Model Based Security Testing
Selected Considerations

Keynote at SECTEST @ ICST 2011

Ina Schieferdecker, Jürgen Großmann, Axel Rennoch
Fraunhofer FOKUS

25 March 2011, Berlin

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Outline

- Sketch of Model-Based Security Testing
- Overview of DIAMONDS Project

Outline

- Sketch of Model-Based Security Testing
- Overview of DIAMONDS Project

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

Model-based security testing - Goals

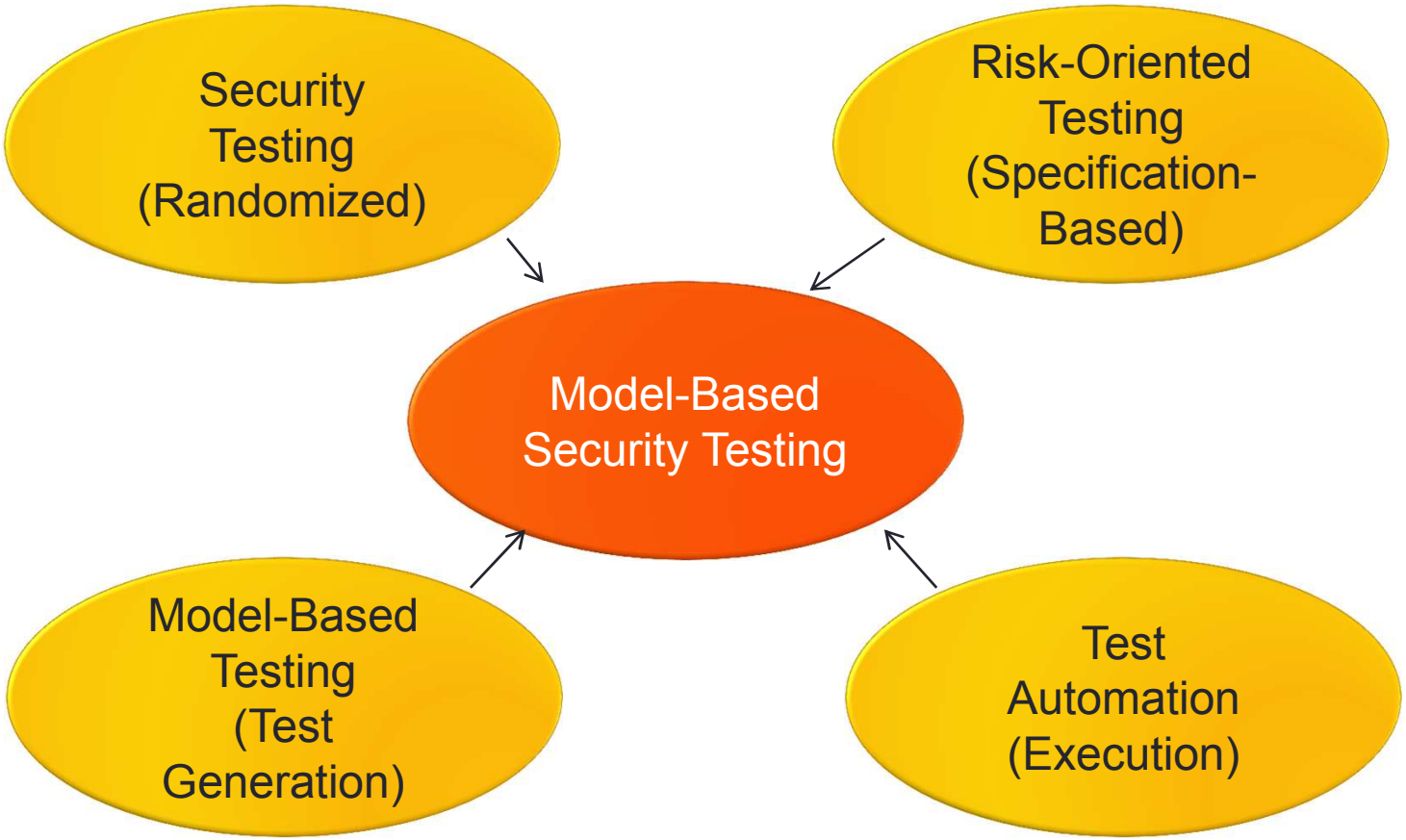
- Provide
  - Objective
  - Transparent
  - Repeatable
  - Automated
- security tests that focus on system specifications and related risks

Model-based security testing

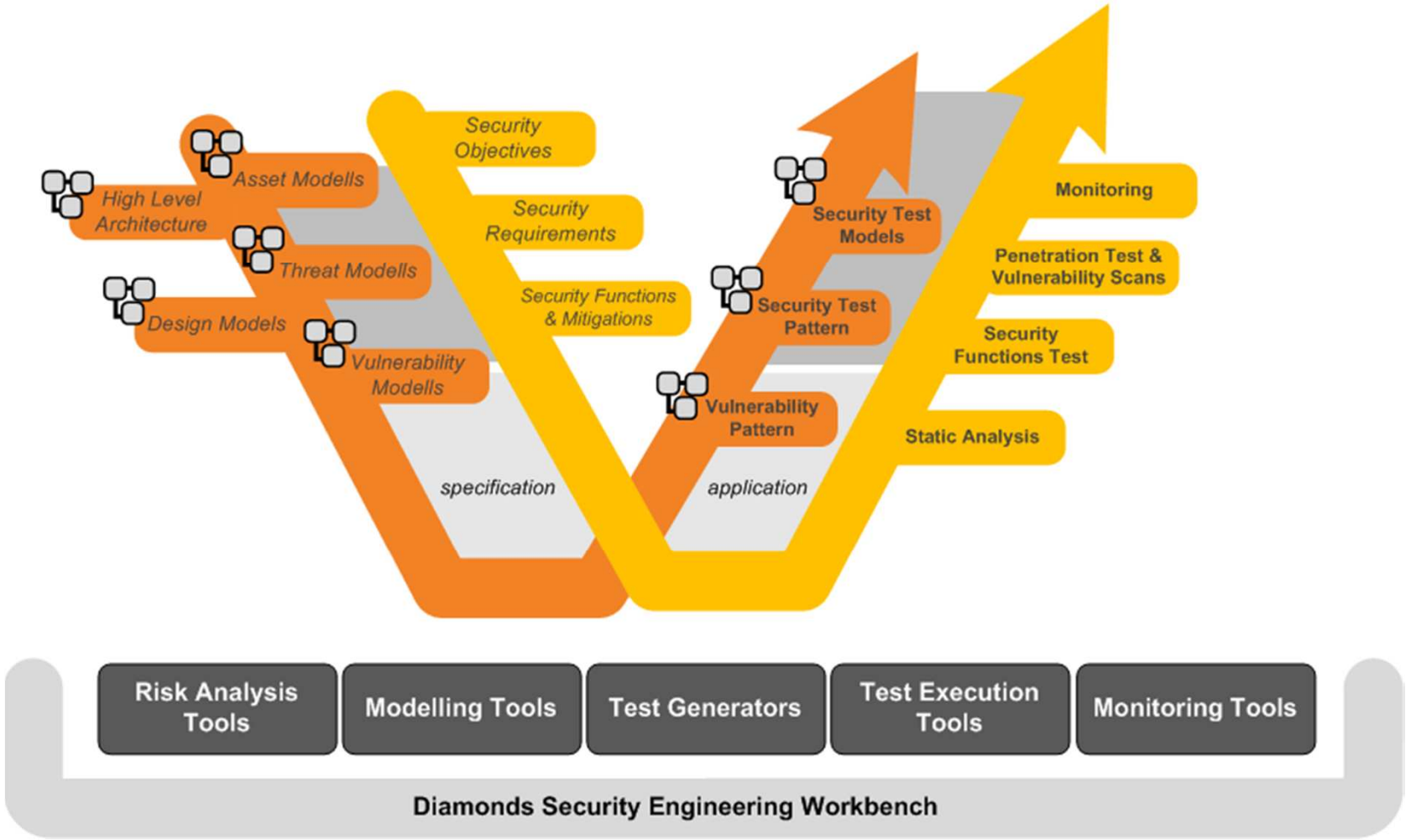
- Using functional system and test **models** with **security annotations**
  - determine the test architecture
  - derive test suites that cover modelled security functions and protocols
  - drive test-case selection and generation to cover critical security functions according to their criticality
- Derive test suites from **environment models** or system models that reflect the logical and physical environment
  - automated vulnerability search for complex system configurations (e.g. from deployment models)
  - systematic test generation from attack models. (e.g. abuse case or misuse case models) or environment models (e.g. protocols)
  - integrating risk models (threat and asset models) to identify, generate and select test cases

J. Jürjens, “Model-based security testing using umlsec: A case study,” Electr. Notes Theor. Comput. Sci., vol. 220, no. 1, pp. 93–104, 2008.
M. Zulkernine, M. F. Raihan, and M. G. Uddin, “Towards model-based automatic testing of attack scenarios,” in SAFECOMP, ser. Lecture Notes in Computer Science, B. Buth, G. Rabe, and T. Seyfarth, Eds., vol. 5775. Springer, 2009, pp. 229–242.

Combination of Approaches



Model-based Security Testing Process



Model-based fuzzing

- Fuzzing originally describe the generation of randomly generated test vectors (Miller et. Al. in the early 1990s)
- **Random fuzzing:** has close to zero awareness of the tested interface.
- **Mutation based fuzzing:** mutate existing data samples to create test data , breaks the syntax of the tested interface into blocks of data, which it semi-randomly mutates.

- **Model-based fuzzing:**
  - uses models of the input domain (protocol models, e.g. context free grammars), for generating systematic non-random test cases
  - in security testing purposes, the models are augmented with intelligent and optimized anomalies that will trigger the vulnerabilities in code.
  - finds defects which human testers would fail to find

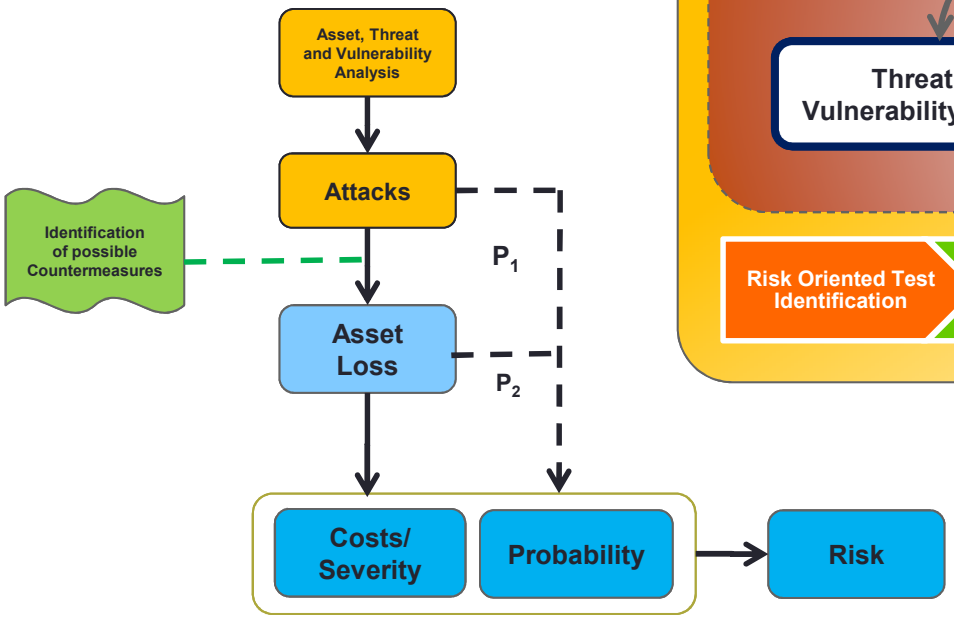
Ari Takanen, Jared D. DeMott, and Charles Miller: Fuzzing for Software Security Testing and Quality Assurance; ISBN 978-1-59693-214-2
Copyright 2008
PROTOS project. www.ee.oulu.fi/protos

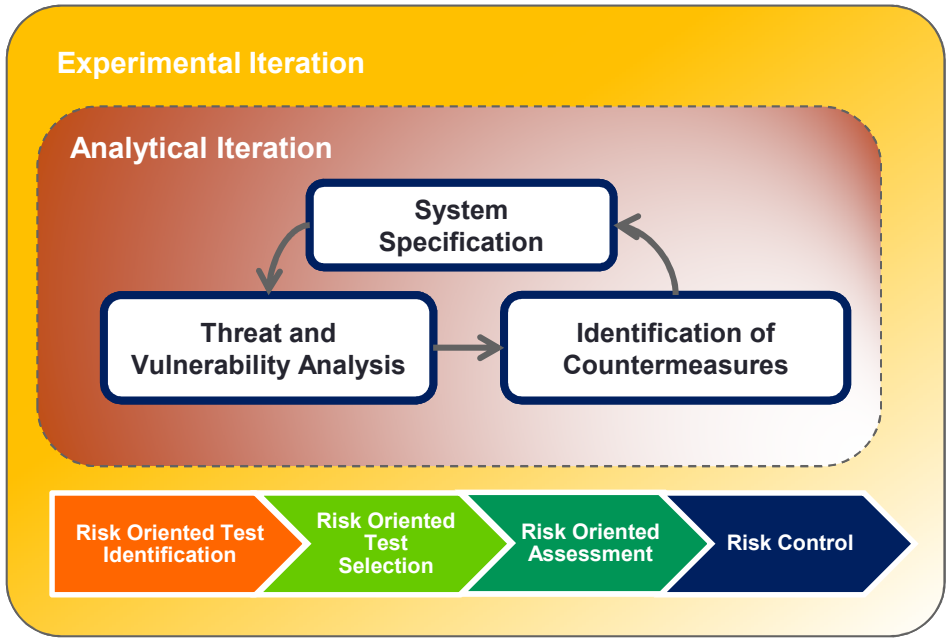
Risk-oriented testing

uses risk analysis results to optimize the test process

Qualification of:

- test identification
- test selection
- test assessment
- risk control

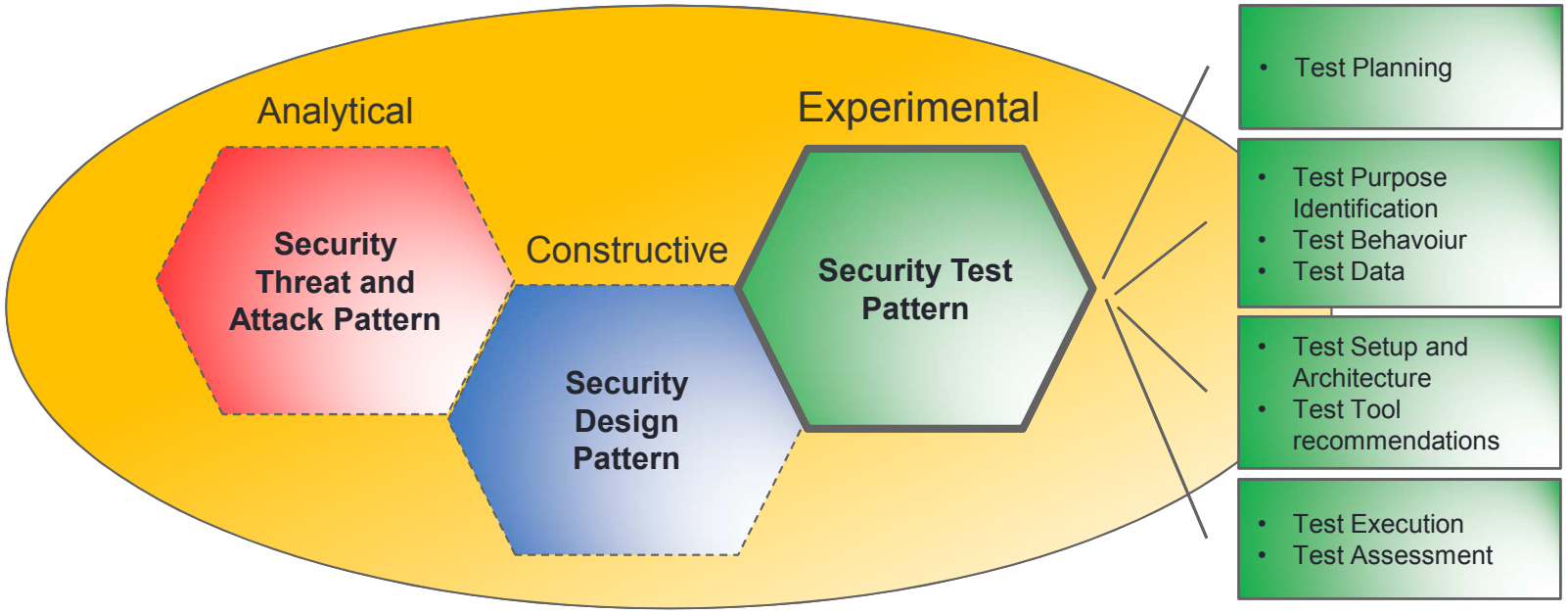




Stallbaum, H., Metzger, A., Pohl, K. *An Automated Technique for Risk-based Test Case Generation and Prioritization*.
Amland, S., *Risk Based Testing and Metrics: Risk analysis fundamentals and metrics for software testing.*
Souza, E.; Gusmao, C.; Alves, K.; Venancio, J. & Melo, R. *Measurement and control for risk-based test cases and activities*

Security test pattern

capture expert knowledge on what to test in which context



- **Pattern:**
  A (formalized) solution to a problem that arises (repeatedly) in a specific context

- **Security Threat and Attack Pattern:** vulnerability assessment, risk determination, attack pattern, requirements identification
- **Security Design Pattern:** security services, mitigations, design guidelines for countermeasures
- **Security Test Pattern:** security tests and assessments

Challenges in model-based security testing

Smart fuzzing is much more effective with respect to vulnerability detection:

random-based fuzzers detect **10%** of the vulnerabilities
mutation-based fuzzers detect around **50%** of the flaws.
smart or **model-based** fuzzing approaches can detect up to **80-90%** of the flaws

Challenge:
smarter models with effective strategies to support root cause analysis
reduce efforts in model development and maintenance

Takanen, DeMott and Miller: "Fuzzing for Software Security Testing and Quality Assurance"

Model-based security functional testing

systematically **combine functional and security aspect** for test generation and test assessment

Challenge:
translating high level security properties to code level test case specifications and vulnerabilities.
providing intuitive and industrial grade modelling paradigms

Systematic **integration of risk models** (threat and vulnerability models) with **test generation methods**

tuning test generation and execution efforts with respect to technical and business risks

Challenge:
finding adequate metrics and coverage criteria to effectively and trustworthy qualify risk and functional related testing aspects

Outline

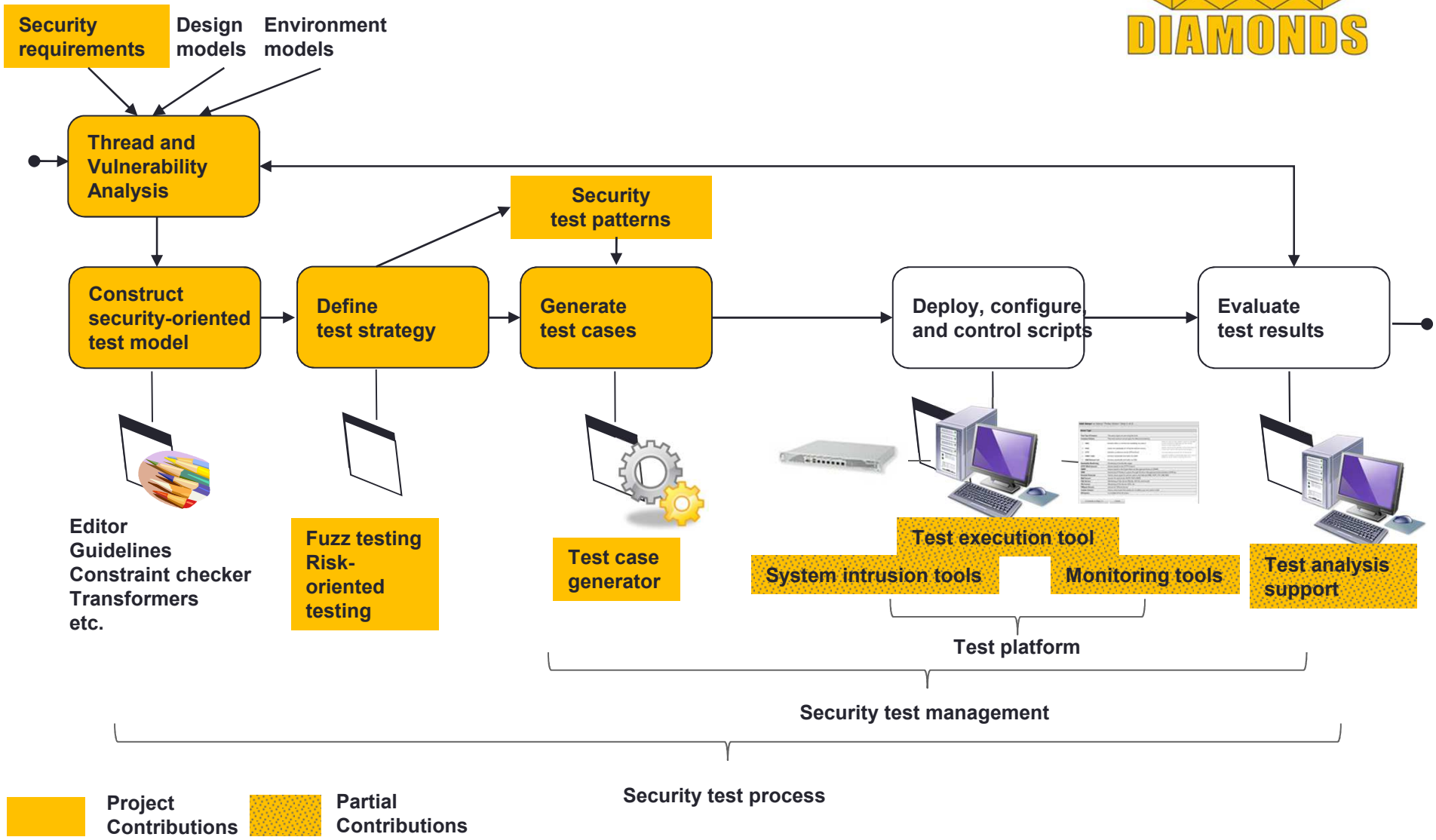
- Sketch of Model-Based Security Testing
- Overview of DIAMONDS Project

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

DIAMONDS Issues

- Security engineering is increasingly challenged by
  - the openness,
  - dynamics, and
  - distribution of networked systems
- Most verification and validation techniques for security have been developed in the framework of static or known configurations, with full or well-defined control of each component of the system.
- This is not sufficient in networked systems, where control and observation of remote (sub) systems are dynamically invoked over the network.

Model-based security testing - sketch



ITEA 2 Diamonds Project

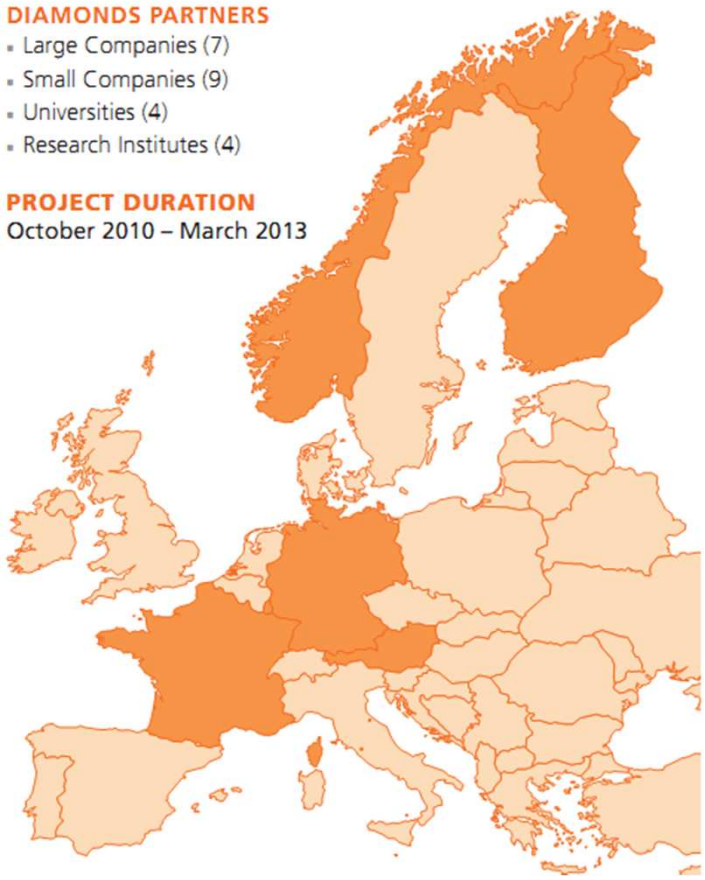
Diamonds will enable efficient and automated security testing methods of industrial relevance for highly secure systems in multiple domains (incl. e.g. banking, transport or telecommunication).

Business Value

- Multiple Domains
- Pre-Standardization Work
- Novel Integration of Testing, Security and Risk-Analysis

Expected Results

- Security Fault Models
- Risk-driven Security Testing Methodology
- Model-Based Security Test Techniques
- Security Test Patterns Catalogue



ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

DIAMONDS Case Studies

- Banking
- Smart Cards
- Industrial Automation
- Automotive
- Radio Protocols
- Telecommunication Infrastructures

ITEA 2
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

G&D: Banknote Processing Machines

Summary

- Security testing is
  - needed
  - challenging
- Systematic and automated security testing
  - **Model-based fuzzing** (smart fuzzing) using models on the data and behaviour that is being mutated (protocol models, data models) in such a way that the number of test cases are significantly reduced.
  - **Risk-oriented testing** uses risk analysis results for test identification, test selection and test assessment to prioritize and optimize the test process
  - **Security test pattern** catalogue capturing expert knowledge on what to test in which context (kind of system, security goal) and allow the reuse of this knowledge within a slightly different context

Contact

Prof. Dr.-Ing. Ina Schieferdecker

Phone: +49 30 34 63 7241
Mobile: +49 175 260 30 21
Email: ina.schieferdecker@
fokus.fraunhofer.de

FOKUS
Fraunhofer Institute for Open
Communication Systems FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

Tel: +49 (30) 34 63 – 7000
Fax: +49 (30) 34 63 – 8000

Web: www.fokus.fraunhofer.de