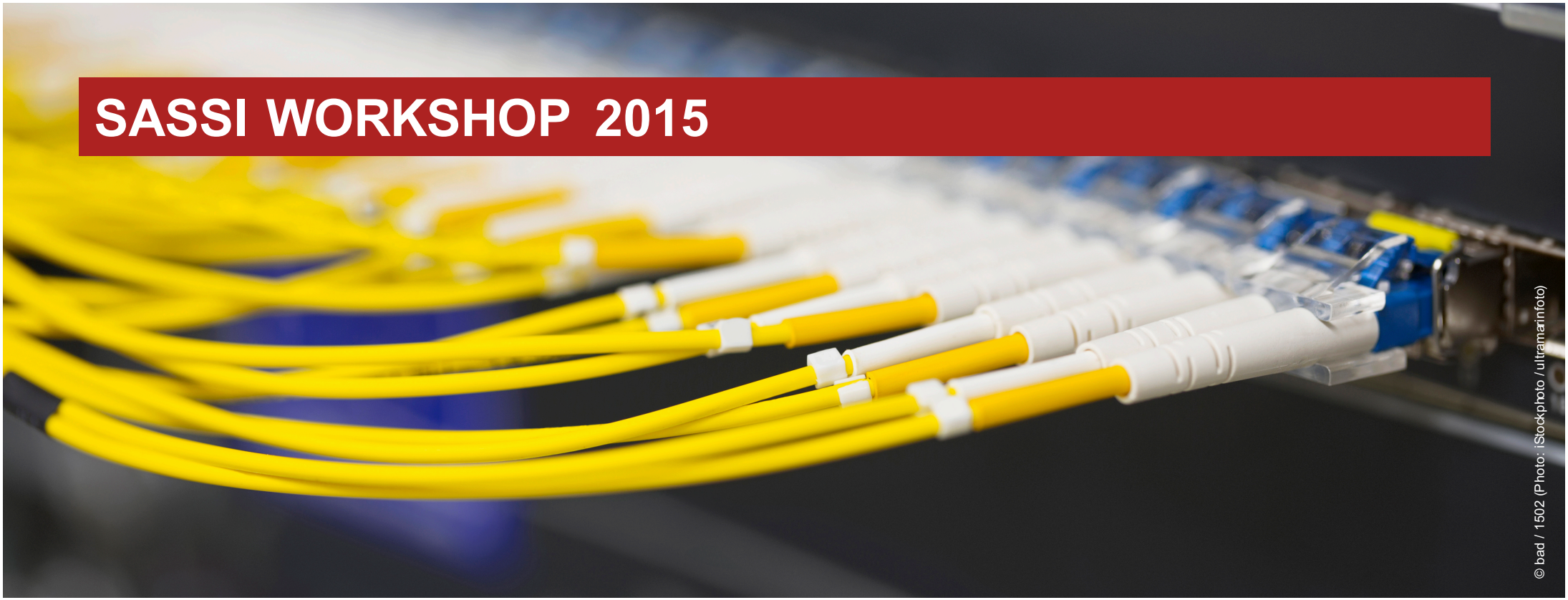
SASSI WORKSHOP 2015



Security Assessment for Systems, Services, and Infrastructures (SASSI 2015)

15 – 16 September 2015

Fraunhofer FOKUS

SASSI '15

INTRODUCTION

Security Assessment for Systems, Services and Infrastructures

Mobile devices, industrial equipment and facilities, smart grids, and even vehicles are connected via the Internet and becoming accessible and thus vulnerable to security breaches and hacker attacks. Software that runs this kind of system is exposed to a large number of different threats that pose special requirements on the quality and robustness of the software. These requirements can only be identified and met if security and privacy risks and their impact are systematically considered already during the early phases of the software development and quality assurance processes. A systematic and capable security risk and quality assessment program and its tight integration within the software development life cycle are key to building and maintaining secure and dependable software-based infrastructures. The SASSI workshop will provide a forum to discuss innovative approaches to security assessment, security testing and security certification for software-based systems. Experts from industry and academia will present and discuss their solutions to key issues like legal-risk analysis, security risk analysis, risk-based engineering, vulnerability testing, model based security testing, standardization, and certification. The workshop has a special focus on the interaction between innovations and industrial requirements, especially when security meets the demands of cost efficiency and scalability. The contributions originate from industrial practice and are complemented by industry grade research results from national and international research projects.

Fraunhofer FOKUS

SASSI '15

DAY 1 – TUESDAY, SEP. 15, 2015

KEYNOTE

SESSION 1: SECURITY RISK & COMPLIANCE ASSESSMENT

Fraunhofer FOKUS

SASSY '15

DAY 1 – TUESDAY, SEP. 15, 2015

SESSION 1: SECURITY RISK & COMPLIANCE ASSESSMENT

Fraunhofer FOKUS

SASSY '15

DAY 2 – WEDNESDAY, SEP. 16, 2015

SESSION 2: SECURE SOFTWARE DEVELOPMENT

Fraunhofer FOKUS

SASSY '15

DAY 2 – WEDNESDAY, SEP. 16, 2015

SESSION 3: SECURITY TESTING AND VALIDATION

Fraunhofer FOKUS

SASSY '15

SASS15

Living risk-based security at SAP, the solved challenges and the open ones

Paul El Khoury, SAP

Abstract:

SAP as the world 3rd largest software company offers solutions running in Mobile, Cloud and On Premise environments. As market leader for business applications, SAP shares the responsibility with customers and partners for securing its solutions. The SAP Secure Software Development Lifecycle is a risk-based process used to ensure a software is free of known vulnerabilities and guaranteeing the appropriate level of security for shipped products. The security risk assessment parts of this process, namely SECURIM and Threat Modeling, used per product to identify and manage product-specific security risks, define the targeted level of trust and build a security test plan. This talk will detail the materialization of these methods at SAP worldwide and highlight the next upcoming challenges with examples from Cloud and Internet of Things scenarios.

Vita:

Dr. Paul EL KHOURY joined SAP SE in 2006 and is currently co-owner of the SAP Product Standard Security. He leads the Product Security Risk Identification and Management as part of the SAP Secure Software Development Lifecycle and is an SAP security evangelist. Prior, Dr. EL KHOURY's major contributions were leading the SAP Threat Modeling methodology, co-defining the secure storage on device used by all SAP mobile applications and holding the position of governor of the SAP patch day from its pilot phase until it was rolled out to customers. He received his MSc and his Ph.D. in Computer Science from the Université of Claude Bernard Lyon 1. He has authored various scientific publications and patents in the field of software security.

Fraunhofer FOKUS

SASS '15

Living risk-based security at SAP, the solved challenges and the open ones

Dr. Paul El Khoury – CISSP
Co-Owner of SAP Product Standard Security, SAP SE

September 2015

Security Assessment for Systems, Services, and Infrastructures (SASSI 2015)

SAP – Helping the world run better!

For More than 40 Years, SAP Has Helped the World Run Better and Improve People's Lives

For the world

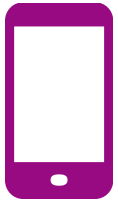
74%

of the world's transaction revenue touches an SAP system

For business

98%

SAP customers represent 98% of the top 100 most valued brands in the world



For you

97%

Mobile solutions from SAP reach 97% of the world's mobile subscribers via text messaging

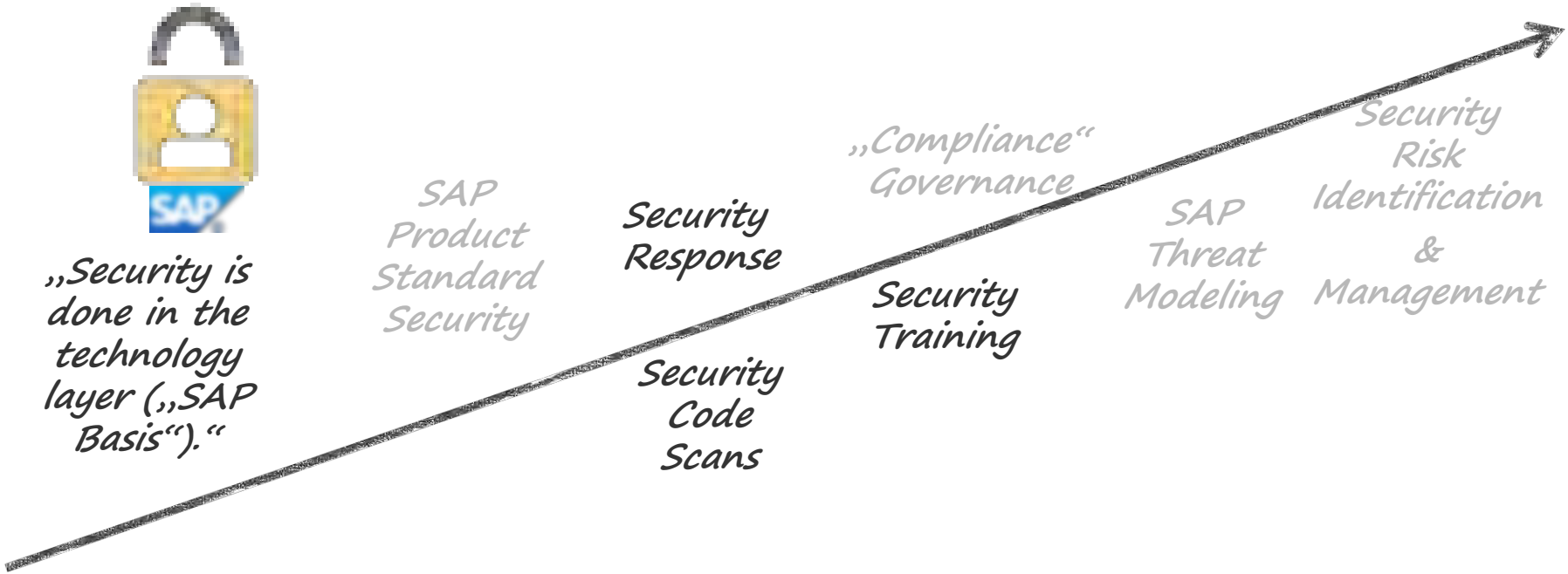
Who am I?

- ✓ **Joined SAP in 2006**
- ✓ **Holds a Ph.D. in Computer Science from the Université of Claude Bernard Lyon 1**

- ✓ **Is currently co-owner of the SAP Product Standard Security**
- ✓ **Leads the Product Security Risk Identification and Management**

- ✓ ***Earlier:***
  - ✓ Lead SAP Threat Modeling methodology,
  - ✓ Co-defined the secure storage on device used by all SAP mobile applications
  - ✓ Have held the position of governor of the SAP patch day

We have come a long way …



Awareness … Needs … Tools … Skills … Accountability …

We have come a long way …

Awareness … Needs … Tools … Skills … Accountability …

SAP Product Innovation Lifecycle an over-simplified view

The past

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool

*For every deviation a description of the **risk** taken and action item with a name and a date… are added to the tool supporting the PIL*

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool
- Development Teams **executes** on the plan

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool
- Development Teams **executes** on the plan
- Development Teams may have **implemented** or **deviated** from their agreed plan

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool
- Development Teams **executes** on the plan
- Development Teams may have **implemented** or **deviated** from their agreed plan

*For every deviation a description of the **risk** taken and action item with a name and a date… are added to the tool supporting the PIL*

SAP Product Innovation Lifecycle an over-simplified view

The past

- Central Product Security Team (CPST) **defines** the **Product Standard Security** serving as baseline
- Development Teams **Plan** their compliance to the **Product Standard Security** and store the compliance in **Product Standard Compliance** tool
- Development Teams **executes** on the plan
- Development Teams may have **implemented** or **deviated** from their agreed plan
- If CPST during security validation **finds no violation of agreed security level** then shipment is authorized

“Winter is coming!...” (John Snow - Games of Thrones)

- **SAP’s strategy embarked with speed into Mobile application development**
- **SAP acquired several mid-to-large size companies with divers software portfolio**
- **SAP’s strategy promoted SAP HANA to partners and strengthen partnership offerings**
- **SAP’s strategy embarked with speed into Cloud and recently into Internet of Things offering**

Wind of change…

- **#1: Ownership of the (security) risk moves with the Product Owners / Service Owners, i.e. CPST main objective is primarily “advising” rather than primarily “governing”**

Wind of change…

- **#1: Ownership of the (security) risk moves with the Product Owners / Service Owners, i.e. CPST main objective is primarily “advising” rather than primarily “governing”**
- **#2: Refine the way security risks are identified and managed**

Very important 3rd fact that we considered!

Developers are creators not builders!

Wind of change…

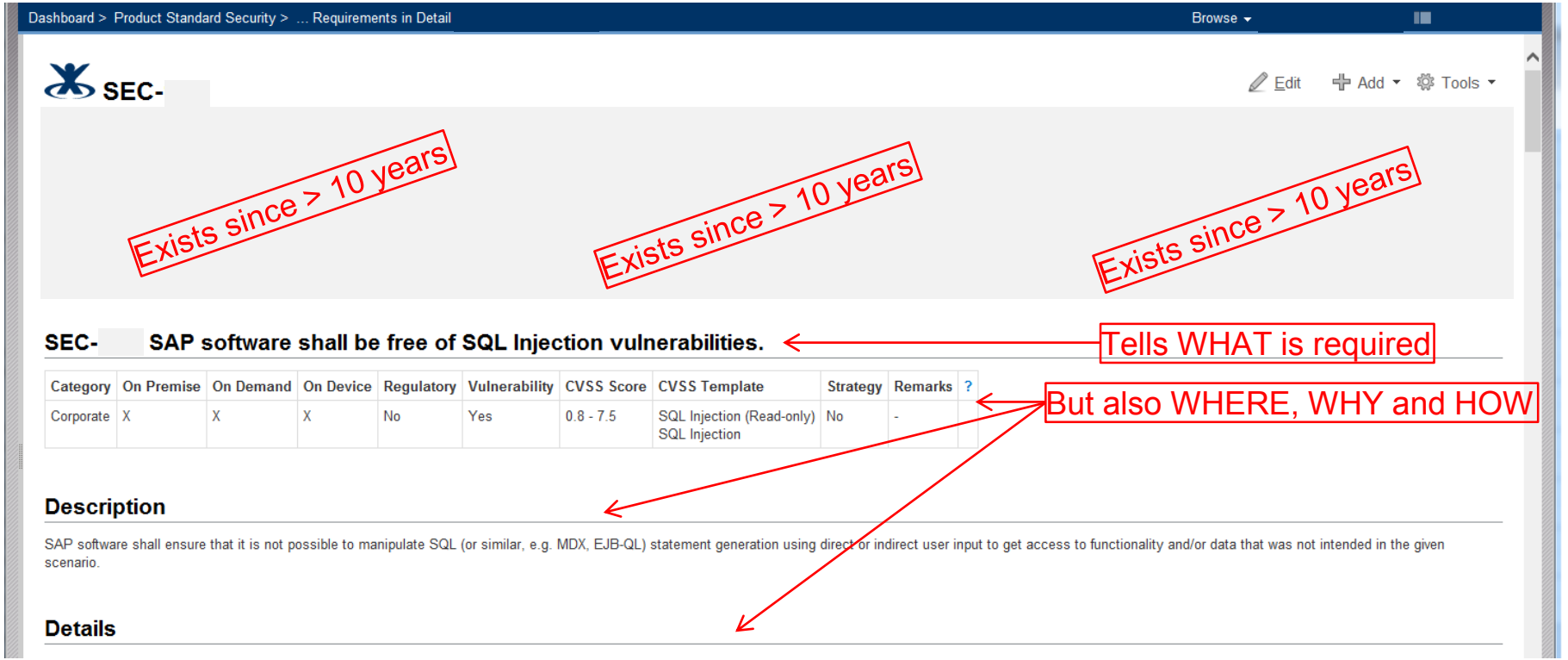
- #1: Ownership of the (security) risk moves with the Product Owners / Service Owners, i.e. CPST main objective is primarily “advising” rather than primarily “governing”
- #2: Refine the way security risks are identified and managed
- **#3: Invest in the people: Need to strengthen security experts, up skill and enable all the development teams**
  - Creating a collaboration environment and a network of security experts
  - Creating a reliable channel for disseminating security information
  - Allowing easier access to the huge security knowledge base
  - Identifying security risks, understanding the underlying impact and managing them appropriately
  - Teach methods for building misuse cases and thinking like hackers
  - Teach how to build security test plans
  - …

Very important 4th fact - external to SAP!

Declare compliance to ISO 27034

The common denominator: SAP Product Standard Security

A Requirement Example



SAP Product Standard Security

Could no longer serve as a standalone planning means

Feedback / Design thinking statements

From Developers, Architects and Security Experts

- Uncover the security threats and create transparency to decision makers
- Improve targeted security test cases / Improve true-positives in Code Scanning
- Up skill the development team and fits to our development

SAP Threat Modeling and Security Risk Identification & Management

SAP Threat Modeling

- is a systematic approach to uncover security threats at design time to reach a secure design
- outcome is targeted for architects, developers and security experts

Security Risk Identification & Management

- is a method based on SAP Threat Modeling
- outcome is targeted for decision makers, lead architects and security experts

Analyzing Risks: Security Risk Identification & Management + SAP Threat Modeling
Comparison

Security Risk Identification & Management

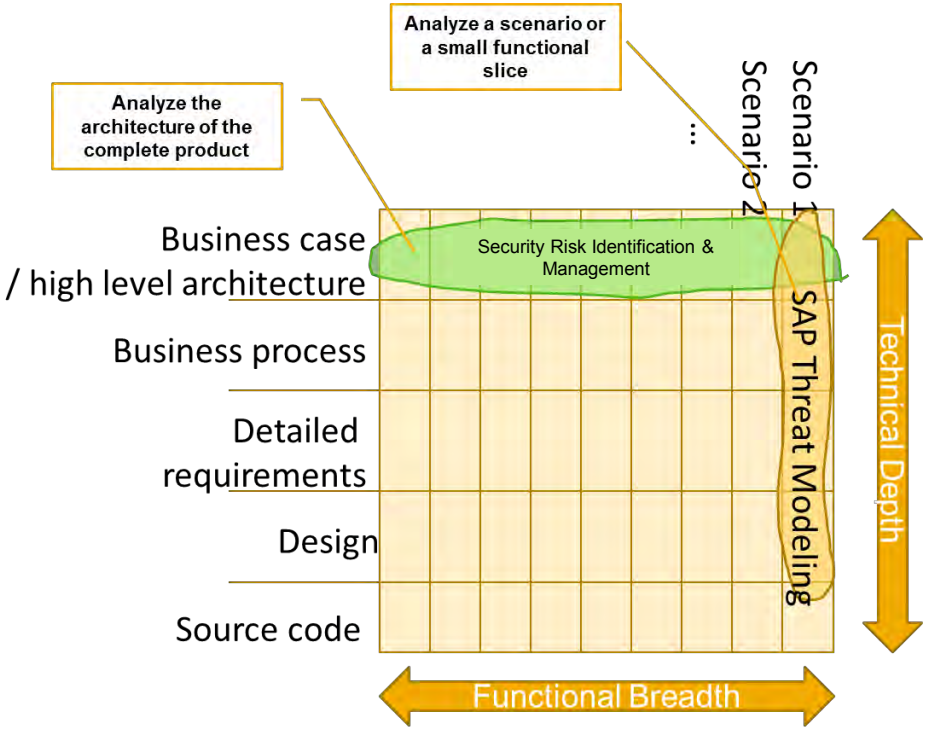
- Focus on complete product / service
- Analyze according to 10 security themes
- Document risk and risk response

=> High-level approach

SAP Threat Modeling

- Focus on critical scenarios
- Analyze these scenarios in detail
- Document threats, their risk, proposed mitigations and test cases

=> Very detailed, no coverage for huge applications

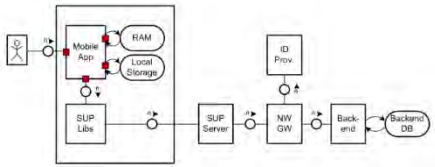


The common methodology

- Self-contained
- Timeboxed
- As simple as possible
- Clear workshop structure
- Clear outcome and documentation
- Decision and Follow-Up
- Mitigations by the Program

1

Understand the Architecture



Asset and Software centric view

2

Analyze potential Threats

Add an Attacker centric view

3

Put Threats on the Backlog & reprioritize

The seven steps of Security Risk Identification & Management

Risk Identifcation

1. Get common understanding about the architecture
2. Define the assets to be protected
3. Identify all risks in context of the product

Risk Analysis

4. Describe the risk incl.impact and mitigation alternatives
5. Rate the risks
6. Write documentation and present the risks to PO
7. Decide on the risks and document decisions

Workshop settings

Mandatory:
Program Lead Architect
Security Expert

Optional:
Lead Developer(s)
Product Owner (PO)

Standardizing the Methods Across SAP

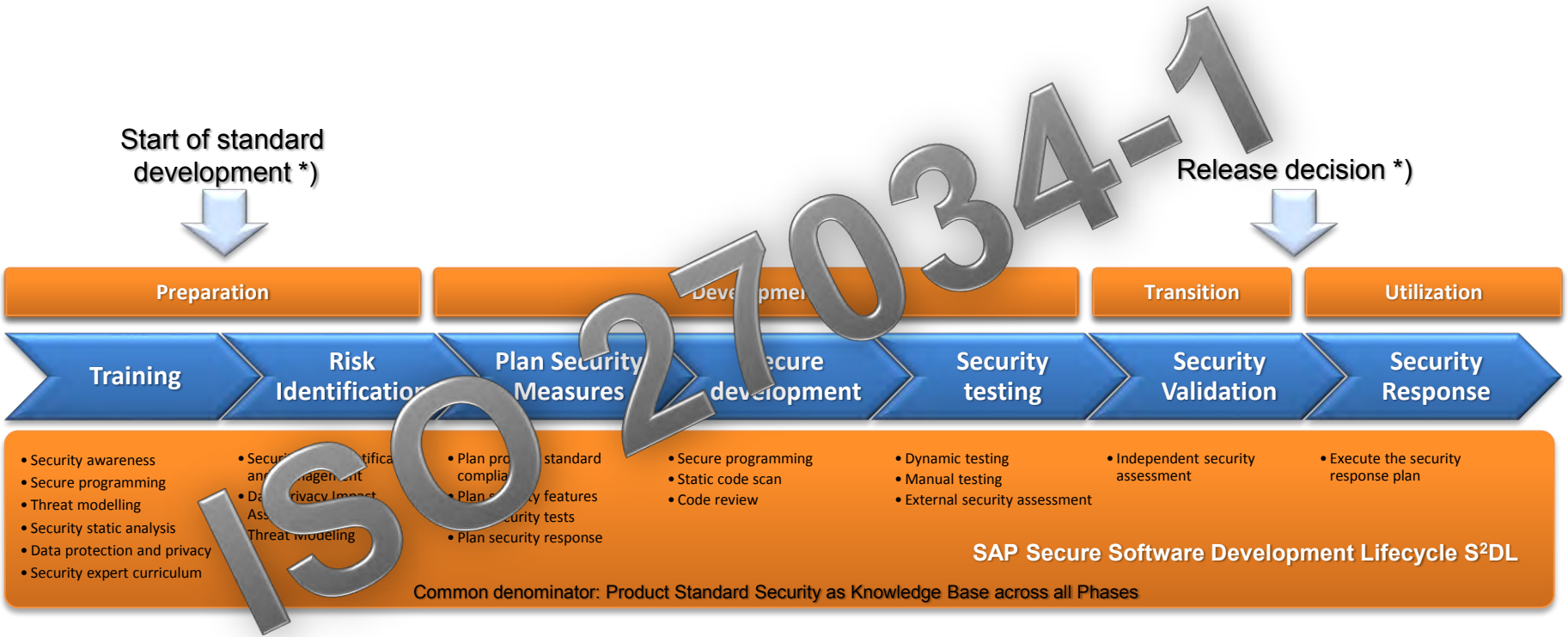
For SAP Threat Modeling

- 3 days class room training (200+ experts trained)
- Experts support projects across their development line
- Results and Decisions are reusable / understandable

For Security Risk Identification & Management

- Blended Learning with a prerequisite to have a certified Threat Modeling expert as a Security Risk Identification & Management lead

SAP Secure Software Development Lifecycle S²DL



*) In accordance to new I2M decision points

Open challenges

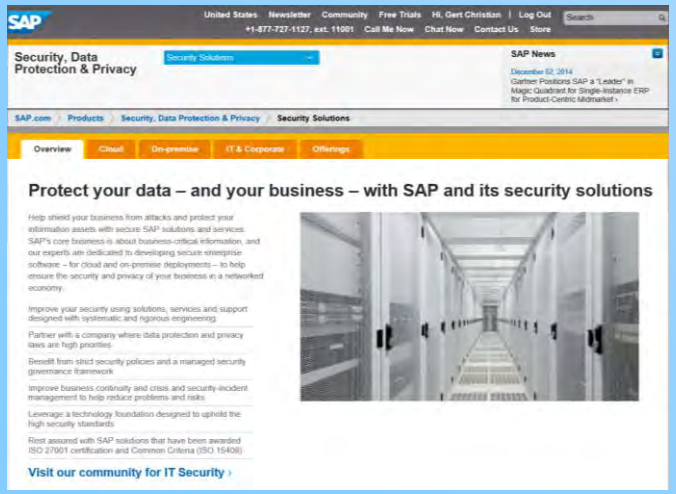
- ✓ Cloud Solutions
  - ✓ Development and the hosting of software are tightly integrated
  - ✓ Even shorter development and release time-frames
- ✓ Security Monitoring plan with SAP Enterprise Threat Detection
  - ✓ Creating “monitoring plan” from SAP Threat Modeling reports
- ✓ Internet of Things
  - ✓ Security Threats are standard, but the capabilities and solutions have a high dependency on devices and scenarios!

Summary

- ✓ The current SAP **S²DL** is a Risk-Based Security process
  - ✓ It helps SAP to scale with secure development to the various use cases
  - ✓ Reaching Risk-Based Security at SAP required a specific organizational infrastructure
- ✓ Security Risk Identification & Management and SAP Threat Modeling are the heart of Risk-Based Security process
  - ✓ Same methodology to identify security risks by different target user groups
  - ✓ Threat Modeling on the architecture for critical use cases
  - ✓ Security Risk Identification & Management for a complete product or solution
- ✓ Suitable risk description and rating focusing on affected assets and potential cost

Where to Find More Information

www.sap.com/security



- Cloud
- On premise
- IT & Corporate
- Offerings

www.sap.com/security

Thank you!

Dr. Paul El Khoury, CISSP
Co-Owner of SAP Product Standard Security, SAP SE

paul.el.khoury@sap.com

SASS15

Security issues in financial cloud environments

Volker Krummel, Wincor Nixdorf

Abstract:

On the first sight, Secure Financial Cloud seems to be a contradiction in itself. Concepts of open environments like cloud computing typically do not address challenges like thorough security concepts. Designing security architectures for arbitrary cloud environments seems to be a hard problem. In our research project “Securing the Financial Cloud (SFC)" we are researching approaches and solutions for a special cloud environment, the so called “financial cloud". In this talk I would like to present the actual status of our research and discuss interesting challenges.

Vita:

Volker Krummel is a Security Professional at Wincor Nixdorf since 2008. He received his PhD in the area of cryptography from the University of Paderborn in 2007. At Wincor Nixdorf he is responsible for the IT-Security Research. He is the project leader and specialist at several publicly funded cooperative research projects in the areas of secure cloud computing, IT Forensics and Risk Management. His research interests cover Cryptography, Computer Algebra and Information Theory, IT-Security Analysis and IT-Forensics.

Fraunhofer FOKUS

Security Issues in Financial Cloud Environments
where no bank has gone before …

Dr. Volker Krummel
CTO-Office – Research & Innovation

Wincor Nixdorf International

SASSI Workshop 2015, Berlin

WINCOR NIXDORF
EXPERIENCE MEETS VISION.

WINCOR NIXDORF

Target Incident(s) 2013 & 2014

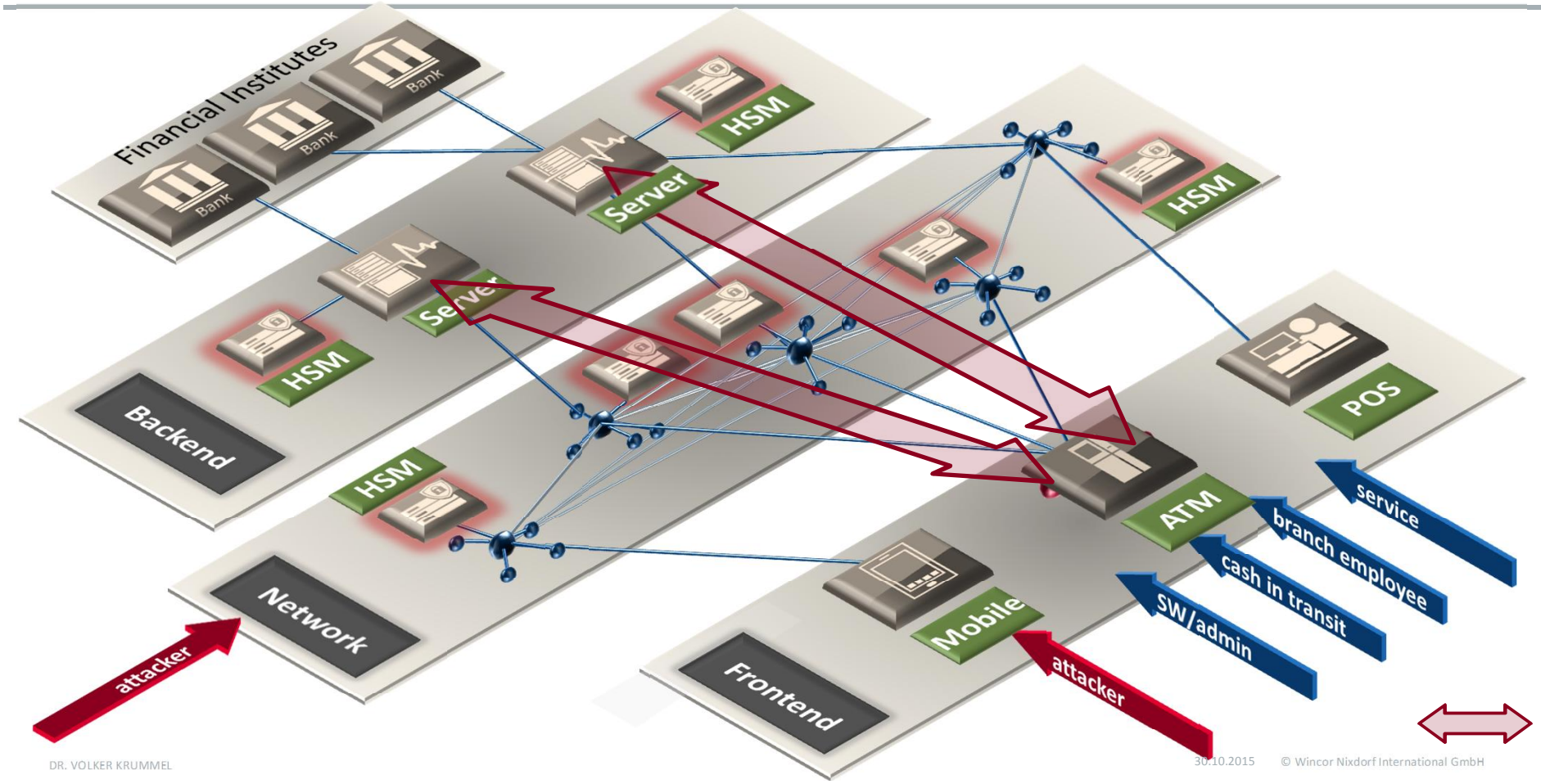
- Malware on POS Terminals
- ca. 3 weeks
- data of ca. 40 mio credit cards were stolen
- Business: ca. 18-35 Dollar per data set
- personal data of ca. 70 mio customers stolen
- direct impact on business

J.P. Morgan Chase & Co Incident 2014

- Malware in IT System
- ca. 2 months
- Prey: ca. 76 mio private credit card data and 7 mio business customers
- until now no criminal usage of data

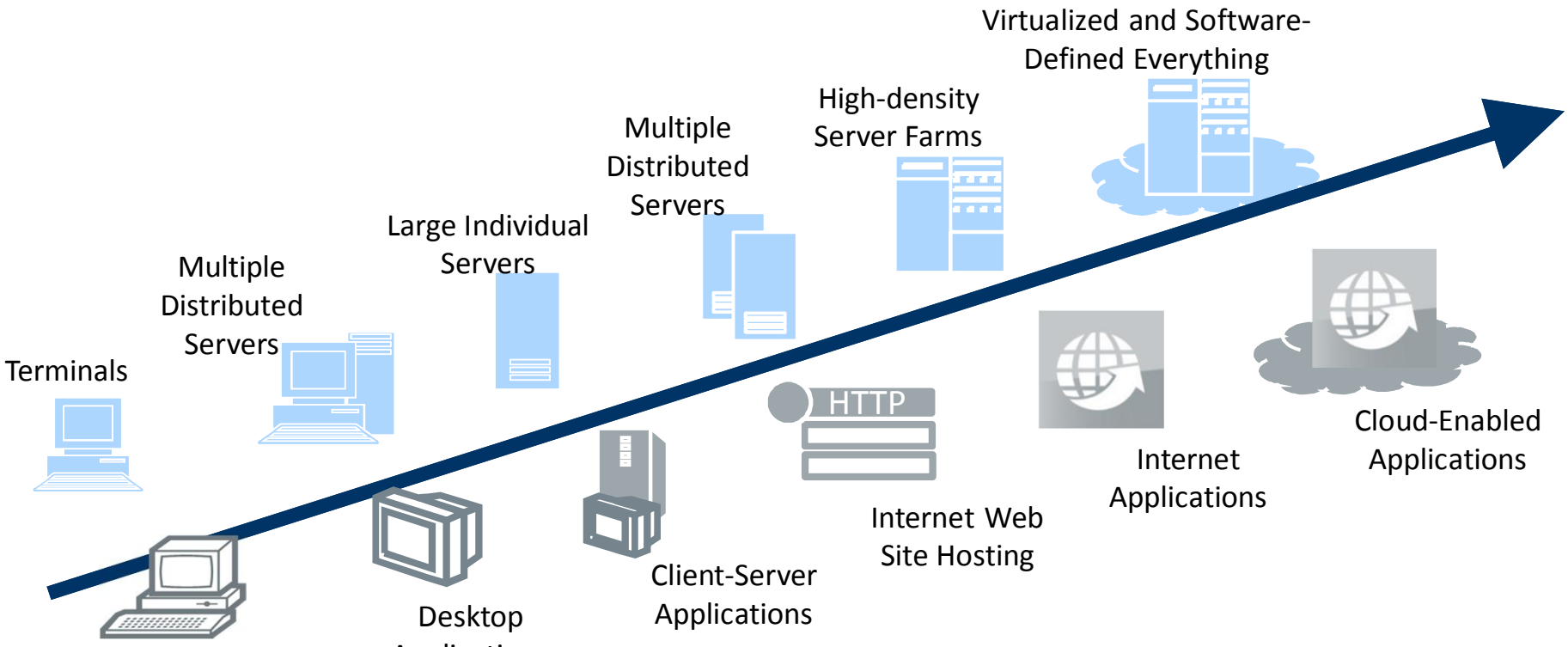
The Classical Financial Infrastructure

WINCOR NIXDORF



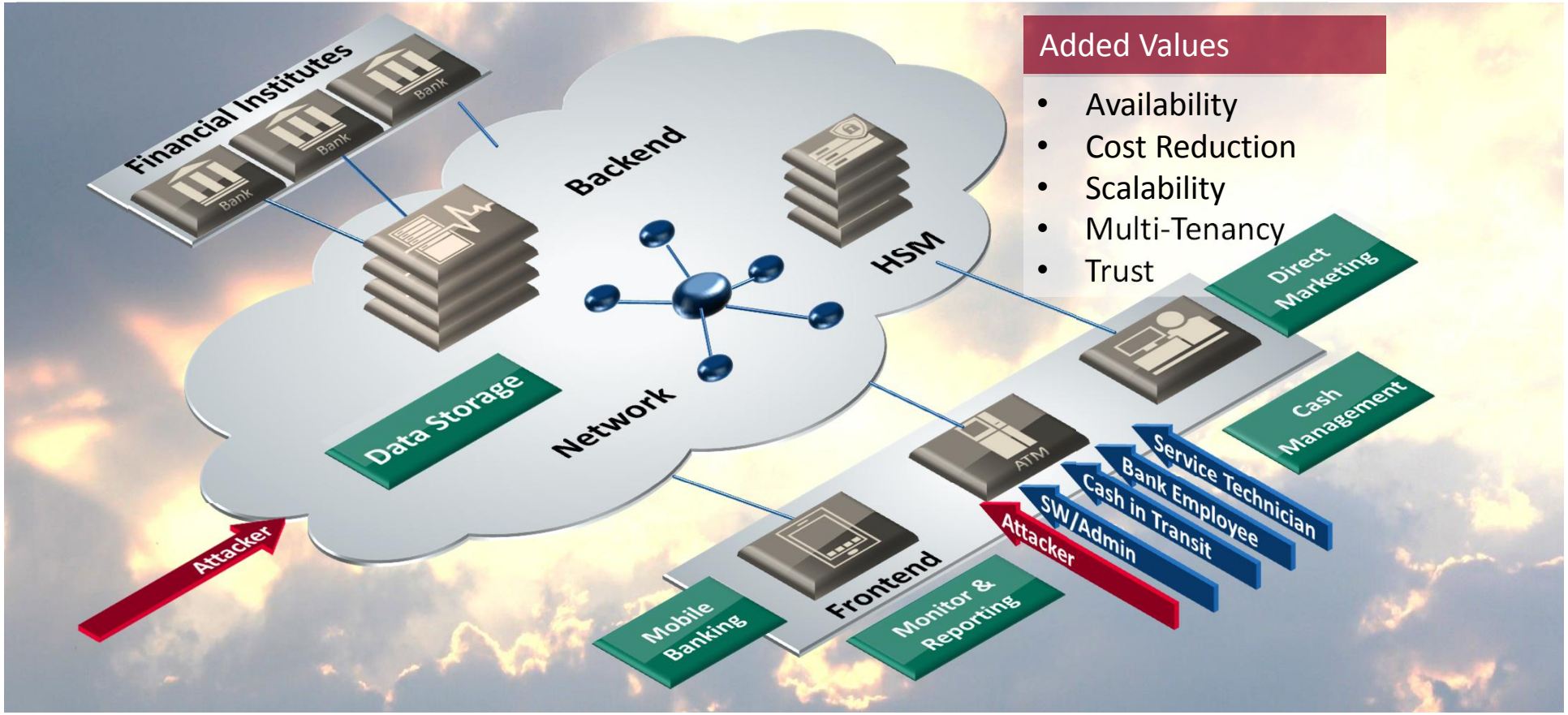
Evolution of the IT-Infrastruktur

WINCOR NIXDORF



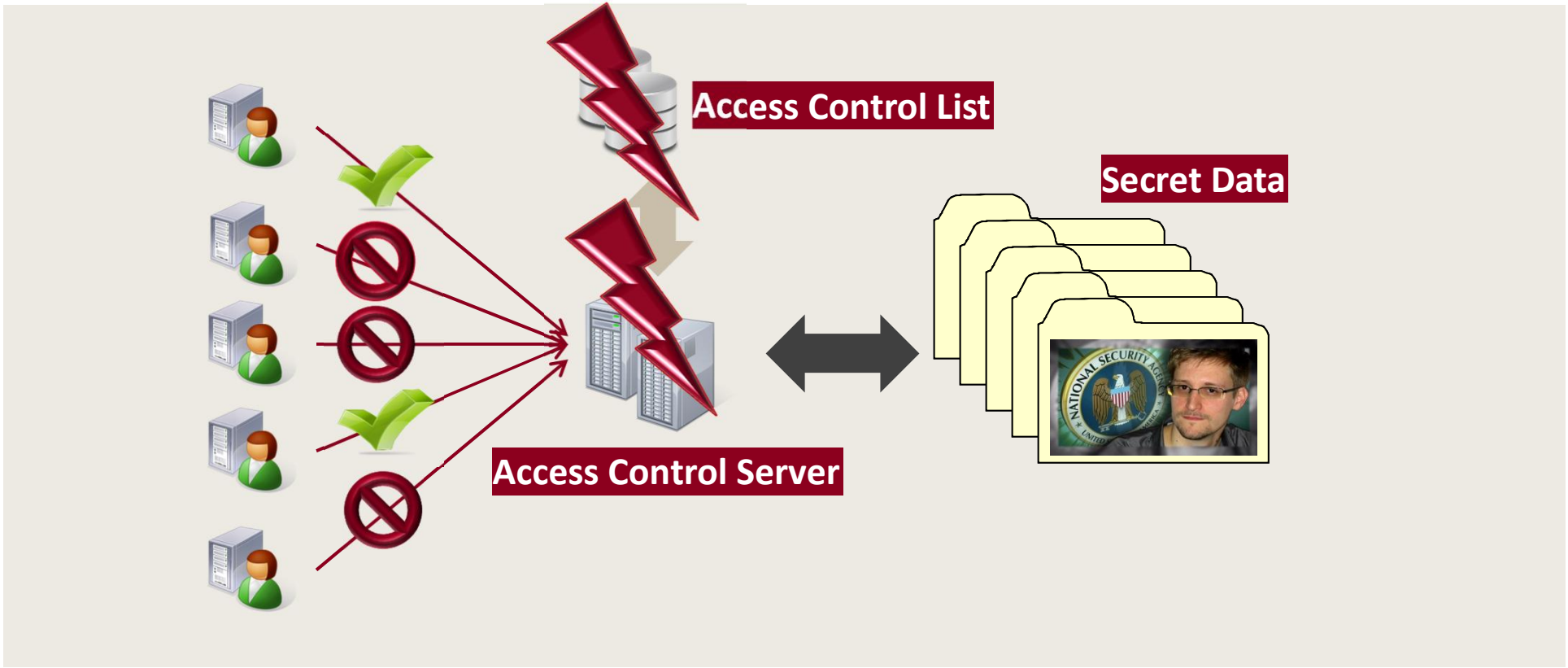
Financial Infrastructure of the Future

WINCOR NIXDORF



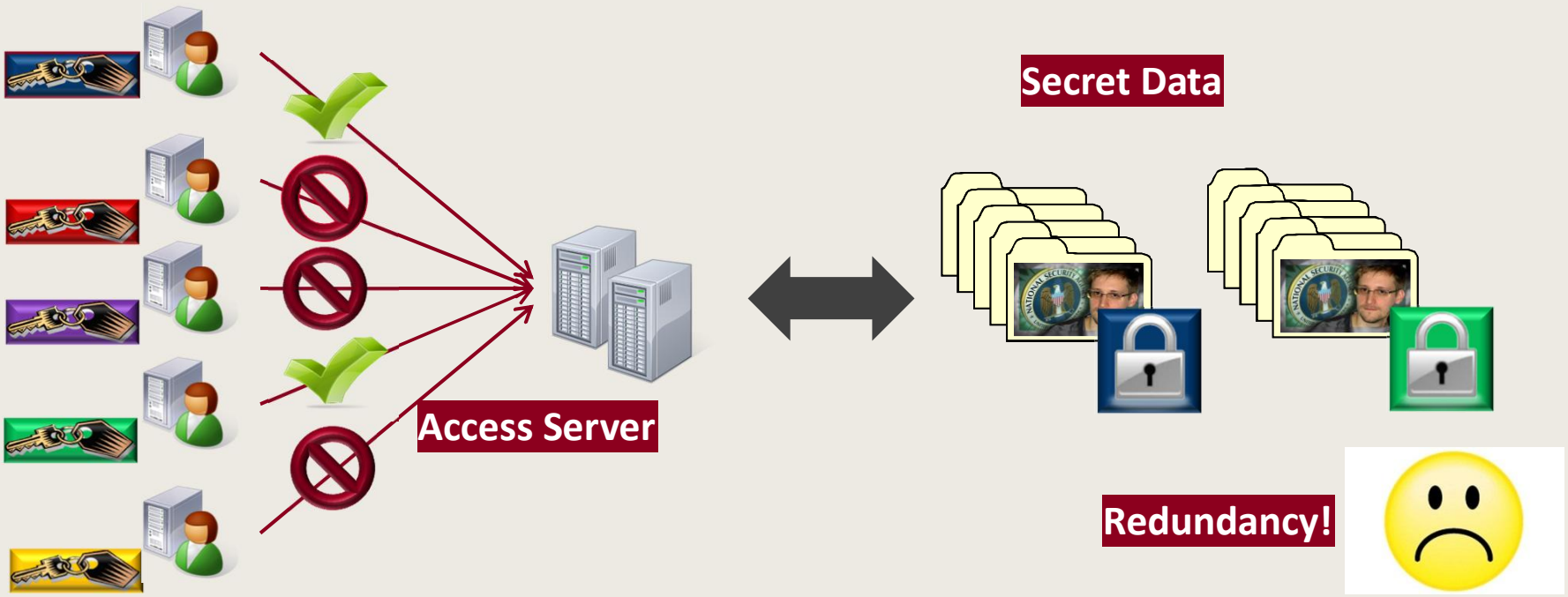
Classical Access Control Server is not appropriate

WINCOR NIXDORF



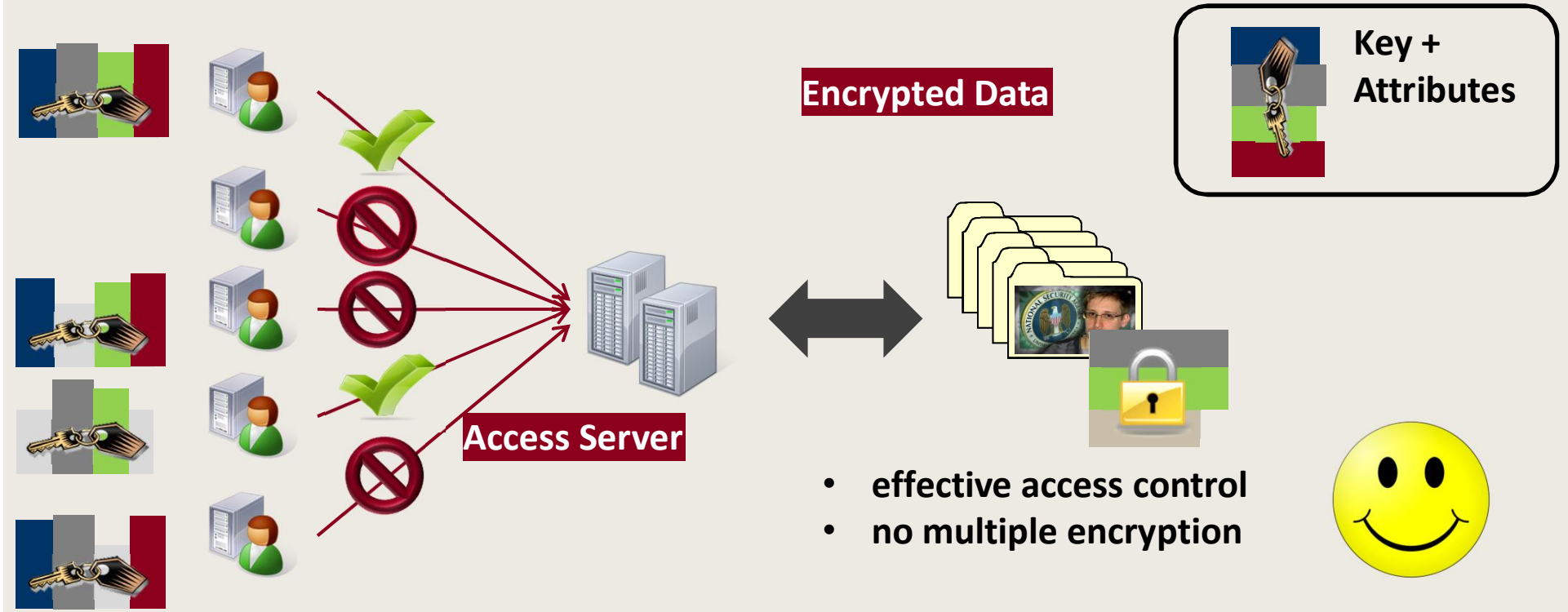
Access Control based on classical Encryption

WINCOR NIXDORF



Access Control based on Attribute Based Encryption

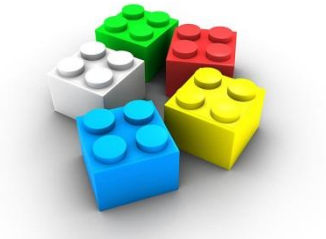
WINCOR NIXDORF



Building Blocks

WINCOR NIXDORF

1	Crypto	Security Proofs	Efficient Implementation
2	Implementation	Verification	
3	SW & HW Setup	Side Channel Analysis & Invasive Attacks	
4	System	Security Analysis (Stride)	Cloud Architecture
5	Certification	Understanding Formal Process	Understanding Practical Process	Prepare Certification
6	Operation	Monitoring, Security Processes	



WINCOR NIXDORF

Algorithms for Attribute Based Encryption (ABE) are very complex

- large number of parameters with dependencies
- large variety of algorithms and building blocks
- bilinear pairings on elliptic curves defined over finite extension fields
- Security Proof

Optimization for Speed

- adapt to different plattforms like embedded hardware, smartcards, HSM
- currently no support by Crypto Coprocessors

Optimization for Security

- balancing key length
- Implementation secure against side channel attacks

Extensions

- Searchable Encryption
- …

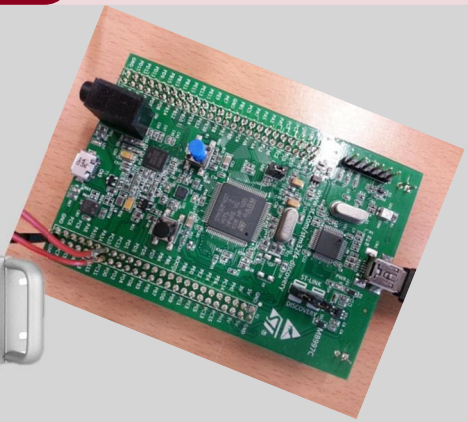
Parameter

Security Level: 128 bit (80 bit)

EC Group size: 256 bit (160 bit)

finite field size: 3248 bit (1248 bit)

embedding degree: 12



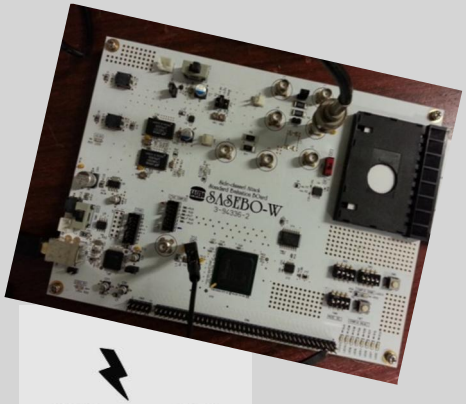
Implementation and Hardware & Software Setup

WINCOR NIXDORF

Optimization
- Speed & Code Size
- Copocessor Design

Verification of Correctness
- Source Code Review
- Test vectors (reference implementation)

Side Channel Analysis
- Power Analysis (SPA & DPA)
- Reference Setup (Sasebo / Sakura boards)



„magic“ Analysis

KDV1

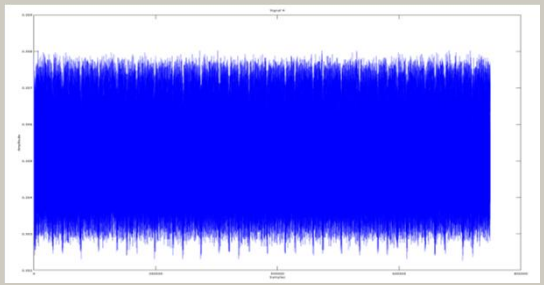
Target Platform:

ARM Cortex M4 @ 168 MHz, 1MB flash, 192 KB RAM,

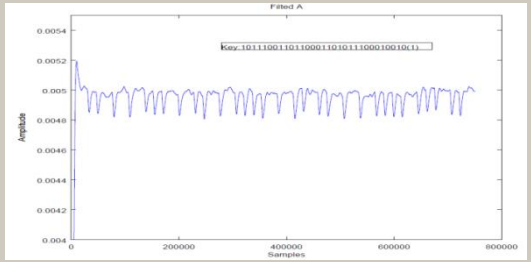
Code Size: ca. 180KB

Performance: 1.5 sec / pairing

Power Measurement



Key Exposure



KDV1 Update + Oszilloskop Kurven

Krummel, Dr., Volker; 09.09.2015

System & Certification

WINCOR NIXDORF

Architecture (approx. 50 req.)

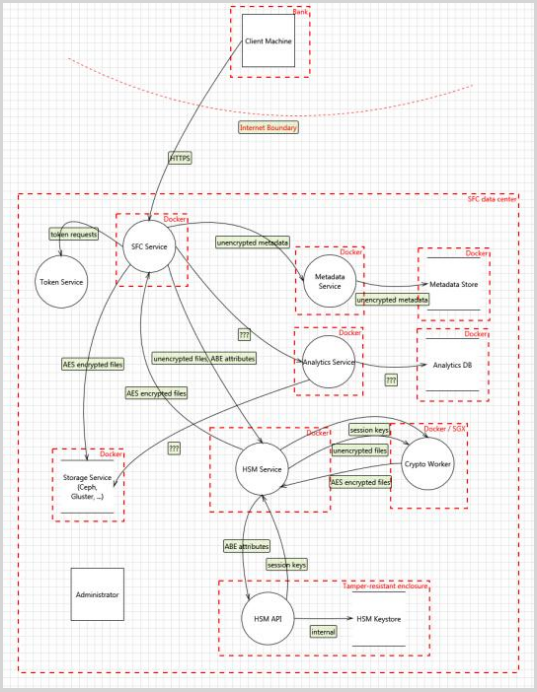
- elastic resources
- distributed storage

Security Analysis (approx. 120 req.)

- API Attacks -> HSM
- Threat Model (STRIDE, Attack Trees)

Certification

- relevant standards (CC, PCI-DSS, MaRisk (BaFin))
- Customer Interviews & Report
- CC Security Target (Redefinition of TOEs)
- Knowledge about practical aspects



Detail of the Threat Model (Draft)

Securing the Financial Cloud (SFC)

WINCOR NIXDORF

How can the Financial Infrastructure look like in the future?

Aim of the project

- novel cloud-based approaches for financial transactions
- security as the most important property
- novel cryptographic techniques
- added values: availability, cost reduction, scalability, multicliant architecture and trust
- trust as the key factor for succesfull business

Partner

WINCOR NIXDORF · utimaco® a member of the Sophos Group · escrypt Embedded Security · janz IT Consulting · Solutions · Services · UNIVERSITÄT PADERBORN Die Universität der Informationsgesellschaft · achelos the experts in eID

Government-funded by

Bundesministerium für Bildung und Forschung

FKZ: 16KIS0058K

Guide Banking Data Centers Into a Secured Future
Preventive Crisis and Risk Management for Data Centers

WINCOR NIXDORF

Conceptual and technical development of an integrated framework to preventively manage risks and crises for data centers of system relevant banks

Aim of the project

- Risk analysis, risk reduction
- Check of compliance with norms and guidelines
- Detection of threats in real-time, semi automatic crisis intervention
- Risk controlled security tests and measurements
- Simulation of thread scenarios and crises situations

Partner

WINCOR NIXDORF · HypoVereinsbank UniCredit Group · Fraunhofer FOKUS · XiV-CONSULT

Supervised by

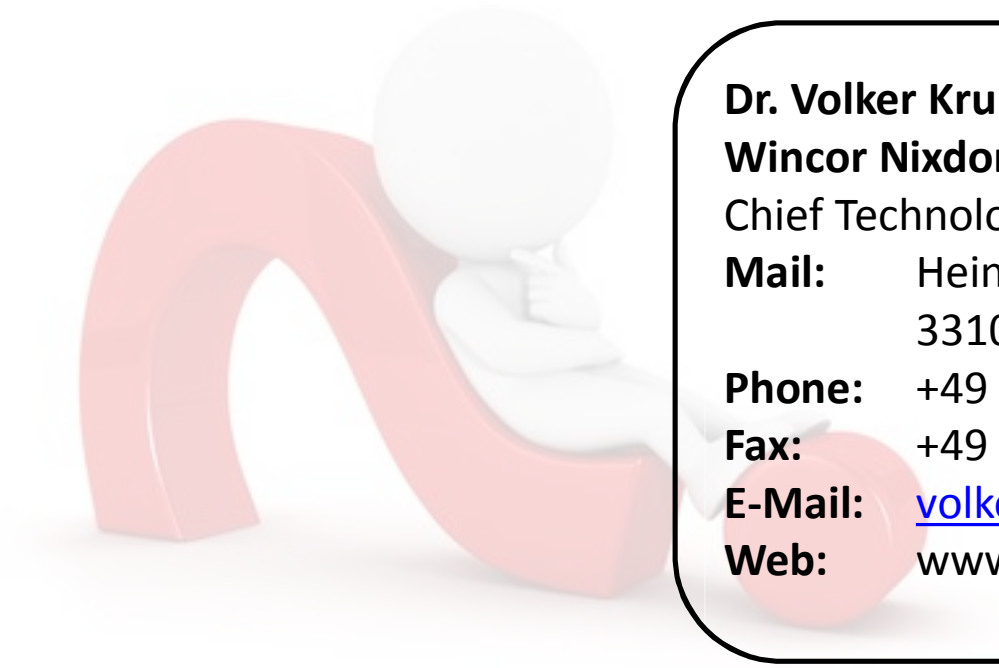
VDI | VDE | IT

Government-funded by

Bundesministerium für Bildung und Forschung

WINCOR NIXDORF

Thank You for Your Attention!



Dr. Volker Krummel
Wincor Nixdorf International GmbH
Chief Technology Office | Corporate Research Security
Mail: Heinz-Nixdorf-Ring 1,
33106 Paderborn, Germany
Phone: +49 (5251) 693 - 6216
Fax: +49 (5251) 693 - 6309
E-Mail: volker.krummel@wincor-nixdorf.com
Web: www.wincor-nixdorf.com

SASS15

Risk monitoring of an pseudonymisation service based on TRICK Service

Ben Fetler, itrust consulting

Abstract:

TRICK Service (Tool for Risk management of an ISMS based on a Central Knowledge base) is a risk assessment & management web application for identification, analysis and estimation of assets, threats, vulnerabilities, risk scenarios and security measures. TRICK Service enables to determine a list of security measures to implement in order to reduce the impact or the occurrence likelihood of possible risk scenarios.The presentation illustrates how risk parameter like security implementation rates, threats likelihood, and impact values are calculated in real time with inputs from security monitoring tools, so that the current risk situation is reflected. Lessons learned from applied risk monitoring on an itrust consulting service providing pseudonymisation for student evaluation tests are discussed.

Vita:

Ben Fetler, Owner of a Master's degree (Reutlingen University) in Business Information Systems, is a part of itrust since 2012. During 2 internships at itrust consulting, he developed beneath others models to measure the uncertainty of risk estimations and the maturity of security measures coming from ISO/IEC 27001. Today he mainly assists service providers to get ISO/IEC 27001 certified and conducting risk analyses. Additionally he is member of the technical committee ISO/TC 262 – Risk Management and product owner of the risk analysis tool TRICK Service. Currently he is involved in a national research project to develop a real time risk monitoring system.

Fraunhofer FOKUS

SASS '15

Risk monitoring of a pseudonymisation service based on TRICK Service

Speaker: Ben Fetler

Authors: Ben Fetler, Steve Muller

Agenda

Introduction to TRICK Service & ÉpStan project

Real-time risk assessment

Conclusion and outlook

Introduction
TRICK Service

itrust consulting

TRICK Service

Tool for **R**isk management of an **I**SMS based on a **C**entral **K**nowledge base

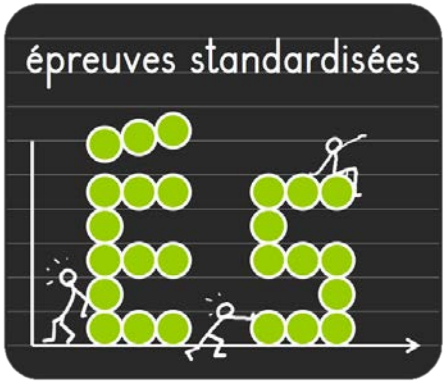
itrust consulting

Core principles

- Risk management following ISO/IEC 27005;
- Quantitative assessment of likelihood and impact of different risk scenarios;
- Use of a “Risk Reduction Factor” (RRF) which enables to quantify the influence of security measures on the losses caused by threats to assets;
- Cost-effectiveness of security controls; TRICK Service considers the Return On Security Investment (ROSI) and derives a prioritised action plan.

Introduction
ÉpStan

itrust consulting



UNIVERSITÉ DU LUXEMBOURG

Luxembourg's national school monitoring programme

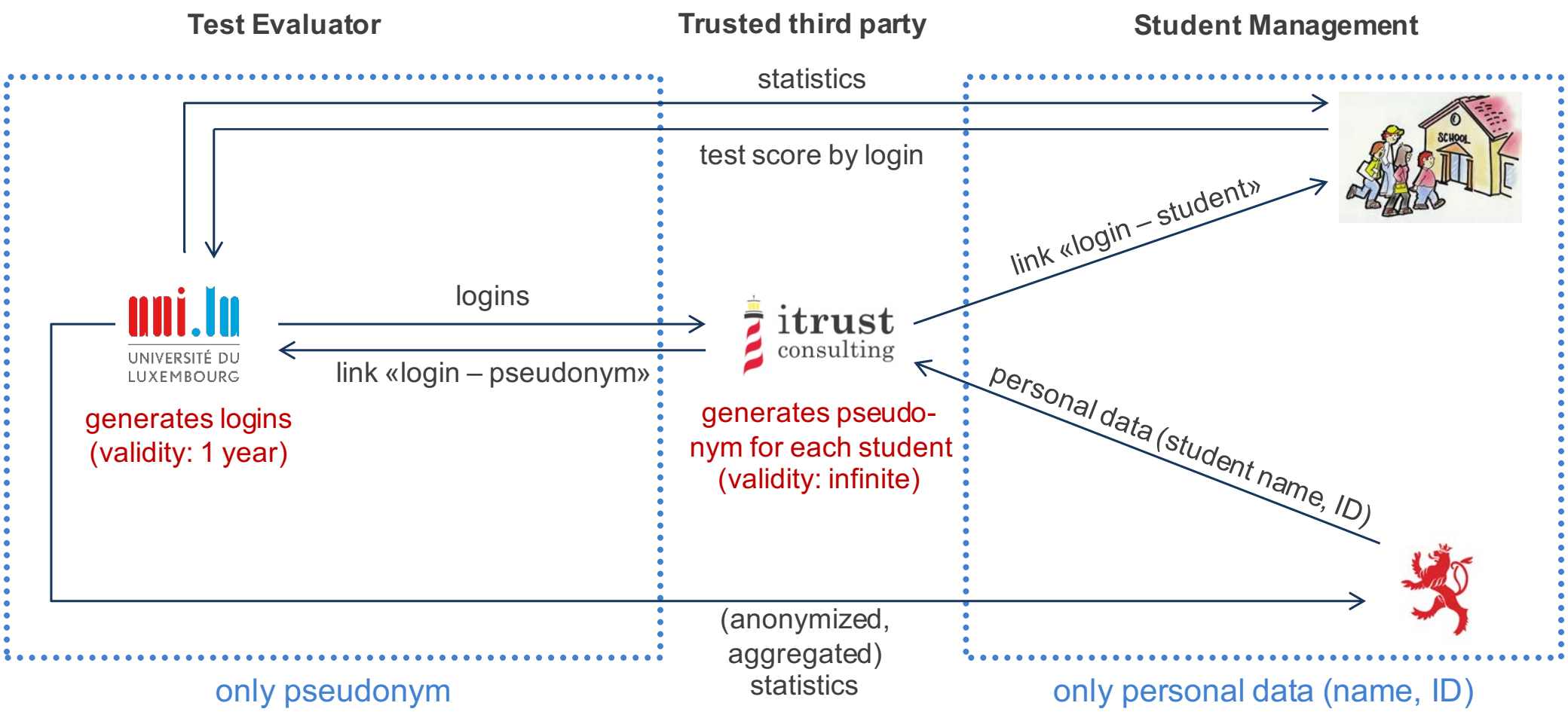
Introduction
ÉpStan

itrust consulting

Requirement:
University and Ministry shall not make link between results and student.

Solution:
Involve a trusted third party (TTP) offering a pseudonymisation service.

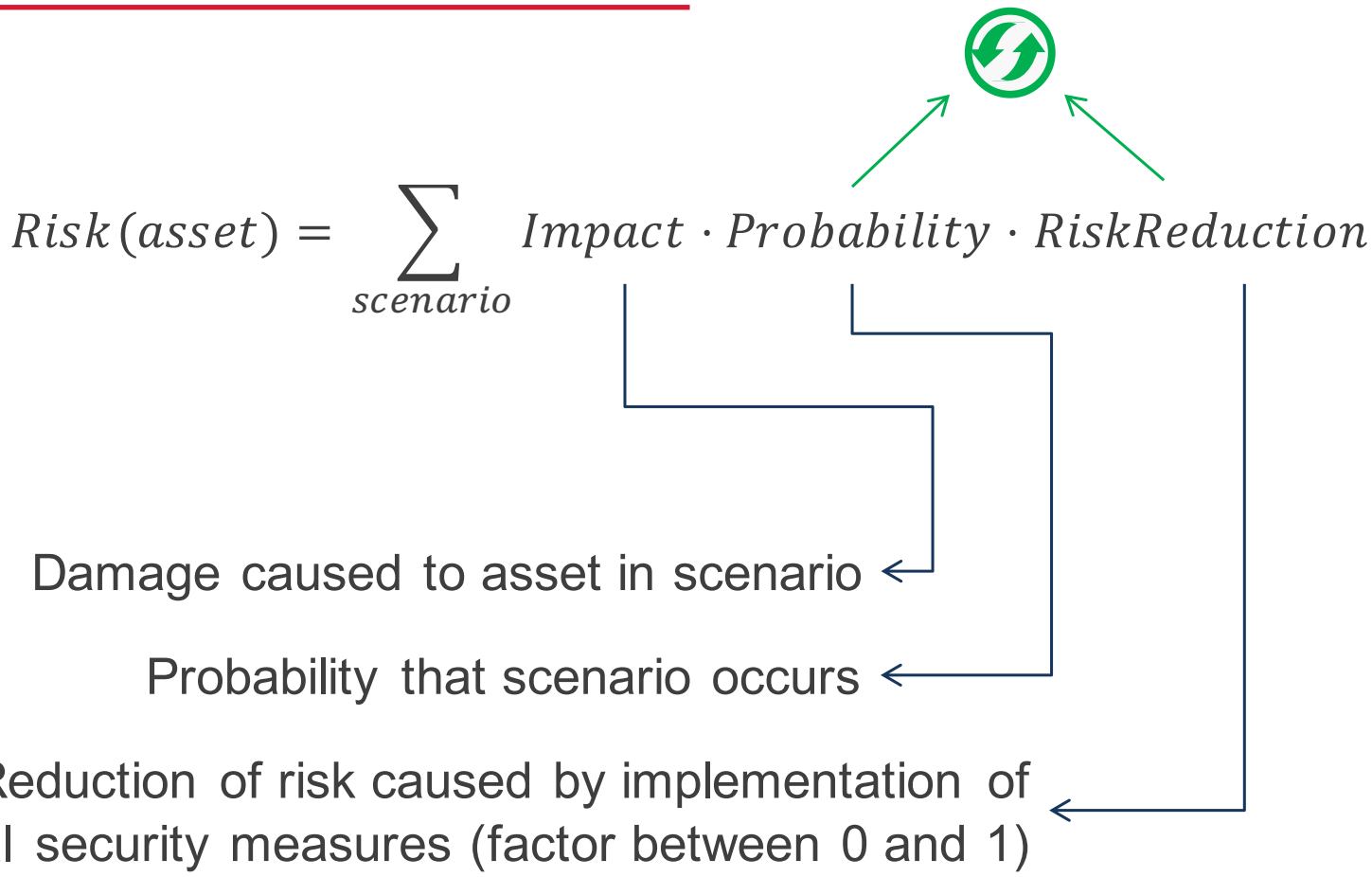
Introduction
ÉpStan



Real-time risk assessment

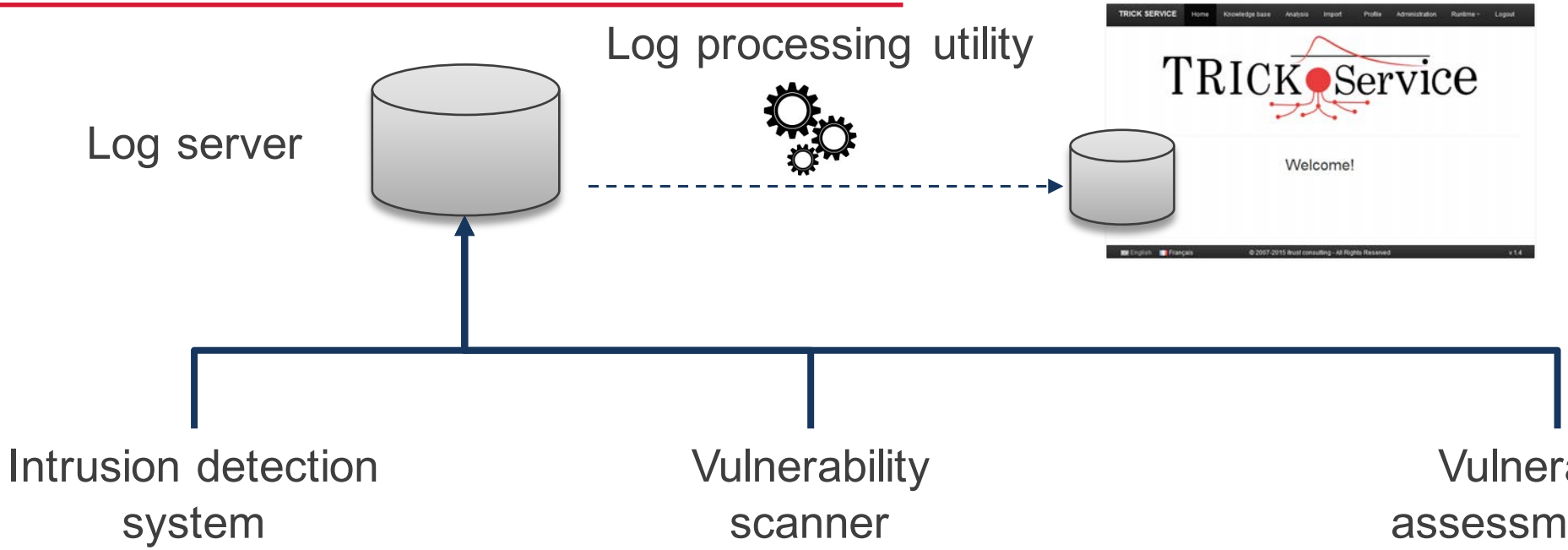
Risk computation

Dynamically



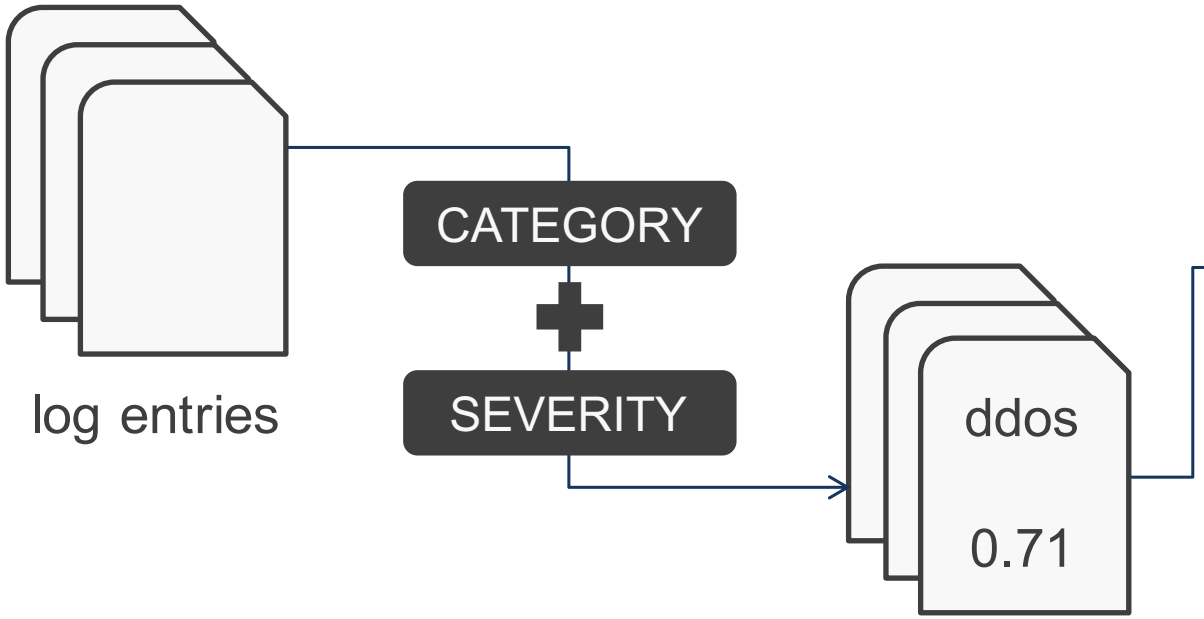
Real-time risk assessment

Strategy



Real-time risk assessment

Log processing utility

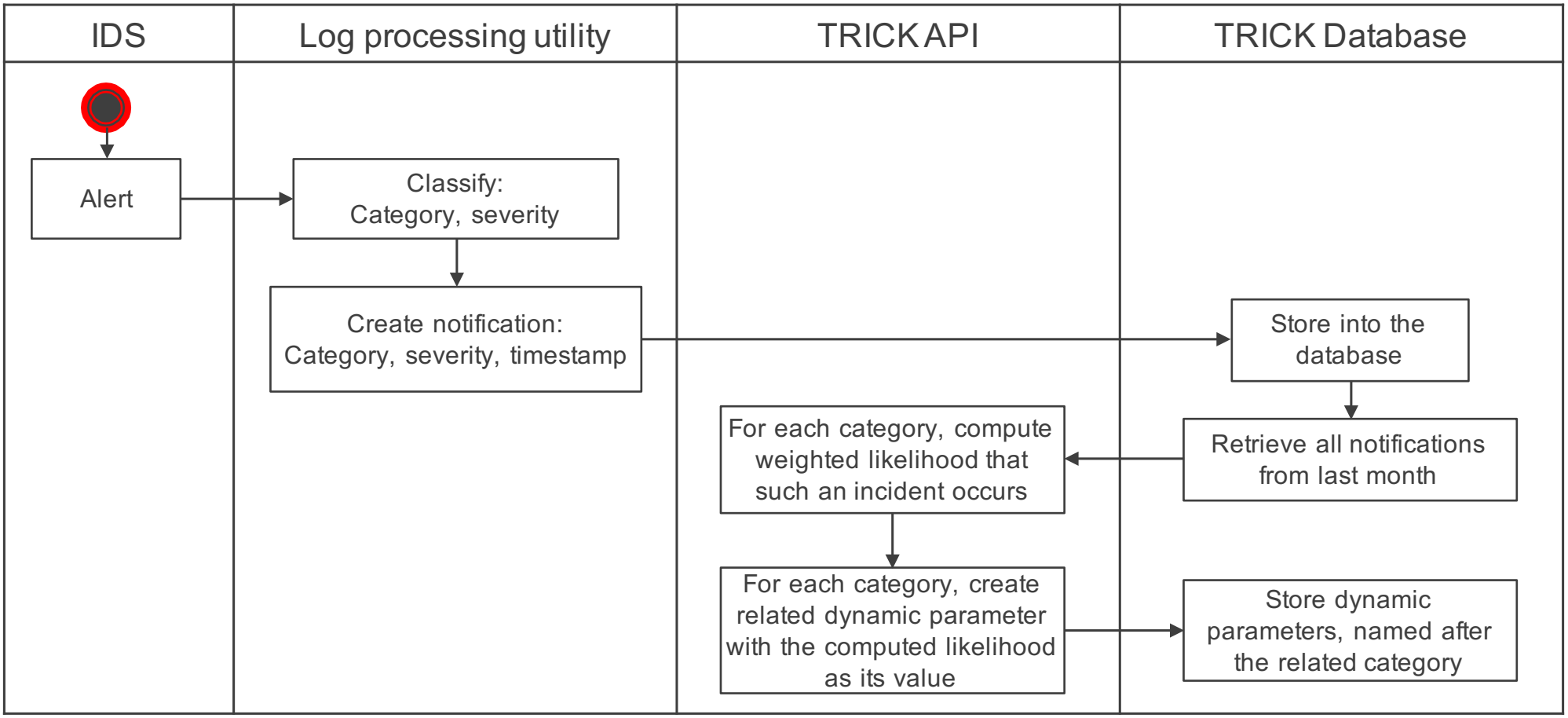


Category	Probability
ddos	$\Pr[ddos]$
malware	$\Pr[malware]$
diskfailure	$\Pr[diskfailure]$
...	...

- $\Pr[category]$ increases with each log entry
  (the higher the severity, the higher the increase)
- $\Pr[category]$ decreases with time

Real-time risk assessment

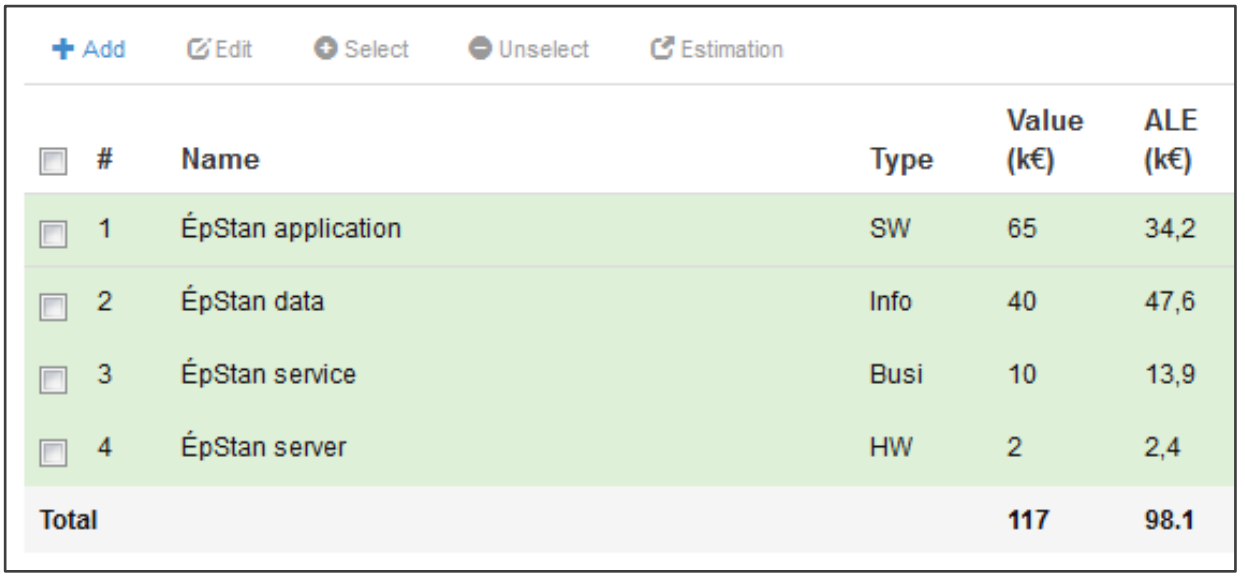
PoC - Intrusion detection system



Real-time risk assessment

TRICK Service: dynamic likelihood

itrust consulting



+ Add Edit Select Unselect Estimation

#	Name	Type	Value (k€)	ALE (k€)
1	ÉpStan application	SW	65	34,2
2	ÉpStan data	Info	40	47,6
3	ÉpStan service	Busi	10	13,9
4	ÉpStan server	HW	2	2,4
Total			**117**	**98.1**

- Definition of all ÉpStan-related assets
- Automatic real-time estimation of Annual Loss Expectancy (ALE)
  ALE = impact · likelihood

Real-time risk assessment

TRICK Service: dynamic likelihood

Scenario	Imp. (k€)	Pro. (/y)	ALE (k€)
A_all - Complete loss, including backup	i6	ids_malware*0.05+ ids_disk_failure_db	15,2
C3 - Accidental disclosure	i7	p3	11,5
A_1 - Partial loss or temporary	i4	ids_ddos*0.1	5,1
I3 - Accidental manipulation	i5	p4	5
C1 - Partial theft coming from external	i6	ids_login_bruteforce_db*0.1	4,4

Impact		Probability	
i0	2 k€	p0	1/100y
i1	4 k€	p1	1/50y
i2	10 k€	p2	1/30y
i3	16 k€	p3	1/16y
i4	25 k€	p4	1/10y
i5	50 k€	p5	1/5y
i6	100 k€	p6	1/3y
i7	200 k€	p7	1/2y
i8	400 k€	p8	1/y
i9	800 k€	p9	2/y
i10	1 600 k€	p10	3/y

- Support for expressions in ‘likelihood’ field involving variables resulting from log processing utility
- ALE is updated in real-time

Real-time risk assessment

TRICK Service: dynamic risk reduction

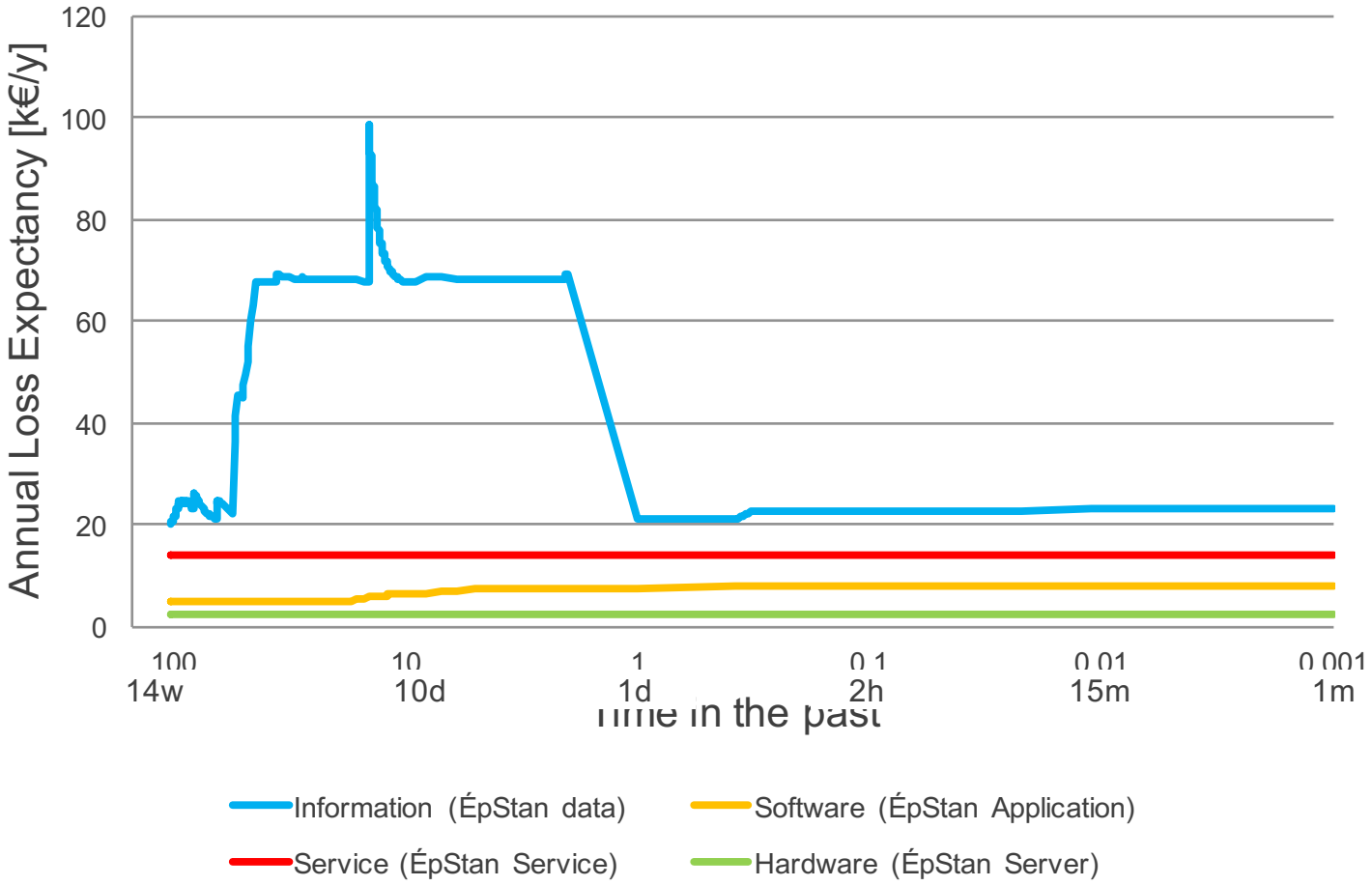
IR = Implementation Rate

Ref	Domain	Status	IR (%)	IW (md)	EW (md)	INV (k€)	LT (y)	IM (md)	EM (md)	RM (k€)
12.5	Control of operational software									
12.5.1	Installation of software on operational systems	AP	ids_patch_mgmt	2	0	0	5	0,2	0	0
12.6	Technical vulnerability management									
12.6.1	Management of technical vulnerabilities	AP	50	1	0	0	1	0,1	0	0
12.6.2	Restrictions on software installation	AP	50	0	0	0	5	1	0	0
12.7	Information systems audit considerations									
12.7.1	Information systems audit	AP	70	0	0	0	5	1	0	0

- Implementation rate with support for expressions
- Real-time update of implementation rate

Real-time risk assessment

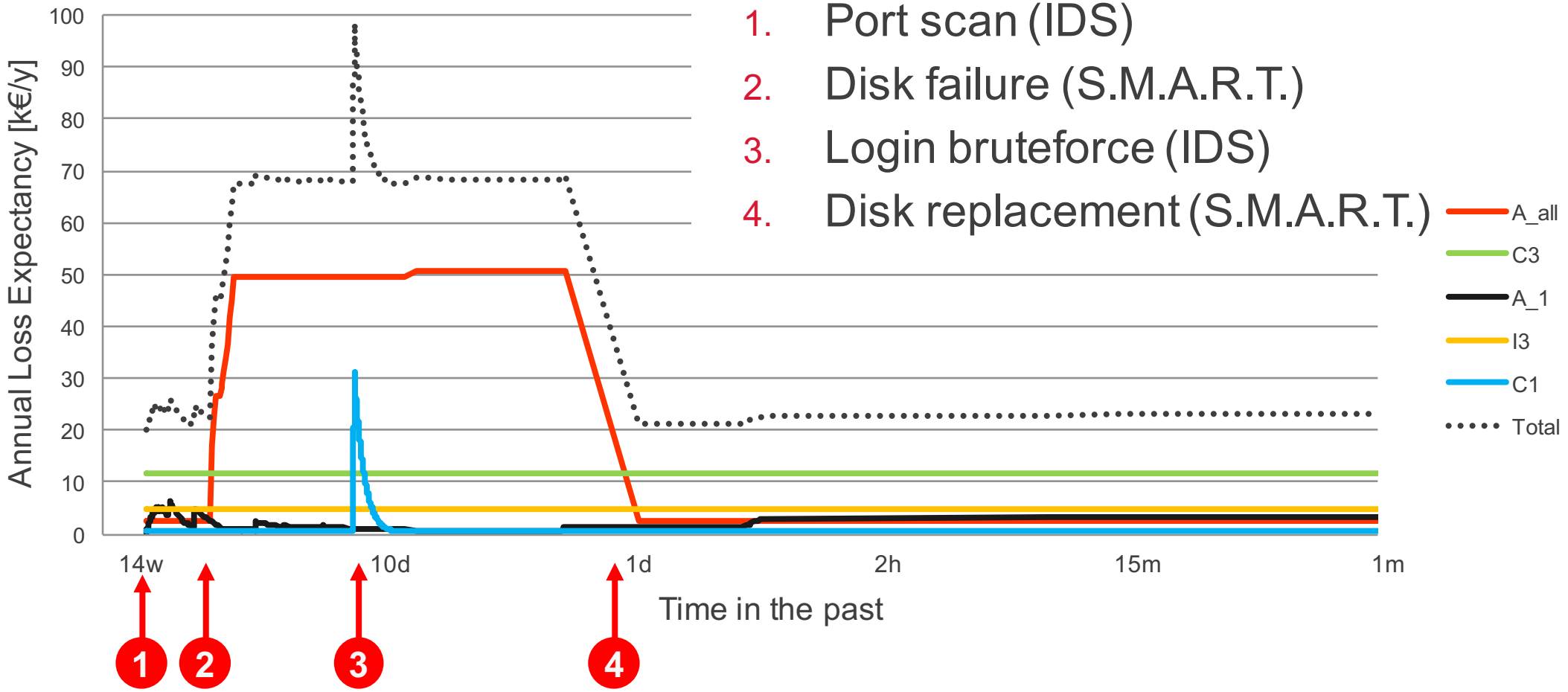
TRICK Service: Cockpit



- Real-time graph displaying ALE per asset type
- Logarithmic time scale to put focus on recent past
- Click on asset type opens up detailed view (see next slide)

Real-time risk assessment

TRICK Service: ALE evolution of «Information» assets



1. Port scan (IDS)
2. Disk failure (S.M.A.R.T.)
3. Login bruteforce (IDS)
4. Disk replacement (S.M.A.R.T.)

Conclusion and outlook

- Real added value: Having view on current risk situation & its impacts;
- Use logs of several information security tools;
- Apply real-time risk assessment to Industrial Control System environment;
- Define generic expressions for dynamic likelihood and risk reduction computation;
- Add asset dependency functionality.

SASS15

The attack navigator – Finding and defending against socio-technical attacks

Christian W. Probst, Tresspass

Abstract:

Industry must react to both existing and unknown attacks on software and intelectual property. These attacks involve physical, virtual, and socio-technical components. Risk assessment is used to prioritize the use of defense resources. The TREsPASS project has developed the concept of an attack navigator that uses system maps and attacker profiles to identify attacks. The attack navigation on system maps is based on invalidation of organisational policies, resulting in weighted attack trees to guide risk assessment and governance using typical attacker profiles.

Vita:

Christian W. Probst is an Associate Professor in the Department of Applied Mathematics and Computer Science at the Technical University of Denmark, where he works in the section for Language-Based Technologies. Christian is technical co-lead of the TREsPASS project. In his work he addresses safety and security properties of systems and organisations, most notably insider threats. He is the creator of ExASyM, the extendable, analysable system model, which supports the identification of insider threats in organisations.

Fraunhofer FOKUS

The Attack Navigator

predict
prioritise
prevent
TREsPASS

Finding socio-technical attacks and defending against them

Christian W Probst

SEVENTH FRAMEWORK PROGRAMME

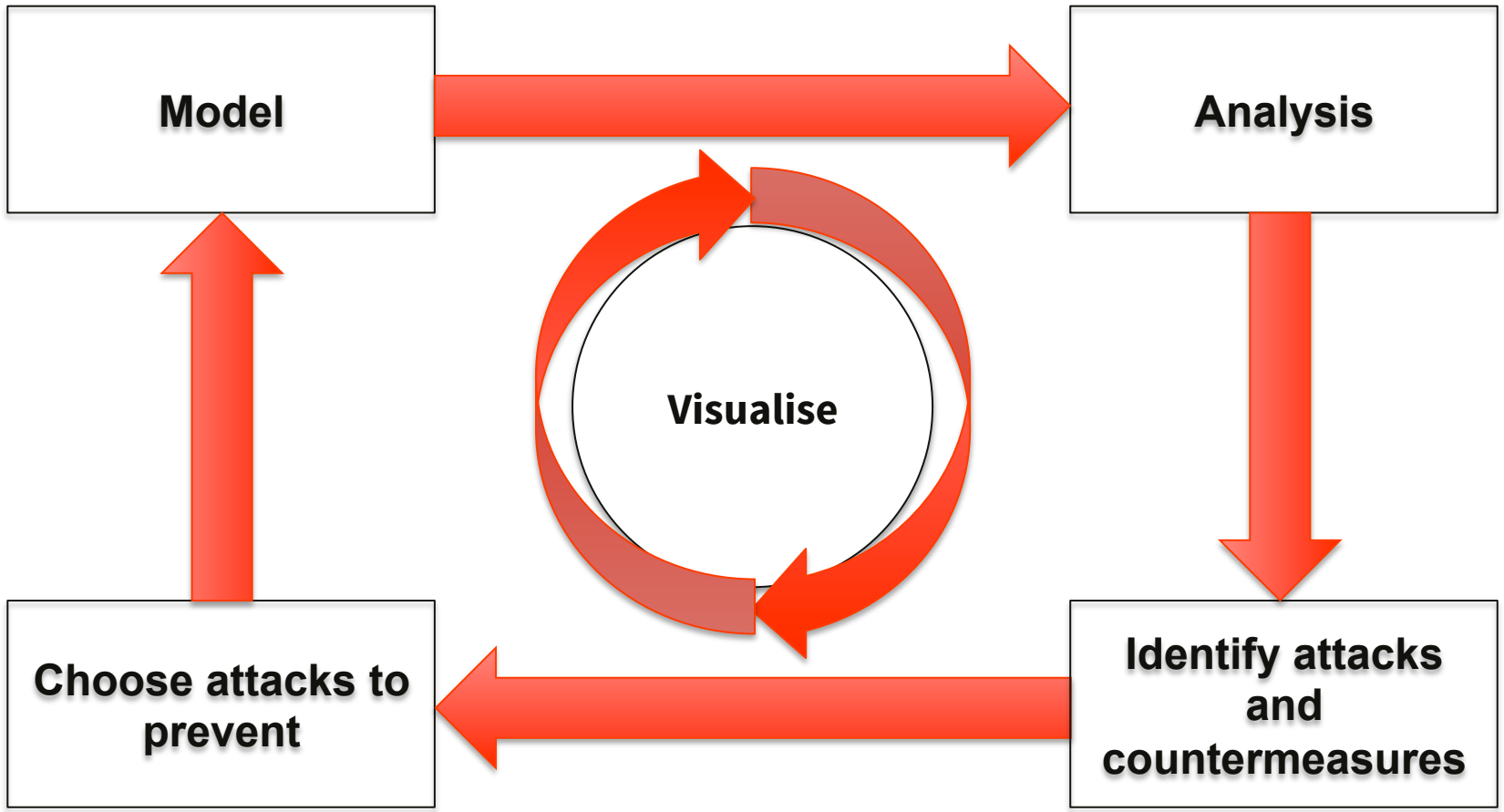
European Commission

September 14, 2015

SASSI

The TRESPASS Approach to Risk Assessment

- Information security threats to organisations have changed completely over the last decade
- New attacks cleverly exploit multiple organisational vulnerabilities, involving physical security and human behaviour.
- Defenders need to make rapid decisions regarding which attacks to block, as both infrastructure and attacker knowledge change rapidly.


Model
Analysis
Visualise
Choose attacks to prevent
Identify attacks and countermeasures

THREAT AGAINST SYSTEM

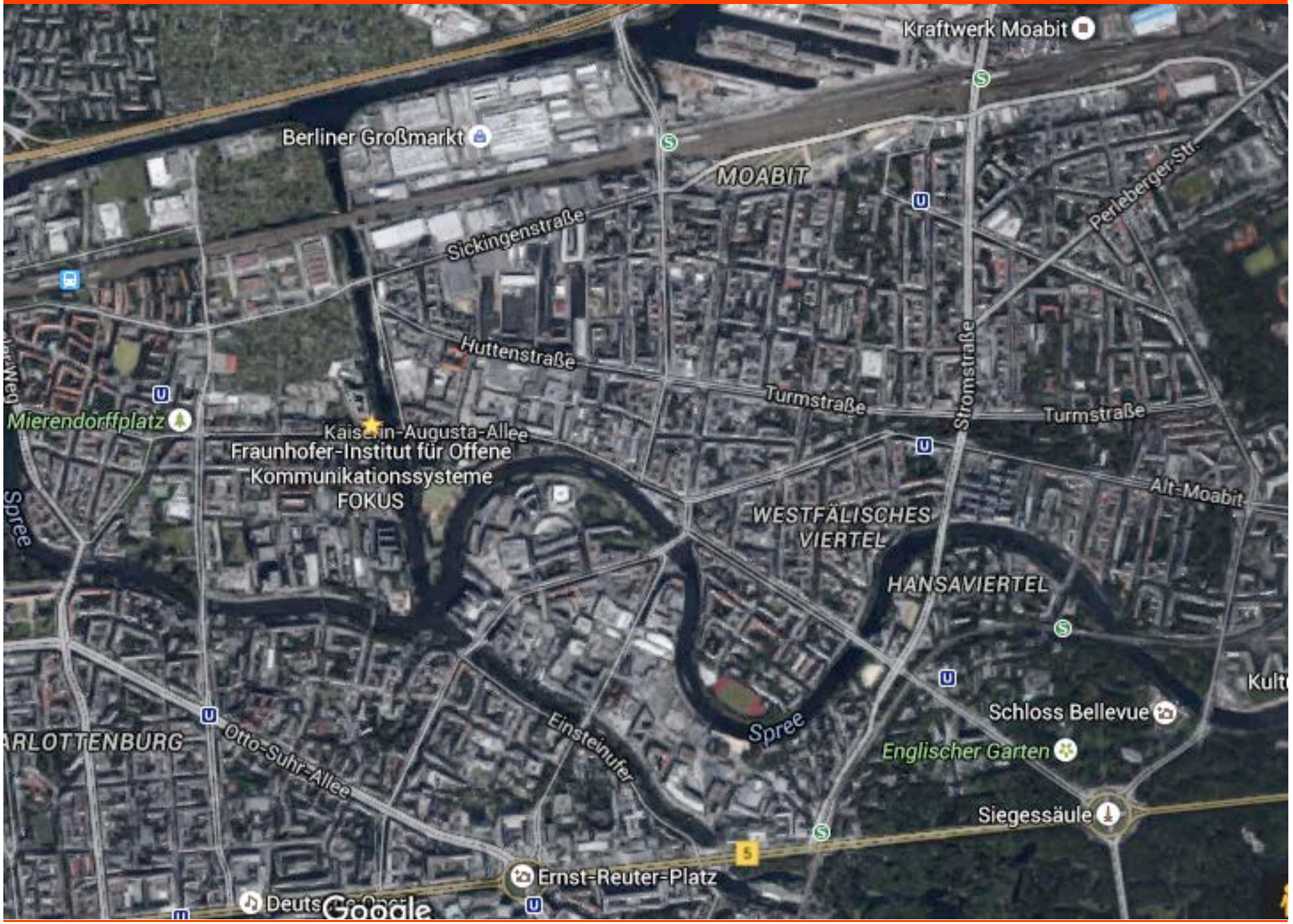
SUM OF THREATS AGAINST SUBYSTEMS

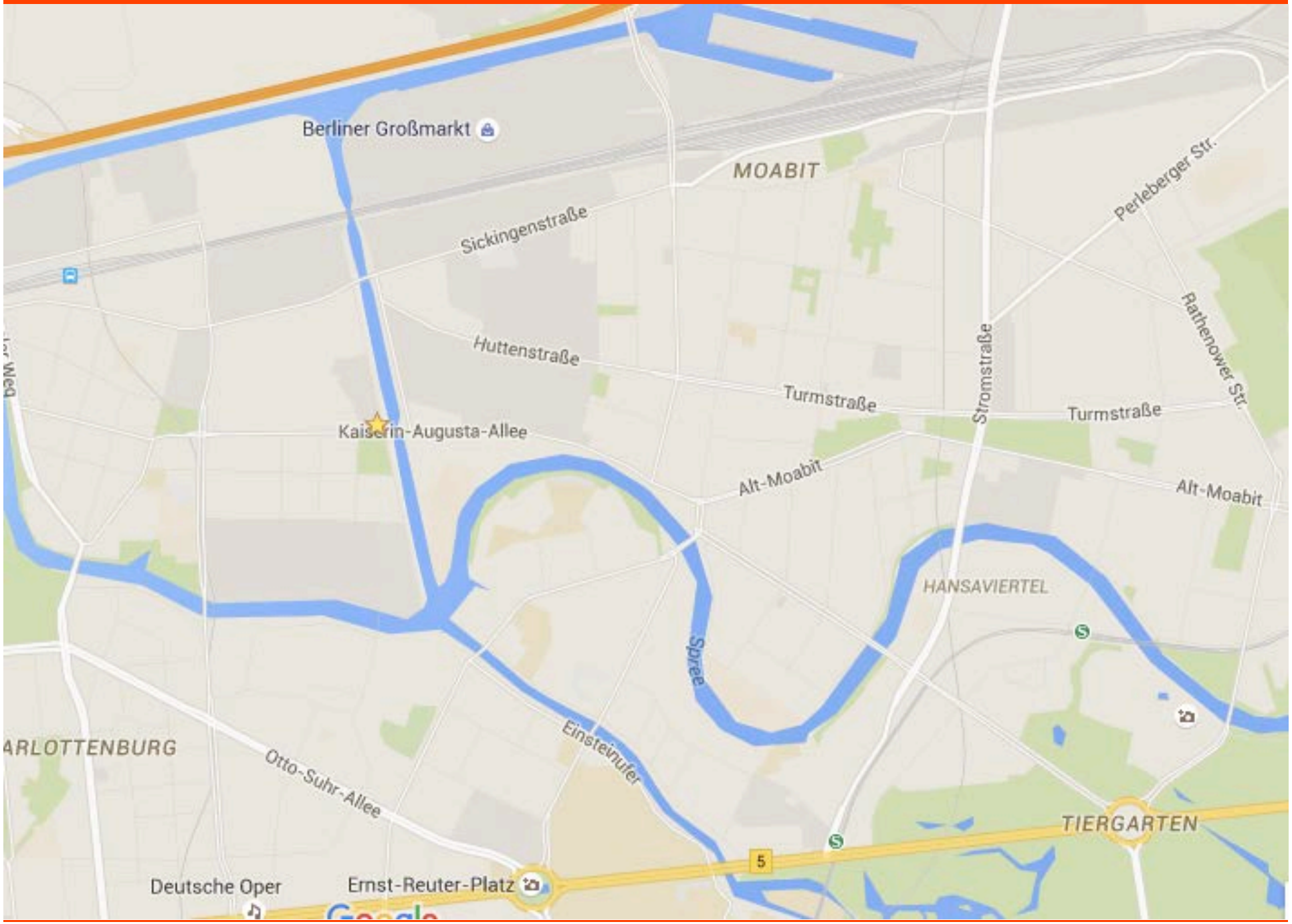
Software Systems

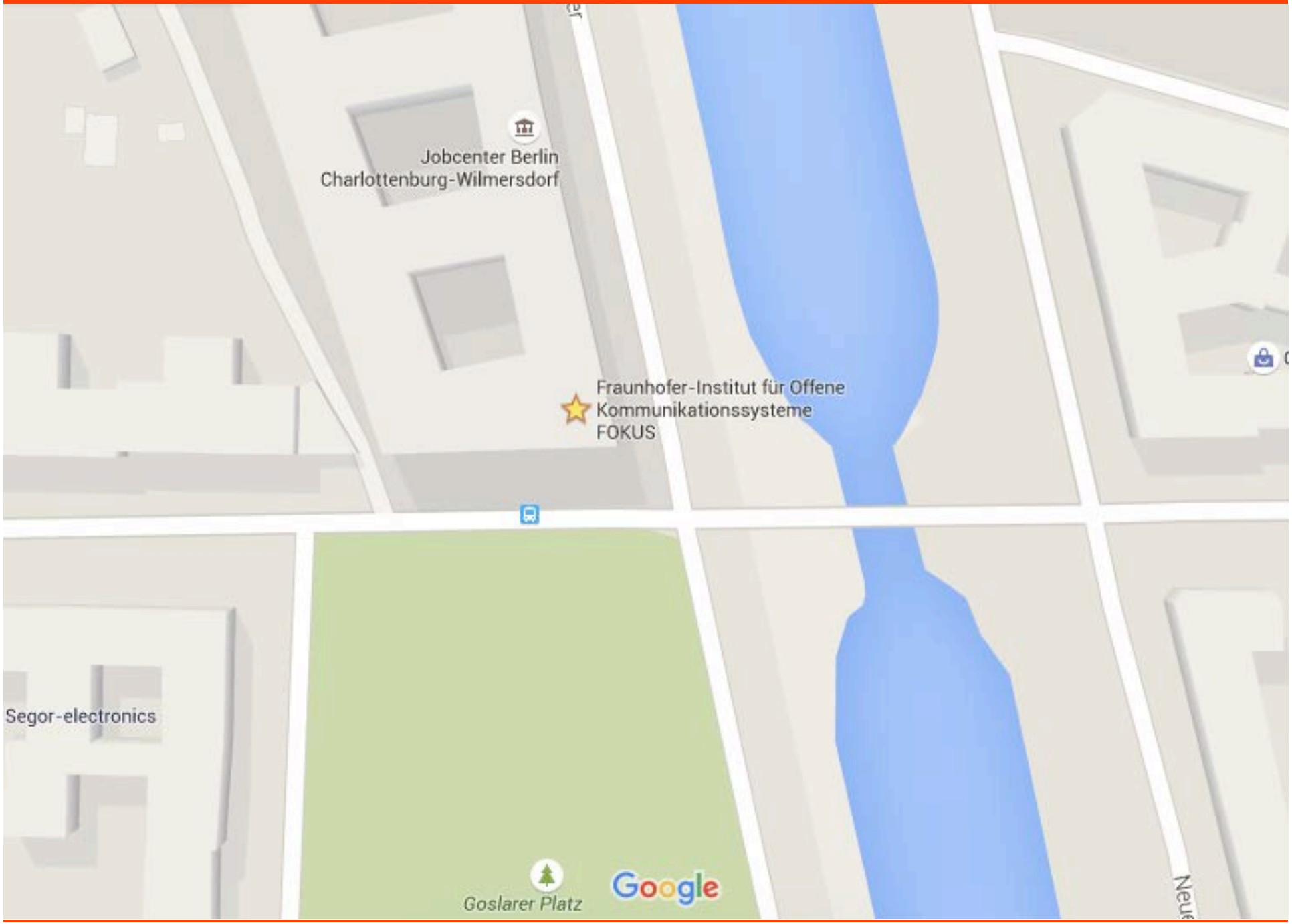
- Systems are not pure systems anymore
- Mixture of hardware, software, data, connections, human operators
- And their interactions

A dumb goal
is better than
the best tactics.

Günter Netzer


Kraftwerk Moabit
Berliner Großmarkt
MOABIT
Sickingenstraße
Perleberger Str.
Huttenstraße
Turmstraße
Stromstraße
Turmstraße
Mierendorffplatz
Kaiserin-Augusta-Allee
Fraunhofer-Institut für Offene Kommunikationssysteme FOKUS
Spree
WESTFÄLISCHES VIERTEL
Alt-Moabit
HANSAVIERTEL
Schloss Bellevue
Englischer Garten
Einsteinufer
Spree
Otto-Suhr-Allee
Siegessäule
Ernst-Reuter-Platz
Google

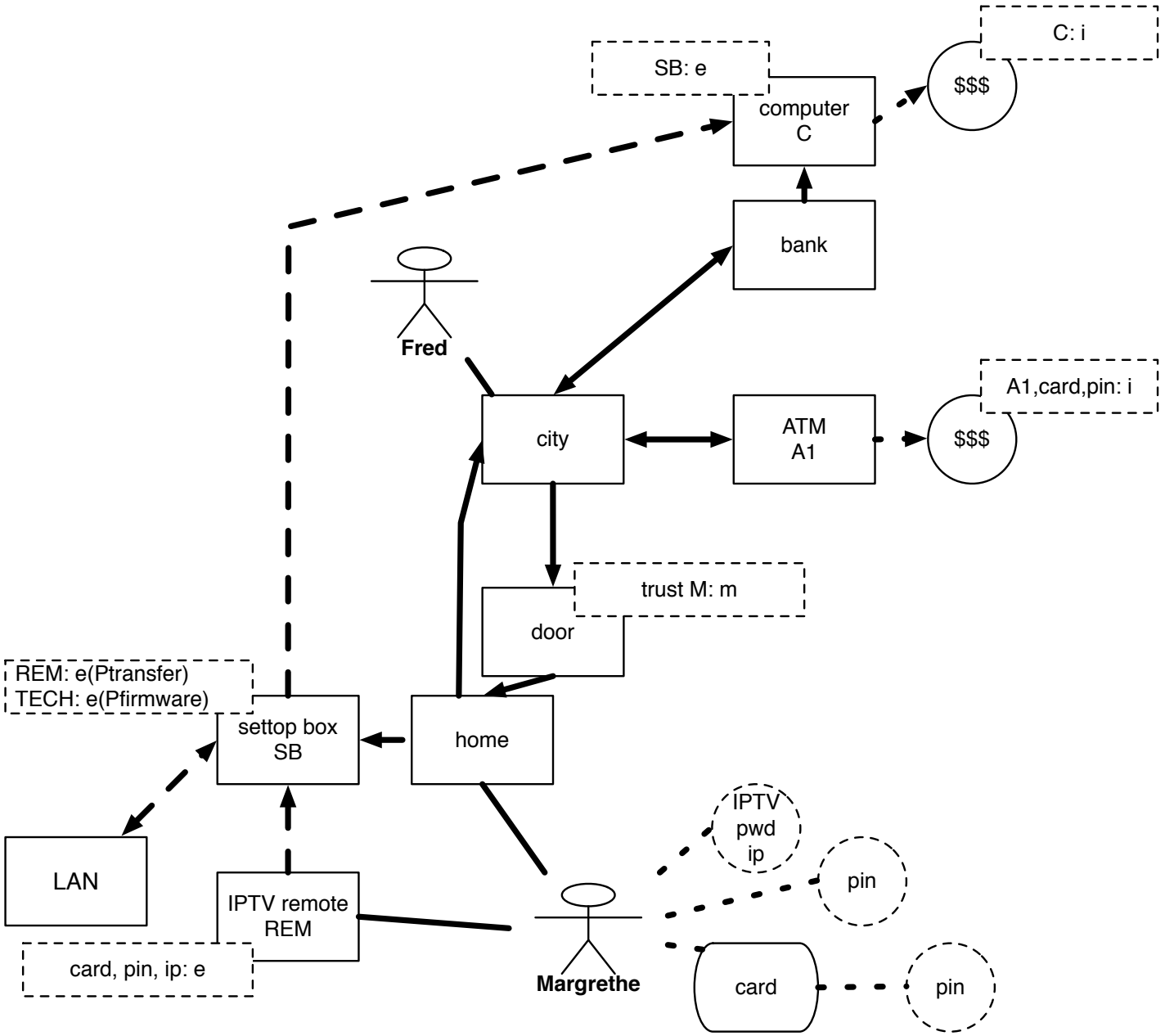

Berliner Großmarkt
MOABIT
Sickingenstraße
Perleberger Str.
Rathenower Str.
Huttenstraße
Turmstraße
Stromstraße
Turmstraße
Kaiserin-Augusta-Allee
Alt-Moabit
Alt-Moabit
HANSAVIERTEL
Spree
Einsteinufer
ARLOTTENBURG
Otto-Suhr-Allee
TIERGARTEN
5
Deutsche Oper
Ernst-Reuter-Platz


Jobcenter Berlin
Charlottenburg-Wilmersdorf
Fraunhofer-Institut für Offene
Kommunikationssysteme
FOKUS
Segor-electronics
Goslarer Platz
Google

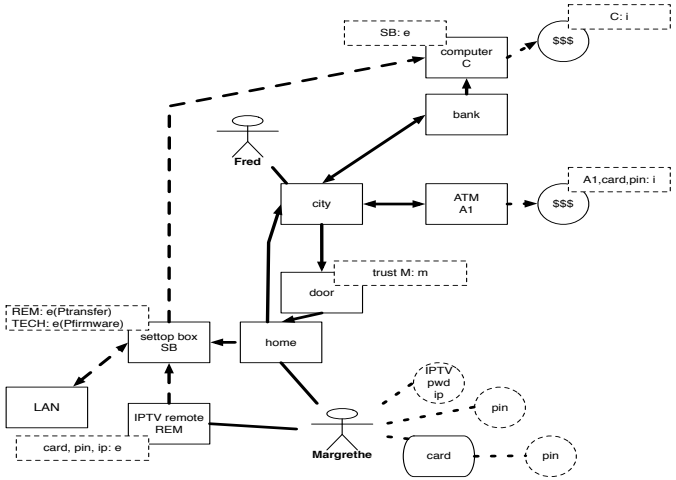

New perspectives on the scenario:


CITY

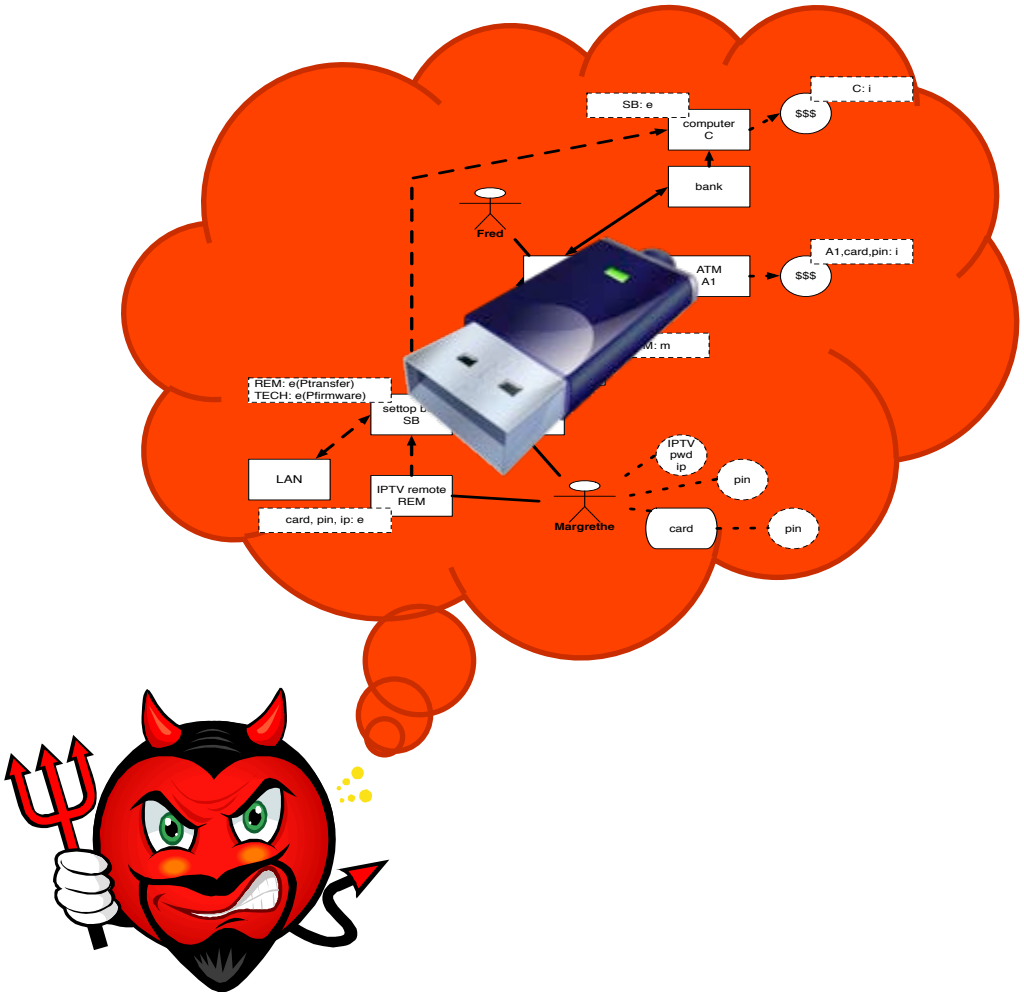
Green = artifacts and devices Blue = data and applications Yellow = business roles and actors

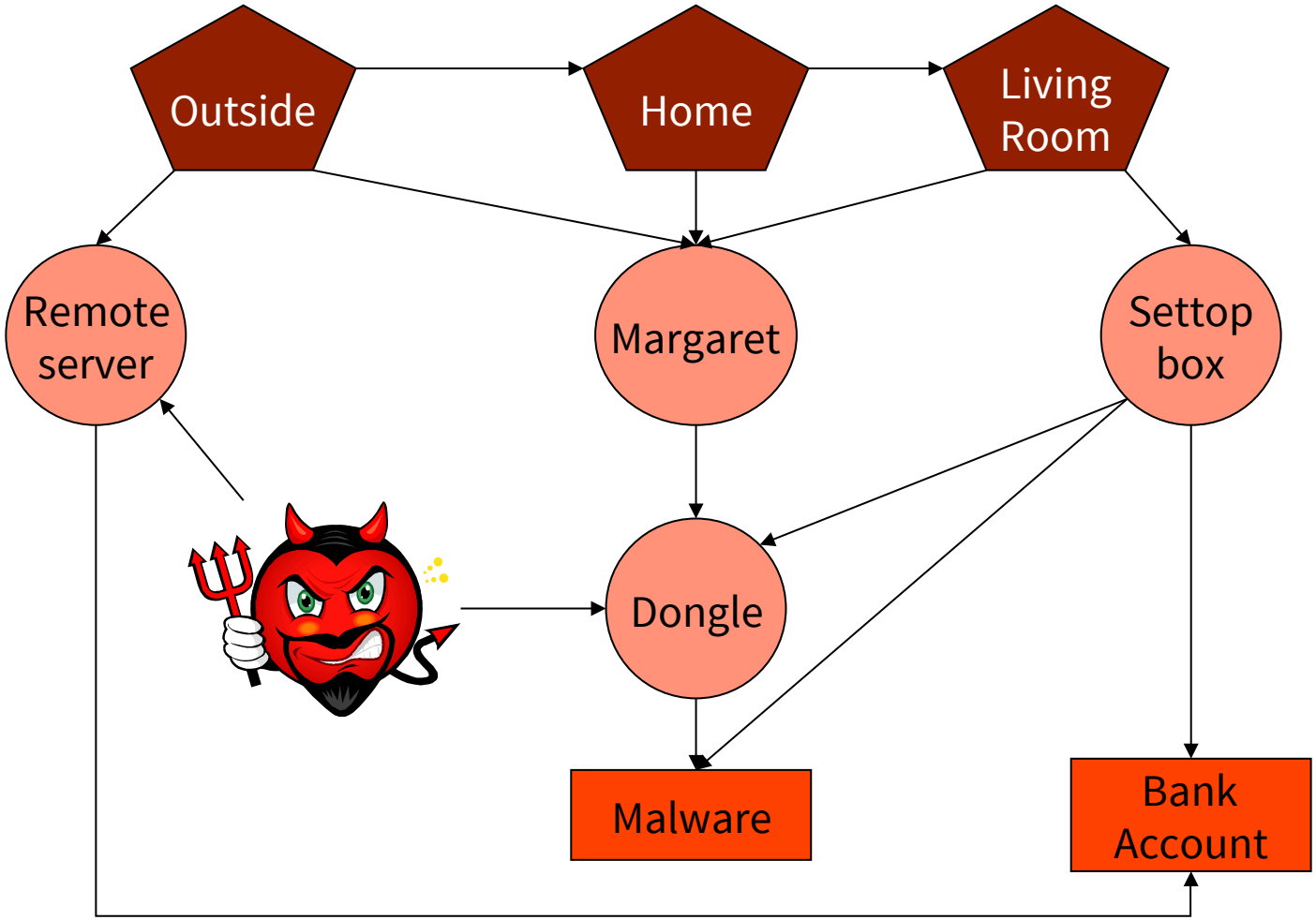

SB: e
computer
C
C: i
$$$
bank
Fred
city
ATM
A1
A1,card,pin: i
$$$
trust M: m
door
REM: e(Ptransfer)
TECH: e(Pfirmware)
settop box
SB
home
LAN
IPTV remote
REM
card, pin, ip: e
Margrethe
IPTV
pwd
ip
pin
card
pin

How do I get the money?



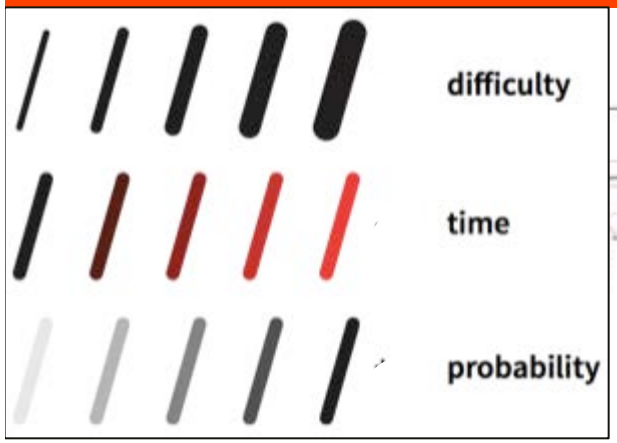
Ask the TRE$_S$PASS Attack Navigator!

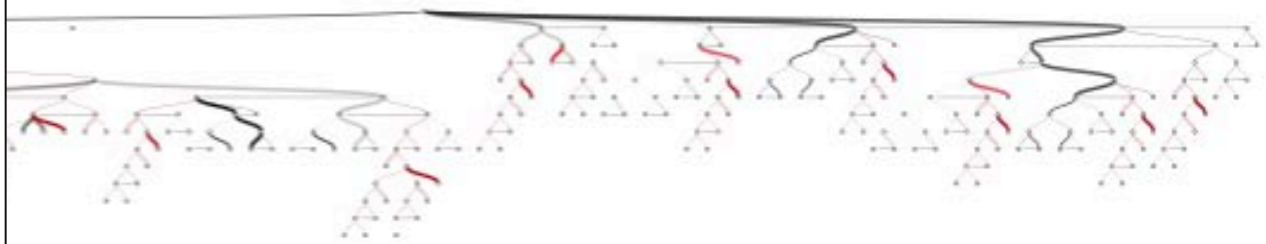


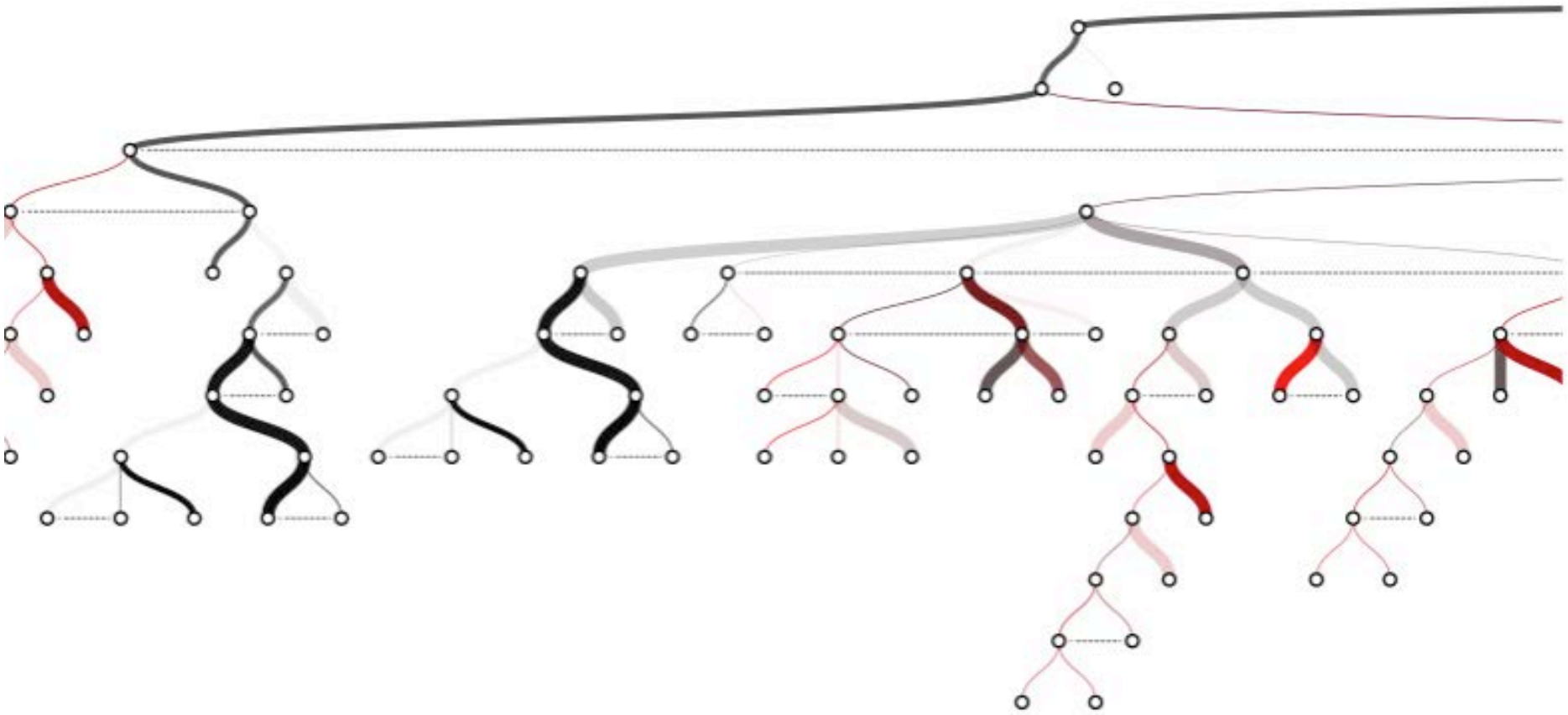

Outside
Home
Living Room
Remote server
Margaret
Settop box
Dongle
Malware
Bank Account

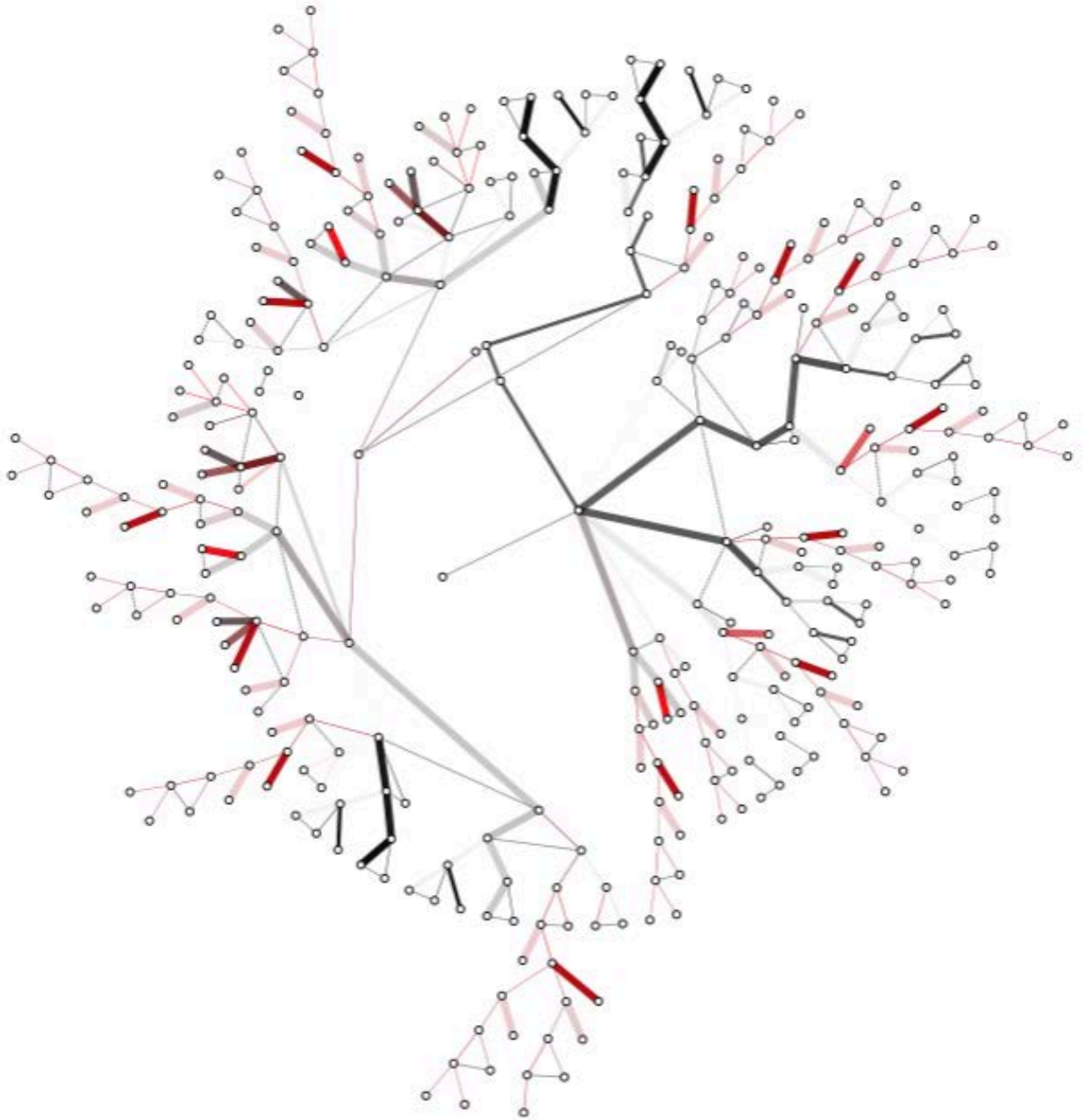
The Attack Navigator

- Tool to support prediction, prioritisation, and prevention of complex attack scenarios.
- Also an environment where all tools developed within the project can be viewed, accessed and connected.
- Generates attacks that represent routes of attackers


difficulty
time
probability



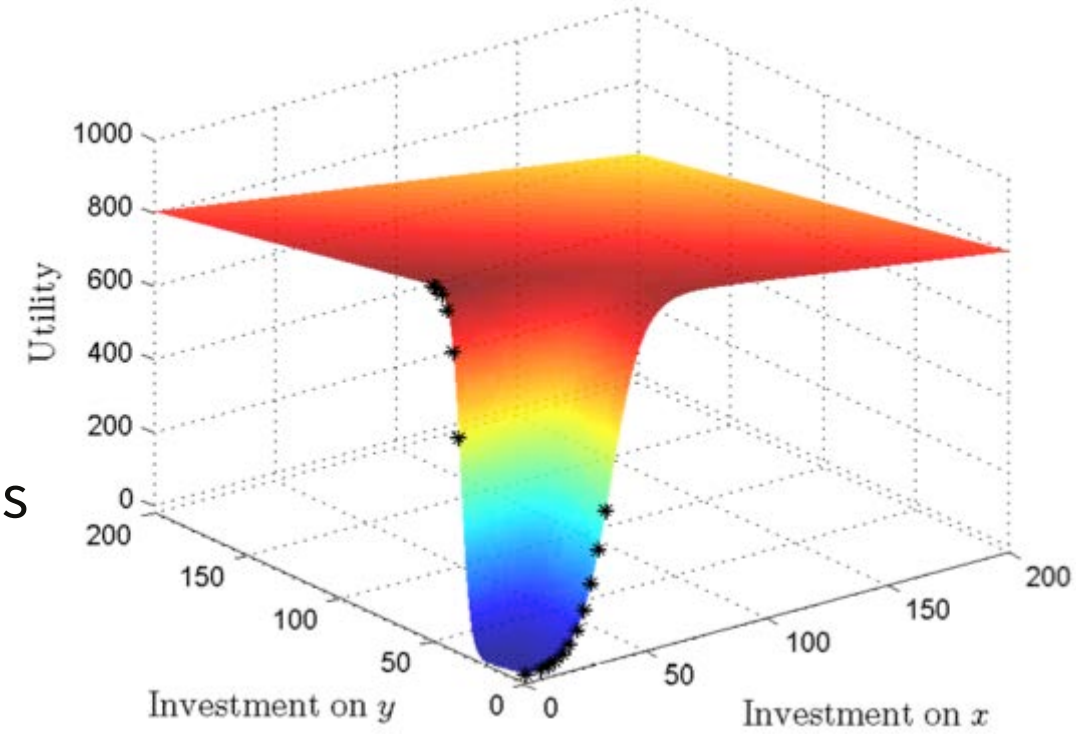




Analyse adversary profiles and strategies

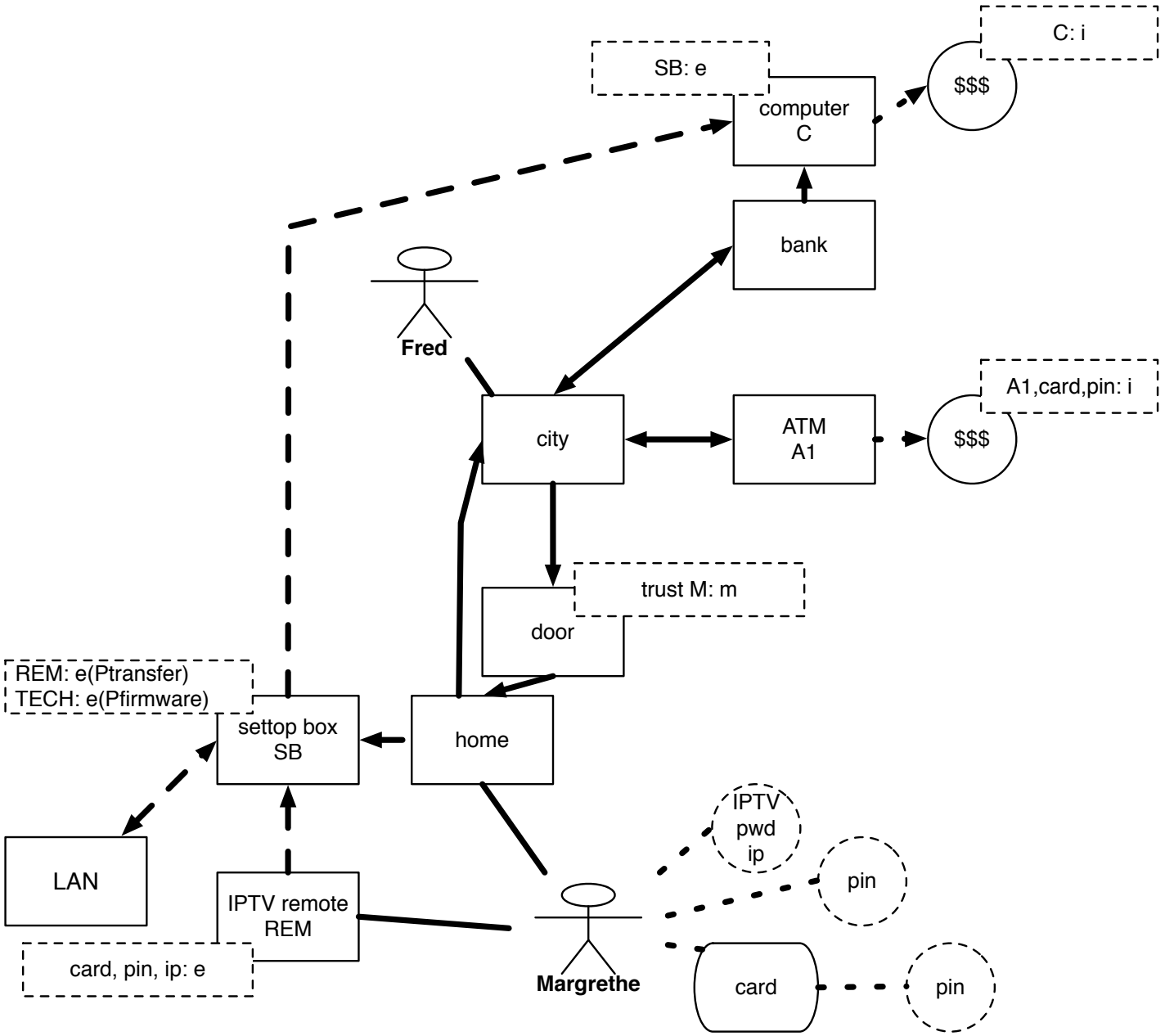
Example parameters

- Goals (utility function)
- Skill
- Budget
- Time
- Initial knowledge/access



Relevance for Security Assessment

- For technical systems, the models can be automatically extracted
- The model can be applied to software systems
- Orthogonal to static analysis


SB: e
computer
C
C: i
$$$
bank
Fred
city
ATM
A1
A1,card,pin: i
$$$
trust M: m
door
REM: e(Ptransfer)
TECH: e(Pfirmware)
settop box
SB
home
LAN
IPTV remote
REM
card, pin, ip: e
Margrethe
IPTV
pwd
ip
pin
card
pin

Conclusions

- We need new concepts to guide risk assessment.
- Attack navigator uses organisational maps and attacker profiles
  - To identify attacks involving several domains
- Identifies attacks to guide risk assessment and governance
- Serious play and novel visualisation techniques identifying and refining models

Contact

predict
prioritise
prevent
TREsPASS

www.trespass-project.eu

contact@trespass-project.eu

Contact us to join our public mailing list!

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 318003 (TREsPASS). This publication reflects only the author's views and the Union is not liable for any use that may be made of the information contained herein.

SEVENTH FRAMEWORK PROGRAMME

European Commission

SASS15

Threat modelling using attack trees

Jan Willemson, Cybernetica

Abstract:

The concept of hierarchical risk assessment has been around a few decades, but the corresponding methods for this kind of approach are still very immature. In this talk we will take a particular look at attack trees and the challenges one has to tackle when trying to build an attack tree based threat model. We will talk about the root node identification, choosing the correct level of abstraction, quantitative risk assessment and the limitations of the attack tree methodology.

Vita:

Jan Willemson has been working on data security and cryptography since 1998 when he joined Cybernetica. He defended his PhD thesis on digital time-stamping at Tartu University (Estonia) in 2002 and has since been active in a variety of research areas including socio-technical risk analysis, secret-shared multi-party computations, security economics and attack trees. He is an author of more than 40 research papers published in major international venues.

Fraunhofer FOKUS

CYBERNETICA

predict
prioritise
prevent
TREsPASS

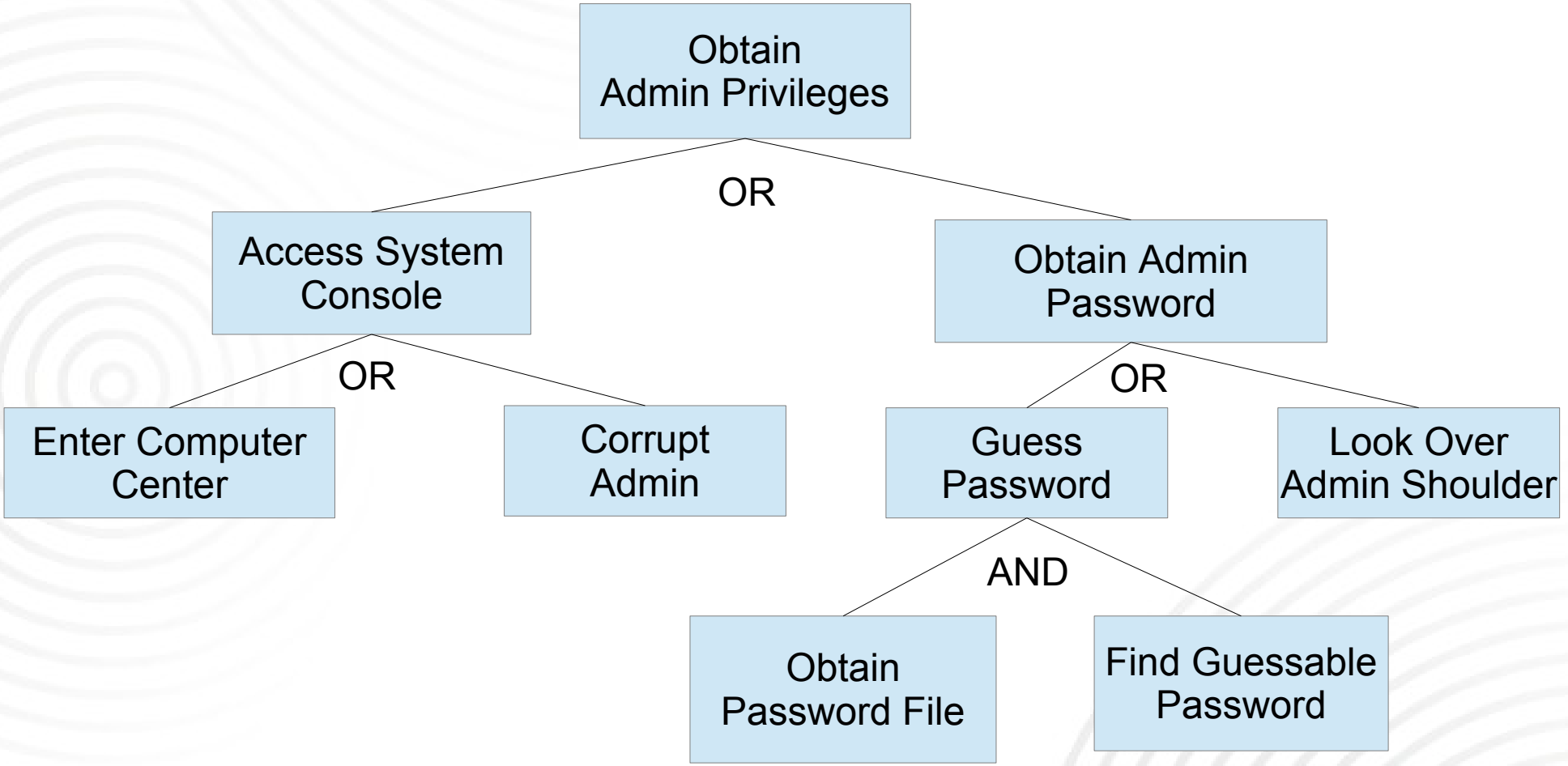
Threat modelling using attack trees

Jan Willemson

European Commission
SEVENTH FRAMEWORK PROGRAMME

15.09.2015

Threat logic trees (Weiss 1991)



predict prioritise prevent TRESPASS

CYBERNETICA

Attack trees (Schneier 1999)

- Hierarchical threat modelling paradigm
- Start from the *root attack*
- **For** every leaf node that is not yet simple enough **do**:
  - Split it into simpler attacks, either *one* or *all* of which are required to implement the parent node
  - Call these OR and AND nodes, respectively
  - **Loop**

Devil is in the details

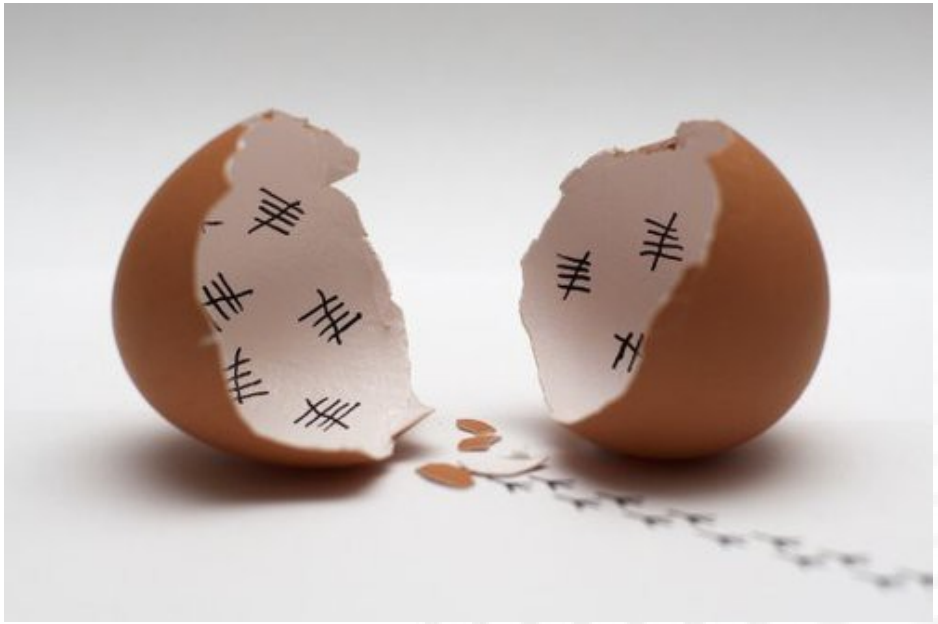
- How do you select the root node?
- When do you break out of the loop?
- What is the correct splitting of the attacks?

Selecting the root node

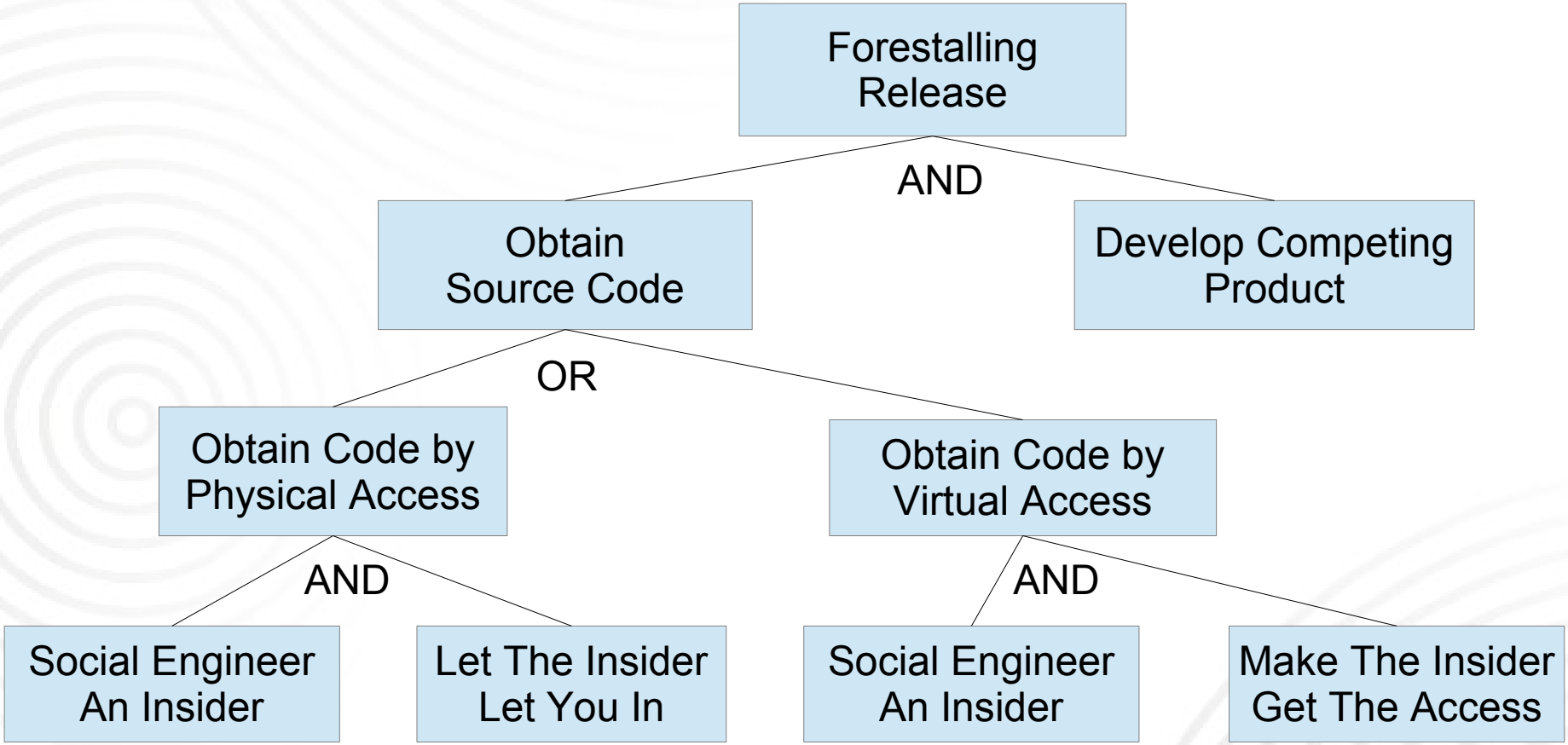
- Attack tree method takes the *attacker's viewpoint*
- Hence, the root node should reflect the attacker's target, not the defender's assessment of his assets
- You have to know what the attacker is after
  - Money? Assets to sell?
    - If so, you have to estimate, how much the assets are worth for the *attacker*, not *you*
  - Fame?
  - Satisfying his curiosity?
  - Causing damage or disruption?
- Different attacker goals may give totally different attack trees

Breaking out of the loop

- One should end the splitting process when it becomes possible to estimate parameters of the attacks
  - Cost
  - Probability of success
  - Probability of getting caught
  - Potential penalties
  - Technical skill required
  - Social skill required
  - Time required



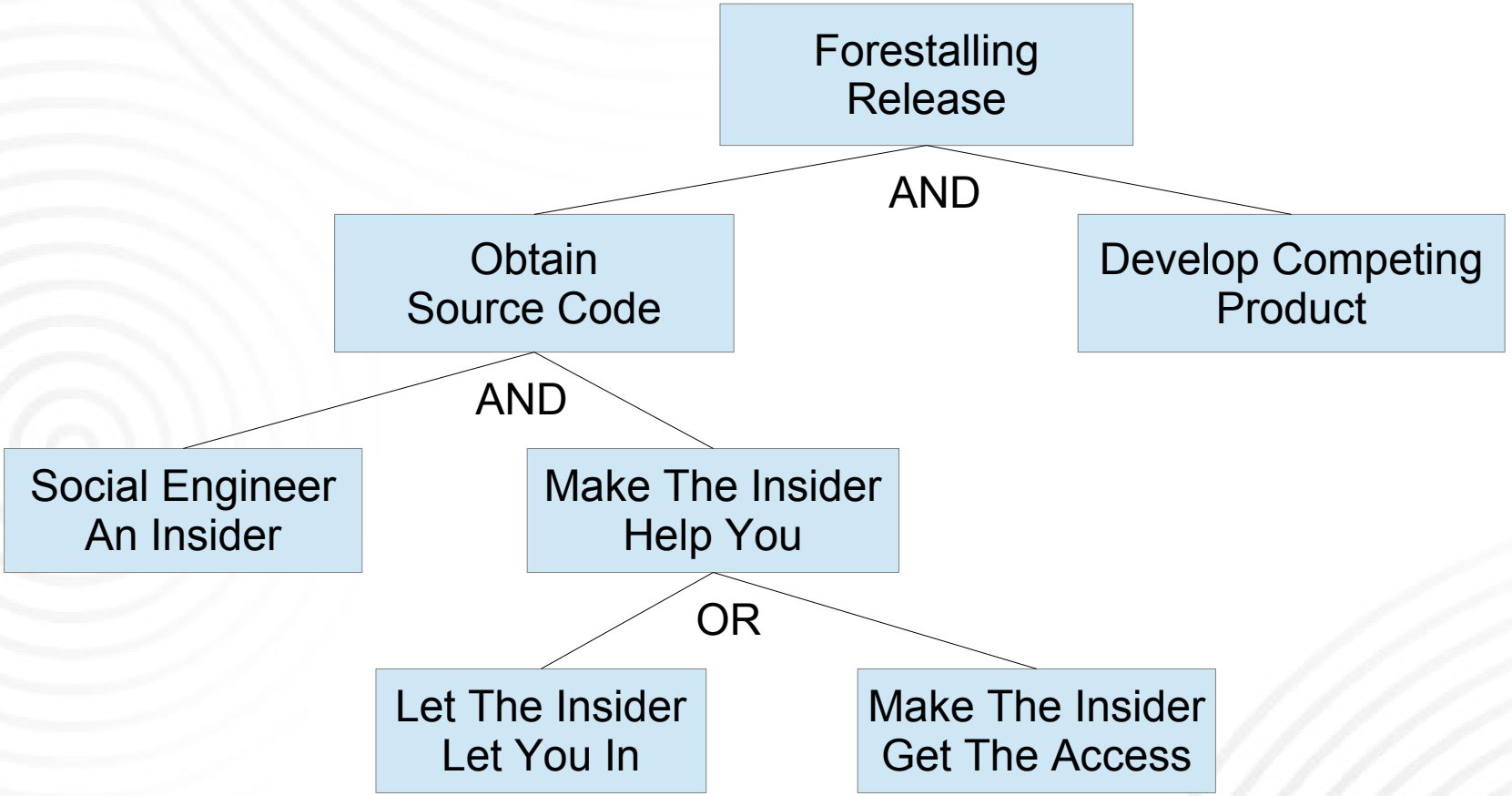
Splitting the attacks



predict prioritise prevent TRESPASS

CYBERNETICA

Splitting the attacks



predict prioritise prevent
TRESPASS
CYBERNETICA

What to do with the attack tree?

- When the elementary attacks are assigned parameter values, quantitative questions can be asked and answered:
  - What is the cheapest attack?
  - What is the attack requiring the smallest skill set?
  - What attack is the most profitable one for the attacker?

predict
prioritise
prevent
TREsPASS

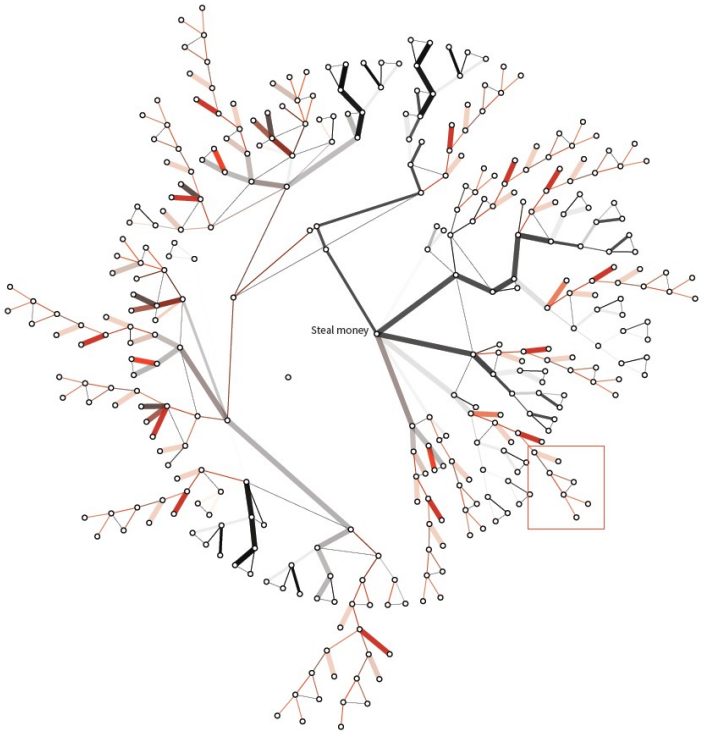
CYBERNETICA

Developing the attack tree – Cathedral or Bazaar?

- Cathedral approach
  - “The Bishop” is drawing the tree and others are giving feedback
  - “The Bishop” will decide which comments to implement
- Bazaar approach
  - All the views are equal
  - Comments are voted on, discussed until consensus, or alike
- Both literature and experience seems to show that the Cathedral approach works better
- Anyway, there is no canonical representation of the attack tree

Developing the attack tree – TREsPASS perspective

- Ideally, the end user does not need to see the attack tree at all
- The user thinks in terms of his environment
  - Assets
  - Actors
  - Access policies
  - Processes
- TREsPASS aims to prove that based on the environment description, building and analysing the attack tree can be done automatically



predict
prioritise
prevent
TREsPASS

CYBERNETICA

Challenges and conclusions

- Attack trees tend to grow large
- There is no canonical representation
- Parameter values are hard to estimate
- It is not clear which level of abstraction is a good one
- Still, I believe attack trees exist in nature, so studying them is inevitable
- Even if quantified risk assessment on top of attack trees proves mission impossible, attack trees will still be a valuable aid to visual reasoning about the risks

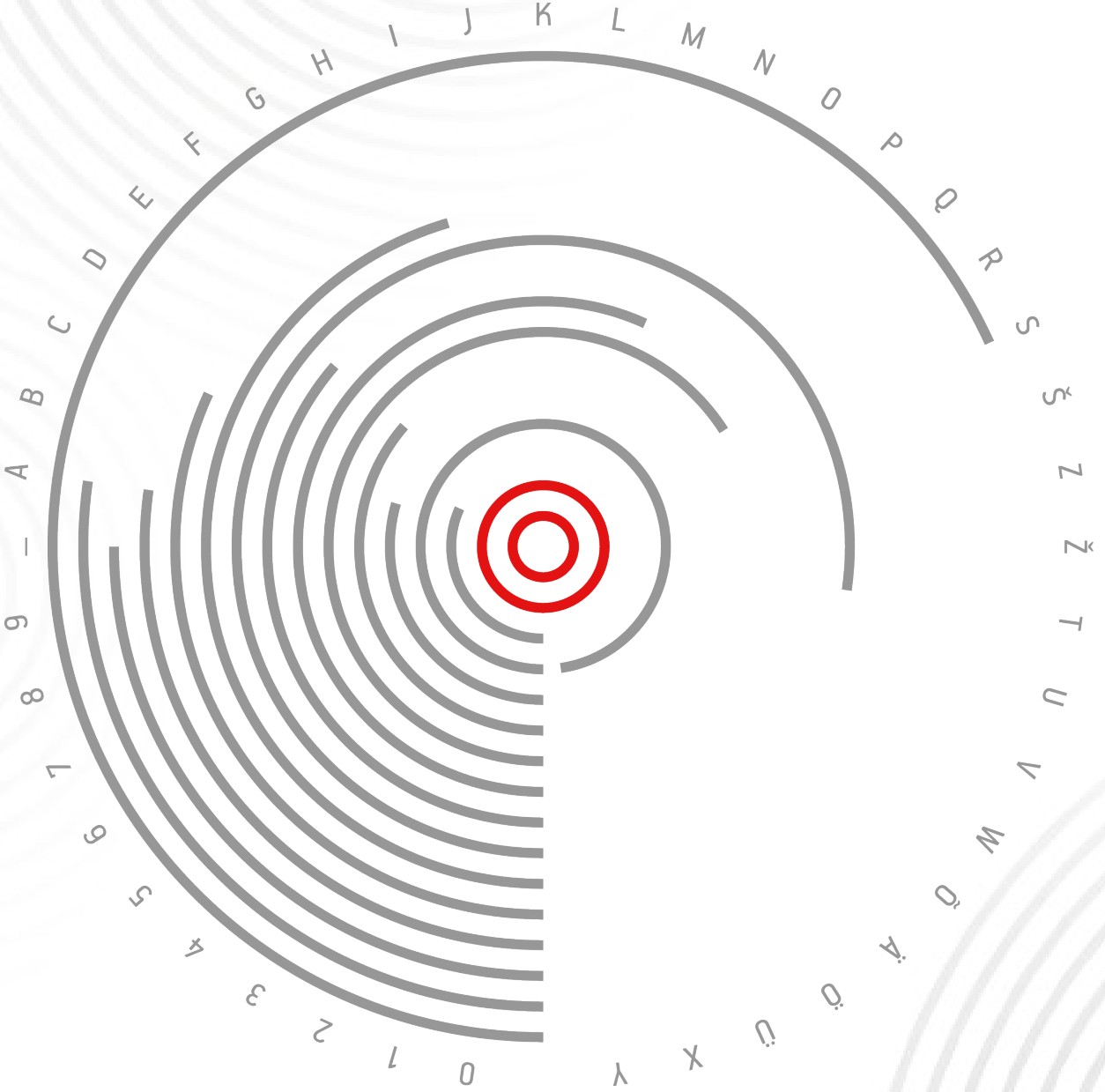
Thank you!

- Questions?
- janwil@cyber.ee
- http://www.cyber.ee
- http://trespass-project.eu

predict
prioritise
prevent
TRESPASS

CYBERNETICA

CYBERNETICA



CYBERNETICA

SASS15

Tool-supported cyber-risk assessment

Bjørnar Solhaug, SINTEF ICT

Abstract:

This tutorial gives an introduction to cyber-risk assessment and demonstrates how it can be conducted using the CORAS risk assessment tool. The presentation includes an introduction to the essential elements that we need to understand in order to assess cyber-risk in a methodic and adequate manner: What is a cyber-system, what is a cyber-threat, what is cybersecurity, and what is cyber-risk?

Vita:

Bjørnar Solhaug is a senior researcher at SINTEF ICT in Norway and holds a PhD in information science from the University of Bergen. His research interests include risk analysis, threat odelling, information security, cybersecurity, trust management and formal languages. He is has contributed to the development of the CORAS method and is one of the authors of the book "Model-Driven Risk Analysis: The CORAS Approach" (Springer, 2011).

Fraunhofer FOKUS

SASS '15

Tool-Supported Cyber-Risk Assessment

Security Assessment for Systems, Services and Infrastructures (SASSI'15)

Bjørnar Solhaug (SINTEF ICT)

Berlin, September 15, 2015

RASEN

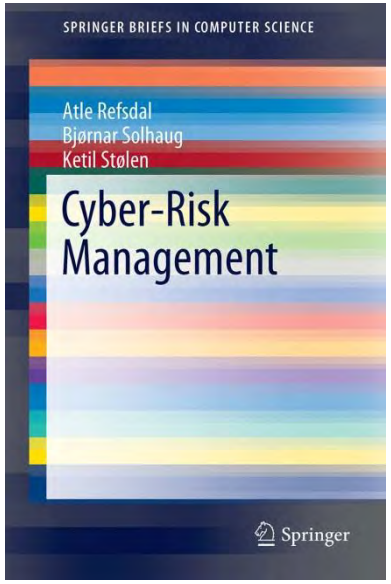
SEVENTH FRAMEWORK PROGRAMME

Me

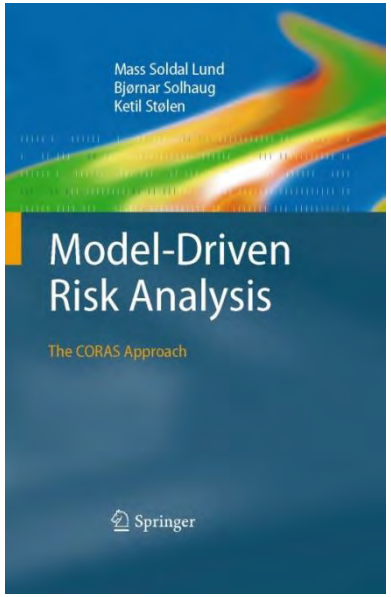
- Bjørnar Solhaug
  - Bjornar.Solhaug@sintef.no
  - www.solhaugb.byethost11.com
- Research scientist at SINTEF ICT since 2010
  - www.sintef.no
- MSc in Logic, Language and Information, University of Oslo, 2004
- PhD in Information Science, University of Bergen, 2009
- Co-author of two books:
  - *Cyber-Risk Management* (Springer, 2015)
  - *Model-Driven Risk Analysis – The CORAS Approach* (Springer, 2015)

Background to this Tutorial

- Atle Refsdal, Bjørnar Solhaug and Ketil Stølen: *Cyber-Risk Management* (Springer, 2015)



- Mass Soldal Lund, Bjørnar Solhaug and Ketil Stølen: *Model-Driven Risk Analysis – The CORAS Approach* (Springer, 2011)



- CORAS resources, including free tool download and demo video: http://coras.sourceforge.net

Relevant Standards

- **ISO 31000** – Risk management – Principles and Guidelines (2009)
- **ISO/IEC 27000** – Information technology – Security techniques – Information security management systems – Overview and vocabulary (2014)
- **ISO/IEC 27001** – Information technology – Security techniques – Information security management systems – Requirements (2013)
- **ISO/IEC 27005** – Information technology – Security techniques – Information security risk management
- **ISO/IEC 27032** – Information technology – Security techniques – Guidelines for cybersecurity

Overview

- Risk assessment
  - Background terminology
  - Risk assessment process
- Cyber-risk assessment
  - Cybersecurity and cyber-risk terminology
  - Cyber-risk assessment process
- Example and demo
  - Smart Grid example
  - Demo of CORAS tool

Risk Assessment

What is Risk?

- Health
- Safety
- Security
- Compliance (legal and regulatory)
- Environmental protection
- Product quality
- Reputation
- Defense
- Finance
- …

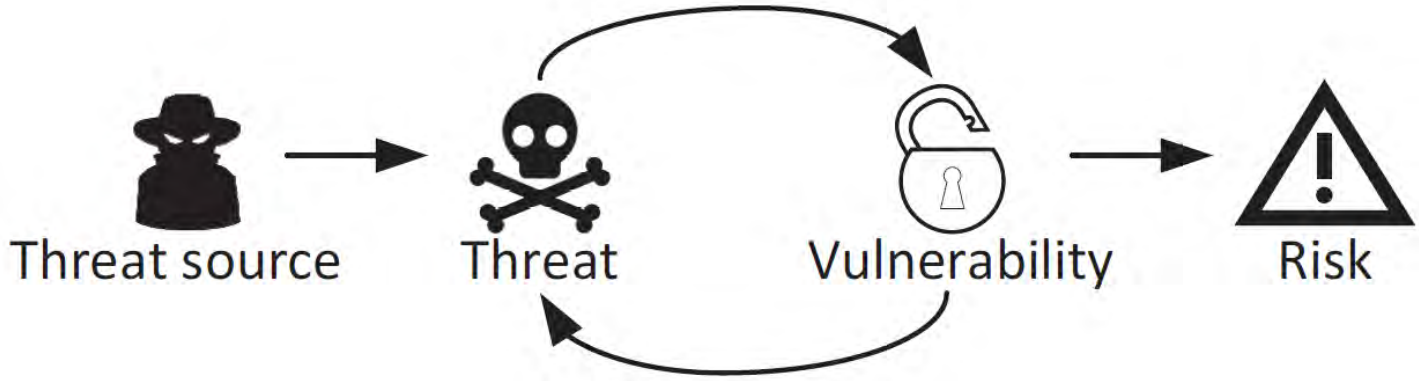
- What do we want to protect?
- What do we want to achieve?
- What do we want to protect from?

Definitions 1/2

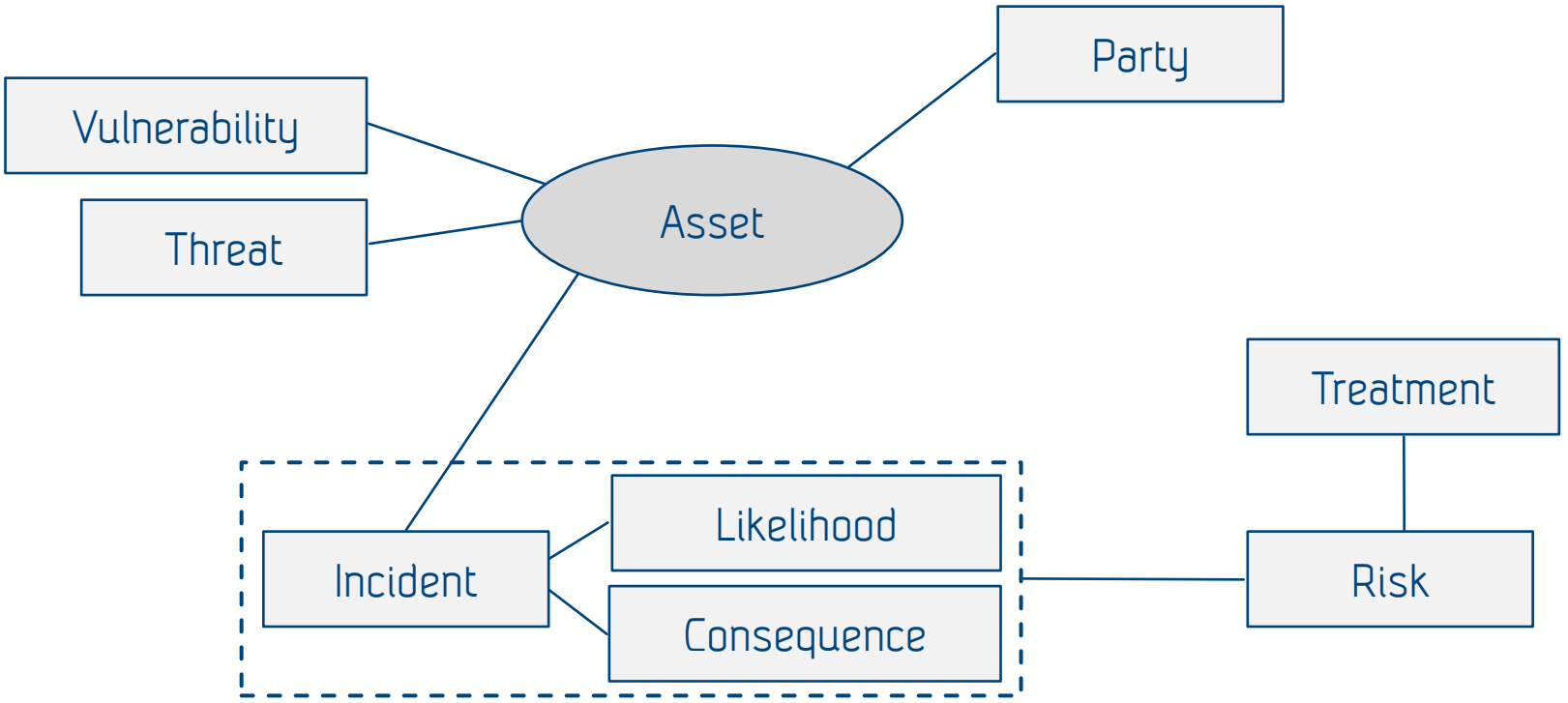
- A **risk** is the likelihood of an incident and its consequence for an asset
- An **incident** is an event that harms or reduces the value of an asset
- An **asset** is anything of value to a party
- A **party** is an organization, company, person, group or other body on whose behalf a risk assessment is conducted
- A **likelihood** is the chance of something to occur
- A **consequence** is the impact of an incident on an asset in terms of harm or reduced asset value
- **Risk level** is the magnitude of a risk as derived from its likelihood and consequence

Definitions 2/2

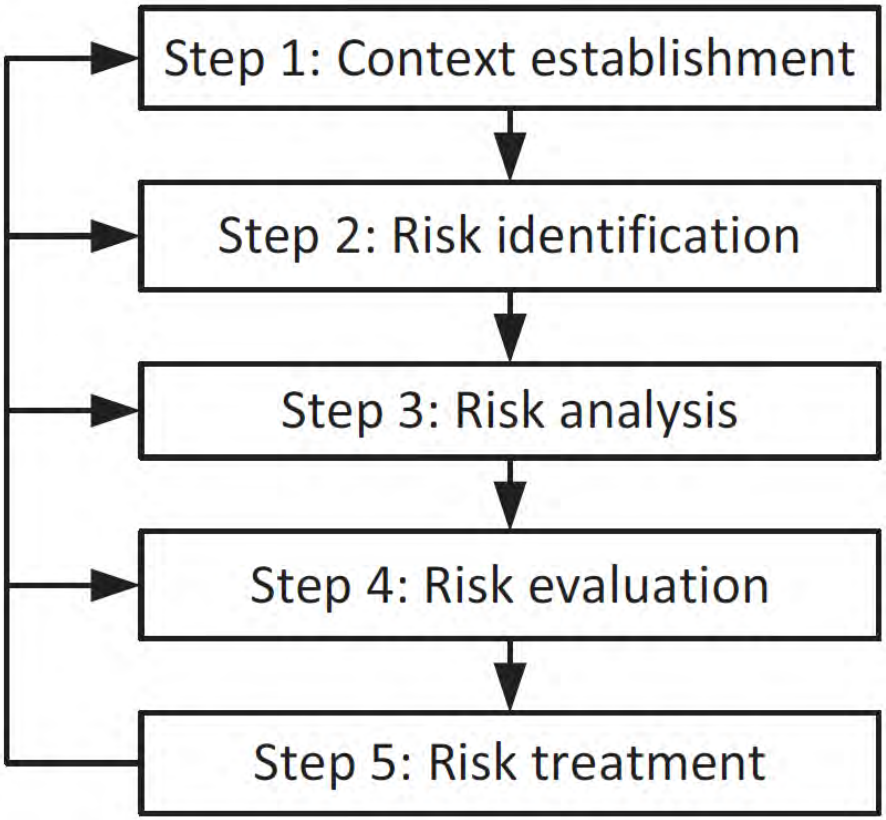
- A **vulnerability** is a weakness, flaw or deficiency that can be exploited by a threat to cause harm to an asset
- A **threat** is an action or event that is caused by a threat source and that may lead to an incident
- A **threat source** is the potential cause of an incident
- A **treatment** is an appropriate measure to reduce risk level



Concept Overview



Risk Assessment Process



Cyber-Risk Assessment

Cyberspace and Cyber-Systems

- Cybersecurity concerns systems that make use of cyberspace

- A **cyberspace** is a collection of interconnected computerized networks, including services, computer systems, embedded processors and controllers, as well as information in storage or transit
  - For most organizations and other stakeholders, cyberspace is for all practical purposes synonymous with the Internet
  - The Internet is a global cyberspace in the public domain
- A **cyber-system** is a system that makes use of a cyberspace
  - A cyber-system may include information infrastructures, as well as other entities that are involved in the business processes and other behavior of the system
  - Cyber-systems are therefore part of the structure of most organizations

Cybersecurity

- **Cybersecurity** is the protection of cyber-systems against cyber-threats
  - Cyber-threats are those that arise via a cyberspace, and are therefore a kind of threat that any cyber-system is exposed to
- A **cyber-threat** is a threat that exploits a cyberspace
  - A cyber-threat can be *malicious*
    - For example DoS attack and malware injection attacks that are caused by intention
  - A cyber-threat can be *non-malicious*
    - For example system crash due to programming error, or some accidental loss of Internet connection

Remark on Cybersecurity

- What defines cybersecurity is not what we seek to protect, but rather what we seek to *protect from*

- Cybersecurity is not defined by the kinds of assets that are to be protected, but rather by the kinds of *threats* to assets
  - The assets of concern depend on the organization and the cyber-system in question
  - Often, cybersecurity concerns the protection of information assets and information infrastructure assets
  - However, cybersecurity must not be confused with information security or critical infrastructure protection

Cybersecurity vs. Information Security

- Information security is the preservation of confidentiality, integrity and availability of data
  - Information can come in any form: Electronic, material, knowledge, …
- Information in all formats need to be protected from threats of any kind
  - Physical, human, technology related, natural causes, …
- Cybersecurity concerns the protection from threats that use cyberspace
  - Various forms of information assets are relevant, but also others like information infrastructures, compliance, revenue, …

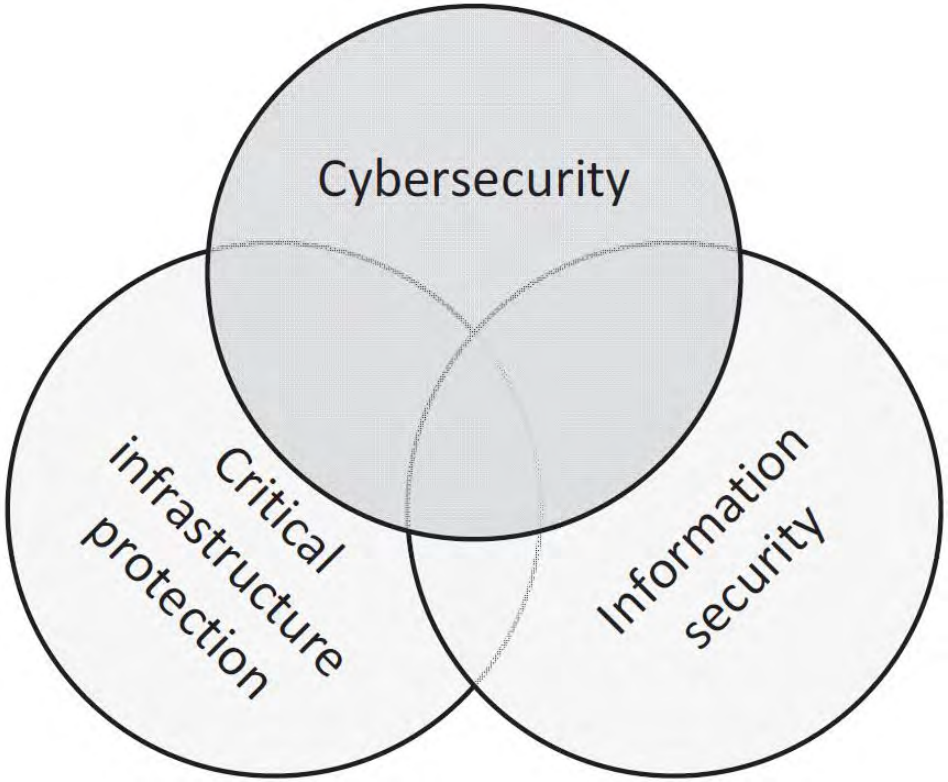
- There is overlap between the two, but:
  - Cybersecurity goes beyond information security
  - Information security goes beyond cybersecurity

Cybersecurity vs. Critical Infrastructure Protection

- Critical infrastructure protection (CIP), or infrastructure security, is concerned with the prevention of the disruption, disabling, destruction or malicious control of infrastructure
  - Telecommunication, transportation, finance, power supply, emergency services, …
- Many critical infrastructures use cyberspace and are therefore cyber-systems
  - Cybersecurity often involves CIP, but is not limited to CIP
  - CIP may involve cybersecurity, but only when the infrastructure is a cyber-system

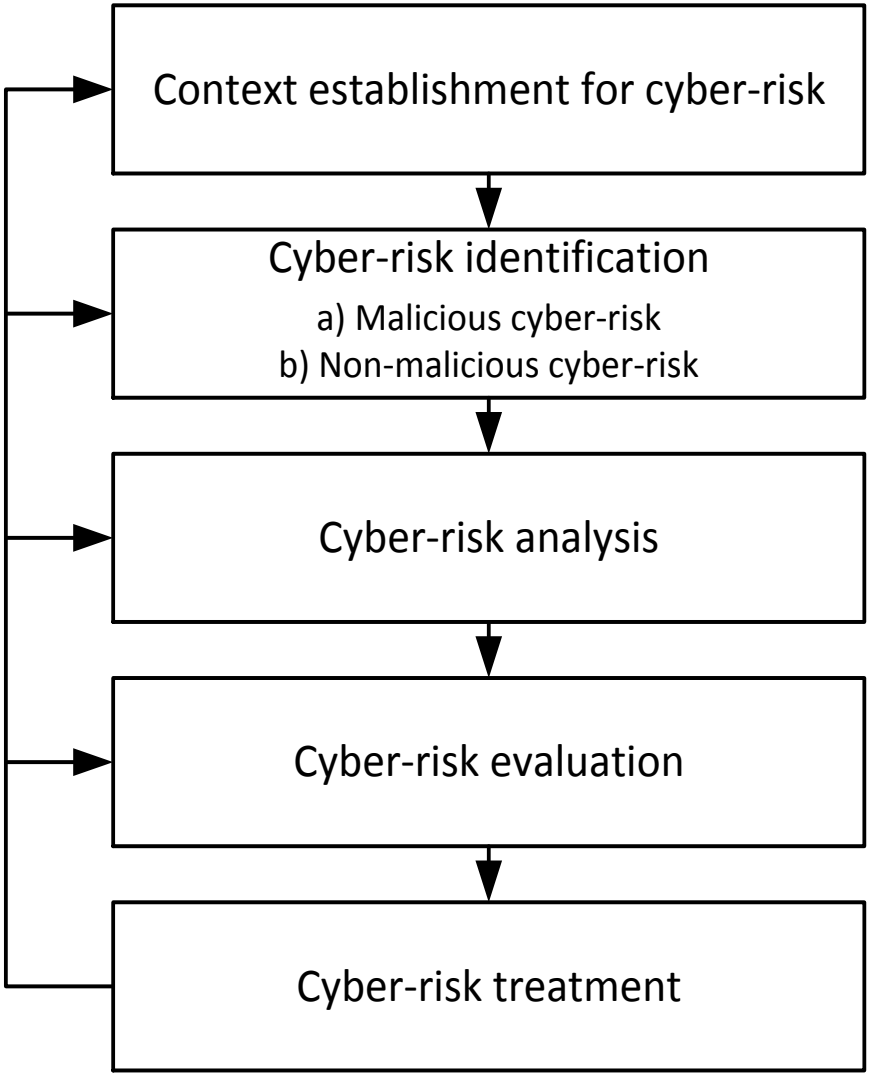
- There is overlap between the two, but:
  - Cybersecurity goes beyond CIP
  - CIP goes beyond cybersecurity

Cybersecurity vs. Information Security and CIP



Cyber-Risk Assessment

- A **cyber-risk** is a risk that is caused by a cyber-threat
- We distinguish between
  - Malicious cyber-risk
  - Non-malicious cyber-risk



Identification of Malicious Cyber-Risk

- Malicious cyber-risks are caused by adversaries with intent
- We need to understand
  - Who or what is the threat source (attacker)?
  - What is the motive and intention?
  - What resources are required?
  - Which skills are required?
  - Which vulnerabilities can be exploited?
  - …
- There are many helpful sources of information
  - Logs, monitored data, security testing, …
  - OWASP, CAPEC, CWE, annual security reports, standards, …

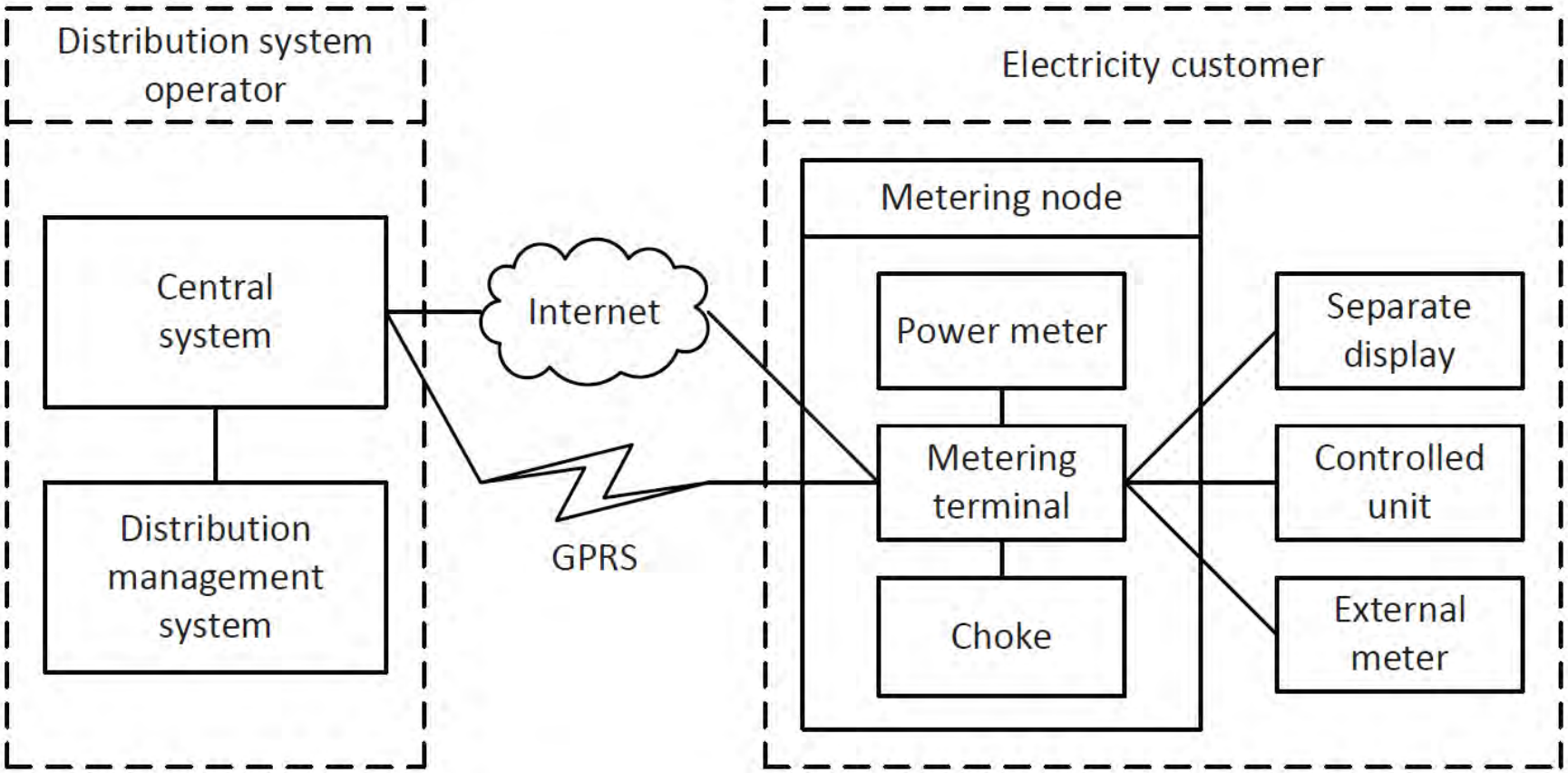
Identification of Non-Malicious Cyber-Risk

- Normally, there is no intent behind non-malicious risks

- To avoid getting overwhelmed during the risk identification, we recommend to start with the assets to identify incidents
- Aspect to take into account:
  - How are assets stored and represented, and how are they related to the target?
    - E.g., how is information stored and processed in the system and in cyberspace, which users and applications have access to read and modify, how is the information transmitted,…?
  - Use logs and monitored data, investigate technical parts of the system, as well as cultures, routines, awareness, etc. of the organization and personnel
  - Take into account unintended external threats
  - Use relevant sources such as ISO 27005 and NIST guide for conducting risk assessments

Example and Demo

Advanced Metering Infrastructure (AMI) of a Smart Grid



CORAS Risk Modeling

- CORAS is a model-driven approach to risk assessment based on ISO 31000
  - Method
  - Language
  - Tool
- The CORAS language is a graphical language for risk identification and modeling
  - Formal syntax: The grammar is precisely defined and implemented in the tool
  - Formal semantics: Mathematical interpretation that enable rigorous analysis
  - Natural language semantics: Any diagram can be systematically translated to paragraphs in English prose
  - Comes with a calculus with rules for calculation, reasoning and consistency checking

CORAS Diagram Elements

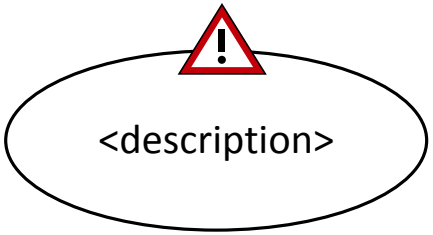
Party

Asset

Malicious threat source

Non-malicious threat source

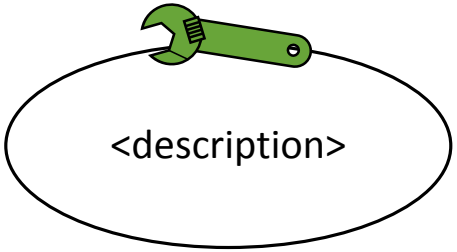
Non-human threat source



Threat

Vulnerability

Incident



Treatment

Risk

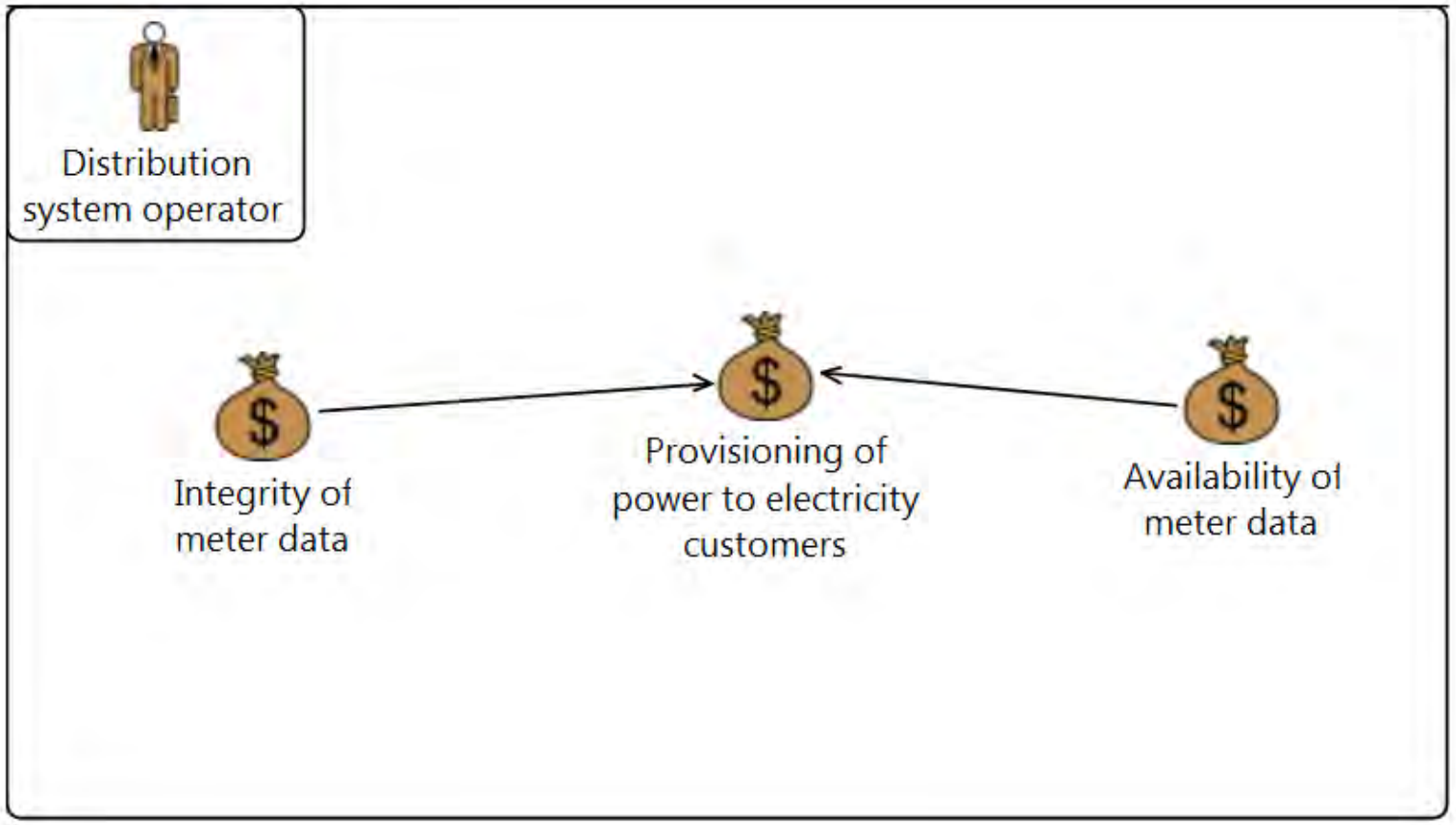
CORAS Diagrams

- The CORAS language supports all steps of the risk assessment process
- Different kinds of diagrams support different steps
  - **Asset diagrams** for identifying and documenting assets during context establishment
  - **Threat diagrams** for risk identification and risk analysis
  - **Risk diagrams** for risk evaluation
  - **Treatment diagrams** for treatment identification
  - **Treatment overview** diagrams for documenting treatments

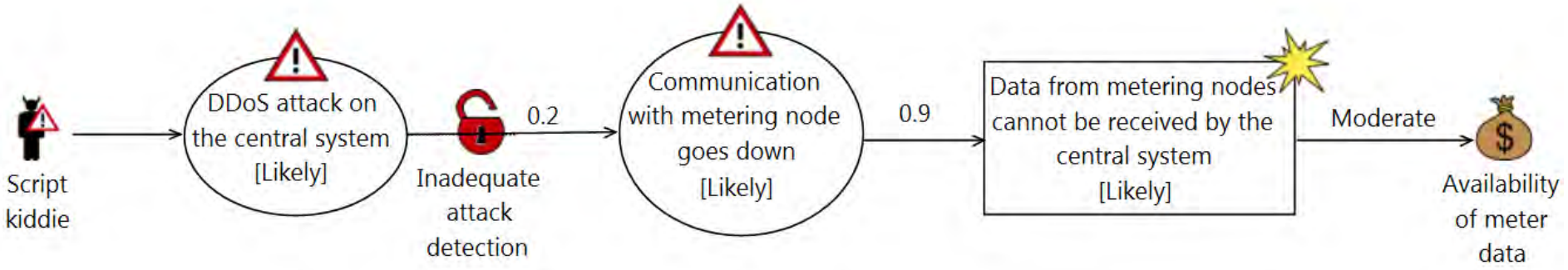
AMI Example: Party and Assets

- The party for the analysis is the distribution system operator
- Assets:
  - Integrity of meter data
    - The integrity of meter data should be protected all the way from **Power meter** to **Distribution system operator**
  - Availability of meter data
    - Meter data from **Metering node** should be available for **Distribution system operator** at all times
  - Provisioning of power to electricity customers
    - Power should only be switched off or choked as a result of legitimate control signals from **Central system**

CORAS Asset Diagram



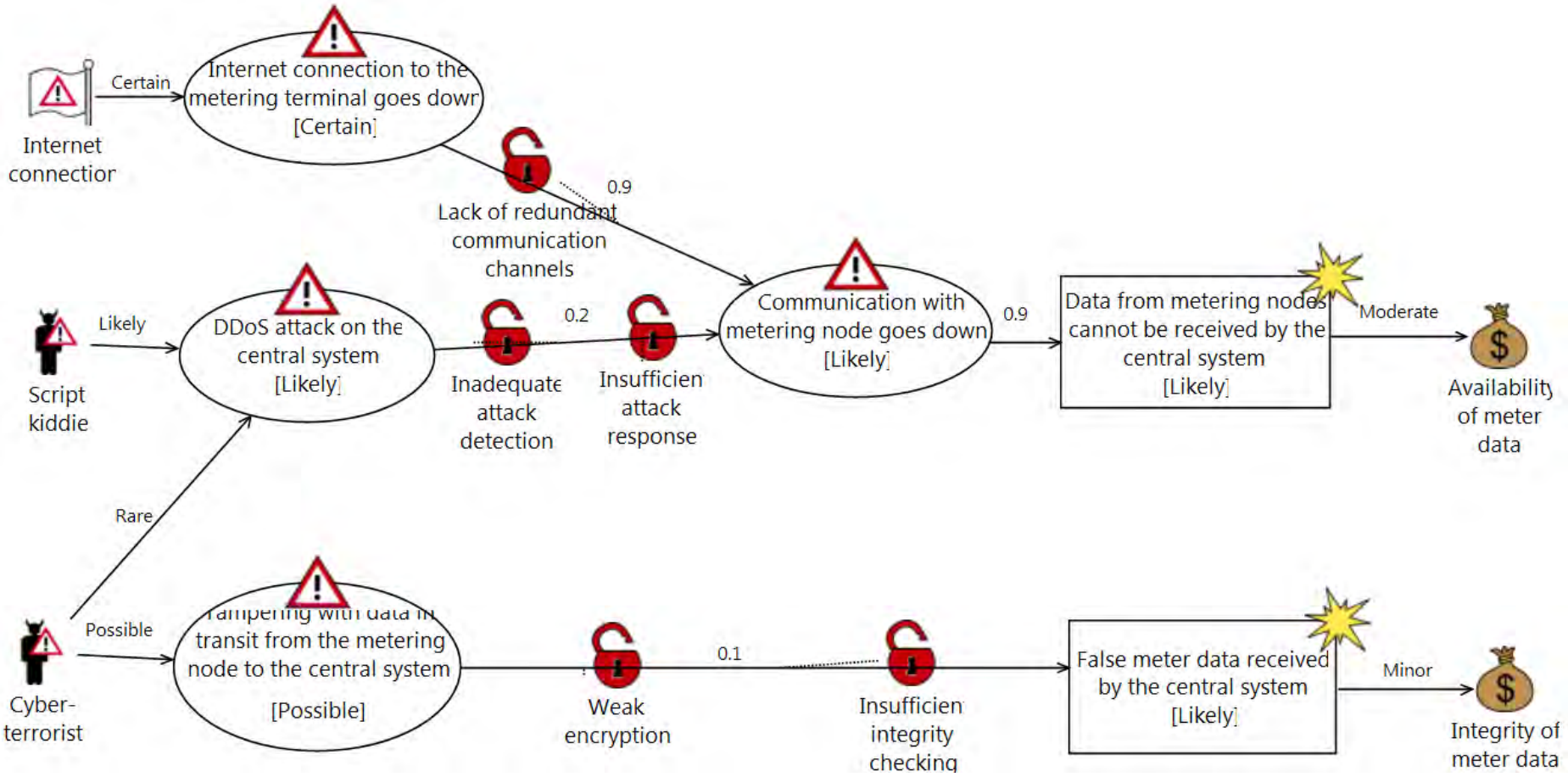
CORAS Threat Diagram



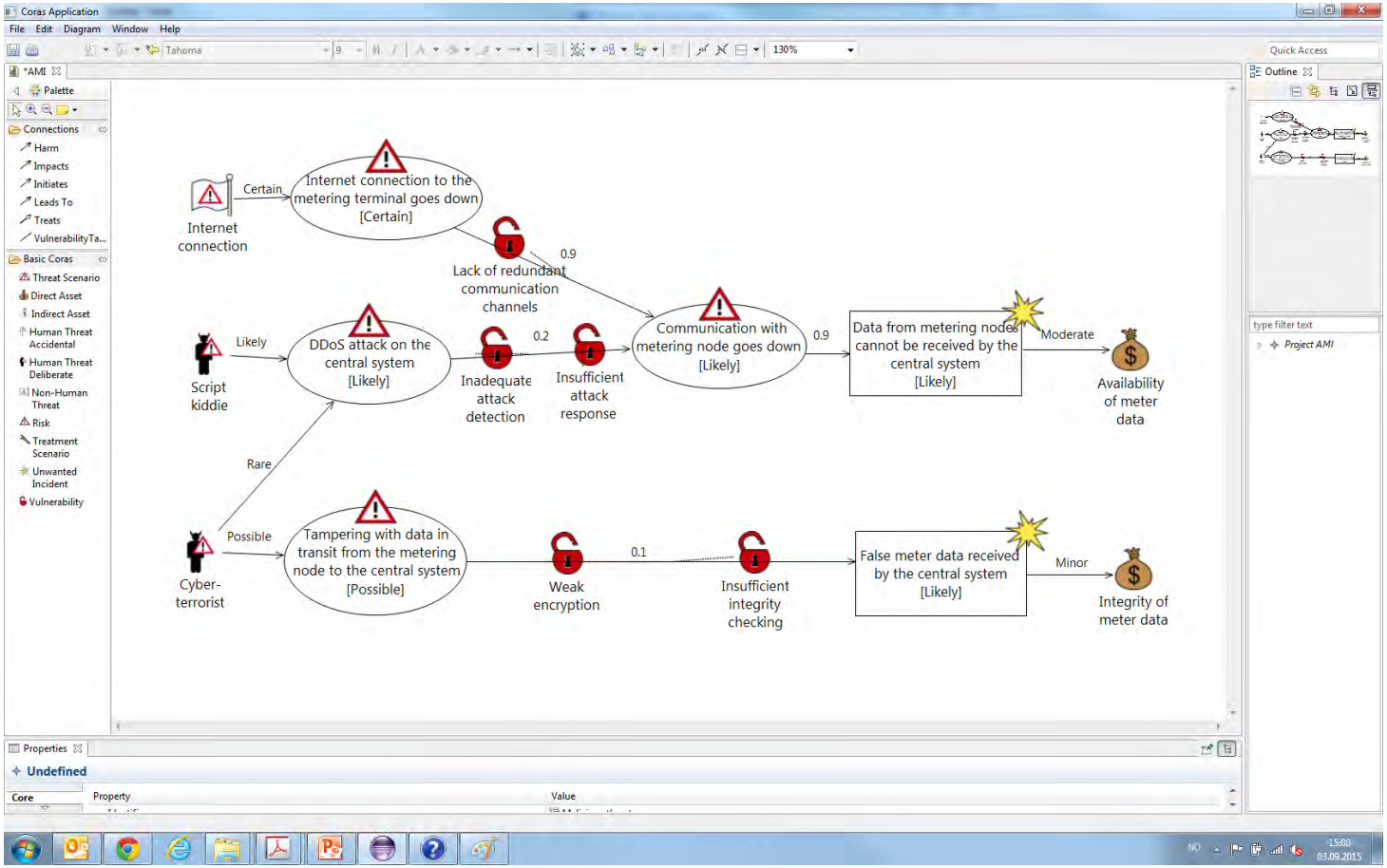
Likelihood Scale

Likelihood	Description	Frequency interval
Seldom	Less than 1 time per 10 years	[0, 0.1›:1y
Unlikely	1-10 times per 10 years	[0.1, 1›:1y
Possible	2-12 times per year	[1, 13›:1y
Likely	13-60 times per year	[13, 60›:1y
Certain	More than 60 times per year	[60, ∞›:1y

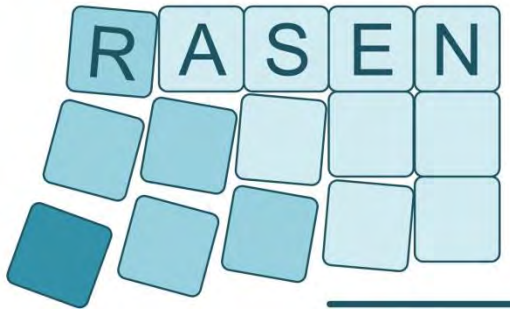
CORAS Threat Diagram



Live Demo


Coras Application
File Edit Diagram Window Help
Palette
Connections
Harm
Impacts
Initiates
Leads To
Treats
Basic Coras
Threat Scenario
Direct Asset
Indirect Asset
Human Threat Accidental
Human Threat Deliberate
Non-Human Threat
Risk
Treatment Scenario
Unwanted Incident
Vulnerability
Outline
Properties
Undefined
Core
Property
Value
Internet connection
Certain
Internet connection to the metering terminal goes down [Certain]
Lack of redundant communication channels
0.9
Likely
Script kiddie
DDoS attack on the central system [Likely]
Inadequate attack detection
0.2
Insufficient attack response
Communication with metering node goes down [Likely]
0.9
Data from metering node cannot be received by the central system [Likely]
Moderate
Availability of meter data
Rare
Possible
Cyber-terrorist
Tampering with data in transit from the metering node to the central system [Possible]
Weak encryption
0.1
Insufficient integrity checking
False meter data received by the central system [Likely]
Minor
Integrity of meter data

Thank You!



Compositional Risk Assessment and Security Testing of Networked Systems

www.rasenproject.eu

SEVENTH FRAMEWORK PROGRAMME

SASS15

RACOMAT – Risk-based Security testing for networked systems

Johannes Viehmann, FraunhoferFOKUS

Abstract:

The iterative RACOMAT process combines risk assessment and automated security testing in both ways: Test-Based Risk Assessment (TBRA), which tries to improve risk assessment with the results of security tests and Risk-Based Security Testing (RBST), which tries to optimize security testing with results of risk assessment. The RACOMAT tool implements the entire RACOMAT process. It supports risk analysts and testers in each step without having trouble with different tools, offering a seamless continuous workflow with a high level of automation.

Fraunhofer FOKUS

SASS '15

Risk Assessment and Security Testing of Large Scale Networked Systems with RACOMAT

Johannes Viehmann 2015

Fraunhofer FOKUS

Overview

Risk Assessment and Security Testing
of Large Scale Networked Systems with RACOMAT

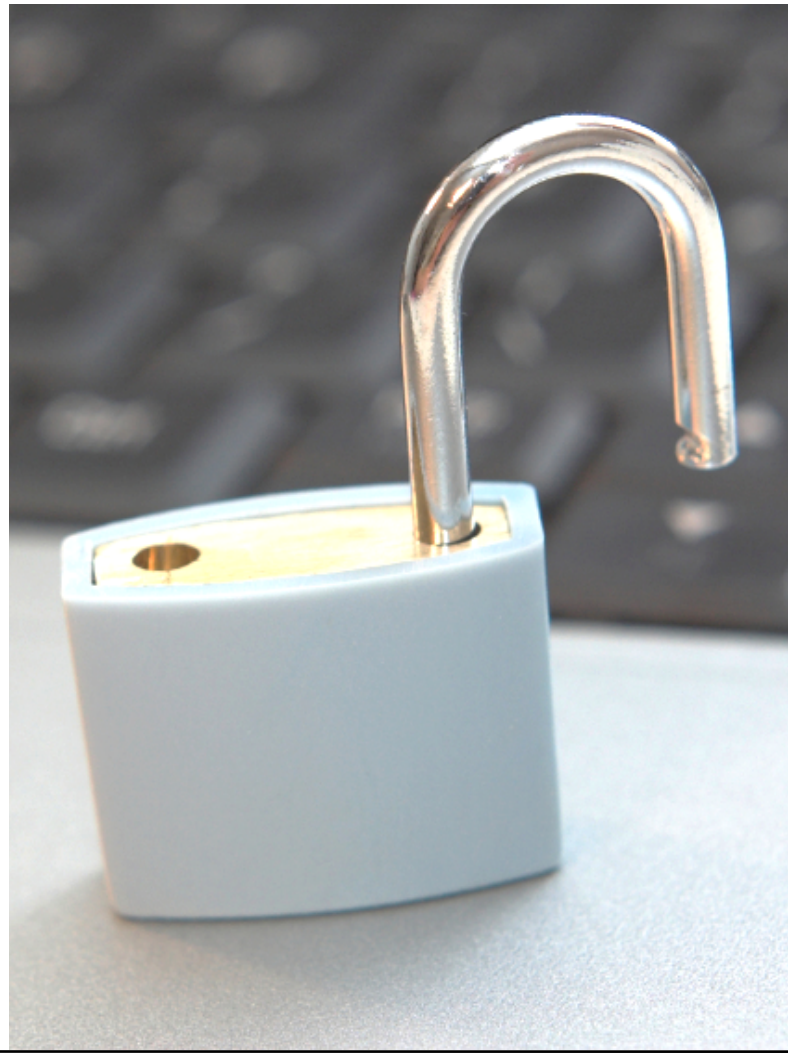


Table of Content

- Introduction
- State of the Art
- Problems and Challenges
- Initial Risk Assessment and Refining the Risk Picture
- Automated Risk-Based Security Testing
- Test-Based Risk Assessment
- High Level Composition
- Conclusion and Future Work

Introduction – Risk Assessment and Security Testing

Definition

- Risk assessment is a part of risk management and means to identify, analyze and evaluate risks
- Security testing is one possibility to analyze risks

Why Risk Management is required

- In the real world, perfect security often cannot be achieved
  - There are residual risks for any complex ICT-System
- Risk assessment and risk treatment can help to create trust by:
  - Communicating residual risks
  - Help to implement safeguards and treatments for to high risks in order to reduce the risks

Introduction – the Case Study

The RASEN Research Project

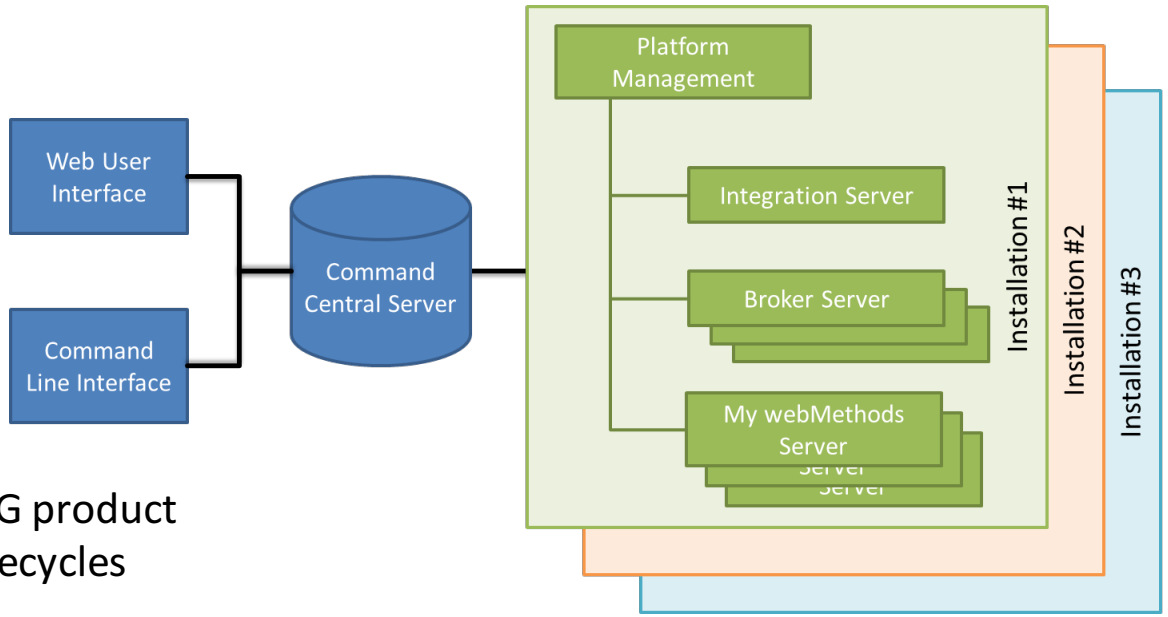
- European project with 7 partners in four countries
- Three industrial case studies

RASEN
Compositional Risk Assessment and Security Testing of Networked Systems

SEVENTH FRAMEWORK PROGRAMME

The Software AG Case Study

- Software under analysis is called Command Central
  - Part of webMethods tool suite by Software AG
  - Uses SAG Common Platform and OSGi framework
  - Intended to manage Software AG product installations throughout their lifecycles



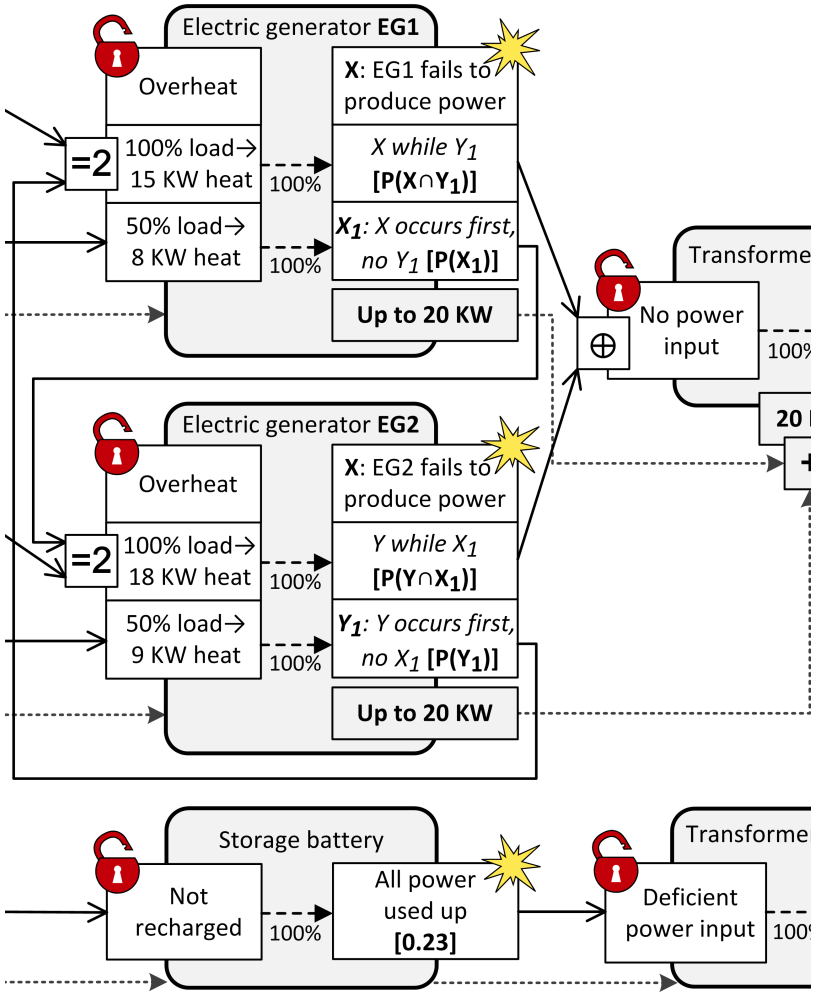
State of the Art – Risk Assessment and Security Testing

There are lots of methods, libraries and tools for

- Risk Assessment
  - Standard: ISO 31000
  - FMEA/FMECA, FTA, ETA, CORAS …
  - Catalogues of common risk artifacts
    - CWE, CAPEC (Mitre), BSI IT-Grundschutz
- Testing and security testing
  - Standard: ISO 29119
  - Automated testing, fuzz testing …

There is less literature and support for the combination of Risk Assessment and Security testing

- Test-Based Risk Assessment (TBRA)
- Risk-Based Security Testing (RBST)
- Combination of TBRA and RBST



Problems and Challenges

Risk assessment might be difficult and expensive

- Hard for large scale systems
- Is highly dependent on the skills and estimates of analysts

→ We have to find ways to make risk assessment more objective

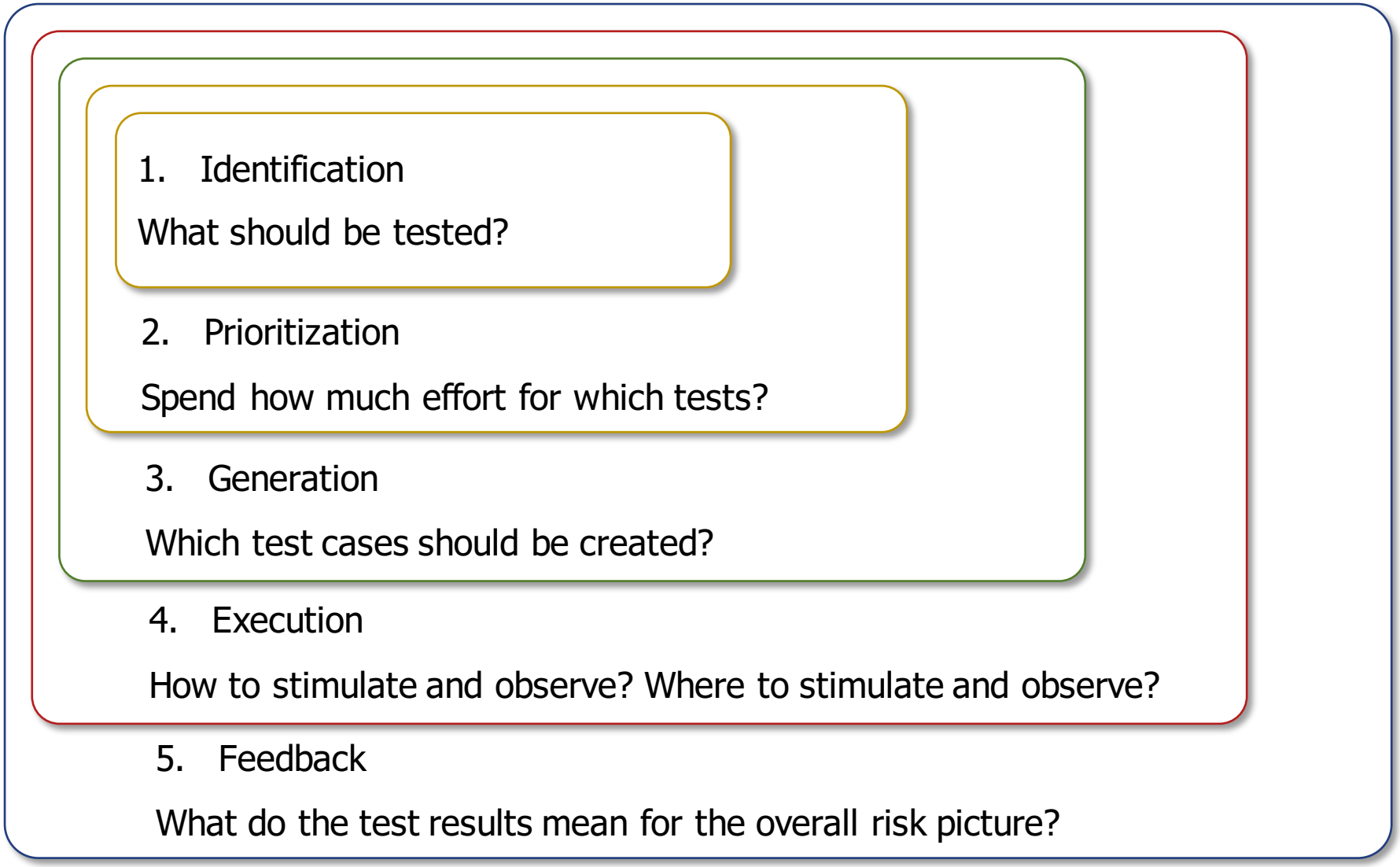
- ➢ e.g. with security testing

Security testing might be difficult and expensive, too

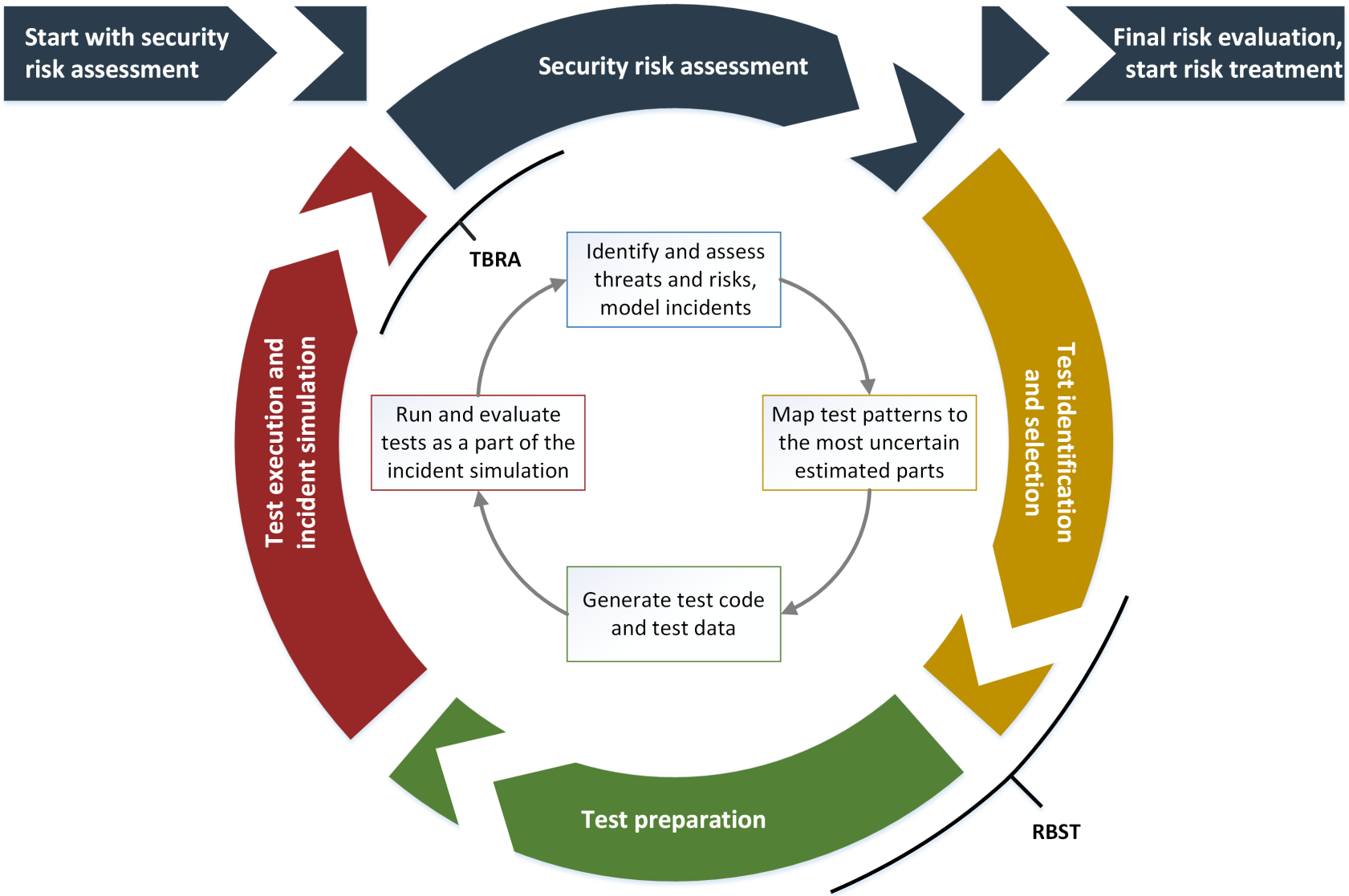
- Testing for unwanted behavior – there is no specification what to expect
- Even highly insecure system can produce lots of correct test verdicts if the "wrong" test cases have been created and executed
- Manual testing is error prone and infeasible for large scale systems

→ Automate security testing using risk assessment?

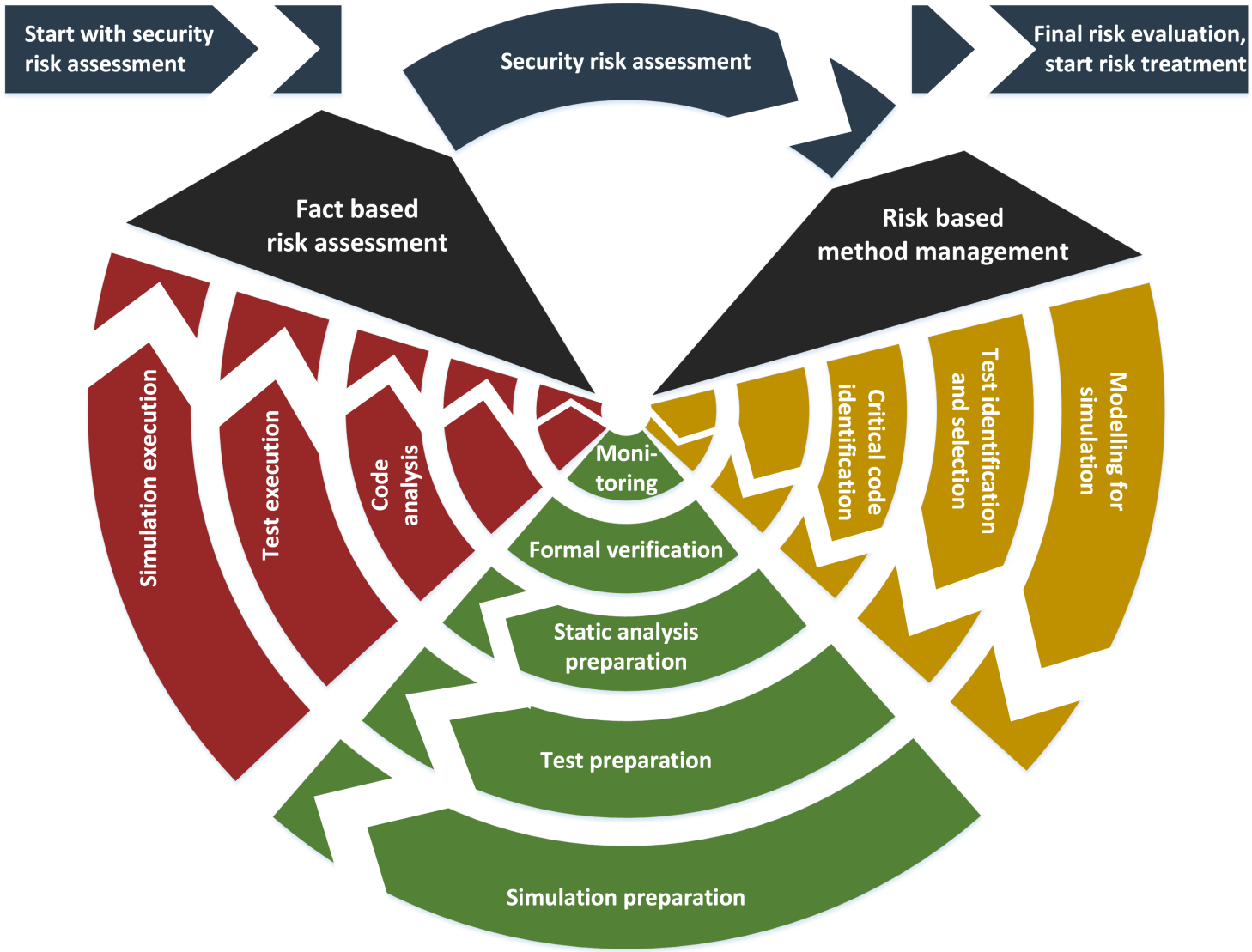
Problems and Challenges – Combined Risk Assessment and Testing Process



Problems and Challenges – Iterative Risk Assessment Process



Problems and Challenges – Iterative Risk Assessment Process



Problems and Challenges – The Case Study

Software AG wishes:

- Get a realistic picture of the overall risks associated with Command Central and the other Software AG products
  - Command Central is used in many different contexts for managing various systems
  - There should not be an expensive complete risks assessment required for each scenario

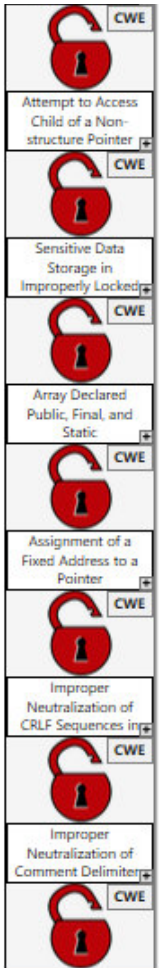
- Manual analysis methods are generally regarded to be not feasible
  - Software AG products are complex
  - There is only a limited budget

➢ As much automation and reusability as possible!

Initial Risk Assessment

Manual high level analysis

- Establish the context
- Identify risks
  - Joint workshop of the Command Central product development team and security experts
  - Product under investigation and potential vulnerabilities modelled in ARIS tool
  - Used the existing Mitre CWE database
- Results:
  - Long lists of weaknesses for about 30 components
    - Not analyzed if the weaknesses actually exist
    - Not investigated how likely it is that the existing ones would actually be exploited or what the consequences might be



Refining the Initial Risk Picture

The initial risk identification contains not enough information to enable automated testing:

- Requires a low level risk assessment
  - Connection between risk analysis artefacts and system components
    - Where to stimulate?
    - Where to observe?
  - ➢ Create a model that has both system information and risk information
  - ➢ Lots of manual work to create such a model?

We decided to develop a tool for this step and the entire combined TBRA and RBST process in order to keep the manual effort as low as possible:

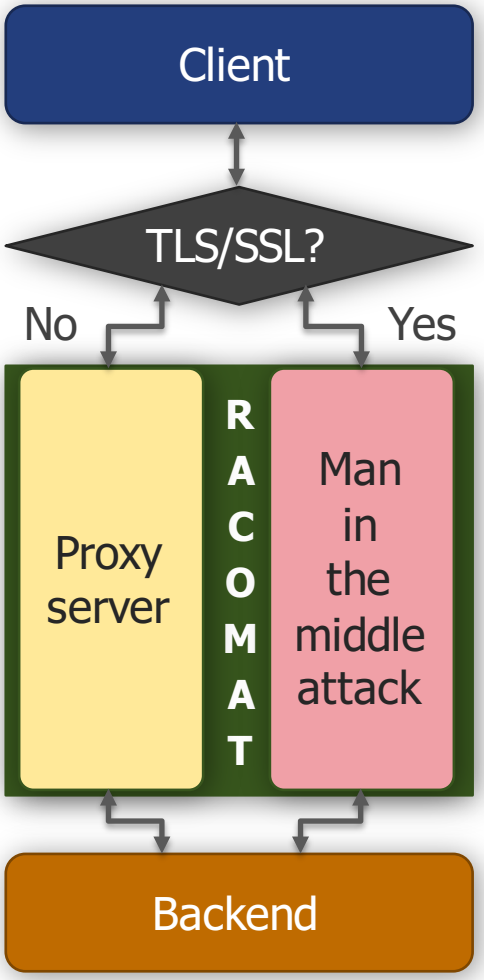
- The RACOMAT tool
  - Stand alone application
  - Also Visual Studio plug-in

Refining the Initial Risk Picture with RACOMAT

Generate system models with low level risk information

- Automated static analysis of components
  - Generate models for testable interfaces
    - HTML pages, source code, compiled libraries or programs …
  - Threat interfaces with input and output ports
    - Suggests typically related risk artefacts (e.g. vulnerabilities) for the identified interfaces
- For Command Central static analysis fails
  - Web interface has lots of scripts – hard to parse
  - The user interfaces are generated dynamically based on the session state

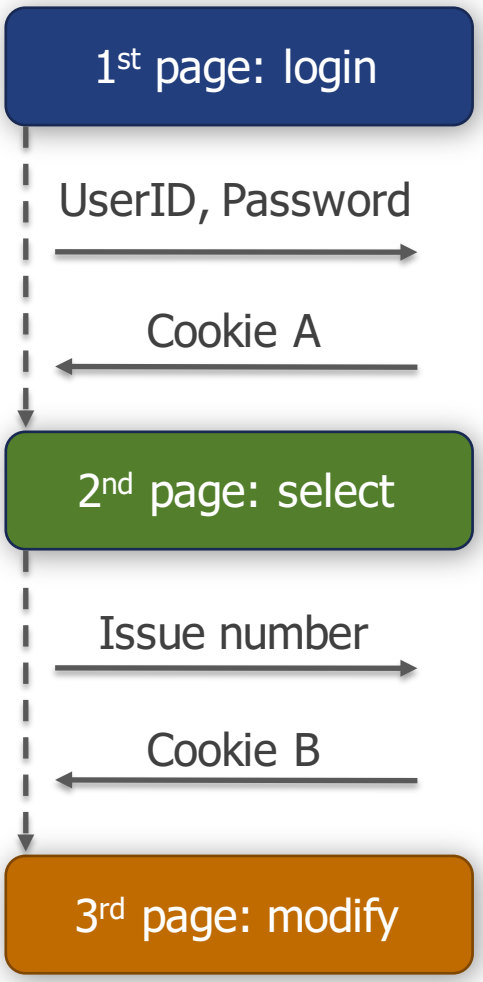
➢ Dynamical interface analysis required
  - Observe network traffic while the system is used and generate thread interface models
  - Semi automated



Refining the Initial Risk Picture with RACOMAT

What RACOMAT dynamic analysis does

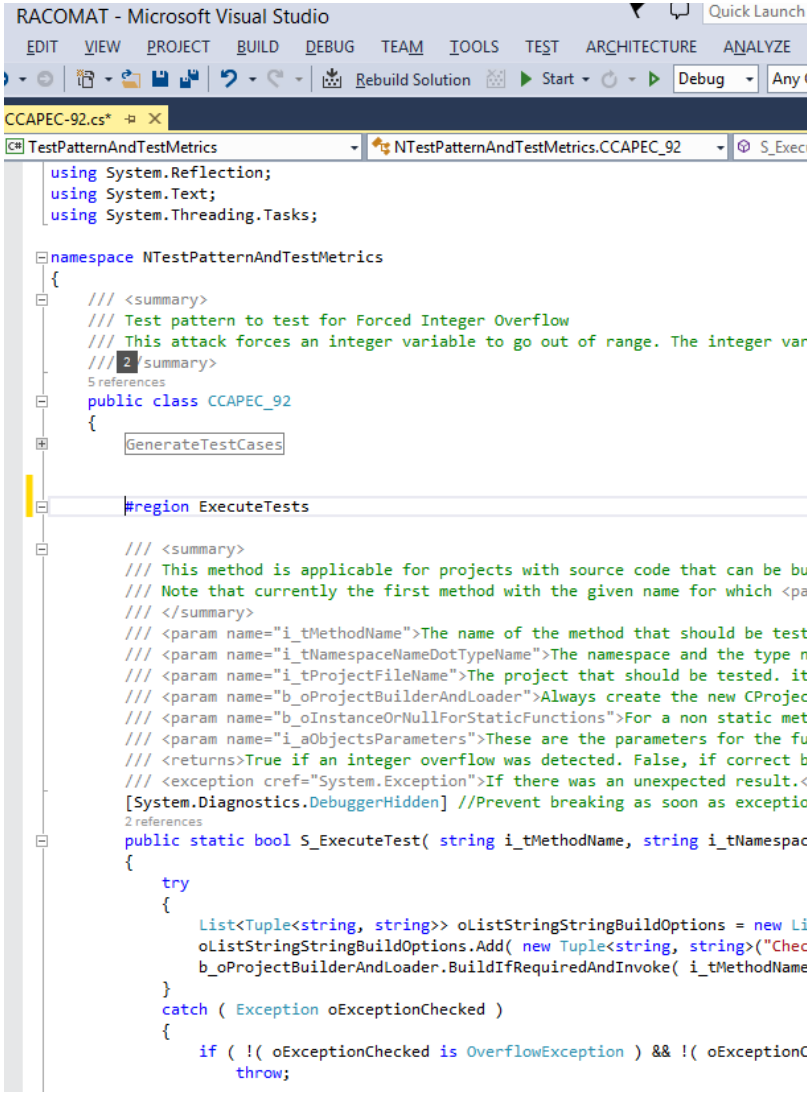
- Analyzes data exchange
  - Authentication
  - Cookies
  - Parameters (Url, multipart, JSON, SOAP …)
- Generates state dependent threat interface models
  - Input / output ports
    - Type information, Values
  - Suggests lists of typically related weaknesses for each port
    - From CWE database and type information
    - From initial risk assessment
- Models relations between threat interfaces
  - How to get to a certain state
    - e.g. authenticate, set cookie



Automated Risk-Based Security Testing

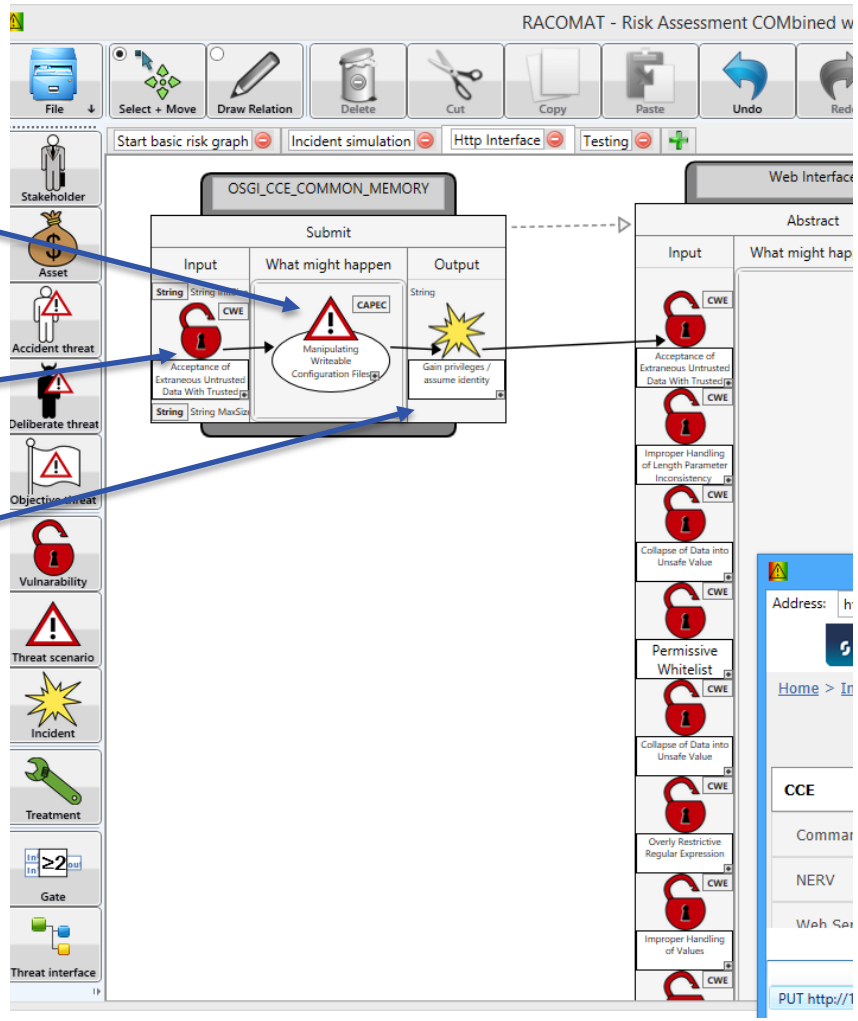
Basic ideas

- Security testing means attacking the SUT
- ➢ Attack patterns describe exactly, how such an attack could be made
- ➢ CWE weaknesses contain links to typically related CAPEC attack patterns
  - ➢ Add CAPEC attack patterns to the system and risk model
- Problem: Attack patterns are designed for human beings
  - – Implementing them requires a lot of manual work
- ➢ Introduce reusable Security Test Patterns
  - ➢ Machine interpretable, executable
  - ➢ Attached to CAPEC attack patterns for minimal instantiation effort



Automated Risk-Based Security Testing with RACOMAT

- RACOMAT uses the combined system and risk model to instantiate test patterns
  - Attack patterns indicate which test patterns should be used
    - Priority of tests can be calculated based on likelihood and consequence values
  - Vulnerabilities indicate where to stimulate the SUT
  - Unwanted Incidents can be introduced in order to determine what should be observed to get some verdict
  - ➢ Complete automation often achievable
- Problem: There are only a few test pattern available
  - Implementing generic reusable test pattern is challenging
  - Currently not really saving manual effort
  - ➢ Vision: create an open security test pattern library



Test-Based Risk Assessment

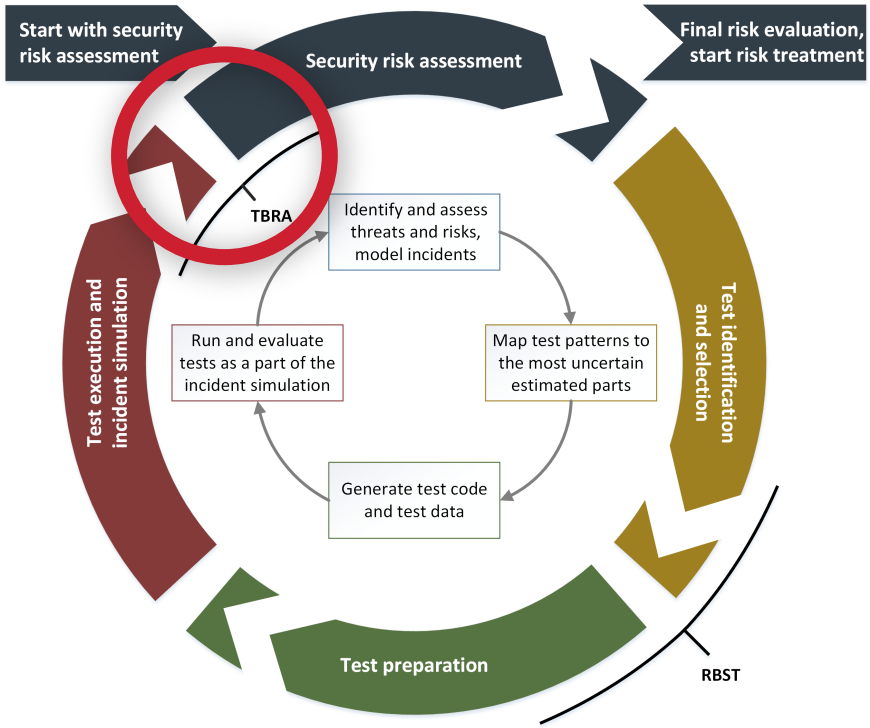
There are basically two types of test based updates to the risk model

- Introduce new unwanted incidents and vulnerabilities discovered while testing

- Update likelihood values based on test results
  - ➢ Use security testing metrics

➢ RACOMAT supports both in a semi-automated fashion

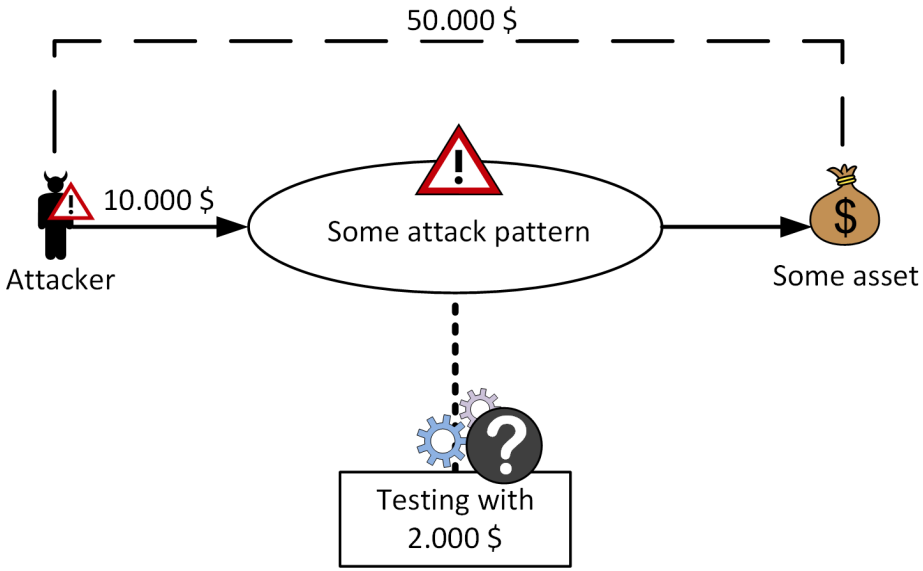
  - Problem: How to deal with test results that did not find any unwanted incidents?
  - Problem: There are only a few good security testing metrics available at the moment



Test-Based Risk Assessment with Testing Metrics

Example for an efficiency metric:

- Idea: Try to figure out *P* indicating how likely it is that an attacker will apply the tested attack pattern successfully
  - In future simulations, that likelihood *P* will be used instead of testing the component again
- Input:
  - *R*: testing results: number of times unwanted incident was triggered
  - *T*: how much budget was spend for testing
  - *A*: estimated budget of deliberate human threats for such an attack
- A metric could define a function to calculate a probability value like that the attack will occur, e.g.:
  - $P = \left(1 - \frac{1}{(\sqrt{2})^{A*(1+R)/T}}\right)$



R \ T	2000	4000	10000	20000
0	0,82	0,58	0,29	0,16
1	0,97	0,82	0,50	0,29
2	0,99	0,92	0,65	0,41

High Level Composition

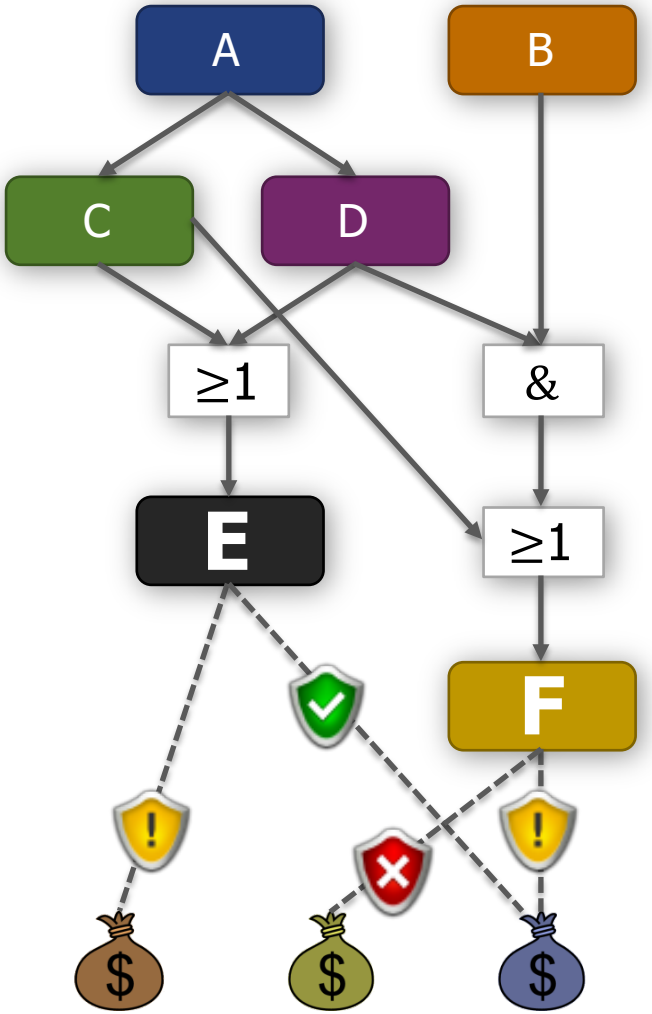
Refining the risk picture and testing produce detailed risk models

- Required to get more objective picture, but too much information
- For risk management, typically more high level results are wanted
- The same components and systems may be used in different scenarios and contexts

➢ Aggregate risk analysis results
  ➢ RACOMAT uses simulations to calculate high level risk values

➢ Model the different contexts
  ➢ Use CVE vulnerabilities database for common software components

➢ Do compositional risk assessment
  ➢ Requires manual modelling?



High Level Composition with Tags

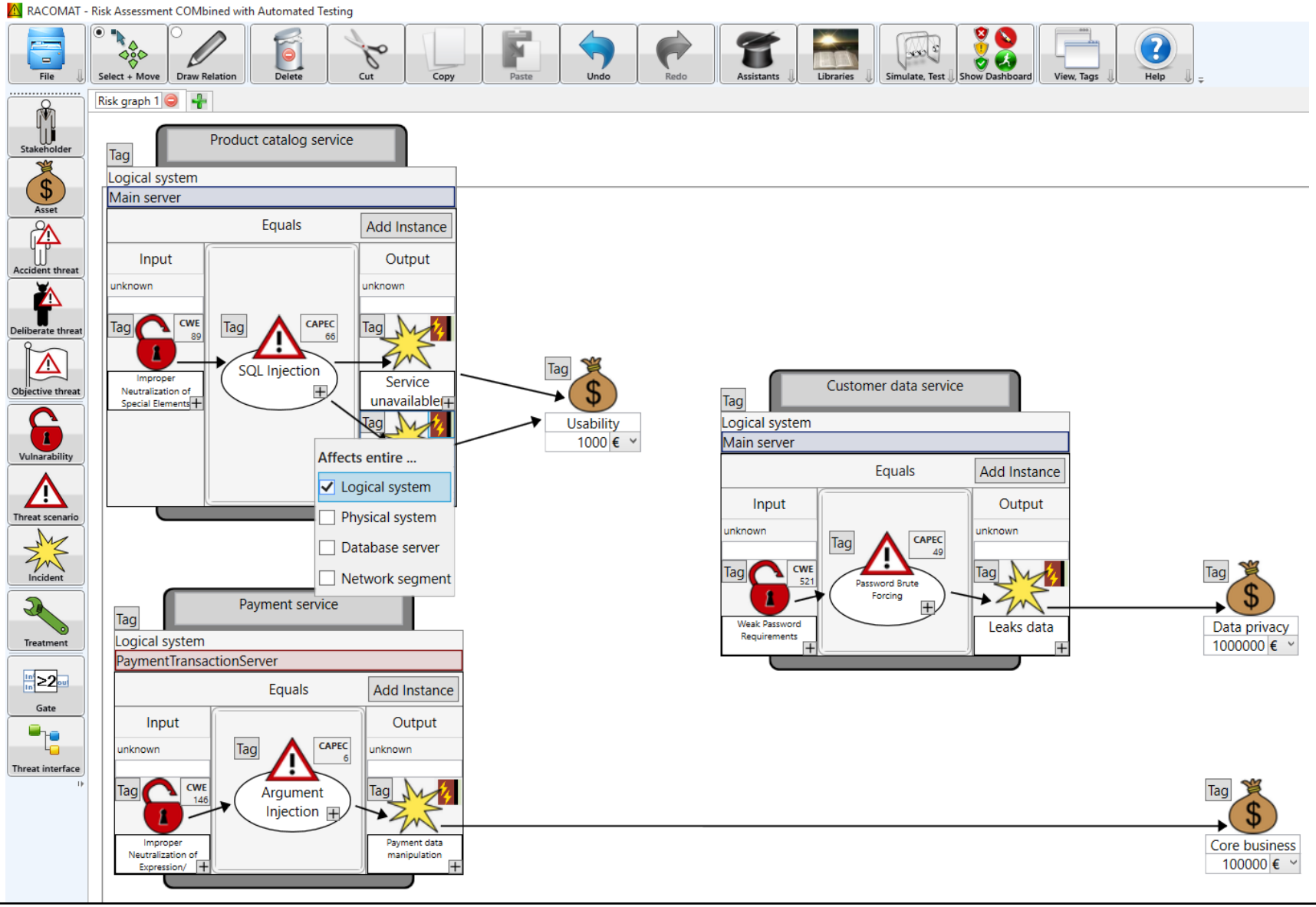
For large scale systems, graphical modelling might become unintuitive

- Analysts will probably get lost simply because the models get to complex

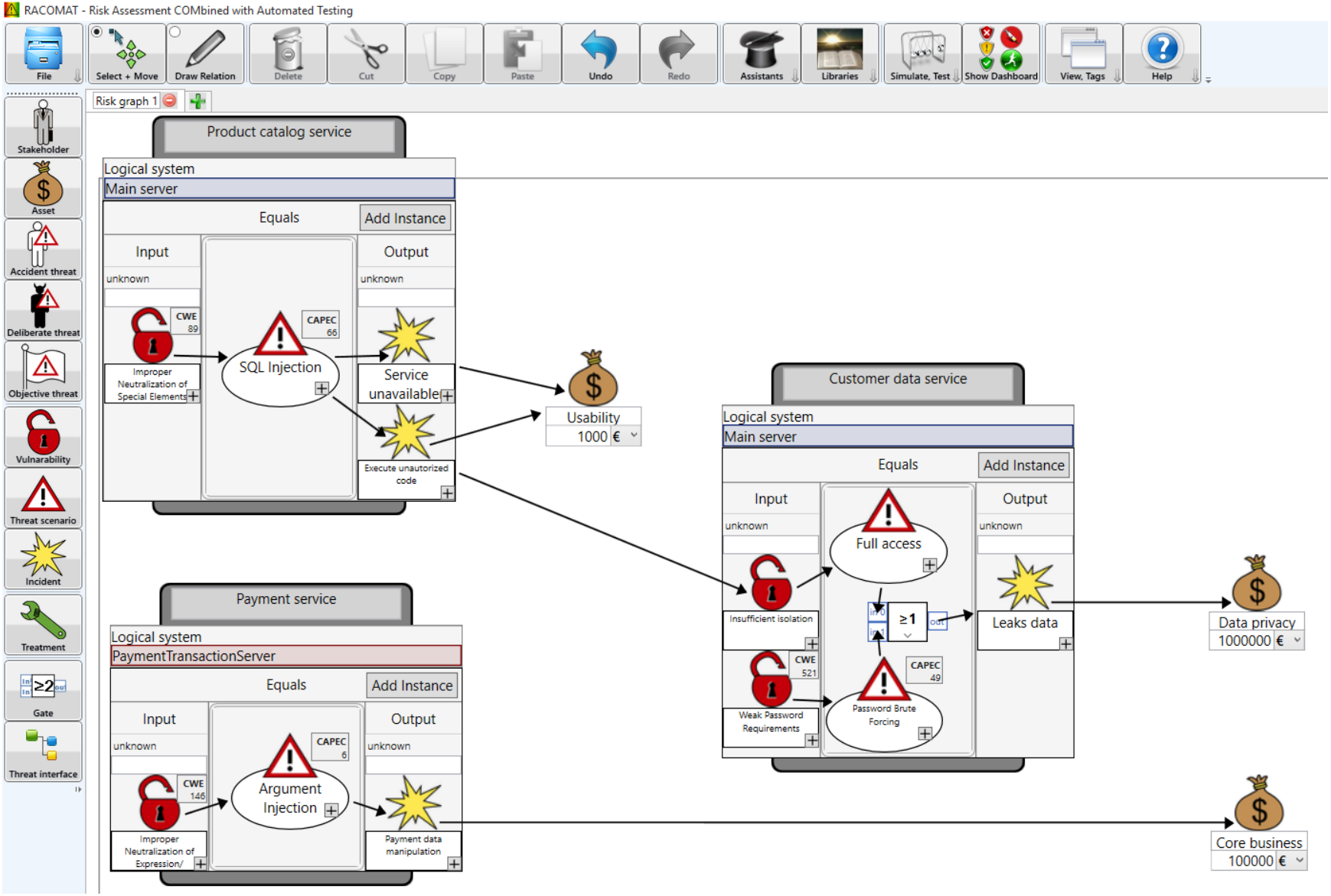
Idea: Model isolation and scope with tags

- Isolation tags with categories and values to model involved entities
  - Component, Product
  - Configuration
  - Physical system, Logical system, Network segment
  - Database, Database server
  - Operating system, Programming language, Framework
  - Third party API / library
- Scope tags indicate which entities are eventually affected by incidents / faults

High Level Composition with Tags



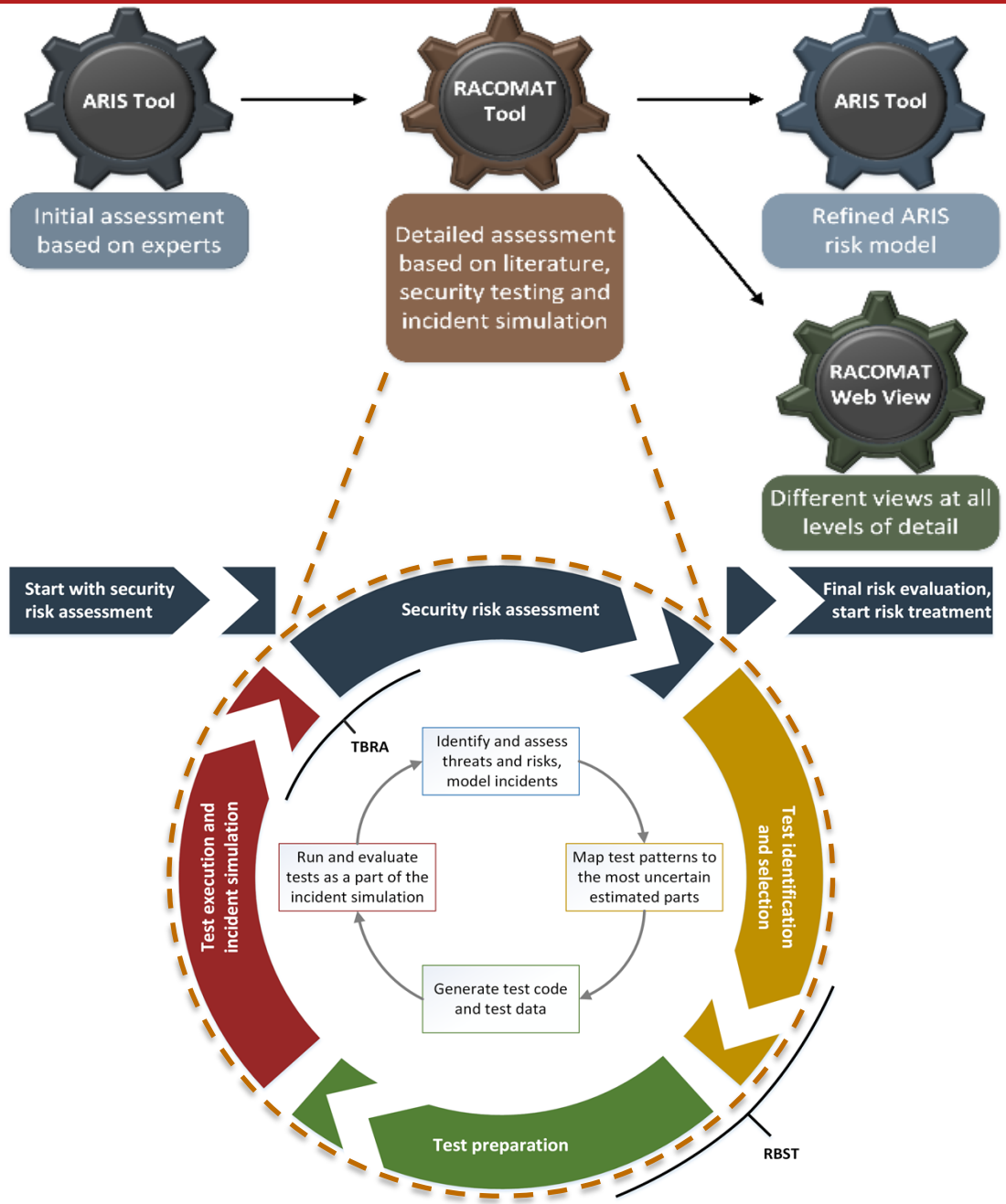
High Level Composition with Tags



Case Study Workflow

Final results are exported from the RACOMAT tool in two formats:

- ARIS exchange format (JSON) at the same level of detail which the initial risk assessment provided
- XHTML format at different level of details
  - Risk graphs
  - Test results
  - Dashboards (basically intended to support management decisions)

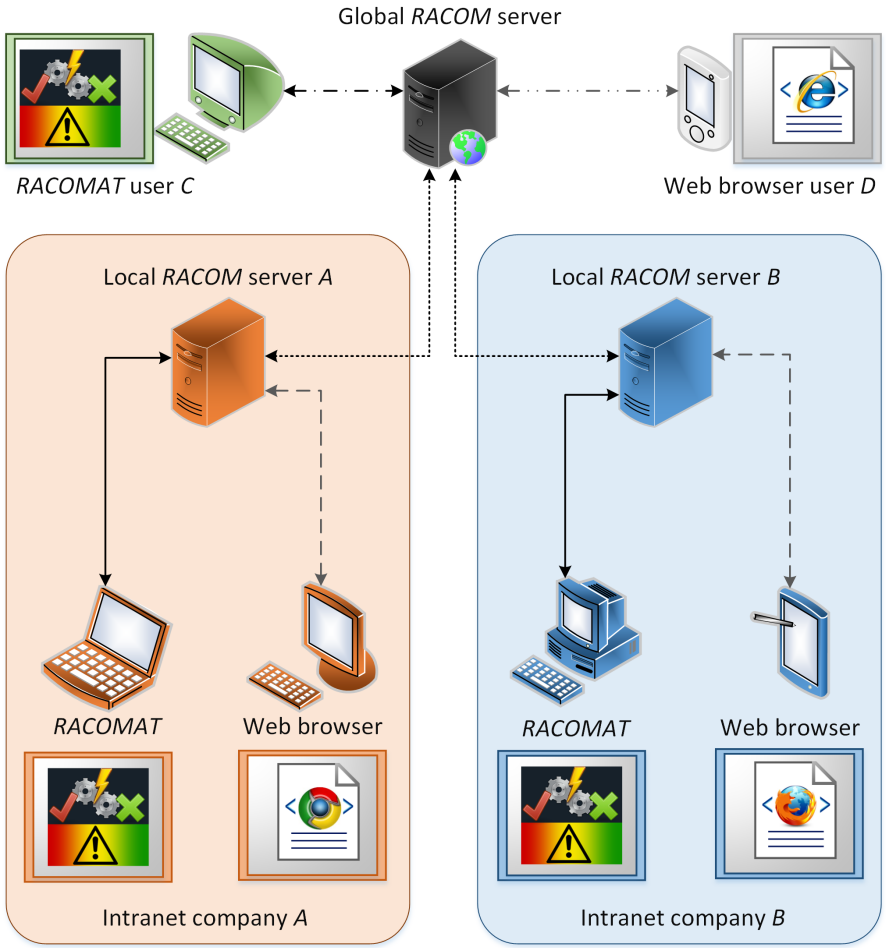


Conclusion and Future Work

Observations

- Combined risk and system models are a good base for automated security testing
  - Creating such models does not require much manual work
  - Automation highly depends on good reusable existing artefacts
  - Problem: No adequate databases of test pattern and testing metrics available

- Future work
  - Complete the Software AG case study within the next five months
  - Development of RACOM server
    - Sharing test patterns, testing metrics
    - Sharing reusable threat interfaces for entire components or programs



Questions, Remarks?

Thanks a lot for the attention!

Johannes Viehmann 2015

Contact

Fraunhofer Institute for Open Communication Systems FOKUS

Kaiserin-Augusta-Allee 31
10589 Berlin, Germany

www.fokus.fraunhofer.de

Johannes Viehmann
Researcher
johannes.viehmann@fokus.fraunhofer.de

System Quality Center SQC

http://s.fhg.de/sqc

Dr. Tom Ritter
Head of competence center SQC
tom.ritter@fokus.fraunhofer.de

Friedrich Schön
Head of competence center SQC
friedrich.schoen@fokus.fraunhofer.de

SASS15

Risk Management in the Development Process

Armin Lunkeit, OpenLimit

Abstract:

With Industry 4.0 and Internet of Things embedded systems continue to gain importance. Hardware costs decline and the need for new intelligent devices increases. Pressure to innovate and high speed of development dominate in engineering. New systems provide complex functions such as secure communication via non-secure networks. The presented report outlines an approach for the management of existing development risks for products provided with IT security functionality within tight time and budget targets.

Vita:

Armin Lunkeit was born in 1978 and is a German national. As Chief Technology Officer, he has been a member of the board of management of the OpenLimit Group since December 2007. He studied microsystems technology at the Technical College of Technology and Economics in Berlin, from where he graduated in 2002 as a chartered engineer. Mr. Lunkeit entered the field of software development in 2000. After concluding his studies, he worked as a developer for Kithara GmbH. Mr. Lunkeit worked at OpenLimit SignCubes GmbH in product development from June 2003 until he took over his present position.

Risk Management in the Development Process
A Progress Report

Armin Lunkeit

Introduction

- Industry 4.0 and IoT gain importance of Embedded Systems
- Hardware Costs decline
- Pressure to innovate and high speed of development dominate
- Part of the Game: Smart Meters and Smart Meter Gateways with distinct communication over (insecure) networks

Introduction

Term	Definition
Risk	The likelihood of an unwanted incident and its consequence for a specific asset. (see CORAS)
Risk Management	Coordinated activities to direct and control an organisation with regard to risks. (see CORAS)

Introduction

Examples of Development Risks

- *People Risks* - availability, skill level, experience
- *Size Risks* - handling of large teams, increased complexity in large products
- *Process Risks* - well defined development process
- *Technology / Tool Risks* - new or complex technology increases the risk, availability of reliable tool chains (development environmen, CASE tools)
- *Organisational Risks* - financial stability, organisational threats, change in company focus
- *Estimation Risks / Planning Risks* - resource estimates and product development time
- *Customer Risks* - changes to the customer requirements

Smart Meter Gateway - Overview

- records energy consumption data
- transmits energy consumption data to meter operators
- utilizes a trusted communication channel with the administrator
- part of the critcial infrastructure
- needs to be security evaluated (Common Criteria, ISO 15408) on evaluation assurance level 4+
- 6.000.000 SMGW installations expected until end of 2020

Smart Meter Gateway - Functional Requirements

- Storage of meter data and application of tariffing profiles
- Remote administration channel
- Support of different meter types
- user interface for displaying consumption data
- multi-client capability
- Support of cryptographic algorithms

Smart Meter Gateway - Security Requirements

- Secure Storage of private key material ⇒ HSM chip
  - Pre-Personalization during production
  - Final Personalization during installation
- Secure communication channels ⇒ TLS and symmetric cryptographic protocols
  - TLS to communication partners in WAN, HAN and LMN
  - AES secured crypto in LMN (wireless communications)
- Passive tampering and modification detection
  - Tampering and modification of hard- and software needs to be detected (or be detectable)
- Functional correctness pertaining Technical Guideline TR 03109
  - Manufacturer must demonstrate complete support of required functionality

Smart Meter Gateway - Cost assumption

Cost-Benefit Analysis was published by Ernst & Young in 2013 (Kosten-Nutzen-Analyse für einen flächendeckenden Einsatz intelligenter Zähler), on behalf of Federal Ministry for Economic Affairs and Energy

cost factor	**value**	**amortisation period**
Ferraris meter	25 EUR	16 years
Intelligent meter	80 EUR	13 years
BSI conformant meter	55 EUR	13 years
SMGW with HSM, without communication module	80 EUR	13 years
SMGW with HSM comm. module and meter	175 EUR	13 years
Installation costs per meter	30 - 100 EUR	13 years
Installation costs per Gateway	20 - 90 EUR	13 years
Installation costs per meter / gw. in comb.	40 - 110 EUR	13 years

Smart Meter Gateway - Development Risks

What kind of risks arise?

In general most of the risks listed on slide 5

- *cost limits:* (organisational risks, estimation risks, customer risks)
- *unfinished specification* (organisational risks, estimation risks, people risks)
- *unfinished security requirements* (organisational risks, estimation risks, people risks, tool risks, technology risks)

Development was based on unstable external foundations. How to handle this situation?

Chosen strategy

- Functional and security requirements define the lower bound of a possible development budget
- During analysis, the security requirements were focussed due to their strong influence on development time and budget
- Setup a team of experienced sw-engineers and:
  - perform use-case analysis
  - technology studies
  - set up management framework (requirement lists, define sw-development process)
  - get an idea of the required tool chain

Following slides show some aspects of the security analysis.

Metrics Definition

Tabelle: Defined Metrics

Metric	**Explanation**
Fulfilment of the security requirements formulated	Consideration of whether or not a suggested measure / a combination of several measures addresses a given IT security target
Development and production costs	Costs for the implementation of a suggested measure. Important indicator for staying within the planned development and production budget.
Knowledge and knowledge management	Consideration of whether or not the existing engineering knowledge required for implementation of the identified IT security measure is available.
Availability	Availability of ready partial solutions, use of OTS (off-the-shelf) components

Requirements Analysis and System Design

subject	**questions**
Environmentand Assumptions	■ communication interfaces? ■ communication protocols? ■ Are there any trusted external entities? ■ Public or secure environment?
Security Targets	Should data be stored or exchanged and what are the relevant security requirements?
Adversary Model	■ What might an attacker be capable of? ■ Which interfaces of the system need to be considered in terms of defining the security requirements?
Security Requirements	■ Based on assumptions

Threat Modeling

- set up a model of all interfaces and communication flows
- model closed gaps in the unfinished specification (Techncial Guideline)
- Identification of threats, risks and potential weaknesses
- application of S.T.R.I.D.E classification and D.R.E.A.D risk rating

STRIDE threat categories:

- Spoofing of user identity
- Tampering
- Repudiation
- Information disclosure (privacy breach or data leak)
- Denial of service (D.o.S)
- Elevation of privilege

The categories are:

- Damage - how bad would an attack be?
- Reproducibility - how easy is it to reproduce the attack?
- Exploitability - how much work is it to launch the attack?
- Affected users - how many people will be impacted?
- Discoverability - how easy is it to discover the threat?

Security Concept

Model Type	**Content**
Static Security Model	Covers all aspects of relevance independent from the data flow. - List of Threats and Assets - Physical Security - Firmware Security (Reverse Engineering Protection, Encryption) - Disk Encryption - Used Cryptographic Mechanisms - Key Material
Dynamic Security Model	Covers all aspects of dynamic system behavior. - List of Threats and Assets - communication matrix - resource separation - availability - cryptographic algorithms - network interfaces, updates, administration

Results and Next Steps

- Hardware based on OTS components
- Open Source Operating System (Linux)
- All security features implemented
- Ongoing security evaluation
- Participant in field tests

Lessons learned

- The focus on the security requirements has identified the top development efforts.
- Threat modeling helps in understanding the data flows (because a threat model requires data flows and helps identifying unclear or inconsistent specifications).
- Technology studies before starting an implementation reduces peoples risks (increases knowledge) and technology risks

Questions?

Armin Lunkeit
armin.lunkeit(at)openlimit.com

SASS15

Fast & Furious - A media style of software development

Axel Allerkamp, Axel Springer SE

Abstract:

Building software for a media corporation is different. This talk depicts specifics of the media environment and highlights its role as critical infrastructure. Current attacks on media corporations are discussed and resulting challenges for the software development process are highlighted.

Vita:

Axel Allerkamp holds a diploma in electrical engineering. He has more than 15 years' experience in information-/cybersecurity. He took responsibility for information security in the armed forces and worked as project leader at Fraunhofer SIT. Currently, he is heading the department Crisis Management, Awareness & Security Evaluation (C.A.S.E.) at Axel Springer SE.

Fraunhofer FOKUS

SASS '15

Fast & Furious

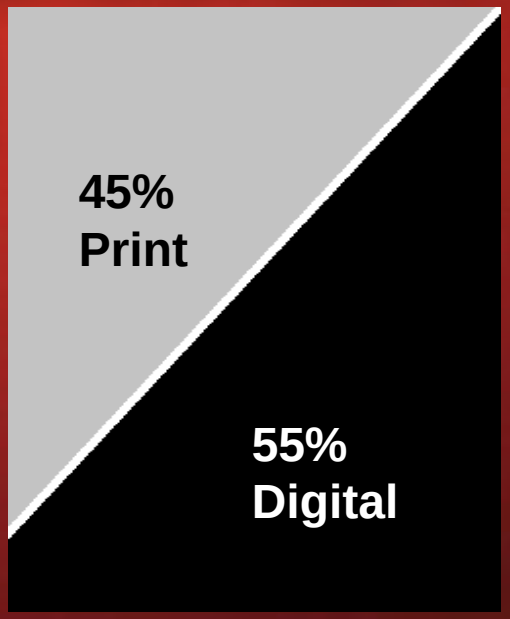
A media style of software development

axel springer

Intro

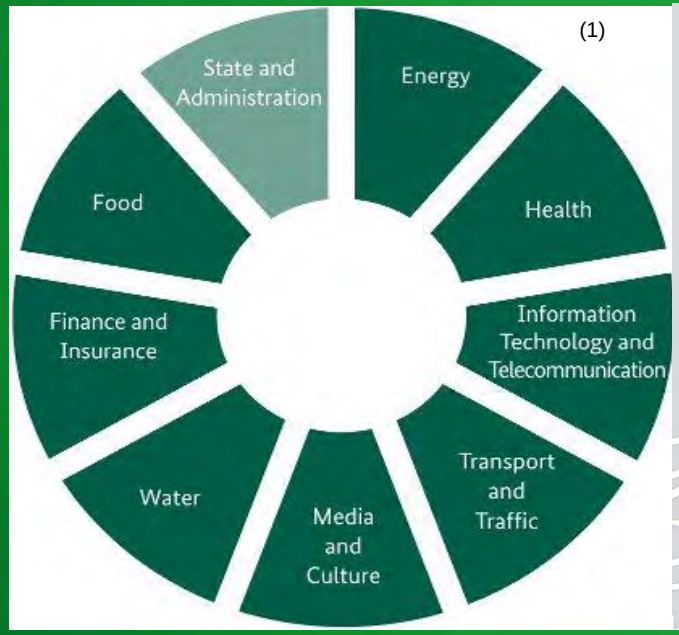
Axel Springer SE

The Leading Digital Publisher

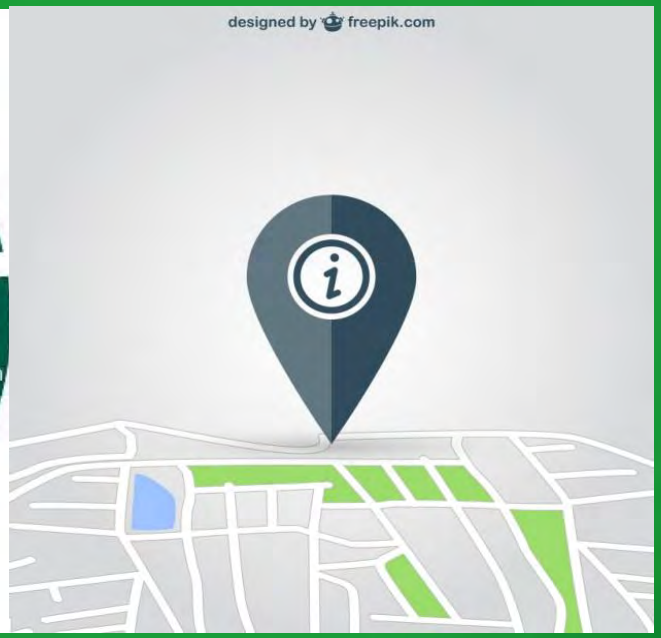


Media

Critical Infrastruktur ?!



(1)



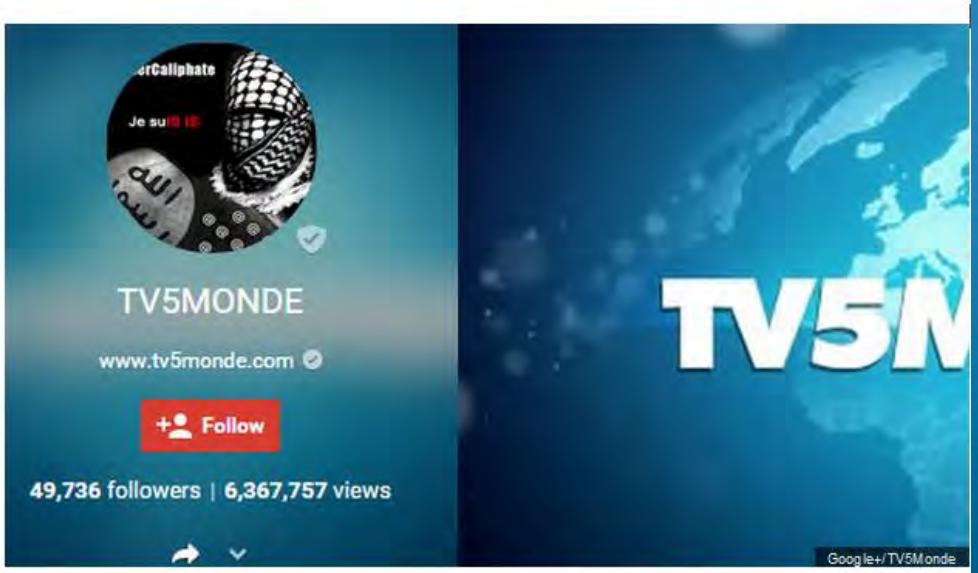
axel springer

Hack@Media

TV5 Monde (fr)

TV5 Monde attack 'by Russia-based hackers'

9 June 2015 | Europe



A cyber attack on the French television network TV5 Monde may have been carried out by Russian-based hackers, police believe.

Jihadist propaganda was posted on the station's website in April by individuals claiming to represent Islamic State.

axel springer

Software development

Fast & Furious

Vielen Dank für Ihre Aufmerksamkeit!

Haben Sie noch Fragen?

axel springer

Axel Allerkamp
Axel Springer SE
10888 Berlin

axel.allerkamp@axelspringer.de

axel springer

(1) http://www.kritis.bund.de/SubSites/Kritis/EN/activities/national/cipimplementationplan/cipimplementationplan_node.html

SASS15

Selecting and deploying risk assessment methods for the development life cycle

Jörn Eichler, Fraunhofer AISEC

Abstract:

Risk assessment is increasingly considered a foundational starting point to develop secure software. Different approaches and methods have been proposed until today. Naturally, not every approach suits a given development organization or project. This talk pinpoints the need for risk assessment in the secure software development lifecycle, depicts properties of several risk assessment approaches, and provides insights on selection and deployment of a matching approach into the development process.

Vita:

Jörn Eichler served several years as developer, analyst, and project manager within international software development and enterprise application integration projects. Focusing on software security he joined the Security Test Lab of Fraunhofer SIT 2008. Since 2013 he is heading the department for Secure Software Engineering at Fraunhofer AISEC.

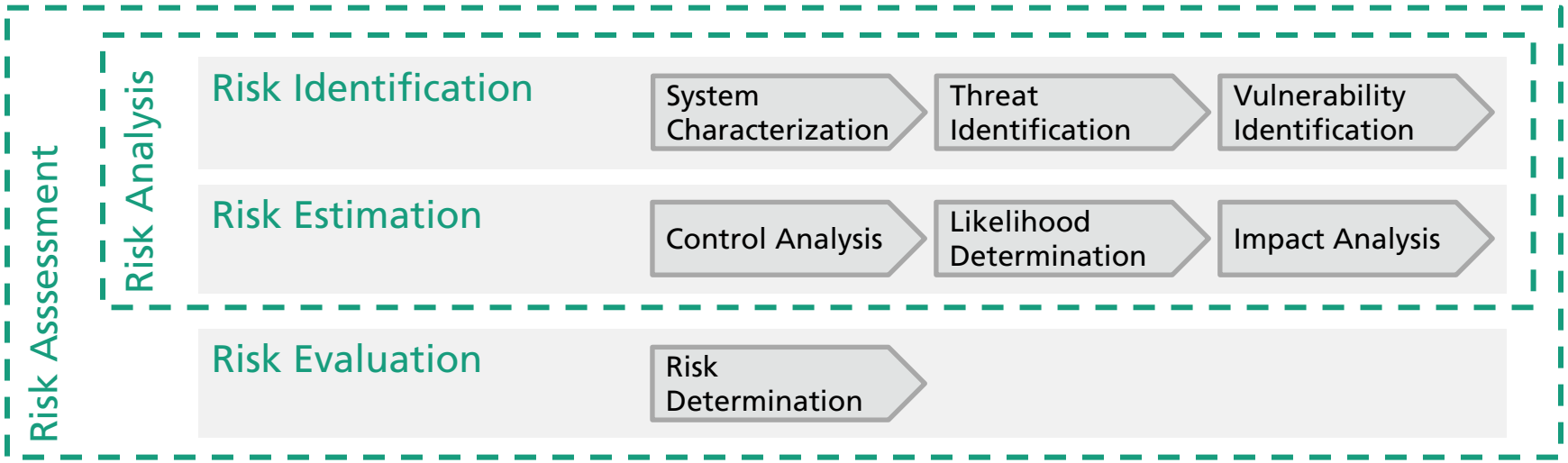
Fraunhofer FOKUS

SASS '15

Selecting and Deploying Risk Assessment Methods for the Development Lifecycle

SASSI-Workshop Berlin

2015-09-16, Dr. Jörn Eichler

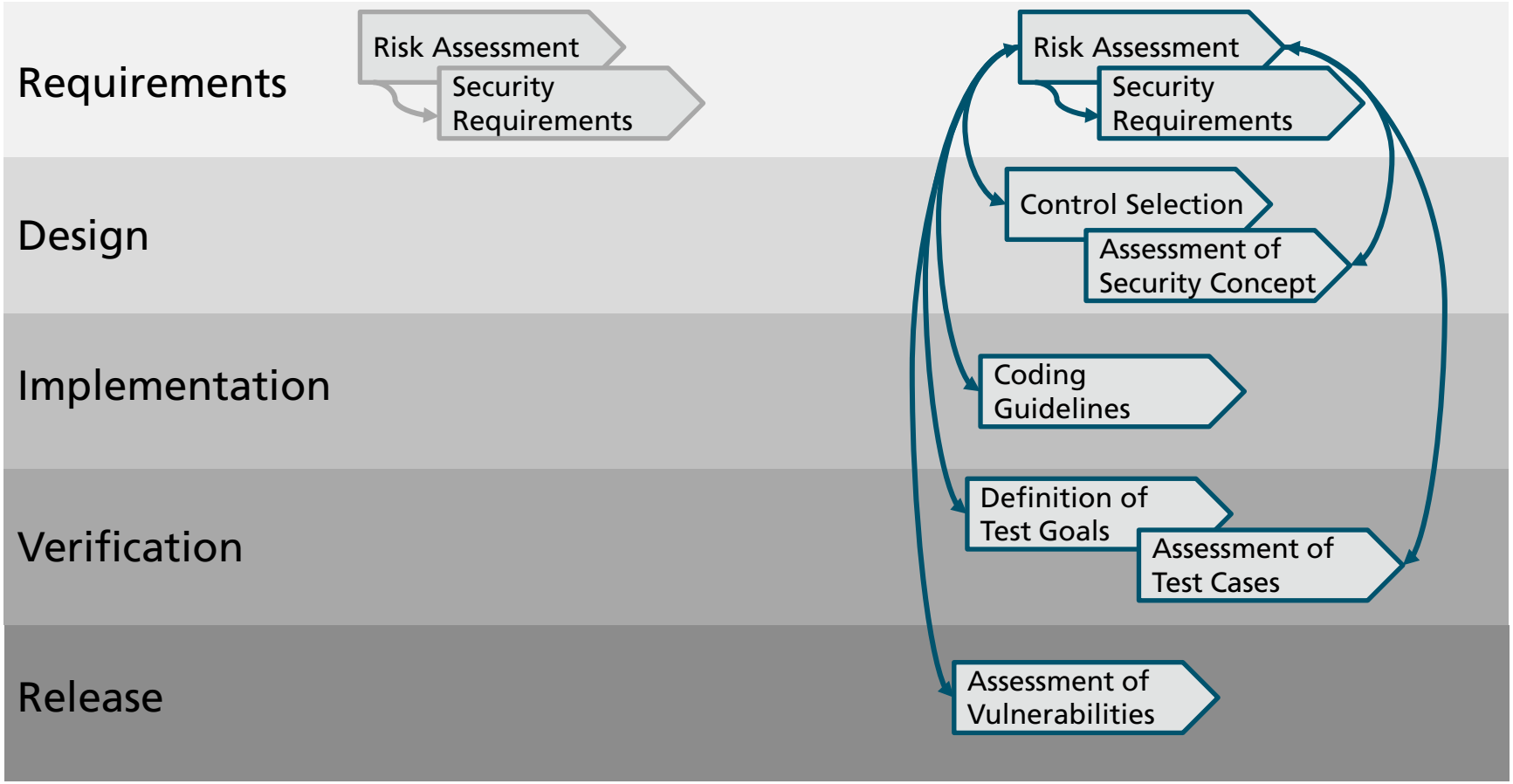


Fraunhofer AISEC

AGENDA

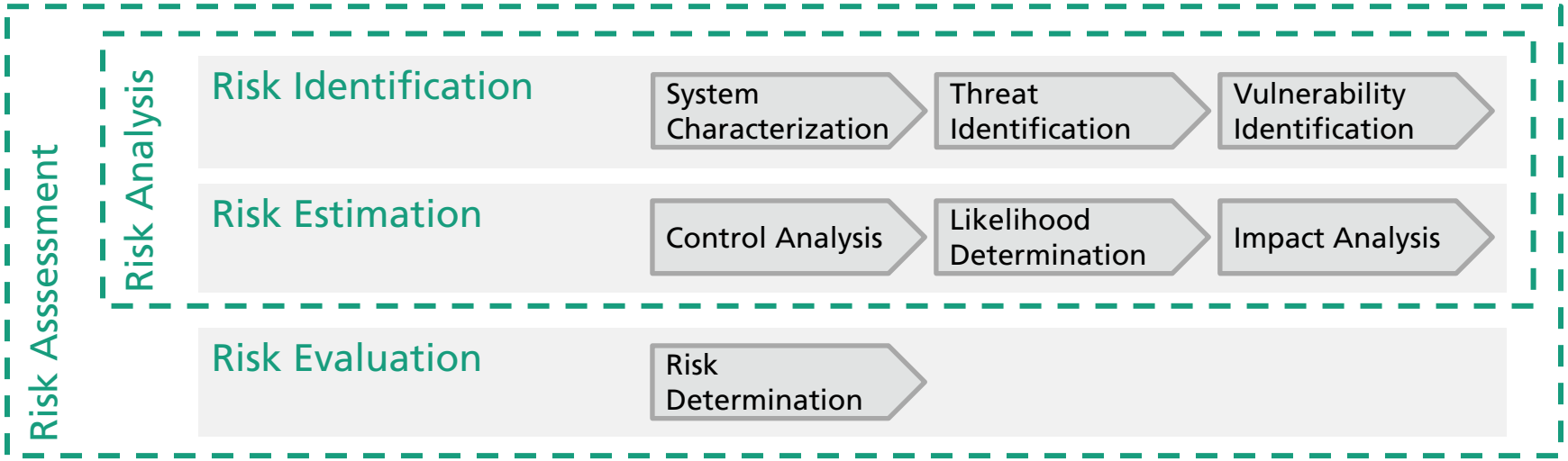
- Motivation
- Essentials
- Comparing approaches
- Tailoring approaches
- Summary

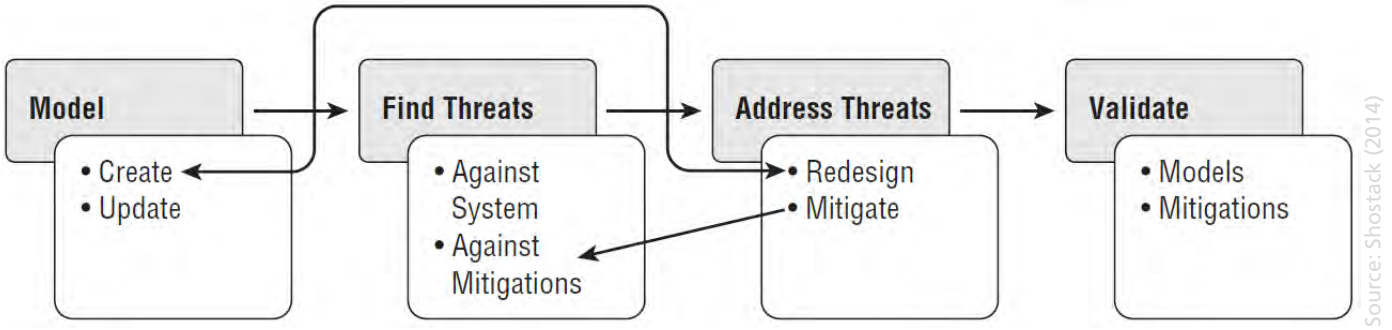
Motivation: The Case for Risk Assessment



Fraunhofer AISEC

Risk Assessment: Essentials





Fraunhofer AISEC

Evaluating Methods for Risk Assessment
(Köster et al. 2009)

Aspect	Criteria	Example: Microsoft
Audience	• Developer and architects • "Real world" environments	Addresses practitioners, rich application experience
Abstraction level	• Different level of abstraction	Multiple levels of data flow diagrams (DFDs)
Collaboration support	• Role model • Asynchronous execution • Knowledge sharing	Supports templates but provides no defined roles and no knowledge base
Evaluation target	• Quantification not required • Ongoing assessments	Focus on concrete scenarios, estimation very weakly supported
Models and techniques	• Specified data structure and notation • Intended vs. current level of security • Reuse of existing model information	DFD and templates provided but intended/current level is not clearly distinguished
Validation and plausibility	• Verification of results • Explication of assumptions • Metrics for assurance level • Tool support with audit trail	Tool provided and assumptions are explicated but verification, assurance level, and audit trail not really

Source: Köster et al. (2009)

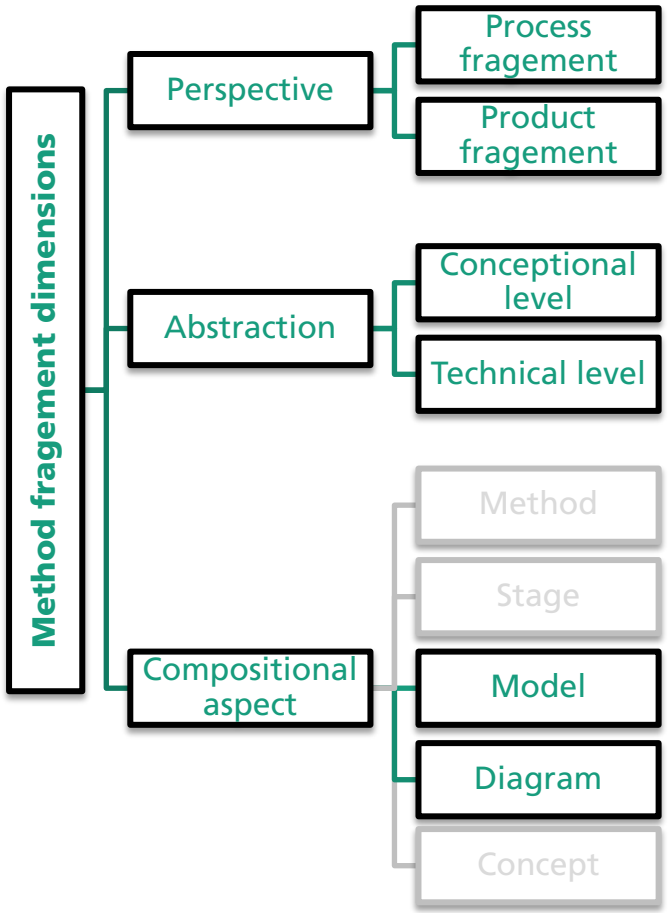
Exemplary Evaluation of Multiple Methods
(Köster et al. 2009)

Aspect	CORAS	OCTAVE	Trike	EBIOS	Microsoft
Audience	◒	◒	○	○	●
Abstraction level	●	○	●	○	●
Collaboration support	◒	◒	○	◒	◒
Evaluation target	◒	◒	◒	○	○
Models and techniques	●	○	●	◒	◒
Validation and plausibility	◒	○	◒	◒	◒

Source: Köster et al. (2009)

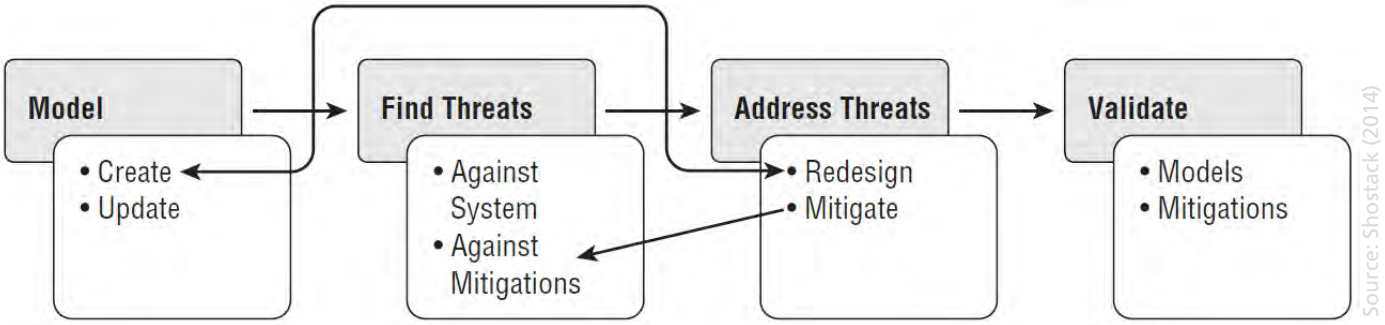
Analyzing and Decomposing Methods
Applying a Method Engineering Framework

- A method
  - is a repeatable procedure
  - that specifies the steps
  - involved in solving a specific problem
- Method Engineering
  - *Selection and assembly* of method fragments to provide adequate methods
  - Situational method engineering "encompasses all aspects of creating a development method for a specific situation" (Brinkkemper 1996)



Fraunhofer AISEC

Exemplary Method Analysis: SDL/A Threat Modeling



Source: Shostack (2014)

Fragment	Short name	Dimensions	Method chunks
F1.1	Diagram creation	pro / conc / dia	C1.1
F1.2	Threat identification	pro / conc / mod	C1.2
F1.3	Selection of mitigations	pro / conc / mod	C1.3
F1.4	Identification of update needs	pro / conc / mod	C1.4
F1.5	Model validation	pro / conc / mod	C1.5
WP1.1	DFDs	prd / conc / dia	C1.1, C1.2, C1.3, C1.4, C1.5
WP1.2	Threats	prd / conc / mod	C1.2, C1.3, C1.4, C1.5
WP1.3	Mitigations	prd / conc / mod	C1.3, C1.4, C1.5
T1.1	Threat modeling tool	prd / tech / mod	(C1.1), (C1.2), (C1.3), (C1.4), (C1.5)

Source: Renatus et al. (2015)

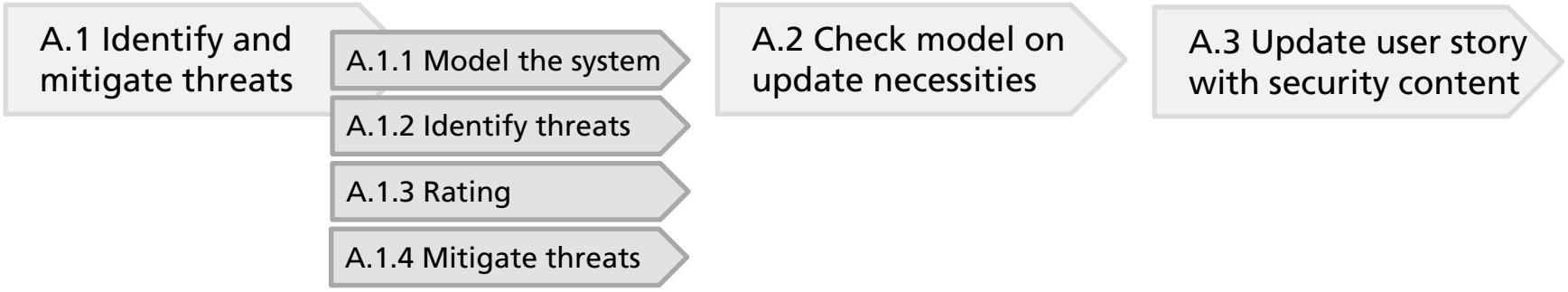
Method Comparison Based on Decomposition

	SDL/A Threat Modeling	AVS Threat Modeling	Threat Modeling Express
Non-Monolicithy	● (5)	● (5)	○ (2)
Segmentation of product fragments	◒	●	○
Additional artifacts	●	●	●
Policies / guidelines	●	◒	○
Scrum modifications: activities / work products	● / ○	○ / ●	○ / ○
Estimations	◒	◒	○

Source: Renatus et al. (2015)

Fraunhofer AISEC

Method Tailoring Based on Decomposition



Chunk	Short name	Process fragment	Work products	Related chunks
C.1	Model system	A.1.1	WP.1	C1.1
C.2	Identify threats	A.1.2	WP.1, WP.2	C1.2
C.3	Mitigate threats	A.1.4	WP.1, WP.2, WP.3	C1.3
C.4	Check update necessities	A.2	WP.1, WP.2	C1.4, C2.2
C.5	Update user story	A.3	WP.1, WP.2, WP.3, WP.4	C2.1
C.6	Rating	A.1.3	WP.1	C2.4

Source: Renatus et al. (2015)

Fraunhofer AISEC

Summary

- Risk assessment is a cornerstone for secure software
- Many activities depend on up-to-date risk assessments
- Therefore, choose your risk assessment method wisely
  - Understand differences between existing approaches
  - Investigate your internal requirements
- COTS methods do not always fit your needs
  - Tailoring increases acceptance and benefit
  - Systematic approaches for analysis and tailoring provide means for streamlined adoption

Contact

Fraunhofer Institute for
Applied and Integrated Security (AISEC)

Parkring 4, 85748 Garching near Munich
Alexanderstr. 9, 10178 Berlin (Berlin Office)

Dr. Jörn Eichler

Head of Department „Secure Software Engineering“

Tel.: +49 89 32299 86-152
Fax: +49 89 32299 86-299

joern.eichler@aisec.fraunhofer.de
http://www.aisec.fraunhofer.de/

References

ISO (2008)	ISO 27005: Information technology — Security techniques — Information security risk management
Köster et al. (2009)	Information security assessments for embedded systems development: An evaluation of methods. *Proceedings of 8th Annual Security Conference*
NIST 800-30 (2002)	Risk Management Guide for Information Technology Systems
Renatus et al. (2015)	Method Selection and Tailoring for Agile Threat Assessment and Mitigation. *Proceedings of the First International Workshop on Agile Secure Software Development (ASSD)*
Shostack (2014)	Threat Modeling: Designing for Security. *Wiley*

SASS15

Automated detection and prevention of Security Vulnerabilities in Multi-Party Web Applications

Luca Compagna, SAP SE

Abstract:

Security testing and validation is a key research area at SAP, aiming to enhance SAP products and processes with cost-effective techniques for automated detection and prevention of security vulnerabilities. In this talk we will first introduce an overview of the main topics in this area (e.g., dynamic analysis fighting injections via E2E taint tracking, open-source vulnerability assessment, automated security checks for best practices) and we will then dig into a few of them to provide more concreteness. In particular, we will target the multi-party web applications domain and present a few techniques---ranging from design-time security protocol analysis to black-box dynamic testing---that we devised to support developers and security experts at SAP over the software development lifecycle of these applications. We will demo these techniques and discuss their pro & cons with special focus on the cost and potential exploitation at SAP.

Vita:

Dr. Luca Compagna joined SAP in 2006. He is Research Expert at SAP Product Security Research, where he is contributing to the SAP research strategy and responsible for various internally- and externally-funded research projects. He received his MSc in Informatics Engineering from the University of Genova and his Ph.D. in Computer Science jointly from the University of Genova and Edinburgh. His area of interests include cyber-security, security engineering, automated reasoning, security testing, and their application to industrial relevant scenarios. He contributed to various projects on information security and he has published various scientific publications in his area of interest.

Fraunhofer FOKUS

SASS '15

Automated detection and prevention of Security Vulnerabilities in Multi-Party Web Applications

SASSI Workshop, Sept, 2015

Luca Compagna, Product Security Research, SAP SE

Product Security Research in a Nutshell…

PEOPLE

30

SECURITY RESEARCHERS & EXPERTS

+7 PhDs

LOCATIONS

SOPHIA ANTIPOLIS

KARLSRUHE

SKILLS

PRIVACY

SECURITY TESTING, VALIDATION & MONITORING

CRYPTOGRAPHY

EDUCATION & GAMIFICATION

MOTTO

SECURITY

AS DIFFERENTIATOR

PROJECTS

9 COLLABORATIVE

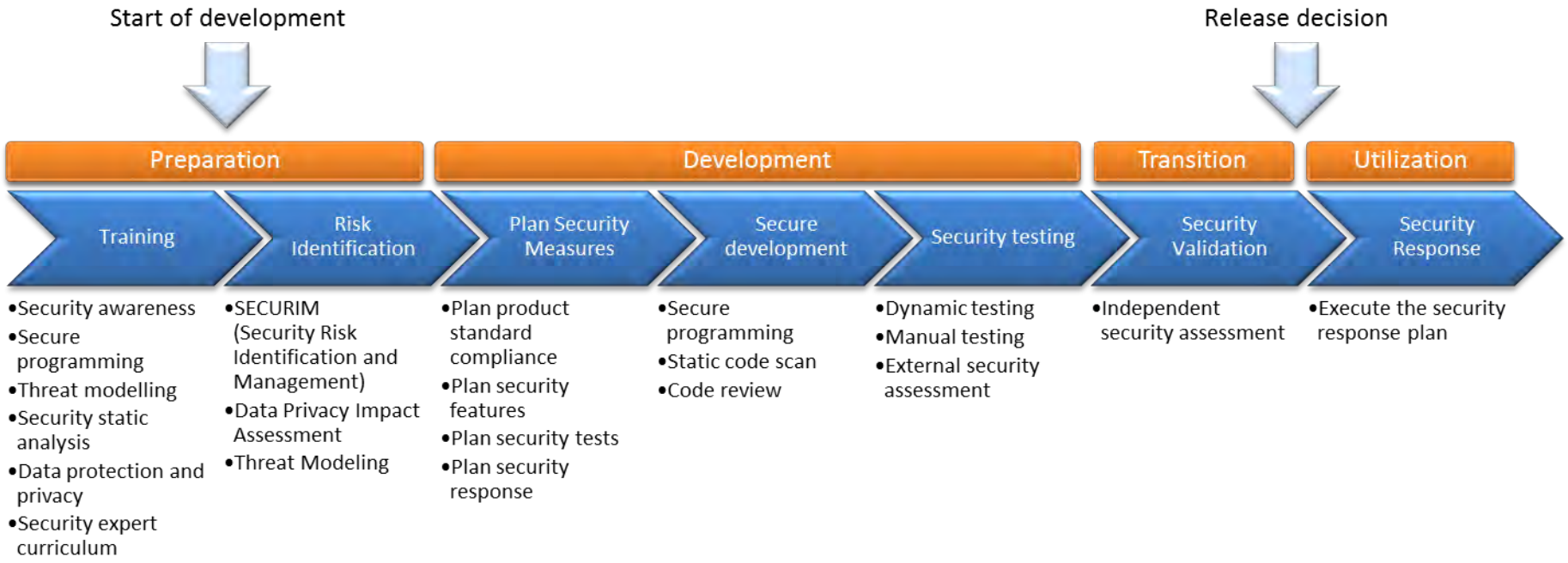
4 INTERNAL

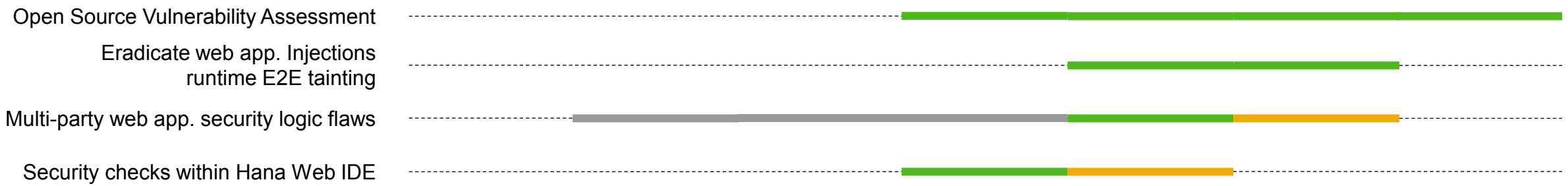
+60 PARTNERS

SAP S2DL process and our research

Business value / Cost

Automation/ Accuracy / Usability / …





Open Source Security Assessment

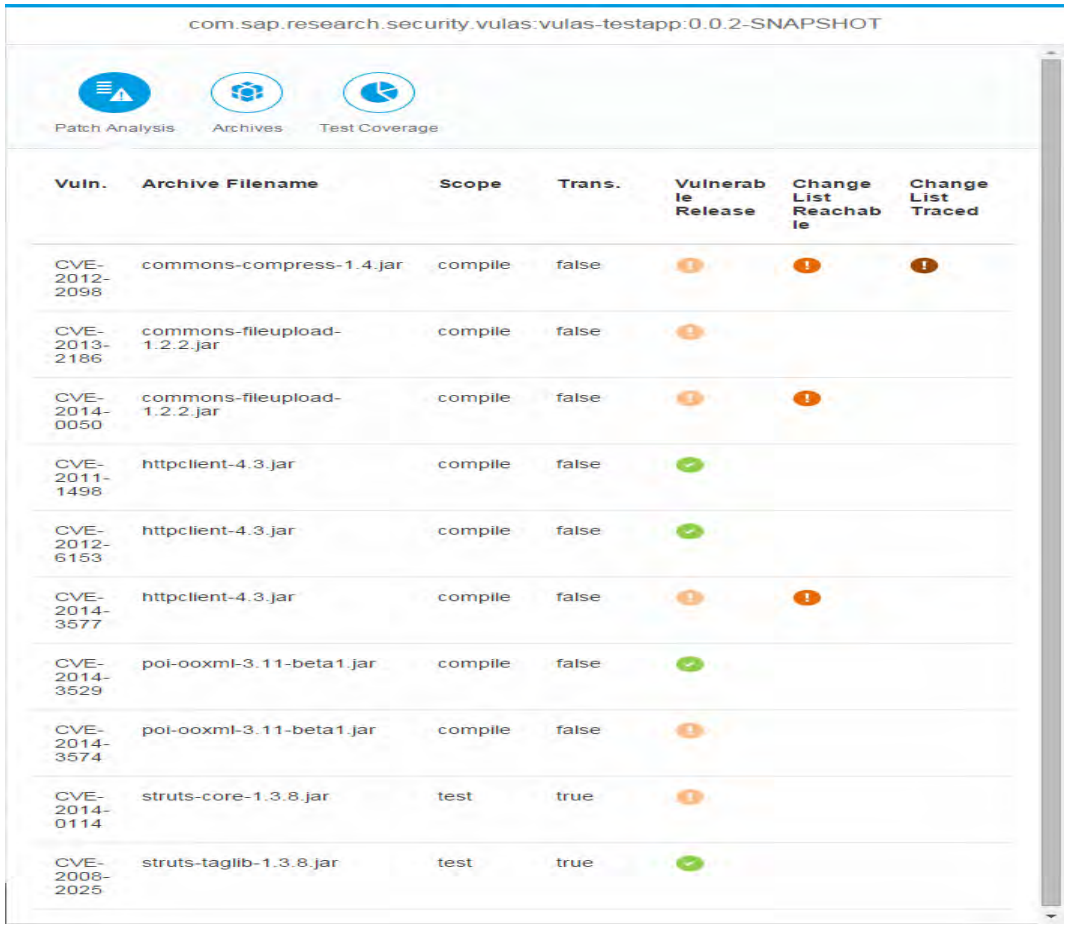
Applications increasingly depend on (complex) open-source software (OSS)

What if a new vulnerability in a bundled OSS is disclosed, e.g., Heartbleed?

Shall immediately create and ship a patch for customers or can wait until next regular app update?

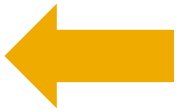
Likelihood that a vulnerable OSS component is exploitable in my application?

1. vulnerable release of OSS comp. bundled with app
2. vulnerable code potentially reachable (static analysis)
3. vulnerable code actually reached (during app tests)



Vuln.	Archive Filename	Scope	Trans.	Vulnerable Release	Change List Reachable	Change List Traced
CVE-2012-2098	commons-compress-1.4.jar	compile	false	!	!	!
CVE-2013-2186	commons-fileupload-1.2.2.jar	compile	false	!		
CVE-2014-0050	commons-fileupload-1.2.2.jar	compile	false	!	!	
CVE-2011-1498	httpclient-4.3.jar	compile	false	✓		
CVE-2012-6153	httpclient-4.3.jar	compile	false	✓		
CVE-2014-3577	httpclient-4.3.jar	compile	false	!	!	
CVE-2014-3529	poi-ooxml-3.11-beta1.jar	compile	false	✓		
CVE-2014-3574	poi-ooxml-3.11-beta1.jar	compile	false	!		
CVE-2014-0114	struts-core-1.3.8.jar	test	true	!		
CVE-2008-2025	struts-taglib-1.3.8.jar	test	true	✓		

Outline

Multi-Party Web Applications (MPWAs) 

Security Threat Identification and Testing: model-driven

Black-Box Security Testing: vulnerability-driven (two slides)

Multi-Party Web Applications (MPWAs)

Many modern web applications relies on TTPs to deliver services to their Users

- e.g., 27% of Alexa top 1000 uses Facebook SSO

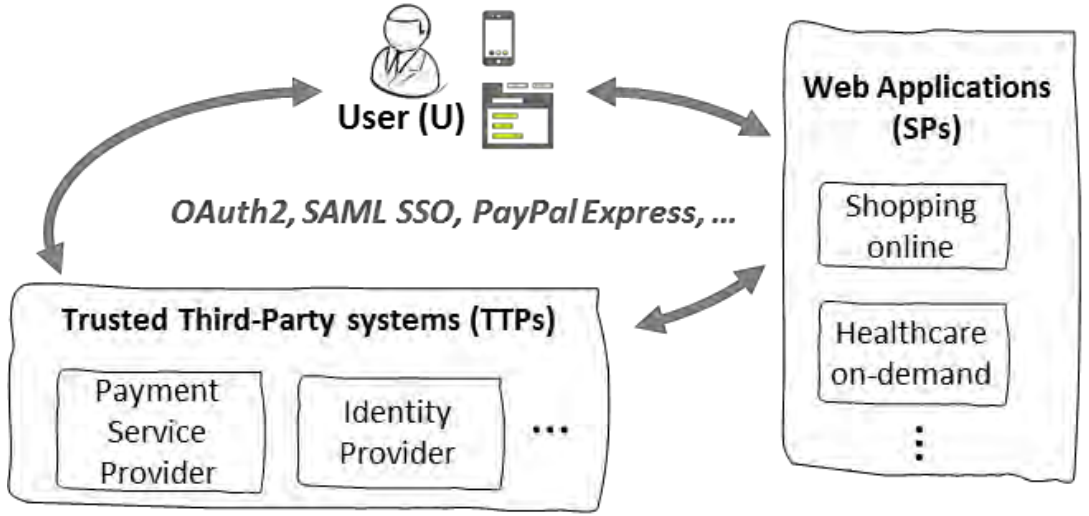
Based on:

- protocols (interoperability)
- *bilateral* trust relationships

TTPs are assumed to be trustworthy

But neither SP nor C are assumed so

E.g., a compromised SP should not impact another one



Illustrative example

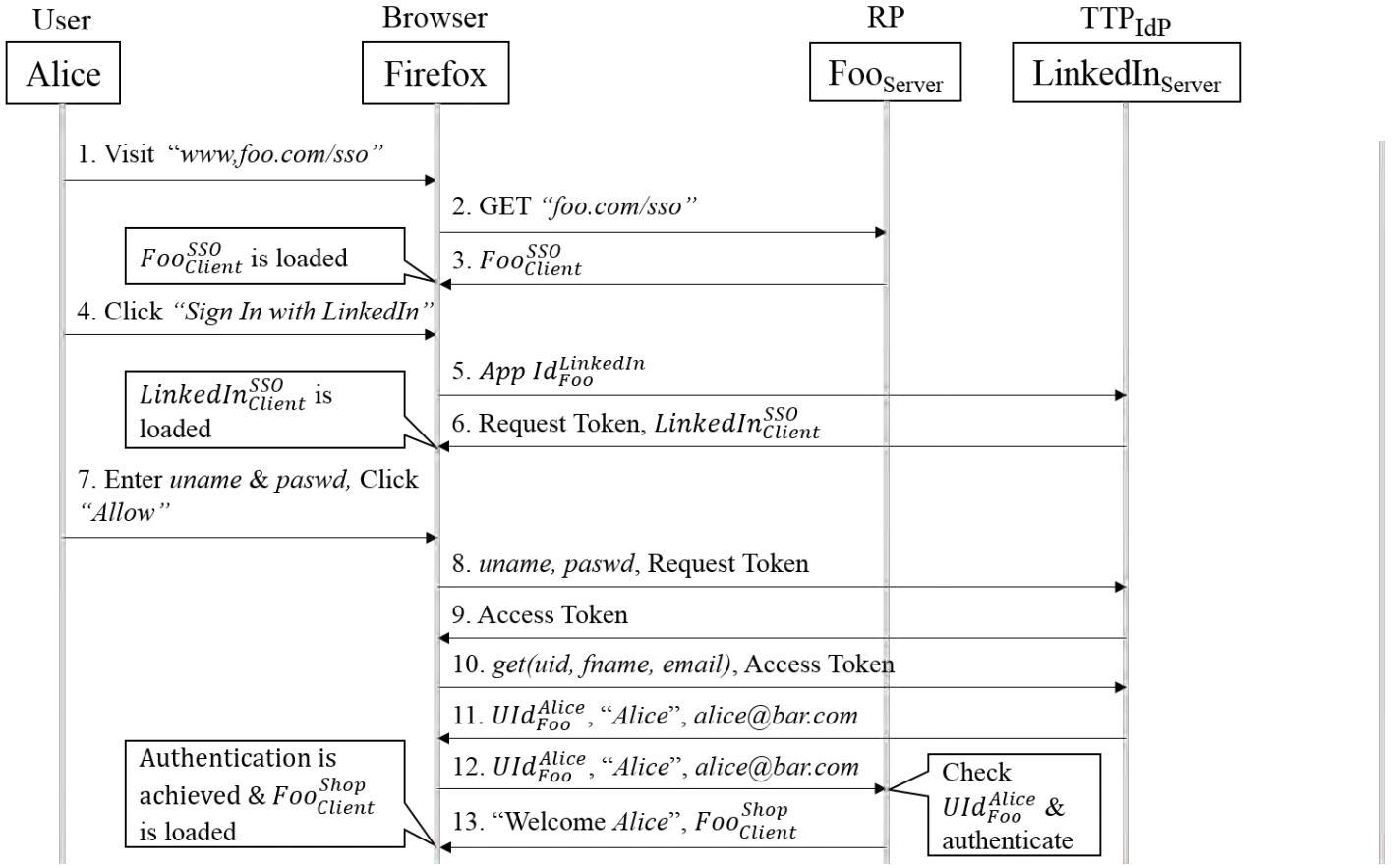
Foo is an **online shop** that relies on

Illustrative example

Foo is an **online shop** that relies on

… **LinkedIn** for social SSO

- Linkedin's Javascript API-based SSO
- OAuth2-based



Illustrative example

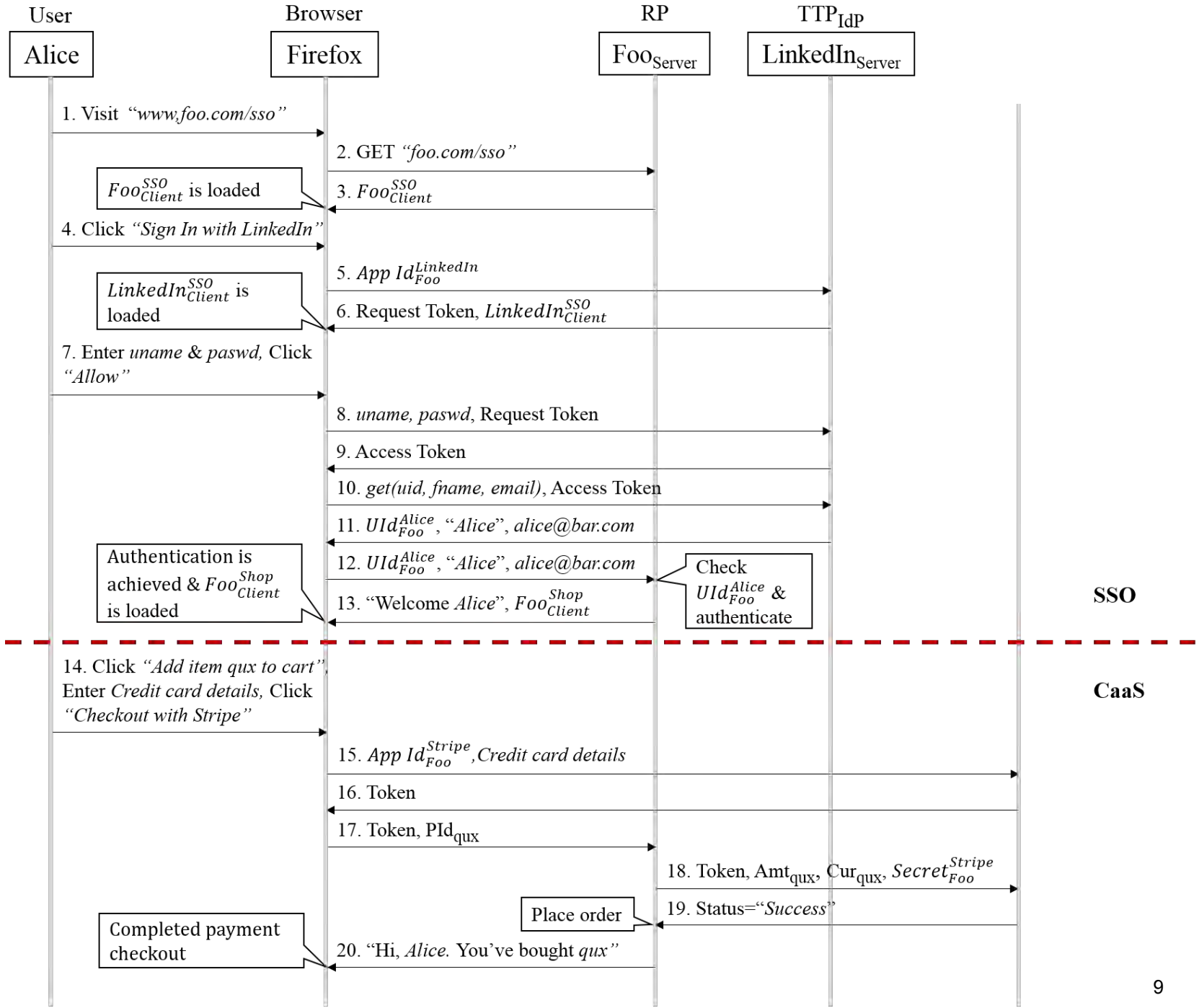
Foo is an **online shop** that relies on

… **LinkedIn** for social SSO

- Linkedin's Javascript API-based SSO
- OAuth2-based

… **Stripe** for payment checkout

- proprietary protocol
- integrated in >17K web-sites



Challenges and Motivations

Several vulnerabilities reported in literature

Mainly **implementation** issues, but also **design** ones

Challenges include:

- highly **configurable** protocols, **interpretation** of the specifications
- **internal requirements**, total cost for development (**TCD**)
  - lack of (security) **testing**, but also
  - lack of **tool support** for developers
- …

Paper	Tech	Application(s)
Sec.4 of [22]	FV	SPs implementing Google's SAML SSO
Sec.5.2.1 of [36]	FV	SPs implementing OAuth 2.0 implicit flow-based Facebook SSO
Sec.IV.A.1 of [30]	BB	PayPal Payments Standard implementation in SPs using os-Commerce 2.3.1 or AbanteCart1.0.4
Sec.V.A of [33]	WB	SPs implementing CaaS solutions of 2Checkout, Chrono-Pay, PSiGate and Luottokunta (v1.2)
Sec.IV.A.2 of [30]	BB	PayPal Express Checkout implementation in SPs using Open-Cart 1.5.3.1 or TomatoCart 1.1.7
Sec.4.2 of [34]	BB	SPs implementing OAuth 2.0 implicit flow-based Facebook SSO
Sec.6.2 of [23]	BB	developer.mozilla.com (SP) implementing BrowserID
Sec.V.C of [24]	FV	CitySearch.com (SP) using Facebook SSO (OAuth 2.0 Auth. Code Flow)
Sec.4 of [21]	FV	SPs implementing Google's SAML SSO
Bug 2 of [1]	M	Github (TTP) implementing OAuth 2.0 Authorization Code flow-based SSO

Legend: FV: formal verification; BB: black-box; WB: white-box; M: manual inspection

[1] Account hijacking by leaking authorization code. http://www.oauthsecurity.com/.
[21] Armando, A., Carbone, R., Compagna, L., Cuellar, J., Pellegrino, G., and Sorniotti, A. From multiple credentials to browser-based single sign-on: Are we more secure? IFIP 2011.
[22] Armando, A., Carbone, R., Compagna, L., Cuellar, J., and Tobarra, L. Formal Analysis of SAML 2.0 Web Browser Single Sign-On: Breaking the SAML-based Single Sign-On for Google Apps. FMSE 2008
[24] Bai, G., Lei, J., Meng, G., Venkatraman, S. S., Saxena, P., Sun, J., Liu, Y., and Dong, J. S. Authscan: Automatic extraction of web authentication protocols from implementations. NDSS 2013
[30] Pellegrino, G., and Balzarotti, D. Toward black-box detection of logic flaws in web applications. NDSS 2014
[33] Sun, F., Xu, L., and Su, Z. Detecting logic vulnerabilities in e-commerce applications. NDSS 2014
[34] Wang, R., Chen, S., and Wang, X. Signing me onto your accounts through facebook and google: A traffic-guided security study of commercially deployed single-sign-on web services. S&P 2012
[36] Wang, R., Zhou, Y., Chen, S., Qadeer, S., Evans, D., and Gurevich, Y. Explicating SDKs: Uncovering assumptions underlying secure authentication and authorization. USENIX 2013

Our (applied) research directions

How can we best detect MPWAs vulnerabilities during the software development lifecycle?

- which techniques?
- are they expressive/accurate enough?
- can they be automated?
- what is the cost-benefit ratio?
- tool support for our developers?
- …

Outline

Multi-Party Web Applications (MPWAs)

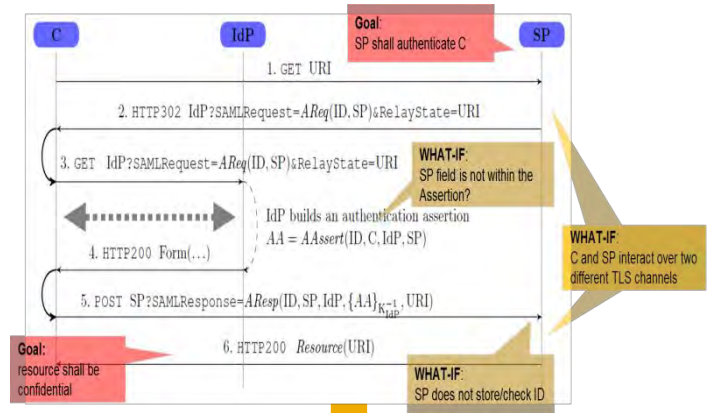
Security Threat Identification and Testing: model-driven 

Black-Box Security Testing: vulnerability-driven (two slides)

Our approach: historical view – Episode 1

AVANTSSAR

Specifications



Formal models

Abstract threats

Input
Output

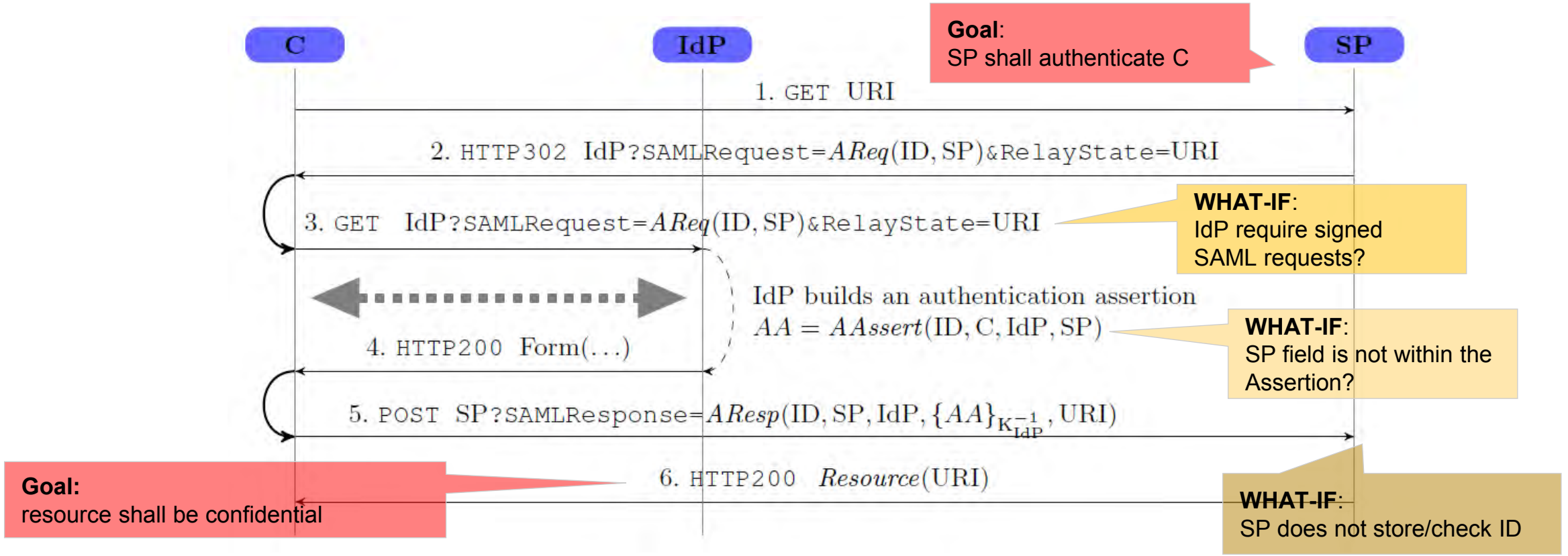
Formal models capture the MPWA

- ❑ one model for each WHAT-IF combination

Model checking to detect vulnerabilities (if any)

- ❑ build-in Dolev-Yao intruder
- ❑ set rewriting underneath
- ❑ LTL expressiveness for goals
- ❑ abstract communication channels as LTL constraints
- ❑ exhaustive exploration of the search space

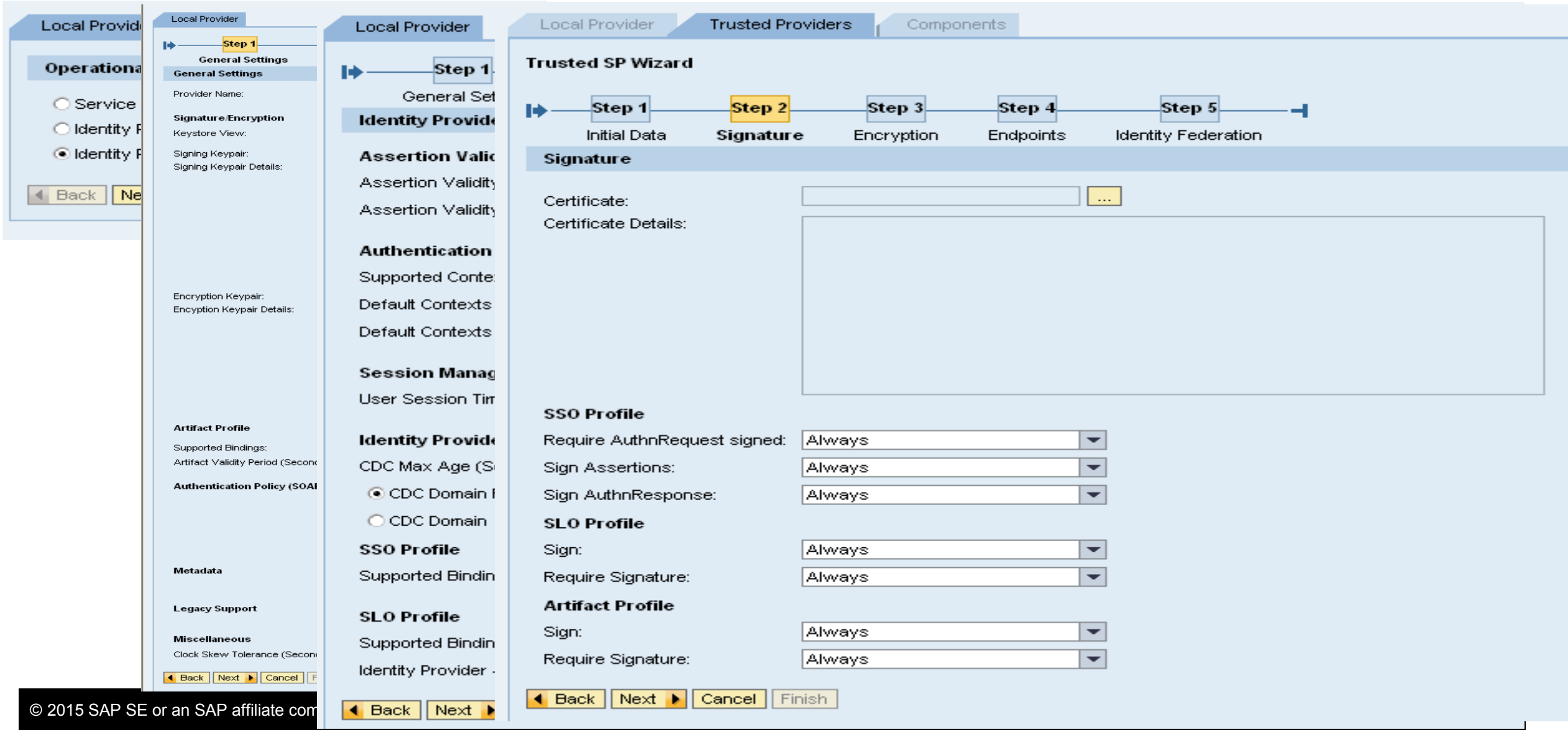
E.g., Developing and deploying SAML SSO



SAML2 comes with many other profiles, protocols, optional attributes, etc...

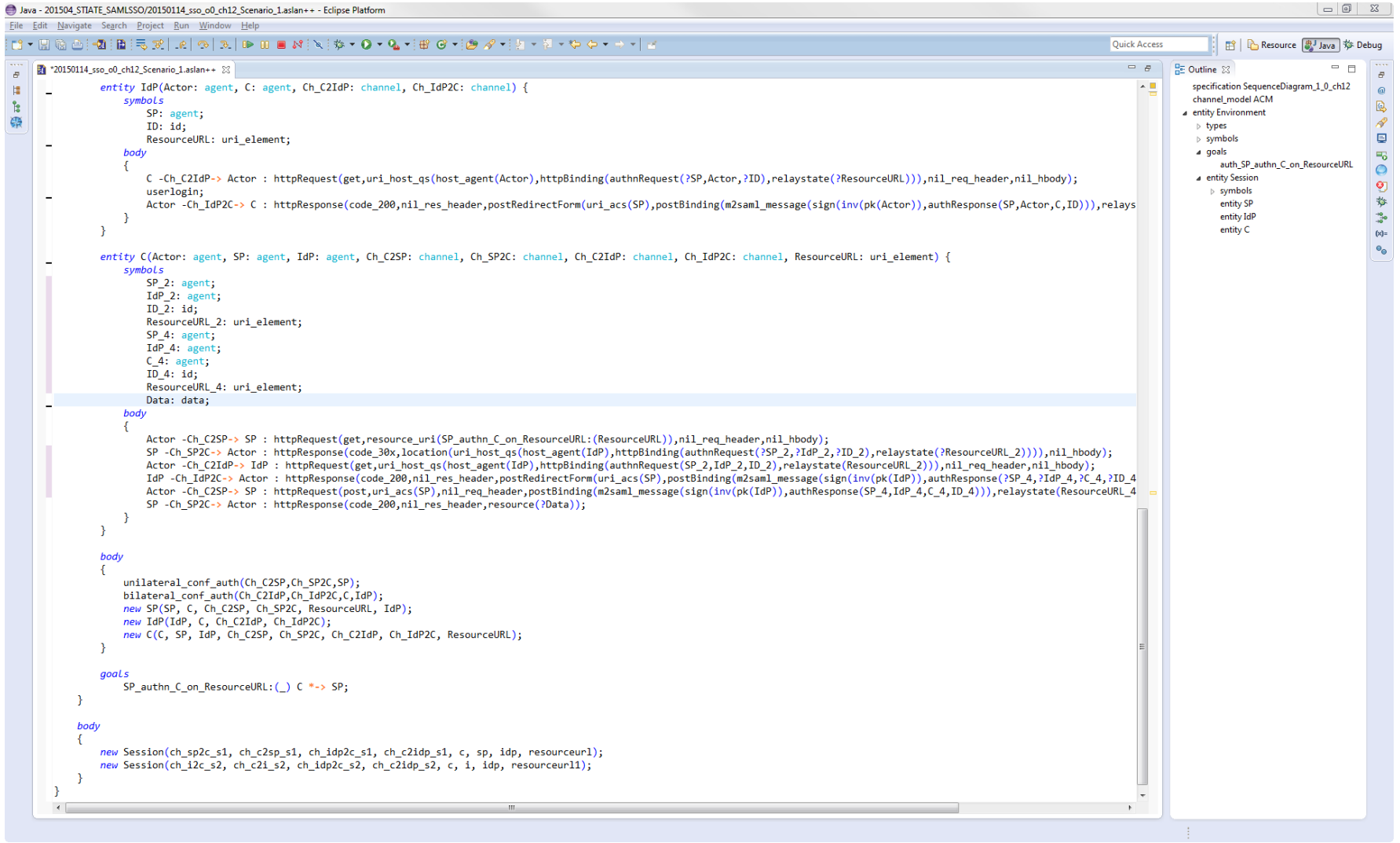
E.g., Developing and deploying SAML SSO

SAP NetWeaver Next Generation SSO



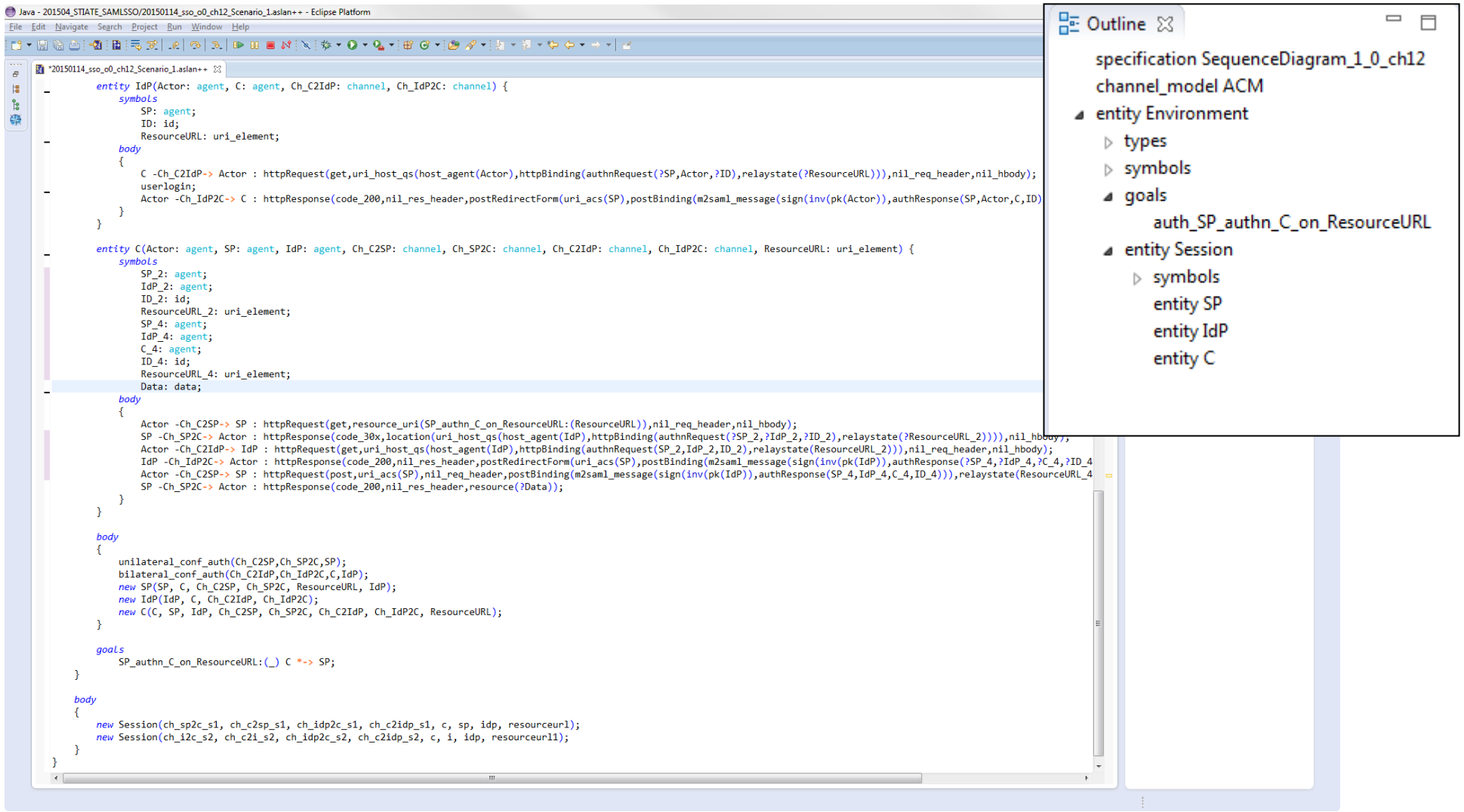
E.g., Developing and deploying SAML SSO

Formal specifications



E.g., Developing and deploying SAML SSO

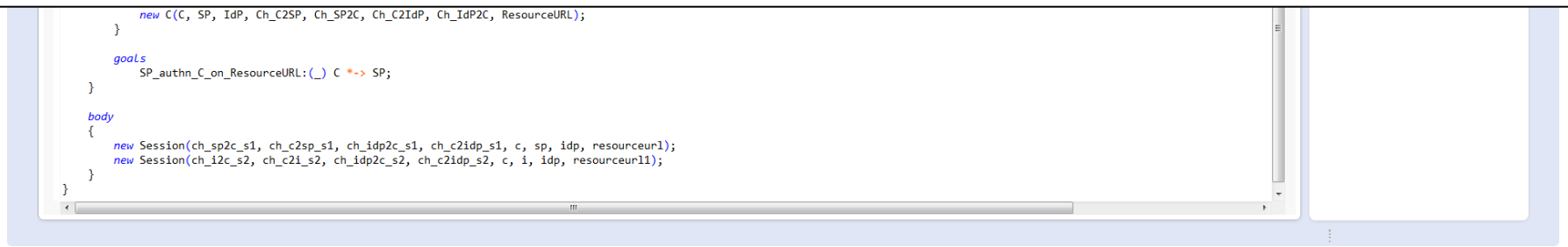
Formal specifications



E.g., Developing and deploying SAML SSO

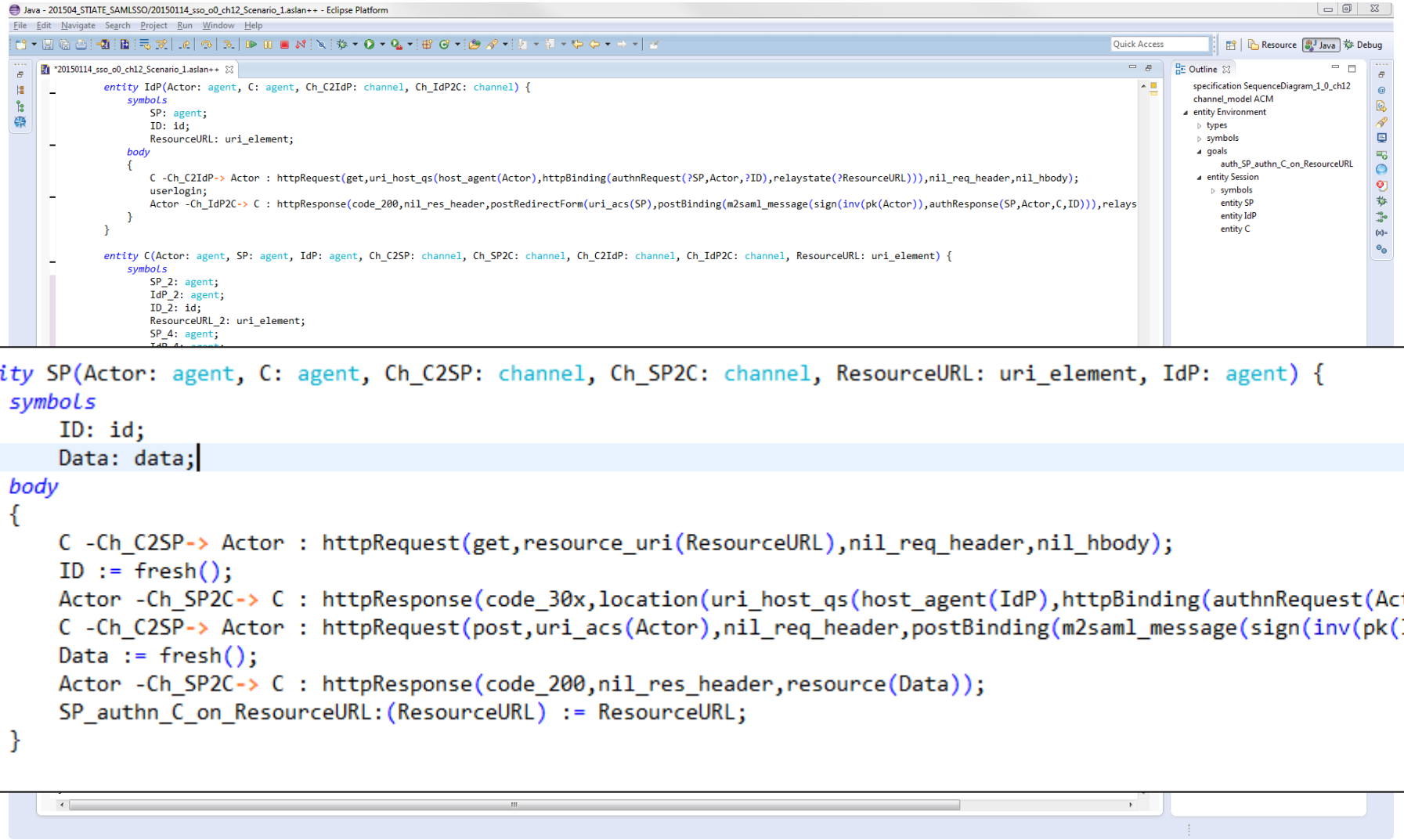
Formal specifications

```
entity C(Actor: agent, SP: agent, IdP: agent, Ch_C2SP: channel, Ch_SP2C: channel, Ch_C2IdP: channel, Ch_IdP2C: channel, ResourceURL: uri_element) {
    symbols
        SP_2: agent;
        IdP_2: agent;
        ID_2: id;
        ResourceURL_2: uri_element;
        SP_4: agent;
        IdP_4: agent;
        C_4: agent;
        ID_4: id;
        ResourceURL_4: uri_element;
        Data: data;
    body
    {
        Actor -Ch_C2SP-> SP : httpRequest(get,resource_uri(SP_authn_C_on_ResourceURL:(ResourceURL)),nil_req_header,nil_hbody);
        SP -Ch_SP2C-> Actor : httpResponse(code_30x,location(uri_host_qs(host_agent(IdP),httpBinding(authnRequest(?SP_2,?IdP_2,?ID_2),relaystate(?Resou
        Actor -Ch_C2IdP-> IdP : httpRequest(get,uri_host_qs(host_agent(IdP),httpBinding(authnRequest(SP_2,IdP_2,ID_2),relaystate(ResourceURL_2))),nil_re
        IdP -Ch_IdP2C-> Actor : httpResponse(code_200,nil_res_header,postRedirectForm(uri_acs(SP),postBinding(m2saml_message(sign(inv(pk(IdP)),authRespo
        Actor -Ch_C2SP-> SP : httpRequest(post,uri_acs(SP),nil_req_header,postBinding(m2saml_message(sign(inv(pk(IdP)),authResponse(SP_4,IdP_4,C_4,ID_4
        SP -Ch_SP2C-> Actor : httpResponse(code_200,nil_res_header,resource(?Data));
    }
}
```



E.g., Developing and deploying SAML SSO

Formal specifications



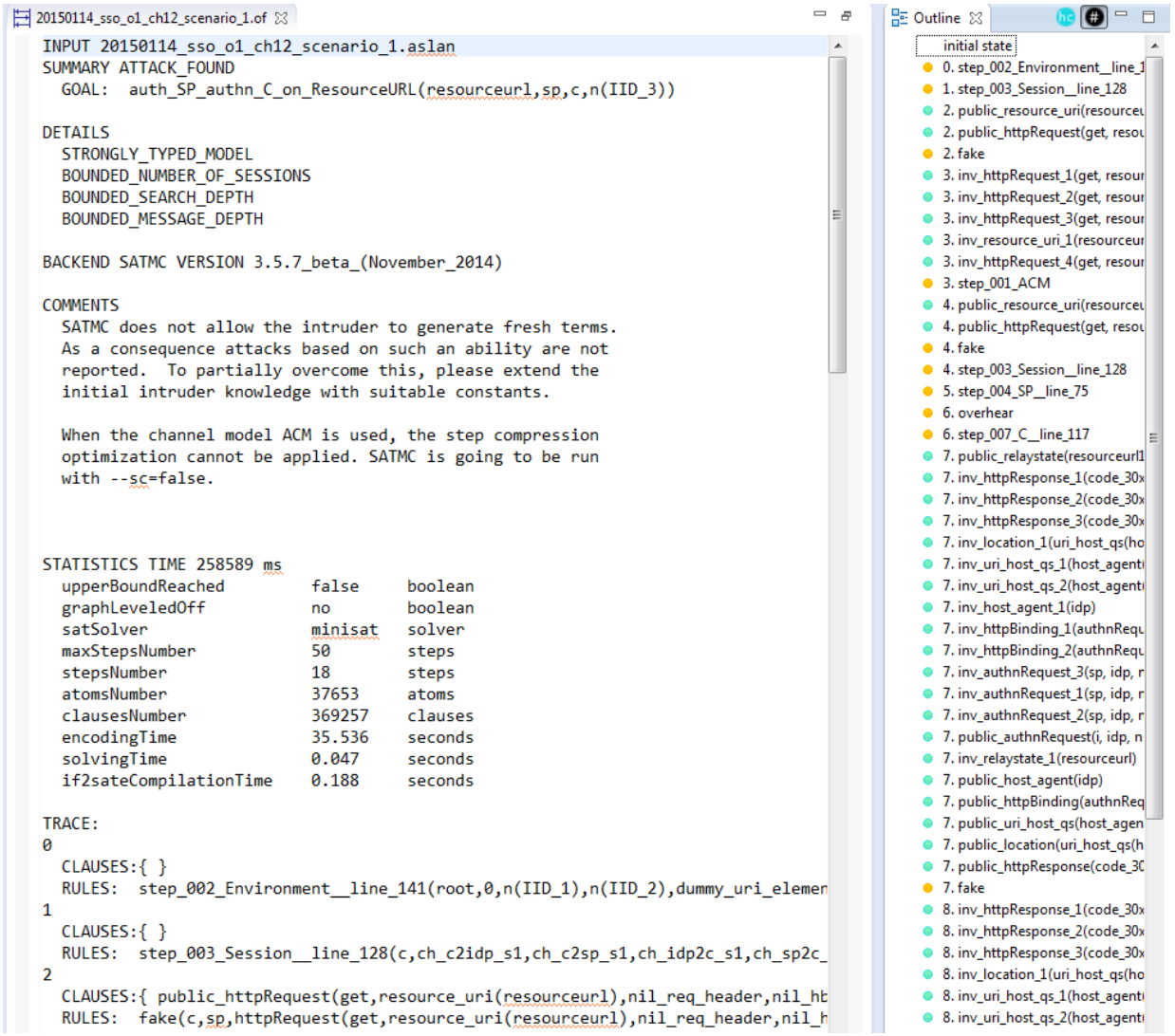

```
entity SP(Actor: agent, C: agent, Ch_C2SP: channel, Ch_SP2C: channel, ResourceURL: uri_element, IdP: agent) {
    symbols
        ID: id;
        Data: data;
    body
    {
        C -Ch_C2SP-> Actor : httpRequest(get,resource_uri(ResourceURL),nil_req_header,nil_hbody);
        ID := fresh();
        Actor -Ch_SP2C-> C : httpResponse(code_30x,location(uri_host_qs(host_agent(IdP),httpBinding(authnRequest(Actor,IdP,ID),relaystate(ResourceURL))
        C -Ch_C2SP-> Actor : httpRequest(post,uri_acs(Actor),nil_req_header,postBinding(m2saml_message(sign(inv(pk(IdP)),authResponse(Actor,IdP,C,ID)))
        Data := fresh();
        Actor -Ch_SP2C-> C : httpResponse(code_200,nil_res_header,resource(Data));
        SP_authn_C_on_ResourceURL:(ResourceURL) := ResourceURL;
    }
}
```

E.g., Developing and deploying SAML SSO

Formal specifications

```
entity Session(Ch_SP2C: channel, Ch_C2SP: channel, Ch_IdP2C: channel, Ch_C2IdP: channel, C: agent, SP: agent, IdP: agent, ResourceURL: uri_element) {
    % [..]
    body
    {
        unilateral_conf_auth(Ch_C2SP,Ch_SP2C,SP);
        bilateral_conf_auth(Ch_C2IdP,Ch_IdP2C,C,IdP);
        new SP(SP, C, Ch_C2SP, Ch_SP2C, ResourceURL, IdP);
        new IdP(IdP, C, Ch_C2IdP, Ch_IdP2C);
        new C(C, SP, IdP, Ch_C2SP, Ch_SP2C, Ch_C2IdP, Ch_IdP2C, ResourceURL);
    }

    goals
        SP_authn_C_on_ResourceURL:(_) C *-> SP;
}

body
{
    new Session(ch_sp2c_s1, ch_c2sp_s1, ch_idp2c_s1, ch_c2idp_s1, c, sp, idp, resourceurl);
    new Session(ch_i2c_s2, ch_c2i_s2, ch_idp2c_s2, ch_c2idp_s2, c, i, idp, resourceurl1);
}
```

E.g., Developing and deploying SAML SSO
Abstract threat



E.g., Developing and deploying SAML SSO

Abstract threat

E.g., Developing and deploying SAML SSO
outcomes from Episode 1

Authentication flaw in SAML-based SSO for Google Apps [1]

Authentication flaw in SAML2 SSO security and SAML errata corrige [2]

Internal consultancy at SAP:

- SAP NetWeaver Next Generation Single Sign-On

[1] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, M. L. Tobarra. Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps. FMSE 2008.

[2] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, G. Pellegrino, A. Sorniotti. An authentication flaw in browser-based Single Sign-On protocols: Impact and remediations. Computers & Security journal

E.g., Developing and deploying SAML SSO
outlook of internal consultancy at SAP

Results

- identified **safe/unsafe** configurations
- **NW NGSSO well designed** and **developed**
- efficient **modus-operandi** with valuable exchanges: business units ↔ researchers

Details

- small deviations from the standard (e.g., InResponseTo): **no issues identified**
- flaw detected in the standard for the **SAML Authentication protocol** used in SP-initiated?
  - cookies strongly mitigate this issue
  - sanitization of ReplayState is extremely important
  - standard asks for integrity of RelayState, but no all vendors do that
  - e.g., Google, simpleSAMLphp, … did not and suffered thus of a serious XSS

Industrial exploitation?

- Formal model / formal analyser

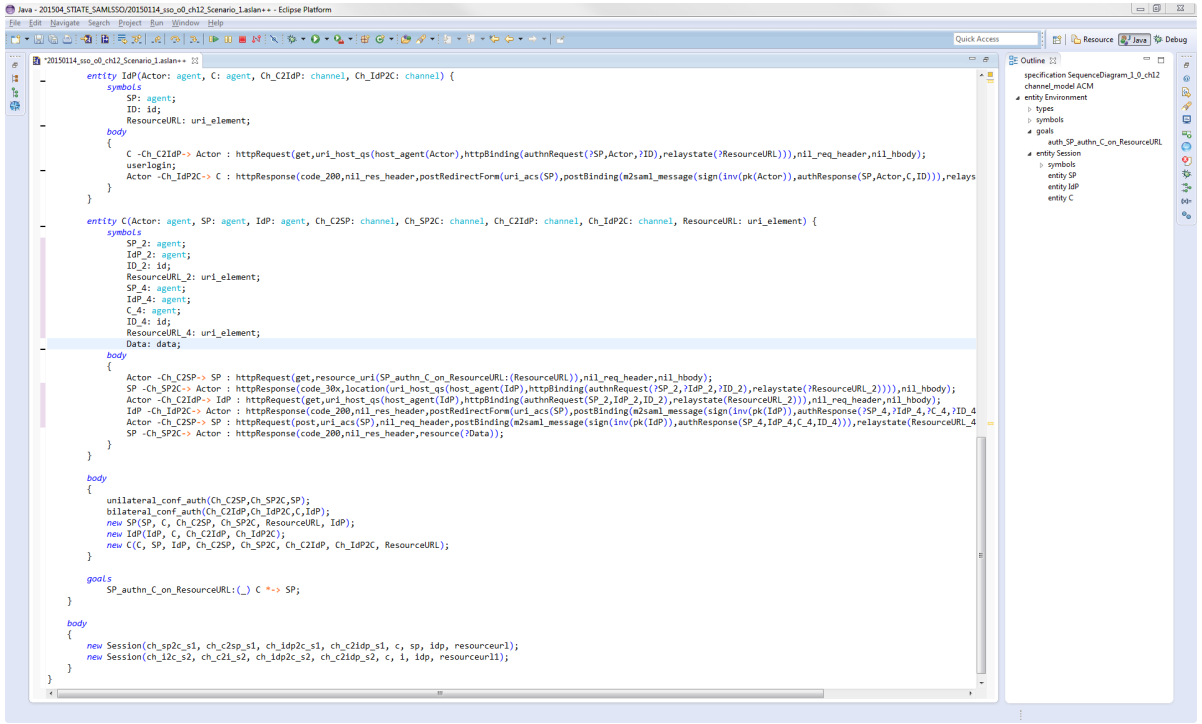
  → Accessibility / Usability

  → Abstraction / Performances

- Industrial landscape and requirements

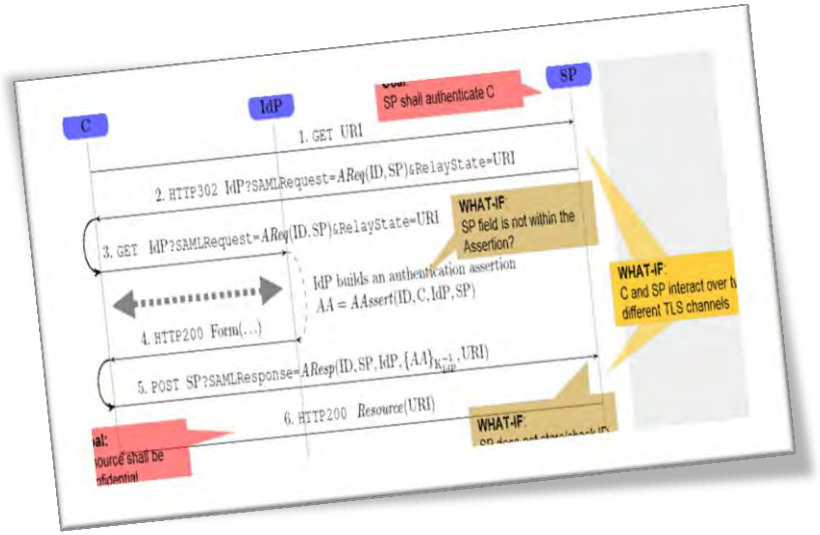
  → Automation / Integration

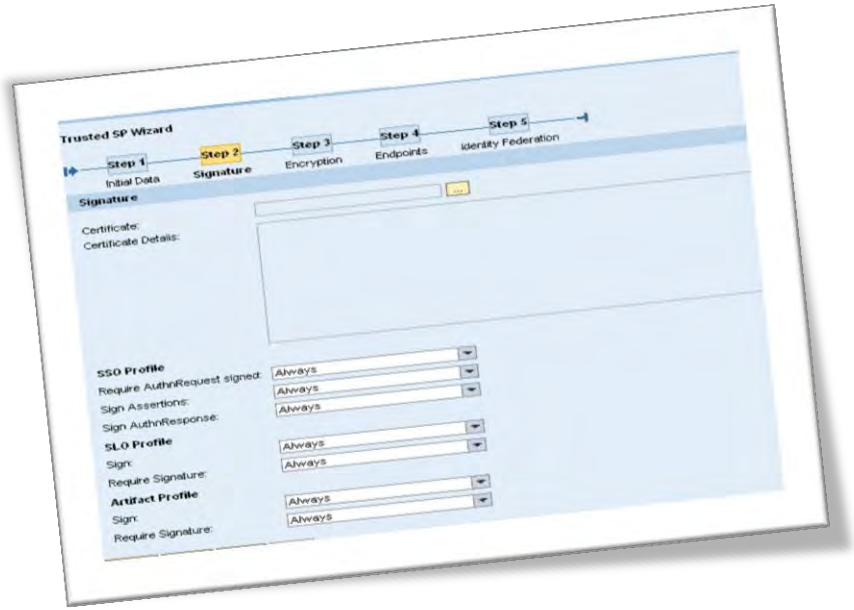
➢ Cost-benefit ratio??



Motivations toward Episode 2

Can we **bridge abstract** and **real world**?



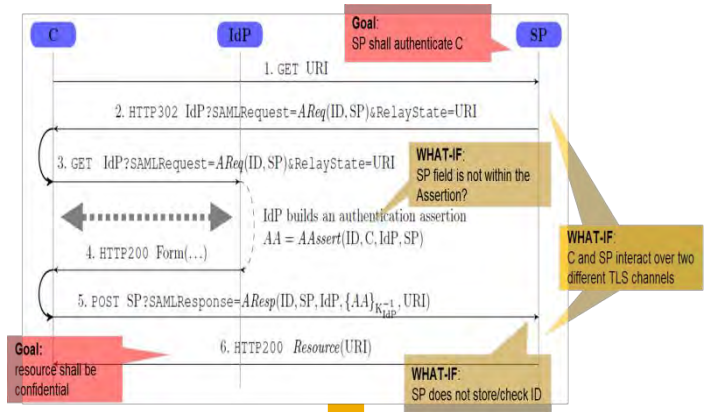


Can we improve **usability**?

Our approach: historical view – Episode 2

SPaCIoS
http://www.spacios.eu

Specifications



Formal models

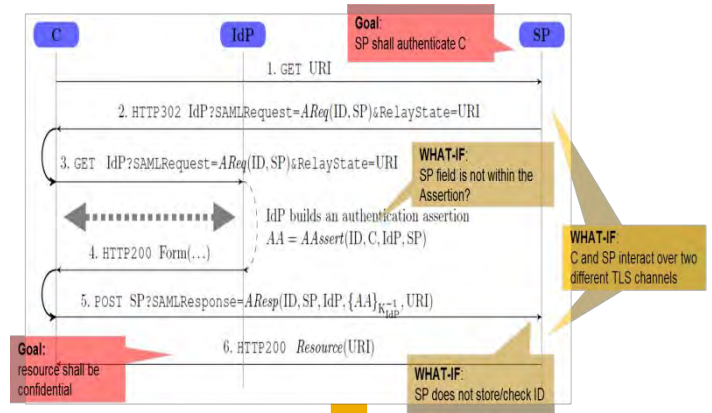
Abstract threats

Input
Output

Our approach: historical view – Episode 2

SPaCIoS
http://www.spacios.eu

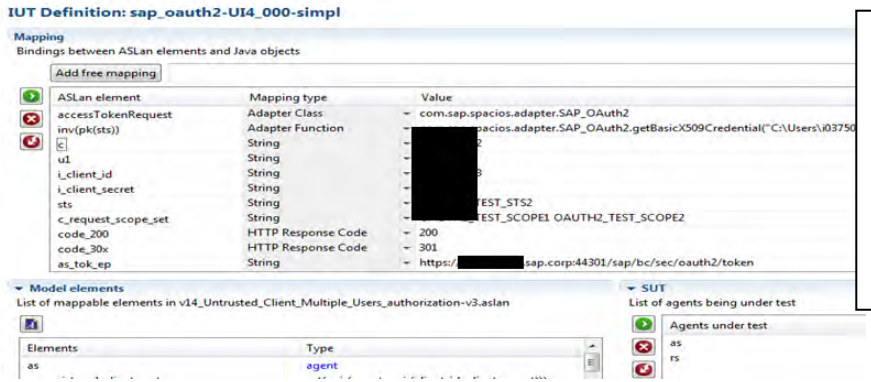
Specifications



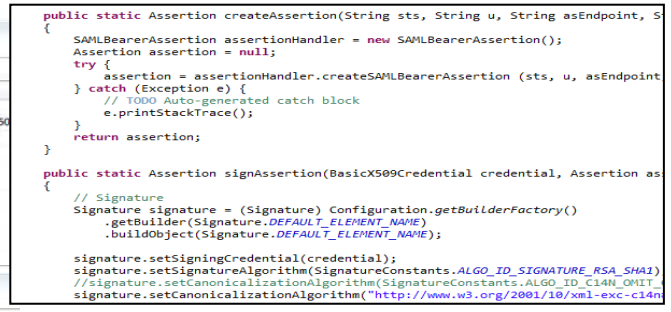
Formal models

Abstract threats

Testing data



Adapter



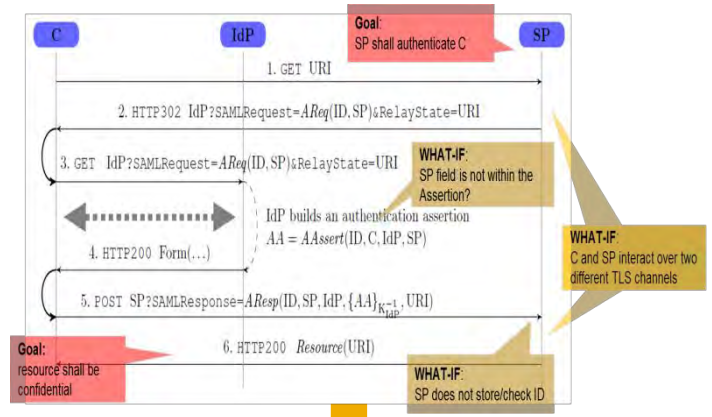
```
public static Assertion createAssertion(String sts, String u, String asEndpoint, S
{
    SAMLBearerAssertion assertionHandler = new SAMLBearerAssertion();
    Assertion assertion = null;
    try {
        assertion = assertionHandler.createSAMLBearerAssertion (sts, u, asEndpoint
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return assertion;
}

public static Assertion signAssertion(BasicX509Credential credential, Assertion as
{
    // Signature
    Signature signature = (Signature) Configuration.getBuilderFactory()
        .getBuilder(Signature.DEFAULT_ELEMENT_NAME)
        .buildObject(Signature.DEFAULT_ELEMENT_NAME);

    signature.setSigningCredential(credential);
    signature.setSignatureAlgorithm(SignatureConstants.ALGO_ID_SIGNATURE_RSA_SHA1)
    //signature.setCanonicalizationAlgorithm(SignatureConstants.ALGO_ID_C14N_OMIT_
    signature.setCanonicalizationAlgorithm("http://www.w3.org/2001/10/xml-exc-c14n
```

Input
Output

Our approach: historical view – Episode 2

http://www.spacios.eu

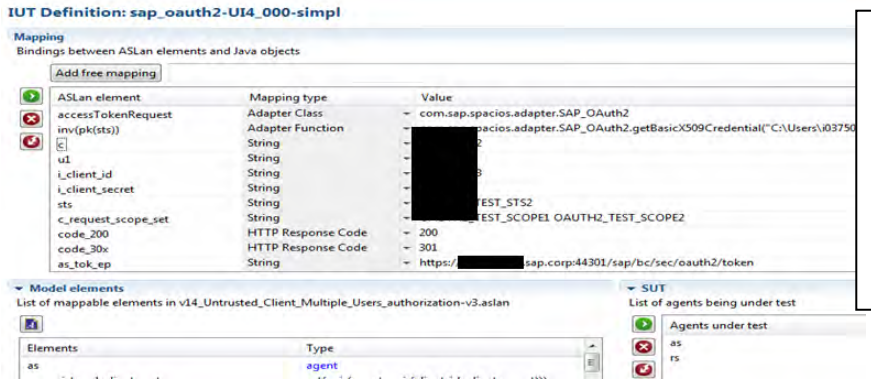
Specifications



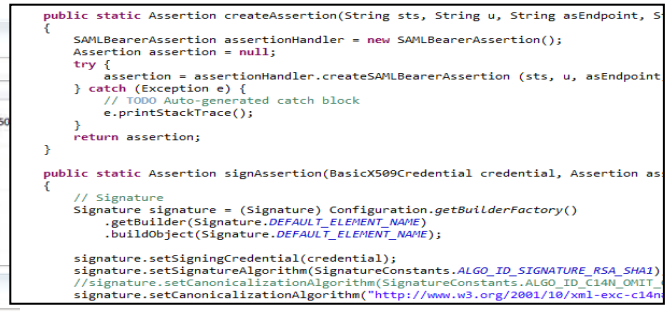
Formal models

Abstract threats

Testing data

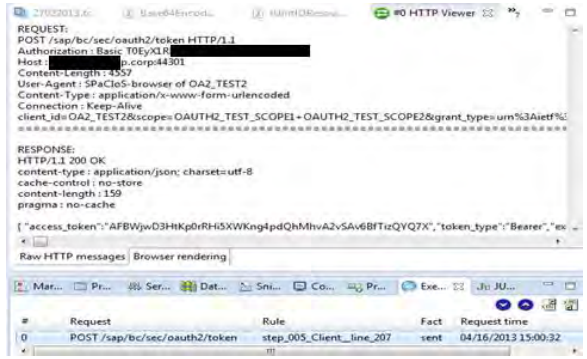


Adapter



```
public static Assertion createAssertion(String sts, String u, String asEndpoint, S
{
    SAMLBearerAssertion assertionHandler = new SAMLBearerAssertion();
    Assertion assertion = null;
    try {
        assertion = assertionHandler.createSAMLBearerAssertion (sts, u, asEndpoint
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return assertion;
}

public static Assertion signAssertion(BasicX509Credential credential, Assertion as
{
    // Signature
    Signature signature = (Signature) Configuration.getBuilderFactory()
        .getBuilder(Signature.DEFAULT_ELEMENT_NAME)
        .buildObject(Signature.DEFAULT_ELEMENT_NAME);

    signature.setSigningCredential(credential);
    signature.setSignatureAlgorithm(SignatureConstants.ALGO_ID_SIGNATURE_RSA_SHA1)
    //signature.setCanonicalizationAlgorithm(SignatureConstants.ALGO_ID_C14N_OMIT_
    signature.setCanonicalizationAlgorithm("http://www.w3.org/2001/10/xml-exc-c14n
```

Concrete test-cases (generation and execution)

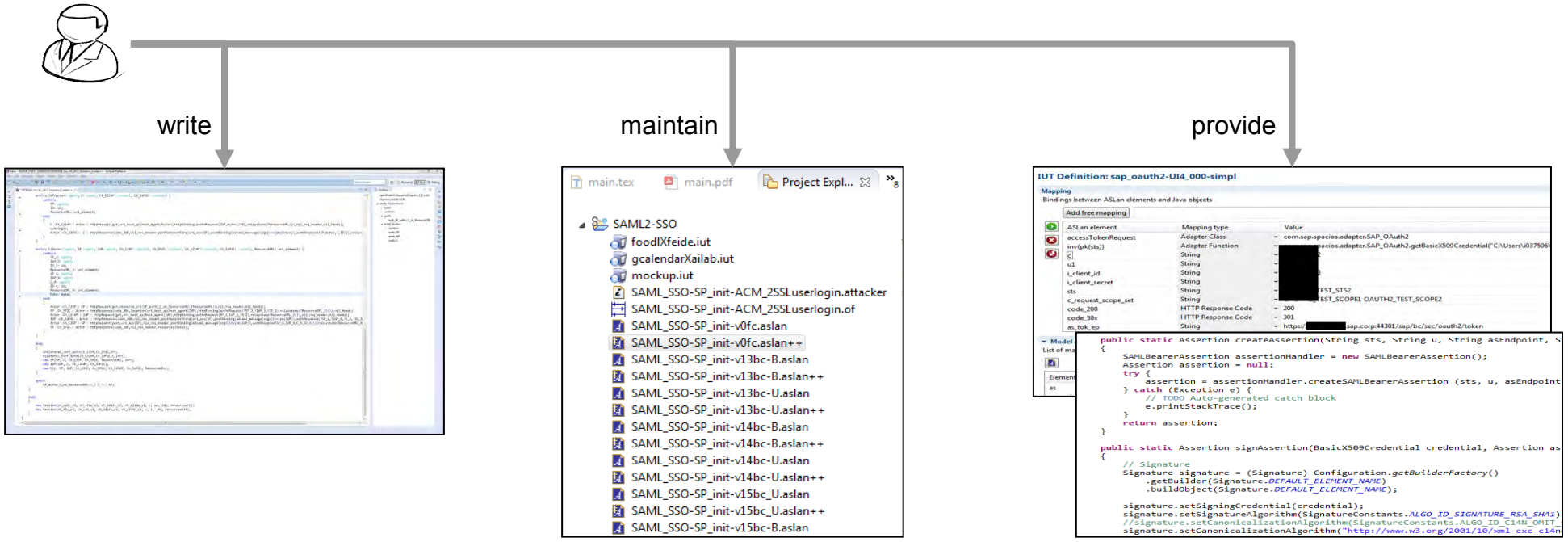


Input
Output

Motivations toward Episode 3

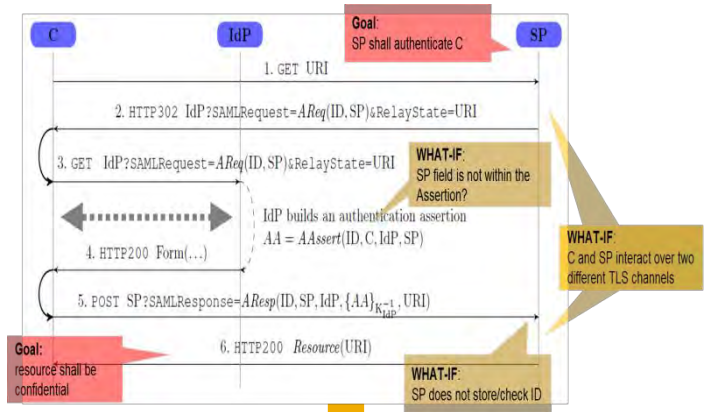
Can we improve **more** on the **usability/accessibility**?

- we have prototype tools **integrated** in a development environment (Eclipse) and able to **test** real systems
- still the tester has to **write** the formal specifications, **maintain** them, provide the **testing data/adapter**, …
- Cost (TCD) is still too high ☹



Our approach: historical view – Episode 3

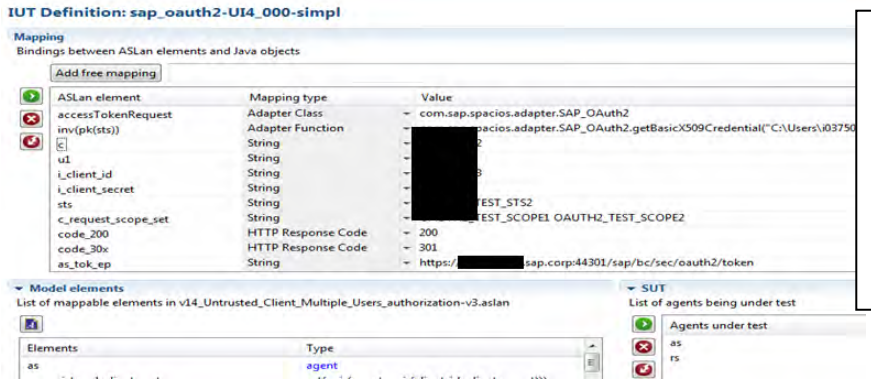
Specifications



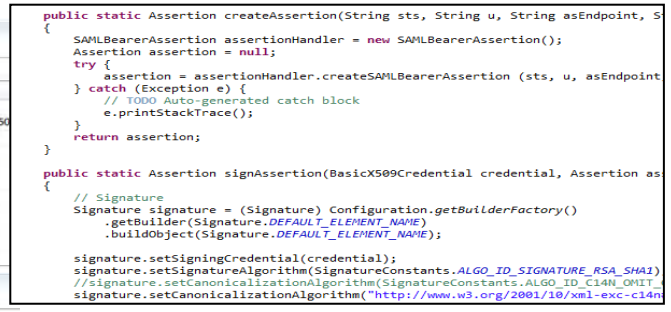
Formal models

Abstract threats

Testing data

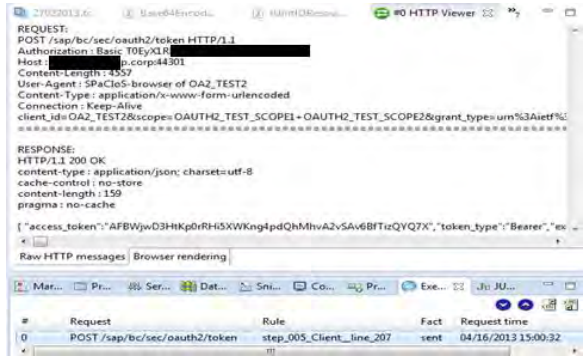


Adapter



```
public static Assertion createAssertion(String sts, String u, String asEndpoint, S
{
    SAMLBearerAssertion assertionHandler = new SAMLBearerAssertion();
    Assertion assertion = null;
    try {
        assertion = assertionHandler.createSAMLBearerAssertion (sts, u, asEndpoint
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return assertion;
}

public static Assertion signAssertion(BasicX509Credential credential, Assertion as
{
    // Signature
    Signature signature = (Signature) Configuration.getBuilderFactory()
        .getBuilder(Signature.DEFAULT_ELEMENT_NAME)
        .buildObject(Signature.DEFAULT_ELEMENT_NAME);

    signature.setSigningCredential(credential);
    signature.setSignatureAlgorithm(SignatureConstants.ALGO_ID_SIGNATURE_RSA_SHA1)
    //signature.setCanonicalizationAlgorithm(SignatureConstants.ALGO_ID_C14N_OMIT_
    signature.setCanonicalizationAlgorithm("http://www.w3.org/2001/10/xml-exc-c14n
```

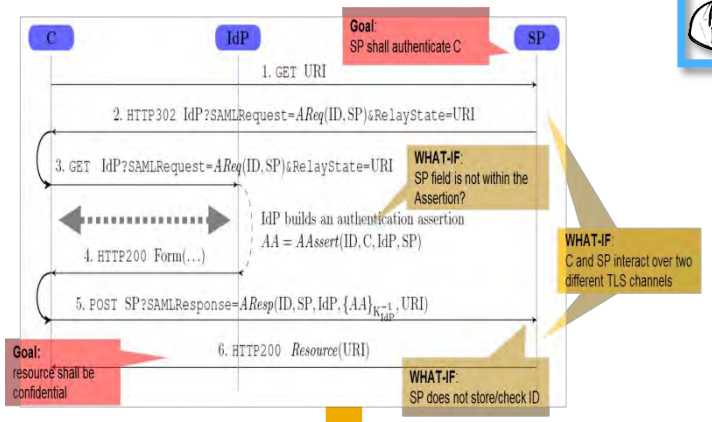
Concrete test-cases (generation and execution)



Input
Output

Our approach: historical view – Episode 3

Sequence diagram (with security annotations)



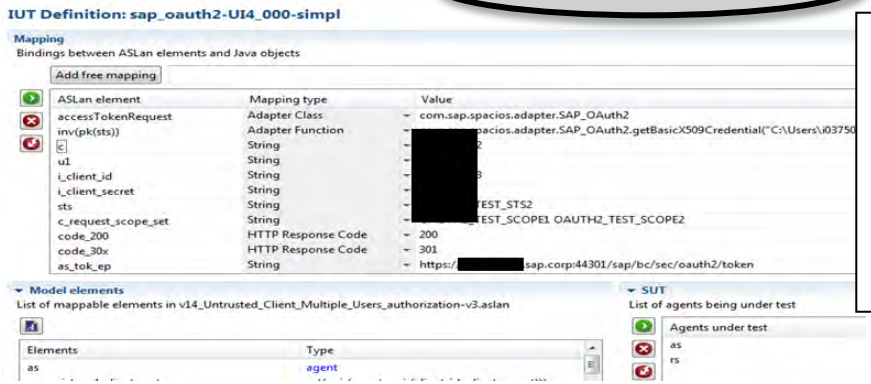
Formal models

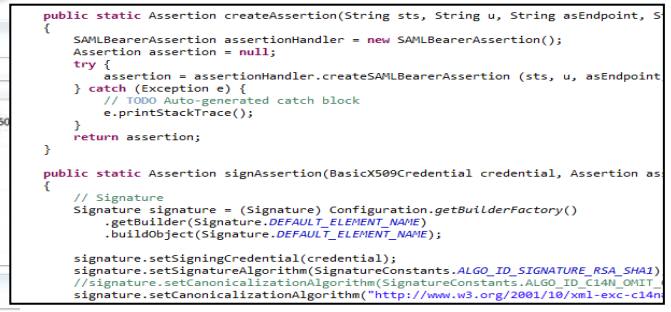
Abstract threats

Testing data

Catalog

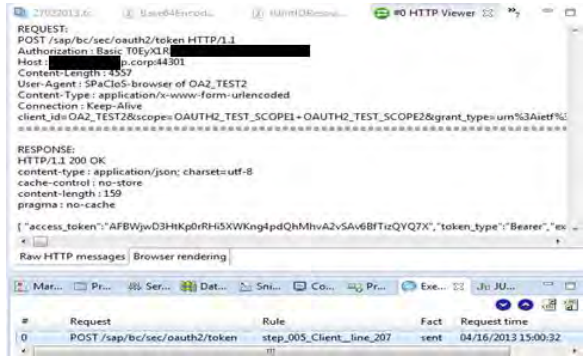
Adapter





```
public static Assertion createAssertion(String sts, String u, String asEndpoint, S
{
    SAMLBearerAssertion assertionHandler = new SAMLBearerAssertion();
    Assertion assertion = null;
    try {
        assertion = assertionHandler.createSAMLBearerAssertion (sts, u, asEndpoint
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
    return assertion;
}

public static Assertion signAssertion(BasicX509Credential credential, Assertion as
{
    // Signature
    Signature signature = (Signature) Configuration.getBuilderFactory()
        .getBuilder(Signature.DEFAULT_ELEMENT_NAME)
        .buildObject(Signature.DEFAULT_ELEMENT_NAME);

    signature.setSigningCredential(credential);
    signature.setSignatureAlgorithm(SignatureConstants.ALGO_ID_SIGNATURE_RSA_SHA1)
    //signature.setCanonicalizationAlgorithm(SignatureConstants.ALGO_ID_C14N_OMIT_
    signature.setCanonicalizationAlgorithm("http://www.w3.org/2001/10/xml-exc-c14n
```

Concrete test-cases (generation and execution)



Input
Output

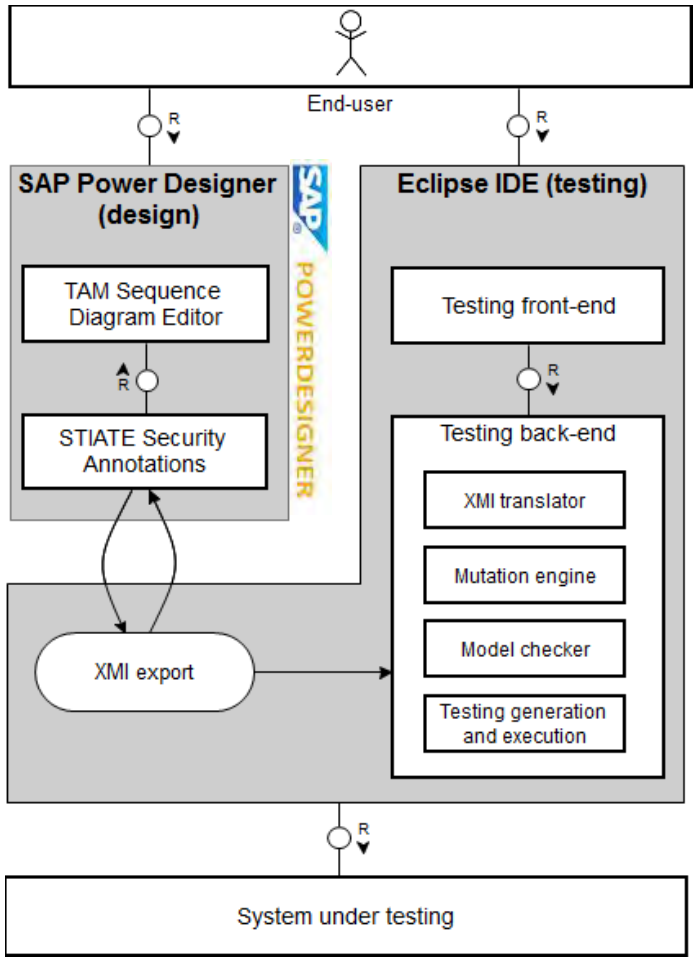
Architecture and status

Proof-of-concept

- integrated within SAP Power Designer
- Mobile payment commercial solution under security assessment

Potential targets

- Architects and development teams integrating a core security protocol
- Security consultants analyzing a customer proprietary protocol (e-payment)
- Standardization bodies designing protocols and reference implementations

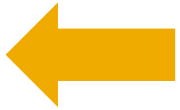


Outline

Multi-Party Web Applications (MPWAs)

Security Threat Identification and Testing: model-driven

Black-Box Security Testing: vulnerability-driven (two slides)



Black-Box Security Testing: vulnerability-driven (two slides)

work-in-progress

Observation

- many attack shares similarities
- capture similarities in executable artifacts

Challenges

- identify similarities
- basic ingredients for executable artifacts
- automation, accuracy, efficiency, …

Paper	Tech	Application(s)
Sec.4 of [22]	FV	SPs implementing Google's SAML SSO
Sec.5.2.1 of [36]	FV	SPs implementing OAuth 2.0 implicit flow-based Facebook SSO
Sec.IV.A.1 of [30]	BB	PayPal Payments Standard implementation in SPs using os-Commerce 2.3.1 or AbanteCart1.0.4
Sec.V.A of [33]	WB	SPs implementing CaaS solutions of 2Checkout, Chrono-Pay, PSiGate and Luottokunta (v1.2)
Sec.IV.A.2 of [30]	BB	PayPal Express Checkout implementation in SPs using Open-Cart 1.5.3.1 or TomatoCart 1.1.7
Sec.4.2 of [34]	BB	SPs implementing OAuth 2.0 implicit flow-based Facebook SSO
Sec.6.2 of [23]	BB	developer.mozilla.com (SP) implementing BrowserID
Sec.V.C of [24]	FV	CitySearch.com (SP) using Facebook SSO (OAuth 2.0 Auth. Code Flow)
Sec.4 of [21]	FV	SPs implementing Google's SAML SSO
Bug 2 of [1]	M	Github (TTP) implementing OAuth 2.0 Authorization Code flow-based SSO

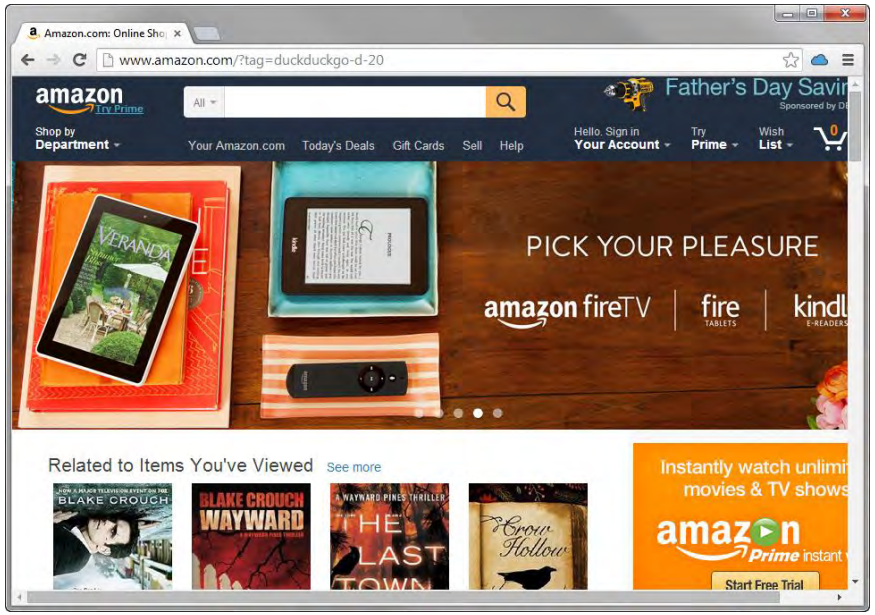
Legend: FV: formal verification; BB: black-box; WB: white-box; M: manual inspection

Proof-of-concept

- integrated with off-the-shelf pentest tool
- available, but cannot be shared yet (need to be published first)
- successful demonstrated against both SSO and Online Shopping scenarios (Cash-as-a-service protocols)

Industrial exploitation?

- Accessibility / Usability + Automation / Integration
  → reasonable
- Business case
  → developer can run our tool
  → security expert can assist in case of findings
- Accuracy / Coverage
  → complete? sound?
  → how we position wrt tools available in the market?
- Cost-benefit ratio



Final remarks

MPWAs is a core business area and security is one of the top challenges (→ Micro-services trend?)

A lot of active research from which industry can take a lot: cost-benefit considerations, though

Investigating two approaches in this area, targeting different phases of the development lifecycle

Thank you

Contact information:

Luca Compagna
luca.compagna@sap.com

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

© 2015 SAP SE oder ein SAP-Konzernunternehmen. Alle Rechte vorbehalten.

SASS15

The many faces of fuzzing

Radek Domanski, Huawei

Abstract:

Fuzzing techniques have been in use for many years as a method to find application bugs. The fuzzing concept has evolved from single random input generation tools to large and complex vulnerability discovery platforms. Nowadays, fuzzing is a fundamental method for security testing of applications, network protocols and structured data parsers. However, due to its own shortcomings, fuzzing methodology is still subject to active research by specialists in the security and testing communities. Due to many variations of fuzzing, it is important to know which approach is best to use for a specific target. In my talk, I will discuss various approaches together with their strengths and weaknesses.

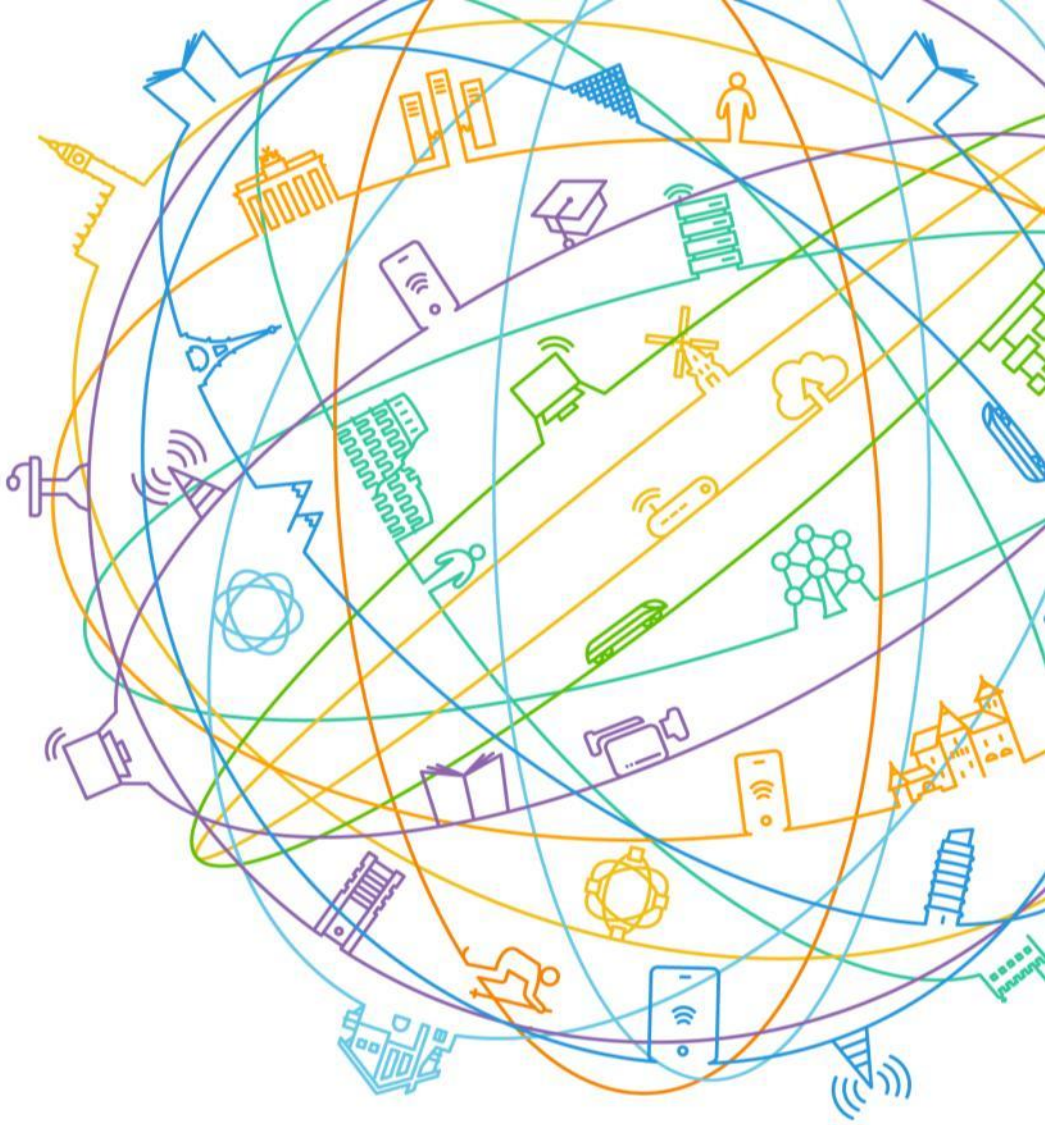
Vita:

Radek Domanski works at Huawei Technologies in the European Research Center in Munich. His research involves methods and techniques that can improve quality of product security testing. He has many years of hands-on experience of security testing focusing on practical security attacks scenarios, especially in the telecom environments. His personal interests include fuzzing, reverse engineering, applications exploitation and systems security.

Fraunhofer FOKUS

SASS '15

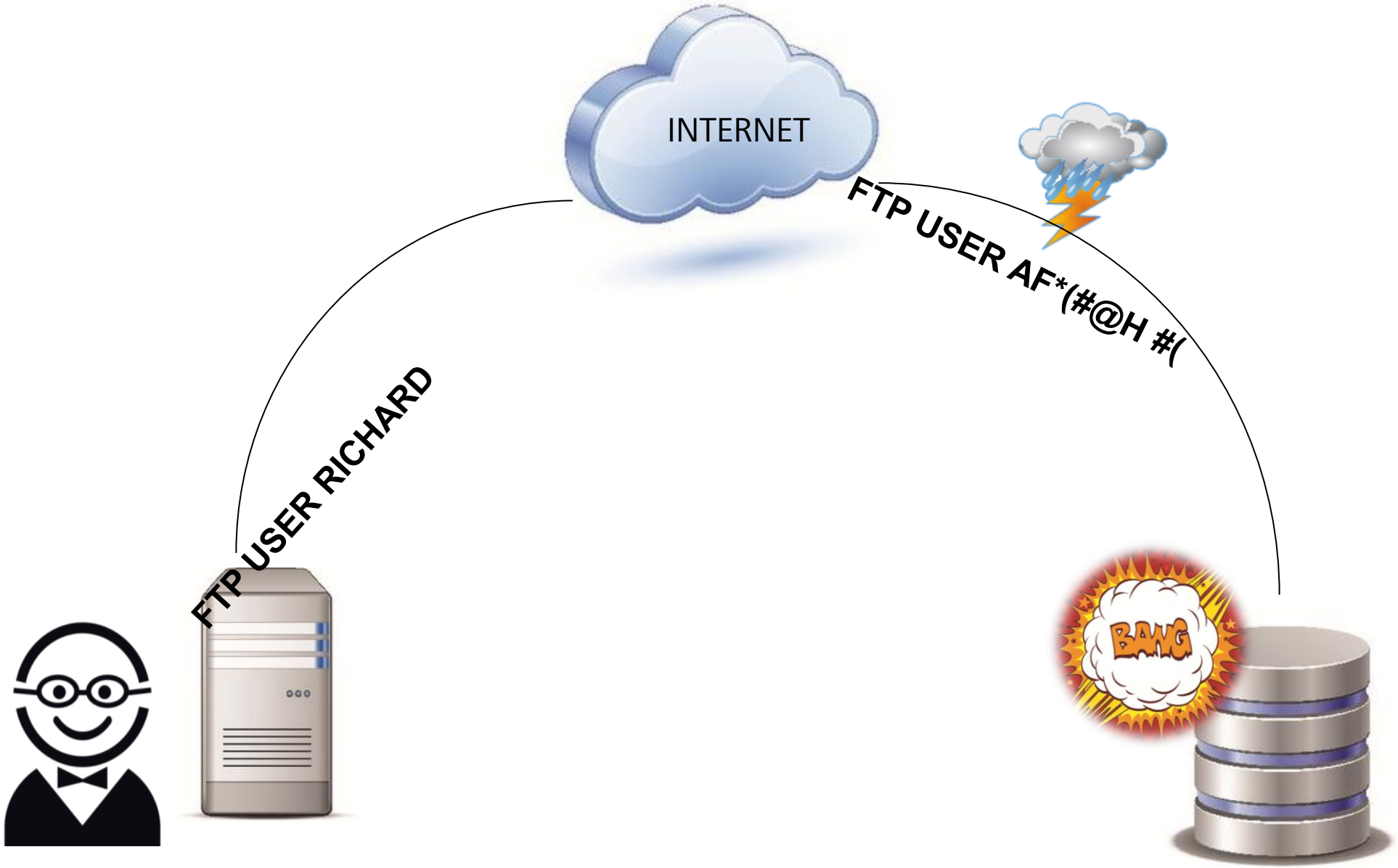
The many faces of fuzzing



SASSI Workshop 2015

Radek Domanski

1990


INTERNET
FTP USER RICHARD
FTP USER AF*(#@H #(
BANG

Research Project

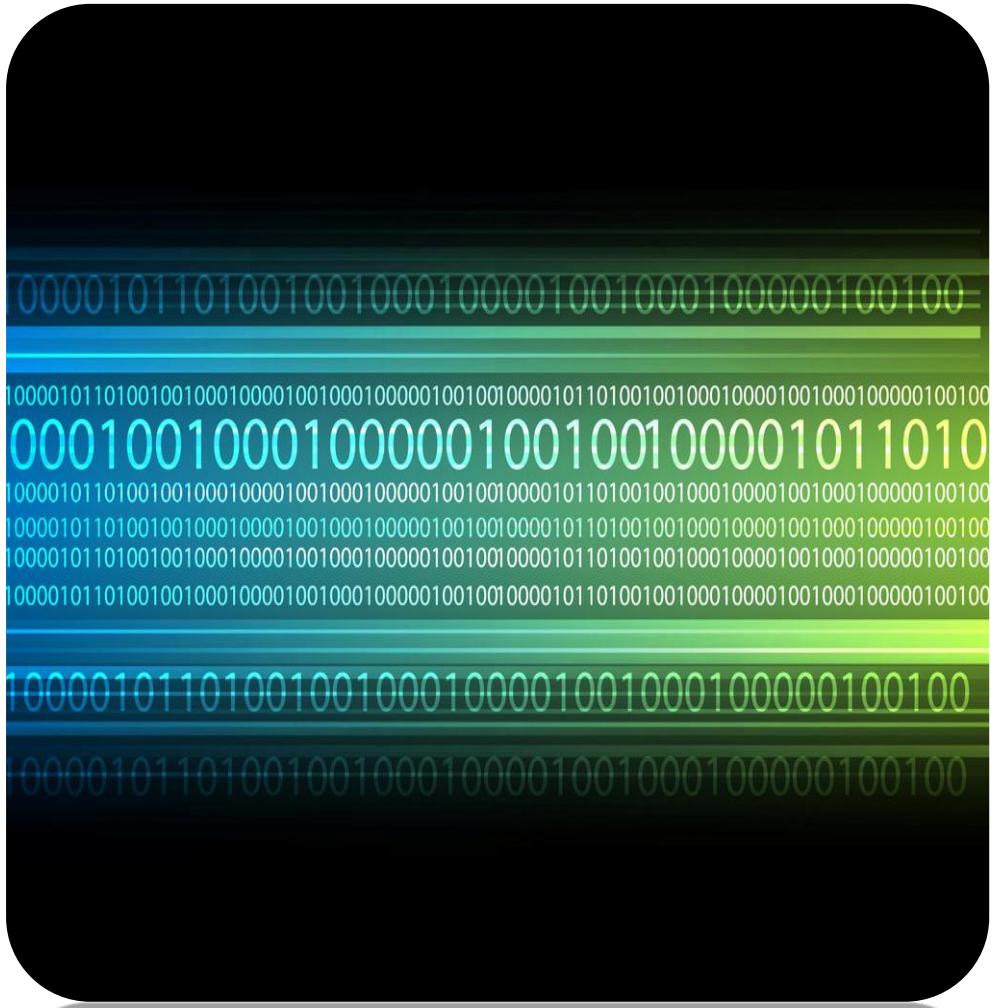
1. Construct a program to generate random characters, plus a program to help test interactive utilities

2. Use these programs to test a large number of utilities on random input strings to see if they crash

3. Identify the strings that crash these programs

4. Identify the cause of the program crashes and categorize the common mistakes that cause these crashes

Reference: “An Empirical Study of the Reliability of UNIX Utilities” (Miller, Fredriksen, So)



$ fuzz 100000 –o outfile | deqn

"The program fuzz is basically a generator of random characters"

"While our testing strategy sounds somewhat naïve, its ability to discover fatal program bugs is impressive"

Fuzz Results

Almost 90 different utility programs on seven versions of UNIX were tested

adb	cat	grep	sql
as	cb	head	telnet
awk	cc	mail	tr
bc	compress	make	vi
bib	diff	sed	wc
calendar	ftp	sort	(…)

More then 24% of those programs crashed

- Bounds checking
- Not checking return codes
- Improper usage of dangerous functions
- Subprocesses
- Bad error handling
- Signed characters
- Race Conditions

Reliability vs Security

"The ability to overflow an input buffer is also a potential security hole, as shown by the recent Internet worm"

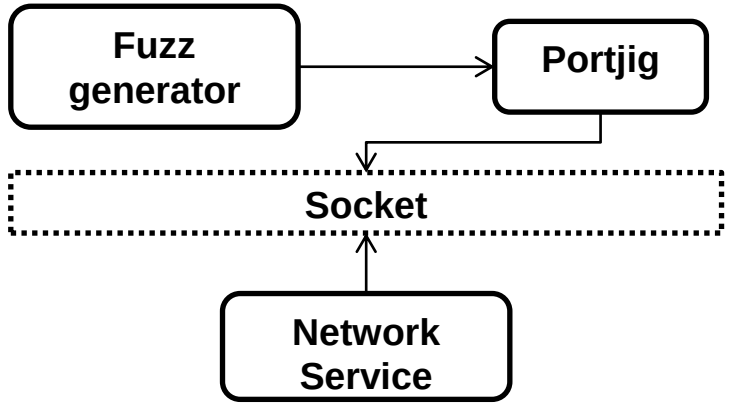
"The suggestion on using random testing to help find security holes is due to one of the anonymous referees"

Retest and new approaches

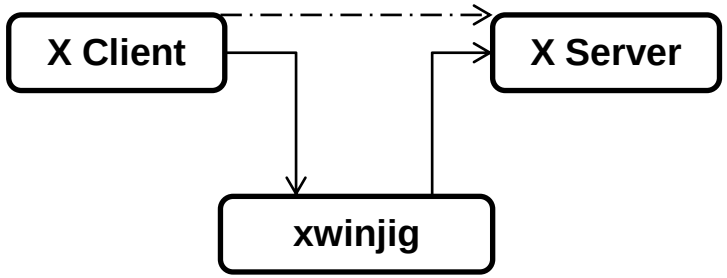
Simple fuzzing

$ fuzz 100000 –o outfile | sort

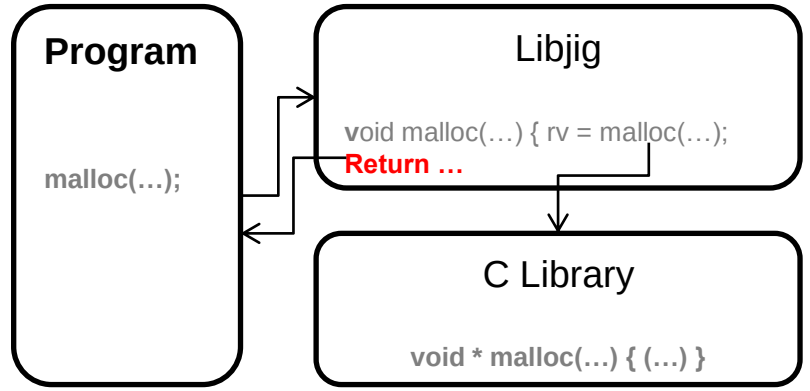
Network Fuzzing



X Window Application Fuzzing

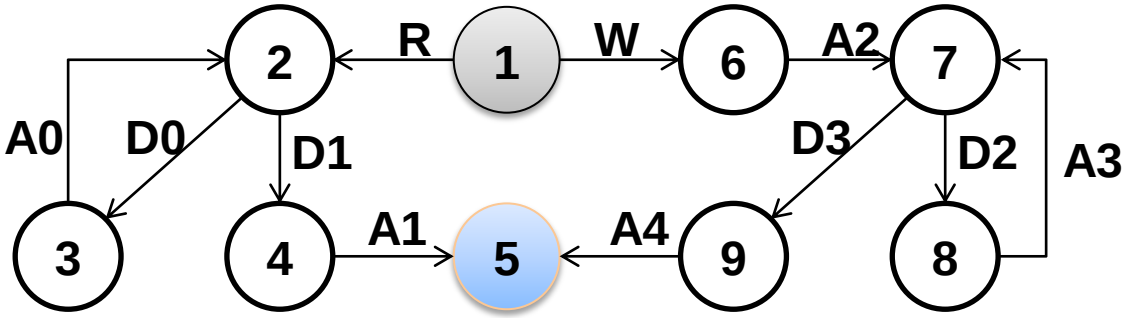


Memory allocation call



Models & Fault Injection

- Improve robustness testing by creating models of protocols, client and server
- Increase test coverage



$$R(D0\ A0)^n\ D1\ A1 + W\ A2(D2\ A3)^m\ D3\ A4$$

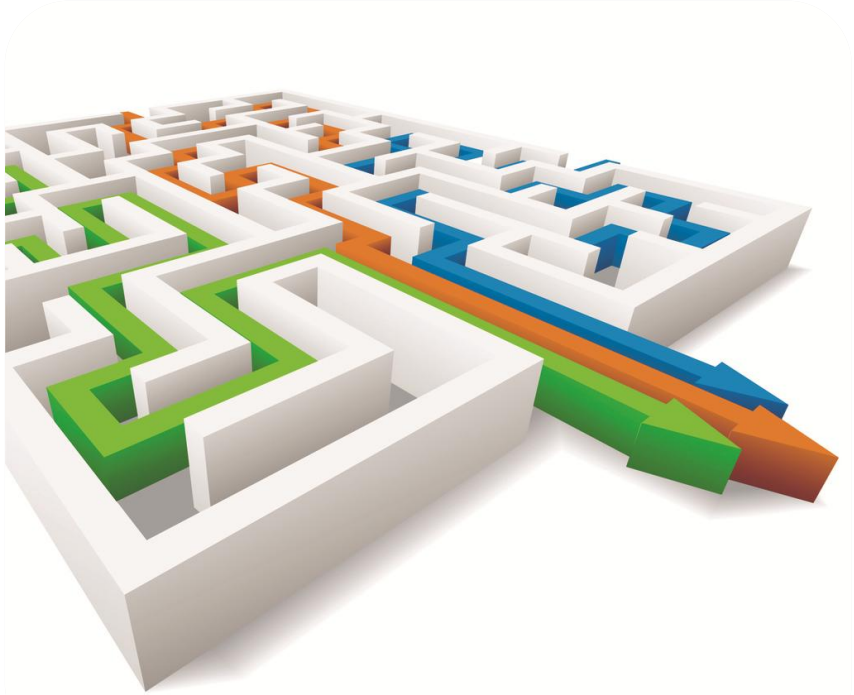
TFTP Write operation

Opcode	Filename	0	Mode	0
^\x00\x01	([^\x00]*)	\x00	([^\x00]*)	\x00$

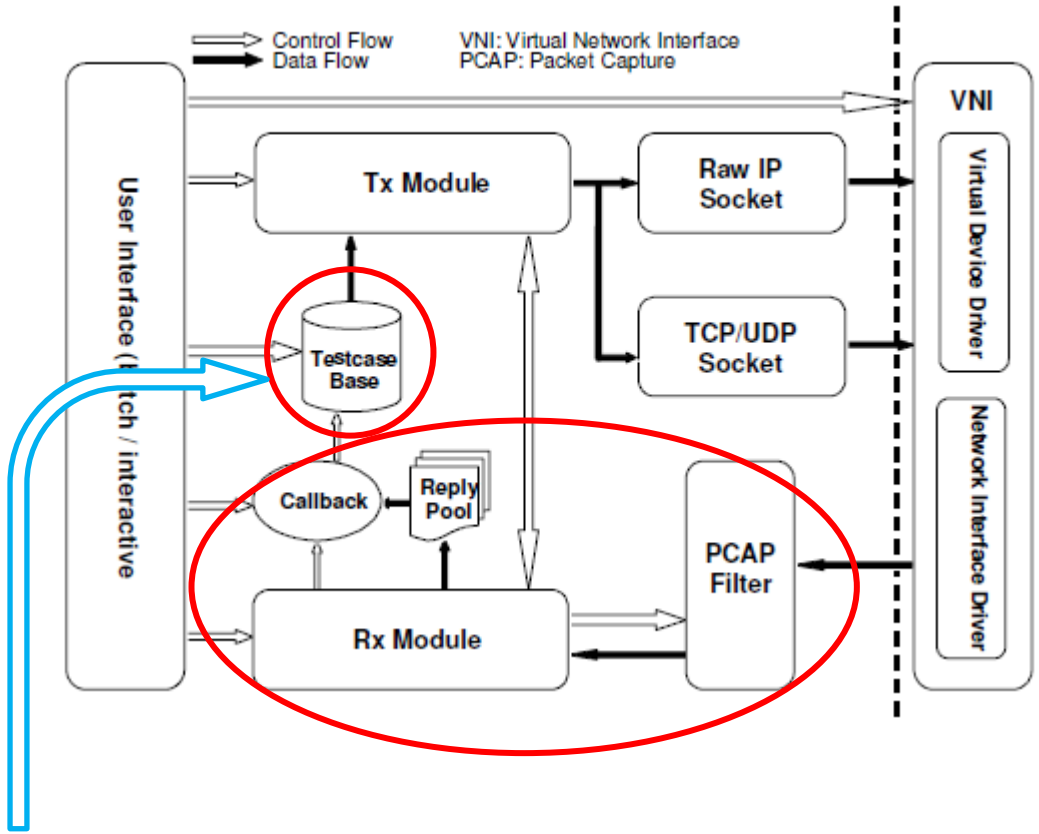
R – Client request a file read
D0 – Server sends a 512-octet data block
A0 – Client acknowledges the previous block
D1 – Server sends the final block, less then 512 octet
A1 – Client acknowledges the final block
W – Client requests a file write
A2 – Server acknowledges the readiness to receive
D2 – Client sends a 512 octet data block
A3 – Server acknowledges the previous block
D3 – Client send the final block, less then 512 octet
A4 – Server acknowledges the final data block

Reference: PROTOS Project

Models



“Complicated protocols often require strict conditions to enter new states. For example, routing protocols like OSPF and BGP would accept routing updates only after the peers established adjacency by exchanging hello messages.”

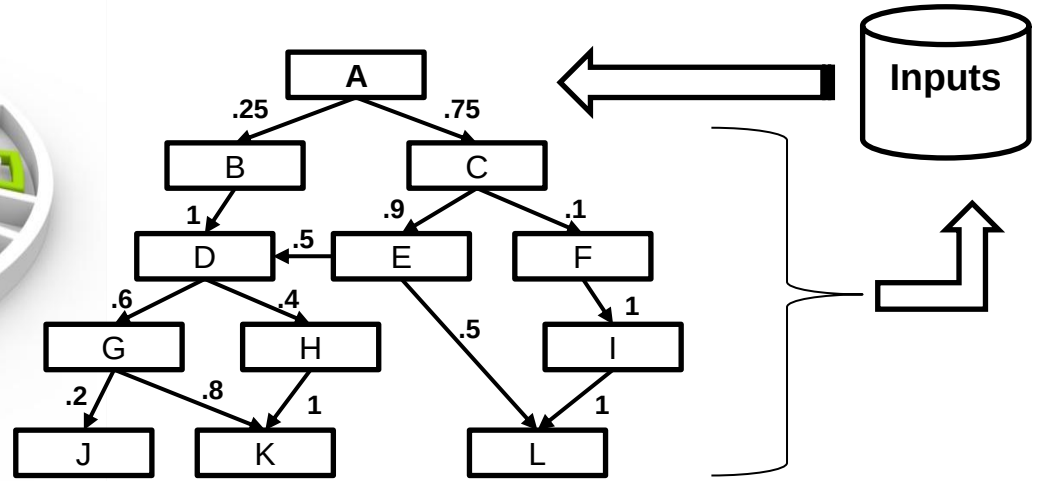


Mutate message, syntax, content and sequence

Reference: “Integrated TCP/IP Protocol Software Testing for Vulnerability Detection” (Xiao, Wang)

Guided Execution

Added intelligence to fuzzing process and input selection method



Build execution flow diagram
Select potentially vulnerable block
Mark all possible transitions leading to selected block
Mark all other transitions as "reject" state
Generate round of random input
At each transition calculate probability of reaching the state for the given round
Reject inputs that led to "reject" state
Select inputs that are the best candidates to reach desired state
Mutate selected inputs for another round

		2 bytes	string	1 byte	string	1 byte
	Generation	**Opcode**	**Filename**	**0**	**Mode**	**0**
1.	0	2326				
2.	1	626F637465741B7B6225				
3.	51	0005	367D			
4.	72	0002	36060628791E32			
5.	78	0002	36060128	00	0A2A3606	
6.	111	0001	NULL	00	0A057C0561	
7.	393	0002	187566	00	266F6374657464	00
8.	547	0001	2E027D1C02006F63746574	00	6E6574617363696964	00

Reference: "Automated vulnerability analysis: Leveraging control flow for evolutionary input crafting" Sparks, Embleton

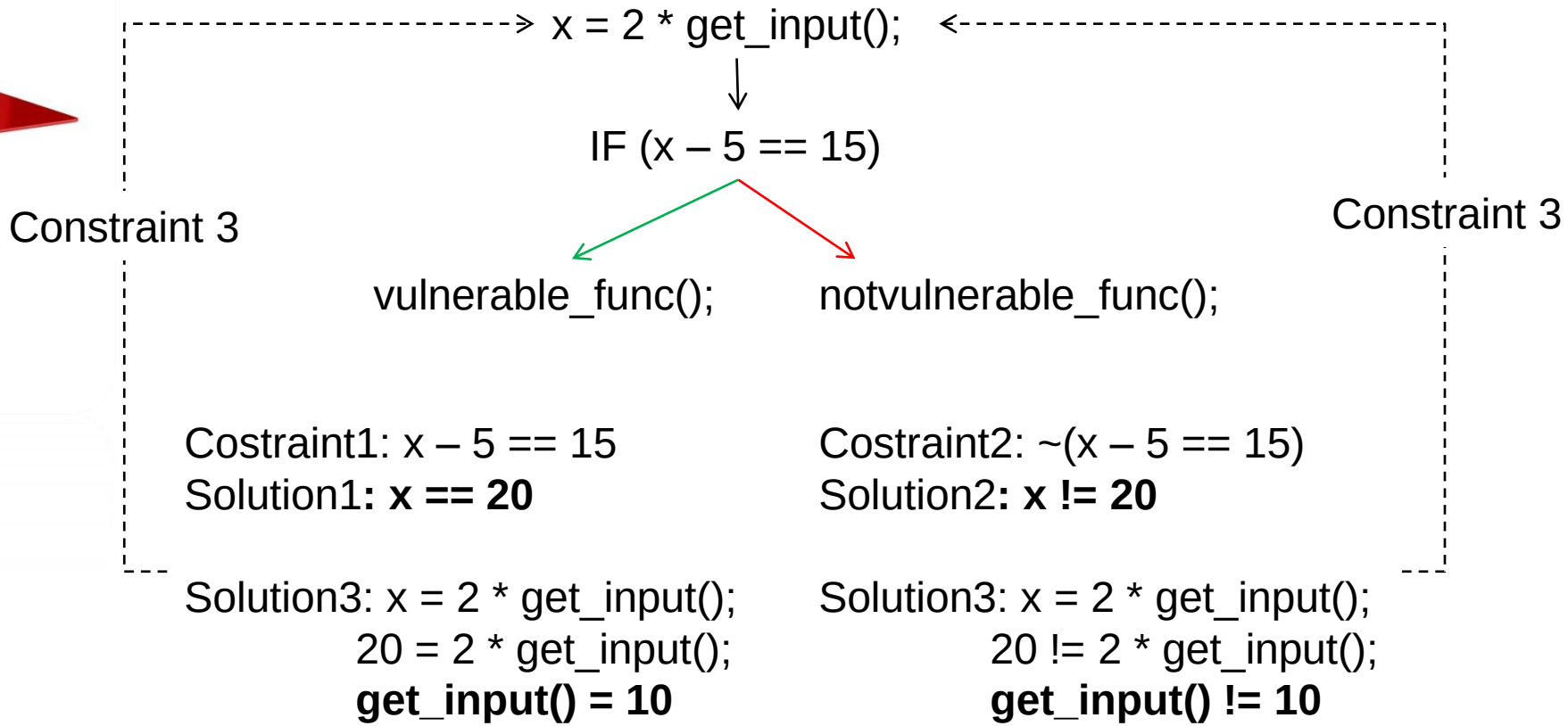
Overcoming code coverage constrains

How to visit all possible paths of the executable?

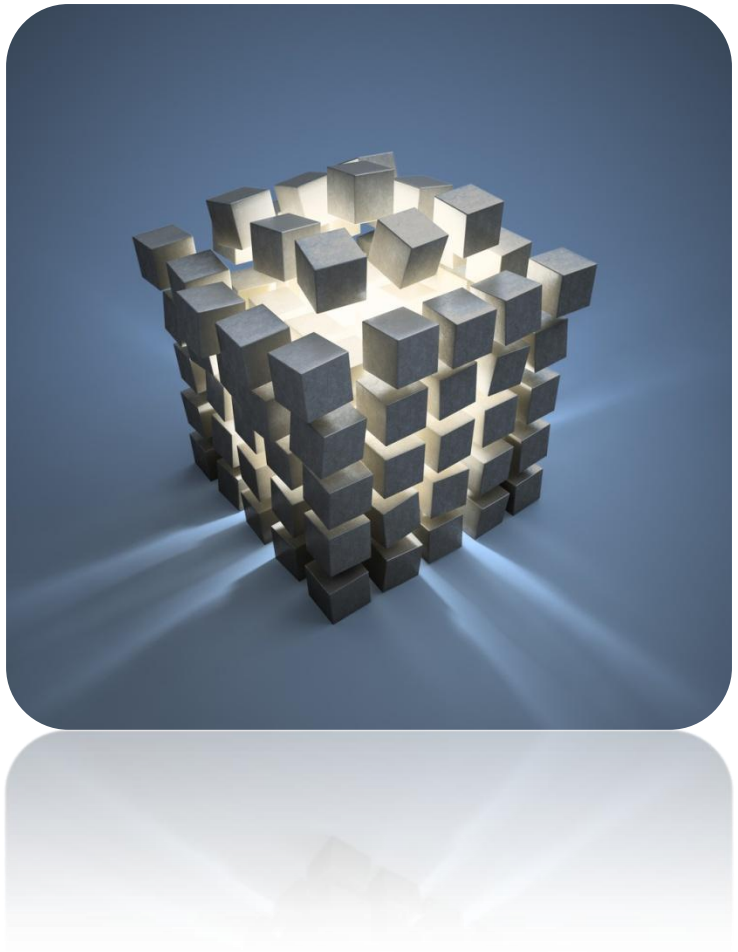
```
X = 2 * get_input();

If (x-5 == 15)
        vulnerable_func();
Else
        notvulnerable_func();
```

Only 1 out of 2^32 possible inputs will lead to the execution of vulnerable_func(). (on 32bit architecture)

Symbolic execution & path constraint solver



White Box Fuzzing



White box fuzzing tools!

KLEE – LLVM based

SAGE – x86 binary

SAGE is used in Microsoft as a core tool for security testing.
It runs 24/7 since 2008 on over 100 machines
Although SAGE is involved last (after static code analysis and other black box testing) it found over 1/3 off all bugs in Windows 7

Not everything is solved yet!

Imprecision in symbolic execution
path explosion
input dependent loops
floating-point instructions
etc.

Automatic Exploit Generation

How to reduce false positives?

```
char dst[10], src[12];
strncpy(dst, src, sizeof(src));
```

Is it a bug?
Yes, on a source code level.
No, on a run time level!*

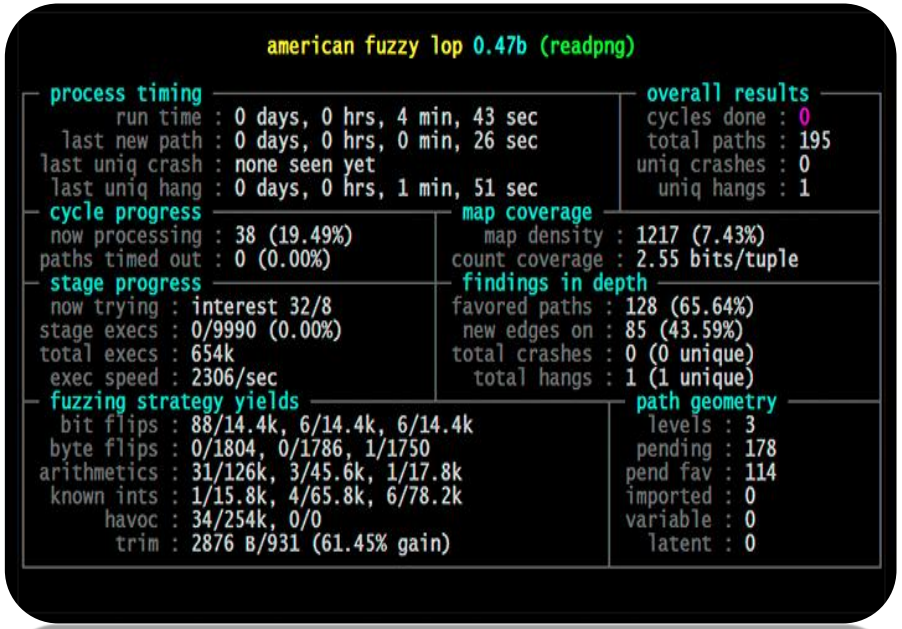
* A lot depends on the compiler. Modern compilers might page-align declared buffers making structures 16 bytes effectively, with programmer not even realizing.

How to reduce false positives?

Prove that the bug is exploitable – that's the P1 bug.

"After the build, we run our tool, AEG, and get a control flow hijacking exploit in less than 1 second. Providing the exploit string to the iwconfig binary, as the 1st argument, results in a root shell." – Carnegie Mellon University

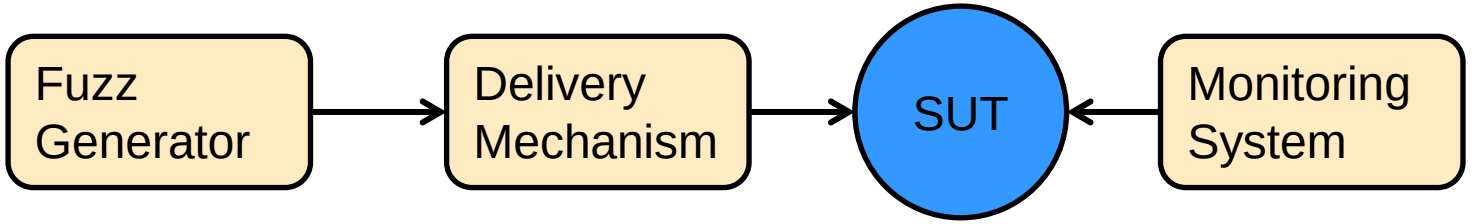
American Fuzzy Lop



“Compared to other instrumented fuzzers, afl-fuzz is designed to be practical:

- **it has modest performance overhead**
- **uses a variety of highly effective fuzzing strategies and effort minimization tricks**
- **requires essentially no configuration, and seamlessly handles complex**
- **real-world use cases - say, common image parsing or file compression libraries.”**

Fuzzers Classification



- mutative
- generative
- template based
- block based
- grammar based
- heuristic based

- files
- environment vars.
- command line
- API
- network interfaces
- OS events

- local monitoring system
- remote monitoring system

Which fuzzer and method should you use?

Thank you

www.huawei.com

SASS15

Combining Security Risk Assessment and Security Testing based on Standards

Jürgen Großmann, Fraunhofer FOKUS

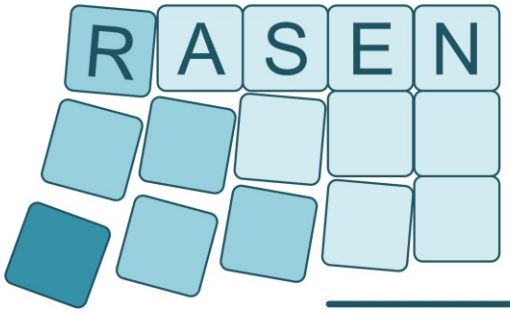
Abstract:

Managing cyber security has become increasingly important due to the growing interconnectivity of computerized systems and their use in society. A comprehensive assessment of cyber security can be challenging as its spans across different domains of knowledge and expertise. For instance, identifying cyber security vulnerabilities requires detailed technical expertise and knowledge, while the assessment of organizational impact and legal implications of cyber security incidents may require expertise and knowledge related to risk and compliance. Standards like ISO 31000 and ISO/IEEE 29119 detail the relevant aspects of risk management and testing and thus provide guidance in these areas. However, both standards do not cover the explicit integration between security risk assessment and security testing. We think however, that they provide a good basis for that. In this paper we show how ISO 31000 and ISO/IEEE 29119 can be integrated to provide a comprehensive approach to cyber security which covers both risk assessment and testing.

Vita:

As a member of the Competence Center "System Quality Center" (SQC) Jürgen Großmann is responsible for validation, verification and testing projects on next generation networks and software technologies for embedded systems. He is an expert on model-based development, model driven testing as well as in security engineering and security testing. Jürgen Großmann has experiences in numerous standardization activities for various standardization bodies, including OMG, ETSI, ASAM and AUTOSAR.

Fraunhofer FOKUS



Compositional Risk Assessment and Security Testing of Networked Systems

Jürgen Großmann
(FhG Fokus)

Combining Security Risk Assessment and Security Testing based on Standards

SASSI Workshop

Berlin, 2015-09-16

SEVENTH FRAMEWORK PROGRAMME

Source: https://www.flickr.com/photos/maerskline/8432240103/in/photostream/
License: https://creativecommons.org/licenses/by-sa/2.0/

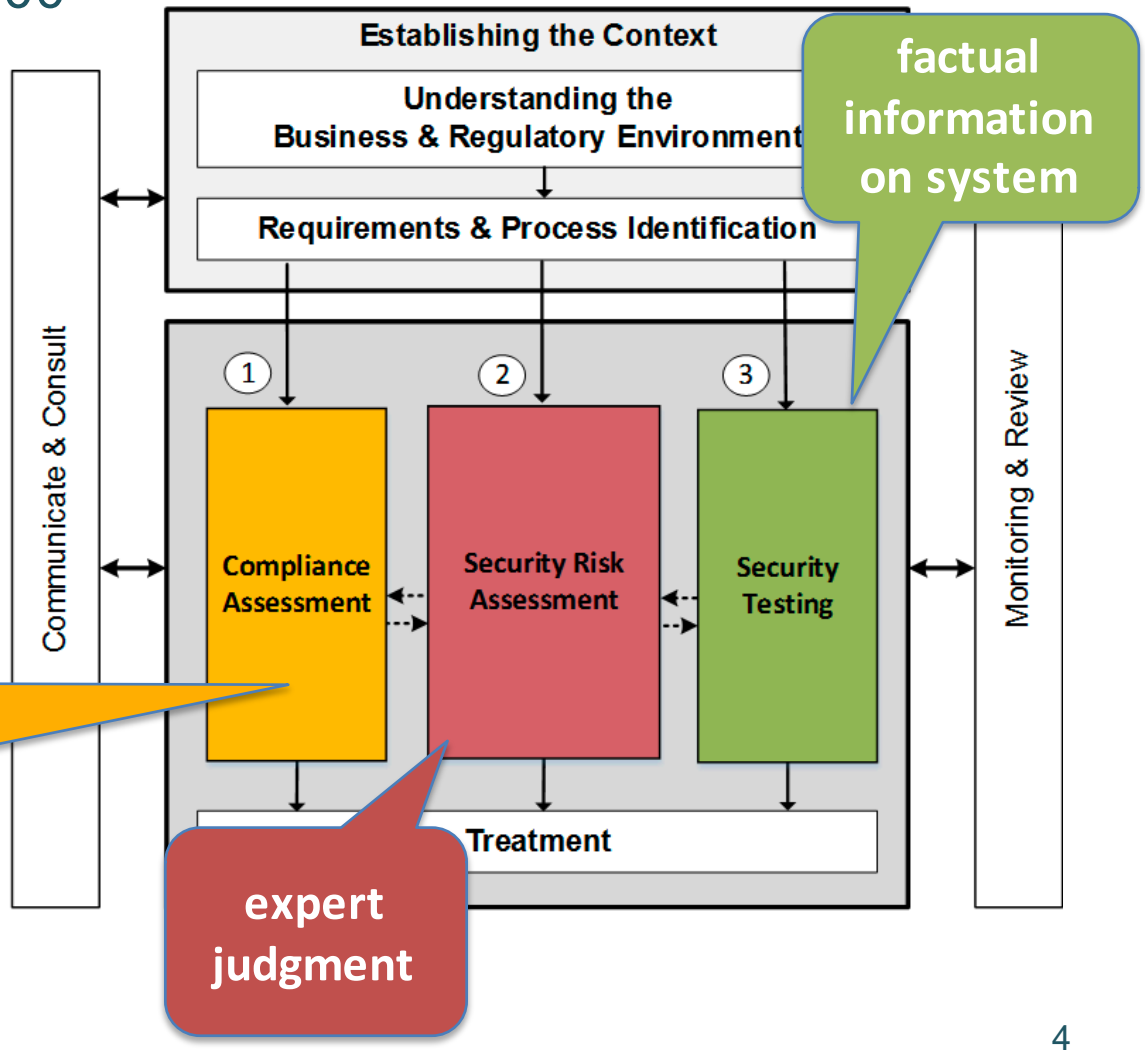
FP7 project **RASEN** (RASEN - 316853)

Developing methods and tools to support **security assessments** for **large-scale networked infrastructures** by considering:

1. technical aspects
2. legal and regulatory aspects
3. uncertainty and risk

The RASEN method for security testing, risk & compliance assessment

- Conforms to ISO/IEC 31000
- Integrates risk assess-ment, compliance assessment and security testing in a meaningful manner
- Addresses management aspects as well as assessment aspects



RASEN

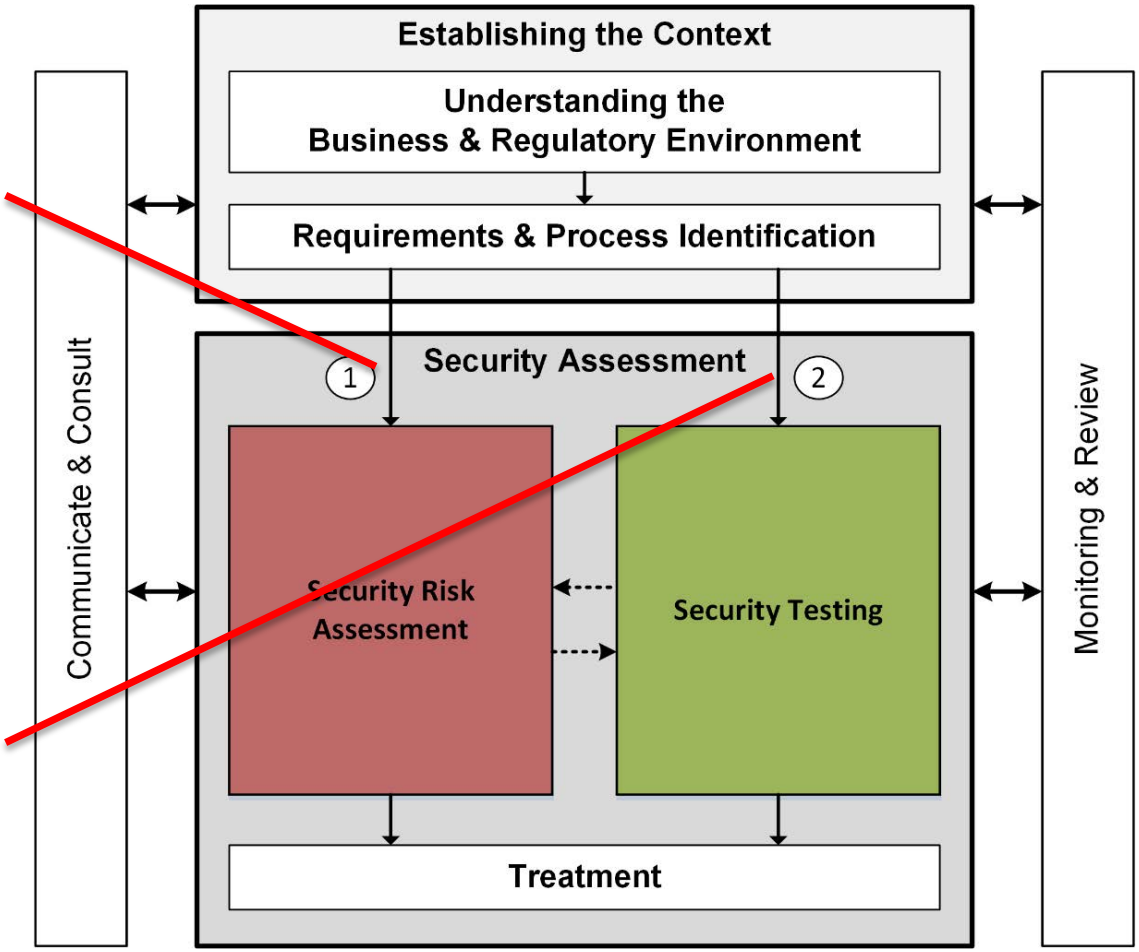
Two main workstreams: Risk assessment and security testing

A test-based security risk assessment process (1)

- starts with the risk assessment
- is used to optimize security risk assessment with empirical data coming from test results or compliance issues.

A risk-based method for security testing (2)

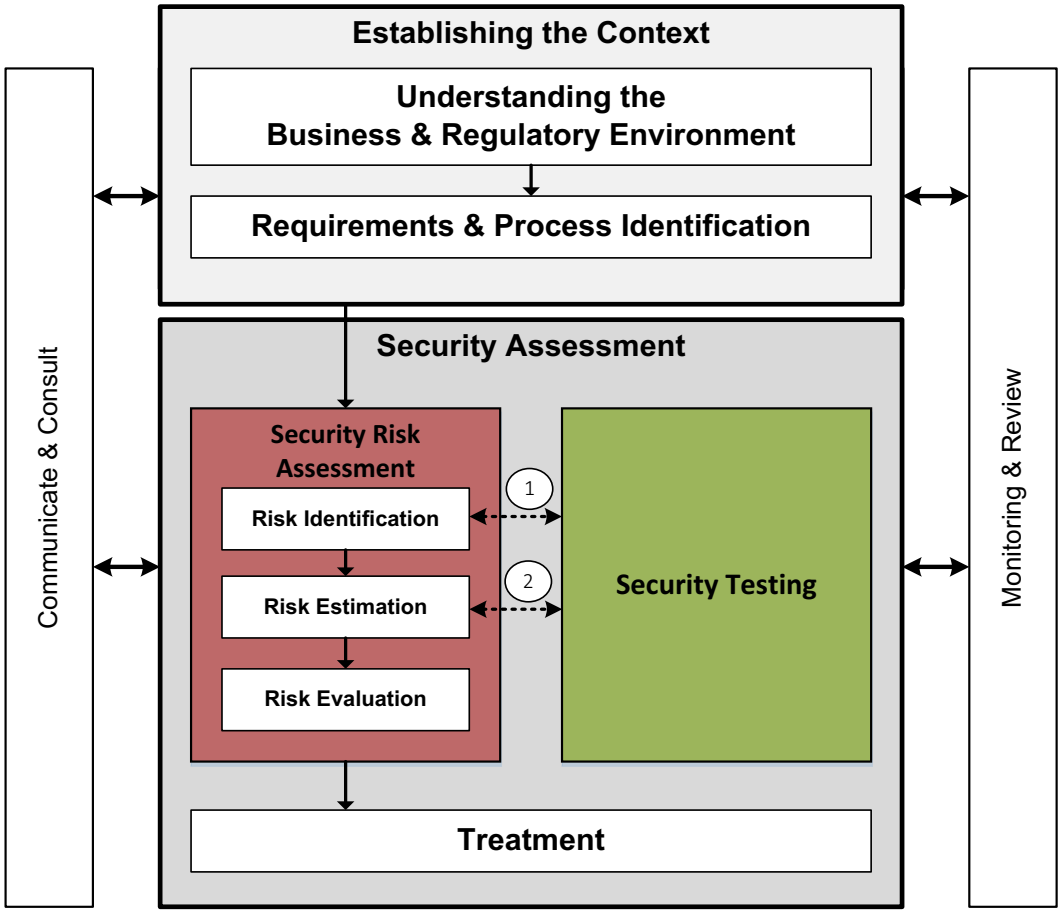
- starts with the identification of issues by security testing or compliance assessment
- focus the compliance and security testing resources on the areas that are most likely to cause concern
- building and prioritizing the compliance measures or testing program around these risks.



Workstream 1: Test-based security risk assessment

1. Test-based risk identification
2. Test-based risk estimation

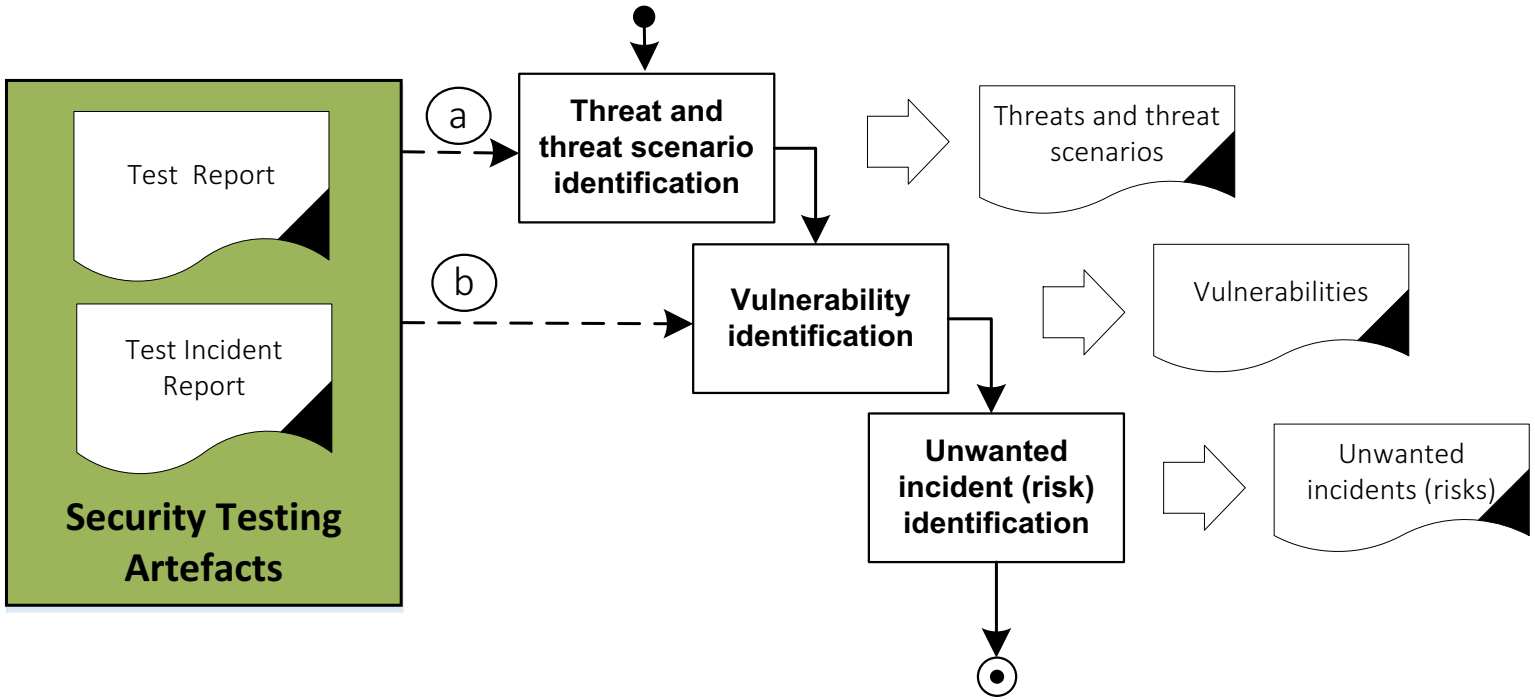
- **Basic idea:** improve risk assessment activities through facts from testing



Test-based risk identification

a) Test-based attack surface analysis

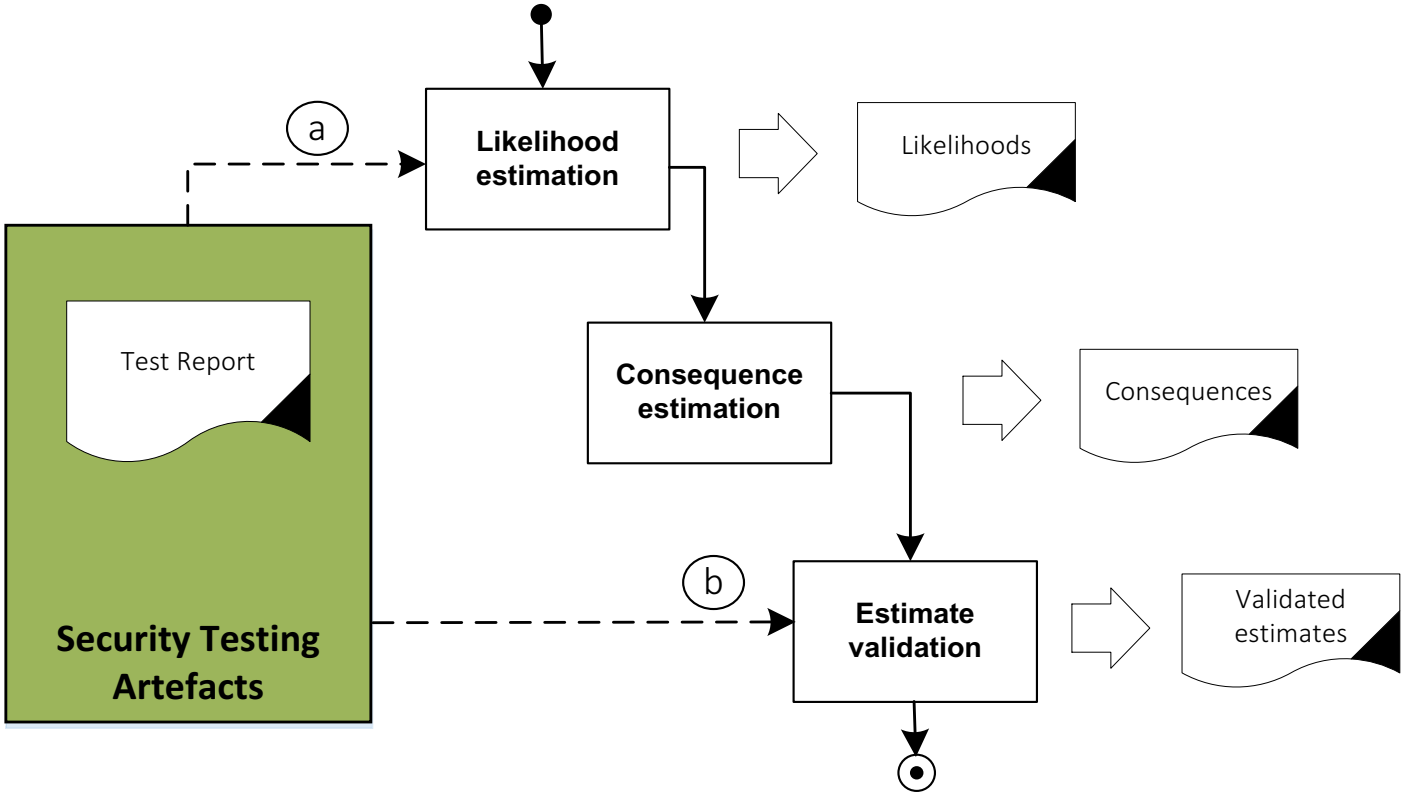
b) Test-based vulnerability identification



Test-based risk estimation

a) Test-based likelihood estimation

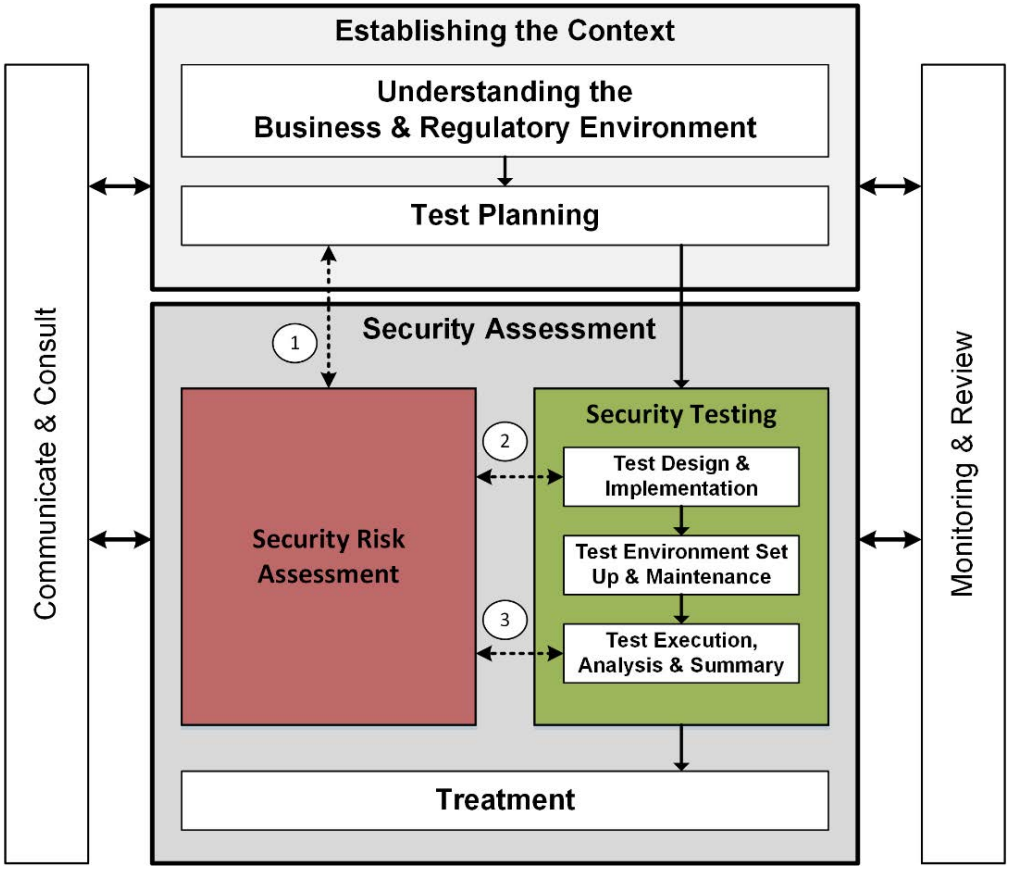
b) Test-based estimate validation



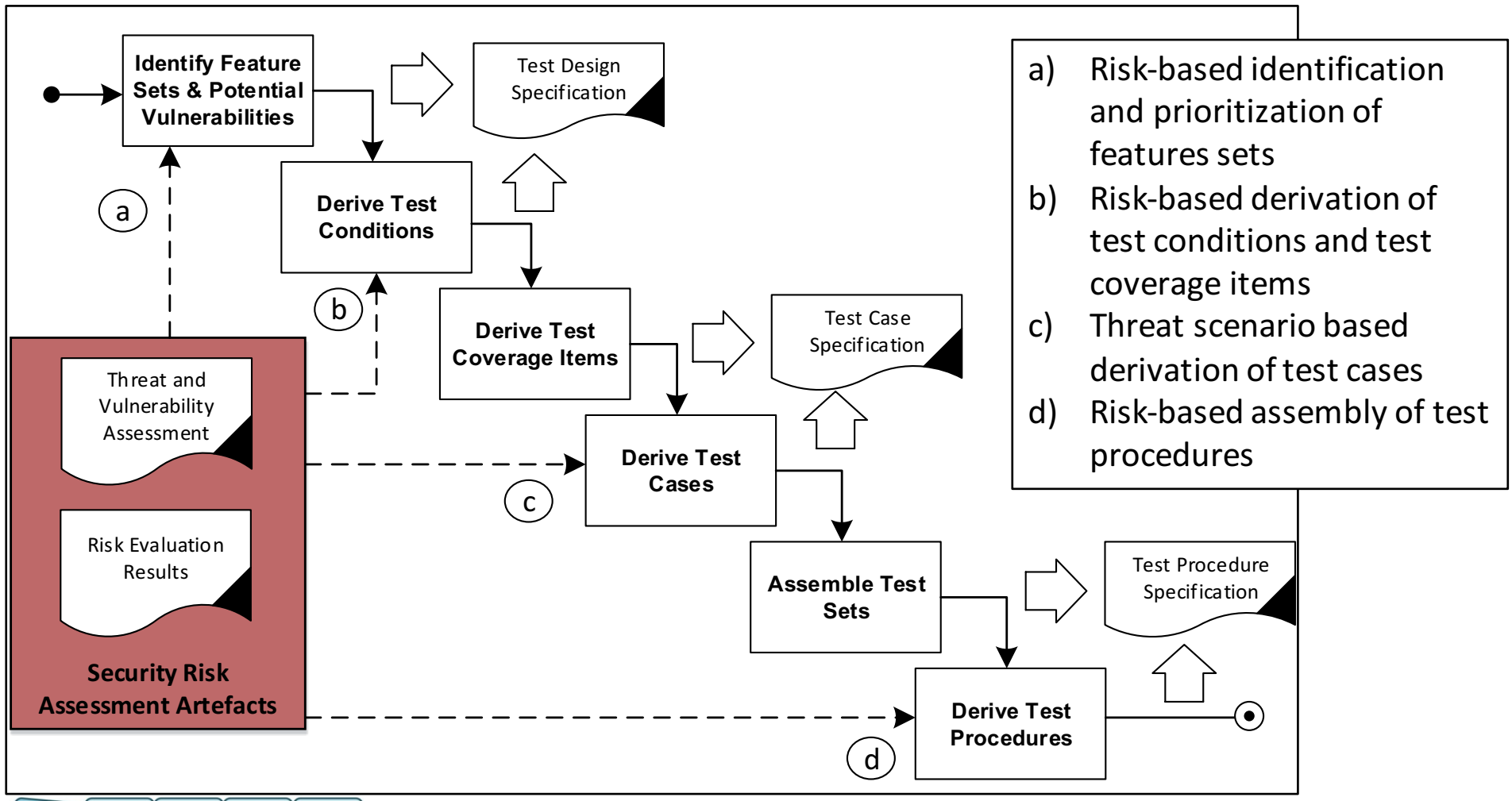
Workstream 2: Risk-based security testing compliant to ISO 29119

1. Risk-based security test planning
2. Risk-based security test design & implementation
3. Risk-based test execution, analysis & summary

- **Basic idea:** focus testing activities on high risk areas



Risk-based security test design and implementation



Activities are specified in detail to provide guidance

Identifier

Environment

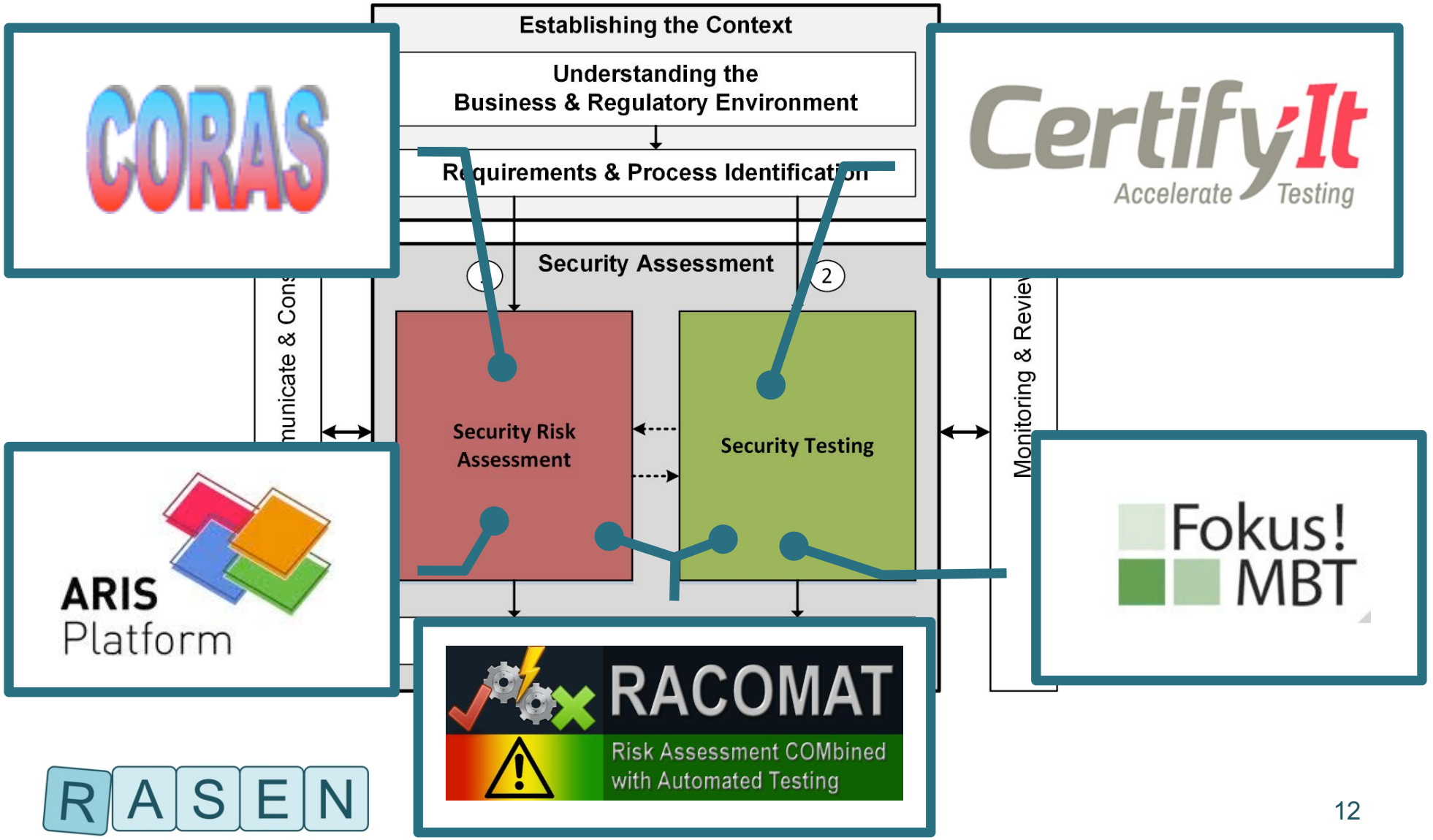
Pre-and Postconditions

Scenario

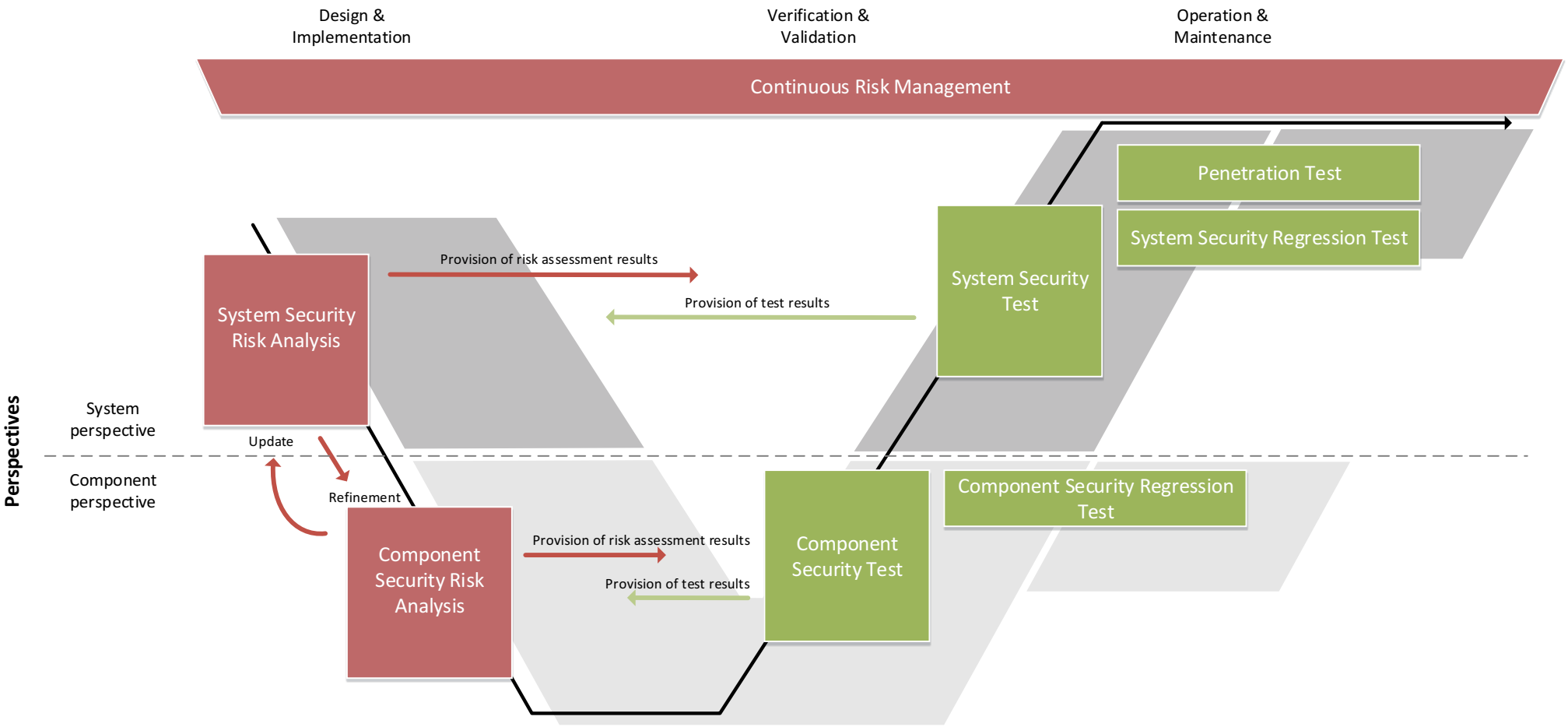
I/O

Name	**Risk-based identification and prioritization of features sets (a)**
Actors	Security Tester (ST), Security Risk Analyst (SRA)
Tools	Test Specification Tool (STST), Security Risk Assessment Tool (SRAT)
Precondition	Security relevant features are documented and the security risk assessment is available
Postcondition	Security relevant features to be tested are grouped with respect to potential vulnerabilities and threat scenarios.
Scenario	1. The Security Tester should identify testable security relevant features that need to be covered by security testing. The security tester classifies the security relevant features by grouping them to form feature sets that each addresses exactly one threat scenario and/or one vulnerability. 2. The Security Tester should prioritize the security relevant feature sets using the risk levels that are associated with the threat scenario and/or vulnerabilities. 3. The Security Tester should document the relations between security relevant feature sets and their associated threat scenarios and/or vulnerabilities (maintain traceability).
Data exchanged/ processed	**In:** *Vulnerabilities, threat scenarios, unwanted incident, likelihoods, consequences, risk level* **Out:** *Prioritized list of testable security relevant features (security feature sets).*

Supported by the **RASEN toolbox** and the **RASEN exchange** format



Mapping to System Lifecycle Phases

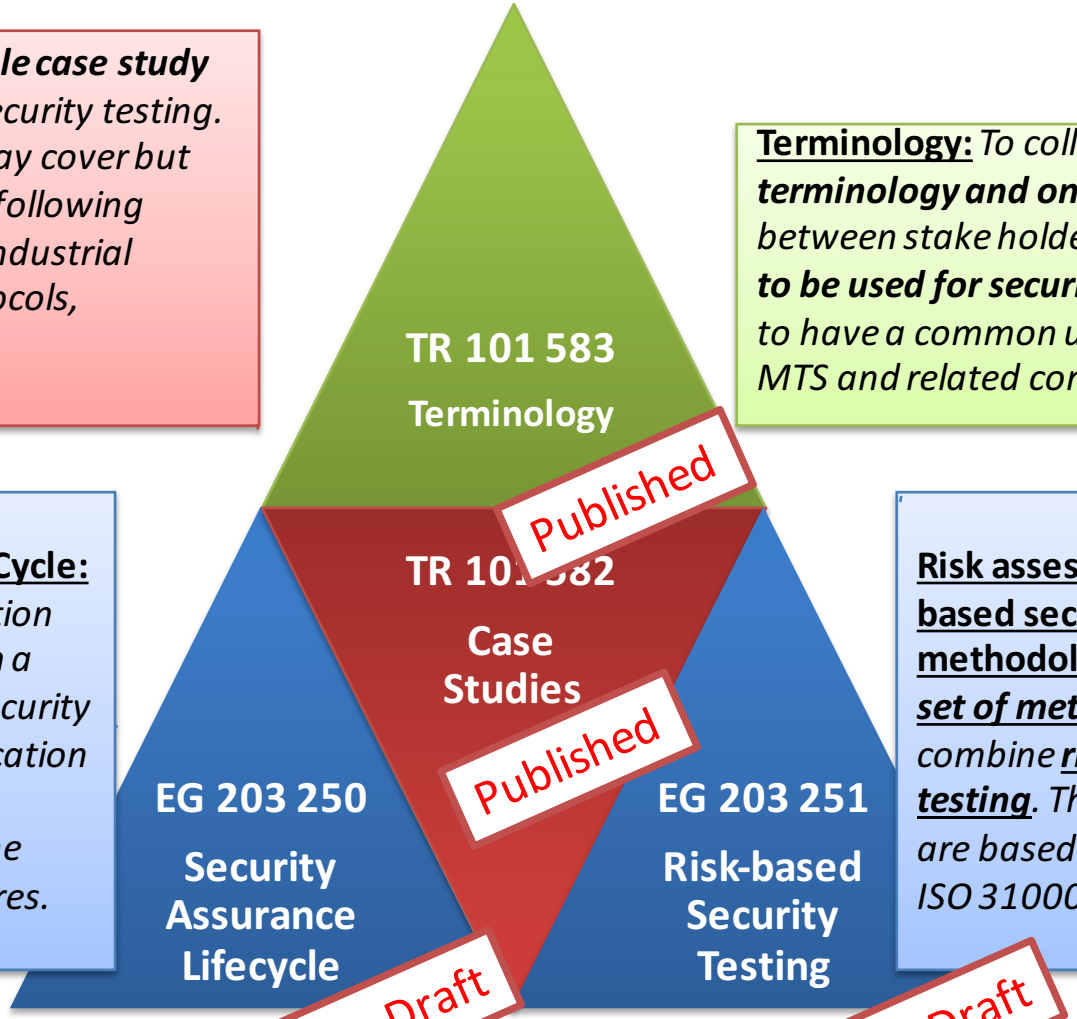


The RASEN method is itself in **standardization**

SEVENTH FRAMEWORK PROGRAMME

Case Studies: *To **assemble case study experiences** related to security testing. Industrial experiences may cover but are not restricted to the following domains: Smart Cards, Industrial Automation, Radio Protocols, Transport/Automotive, Telecommunication*

Terminology: *To collect the **basic terminology and ontology** (relationship between stake holder and application) **to be used for security testing** in order to have a common understanding in MTS and related committees.*



Security Assurance Life Cycle: ***Guidance to** the application **system designers** in such a way to maximise both security assurance and the verification and validation of the capabilities offered by the system's security measures.*

Risk assessment and risk-based security testing methodologies: *Describes a **set of methodologies** that combine **risk assessment and testing**. The methododologies are based on standards like ISO 31000 and IEEE 29119*

RASEN

RASEN method summary

- Covers the integration of security testing and risk assessment
- Is concisely specified and supported by tools
- Is mature and powerful
  - applied to all RASEN case studies
  - integrates with recent risk assessment and testing standards
  - constitutes standardization work item at ETSI

THANK YOU!

Questions and Comments?