

Evolutions de la problématique de sécurité dans les systèmes embarqués

Claire Loiseaux

TRUSTED LOGIC

5, Rue du Bailliage - 78000 Versailles - FRANCE
Tel: +33 1 30 97 25 00 – Fax: +33 1 30 97 25 19
contact@trusted-logic.fr / www.trusted-logic.fr

Agenda

- Where do we stand**
- On java embedded systems**
- On security methodology**
- Illustration**

From Mono application devices to

- ❑ **Multi-application**
 - Isolation, cohabitation, sharing, ...
 - Sensitive or not
 - Life time
- ❑ **Semi-Open systems**
 - Only a portion is open
- ❑ **Open systems**
 - All the applications are downloaded
- ❑ **Multi-operation**
 - The system is controlled by several entities (main issuer + with space location)
- ❑ **Remote administration**
 - Systems can be updated

Already existing scenarios

- Closed and native multi-application smart cards for banking**
- Closed java cards for mobile phones**
- Native Application download on PDA**
- System updates on payment terminals**
- Dynamic algorithm selection for pay TV**

Various Security mechanisms

❑ Hardware/Software

- Tamper evidence/resistance ...

❑ Standard OS features

- Access control, Memory management, Integrity Checks, transaction, crypto, ...Proof of origin

❑ Virtual machine : existing and emerging standards

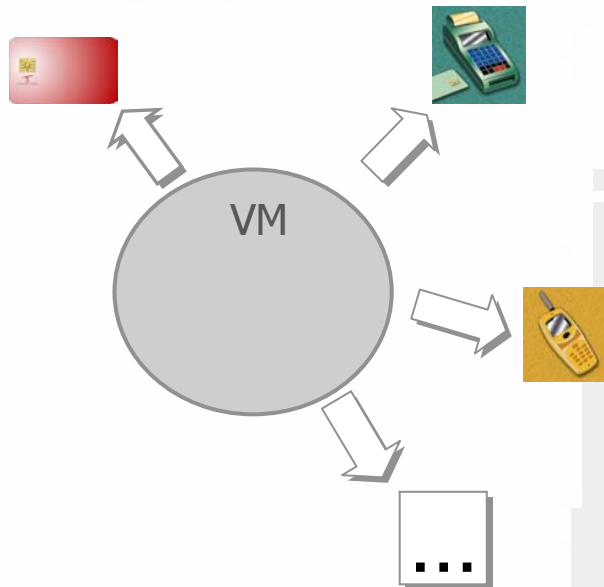
- Java card
- Midp/Stip
- OSGI
- ...

❑ Links to externals

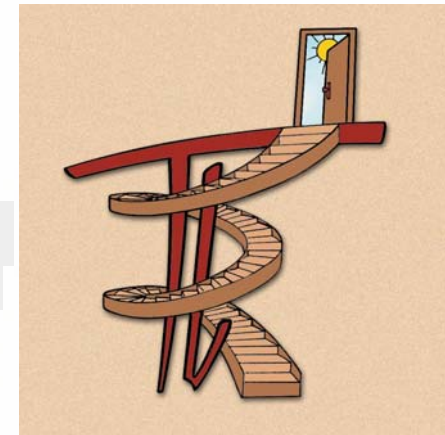
- Secure channels
- Automatic verification
- Key management

Capitalise on

Java Embedded Platforms



And a step by step security methodology



Java for embedded systems

❑ Cards

- Java Card
- Interoperable
- Standard
- CC evaluation + Proprietary schemes

❑ Functional

❑ Standardisation

❑ Interoperable

❑ Security

❑ Terminals

- Functional
- Standardisation
- Proprietary schemes

❑ Gateways

- Functionnal
- Standardisation in progress
- Security to be defined



Java Cards : Where do we stand today?

- ❑ The Java Card™ technology brings unprecedented flexibility to embedded systems
- ❑ The market for Java Cards (banking, mobile telephony, ID, etc.) is now taking off very quickly
 - ➔ *Issuers need to be comforted about the platform's level of security*
- ❑ Java Cards can be secured at least as much as native cards
- ❑ Common Criteria EAL4 & EAL5 certified Java Cards today exist on the market

Java Card vs. Native OS

□ Java Card key benefits

- “Write once, run anywhere”: the interoperability concept
- Post-issuance application download capability

□ The Java Card language structure provides support for:

- Basic properties like firewall, access control, confidentiality, etc.
- Management of sensitive information like keys, PIN codes, etc.
- High level properties like sharing, information flow control, etc.

Java Card vs. Native OS

- ❑ **Today's certificates (EAL4/EAL5) show that Java Card security can be mastered**

- Basic Configurations with limited functionality (no post-issuance download, no DAP, etc.)

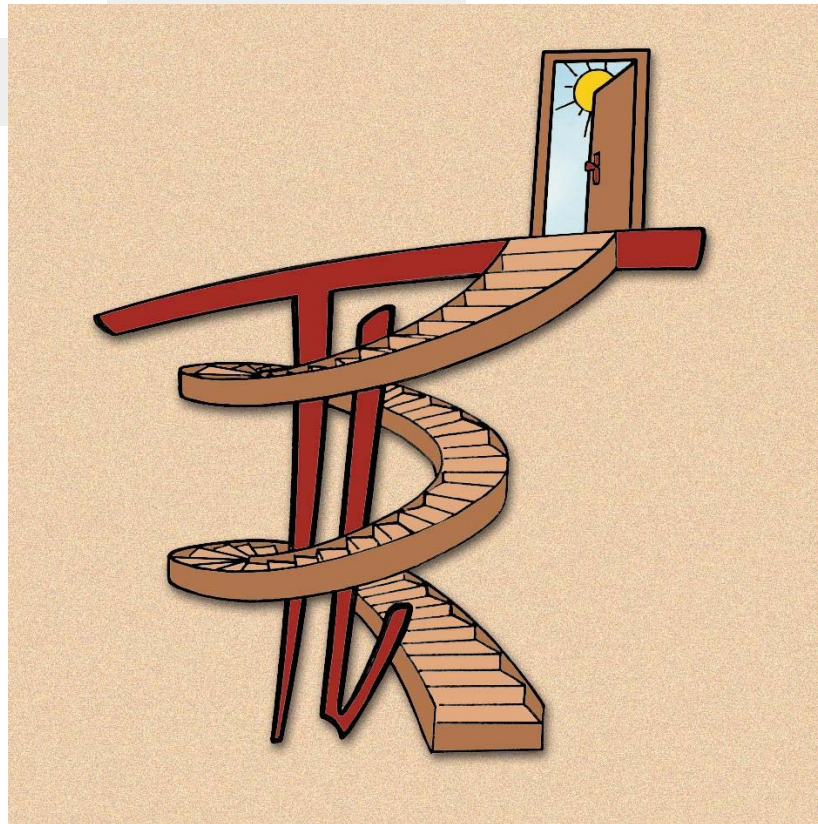
- ❑ **Issuer requirements are getting stronger**

- ❑ **It is worth climbing further up the security ladder:**

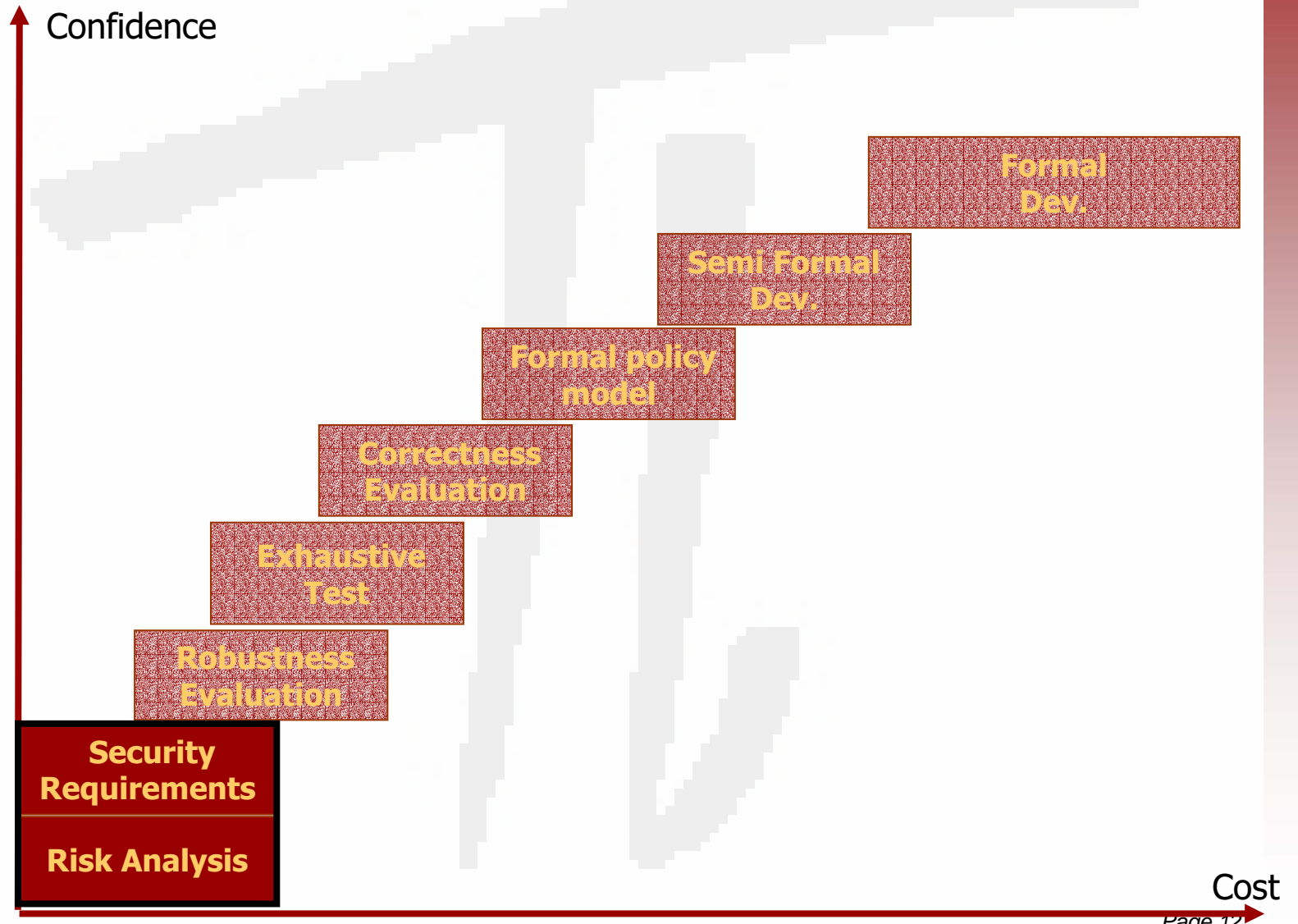
- Java Card Specifications are accessible
- Efforts can be shared and reused
- Investment is worthwhile: challenging parts can be formalized
- Lab expertise is growing larger



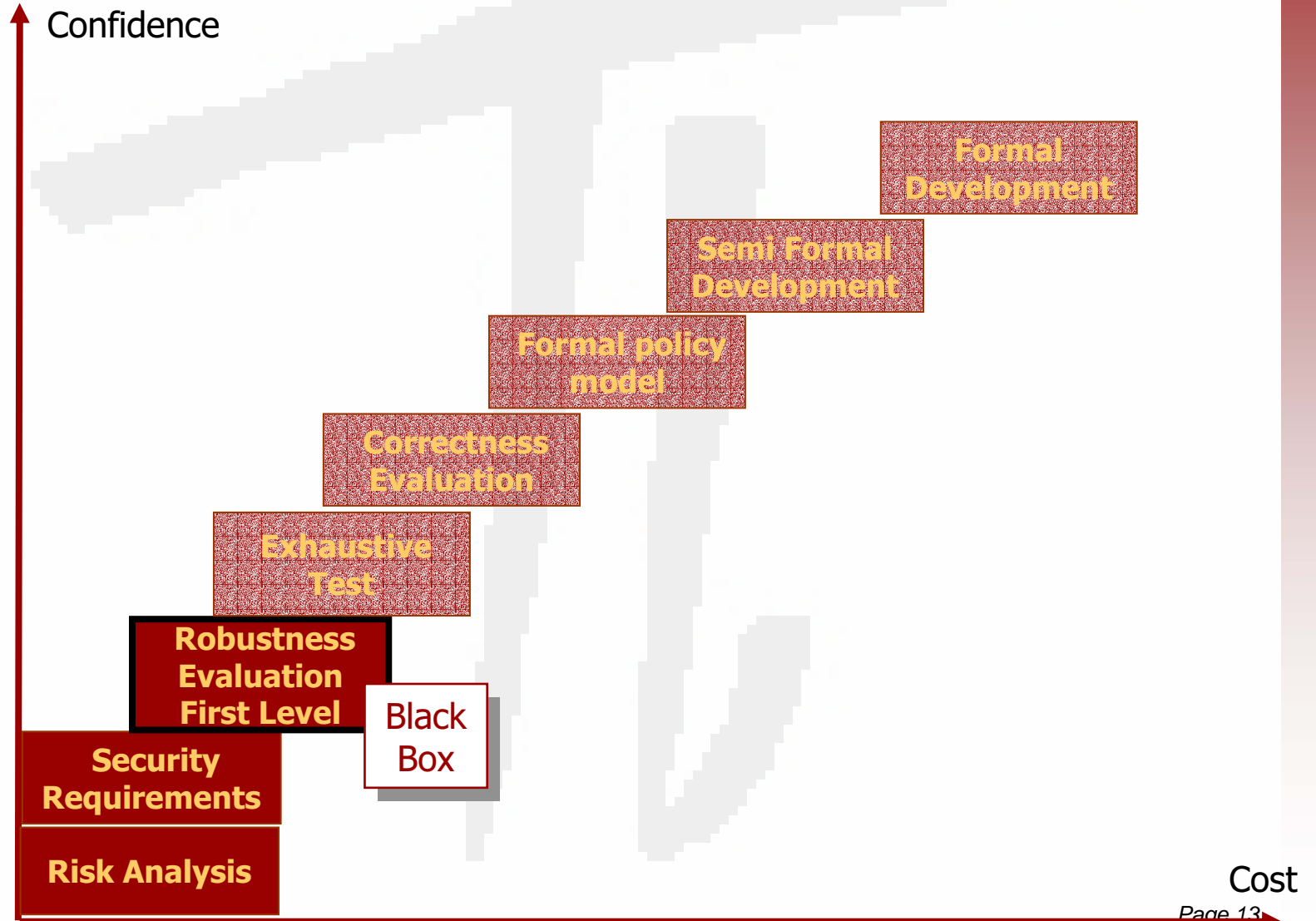
Security : a step by step approach



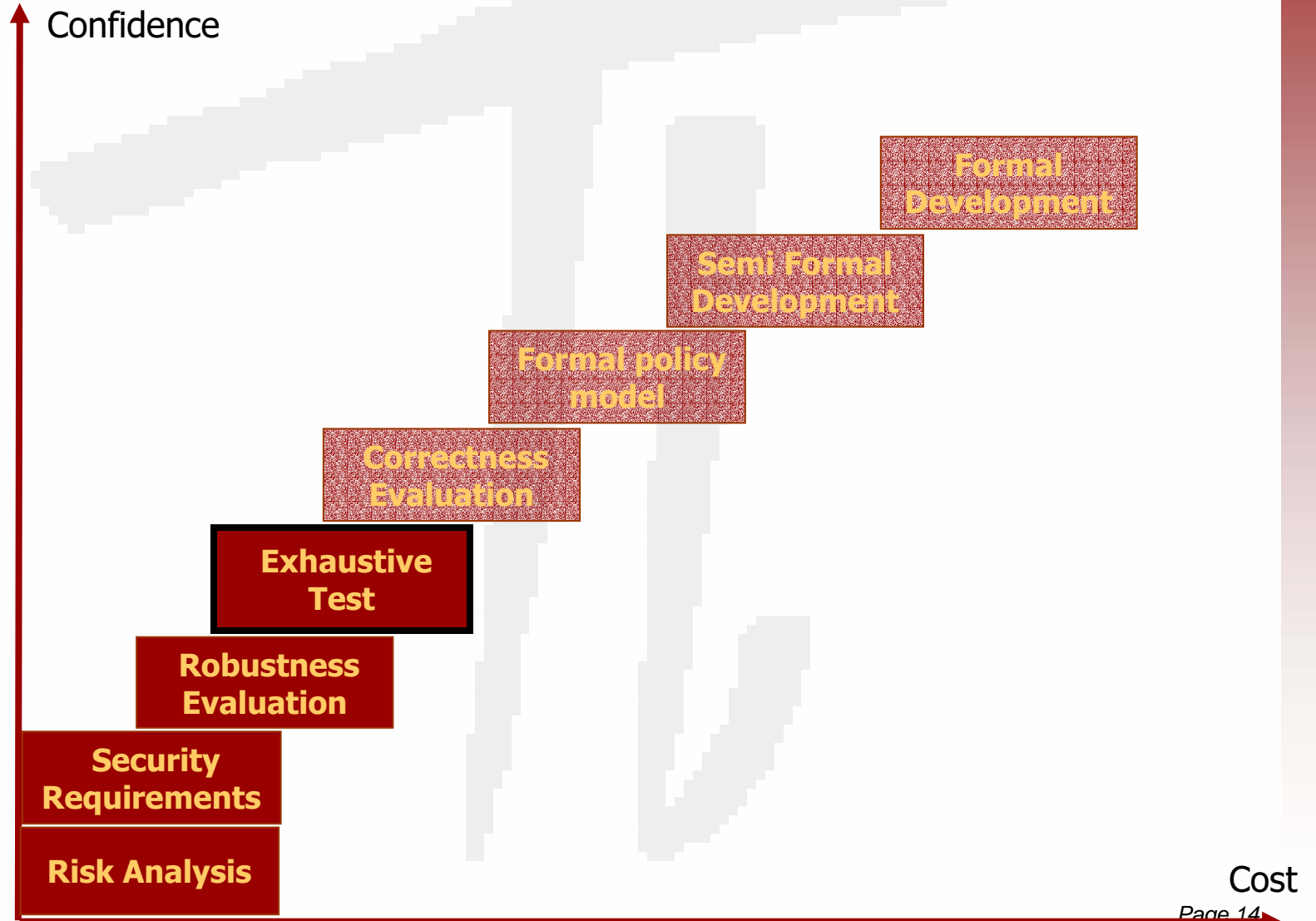
First step: What is my risk ?



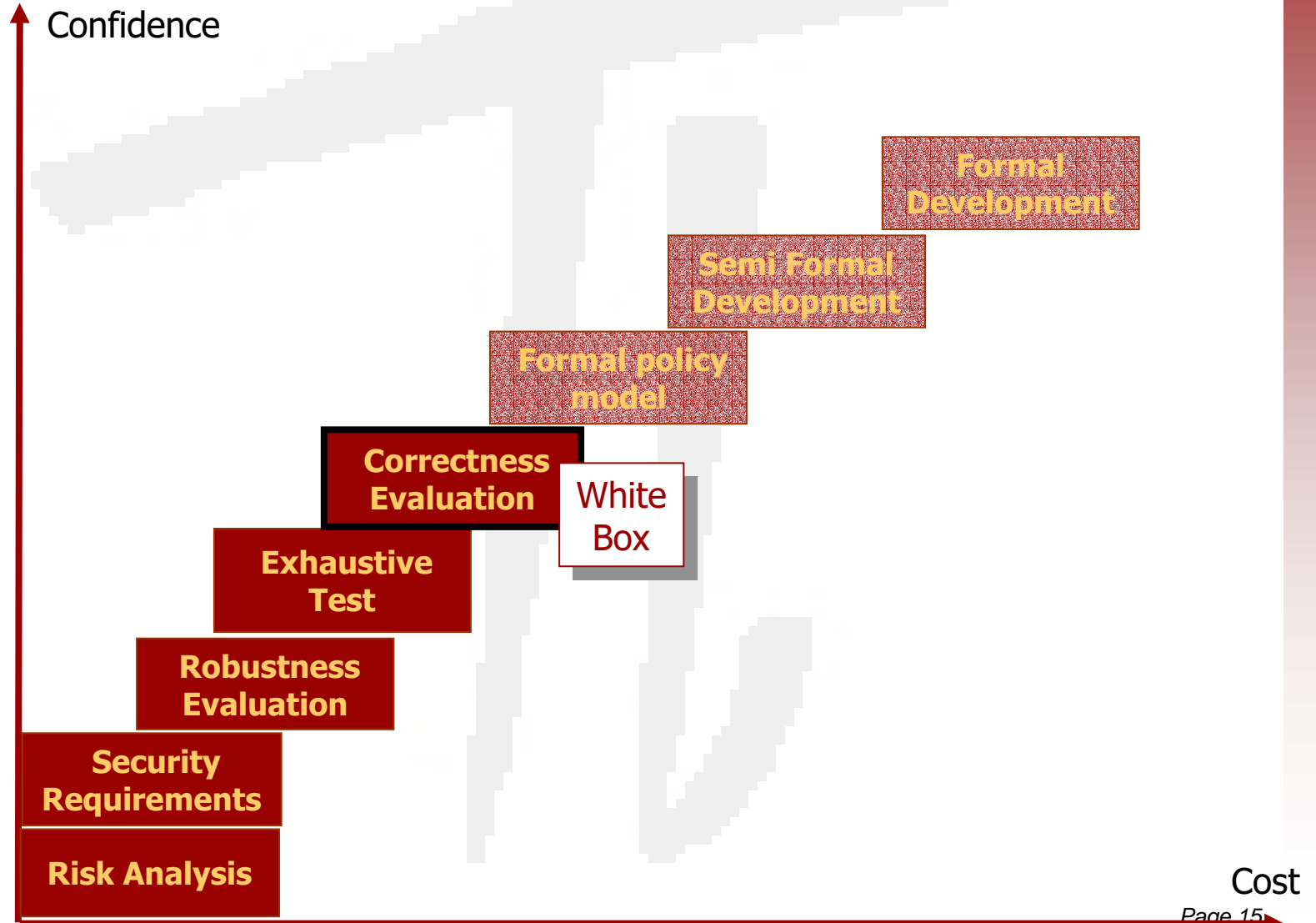
Second step: How robust is it ?



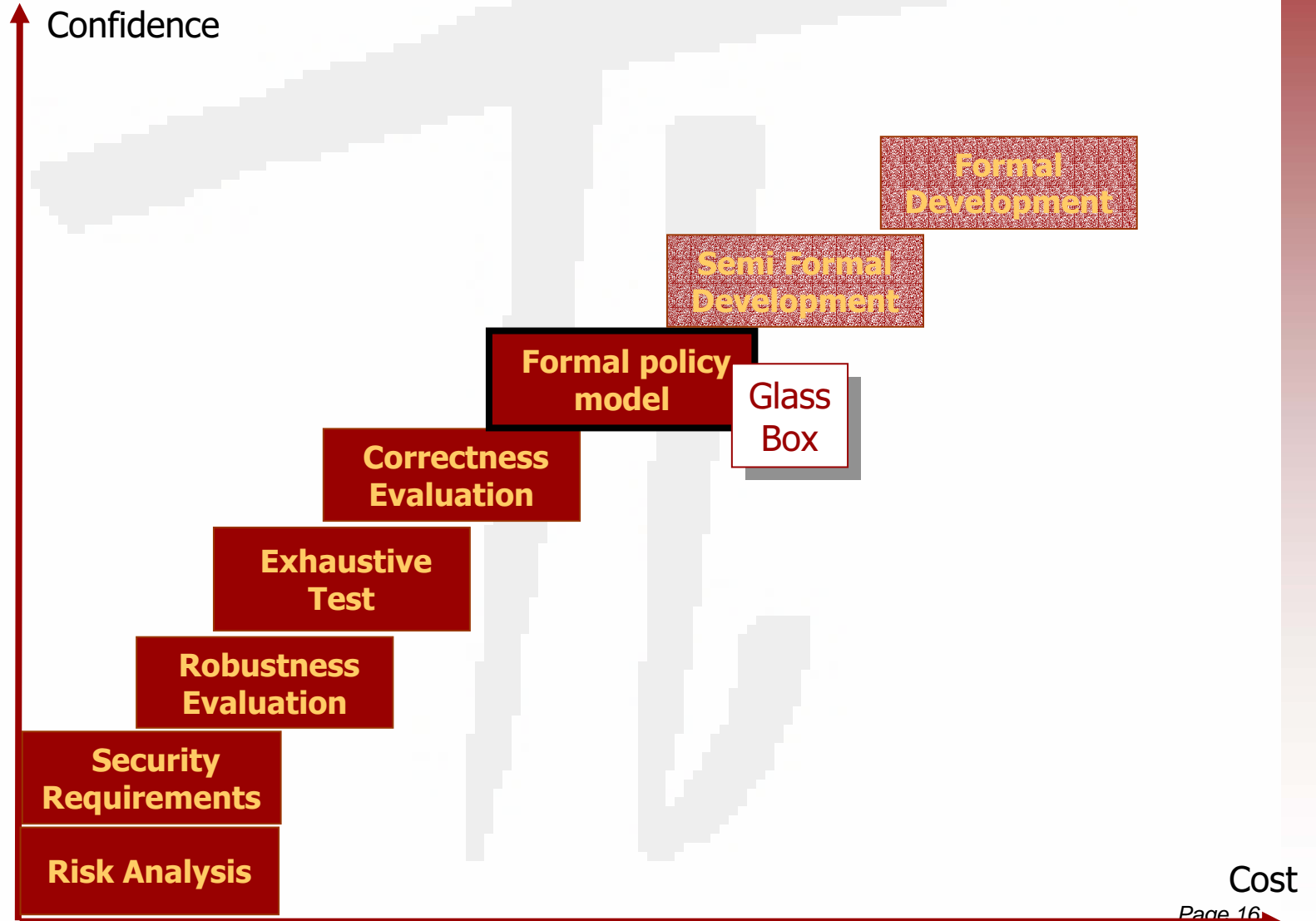
Third Step: Does it work ?



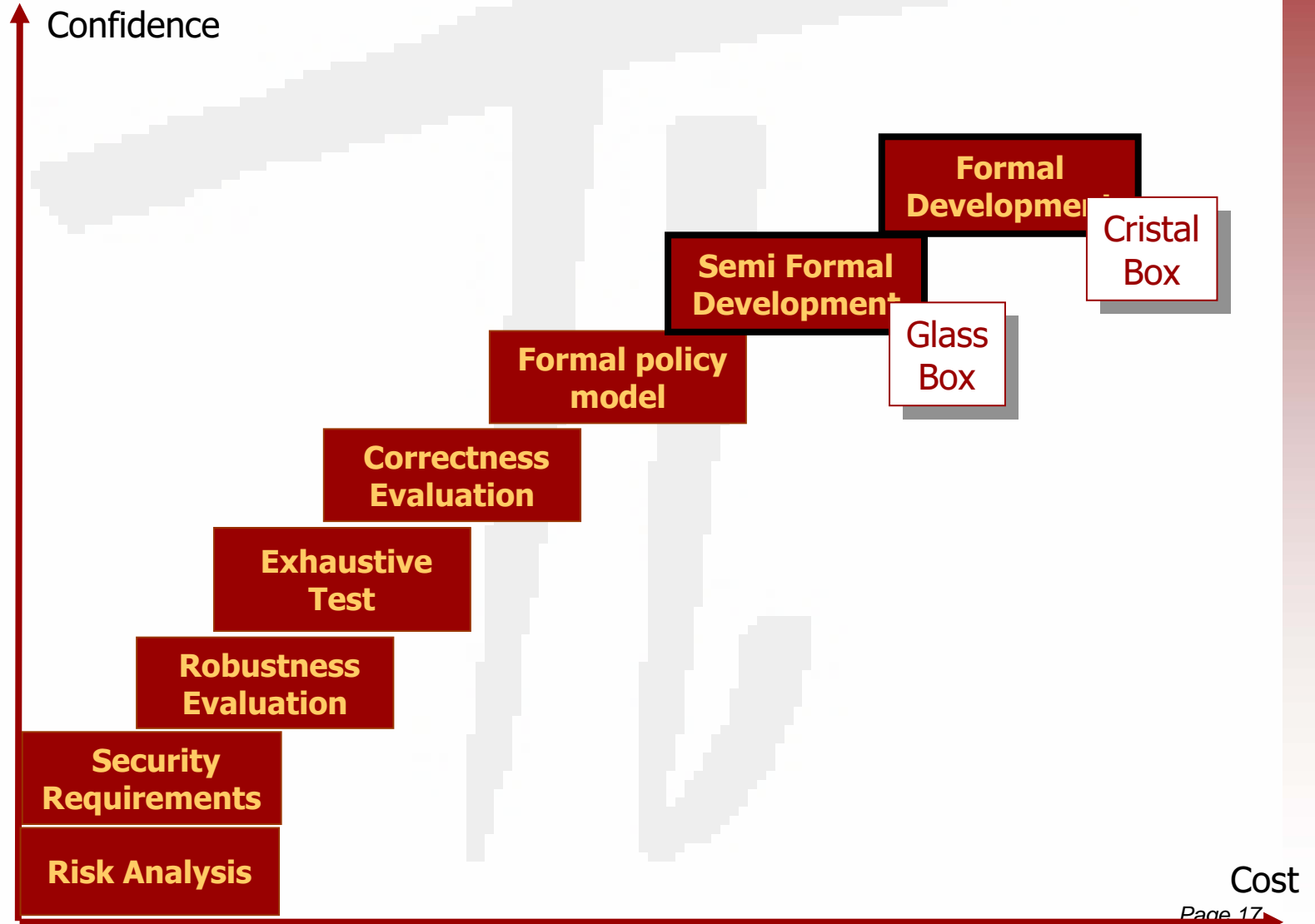
Fourth Step: How did you make it ?



Fifth Step: Prove it !



Furthering semantic analysis Question ?



A stepwise approach

CC Level

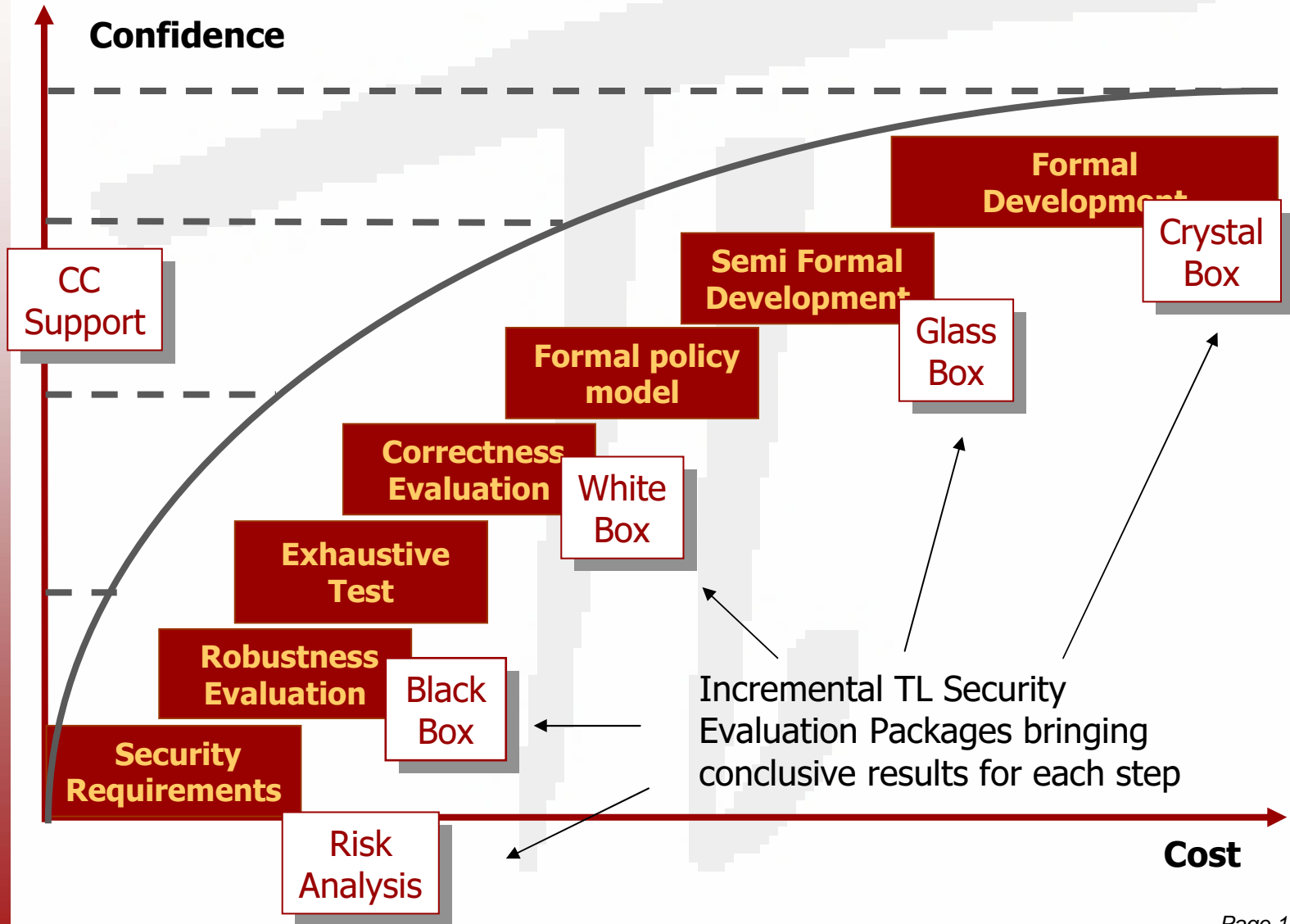
Confidence

EAL7

EAL5

EAL4

EAL1





Risk Analysis

Choose the right security level for the right cost

Inputs

- Documentation of the product
- Use cases and scenarios

Analyze

- The product in its environment
- Possible threats vs possible countermeasures

Outcome

- A report presenting risk scenarios depending on
- Assets protected by the product
 - Conditions of use
 - Protections supported by the product or by its environment





Card Example

- ❑ **One specification for all but**
 - On closed cards we dont need loading capabilities
 - If application dont share anything there is no need of a sophisticated firewall mechanism
- ❑ **Depending on the actors that can load an application on the card, avoid malicious applet can be done by**
 - organisationnal measure
 - Off card automatic procedures
 - Off card certification
 - On card defensive approach
- ❑ **Applications may sensitive to**
 - Confidentiality (health)
 - Integrity (Payment)
 - Replay (DRM) ...

Terminals Example

Composition

- Peripherals : screen, keyboard, sc reader, biometrics, external memories, ...
- Processors : various security levels

On-line / Off-line security

Keeping secrets ?

Cryptography

Tamper evidence / Tamper resistance : what for ?

Connecting services

❑ Domotique : heterogeneous needs

- House protection (alarm, ...)
- Telephone
- Video on demand
- House control (washing machine, ...)
- Good, services purchasing

❑ Automobile

- Follow individual in car float : several persons using one car (taxi, renting, ...)
- Rescue
- Car park Paiement
- Goods reservation and purchasing : cinema, air line tickets,

❑ Need for interoperability and security

A unique methodology applied to various fields

❑ Inside one field Evolution of

- Functional need
- Attack means
- State of the art Countermeasures
- Attacker motivations

❑ Cross sectors problematics

- Multi application,
- Downloading capabilities
- Remote management

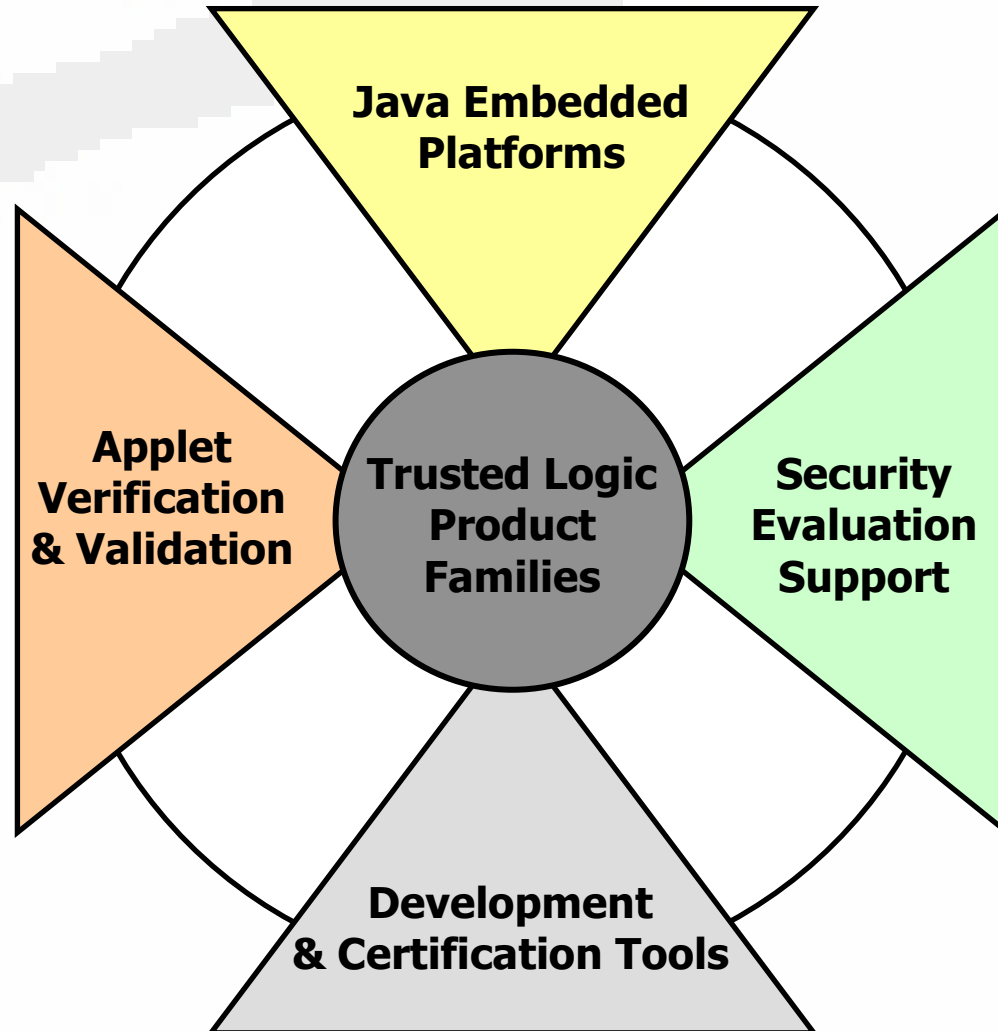
Conclusion :

Design an appropriate solution

- ❑ **Systematic approach based on the definition of a model that**
 - integrates assets, actors, assumptions, organisationnal policies
 - Can simulates scenarios

- ❑ **Objective**
 - Understand and challenge security with a systematic approach
 - Be able to answer any what if question
 - Have an homogeneous security level
 - Be able to adapt the solution to the need.

Combine security components and expertise



Tools supporting Application security

