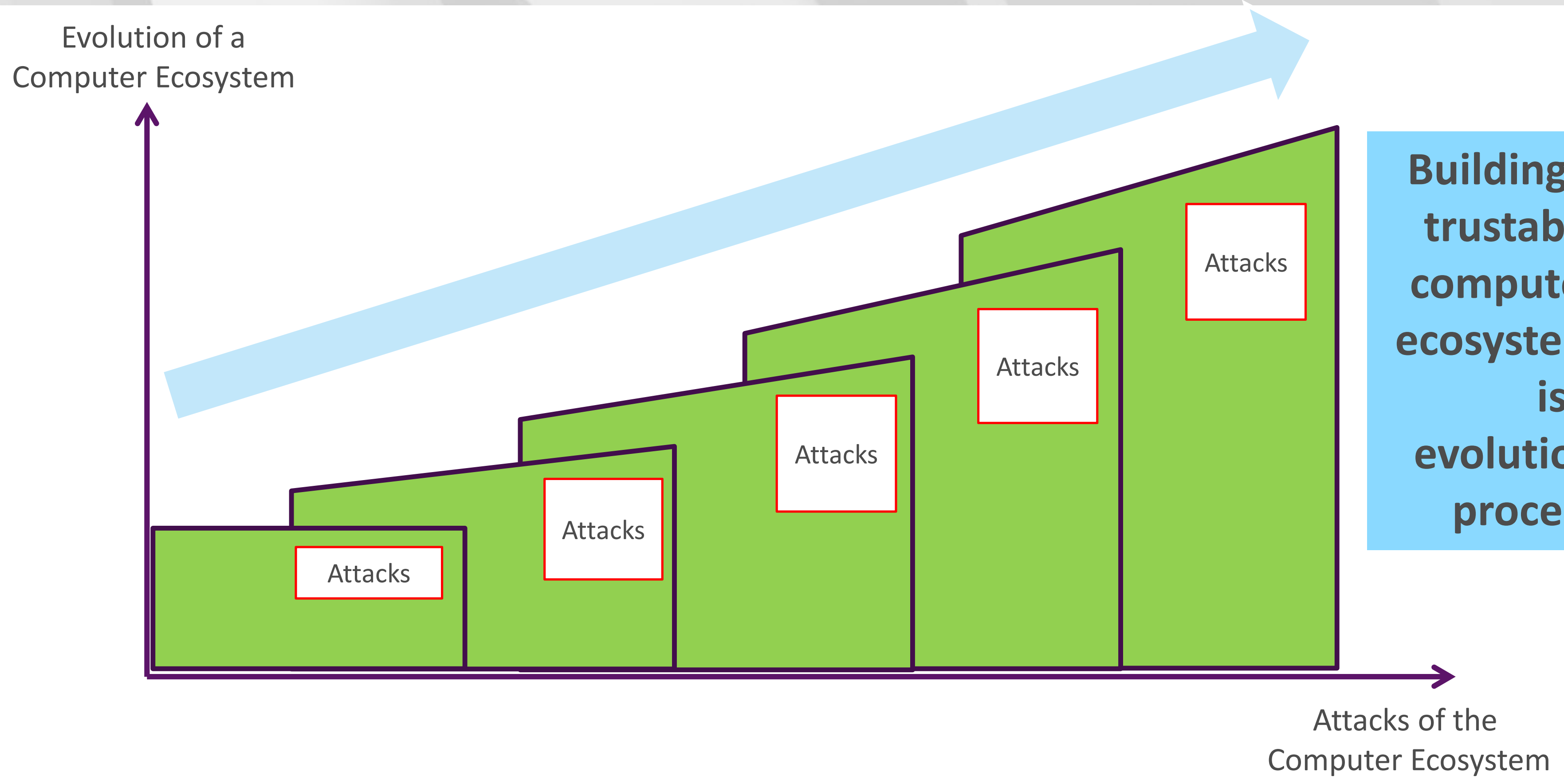
Profiteering

Within 4-5 years, hacks are sold through openly illegal services to unlawful hackers and existing criminal organizations

Eight Major Security Lessons

- Lesson 1: Hostile is a normality, trusted just is a state
- Lesson 2: Vulnerability does always exist
- Lesson 3: Attackers cannot be avoided
- Lesson 4: Attacking has effective methodology
- Lesson 5: Complexity is the biggest enemy of security
- Lesson 6: Software must be protected
- Lesson 7: Security cannot be fixed, but must be dynamic and renewable
- Lesson 8: Need both of pro-active and re-active security

Lesson 1: Hostile is a Normality, Trusted just is a State


Evolution of a Computer Ecosystem
Attacks
Attacks
Attacks
Attacks
Attacks
Attacks of the Computer Ecosystem

Building a trustable computer ecosystem is a evolution process

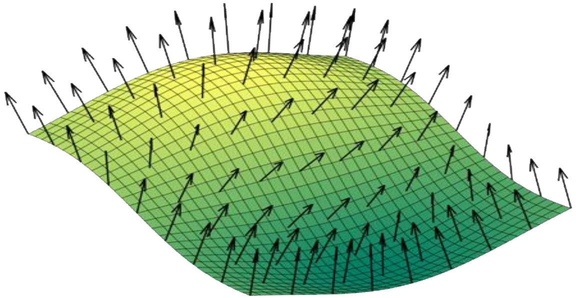
Lesson 2: Vulnerability Does Always Exist

- Vulnerability

  A weakness which allows an attacker to develop and launch an attack

- Attack Surface

  The sum of the different points where an attacker can break a system



- Zero Day Vulnerability and Attack

  Un-exploited and un-known security holes to vendors that can be developed into brand new attacks

Lesson 3: Attackers Cannot Be Avoided

- **For Fun**
  - Unsophisticated Attackers

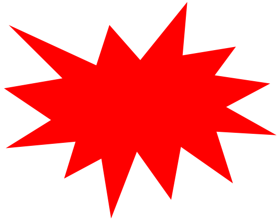
- **For Profit**
  - Cheaters
  - Black Business
  - Organized Crime
  - Terrorist Organizations

- **For Special Interests**
  - Competitors
  - Nation States
  - Terrorist Organizations

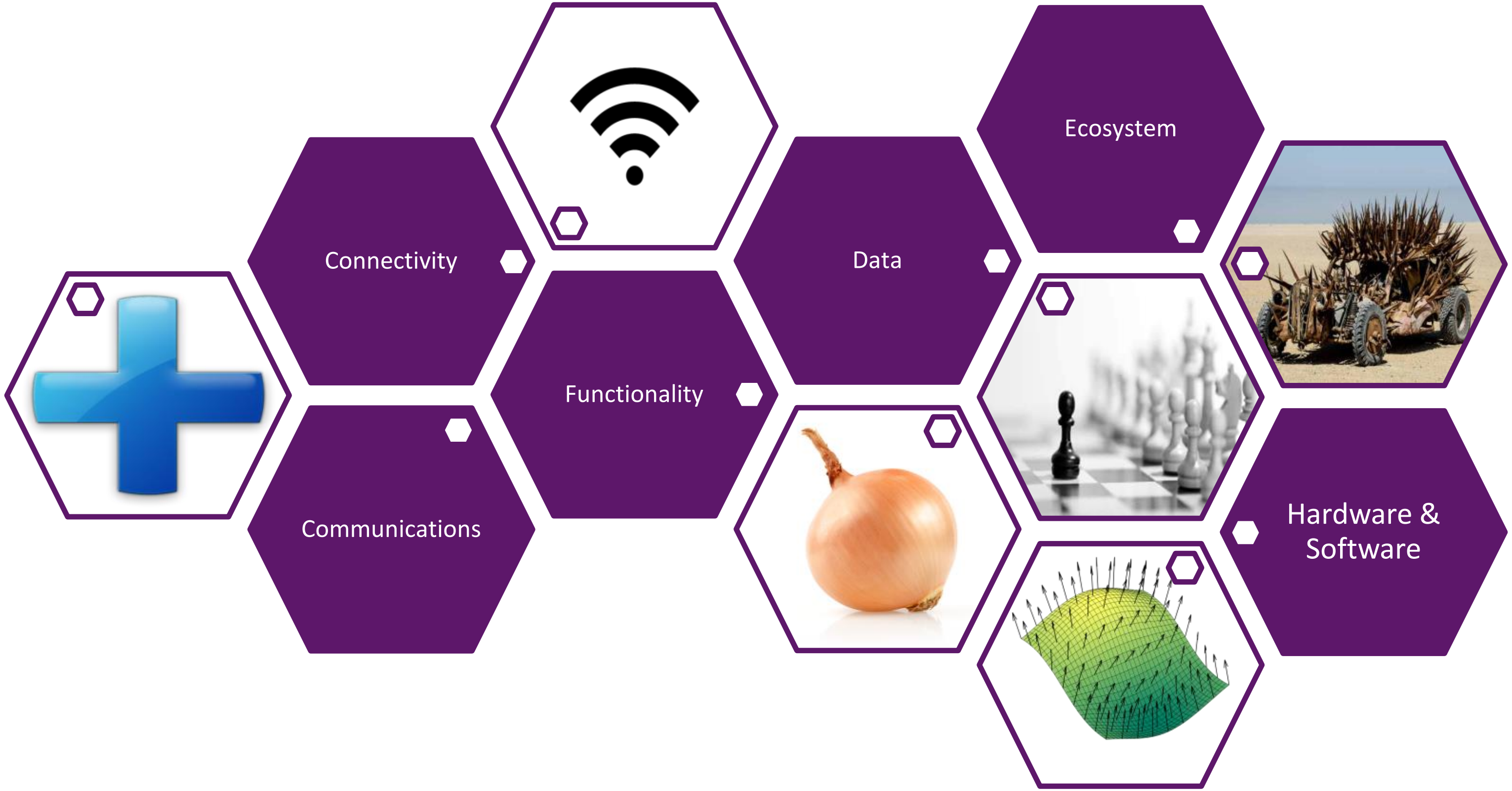
- **For Challenge**
  - Sophisticated Researchers

Lesson 4: Attacking Has Effective Methodology

- Investigation
- Leverage a weakness
- Peel the onion, and discover
- Develop an attack, and rinse and repeat
- Launch the attack



Lesson 5: Complexity is the Biggest Enemy of Security



Connectivity

Communications

Functionality

Data

Ecosystem

Hardware & Software

- Within modern systems, software is the **KIND**

Computing is pervasive and software is everywhere

- Create digitalized information
- Store the information locally and remotely
- Access and process the information
- Distribute and exchange the information
- Interact with users to use the information
- Protect the information

Without protecting software itself, you cannot really protect digitalized information & assets, and business logics & models

Lesson 7: Security Cannot be Fixed, but Must be Dynamic & Renewable

irdeto

If the security of a system breaks, what's your security strategy?

- Static/Fixed security model

  The security is gone and hard to restored

- Dynamic security model

  The security can be renewed and restored immediately in a planned way

Dynamic Security Is Essential

- Business Dynamics
- Attack Dynamics
- Technology Dynamics
- Digital Asset Dynamics
- Software is dynamic

Security needs to be dynamic and evolving

Lesson 8: Need both of Pro-active and Re-active Security

- Proactive Security
  - Design and build-in security and protection
  - Monitor hacker channels to understand attack techniques and methodologies
  - Apply security updates to reset the hacker’s clock
  - Always detecting

- Reactive Security
  - Limits the impact of a breach before it has a significant impact
  - Effective response

- Benefits:
  - Disrupt potential hacks before they happen
  - Mitigate impact of a security breach
  - Minimal disruption of business

What Else Lessons Have We Learned?

- Anything can be hacked given enough time and effort (including HW)
- Perimeter security is not enough
- Adopting best practices is key
- There are lots of reasons behind the motivation to attack
- There is no single protection can stop all attacks. Instead, we have to layer and combine different protection techniques into a protected and interlocked security maze.
- Working together is key
- Use best of breed technologies

- Much more …

How Secure Can a Vehicle Be Today

- Local attacks
- Remote attacks
- Personal Data Theft
- Software Bugs
- Architectural Defects

The Battlefield is new, and fighting and racing between protections and attacks just start

Safety, Security & Privacy

- The security of the vehicle will become as important as its safety
  - Safety and Security get increasingly closer

- Vehicles are generating more and more data that needs protection

- The growing threats of ransomware and terrorism

Where Do We Go From Here?

- Looking again to other industries and learn more
- Best Practices
- Security Audits & Assessments
- Methodology & Ideals
  - Defense in Depth
  - Trust but Verify
  - Least Privilege

What Else Can We Do?

- Defense in depth
  - Root of Trust
  - Chain of Trust
  - Hardware Anchoring
  - Kernel Hardening
  - Only running trusted software
  - Firewall
  - Intrusion Detection (IDS)
  - Anti debugging
  - Software signing
  - And lots more that already exists

- Secure OTA updates
- Secure Telemetry

Cloakware for Automotive – Cockpit System HMI

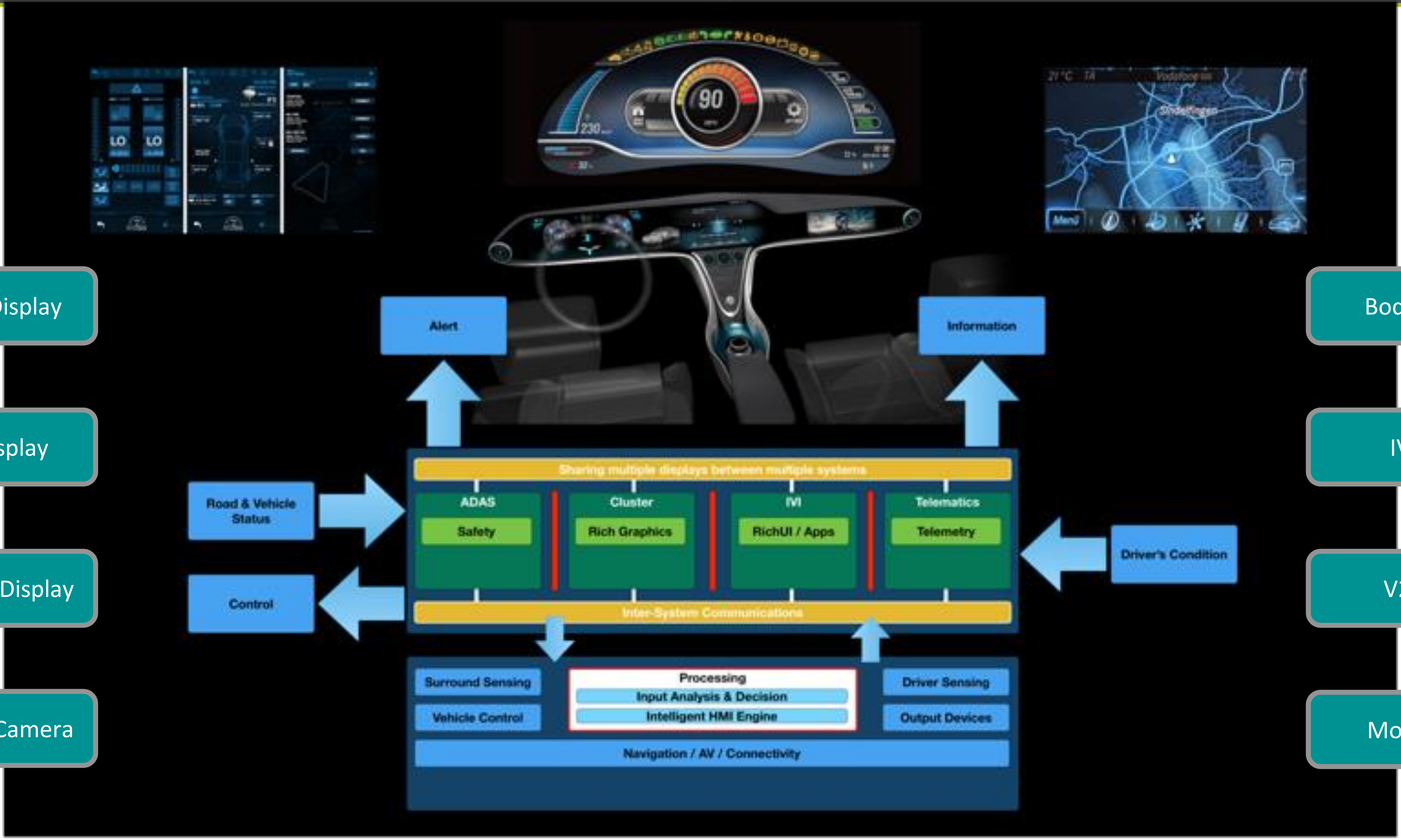
cloakware® for automotive

Heads Up Display

Cluster Display

Information Display

360 degree Camera



Body Control

IVI Apps

V2V / V2I

Mobile Apps

Cloakware for Automotive – Cockpit System HMI

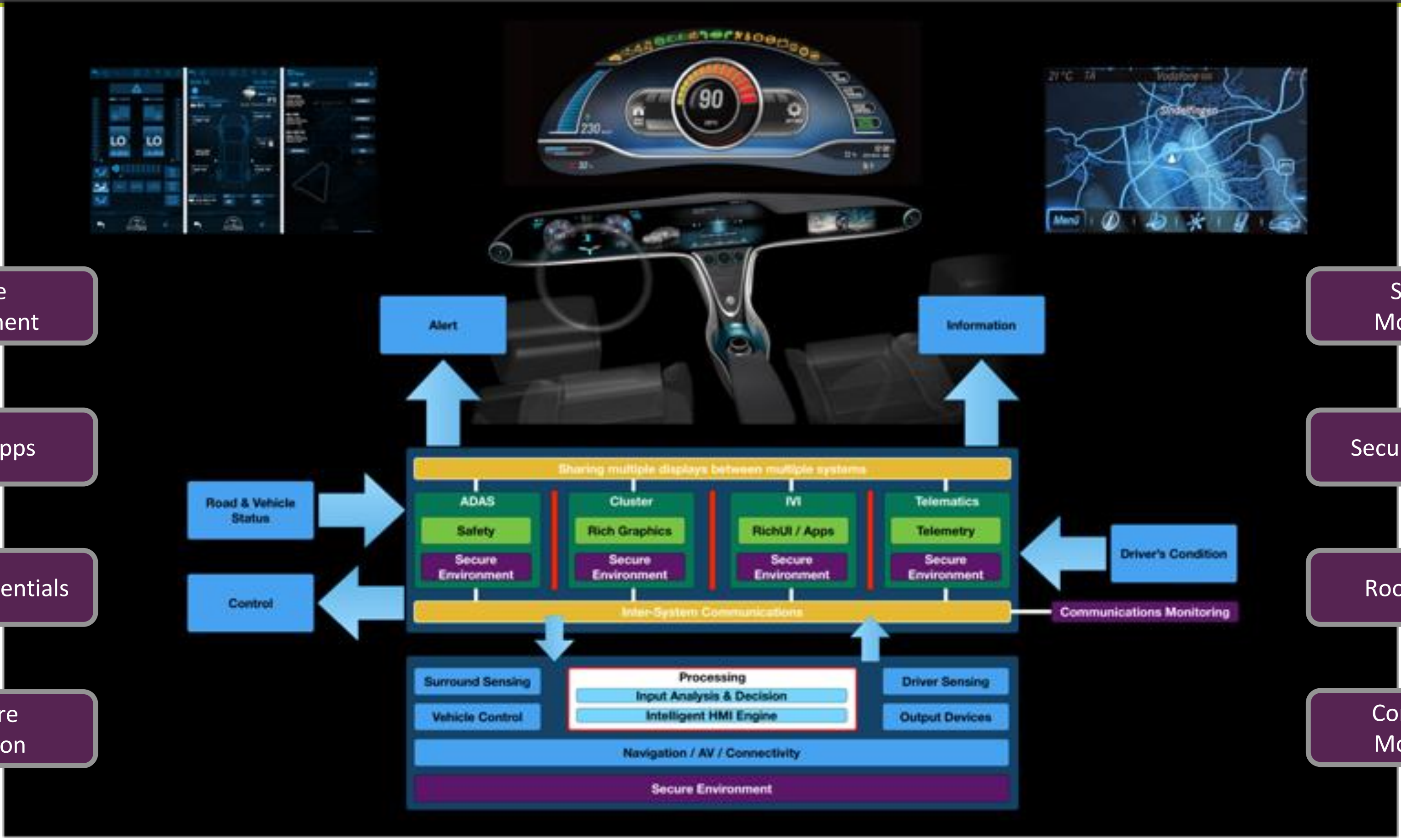
cloakware®
for automotive

Secure Environment

Secure Apps

Keys & Credentials

Software Protection



Security Monitoring

Secure Updates

Root of Trust

Comms Bus Monitoring

Thank you!

Panel Questions: Industrial Challenges

Q1 (by Mattias De Wael)
To what extend do “secure coding guidelines” (such as http://www.oracle.com/technetwork/java/seccodeguide-139067.html for Java) help me make my code secure? Especially becasue this week we learned that attackers just alter your code at runtime, so none of these ‘defence guidelines’ seem to help.

E.g.: protect from SQL injection can be overridden, if one sets a breakpoint before the execution of a “injection protected statement” and just replaces the ‘protected’ code by an arbitrary (malicious) statement.