

Secure Software: Current Research

Jan Jürjens

Competence Center for IT Security
Software & Systems Engineering
Informatics, Technical University Munich



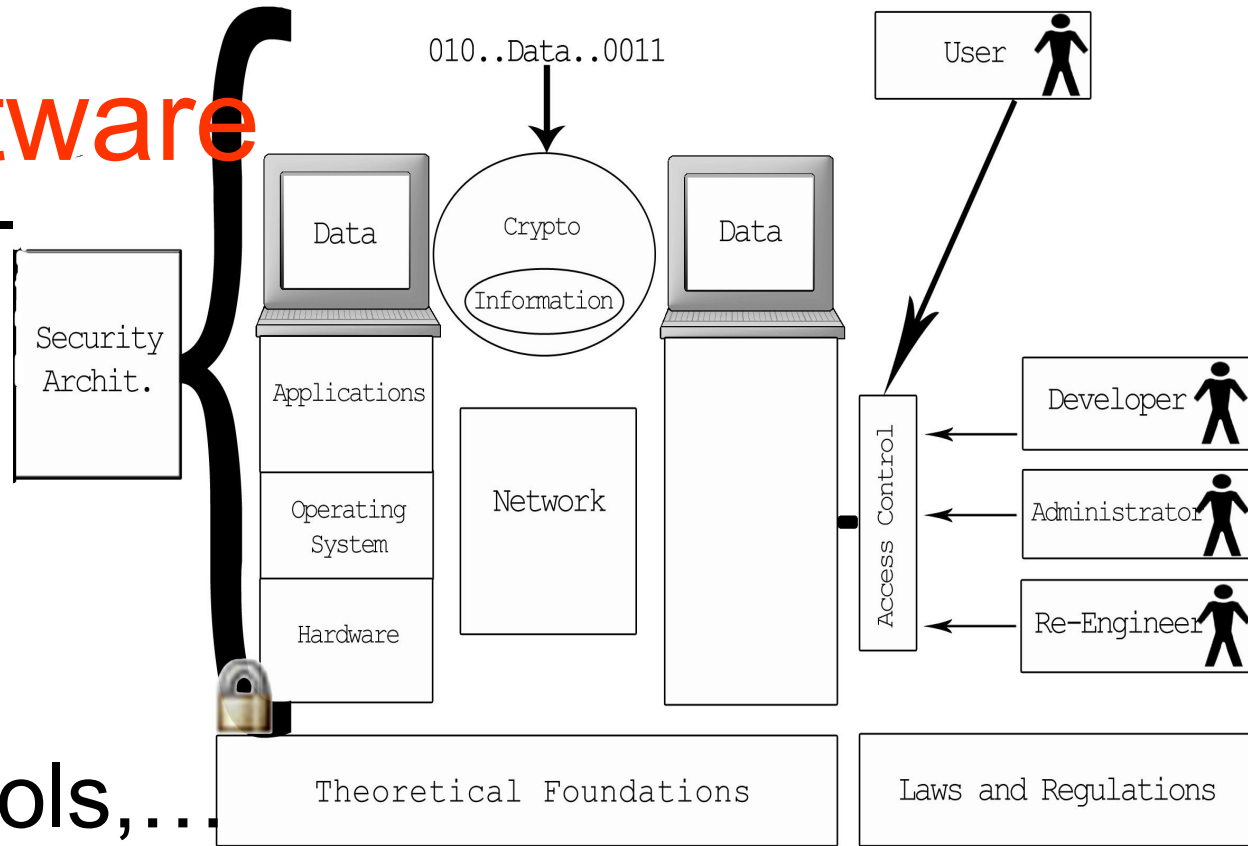
juerjens@in.tum.de



<http://www4.in.tum.de/~juerjens>

Secure Software

Today: strong
security mechanisms:
crypto, protocols,...



Still a problem: **Security requirements.**

- **Hard** to get right.
- How to **transform** into **secure systems** (out of (in)secure components) ?

Security Requirements

Security is **holistic** property:

- Attackers often **circumvent** (not: **break**) mechanisms.
- Rely on system **context**.

„Those who think that their problem can be solved by simply applying cryptography don't understand cryptography and don't understand their problem“
(B. Lampson / R. Needham).



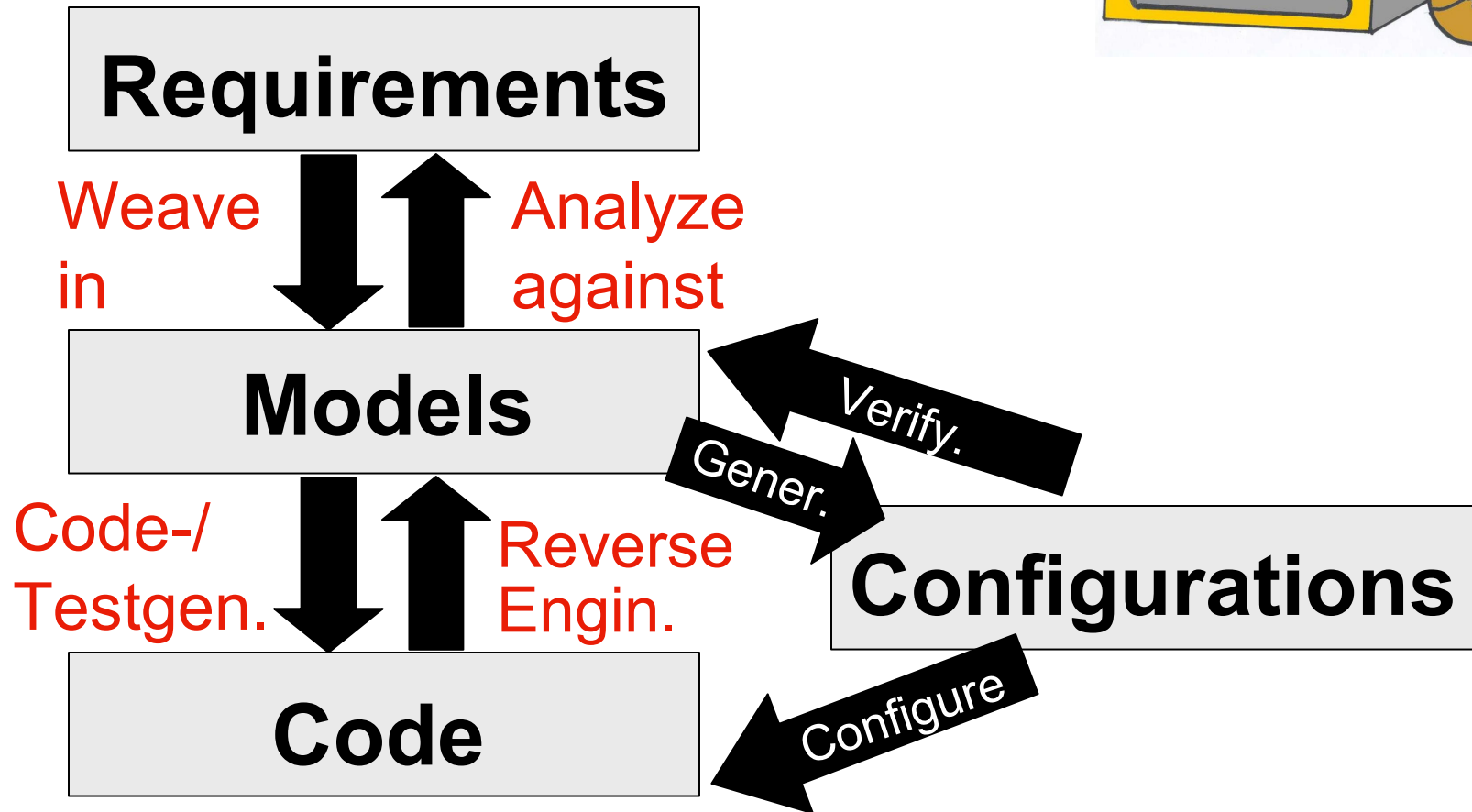
Analyze Artefacts for Security

To **enforce** security requirements:

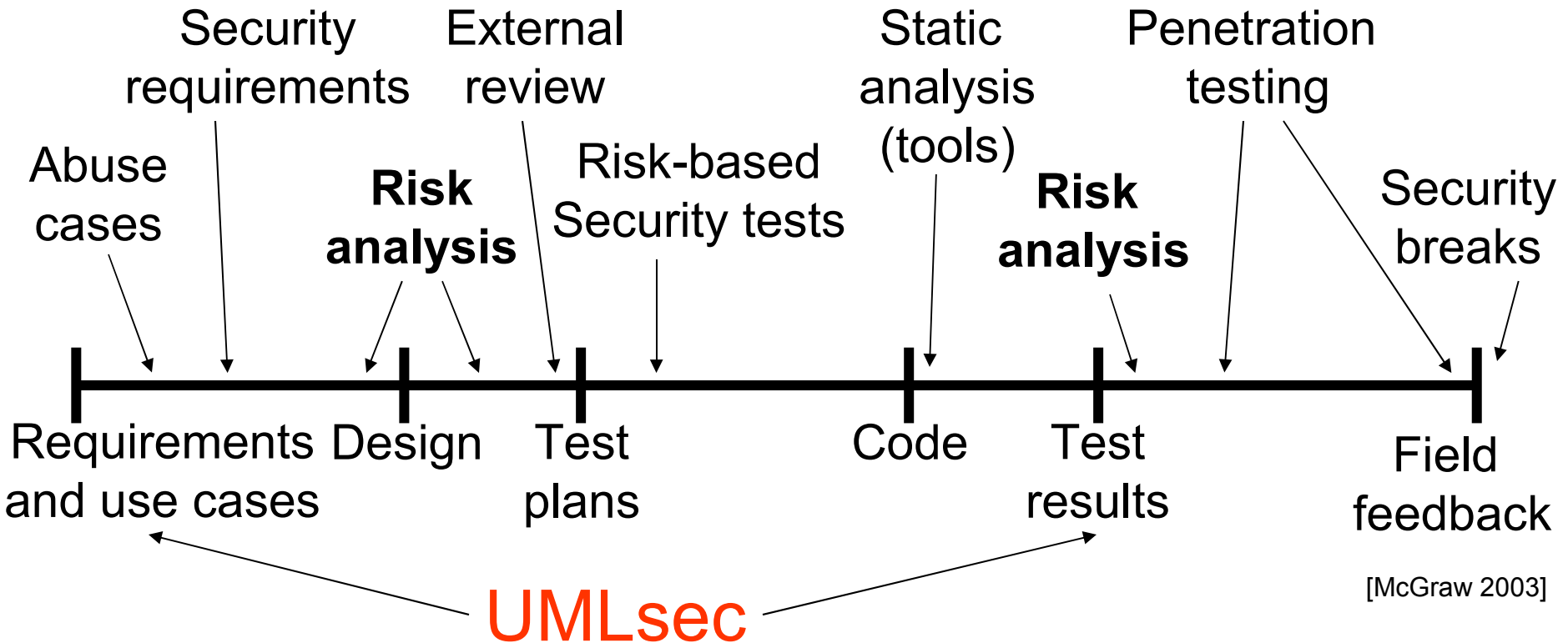
Extract **models** from **artefacts** in **development** and **use** of software and **analyze** against **security requirements**:

- specifications (e.g. UML models)
 - source code
 - configuration data
- ➔ **Tool-supported, theoretically sound, efficient automated security analysis.**

Model-based Security

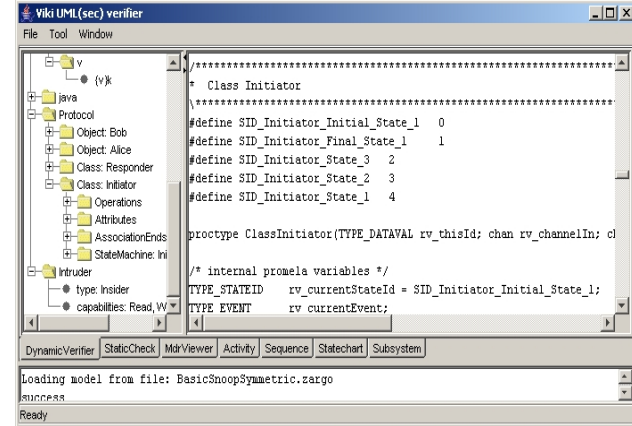


Secure System Lifecycle

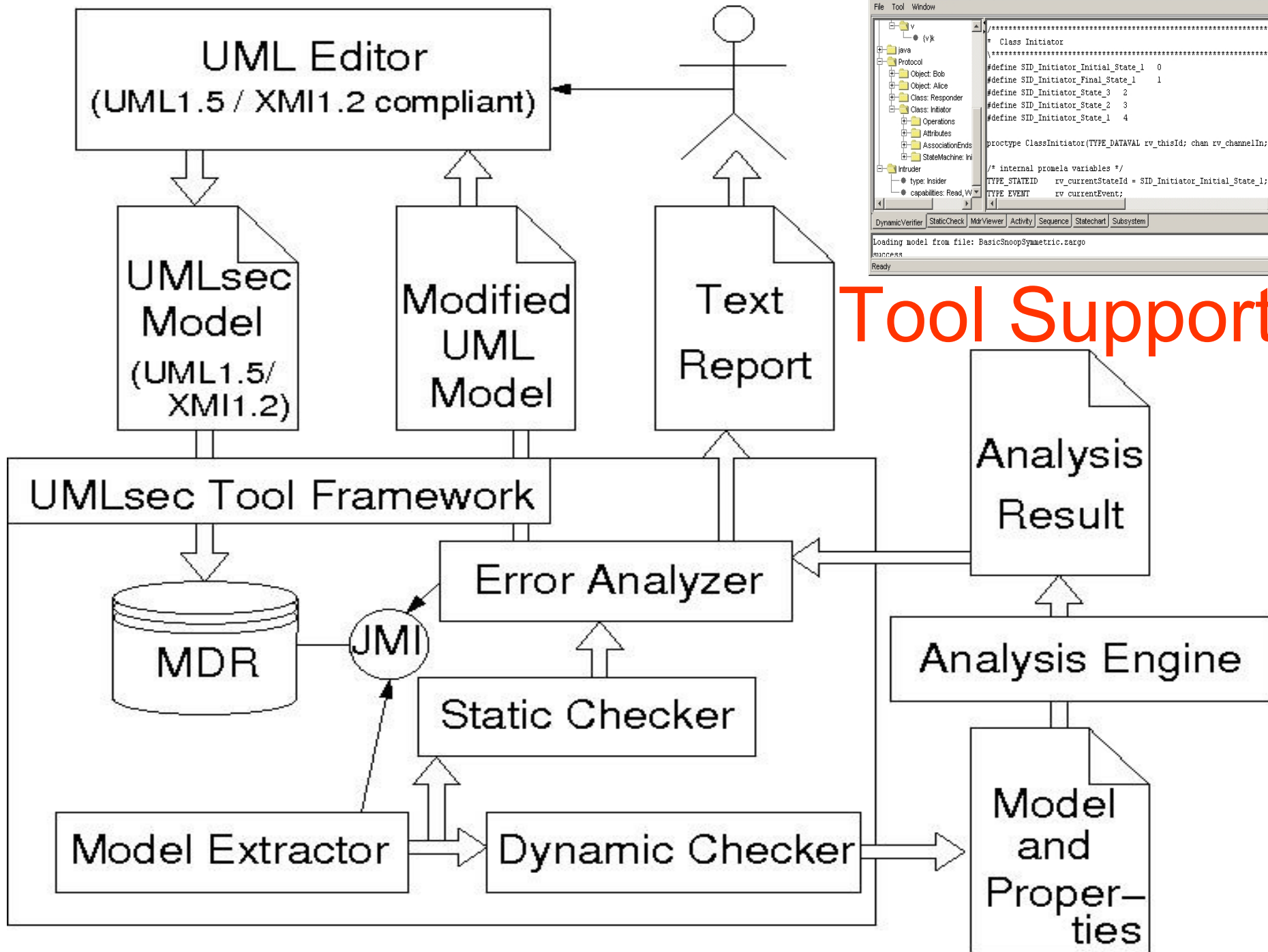


Design: Encapsulate prudent security engineering rules.

Analysis: Formally based, automated, efficient tools.



Tool Support



Security Analysis: Model or Code ?

Model:

- + earlier (**less expensive** to fix flaws)
- + more abstract → **more efficient**
- more abstract → may **miss attacks**
- **programmers** may **introduce** security **flaws**
- even **code generators**, if not formally verified

Code:

- + „the real thing“ (which is executed)
- **Do both**: verify **code against interface** spec.

Some Applications

Analyzed designs / implementations / configurations for

- biometry, smart-card or RFID based identification
- authentication (crypto protocols)
- authorization (user permissions, e.g. SAP systems)

Analyzed security policies, e.g. for privacy regulations.

T-Systems

Allianz

Deutsche Bank

HypoVereinsbank

CEPS™

BMW Group

msg systems

Münchener Rück
Munich Re Group

Bundesministerium
für Bildung
und Forschung

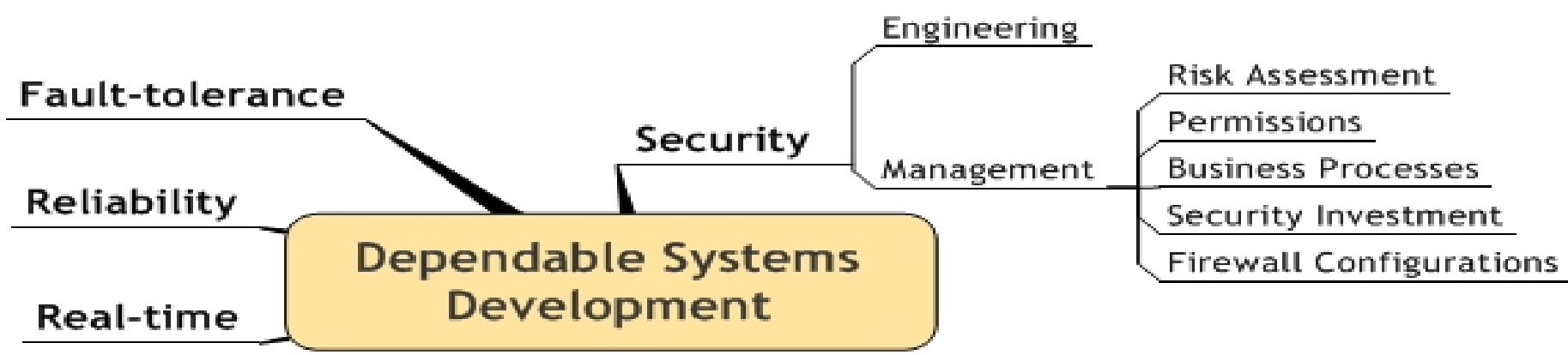
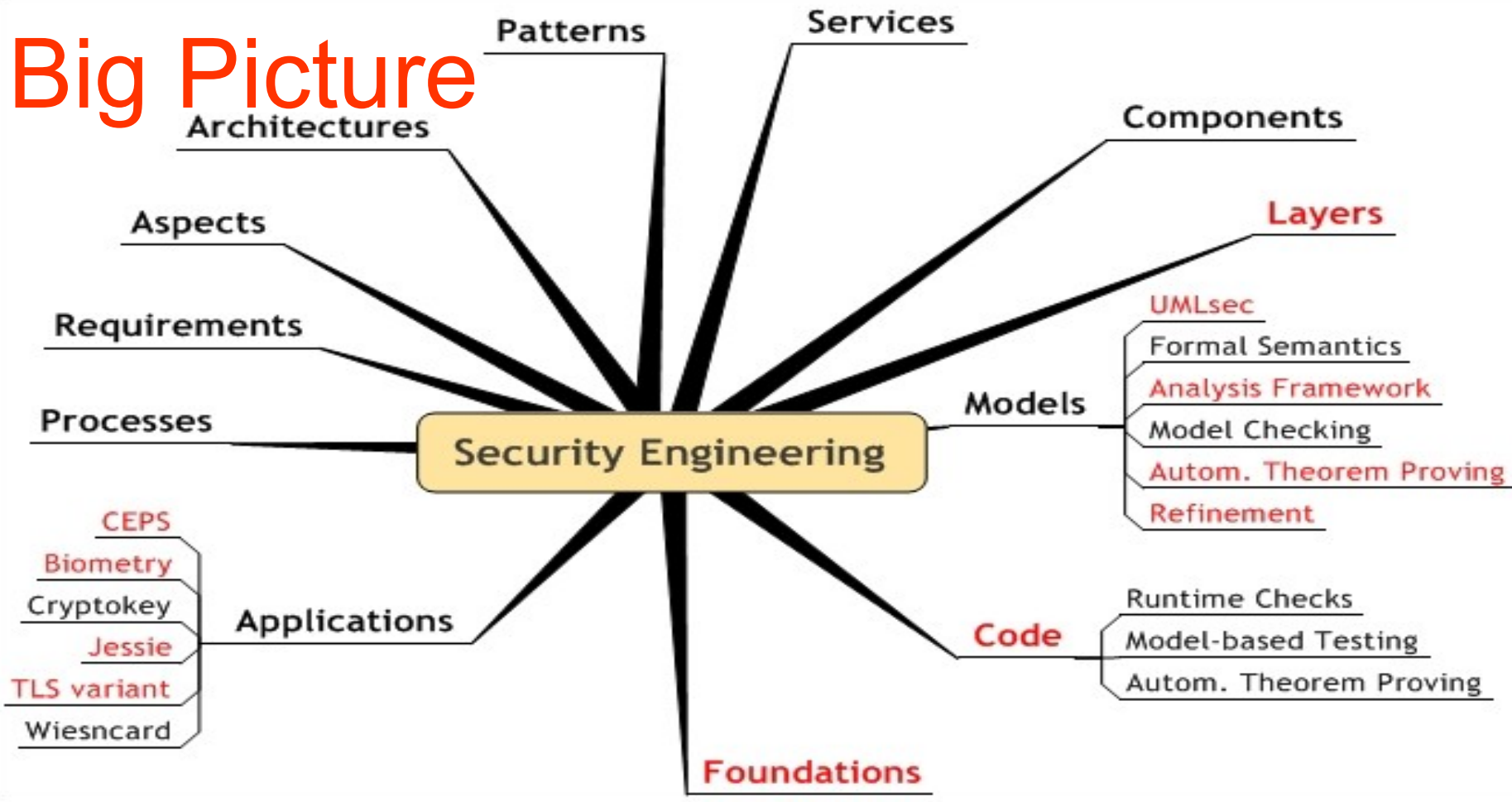
Bundesministerium
der Verteidigung

O₂

infineon

Bundesministerium
für Wirtschaft
und Technologie

Big Picture



GI FG Formale Methoden und Software
Engineering für Sichere Systeme (FoMSESS):
<http://www4.in.tum.de/~fomsess>

Questions ?

juerjens@in.tum.de

<http://www.umlsec.org>

