HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS
–
*Fakultät IV
Wirtschaft und
Informatik*

Competence Center
Information Technology and Management
Institute at the University of Applied Sciences
and Arts Hannover | cc_itm@hs-hannover.de

Service Computation 2021
April 18 – 22, 2021 – Porto, Portugal

The Need of Security Inside a Microservices Architecture in the Insurance Industry

A. Koschel, A. Hausotter, R. Buchta, P. Niemann, C. Rust, C. Schulze, A. Grunewald

Faculty of Business and Computer Science
University of Applied Sciences and Arts, Hannover
Ricklinger Stadtweg 120 30459 Hannover
{arne.koschel | andreas.hausotter}@hs-hannover.de

Agenda

1. Introduction
2. Reference Architecture for Microservices
3. Requirements for German Insurance Companies
4. Authorization and Authentication Patterns
5. Conclusion and Future Work

CC_ITM@HsH

- **Competence Center Information Technology & Management (CC_ITM)**
  - Institute at the University of Applied Sciences and Arts, Hannover
  - Founded in 2005 by colleagues from the departments of **Business Information Systems and Computer Science**
  - Members: **Faculty staff**, **industry partners** (practitioners) of different areas of businesses
- Main objective
  - **Knowledge transfer** between university and industry
- Research topics
  - Management of information processing
  - Service computing, including Microservices, Service-oriented Architectures (SOA), Business Process/Rules Management (BPM/BRM)
  - Cloud Computing

Introduction

The goal of our current research is the security aspect inside a **Microservices Architecture in the Insurance Industry** regarding **Security regulations in Germany**, jointly with our partner companies.

Questions to be answered:

- What are the Security Regulations in Germany and who are the corresponding Authorities?
- How is it determined if an infrastructure needs to obligate to these regulations?
- Which patterns for edge- and service-level authorization exist and what are their pros and cons?
- What considerations need to be taken for choosing specific patterns?

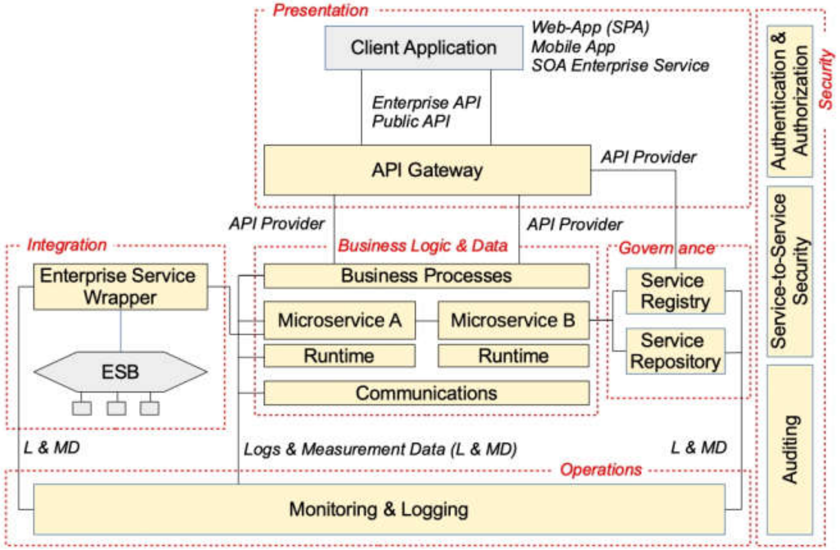
Agenda

1. Introduction
2. Reference Architecture for Microservices
3. Requirements for German Insurance Companies
4. Authorization and Authentication Patterns
5. Conclusion and Future Work

Reference Architecture for Microservices

- **Coexistence**: Legacy applications, SOA and microservices based applications will be operated in parallel for a longer transition period.

- **Security**: Consists of components to comply to general and specific security requirements.



Building Blocks of the Logical Reference Architecture RaMicsV [own representation].

Agenda

1. Introduction
2. Reference Architecture for Microservices
3. Requirements for German Insurance Companies
4. Authorization and Authentication Patterns
5. Conclusion and Future Work

Requirements for German Insurance Companies

Definition of critical infrastructures – Council of the European Union:

“.. **essential for the maintenance** of vital societal functions, health, safety, security, economic or social well-being of people, and the **disruption or destruction of which would have a significant impact** in a Member State...” [1]

Requirements for German Insurance Companies

Determination of critical infrastructures – Federal Office of Information Security (BSI) [2]:

- 7 Sectors: energy, water, food, information technology and telecommunications, health, **finance and insurance**, transport and traffic.
- Examples of critical services: **payment transactions or insurance services**.
- Certain **given thresholds** must be exceeded.
- General example:
  - Contract administration system.
  - Number of life insurance claims per year exceeds €500,000.

Requirements for German Insurance Companies

Obligation to provide evidence in Germany [3][4]:

- **Precautionary measures** to achieve protective goals IT-Security.
- Effort required for securing should be **in proportion to the consequences** of failure.
- Every 2 years to BSI.
- **Federal Financial Supervisory (BaFin) is responsible for supervision** of Banks and financial and insurance providers.
- **Catalog of requirements** for the required evidence published by BSI [5].

Agenda

1. Introduction
2. Reference Architecture for Microservices
3. Requirements for German Insurance Companies
4. Authorization and Authentication Patterns
5. Conclusion and Future Work

Authorization and Authentication Patterns

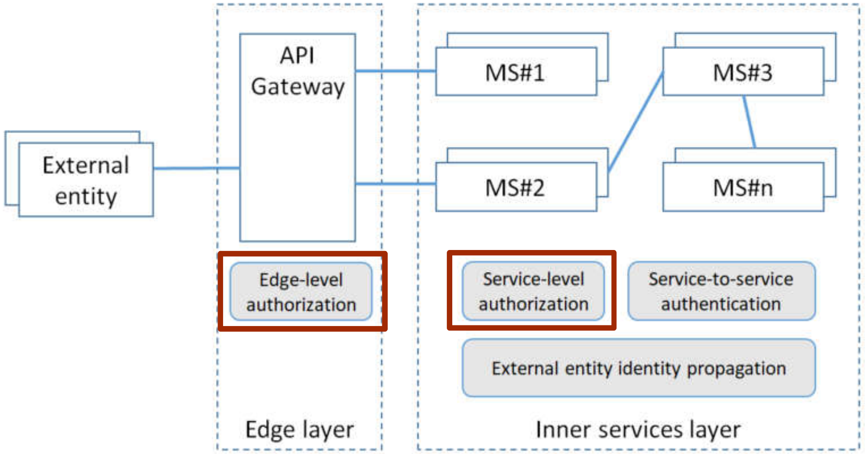
Authentication and Authorization: Differentiation

- Authentication
  - Check and calculate Data.
  - No domain knowledge necessary.
  - **Performance decides about location of this functionality.**
- Authorization
  - Domain knowledge is not necessary for role-based access control (RBAC).
  - Domain knowledge is necessary for access control list (ACL).
  - **Performance *and* access control level of detail decides about the location of this functionality.**

Authorization and Authentication Patterns

Location:

- Edge-level:
  - Functionality inside API Gateway.
  - No contact between Microservice and the functionality.
- Service-level:
  - Functionality inside every Microservice.
  - All functionality will (needs to) be developed by each Microservice-Team.


API Gateway
External entity
MS#1
MS#2
MS#3
MS#n
Edge-level authorization
Service-level authorization
Service-to-service authentication
External entity identity propagation
Edge layer
Inner services layer

Source: [6]

Authorization and Authentication Patterns

Authentication:

- Edge-level:
  - Domain logic development teams have very little involvement.
  - API Gateway development teams have to deal with more complexity.
  - Only one team is responsible, which reduces the risk of security vulnerability.
  - Faster development by lower complexity.
  - Poor scalability due to single point of control.
- Service-level:
  - Domain logic development teams have to deal with more complexity.
  - Higher risk for security vulnerabilities due to multiple development teams.
  - Slower development due higher complexity in any Microservice.
  - Higher scalability, which stresses one of the most important properties of a MSA.

Authorization and Authentication Patterns

Authorization:

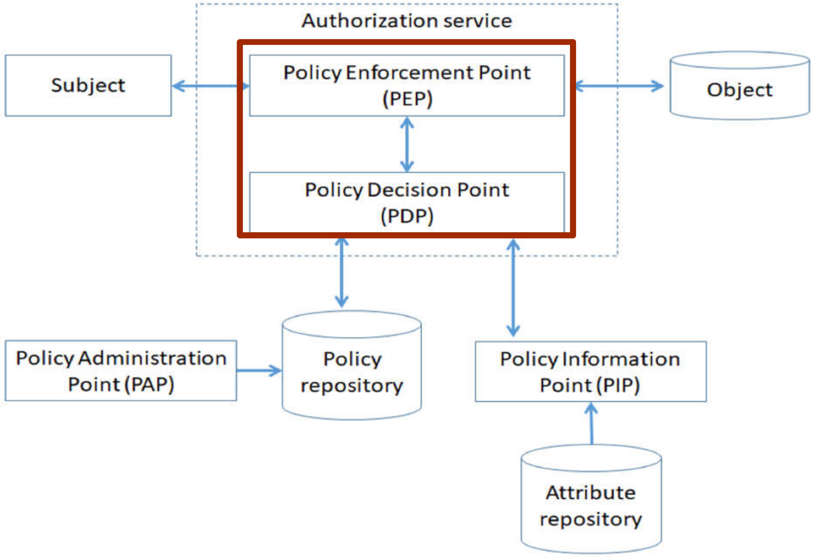
- Edge-level
  - Easy implementation and maintenance.
  - May create problems when scaling.
  - Complex systems can be difficult to design.
  - Back-end Microservices must only be accessible via the API Gateway.
  - Risk of too strong coupling of API Gateway and Microservices.
  - No independent deployment possible.
- Service-level
  - Different patterns on following slides.

Authorization and Authentication Patterns

PDP and PEP:

- Policy Decision Point (PDP): Computes the authorization decision.
- Policy Enforcement Point (PEP): Enforces the authorization decision.

The next patterns are about where PEP and PDP reside in the Microservices environment.


Authorization service
Subject
Policy Enforcement Point (PEP)
Object
Policy Decision Point (PDP)
Policy Administration Point (PAP)
Policy repository
Policy Information Point (PIP)
Attribute repository

Source: [6]

Authorization and Authentication Patterns

Changed general properties compared to edge-level:

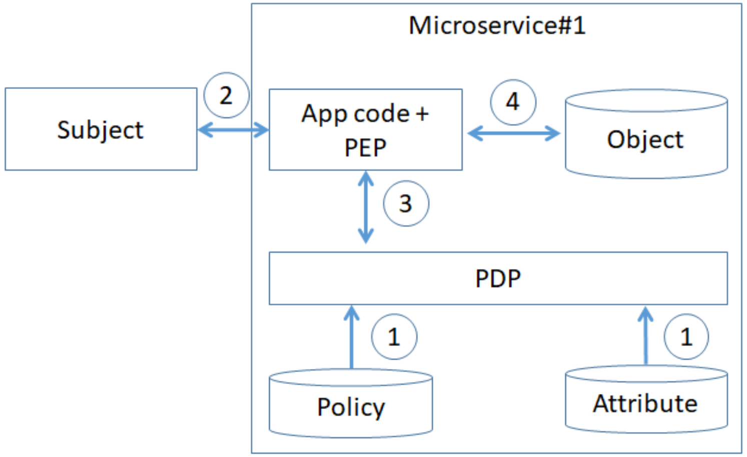
- Responsibility shifts from API development team to the Microservices development team;
- Complex Microservice environments are possible;
- Implementation and maintenance are more complex because changes affect each Microservice.

Authorization and Authentication Patterns

Decentralized pattern:

- PEP and PDP are in the Microservice.
- Properties:
  - Everything controlled by Microservice development team;
  - Optimal for scaling;
  - A lot of effort to implement and maintain
  - Propagating policy and attribute changes to all Microservices.

Recommended for Enterprise Service Bus (ESB) only if performance has the highest priority.



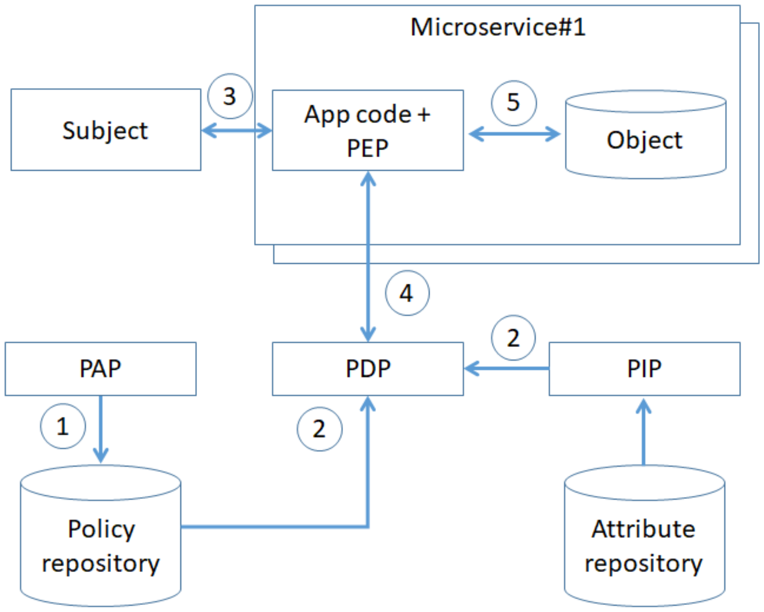
Source: [6]

Authorization and Authentication Patterns

Centralized pattern with single PDP:

- Only PEP is in the Microservice;
- Properties:
  - Every request on Microservice will result in a network call to the PDP;
  - Low effort to Microservice-Team;
  - Need of high-performance PDP component.

ESB could be the PDP component.



Source: [6]

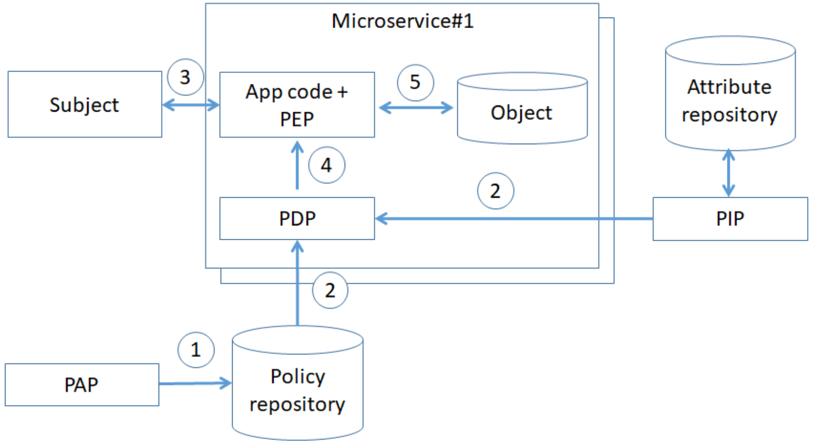
Authorization and Authentication Patterns

Centralized pattern with embedded PDP:

- PDP and PEP are inside the Microservice but embedded within a library;
- Properties:
  - Performance like „Decentralized pattern“;
  - Low effort to Microservice-Team.

ESB could be used for data and attribute sharing.

All other components could make fast decisions within the Microservices.



Source: [6]

Agenda

1. Introduction
2. Reference Architecture for Microservices
3. Requirements for German Insurance Companies
4. Authorization and Authentication Patterns
5. Conclusion and Future Work

Conclusion and Future Work

- Presented here:
  - Legal requirements and general conditions for German Insurance Companies;
  - Initial considerations for architectural security patterns, which address authentication and authorization in a Microservices Architecture.
- Next steps / future work:
  - Adding more guidelines for selection of security patterns;
  - Approach of validity and consistency of embedded policies;
  - Service-to-service authentication;
  - Relevant and current aspects of the protection goals;
  - Deployment options and resulting security domains.

References

[1] The Council of the European Union, “COUNCIL DIRECTIVE 2008/114/EC,” [Online]. Available: https://eur-lex.europa.eu/legalcontent/EN/TXT/HTML/?uri=CELEX:32008L0114. [accessed: 2022- 04-15].

[2] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal ¨Office of Information Security (BSI), , “Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz (BSI-Kritisverordnung - BSI-KritisV) (Regulation for the Determination of Critical Infrastructures according to the BSI Act (BSI-Kritisverordnung - BSI-KritisV)),” [Online]. Available from: https://www.gesetze-im-internet.de/bsi-kritisv/ BJNR095800016.html. [accessed: 2022- 04-15].

[3] Bundesamt für Sicherheit in der Informationstechnik (BSI) - Federal ¨Office of Information Security (BSI), “Act on the Federal Office for Information Security (BSI Act - BSIG) - courtesys translation -,” [Online]. Available from: https://www.bsi.bund.de/SharedDocs/Downloads/EN/ BSI/BSI/BSI Act BSIG.pdf? blob=publicationFile&v=4l. [accessed: 2022- 04-15]

[4] Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) - Federal ¨Financial Supervisory (BaFin), “Versicherungsaufsichtliche Anforderungen an die IT (VAIT) (Insurance Supervisory Requirements for IT (VAIT)),” [Online]. Available from: https://www.bafin.de/SharedDocs/Downloads/DE/Rundschreiben/ dl rs 1810 vait va.pdf? blob=publicationFile&v=5. [accessed: 2022-04-15].

References

[5] Bundesamt fur Sicherheit in der Informationstechnik (BSI) - Federal ¨ Office of Information Security (BSI), “Aufsicht uber Kritische ¨ Infrastrukturen im Finanz- und Versicherungswesen (Supervision of Critical Infrastructures in the Finance and Insurance Industry),” [Online]. Available from: https://www.bsi.bund.de/SharedDocs/ Downloads/DE/BSI/KRITIS/Nachweispruefungen im Finanz-und Versicherungswesen.pdf? blob=publicationFile&v=3. [accessed: 2022-04-15].

[6] A. Barabanov and D. Makrushin, “Authentication and authorization in microservice-based systems: survey of architecture patterns,” CoRR, vol. abs/2009.02114, 2020, [Online]. Available from: https://arxiv.org/abs/ 2009.02114. [accessed: 2022-04-15].

[7] V. C. Hu, D. Ferraiolo, R. Kuhn, A. R. Friedman, A. J. Lang, M. M. Cogdell, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, “Guide to attribute based access control (abac) definition and considerations (draft),” NIST special publication, vol. 800, no. 162, pp. 1–54, 2013.