ULB UNIVERSITÉ LIBRE DE BRUXELLES

BDMA

CentraleSupélec

université PARIS-SACLAY

Self-scalable Moving Object Databases on the Cloud: MobilityDB and Azure

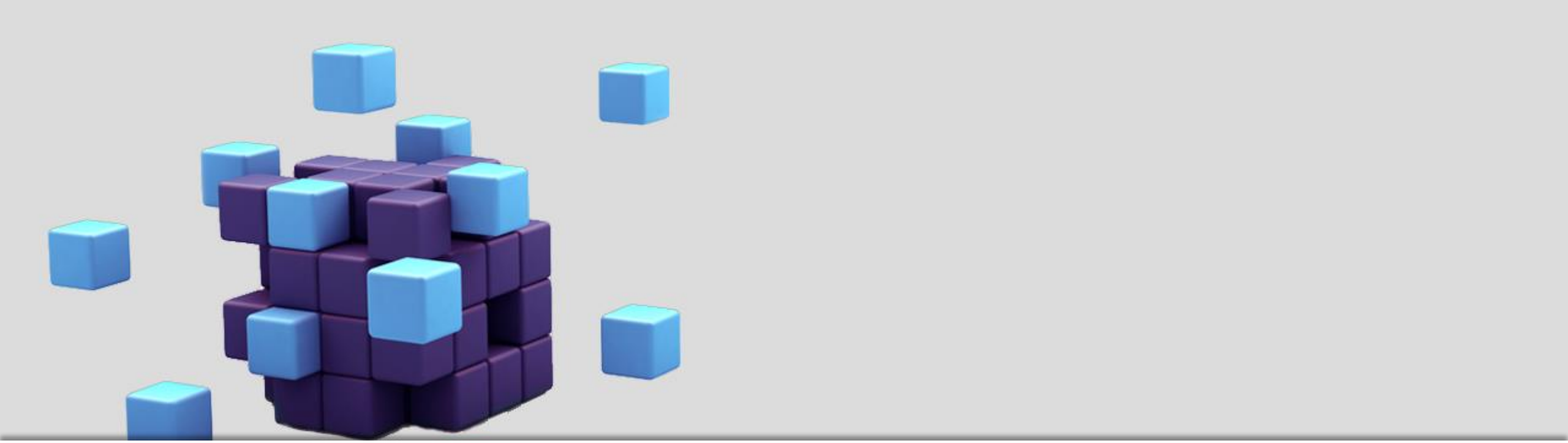
2nd September 2021

Dimitrios Tsesmelis

Advisors:

Esteban Zimányi

Mahmoud Sakr



Motivation & Technical Ecosystem

Moving object data

Data produced by **GPS traces**
Typical examples: fleets of cars, ships and airplanes
Interesting for a wide range of applications (*e.g., fuel consumption reduction of ships*)

Why management of moving data is difficult

Moving data engage both **space** and **time** dimensions

Discrete point representation using relational DBMS

Need for a new DBMS with more **data types** and **operations**

MobilityDB [1]

MobilityDB

tgeompoint, tgeogpoint,
tint, tfloat, ttext, tbool

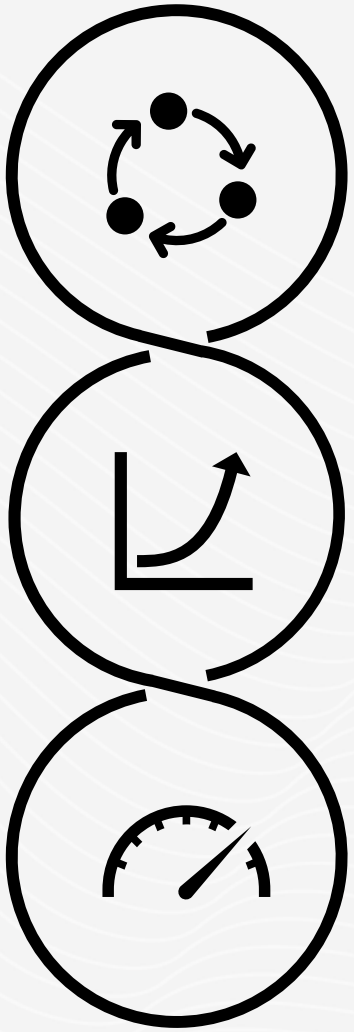
PostGIS

geometry, geography

PostgreSQL

numeric, monetary, character,
data/time, boolean, enum,
arrays, range,
XML, JSON, ...

Our goal



Automation

Exploiting the capabilities of Infrastructure-as-Code and Kubernetes to enable the **automatic deployment** and **management** of the MobilityDB cluster.

Scalability

Move from an on-premises, single-node PostgreSQL server to a **cloud and fully distributed PostgreSQL cluster**.

Elasticity

Applying **autoscaling techniques** to let the system decide **when** and **how** it will scale, according to the observed database workload.

Moving MobilityDB on the cloud

PostgreSQL-as-a-Service solutions

1. MS - Azure Single/Flexible Server
2. AWS - Relational Database Service
3. Google Cloud Platform - Cloud SQL

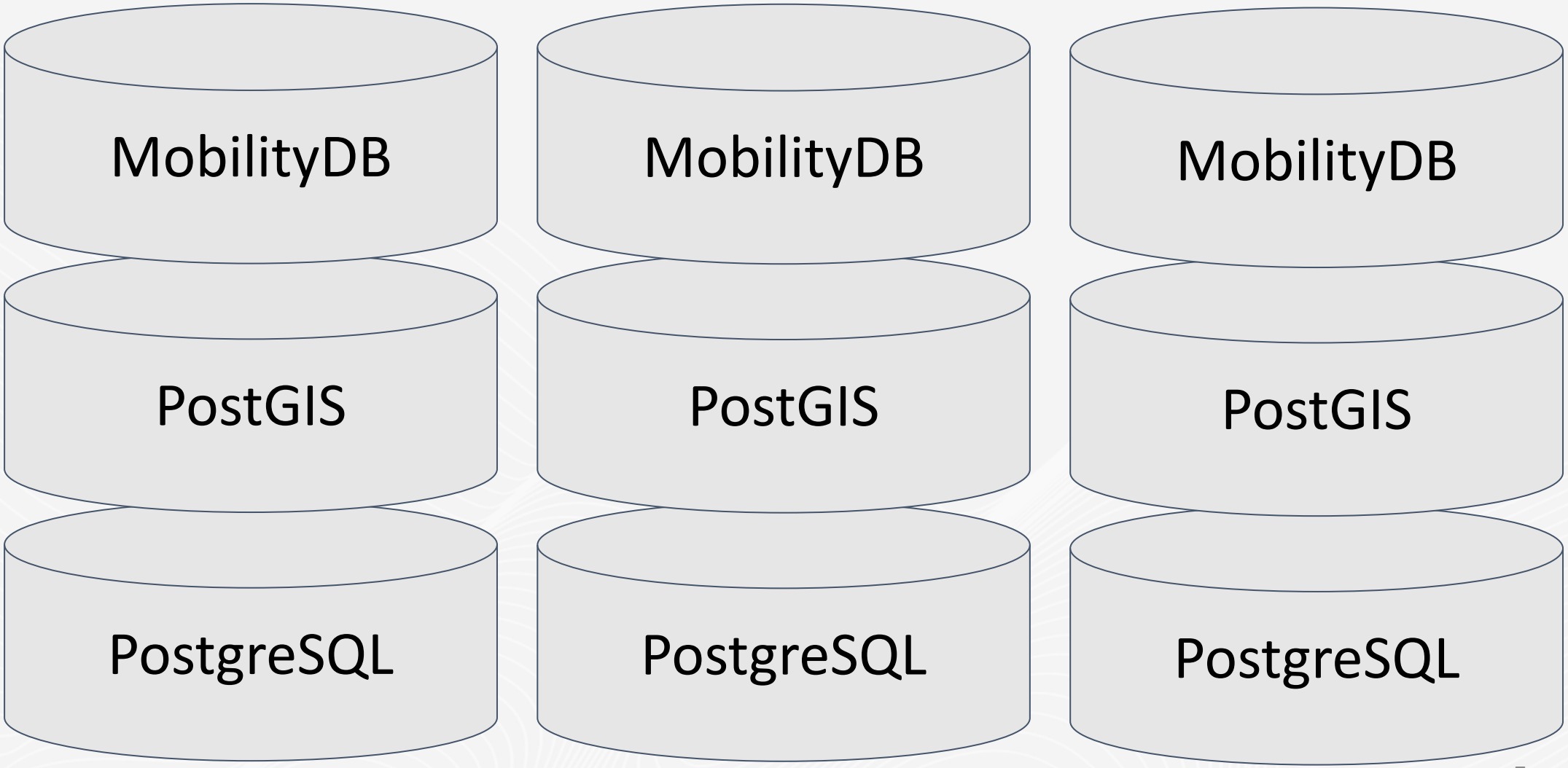
Open-source solutions

1. EnterpriseDB Cloud Native PostgreSQL
2. Crunchy PostgreSQL Operator
3. StackGres Operator

Key features

√	High availability
√	Scheduled maintenance
√	Automated version updates
√	More…
X	No data distribution

MobilityDB at scale [2, 3, 4]



Citus [5]

A PostgreSQL extension

A way to **scale out** PostgreSQL

- Using sharding and **table distribution**
- Parallelizing SQL query execution plans

Citus architecture

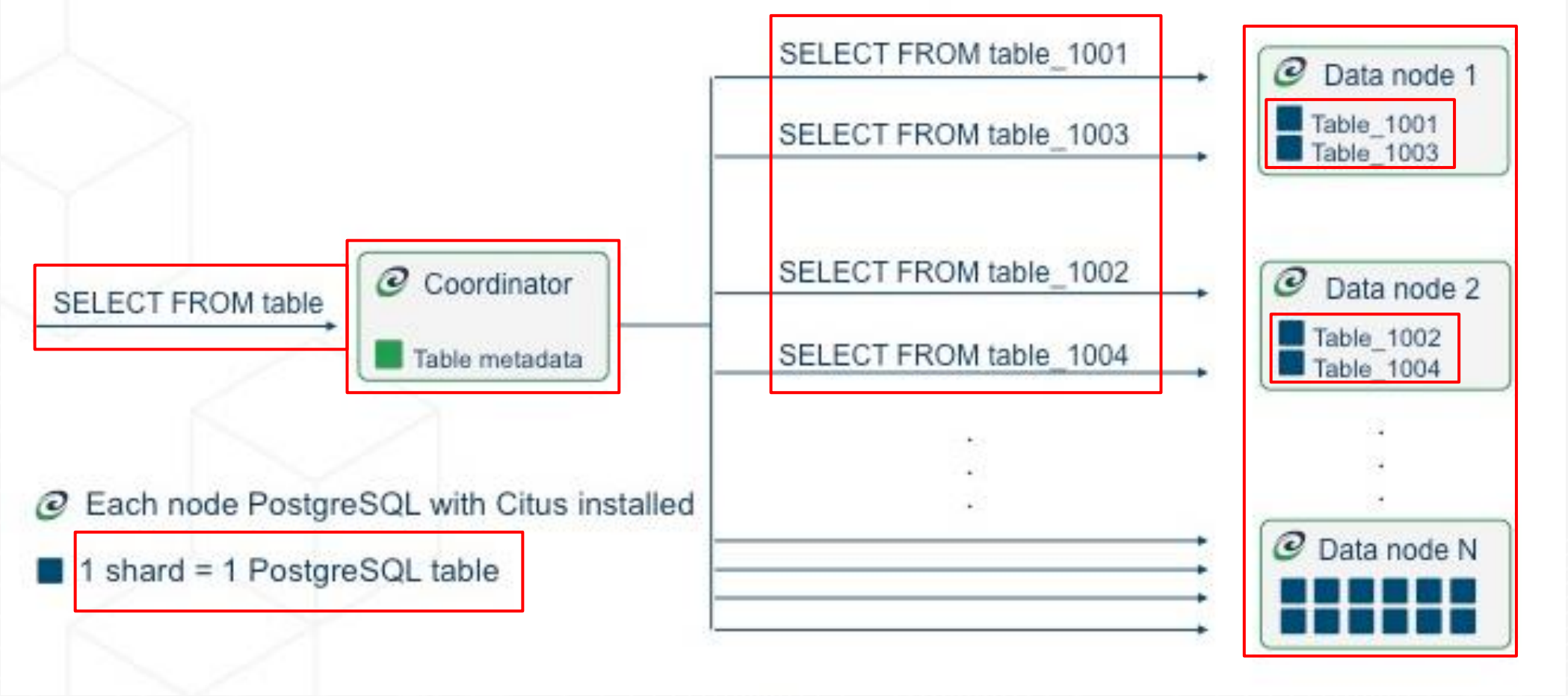


Image taken from Citus documentation

Kubernetes

Kubernetes

A **container orchestration** tool that automates the:

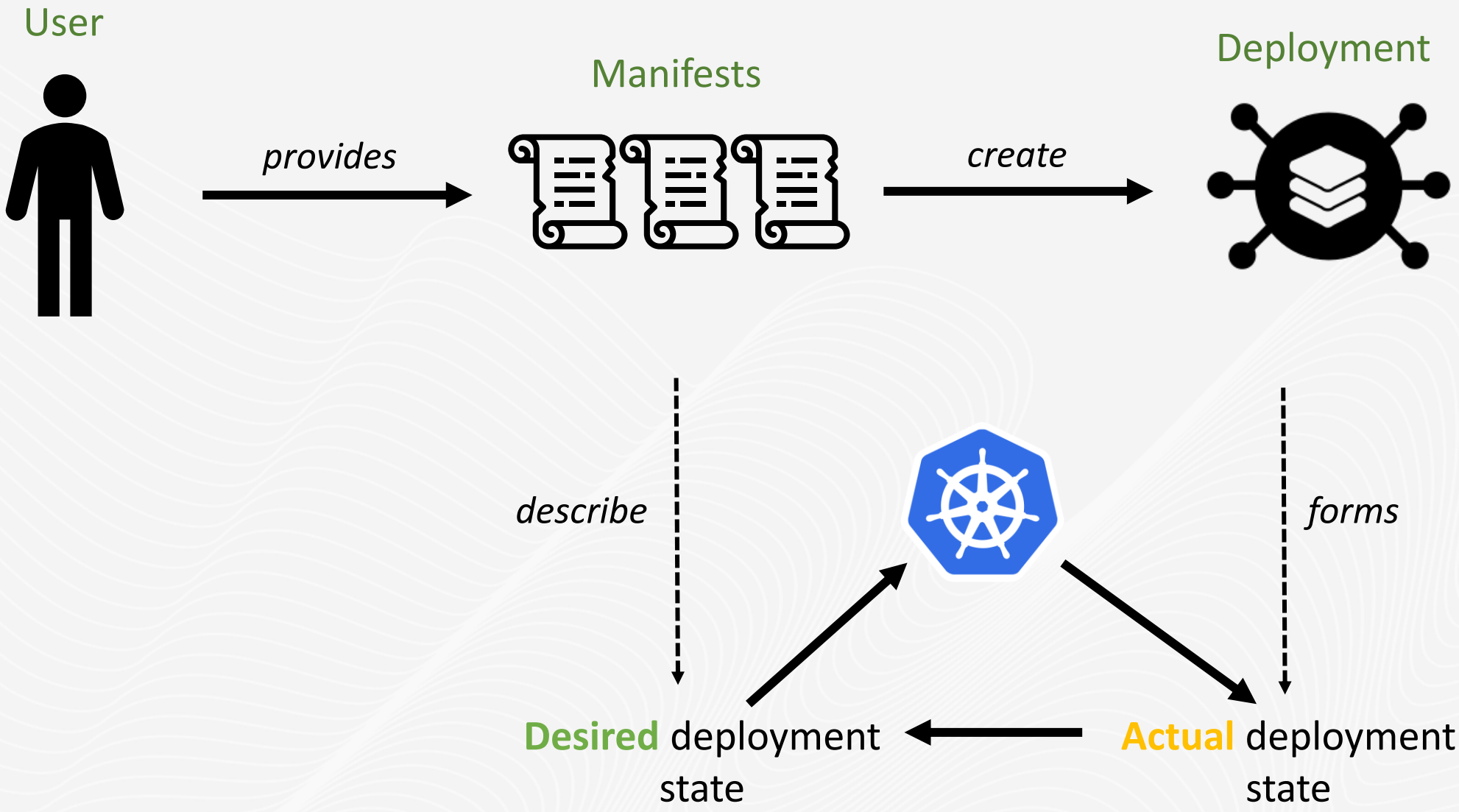
- Deployment
- Management
- Scaling

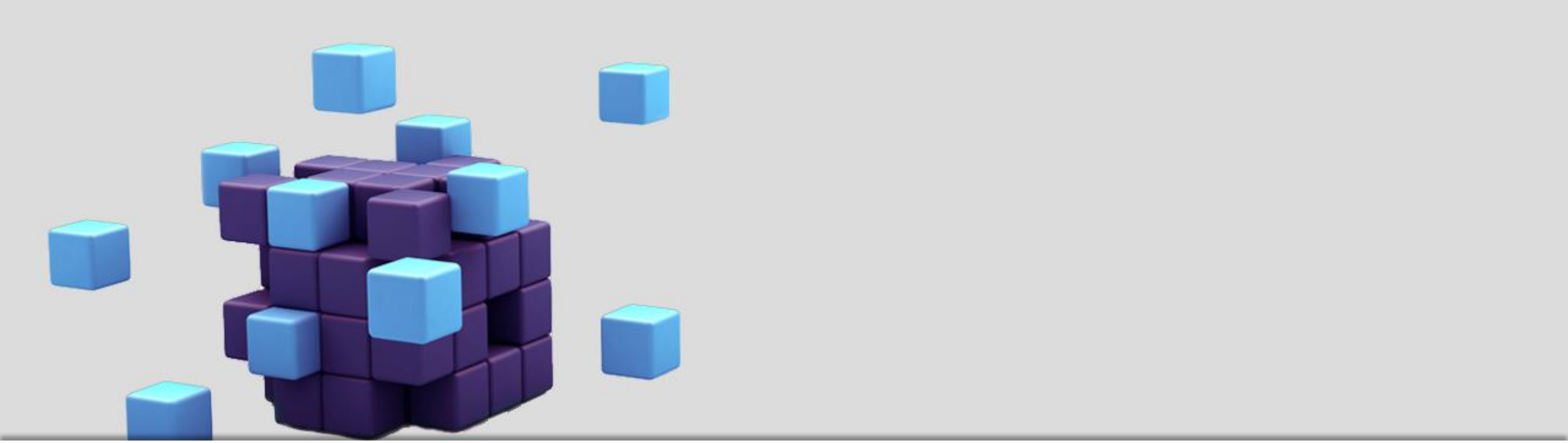
of containerized applications

Key features

- Self-healing
- Load balancing
- Automated updates
- Secret management

How Kubernetes works





Cloud Database Automation and Management

Automating the cluster initialization

MobilityDB

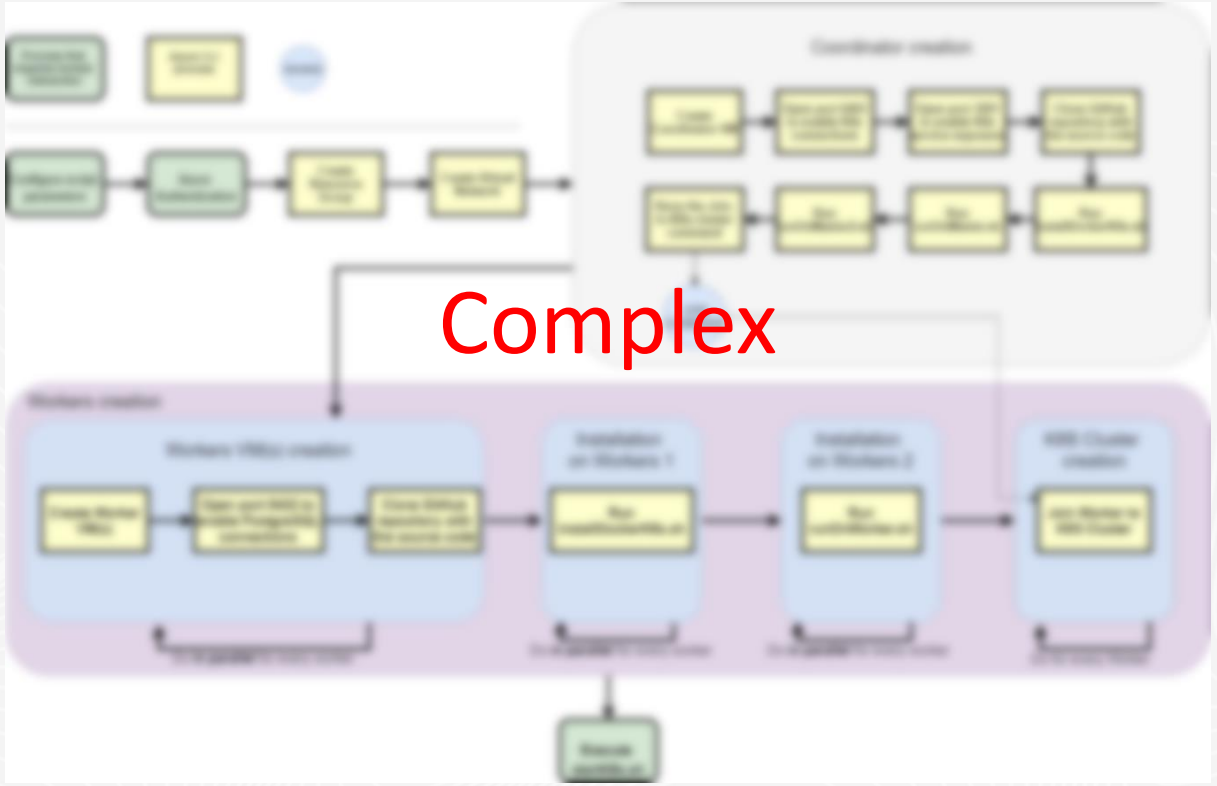
citusdata

Azure

Components

Deploying

Cluster initialization process diagram



Automating the cluster initialization

We provide

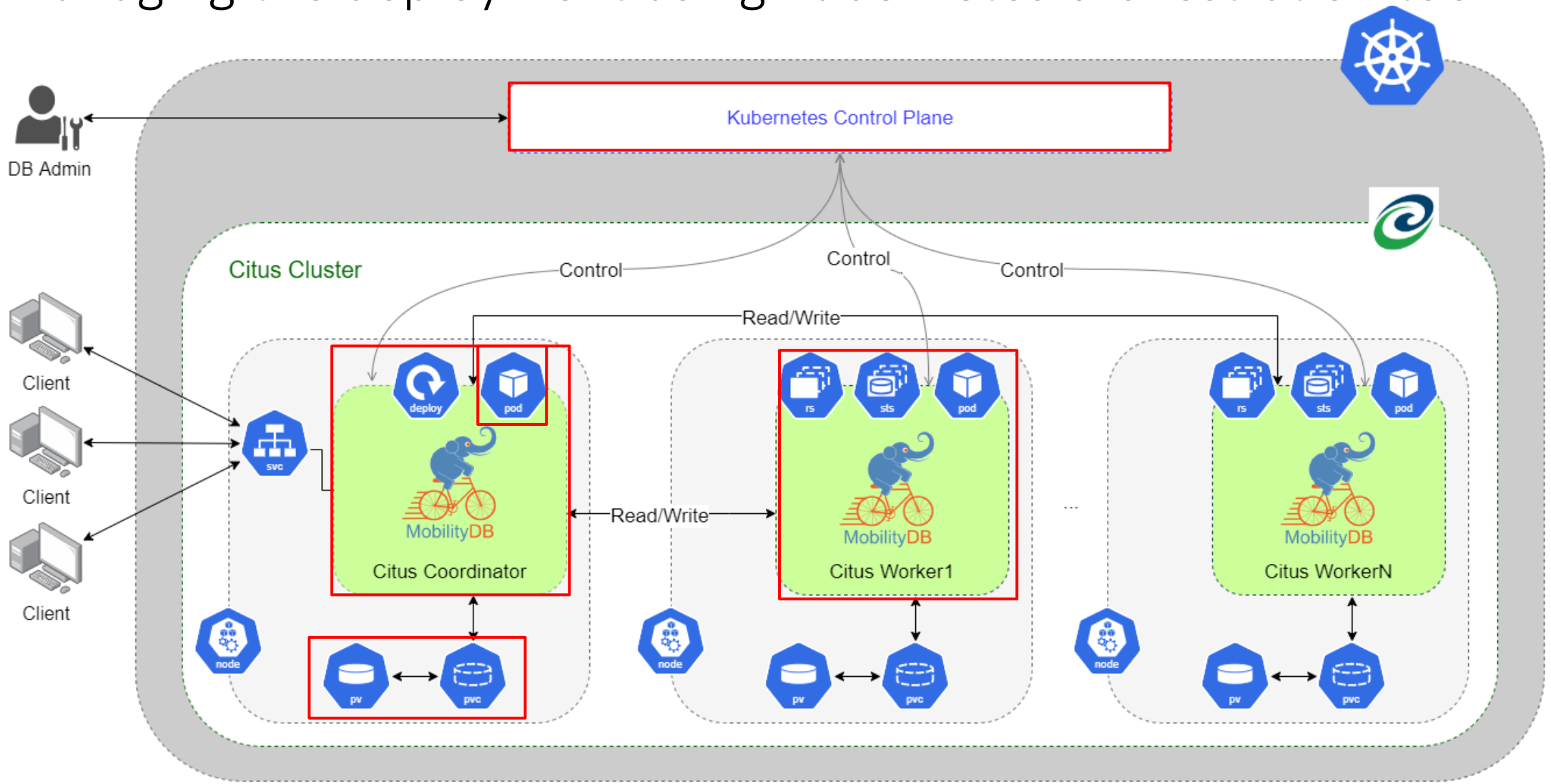
- **An easy to use**
- **Configurable**
- **Scalable process**

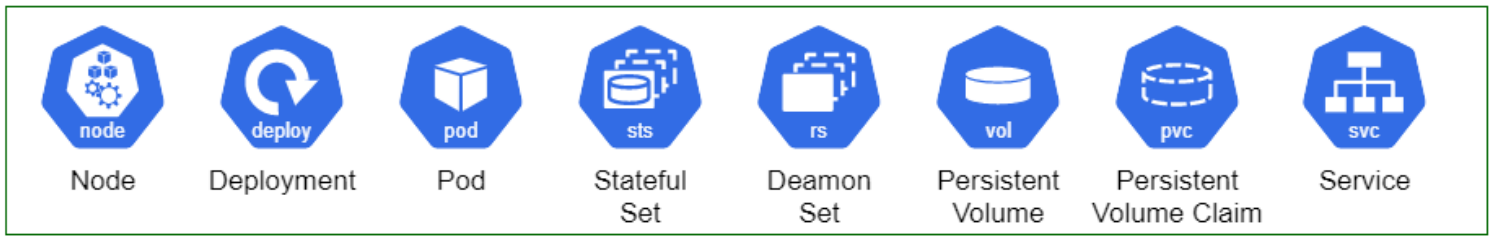
to effectively perform the cluster initialization process

Process key steps

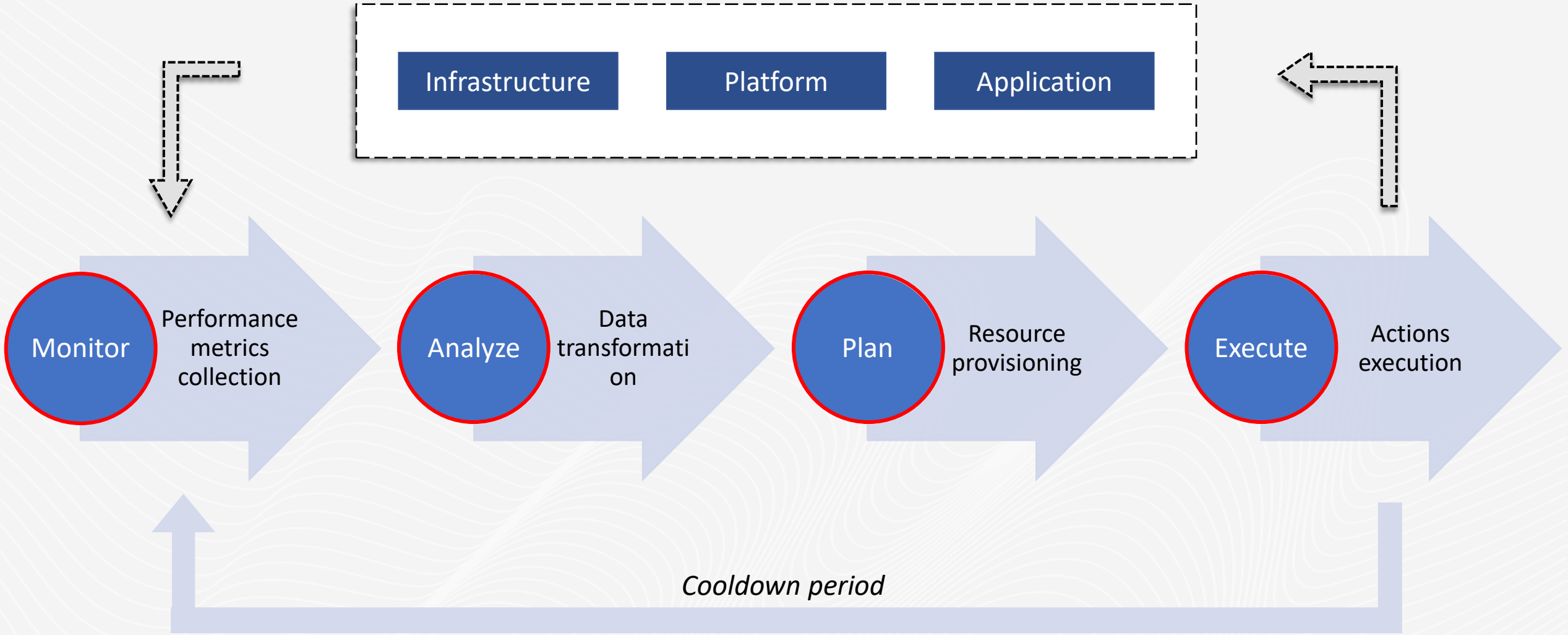
- **Infrastructure deployment and software installation**
- **Kubernetes cluster creation**
- **Citus cluster configuration**

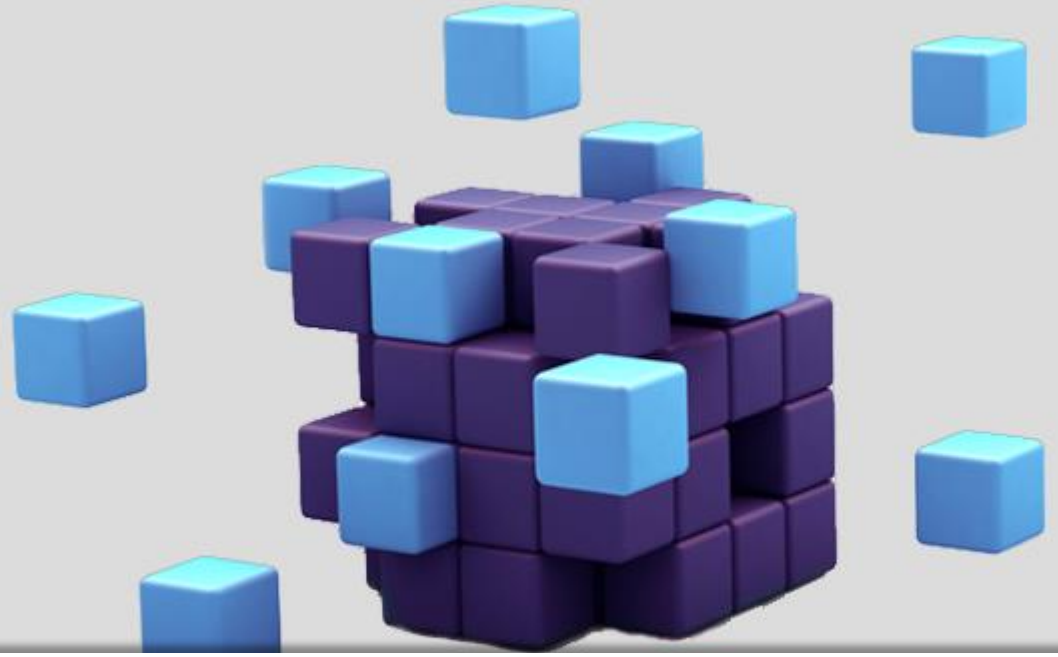
Managing the deployment using Kubernetes orchestration tool





Implementing an autoscaler





Experiments and Results

Dataset and benchmarking

Scalar benchmark [6]

- A benchmark to compare different autoscaling mechanisms
- 2 hours of user requests to the db

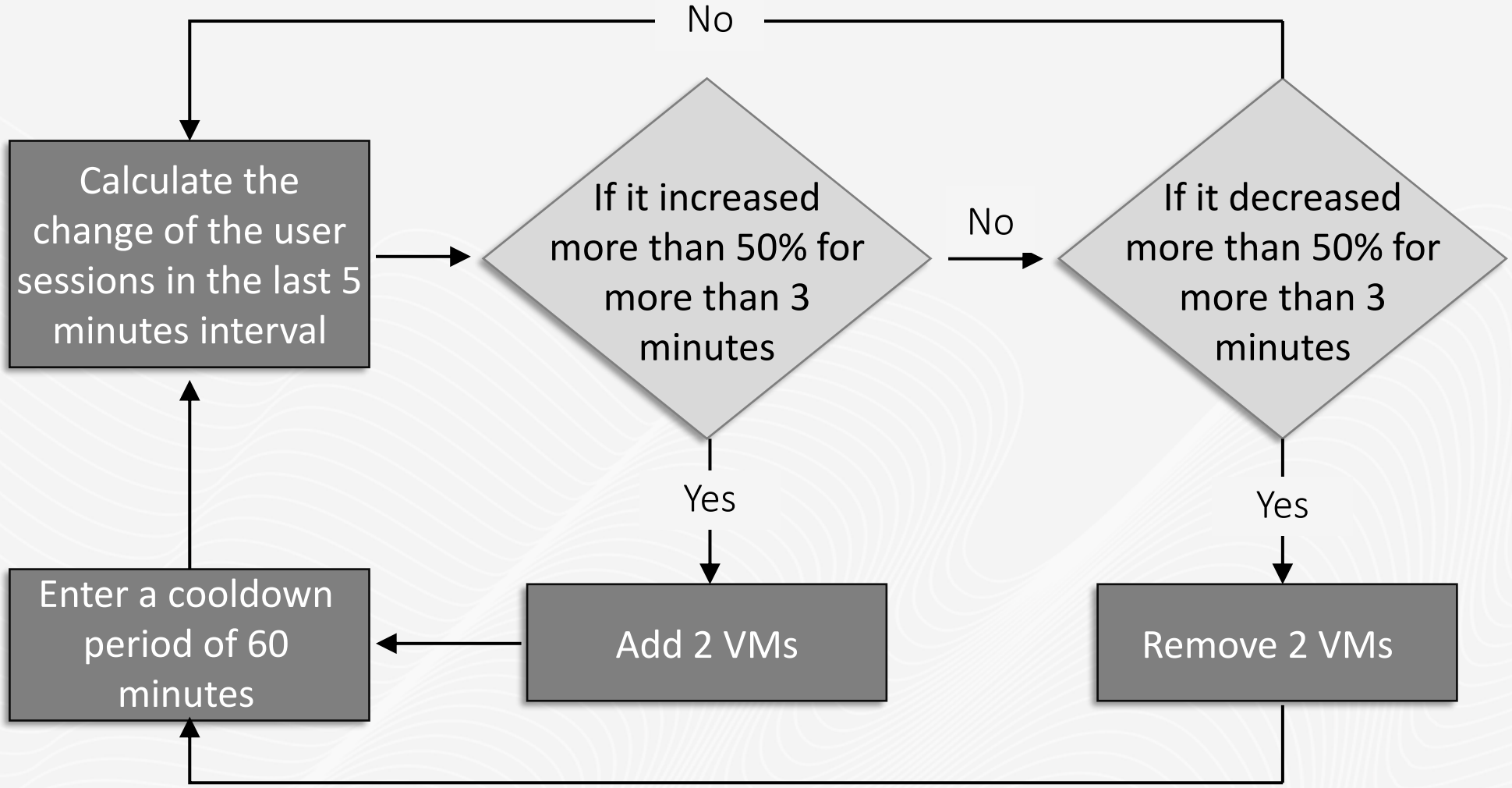
Used to test the **scalable** and **autoscaling** features

BerlinMOD benchmark [7]

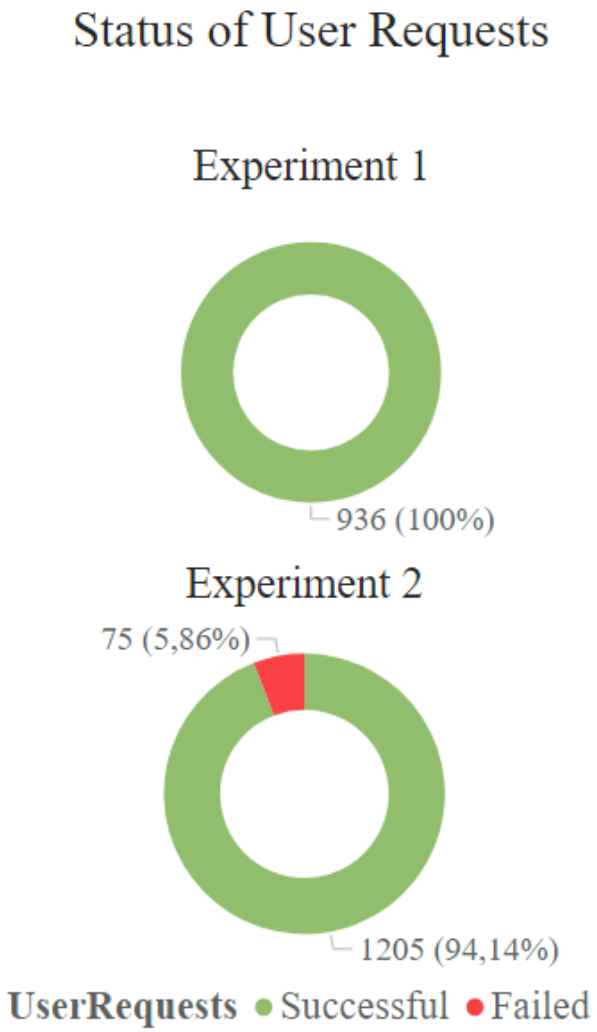
- A benchmark for moving object data, simulating vehicle trajectories
- Up to 30GB in our experiments

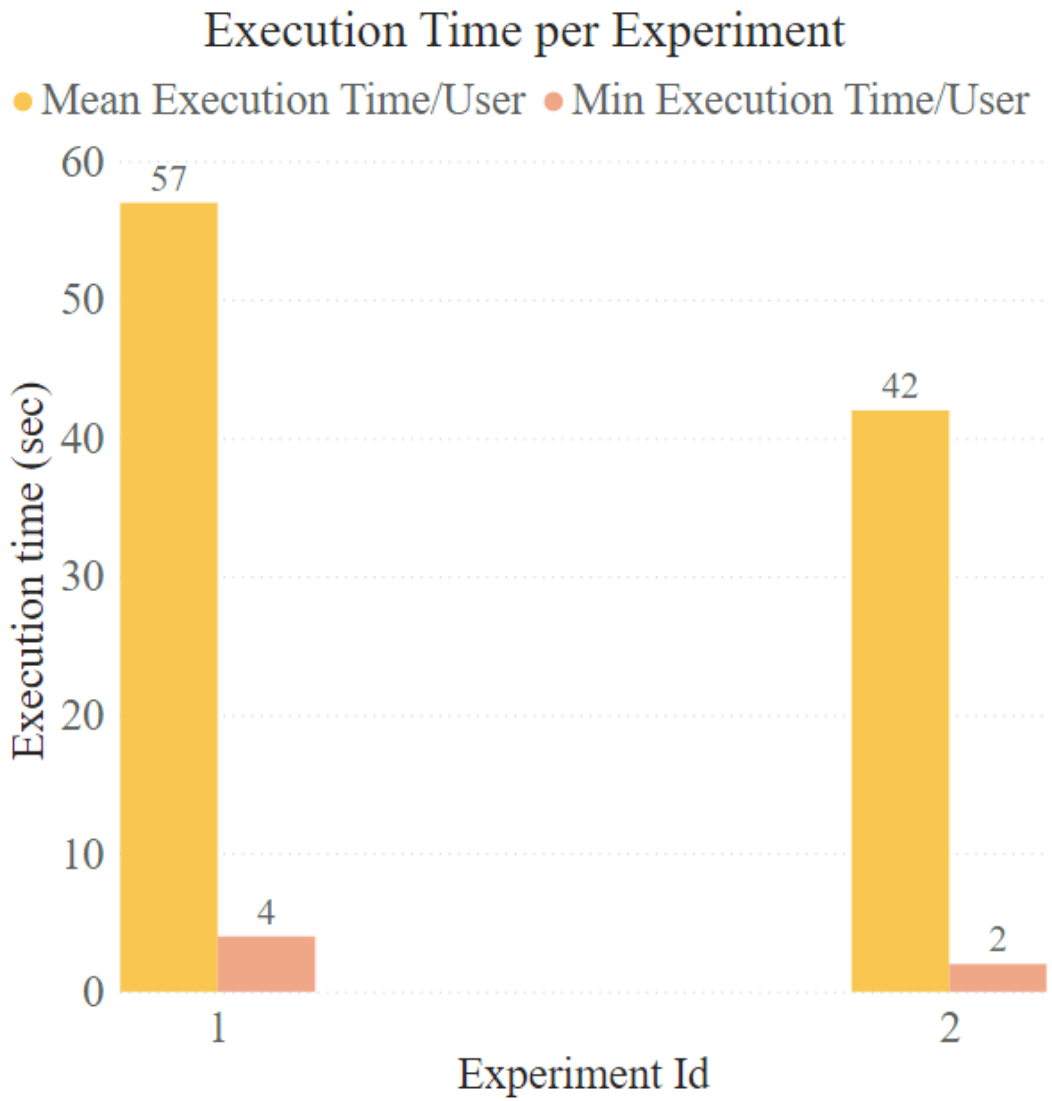
Used as **query workload** for the experiments

Rule-based, database activity autoscaler



Database activity autoscaler results





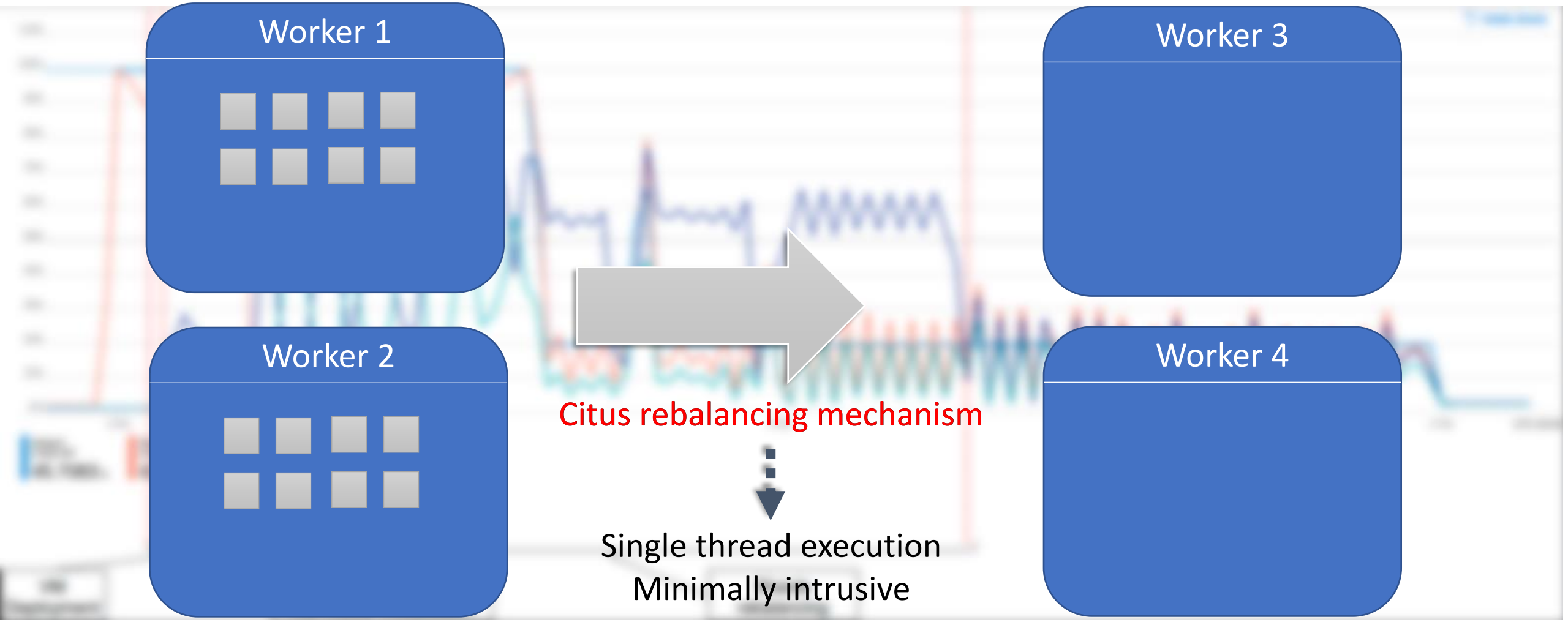
Conclusions

- 28% more users
- 1.3x average speedup
- 2x cost

Experiment 1: Scalar + BerlinMOD

Experiment 2: Scalar + BerlinMOD + Autoscaler

Analyzing the CPU utilization of the VMs



In conclusion, we achieved

Bash scripting

and **Azure CLI**

√ Automation of the deployment and maintenance of MobilityBD on MS Azure

Kubernetes

Implementation of autoscalers as **Python daemon processes**

√ Scalability by enabling users to easily use MobilityBD at scale, assisted by Citus extension

√ Elasticity by providing a tool that automatically adjust the size of the MobilityDB cluster

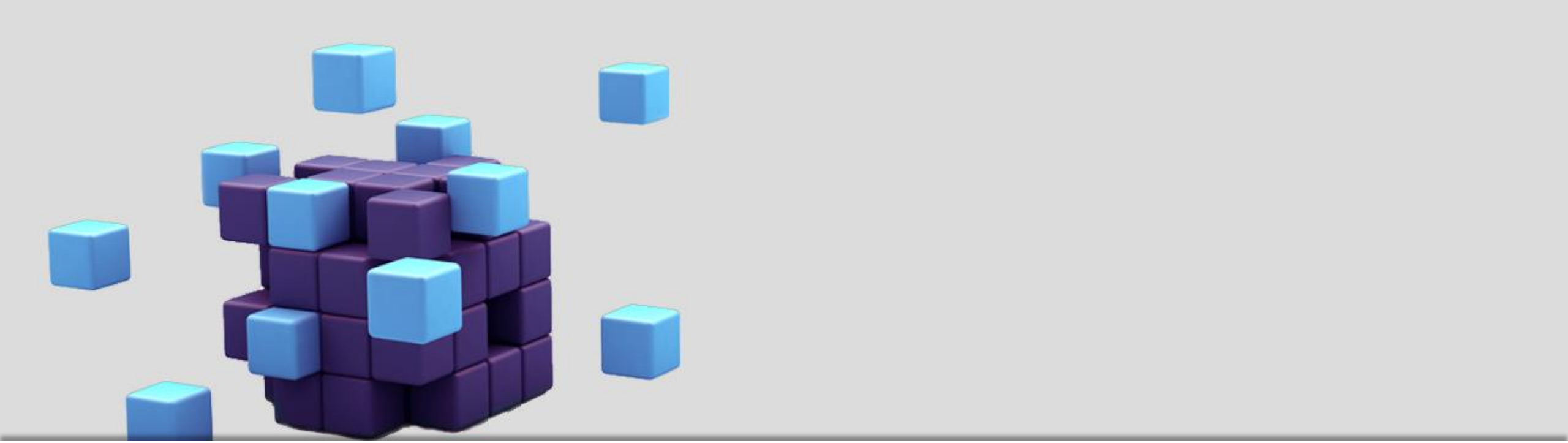
Future Work

Improve **the rebalancing mechanism**

Leverage **multi-cloud architectures**

PostgreSQL **native distributing solutions**

Generalized **groups of autoscalers** (OLAP vs OLTP autoscalers)



Tutorial

References

[1] Esteban Zimányi, Mahmoud Sakr, Arthur Lesuisse, and Mohamed Bakli. Mobilitydb: A mainstream moving object database system. SSTD ’19, page 206–209, New York, NY, USA, aug 2019. Association for Computing Machinery.
[2] Mohamed Bakli, Mahmoud Sakr, and Esteban Zimanyi. Distributed moving object data management in mobilitydb. BigSpatial ’19, New York, NY, USA, 2019. Association for Computing Machinery.
[3] Mohamed Bakli, Mahmoud Sakr, and Esteban Zimányi. Distributed spatiotemporal trajectory query processing in sql. In Proceedings of the 28th International Conference on Advances in Geographic Information Systems, SIGSPATIAL ’20, page 87–98, New York, NY, USA, 2020. Association for Computing Machinery.
[4] Mohamed Bakli, Mahmoud Sakr, and Esteban Zimányi. Distributed mobility data management in mobilitydb. In 2020 21st IEEE International Conference on Mobile Data Management (MDM), pages 238–239, Los Alamitos, CA, USA, jul 2020. IEEE Computer Society.
[5] Citus: https://www.citusdata.com/

References

[6] Wito Delnat, Eddy Truyen, Ansar Rafique, Dimitri Van Landuyt, and Wouter Joosen. K8-scalar: A workbench to compare autoscalers for container-orchestrated database clusters. In 2018 13th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), SEAMS ’18, page 33–39, New York, NY, USA, may 2018. Association for Computing Machinery.
[7] Christian Düntgen, Thomas Behr, and Ralf Hartmut Güting. Berlinmod: A benchmark for moving object databases. The VLDB Journal, 18(6):1335–1368, dec 2009.
[8] GitHub repository: https://github.com/JimTsesm/MobilityDB-Azure