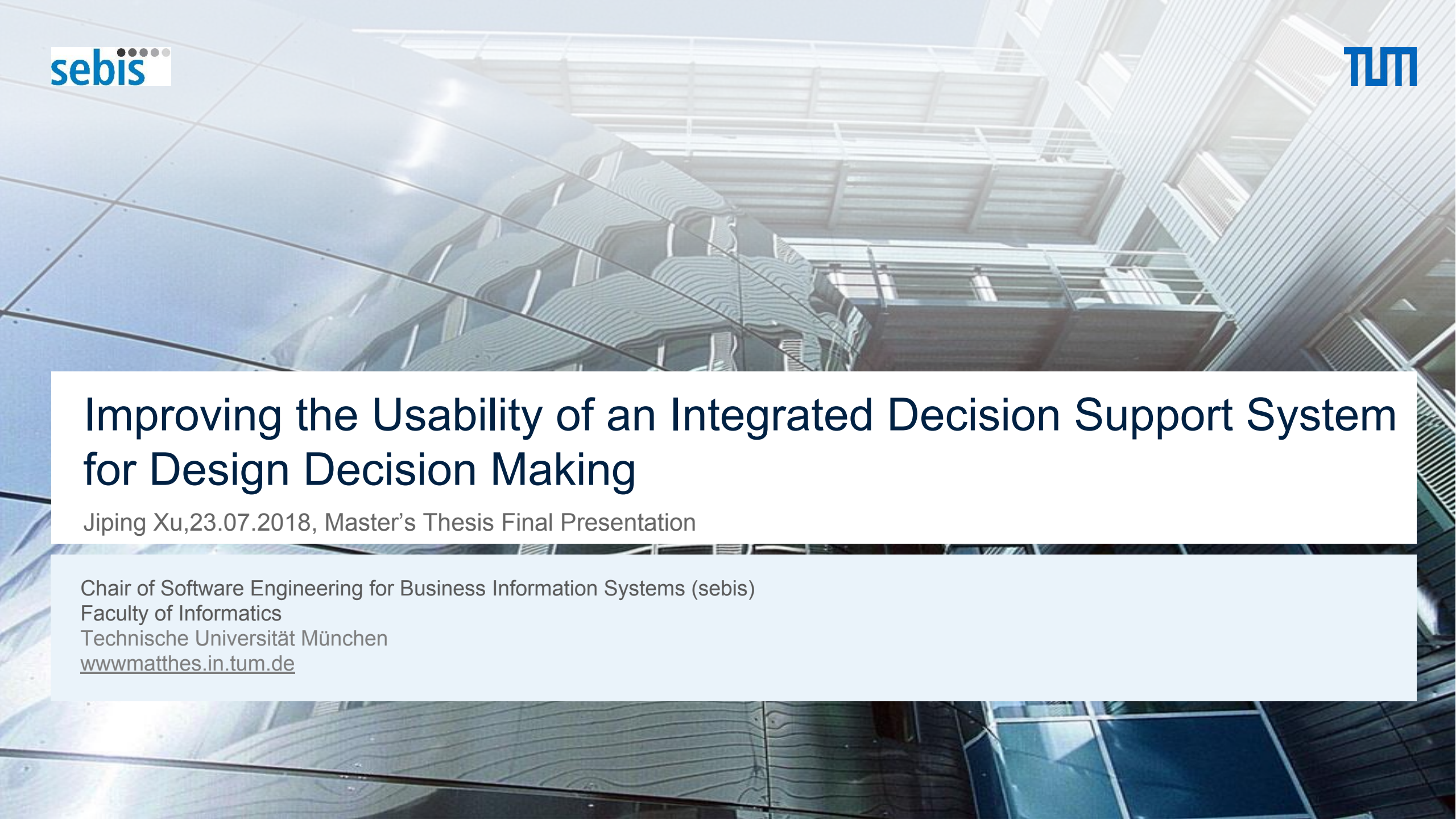


Improving the Usability of an Integrated Decision Support System for Design Decision Making

Jiping Xu,23.07.2018, Master's Thesis Final Presentation

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation
- Architectural Design Decision
- Architecture Management Enabler for Leading Industrial software (AMELIE)

Research Questions

Contribution
- Requirements Analysis
- Enhancing User Interfaces
- Containerization
- End-user and Developer Documentation

Conclusion

© sebis

Motivation

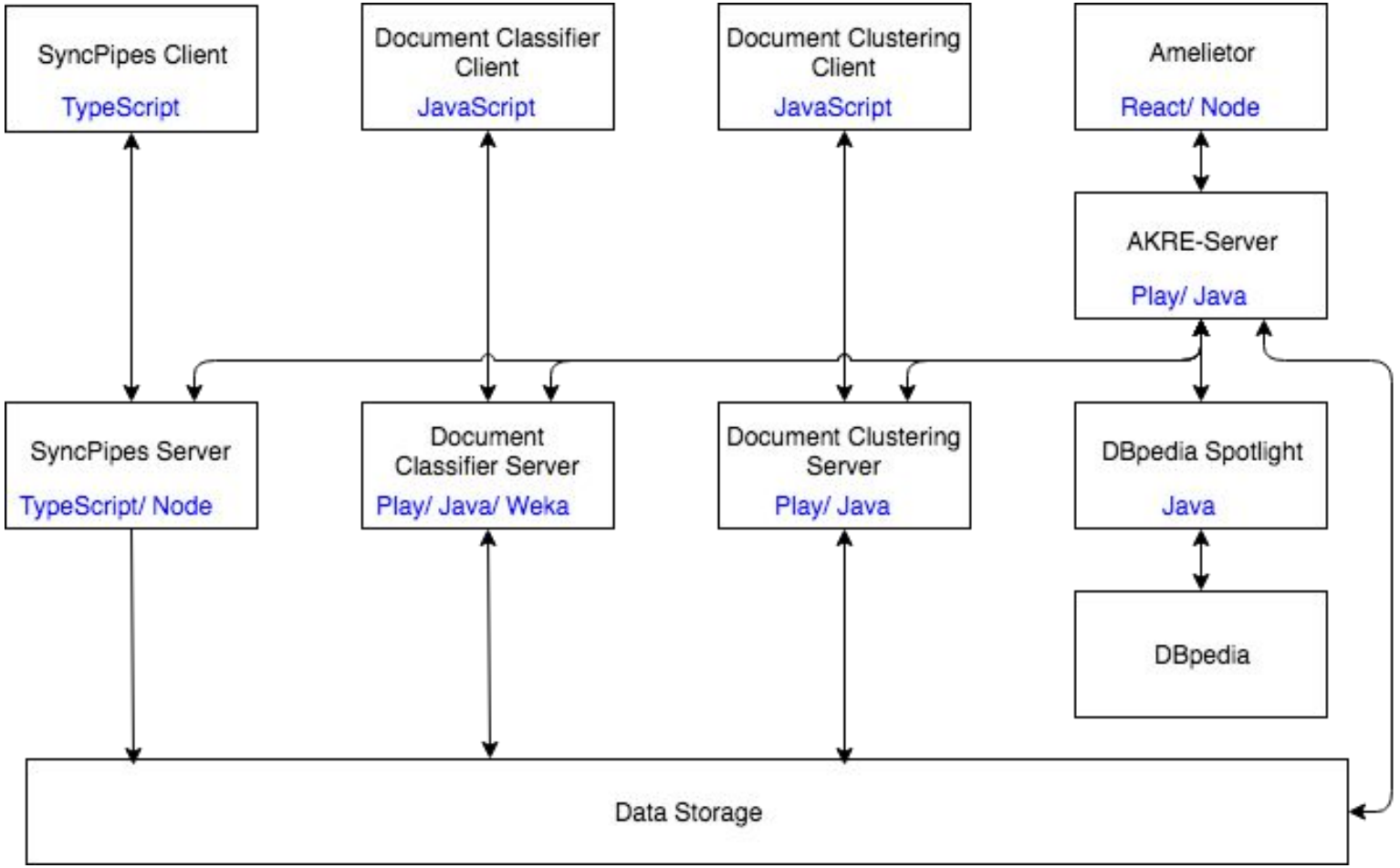
- Software architecture is a set of architectural design decisions [1].
- These design decisions involve important choices on core components to satisfy and balance stakeholders' concerns [2].
- Benefits of explicitly capturing design decisions: lower costs of change, reusability [3], …
- But architects and developers do not document their design decisions: manual effort, extra cost [4].

Design decisions are important to the success of the software, but it is difficult to make good decisions. So there is a need for a system to support architects and developers in the process of design decision making - Architecture Management Enabler for Leading Industrial software (AMELIE).

© sebis

Motivation - AMELIE

- Microservice architecture
- Developed by different people at different time



© sebis

Outline

Motivation

- Architectural Design Decision
- Architecture Management Enabler for Leading Industrial software (AMELIE)

Research Questions

Contribution

- Requirements Analysis
- Enhancing User Interfaces
- Containerization
- End-user and Developer Documentation

Conclusion

© sebis

Research Questions

RQ1: What are the requirements of a decision support system (DSS) for making design decisions?

- A rather new field with emerging models and tools
- Industry partner - Siemens AG: real-world use cases

RQ2: How to improve the *usability* of an existing DSS that supports design decision making?

- Identified stakeholders of the system: end-users and developers
- End-users:
  - Usability of user interfaces
  - Understandability
  - Learnability
- Developers:
  - Understandability
  - Maintainability

© sebis

Outline

Motivation
- Architectural Design Decision
- Architecture Management Enabler for Leading Industrial software (AMELIE)

Research Questions

Contribution
- Requirements Analysis
- Enhancing User Interfaces
- Containerization
- End-user and Developer Documentation

Conclusion

© sebis

Requirements Analysis

- Requirements elicitation based on a literature review
- Practitioners' feedback
- Artifact: a list of requirements
  - 31 requirements related to architectural knowledge management
  - 15 requirements focused on architectural design decisions

Example:

No.	Description	Siemens*	AMELIE**
1	Integration with other tools [5, 6, 9]: automatic or semi-automatic integration with others tools during different phases of software lifecycle.	Y	Y
2	Reasoning capability [6]: basic inference mechanism that can automatically detect inconsistencies, state new questions or perform some fuzzy reasoning.	Y	-
3	Architectural decision consequence analysis [7, 8, 10]: the main consequences of a decision are the changes in the model when a decision is executed. In addition, new decision topics can be introduced.	Y	P

* Is this a real-world use case from our industrial partner?

** Is this already supported by AMELIE?

© sebis

Enhancing User Interfaces (UIs)

- Performed an expert review on the UIs
- Identified 14 major issues
- Redesigned and reimplemented UI elements based on those issues
- Results:
  - Projects Overview
  - Import a Project
  - Quality Attributes
  - Architectural Elements
  - Expertise Matrix
  - Expert Recommender
  - Design Decisions
  - Decision Details

© sebis

Enhancing User Interfaces (UIs)

- Performed an expert review on the UIs
- Identified 14 major issues
- Redesigned and reimplemented UI elements based on those issues
- Results:
  - Projects Overview
  - Import a Project
  - Quality Attributes
  - Architectural Elements
  - Expertise Matrix
  - Expert Recommender
  - Design Decisions
  - Decision Details

© sebis

Enhancing User Interfaces - Projects Overview

Search...

Projects

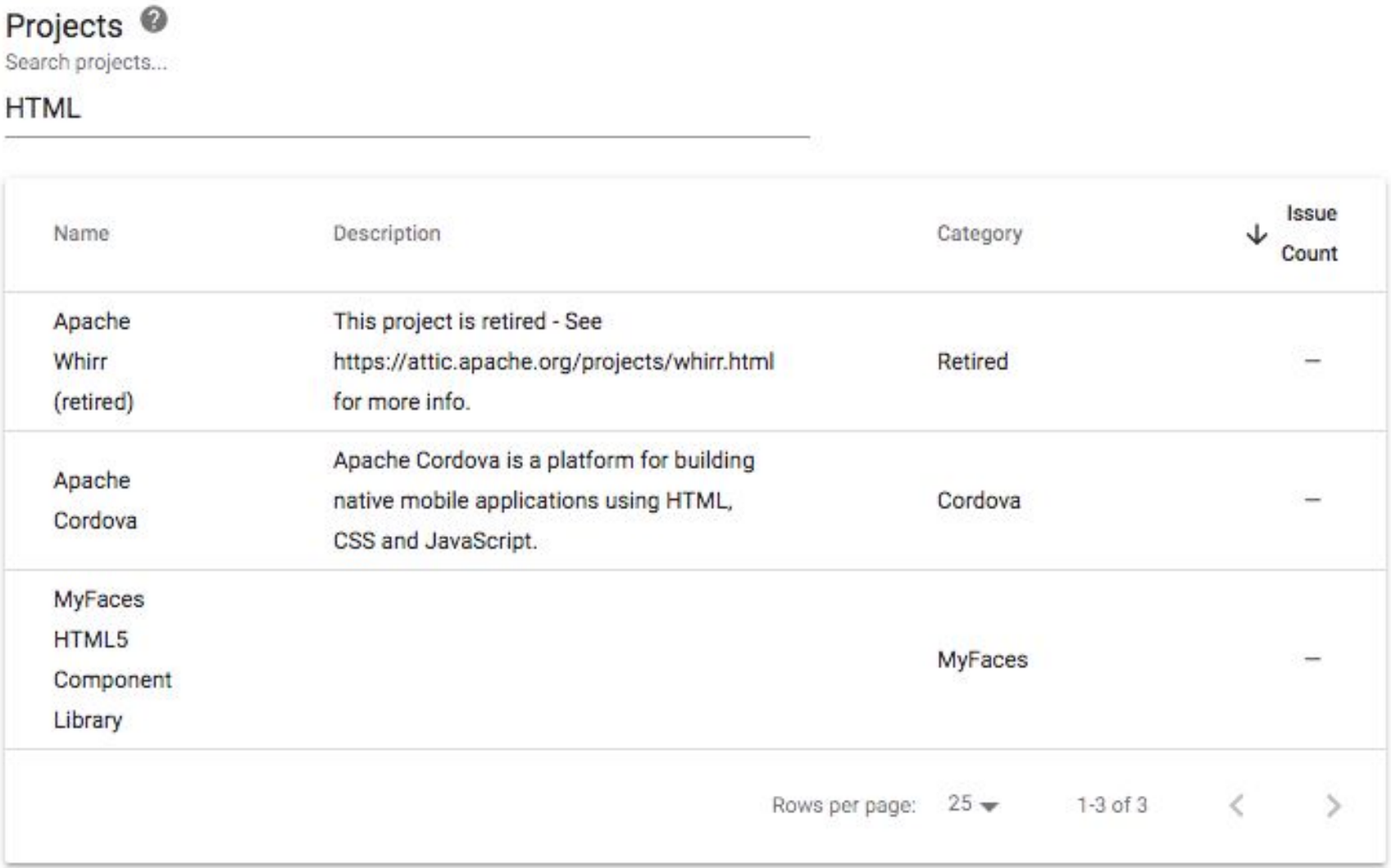
	Project Name	Description	Category	# Issues
☐	Spark	Apache Spark is a fast and general cluster computing system.	Apache Software	22279
☐	NCK	Numerical Control Kernel		16885
☐	Hadoop Common	Hadoop Common is the common library for Apache Hadoop.	Hadoop	12906
☐	Connected Mobility Lab		CML	1232
☐	XmlSchema		Web Services	Import

Identified problems:

- Search field away from table
- Misleading checkboxes
- Multiple types of elements mixed in '# Issues' column
- No pagination: entire table on one page
- Delay during scrolling the table

© sebis

Enhancing User Interfaces - Projects Overview Cont'd

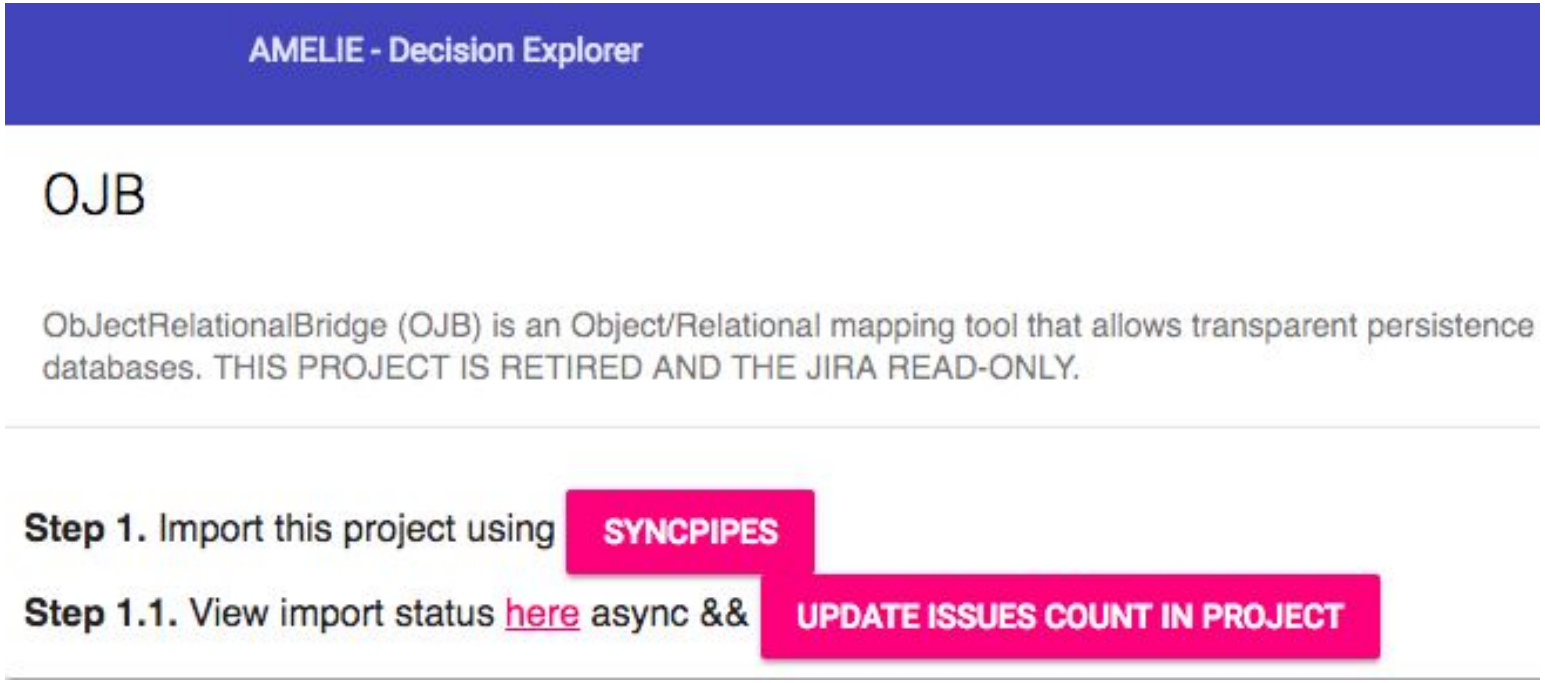


Improvements:

- Repositioned search field ✓
- Removed checkboxes ✓
- Removed import action from the '# Issues' column ✓
- Added customizable pagination ✓
- Replaced the deprecated data table component ✓

© sebis

Enhancing User Interfaces - Import a Project

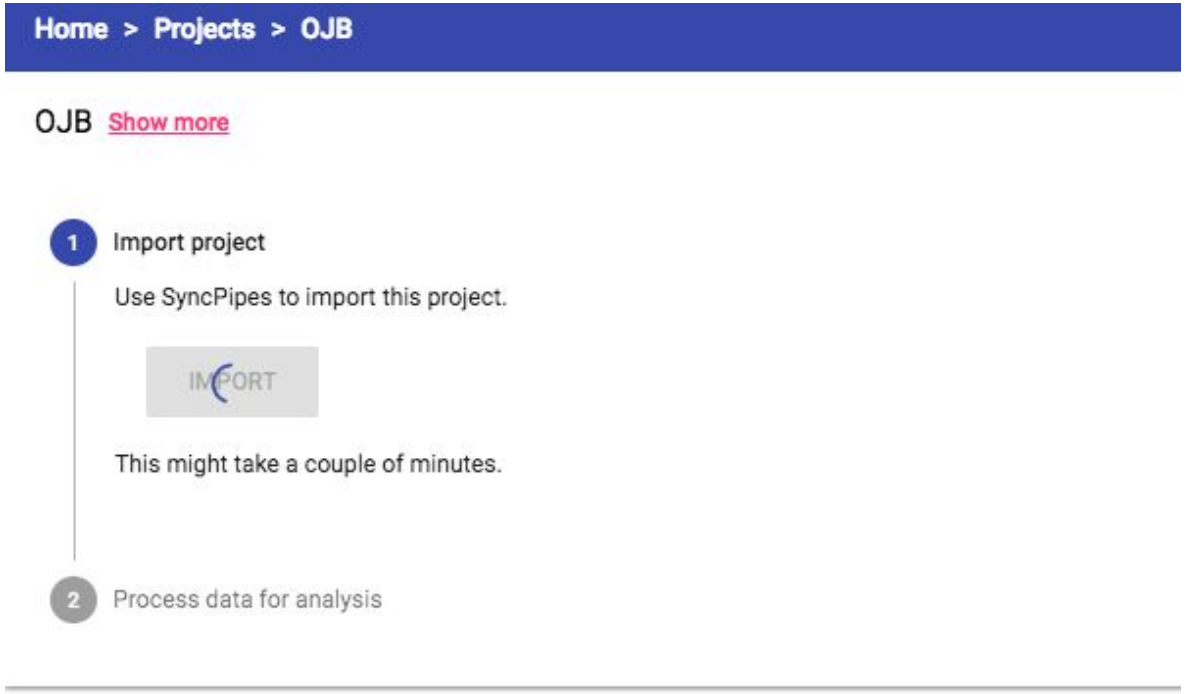


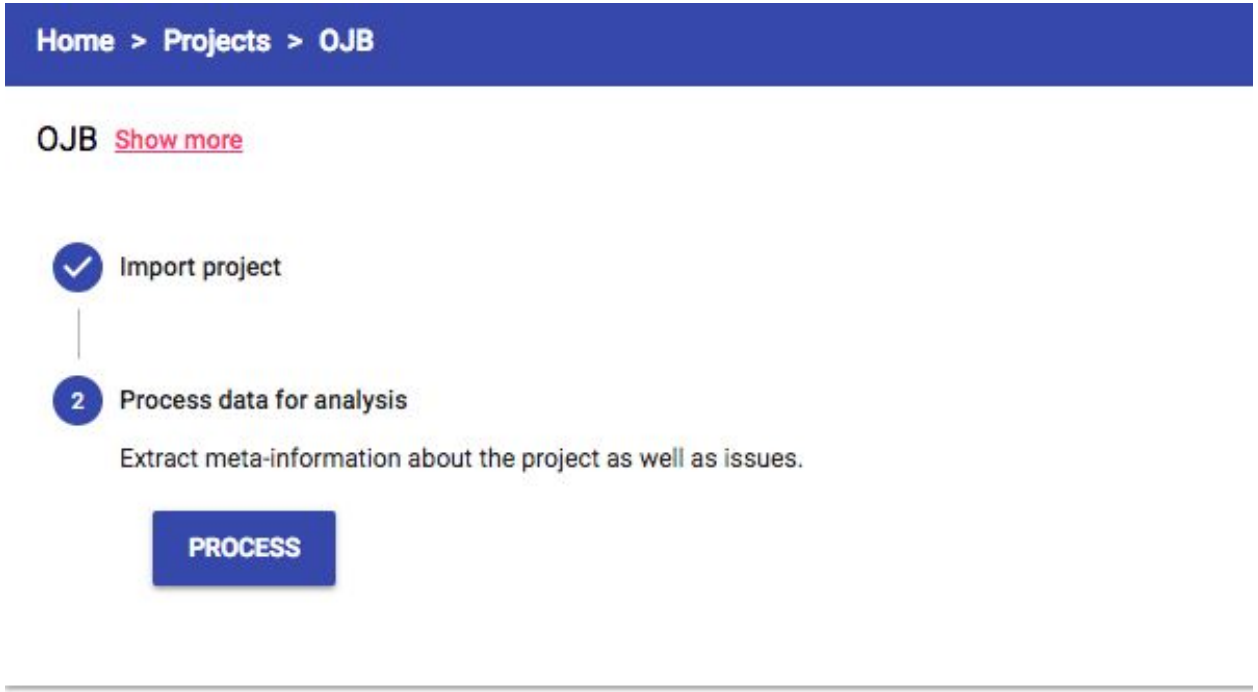
Identified problems:

- Unclear meaning of step 1.1
- Manual state checking in another web application
- Confusing button labels

© sebis

Enhancing User Interfaces - Import a Project Cont'd



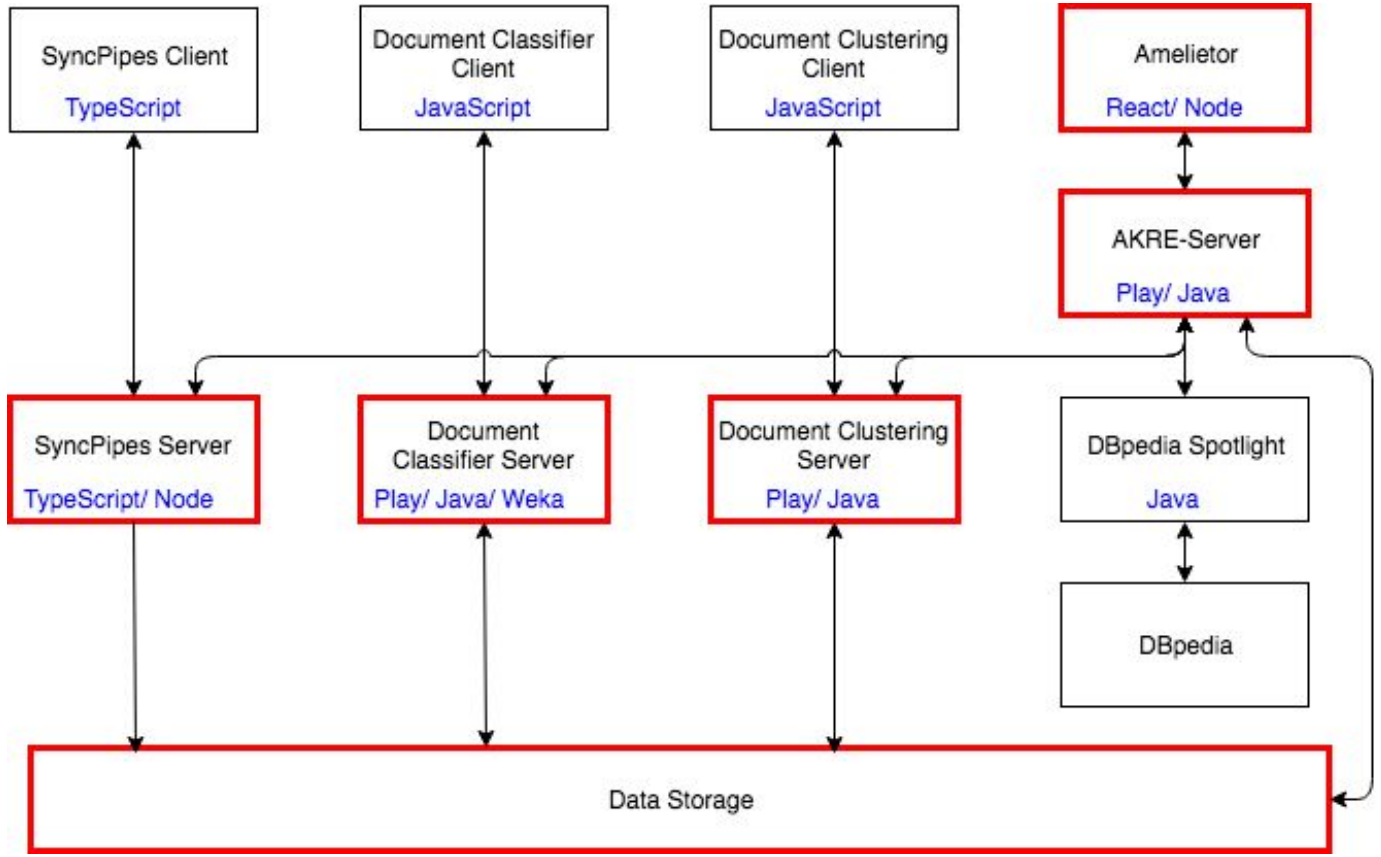


Improvements:

- Restructured the workflow for an intuitive user experience ✓
- Implemented automatic status update and step advancement ✓
- Renamed button labels ✓

Containerization

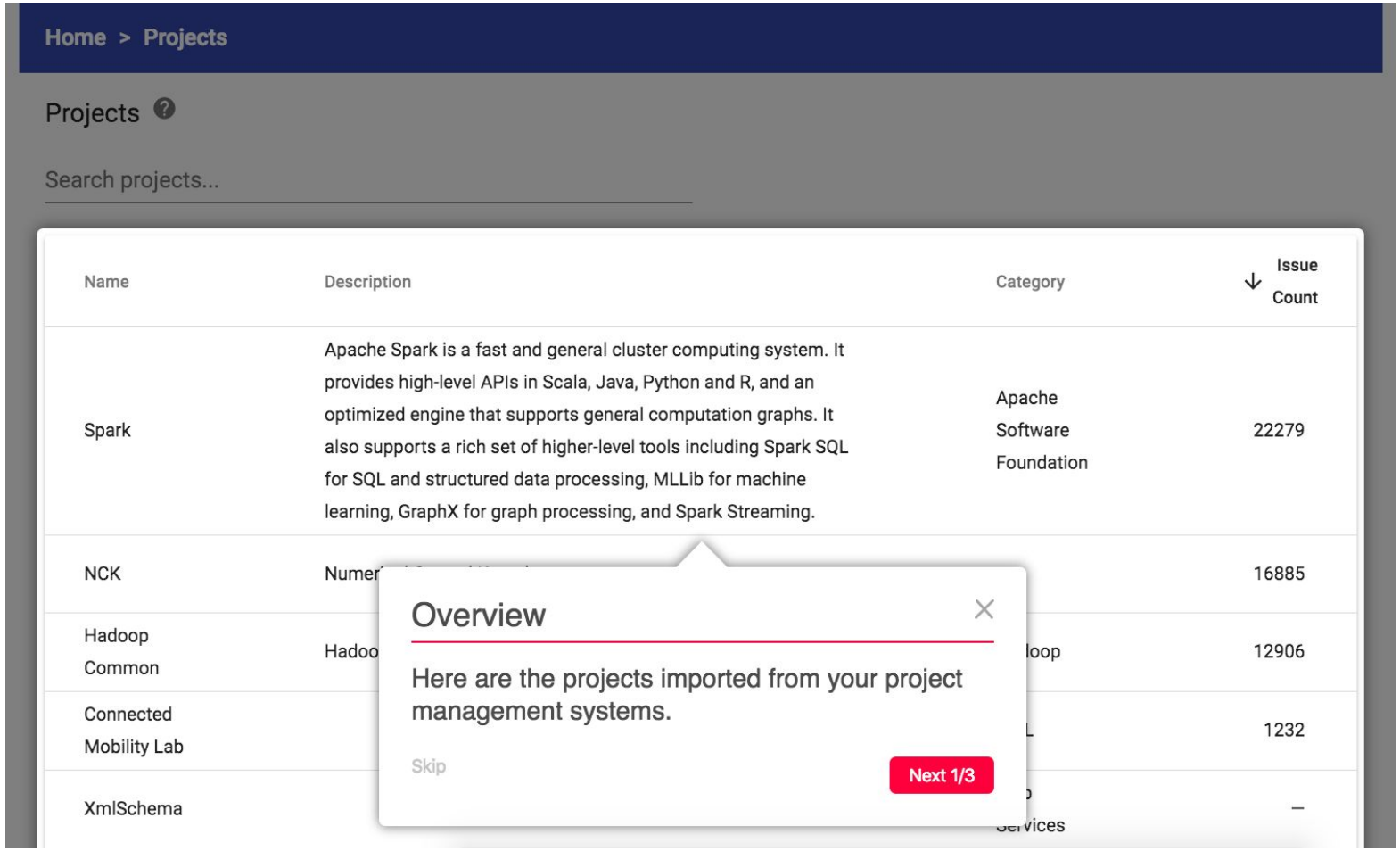
- Individual docker for each microservice
- Shared data storage:
  - One dedicated container for the database
  - Restore from database dumps
- One-click deployment



© sebis

End-user Documentation

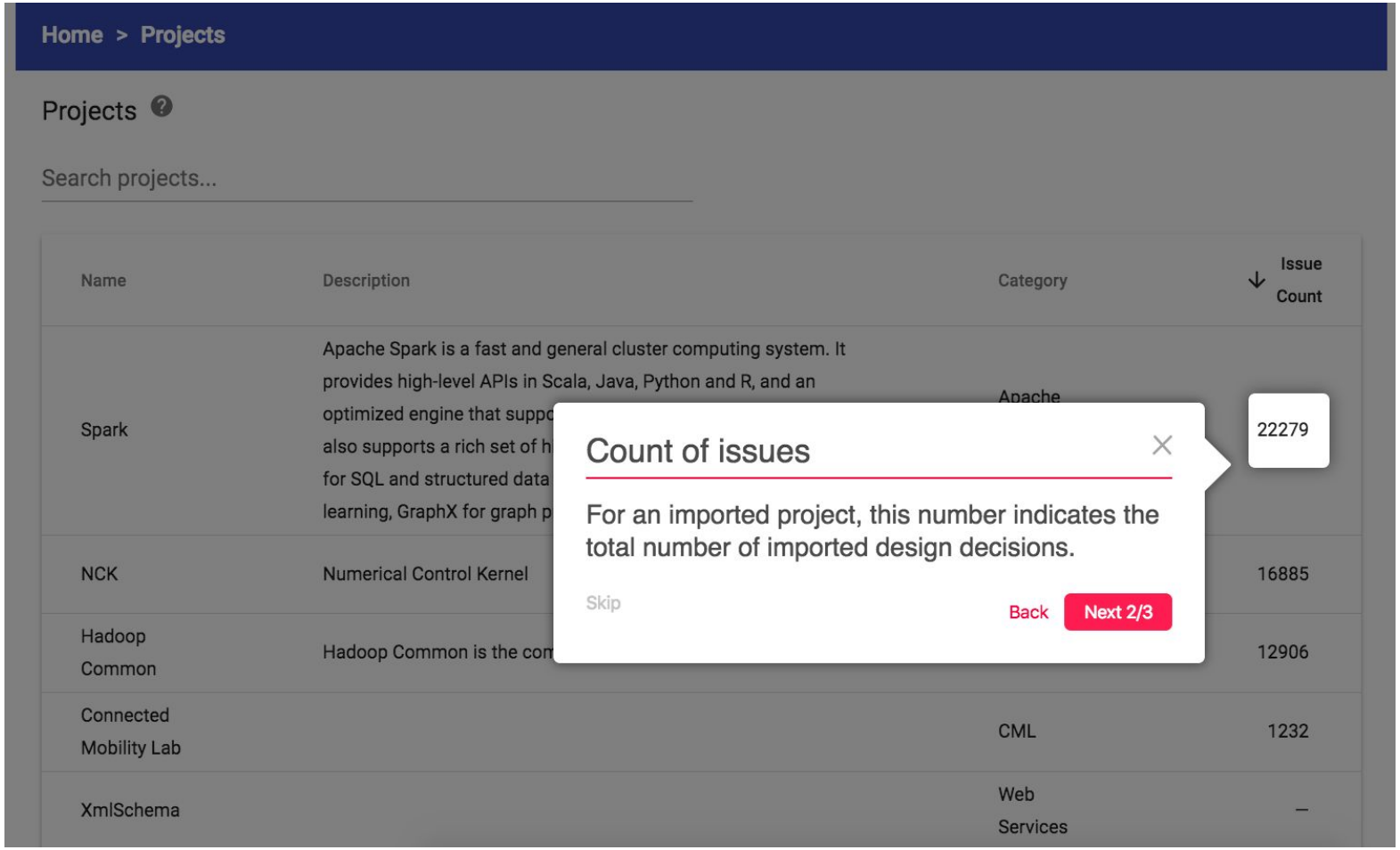
- In-app guided tour: learnability, understandability
  - Implementation: React Joyride library



© sebis

End-user Documentation - Cont'd

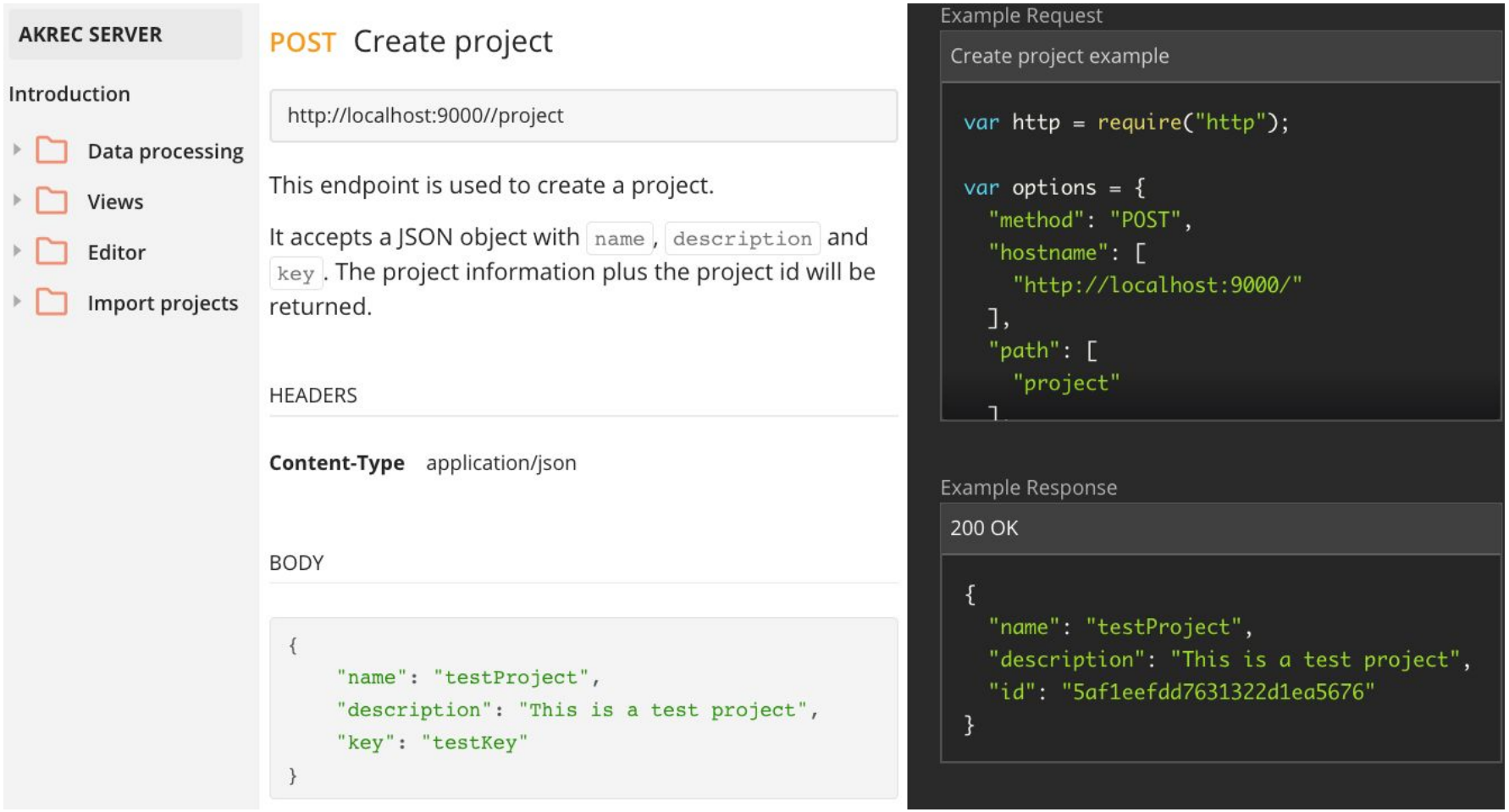
- In-app guided tour: learnability, understandability
  - Implementation: React Joyride library



© sebis

Developer Documentation

- Addresses understandability, maintainability
- API documentation based on Postman Tool
- Instructions for configuring machine learning models



© sebis

Outline

Motivation
- Architectural Design Decision
- Architecture Management Enabler for Leading Industrial software (AMELIE)

Research Questions

Contribution
- Requirements Analysis
- Enhancing User Interfaces
- Containerization
- End-user and Developer Documentation

Conclusion

© sebis

Conclusion

RQ1: What are the requirements of a decision support system (DSS) for making design decisions?

- Requirements elicitation based on a literature review
- Practitioners' feedback
- Researcher's feedback
- Artifact: a list of requirements
  - 31 requirements related to architectural knowledge management
  - 15 requirements focused on architectural design decisions

RQ2: How to improve the *usability* of an existing DSS that supports design decision making?

- End-users:
  - Enhancing user interfaces
  - In-app guided tour
- Developers:
  - Containerization [1] : development, deployment
  - API documentation [2] : development, maintenance
  - Machine learning configuration [3] : development

1. https://hub.docker.com/r/sebischair/amelie/
2. https://github.com/sebischair/akre-server
3. https://github.com/sebischair/DocClusteringWorkbench/wik

© sebis

B.Sc.

Jiping Xu

Technische Universität München
Faculty of Informatics
Chair of Software Engineering for Business Information Systems

Boltzmannstraße 3
85748 Garching bei München

Tel +49.89.289.17132
Fax +49.89.289.17136

matthes@in.tum.de
wwwmatthes.in.tum.de

References

[1] Jansen, A., & Bosch, J. (2005). Software architecture as a set of architectural design decisions. In Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on (pp. 109-120). IEEE.

[2] Zimmermann, O. (2011). Architectural decisions as reusable design assets. IEEE software, 28(1), 64-69.

[3] Tofan, D., Galster, M., Avgeriou, P., & Schuitema, W. (2014). Past and future of software architectural decisions–A systematic mapping study. Information and Software Technology, 56(8), 850-872.

[4] Capilla, R., Nava, F., & Carrillo, C. (2008, September). Effort estimation in capturing architectural knowledge. In Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering (pp. 208-217). IEEE Computer Society.

[5] Tang, A., Avgeriou, P., Jansen, A., Capilla, R., & Babar, M. A. (2010). A comparative study of architecture knowledge management tools. Journal of Systems and Software, 83(3), 352-370.

[6] Capilla, R., Jansen, A., Tang, A., Avgeriou, P., & Babar, M. A. (2016). 10 years of software architecture knowledge management: Practice and future. Journal of Systems and Software, 116, 191-205.

[7] Van Der Ven, J. S., Jansen, A., Avgeriou, P., & Hammer, D. K. (2006, July). Using architectural decisions. In Second International Conference on the Quality of Software Architecture (Qosa 2006).

[8] Jansen, A., Van Der Ven, J., Avgeriou, P., & Hammer, D. K. (2007, January). Tool support for architectural decisions. In Software Architecture, 2007. WICSA'07. The Working IEEE/IFIP Conference on (pp. 4-4). Ieee.

[9] Regli, W. C., Hu, X., Atwood, M., & Sun, W. (2000). A survey of design rationale systems: approaches, representation, capture and retrieval. Engineering with computers, 16(3-4), 209-235.

[10] Tang, A., Jin, Y., & Han, J. (2007). A rationale-based architecture model for design traceability and reasoning. Journal of Systems and Software, 80(6), 918-934.