Asset-Centric Security Risk Assessment of Software Components

Tobias Rauter, Institute for Technical Informatics

2016-01-19

Topic

1. Introduction
2. Approach
3. Evaluation
4. Conclusion and Outlook

Context/Motivation

- Security in complex (software) systems
- Restrict access to critical resources
  - What are these resources?
- Separation of privilege
  - How?
- Security assessment and audits
  - Which parts?

Primary Security Goals of Your Organization

- Protect valuable objects (Assets)
  - value (=> impact)
  - exposure (=>probability)
- Risk

		Likelihood of Incident Scenario				
		Very Low	Low	Medium	High	Very High
Business Impact	Very Low	0	1	2	3	4
	Low	1	2	3	4	5
	Medium	2	3	4	5	6
	High	3	4	5	6	7
	Very High	4	5	6	7	8

Risk Management

- Basically: Identify -> Rate -> Treat
- On organizational level
  - Various (somewhat similar) approaches
  - Here: ISO 27005

Software Security

- ’Security by Design’
- Critical resources?
  - Organizational level assets mapped into SW architecture
- SW components use or protect these assets
  - ’Secondary assets’
  - Useful in organizational level assessment?
  - Asset risks depend on SW components?
- Critical Components?

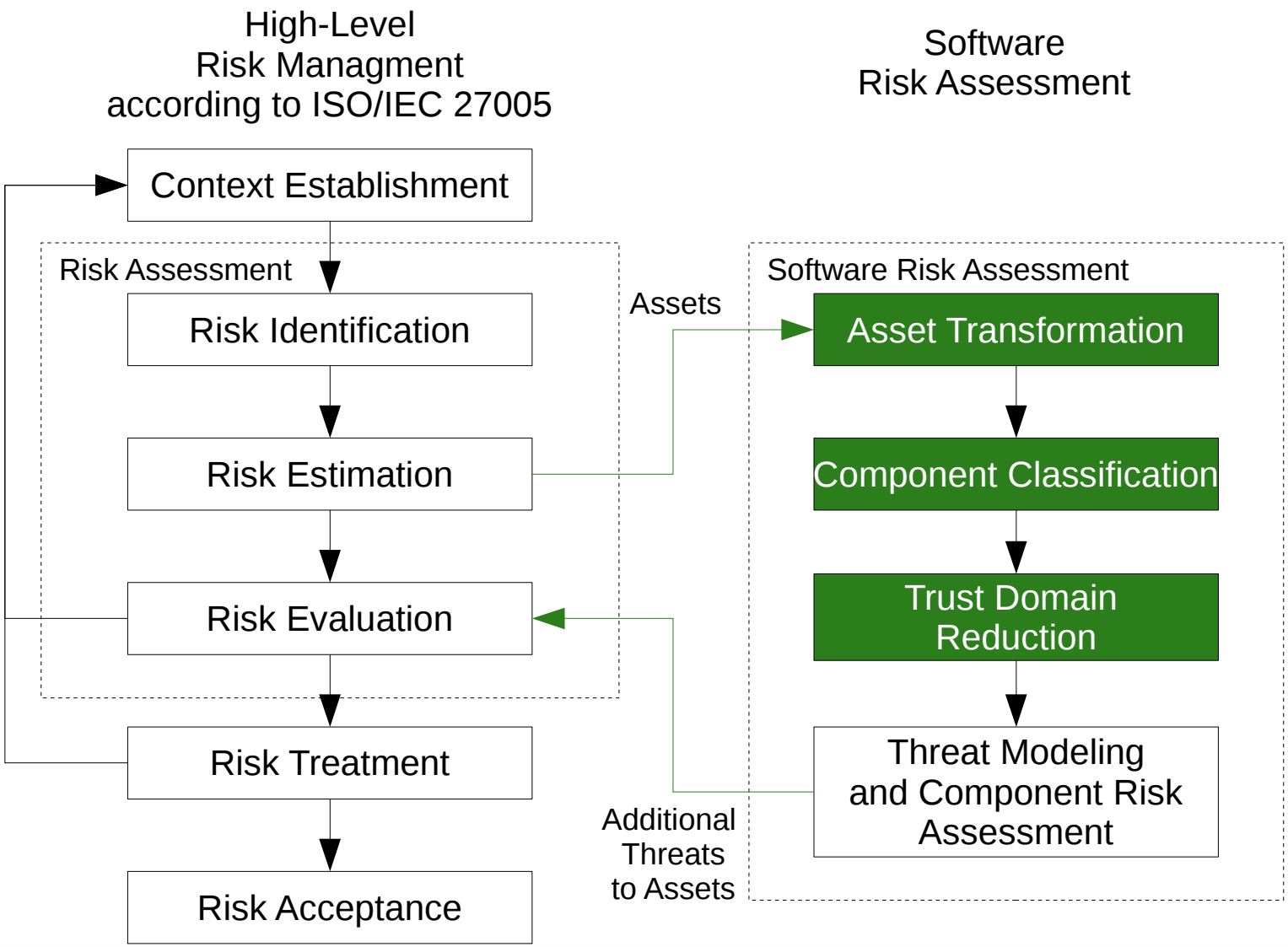
This Work: Combination

- Feed-backed high to low level risk analysis
  - Use information of high level risk analysis in threat modeling
  - Feed-back adjustments on asset risks
- Classification based on criticality (of accessed assets)
- Separation of privilege with special components
- Evaluation with manufacturing system use-case

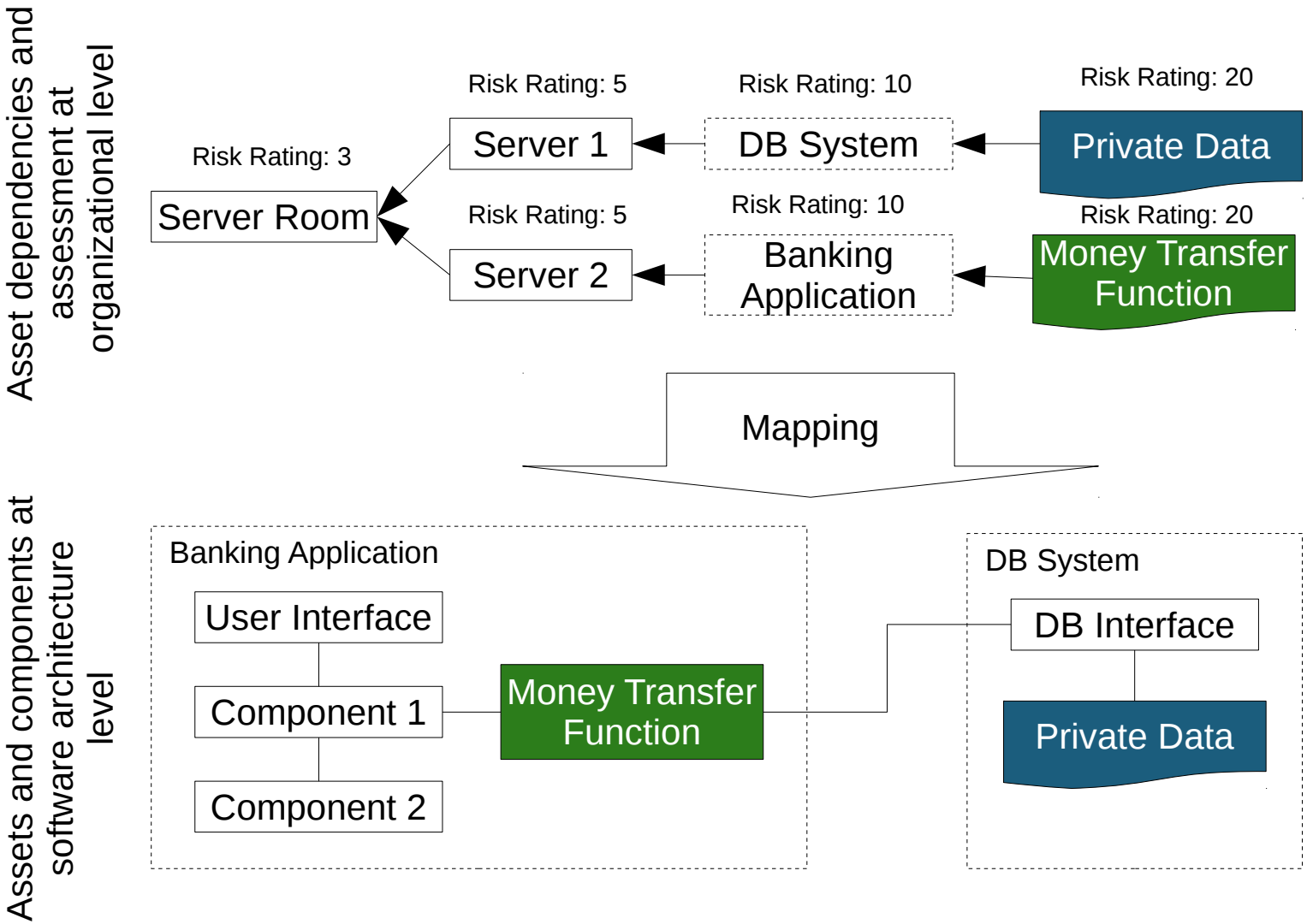
Topic

1. Introduction
2. Approach
3. Evaluation
4. Conclusion and Outlook

Overview


High-Level
Risk Managment
according to ISO/IEC 27005
Software
Risk Assessment
Context Establishment
Risk Assessment
Risk Identification
Risk Estimation
Risk Evaluation
Risk Treatment
Risk Acceptance
Assets
Additional
Threats
to Assets
Software Risk Assessment
Asset Transformation
Component Classification
Trust Domain
Reduction
Threat Modeling
and Component Risk
Assessment

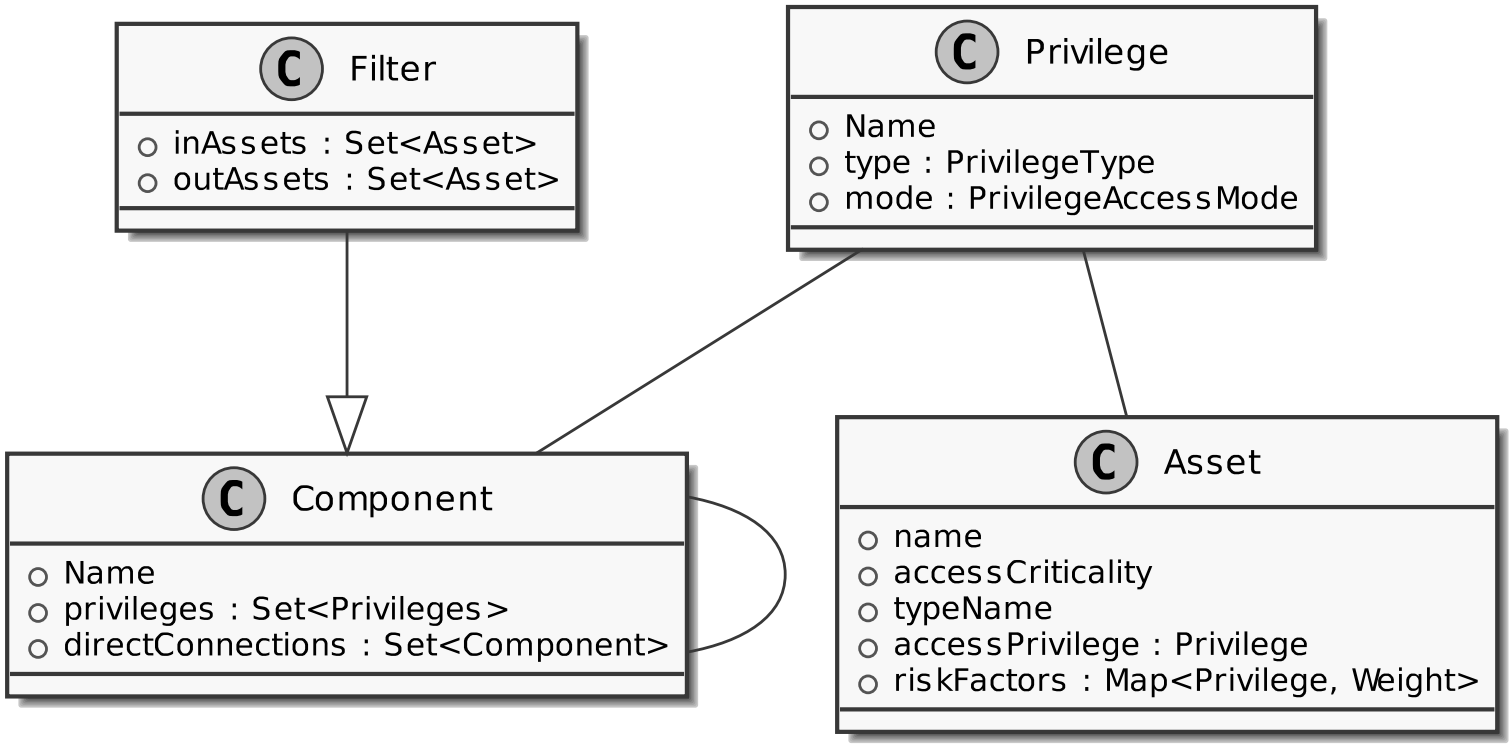
Asset Mapping



Component Model

- Architectural model of software system
- Simple component model with (non-directed) data-flows
  - Software components
  - Assets

Component Model



Component Classification

- Privilege rating
  - Value of accessed assets + ’risk factors’
- Represents impact-part of the risk
  - Probability may result from threat modeling process
- Access via privileges (foundation for future work)

Component Classification: Risk Factor

- Privileges may interfere
  - Access to sensitive data
  - Exposed to network
- 'Risk Factors' are quantified per asset
- Map privilege => value

Component Classification: Privilege Rating

- Privilege Rating *PR* of Component *C*
- 'Value' *Crit()* of all accessed assets *A*
- Risk factors *RF* of this assets for all other privileges *P*

$$PR(C) = \sum_{A=Assets(C)} \left(Crit(A) + \sum_{P=Priv(C)} RF(A,P) \right)$$

Classification: Component Composition

- Merge Privileges
- Calculate Privilege Rating
- Directed information flows
  - Future work
- Restricting components
  - Filter

Filter Components

- Special component
- Transform assets:

$$A \Rightarrow \emptyset(block) \tag{1}$$

$$A \Rightarrow A'(reduction) \tag{2}$$

$$A, B \Rightarrow C(transformation) \tag{3}$$

Filter Components

- Authentication
  - ’All Data’ $\Rightarrow$ ’Data of User X’
- Encryption
  - ’Data’, ’Key’ $\Rightarrow$ ’Encrypted Data’

Trust Domain

- Components that share privileges
- Minimize size (attack surface)
- Add filter components
  - Separation, Reduction
- Iterate until acceptable risk and size

Threat Modeling

- Prioritize
  - High risk components (domains)
  - Protection components (filter on borders)
- Unleashes new threats
  - Feed back to high-level risk management process

Topic

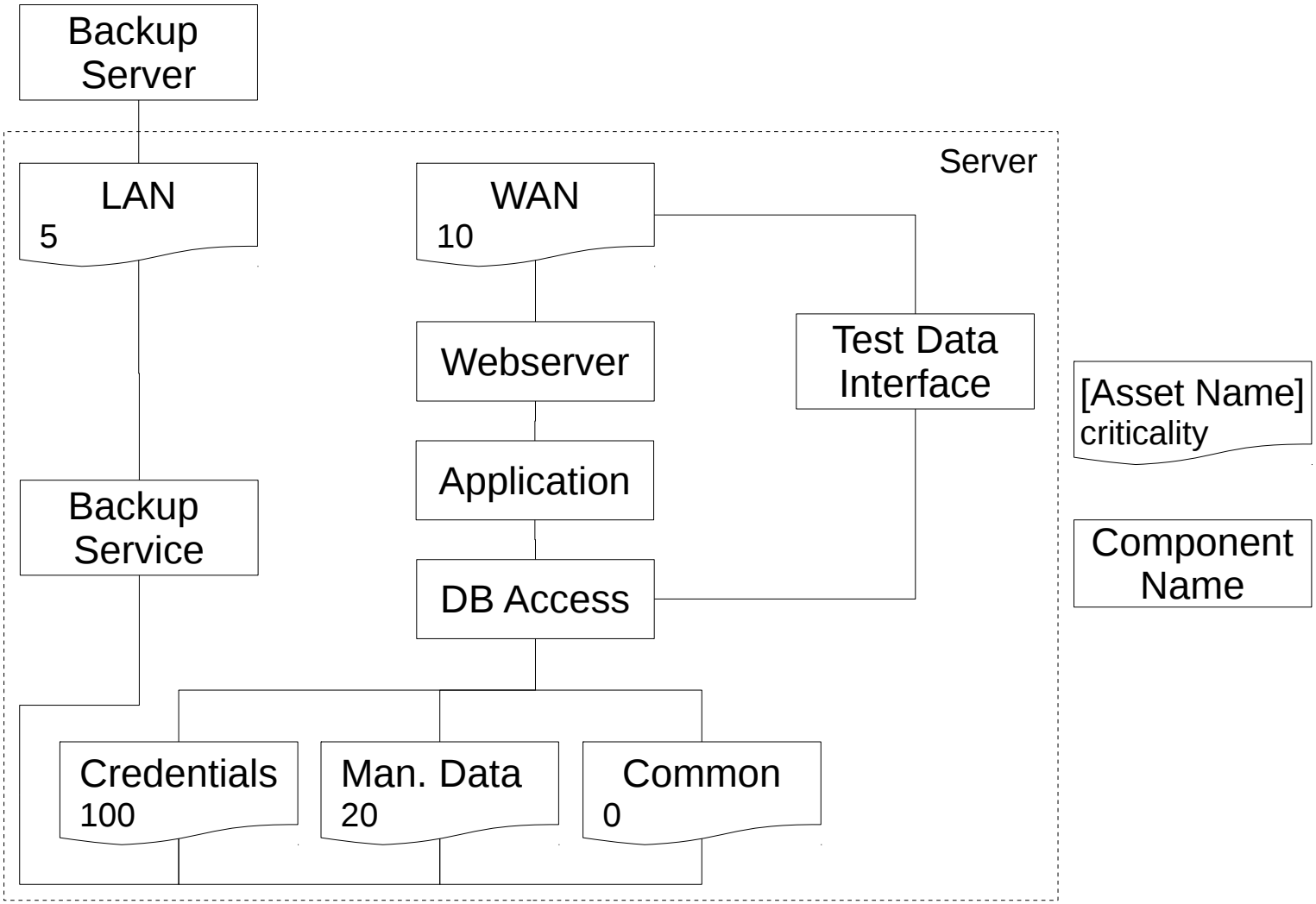
1. Introduction
2. Approach
3. Evaluation
4. Conclusion and Outlook

Use Case

- Manufacturing system
- Embedded control systems
- Manufactures receive test equipment
- Central database (of device vendor)
  - Send production data
  - Get images, certficates, etc.

Central server simplified for this paper

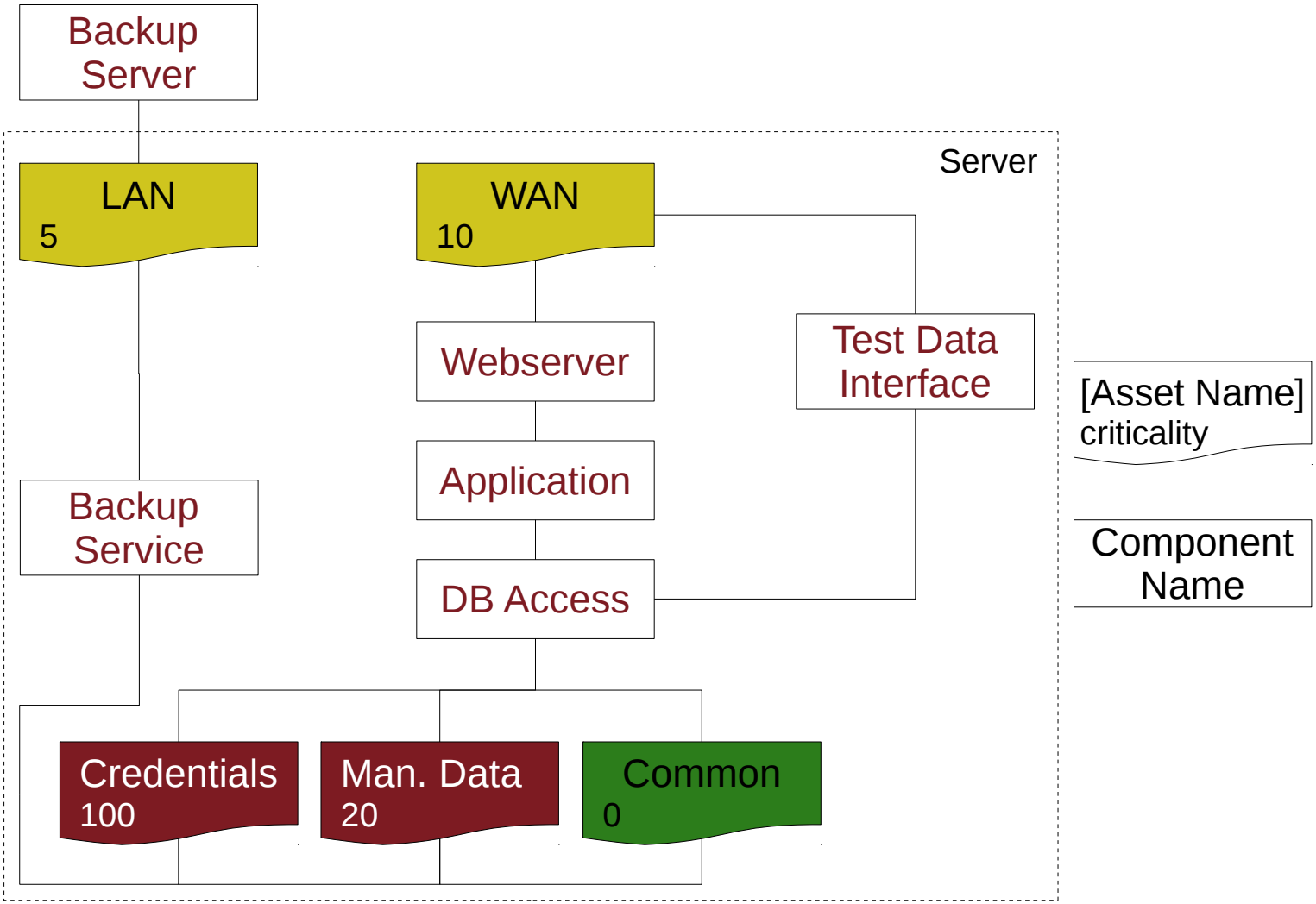
System Overview



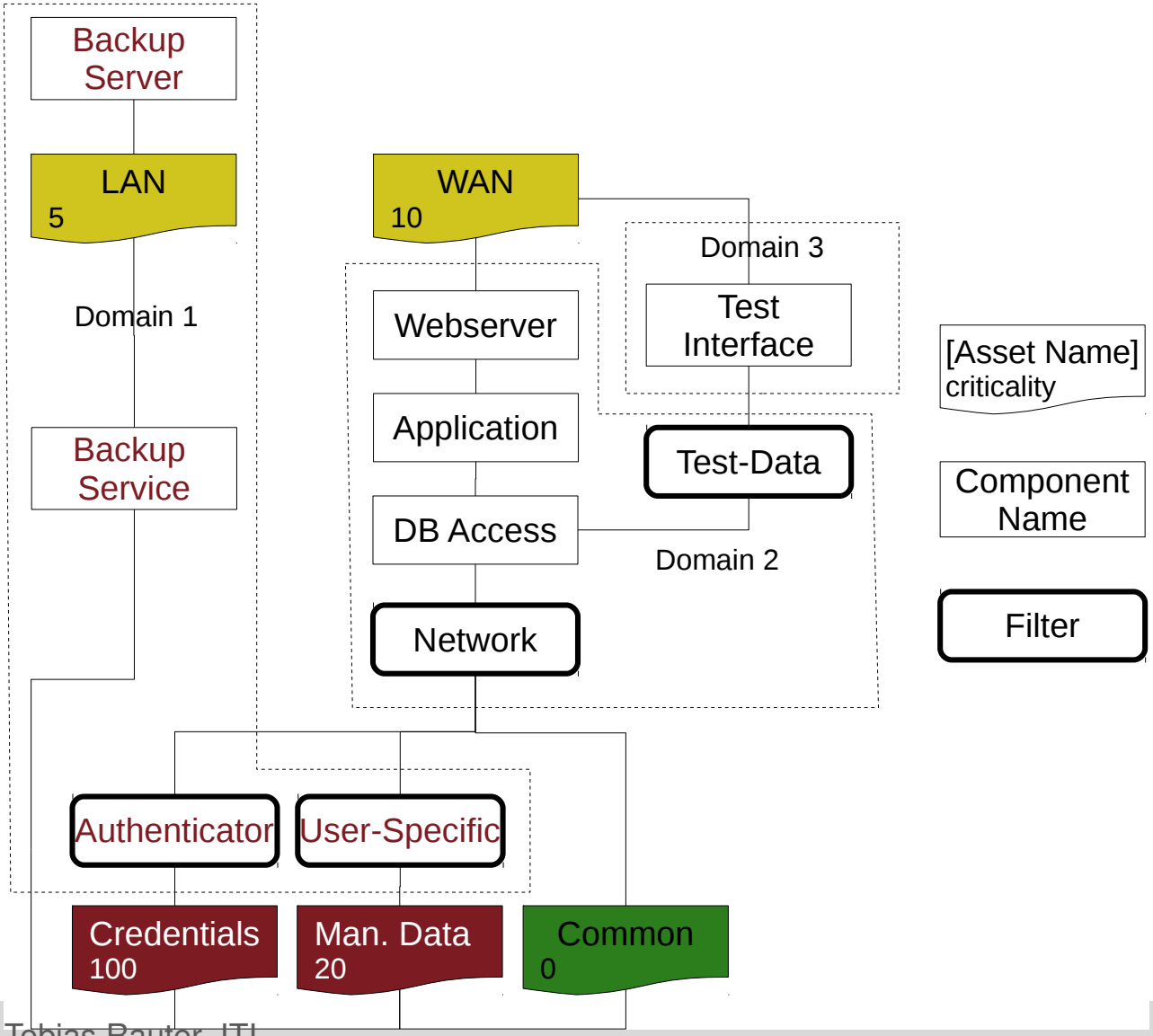
Asset Mapping

Name	Crit(A)	Risk Factors
Credentials	100	Network(WAN), 10
Manufacturing	10	Network(WAN), 5
Common	0	
LAN	10	Network(WAN), 2
WAN	10	
User-Specific Data	5	Network(WAN), 2
Test Data	5	Network(WAN), 2

Component Classification



Trust Domain Reduction



Trust Domain Reduction

Component Name	w/o Filter		with Filter	
	Domain	Criticality	Domain	Crit.
Webserver	0	1120	2	10
Application	0	1120	2	10
DB Access	0	1120	2	10
Test Interface	0	1120	3	8
Backup Service	0	1120	1	125
Backup Server	0	1120	1	125
Authenticator	-	-	1	125
User-Specific	-	-	1	125
Test-Filter	-	-	2	10
Network-Filter	-	-	2	10

Evaluation

- Domain 1
  - Full access
  - Critical
- Domain 2
  - Exposed through internet
  - User-specific data
  - Threat modeling should be done
- Domain 3
  - Relatively few privileges
  - Weakest security requirements

Evaluation cont.

- Component criticality is reduced drastically
- Focus threat modeling efforts
- New threats and assets are feeded back to high-level RM
  - Supplementing threat trees for assets
  - Ease decision for resource allocation and treatment strategies

Topic

1. Introduction
2. Approach
3. Evaluation
4. Conclusion and Outlook

Conclusion

- Risk management in complex SW systems
- Systematic approach
- High-level assessment supports SW-assessment
- Systematic reduction of trust domains
- Prioritized threat modeling
- Feedback for high-level assessment

Future Work

- Directed information flows
- Fine grained privileges (read, write, etc.)
- Find good values for risk factors
- Automate trust domain reduction