Conception de systèmes numériques sécurisés

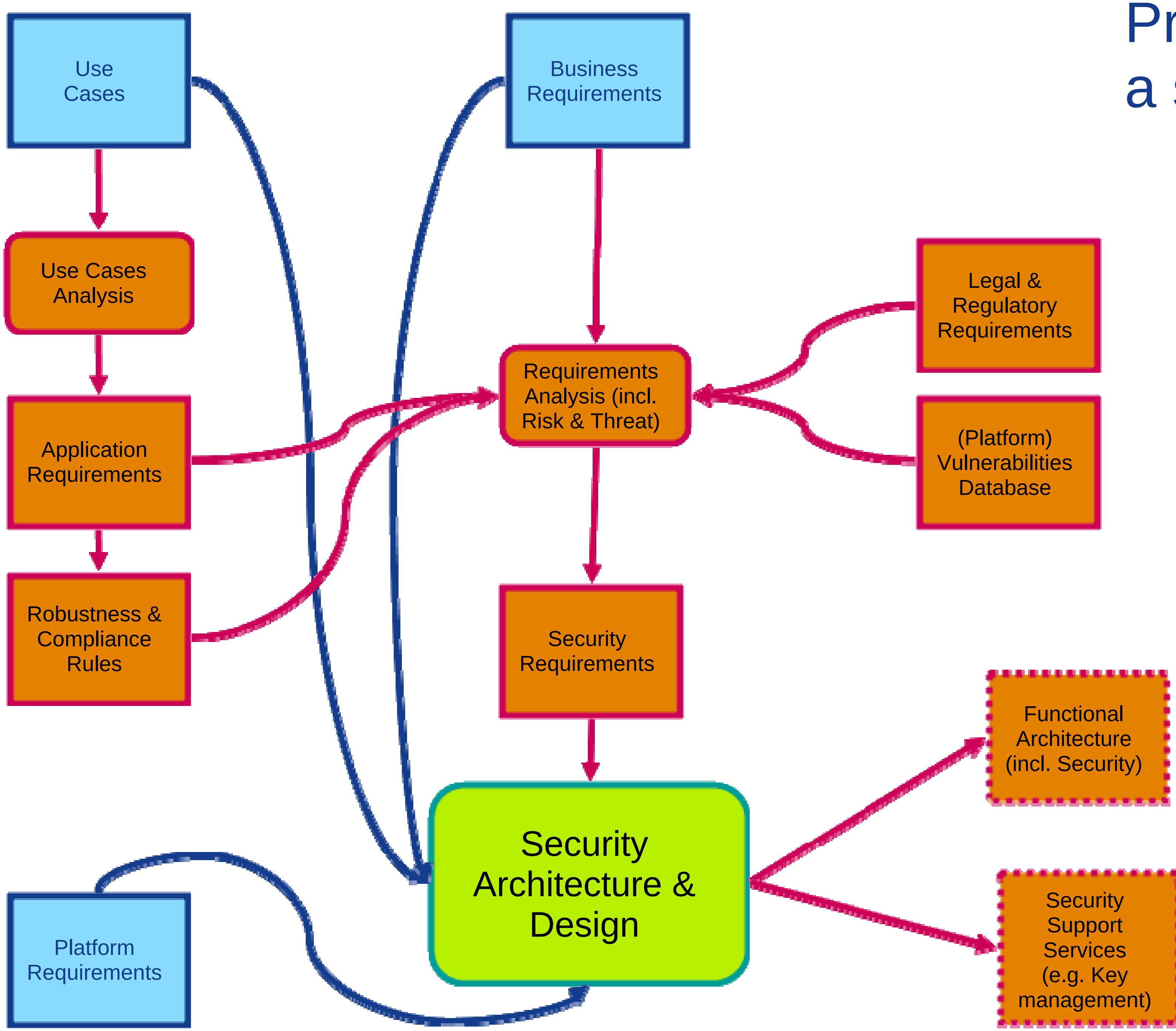
Hugues de Perthuis
hugues.de.perthuis@nxp.com

Objectifs et histoire de la présentation

- Extrait d'une présentation destinée à sensibiliser les architectes et concepteurs de circuits intégrés Grand Public
  - Historiquement peu concerné par la sécurité
  - Systèmes ouverts

Threat analysis

Process to make a secure system



Threat analysis

- Cost of security has to be less than expected loss
  - a system is vulnerable if the effort to break is less than the expected gain
- What do we want to protect ?
  - who are the stakeholders ?
  - which items do they want to be protect ?
  - what is the value of the items ?
- Against whom ?
  - what is likehood of attack
  - what are attackers incentives
  - what are attackers resources

Threats depend on objectives and product

E.g.

- User has interest to break the security or not.
- Third party applications imposed requirements (e.g. DRM, Wifi protection)
- Government imposed (e.g. privacy)

Which assets are to be protected and for how long?

E.g.

- NXP algorithms
- OEM software
- Compressed video
- Credit card info
- Privacy
- 10 year passport life time versus 10 minute stock market data

Product, application and usage characteristics.

Which attacks by whom should be protected against and for how long?

E.g.

- End users
- End users using a script from a sophisticated hacker
- Professional tools like Focused Ion Beam (FIB) machine

Threats depend on objectives and product

E.g.
- User has interest to break the security or not.
- Third party applications imposed requirements (e.g. DRM, Wifi protection)
- Government imposed (e.g. privacy)

Which assets are to be protected and for how long?

E.g.
- NXP algorithms
- OEM software
- Compressed video
- Credit card info
- Privacy
- 10 year passport life time versus 10 minute stock market data

Required technology and processes are determined by these three dimensions

Plus one other dimension:

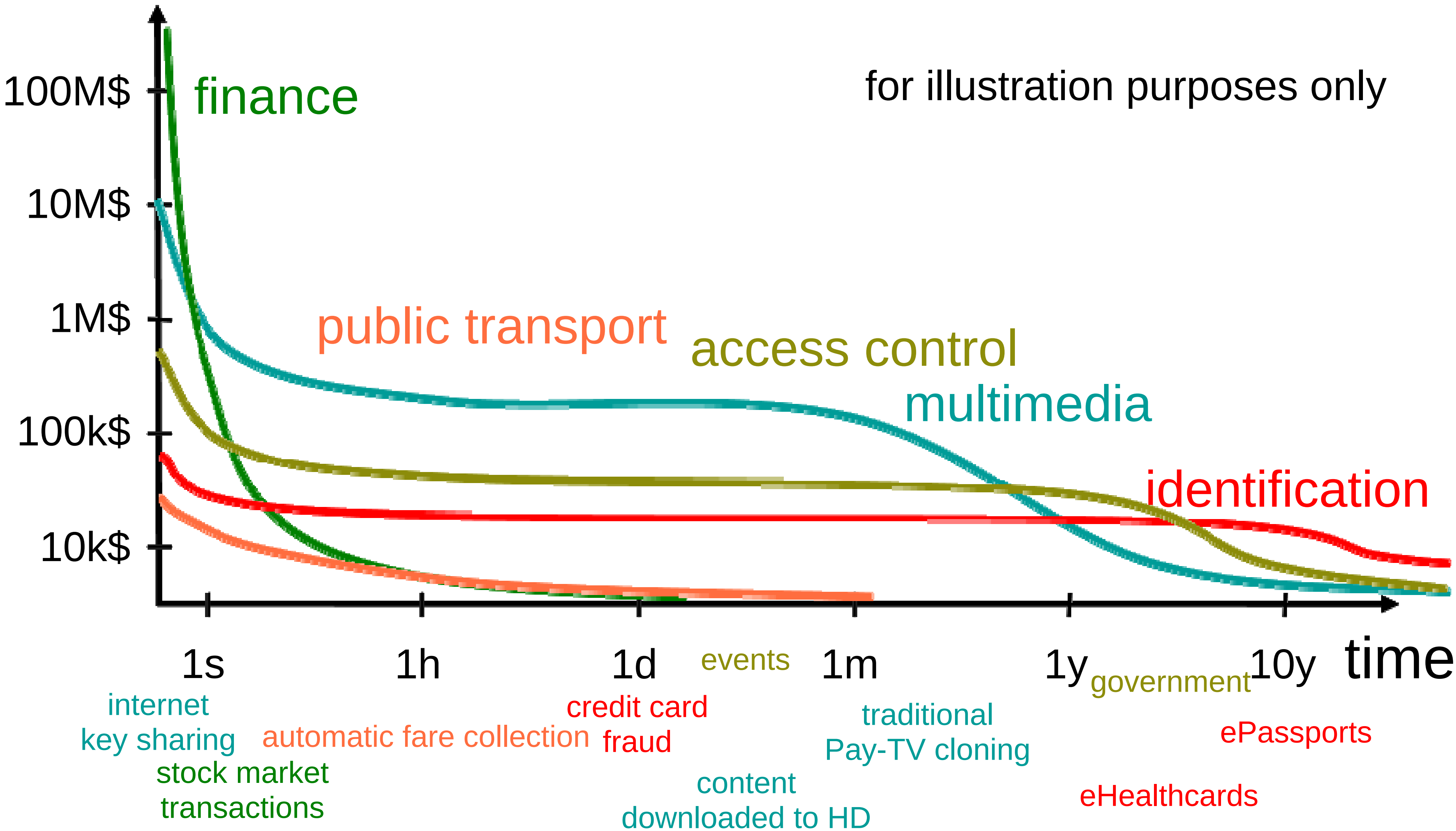
Assurance level:
How well we can underpin that the attacks can be countered.

- End users
- End users using a script from a sophisticated hacker
- Professional tools like Focused Ion Beam (FIB) machine

Product, application and usage characteristics.

and for how long?

Value of an attack depends on market



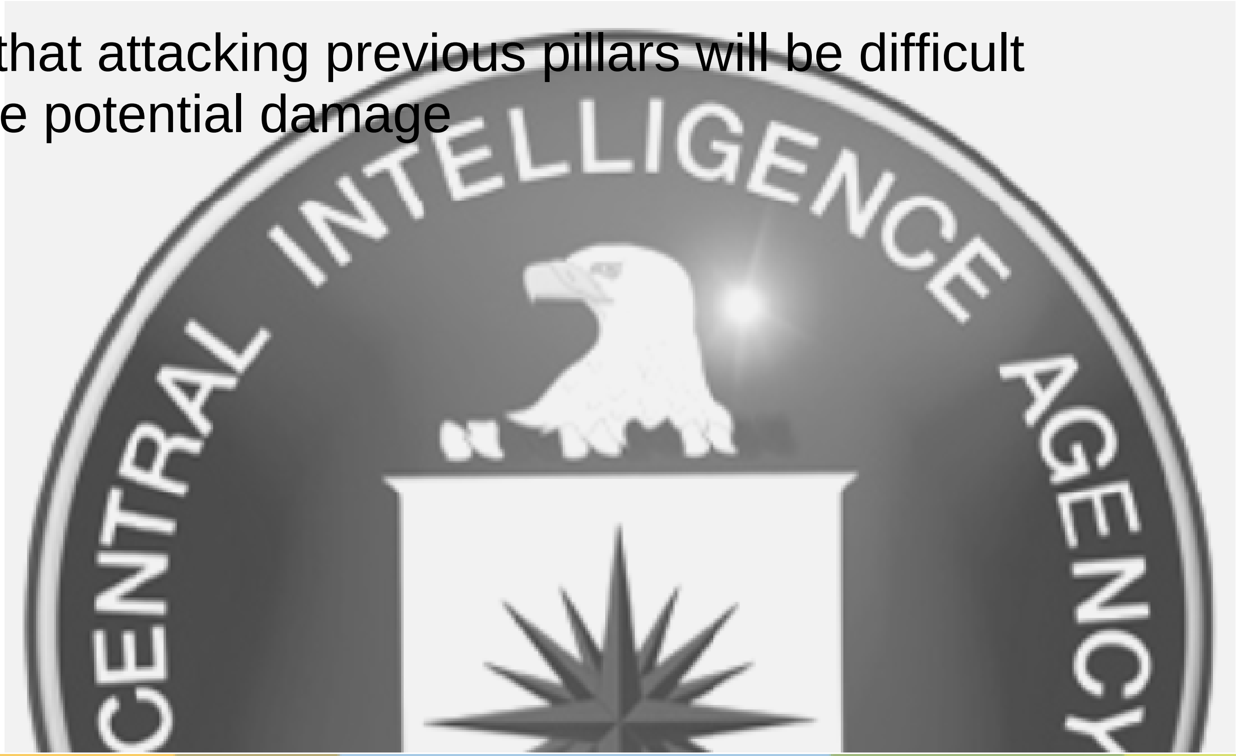
Attackers & Target

- Professional
  - content stealing
  - IP reverse-engineering
  - device cloning /unlicensed usage
  - reputation damage
- Hacker/academia
  - technical challenge
  - peer recognition
- User
  - feature addition/upgrade
  - alternative usage
  - content abuse

Secure design

Secure design pillars

- Confidentiality
  - ensuring that information is accessible only to those authorized
- Integrity
  - ensuring that information has not been modified
- Authenticity
  - ensuring that information comes from trusted source
- Resilience
  - ensuring that attacking previous pillars will be difficult
  - limiting the potential damage



Confidentiality

- Disable or protect access to debug and observability
  - DFX (scan, bist, boundary scan, …)
  - EJTAG, debug info and functions …
  - bury PCB lines
- Build firewall between process, DMA channels and memories
- Limit access to information on “Need to know” basis
  - close access to boot ROM after boot
  - access to keys via direct connection or handle
- Encrypt or obfuscate
  - configuration bits (OTP, …)
  - external memories, internal SRAM and ROM
  - external storage
- Use cryptographic protocols
- …

Integrity

- Detect modification of code and data
  - in memory and external storage
- Use type safe coding
  - check parameters, buffer length
- Check configuration bits, instructions
  - OTP, DMA engine, …..
- Use sensors to check environment
- …

Authenticity

- Give unique ID and secret keys to enable powerful cryptographic services
  - secure storage
  - HW/SW linking
  - …
- Build chain of trust by checking code signature:
  - at boot, during dynamic load, update
- Authenticate messages, files, commands with signatures
  - beware of replay attacks
- Identify DMA initiator to allow access control
- …

Resilience

- Divide system in multiple and isolated components
  - limit area and interfaces
  - virtualization can be an option for SW components
- Use multiple layers of protections
- Never trust SW !
  - unless it is very small and designed with security in mind
- Create fail safe implementation
  - what happens in case of glitch or other corner cases ?
- Use HW cryptographic accelerators
  - help to limit access to secret keys
- Implement complete protocol
- …

Security through obscurity

- Similar to hiding the key under the doormat
- Gives a false sense of security
  - information on the system become an additional threat
  - adds complexity to the design, i.e. failure points
  - does not protect against collaborative attacks

It should be assumed that the entire design of a security system is known to attackers

- security resides in keys
- restraining access to info is however a good policy and sometimes obfuscation is only option

Do not design your own cryptographic system

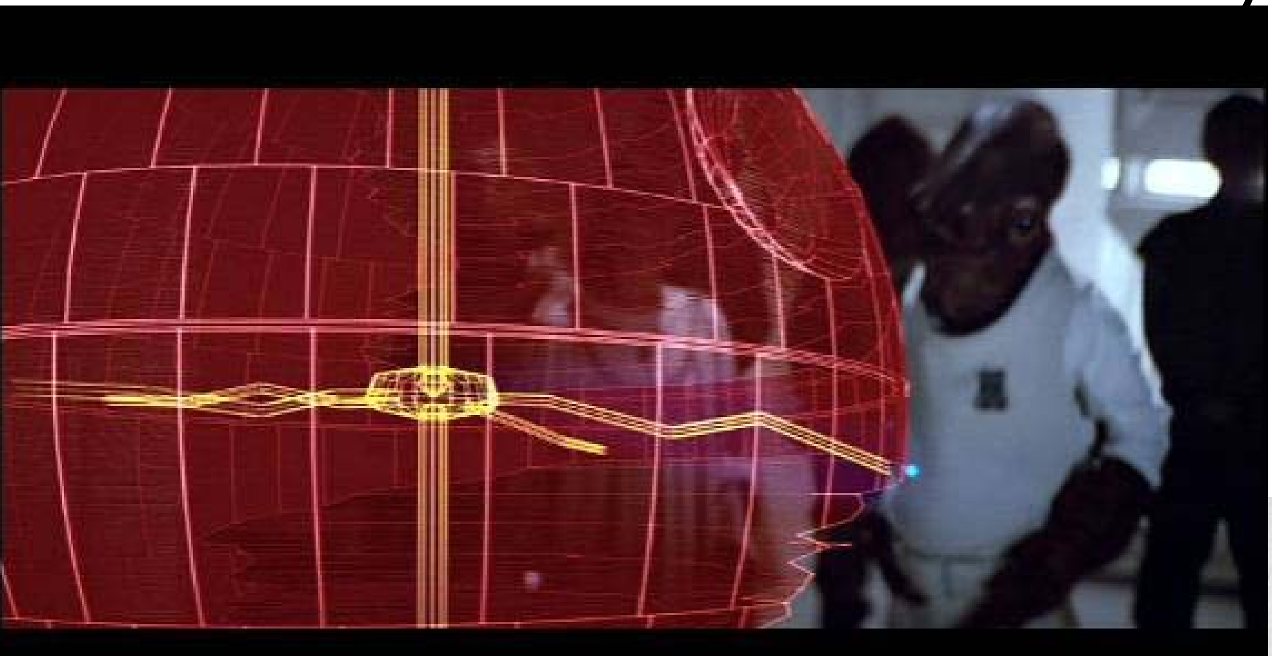
- non reviewed crypto always fails

Design of secure system

- Apply system thinking
  - security should not be an afterthought
  - security can also be compromised/enforced at different stages (design/integration center, fab, ODM, sales, …)

- Layered security
  - each layer will increase the cost of the attack, but assume each will be broken
    - breaking one should not lead to complete collapse
    - use overlapping systems
  - isolate sub-systems
    - deny vulnerable components access to sensitive assets
    - most secure part should have smallest interface area

Illustration of different security approaches



Star Wars VI or security seen by Hollywood

Security not part of system design : single point of failure

System designed with security in mind : Careful isolation of assets with multi-layered protection

Thanks for your attention!