Data Management in Microservices: State of the Practice, Challenges, and Research Directions

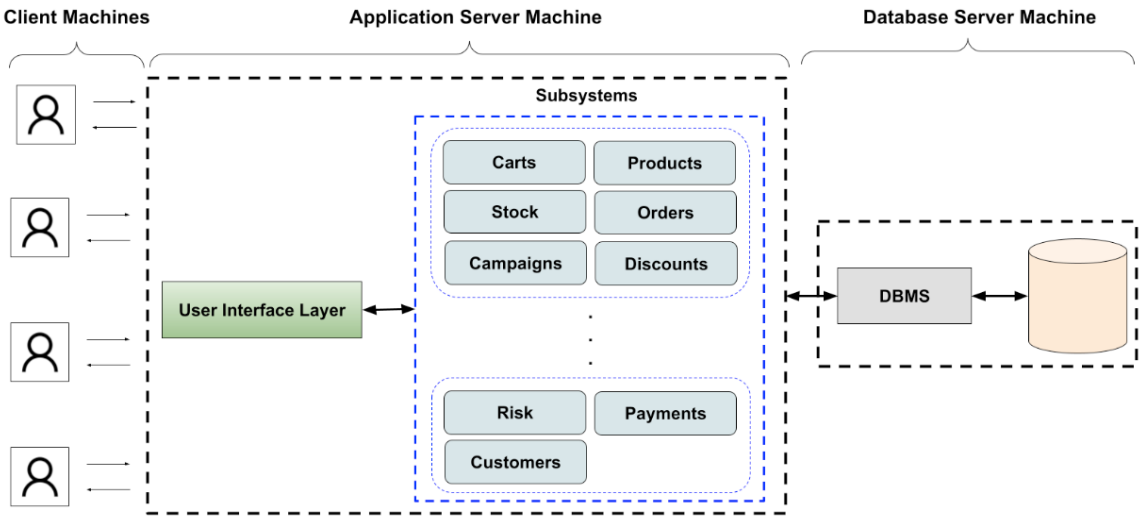
Rodrigo Laigner[1], Yongluan Zhou[1], Marcos Antonio Vaz Salles[1],
Yijian Liu[1], Marcos Kalinowski[2]
[1]University of Copenhagen, [2]PUC-Rio

Presenter: Rodrigo Laigner
PhD Fellow – Data Management Systems Group
Department of Computer Science

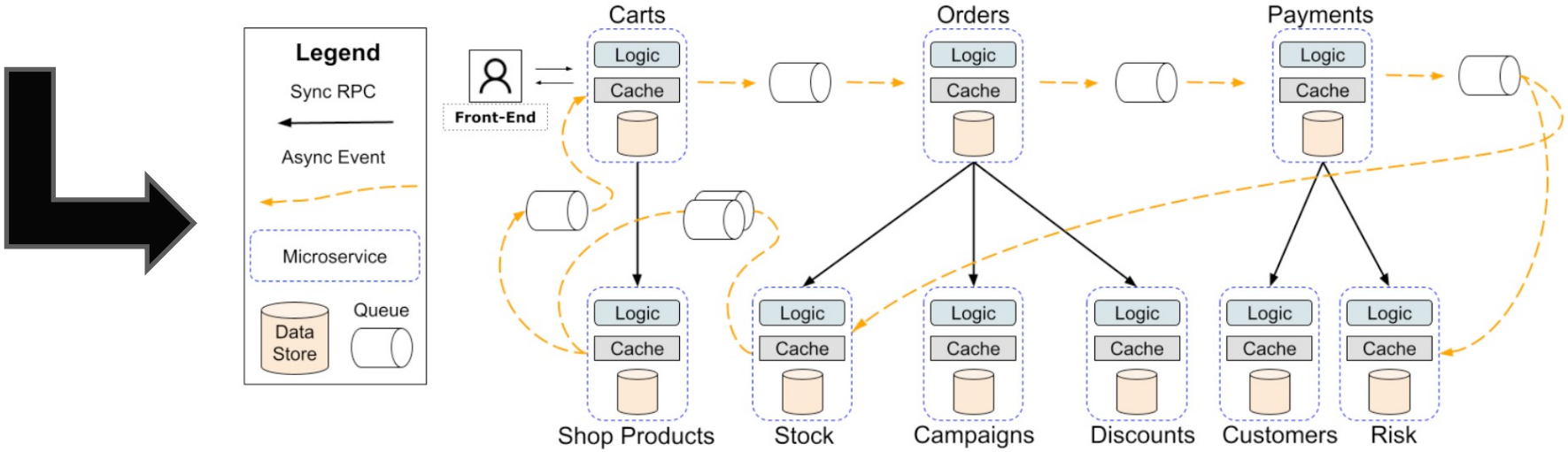
September 6th, 2022

UNIVERSITY OF COPENHAGEN

Background

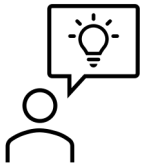


(a) Traditional monolithic architecture

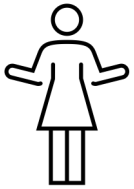


(b) Microservice architecture

Motivation



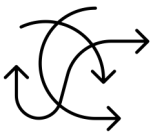
State is partitioned and encapsulated



However, it remains unknown:

- Database system technologies
- Database deployment patterns
- Mechanism to exchange data
- Data consistency semantics

Pressing data management challenges developers face

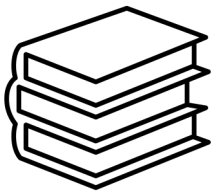


Research Directions

Research Methodology

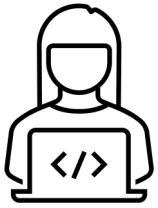
Background and Related Work

- Systematic literature review on peer-reviewed microservice papers
- Identify data management practices



Open-Source Software

- Popular and representative microservice applications selected for code review
- Validate data management practices and identify common pitfalls

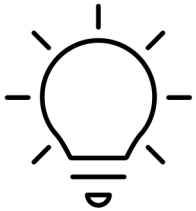


Expert Survey

- Design an online survey
- Practitioners and researches experiences with microservices
- Cross-validate findings from previous steps

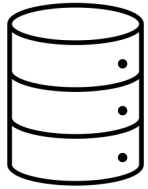
State of the Practice

- Three interrelated explorations
- New insights and evidence on data management practices
- Characterize gaps and challenges



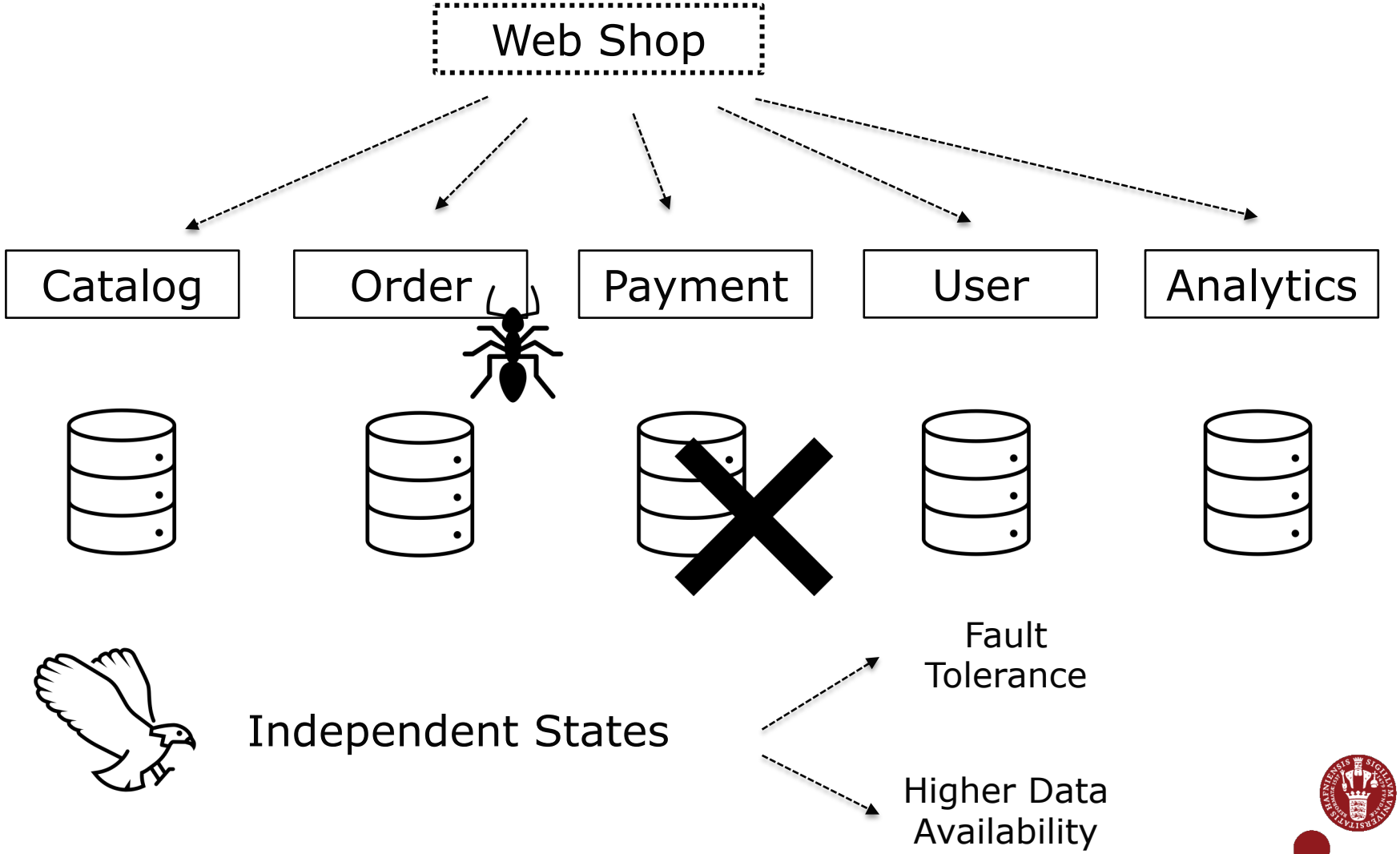
Towards MODB

- Devise a set of features to accommodate challenging requirements
- Making the case for microservice-oriented database systems



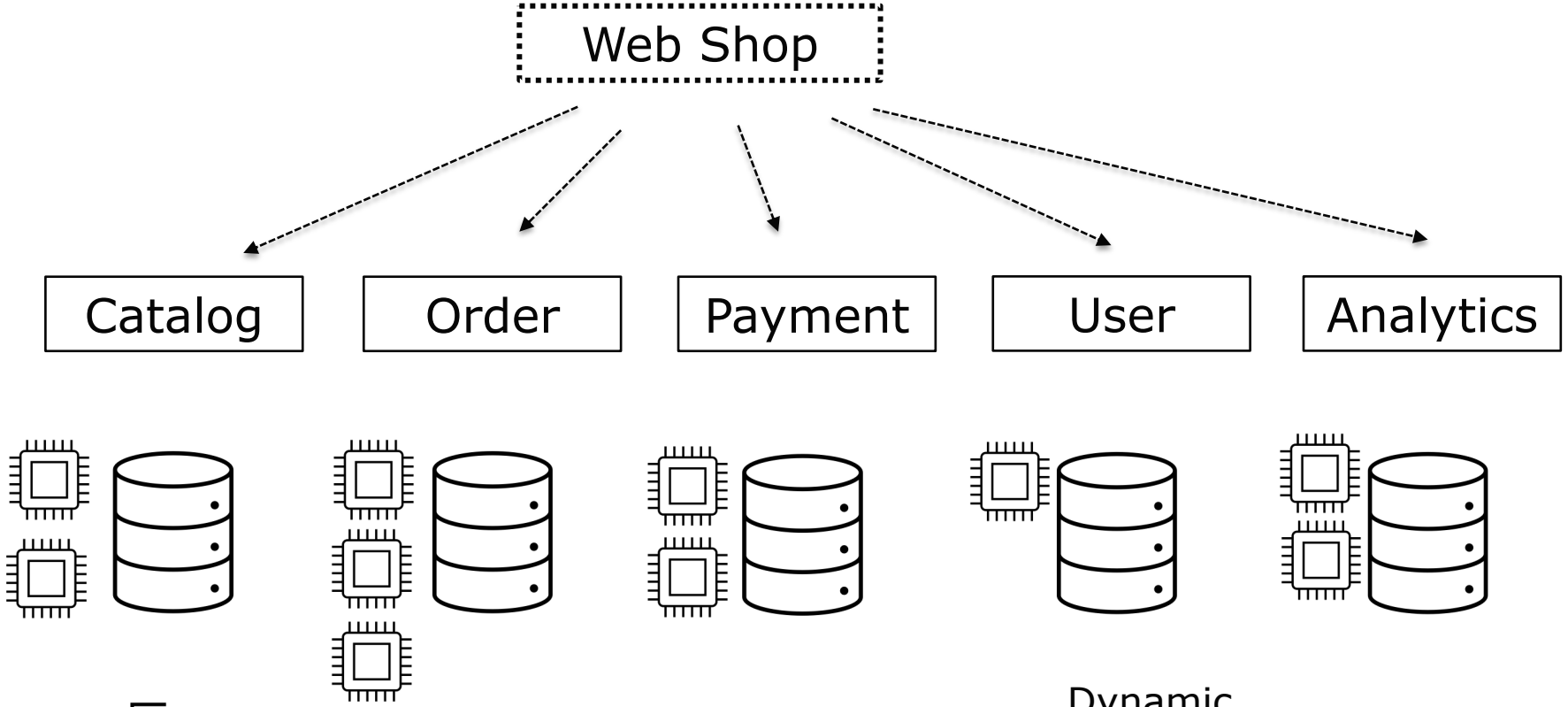
State of the Practice

Motivations



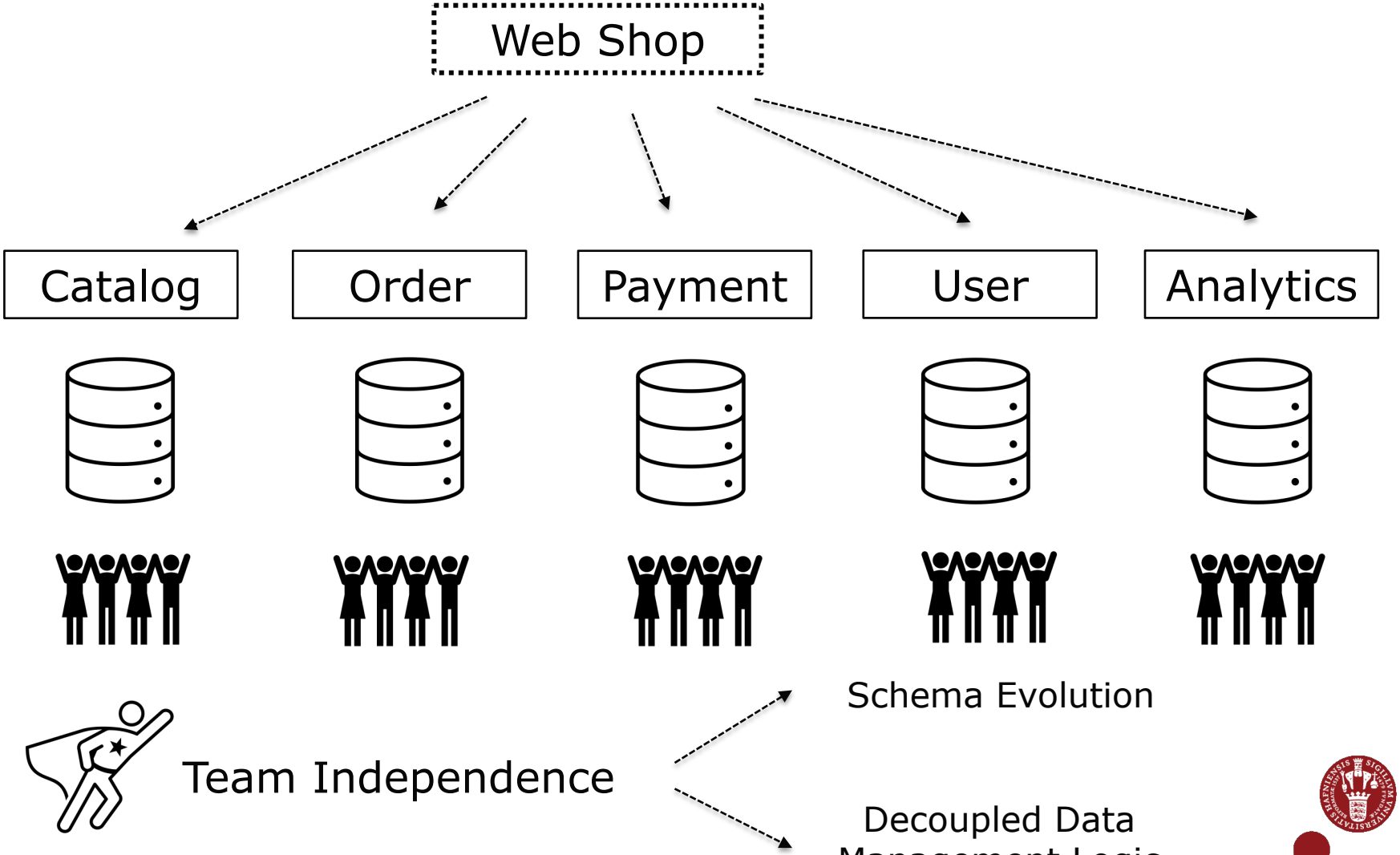
State of the Practice

Motivations



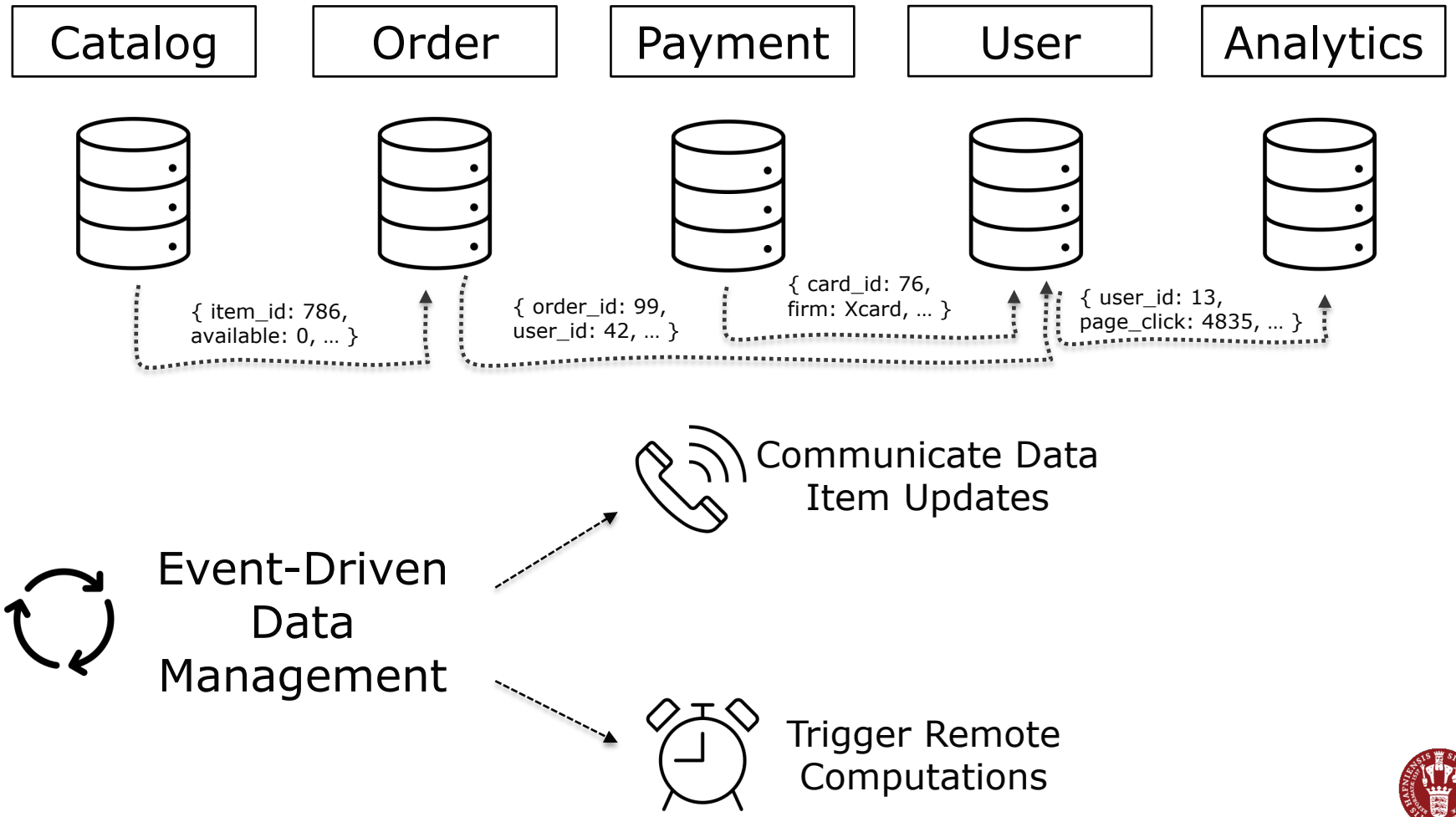
State of the Practice

Motivations



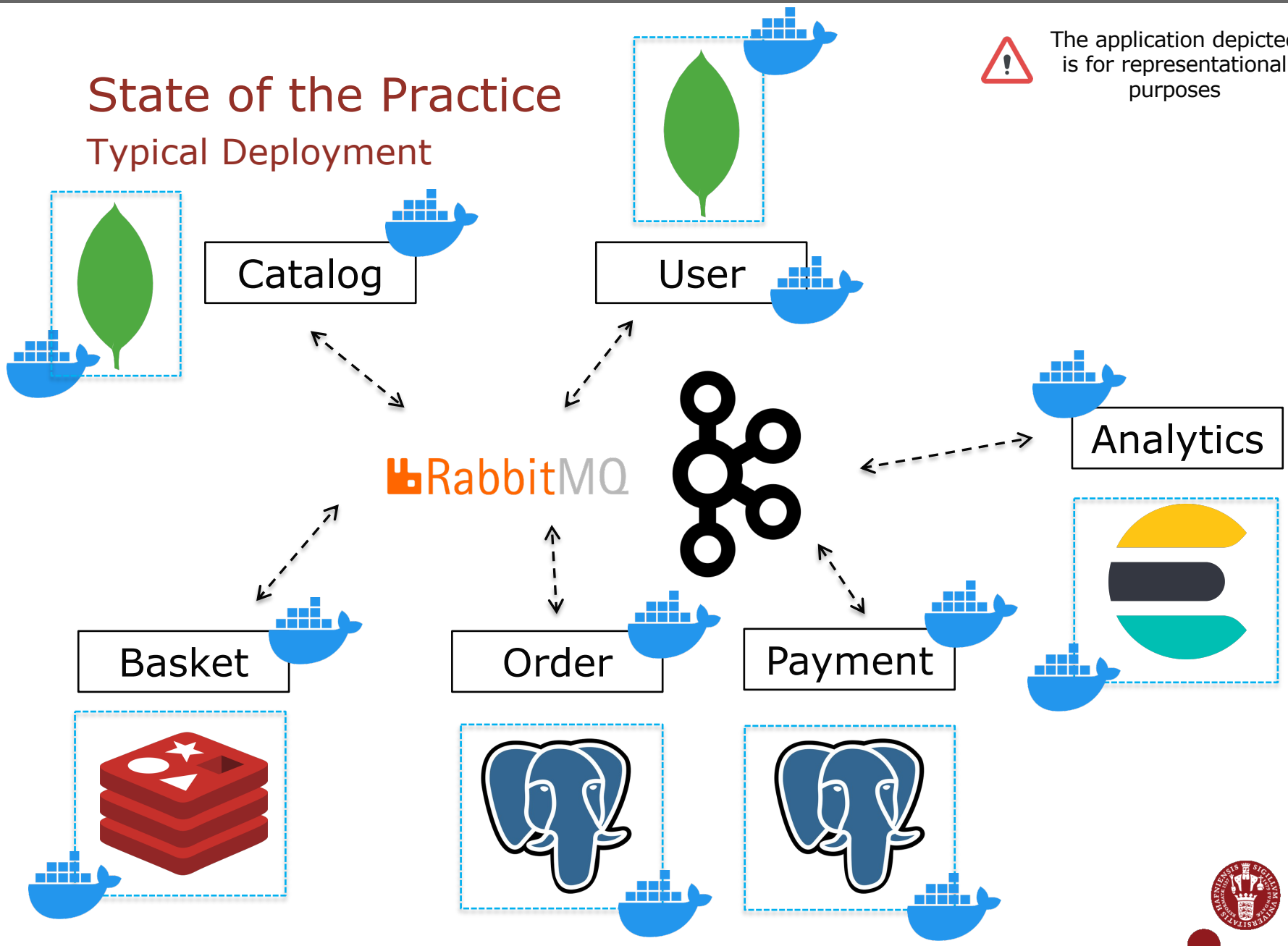
State of the Practice

Motivations



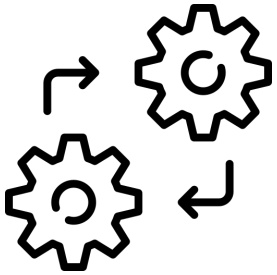
State of the Practice

Typical Deployment



Data Management Challenges

Cross-Microservice Concurrency Control



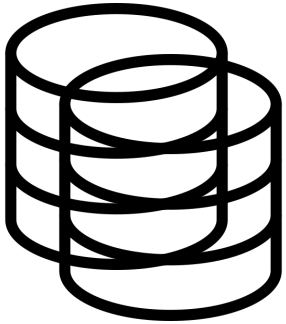
Cross-microservice **synchronizations** are often necessary (e.g., e-commerce)

Distributed commit protocols do not enjoy popularity. **Async** and **non-blocking** interactions are preferred

Developers face challenges on implementing application-level concurrency control and ensuring **correctness**

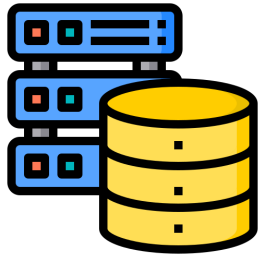
Data Management Challenges

Consistent Cross-Microservice Queries



Cross-microservice queries are often required (e.g., accessing multiple private states)

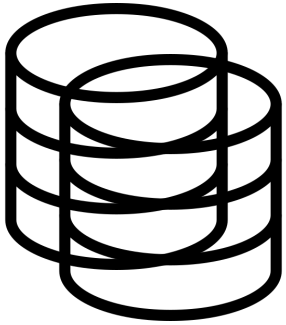
Developers often query and join distinct **private data** for online queries at the application layer



Implementing consistent and efficient data processing is challenging for developers

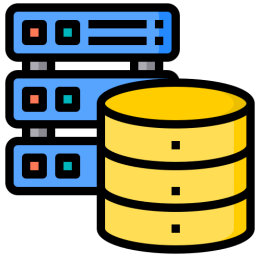
Data Management Challenges

Cross-Microservice Constraints



Cross-microservice constraints are sometimes required (e.g., referential constraint)

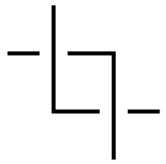
Developers enforce correctness at the application-level (i.e., avoiding "dangling" records)



Error-prone process, developers often eschew off constraint enforcement

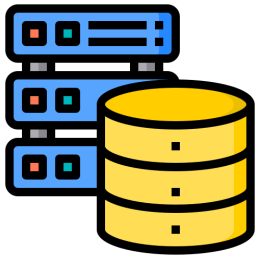
Data Management Challenges

Interleaving of Event Streams



Asynchronous events may impact (i.e., write to) a single or multiple private objects

Events are necessarily processed at the application level (where business logic often belongs to)



Interleaving of events may occur, leading to (often unknown) data anomalies

Towards Microservice-Oriented DBMS

What **features** should a futuristic database system provide?

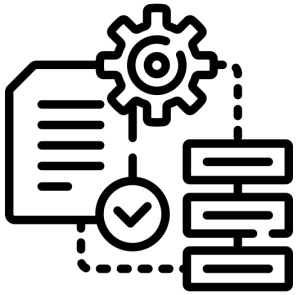
Event awareness (F1 & F2)

Cross-microservice synchronizations (F3 & F4)

Data access on distinct microservice states (F5-F8)

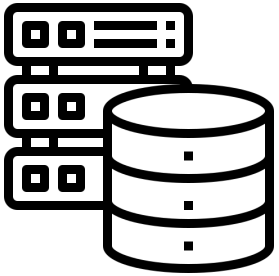
Proper database abstractions (F9)

Towards Microservice-Oriented DBMS



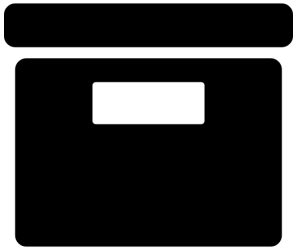
Meta-problem: Application is doing data management!

Which is considered challenging and harmful in some cases



Data management logic **should not be oblivious** to the database

Towards Microservice-Oriented DBMS

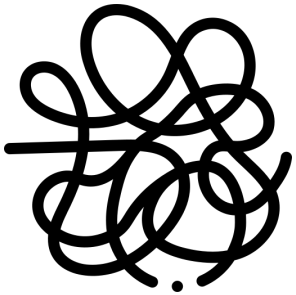


DBMSs designed to interact with

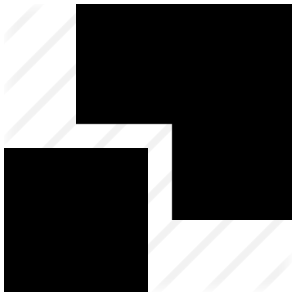
black box applications

DBMSs are oblivious of the **complex interplay** among microservices

Oblivious of the **data movements** (often through events) outside the DBMSs



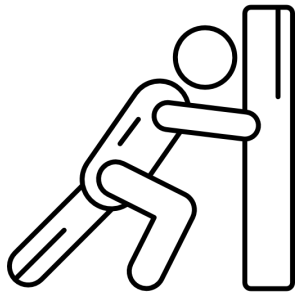
Towards Microservice-Oriented DBMS



Appropriate abstractions!

So, database systems play a **central role** again

Push data management tasks down to the database



Conclusion

- Microservices significantly deviate from traditional monolithic applications
- State-of-the-practice insufficiently meet developers' needs
- Leading them to **encode several pitfalls** in the application layer
- It is time to **rethink** how database systems are designed to tackle this new paradigm

Thank you!