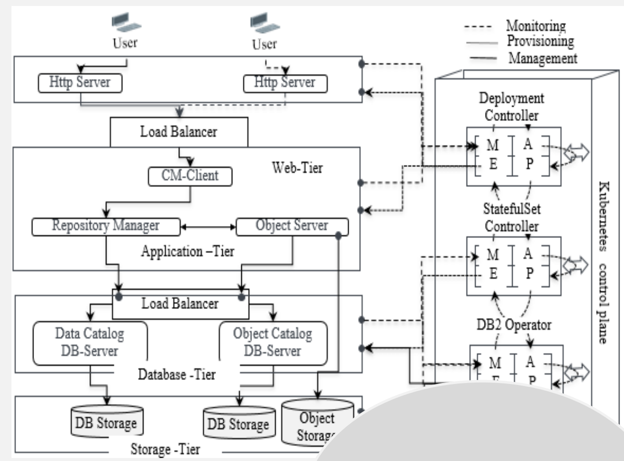
Universität Stuttgart
Institute for Parallel and Distributed Systems
Department: Applications of Parallel and Distributed Systems

Orchestrating Information Governance workloads as stateful services using the Kubernetes Operator Framework



Dipl.-Phys. Cataldo Mega

17th Symposium and Summer School On Service-Oriented Computing (SummerSoC) June 25th - July 1st, 2023

Agenda

- Introduction
- Problem Statement – What is the pain …
  - How to reduce the Information Governance as a cost burden
- Kubernetes Operator Framework
  - Automating Container Orchestration
- Orchestrating stateful IG workloads with Kubernetes
- IBM DB2 Operator Prototype
- Evaluation, Verification Tests & Results
- Conclusion

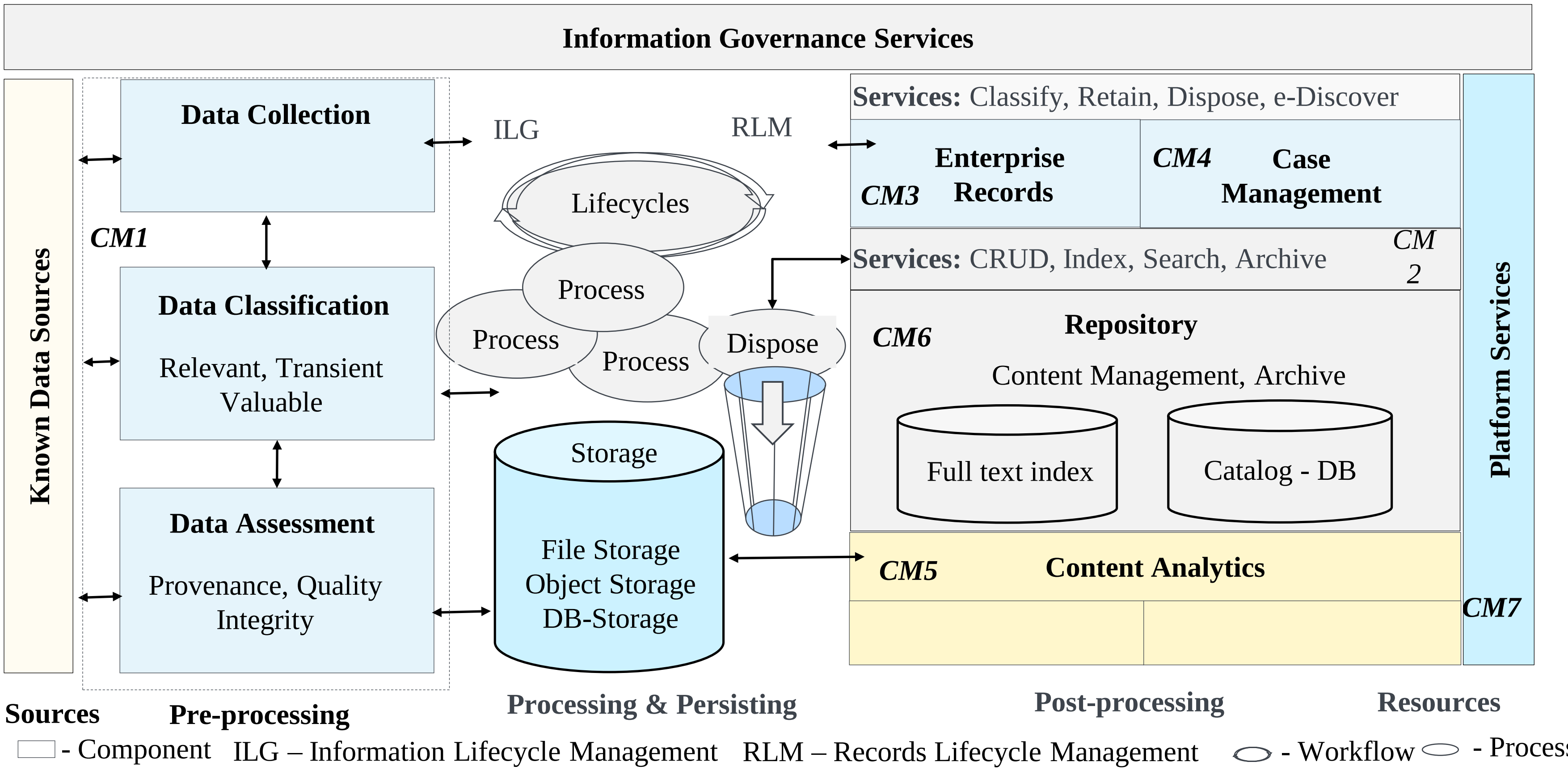
Introduction

- Information governance **(IG) mandates** that companies **preserve business control information** for a specific time, based on **corporate and regulatory compliance.**
- Consequences are:
  - IG requirements complement business requirements and are an additional cost factor
  - By definition IG solutions are complex and stateful – as we will see.

➢ This initiated the **"run into the Cloud"** among these customers

- **Cloud Solutions aim** at the following **Goals**:
  - Pay-as-you-go payment model (no investment capital, no maintenance costs)
  - Built-in **Scalability, H**igh **A**vailability **(HA)** and **D**isaster **R**ecovery **(DR)**
  - **Cost Effectiveness** through exploitation of **M**assive **M**ulti-tenancy, IT-**Elasticity** and **Process Automation**
- **But process automation comes at a cost**
  - ✓ **Transfer** of **Operator Knowledge & Experience** from **Humans to Infrastructure**
  - ✓ **Automating Container Orchestration** using the **Operator Pattern**

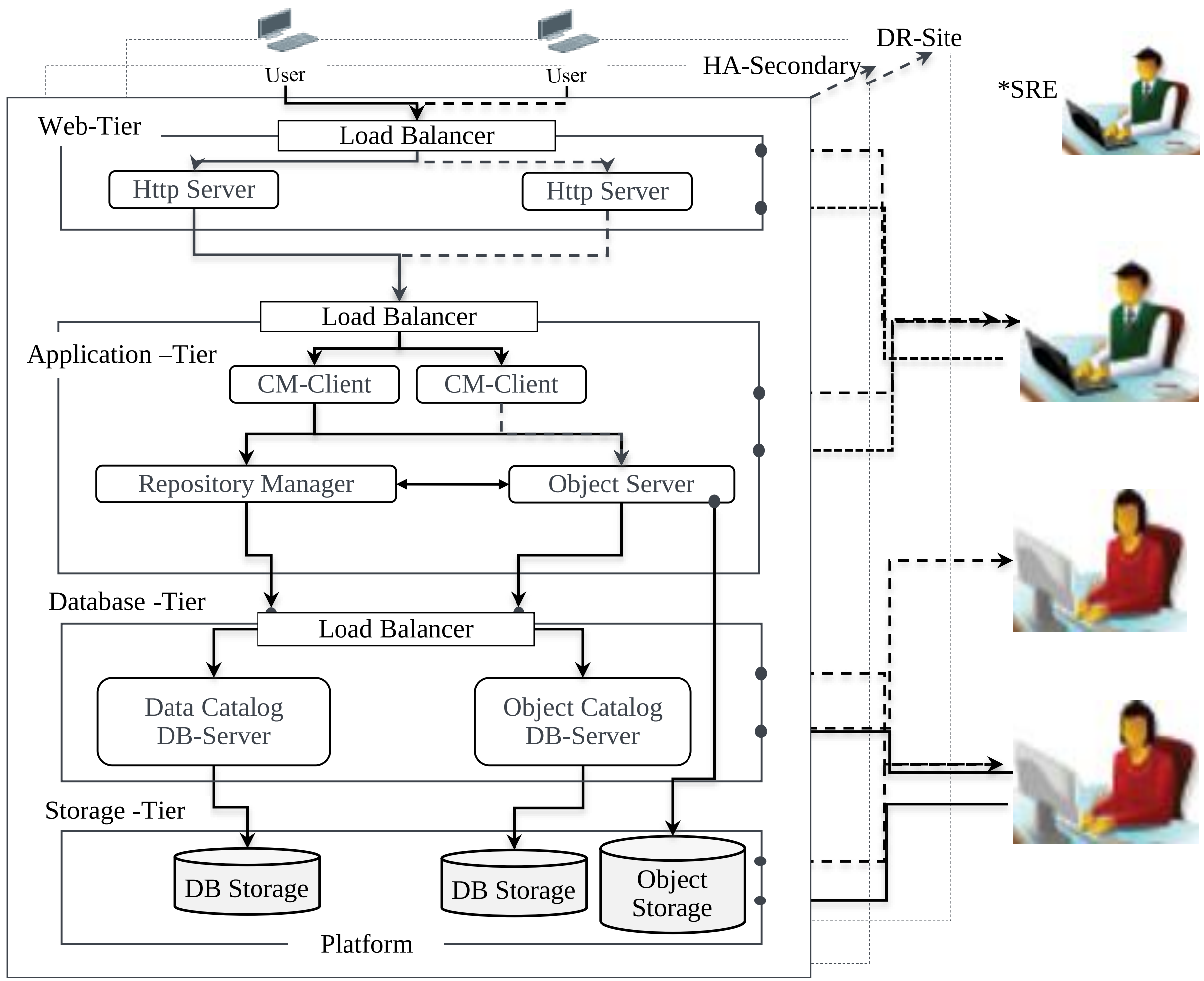
IG Solution Blueprint
IG component model outline



CM1: Data Services: Collection, Classification, Assessment and Ingest.
CM2: Content Services: Access, Index, Search, Retrieval, Security, Management.
CM3: Records Services: Classification, Retention, Disposition, Compliance.
CM4: Case Management Services: e-Discovery, Legal Data Requests, Holds.
CM5: Content Analytics: Classification, Statistics, Reporting.
CM6: Repository services: Information Retrieval, Catalog, Archive.
CM7: Platform Services: Compute, Storage, and Network.

IG Solution Deployment model

Employing scalability, HA and DR capabilities the bare metal way



- Clusters for each tier
- Tier specific cluster management
- Tier specific application lifecycle logic
- Site Reliability Engineering

*Site Reliability Engineering (SRE)

[1] ** SLA: Service Level agreement

Kubernetes Background

- Automating Container Orchestration started around 2015 at **Google** with Kubernetes a container orchestrator given to the **Cloud Native Computing Foundation** (CNCF) and eco-system.
- Kubernetes would originally automate the lifecycle of a stateless application
  - by simply replacing failing app-instances with an identical replica.
  - would not work for stateful applications which were left out of the cloud platform.
- The Operator pattern emerged at **CoreOS then acquired by Red Hat**
  - Problem: automating orchestrations of increasingly complex applications on K8s clusters, including managing Kubernetes itself.
  - Operator work continued at **Red Hat** 2018 Red hat released the open source Operator Framework and SDK.
  - By design an Operator extends K8s to automate the management of the lifecycle of stateful applications providing means to distribute, monitor, maintain, recover, and upgrade the deployed applications through K8s APIs.

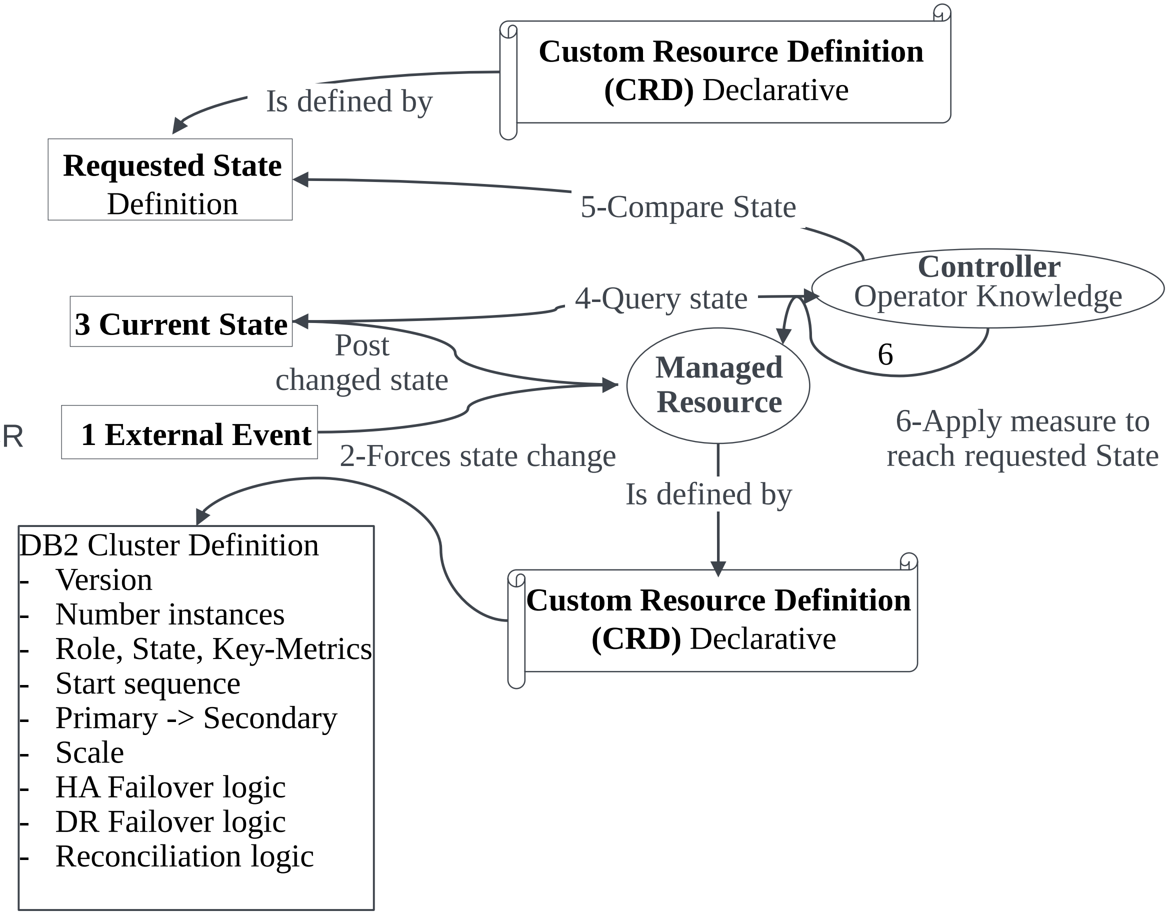
Kubernetes Operator Pattern

Custom Resource Definition (CRD) & Custom Resource (CR)

Kubernetes Operator Foundations Entities

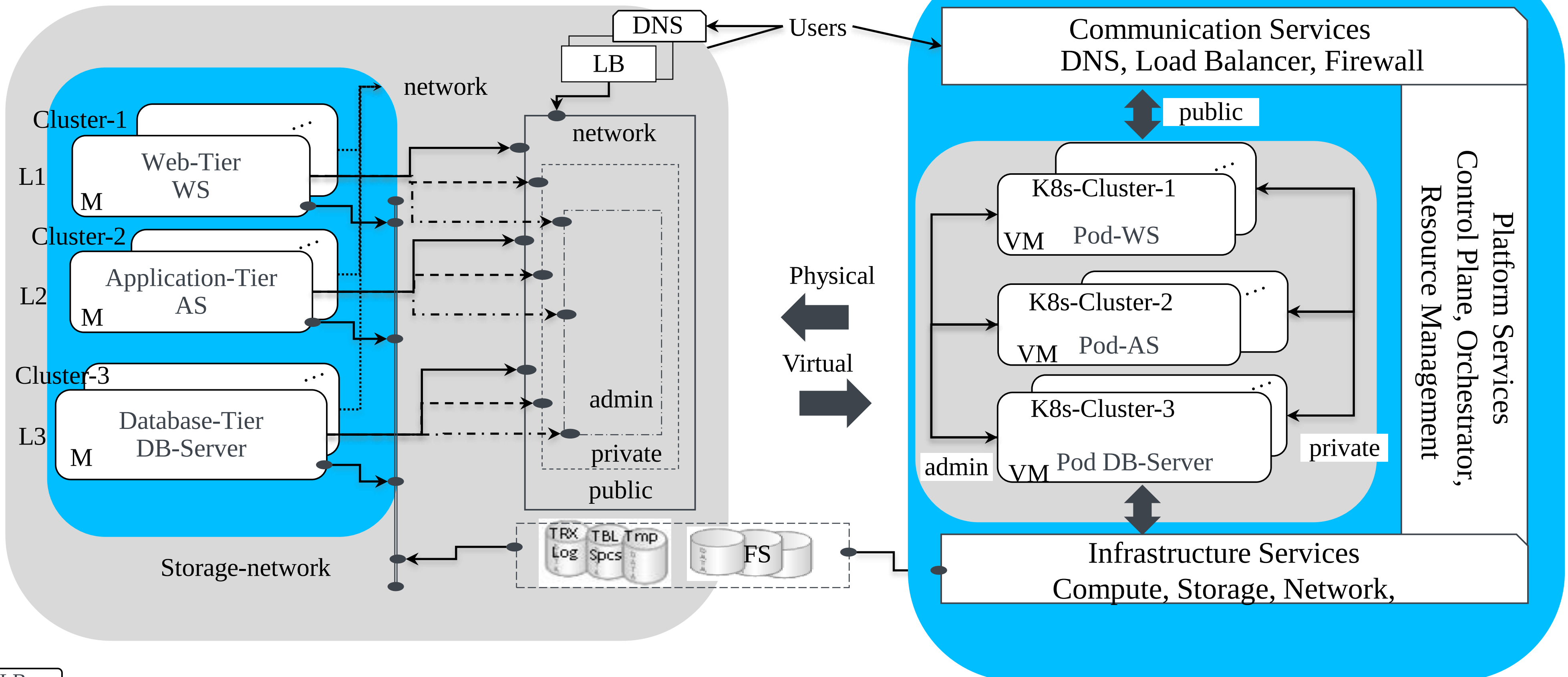
- Kubernetes Operator
  - **Custom Resource (CR) … managed resource.**
    - Extend Kubernetes API with additional types
  - **Custom Resource Definition (CRD)**
    - Define a schema of configuring CR
  - **Controller**
    - Compare the desired state with the current state
    - Apply changes to Kubernetes objects that make up CR

Kubernetes Operator Pattern Outline



IG Solution

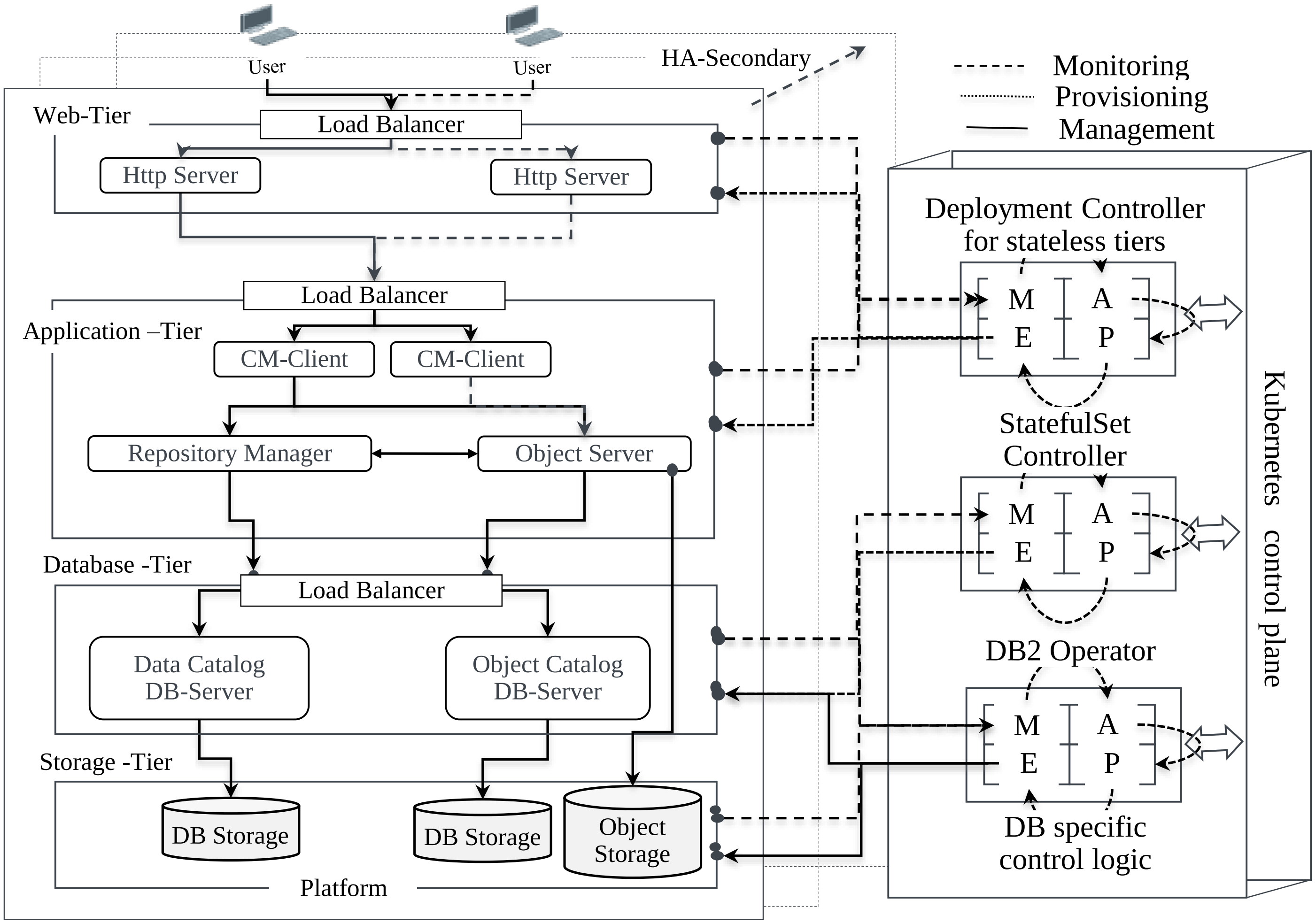
Physical 2 Virtual Mapping



- LB - Physical Load Balancer
- WS - Bare Metal Web-Server
- AS - Bare Metal App-Server
- DB-Server - Bare Metal DB-Server
- Physical Storage-Server
- Physical Network Segments
- Communication Links
- DNS - Domain name Service
- M - Physical Machine
- L - Topology Layer/Tier
- FS - Filesystem Access
- Https - Network Access

IG Solution Deployment Model

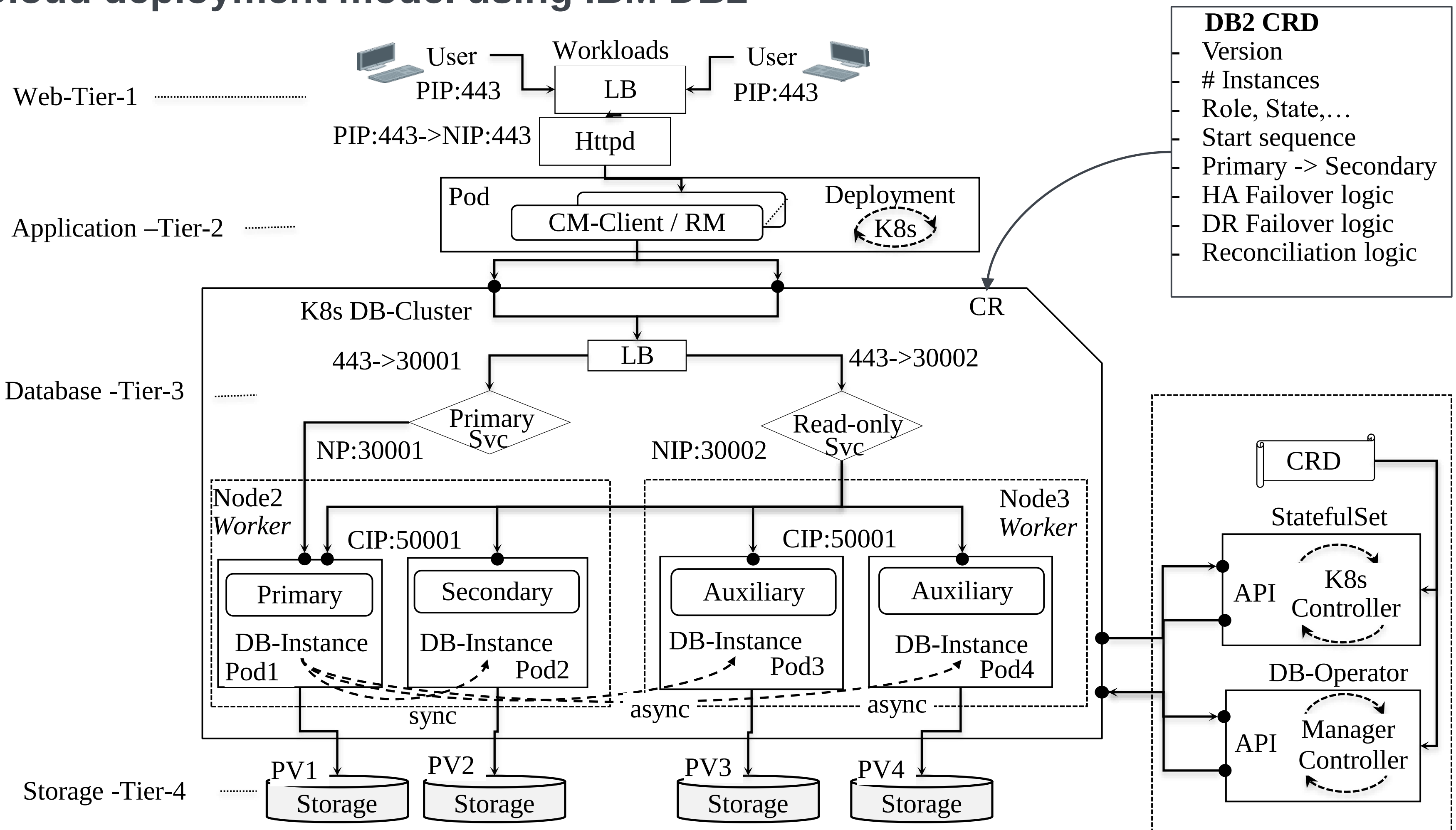
Employing scalability, HA and DR capabilities the Kubernetes way



Universität Stuttgart

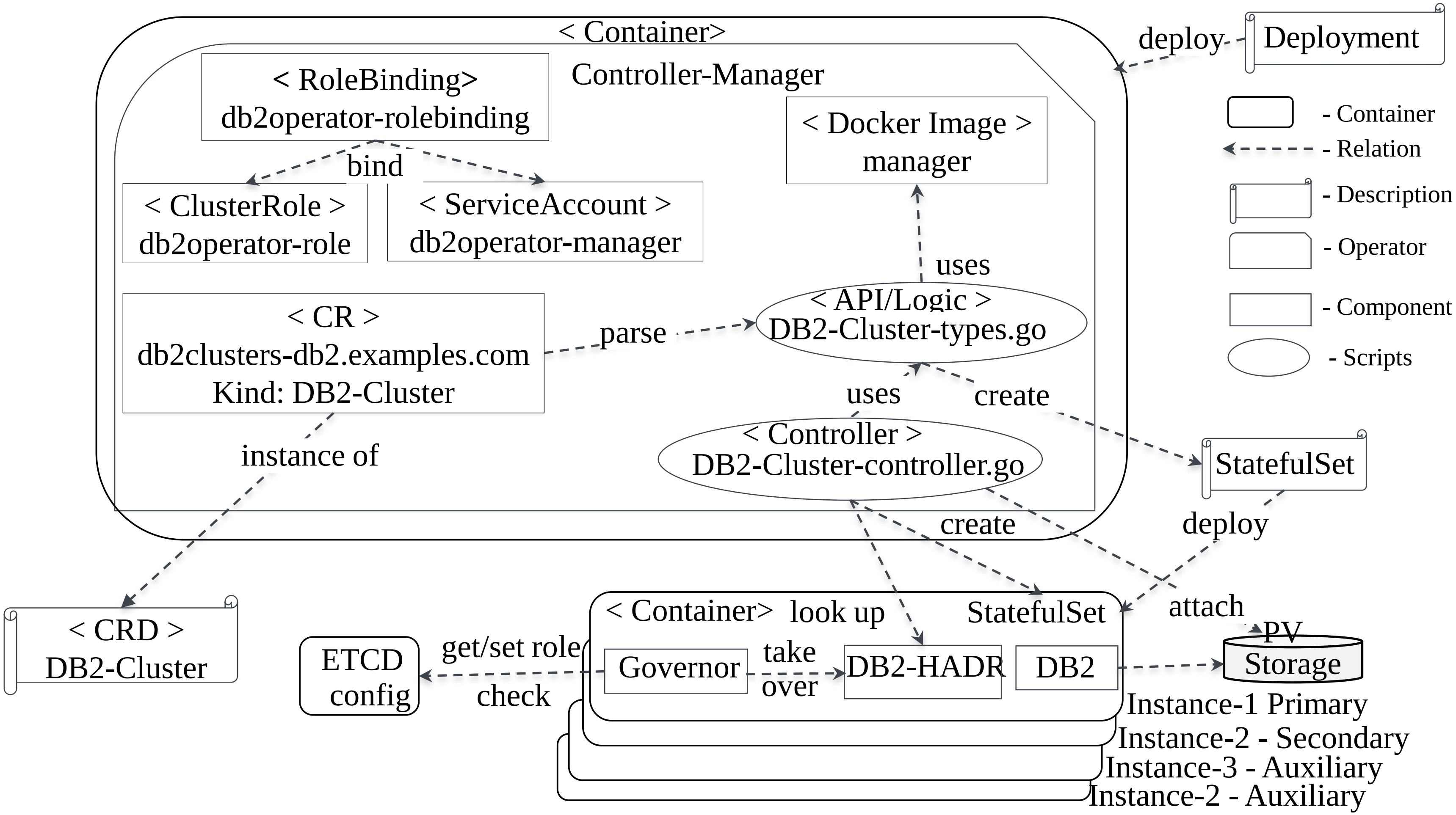
IG Solution

Cloud deployment model using IBM DB2



Kubernetes DB2 Operator Outline

Component Model



Test Scenarios
DB2 Cluster Prototype

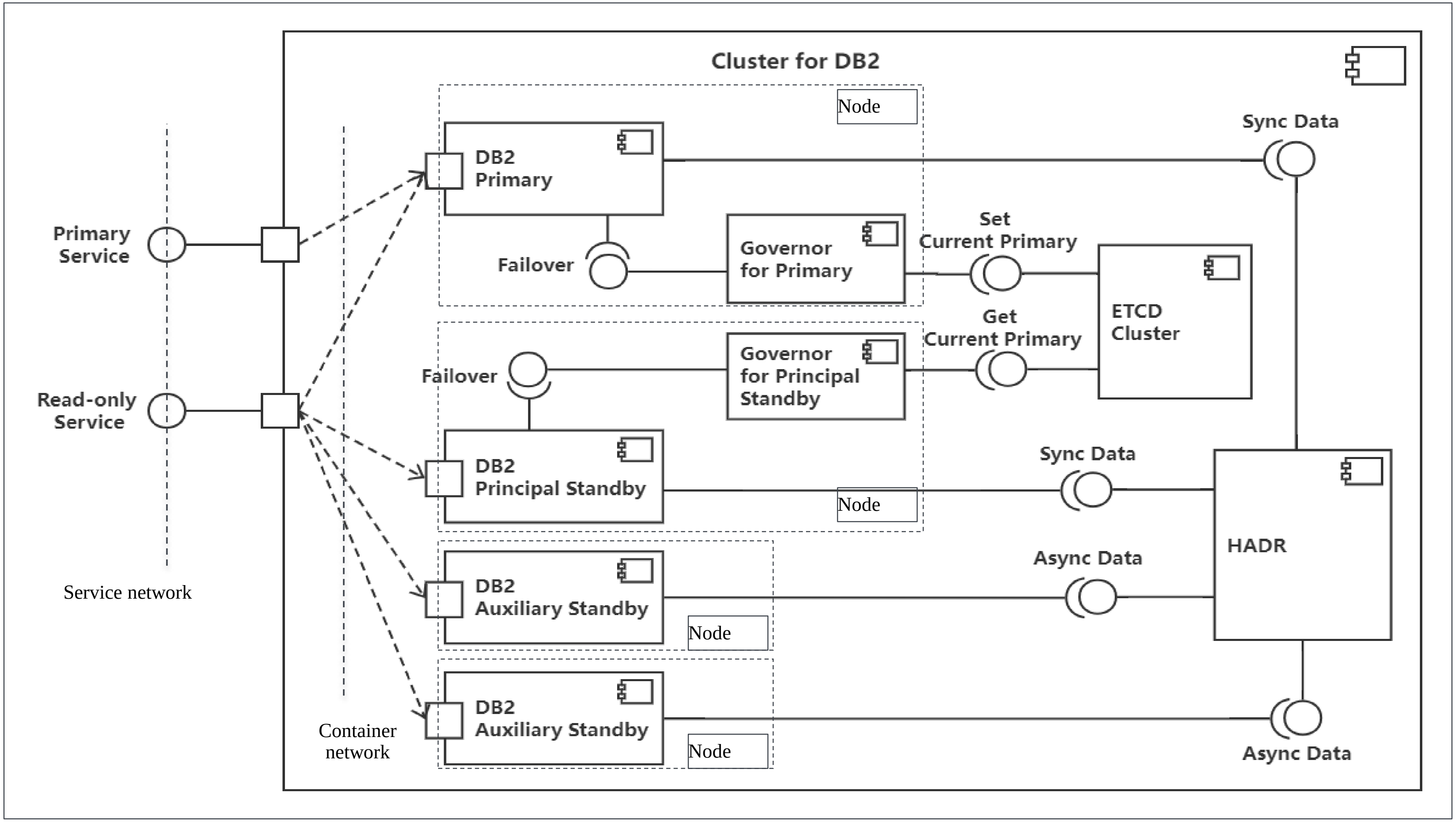
- High Availability (HA)
- Read Scalability
- Disaster Recovery (DR)

- Note: Test results show indicative figures only.

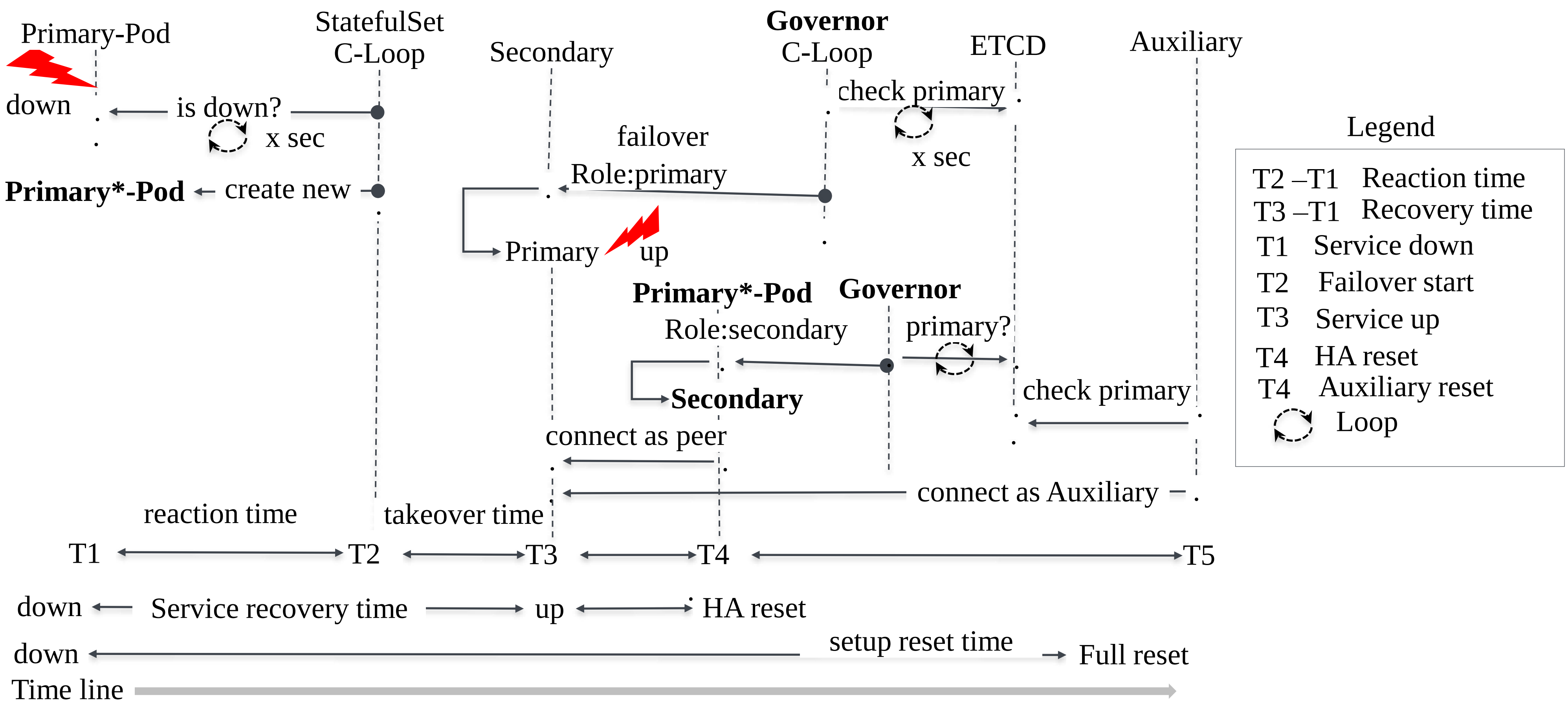
Universität Stuttgart

IBM DB2 Instance Cluster

Cloud component model



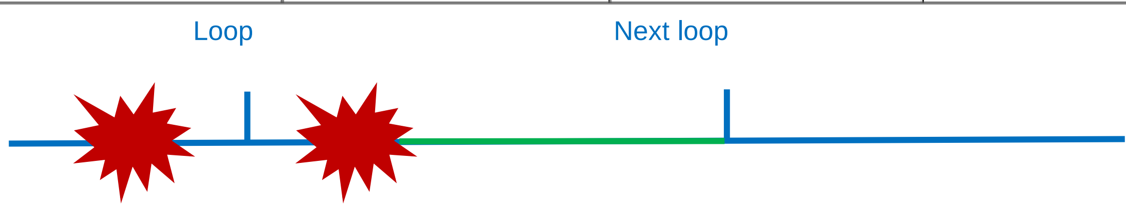
Test Scenarios:
High Availability (HA)*



Test Scenarios

High Availability (HA)*

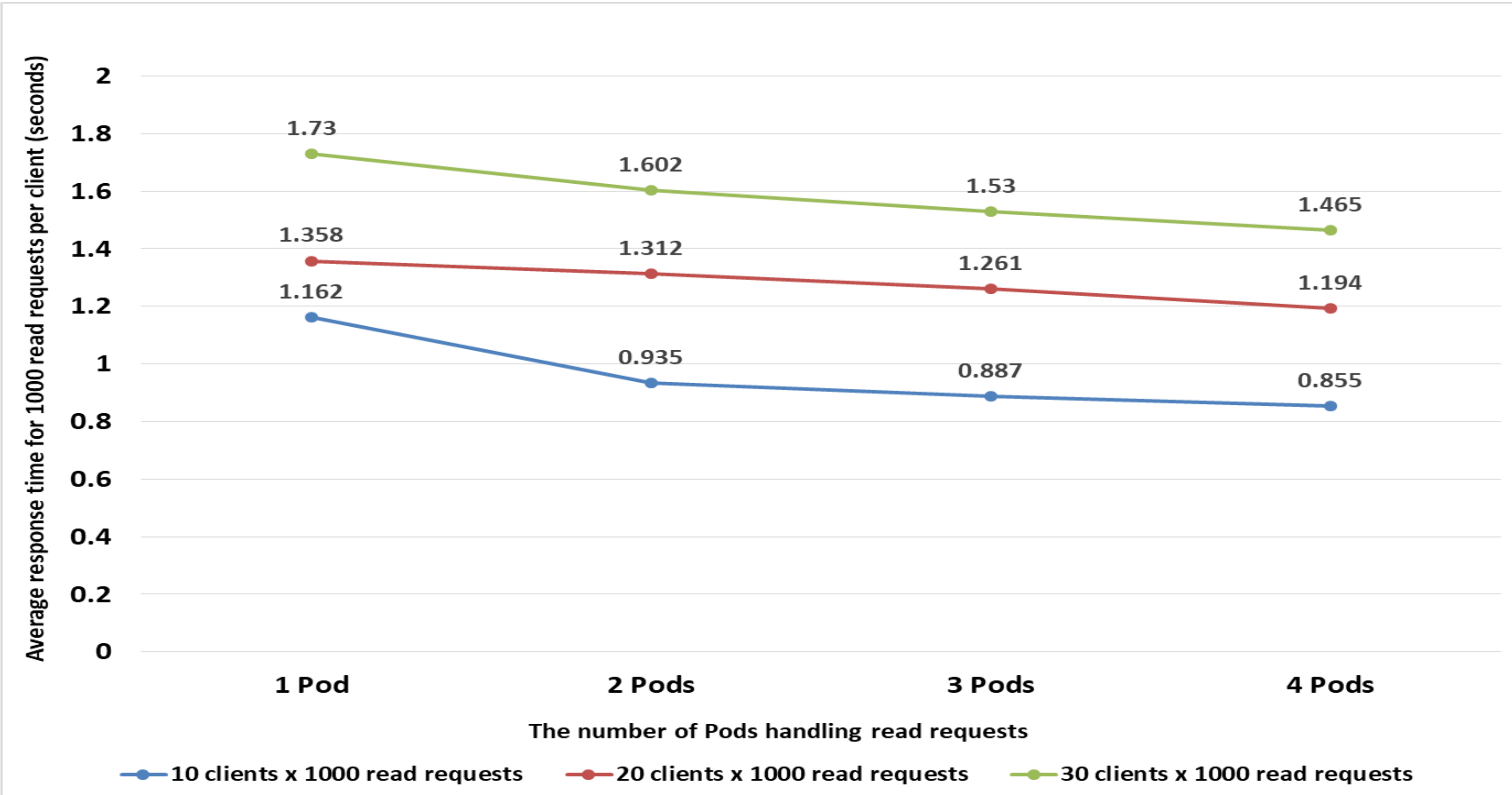
(Unit: seconds)	Reaction Time	Failover Time	Service Outage Time
1	2.859	3.621	6.425
2	1.995	4.322	6.839
3	2.403	9.655	18.180
4	2.066	5.856	13.793
5	2.022	36.309	41.639
6	2.051	9.632	14.555
7	1.720	4.839	9.570
8	2.058	29.886	32.728
9	2.624	9.679	18.111
10	2.059	33.286	34.581
Average	**2.186**	**14.709**	**19.642**



*Note: Test results show indicative figures only.

Test Scenarios
Read-only Workload Scalability*

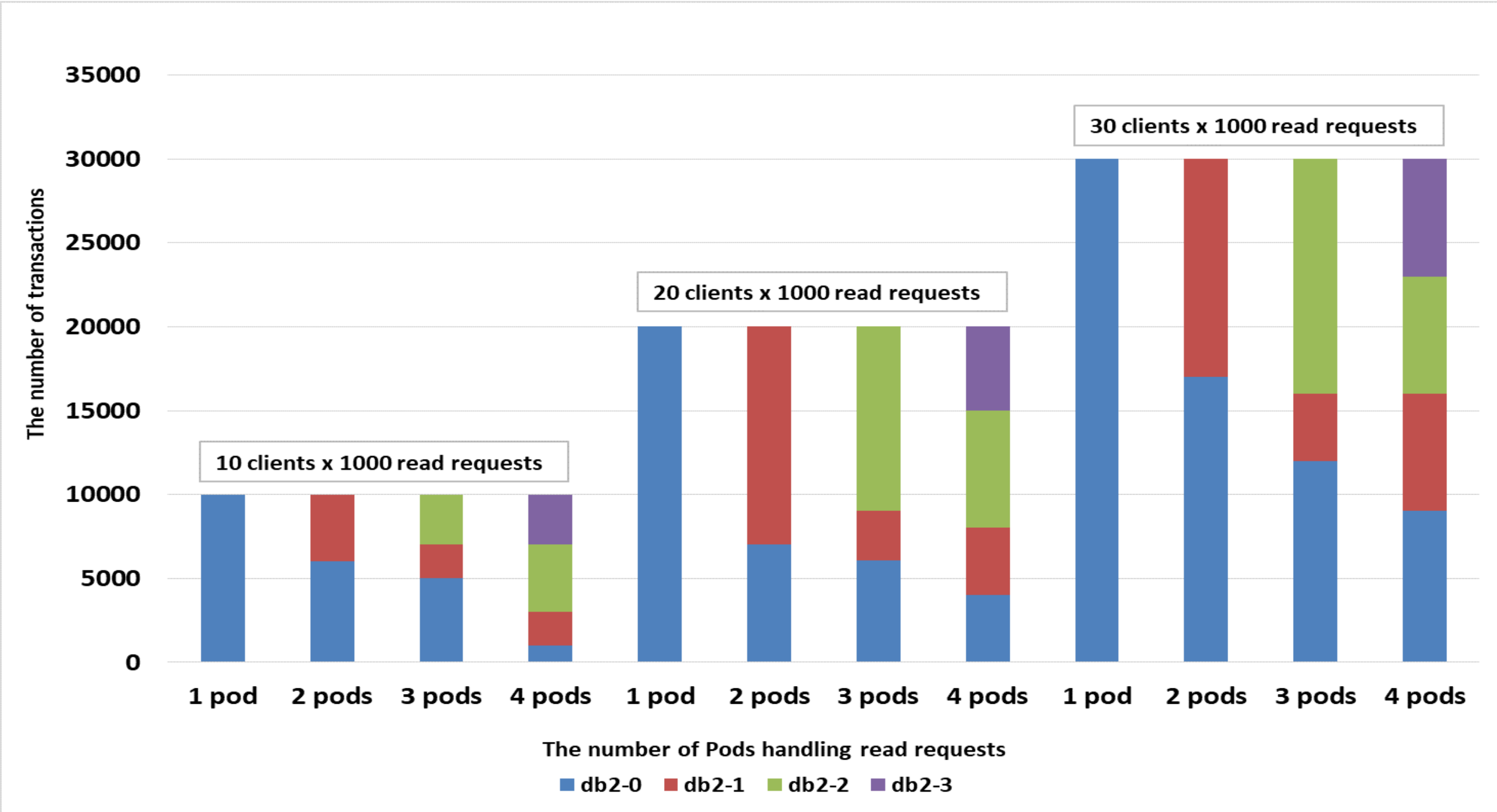
Average read response time by number of users and DB-Instances.



*Note: Test results show indicative figures only.

Test Scenarios
Read-only Workload Scalability*

Read request distribution over the 4 DB2 Instances (Pods)



*Note: Test results show indicative figures only.

Universität Stuttgart

Summary & Conclusions

- ✓ Kubernetes provides an efficient cluster control mechanism that allows dynamic topology changes based on workload demand for stateless applications.
- ✓ Stateful services Kubernetes requires substantial domain/application specific knowledge to be integrated through the K8s operator framework.
- ✓ Eat your own cooking does not apply leave it to the pro's.
- ✓ Migrating legacy solutions into the cloud requires to replace traditional component with new cloud native substitutes where possible. I.e. Traditional RDMBS vs. cloud native RDBMS or equivalent DAAS service.

Universität Stuttgart

Thank you !

Cataldo Mega

e-mail: cataldo.mega@ipvs.uni-Stuttgart.de
phone: +49 (0) 711 685-88 441
www: https://www.ipvs.uni-stuttgart.de/institute/team/Mega/
University of Stuttgart
Institute for Parallel and Distributed Systems
Department: Applications of Parallel and Distributed Systems
Universitätsstraße 38
D-70569 Stuttgart

References

1. X. Wang, Orchestrating data governance workloads as stateful services in cloud environments using Kubernetes Operator Framework, University of Stuttgart, MA Thesis, 2022.
2. C. Trybek. “Investigating the Orchestration of Containerized Enterprise Content Management Workloads in Cloud Environments Using Open Source Cloud Technology Based on Kubernetes and Docker”. University of Stuttgart, MA Thesis, 2021.
3. P. Hagemann, Evaluating dynamic load balancing of ECM workload pattern employed in cloud environments managed by a Kubernetes/Docker eco-system[D]. University of Stuttgart, MA Thesis,. 2021.
4. J. S. Omer Kahani, A. J. Thomas Schuetz, G. G. Hongchao Deng, J. K. Noah Kantrowitz, D. M. Philippe Martin, C. S. Roland Pellegrini. CNCF Operator WHITE PAPER. Tech. rep. July 2021.
5. J. Dobies and J. Wood. Kubernetes Operators; Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. ISBN: 978-1-492-04805-3. Copyright © 2020 Red Hat, Inc. All rights reserved.,
6. Dynamic Provisioning of System Topologies in the Cloud. Ritter, T., Mitschang, B., Mega, C. (2012). In: Poler, R., Doumeingts, G., Katzy, B., Chalmeta, R. (eds) Enterprise Interoperability V. Proceedings of the I-ESA Conferences, vol 5. Springer, London. https://doi.org/10.1007/978-1-4471-2819-9_34
7. C. Mega, T. Waizenegger, D. Lebutsch, S. Schleipen und J. M. Barney, „Dynamic cloud service topology adaption for minimizing resources while meeting performance goals,“ IBM J. Res. Dev., Bd. 58, 2014.