

# Introduction to MongoDB

E-PORTFOLIO VON ROBIN WEISENBURGER



# Gliederung

2

1. Allgemeines zu MongoDB
2. Aufbau von MongoDB
3. MySQL vs. MongoDB
4. Anwendungszwecke
5. Installation MongoDB Server (DEMO)
6. CRUD Befehle (DEMO)
7. Weitere Befehle (DEMO)



# Was ist MongoDB?

3

- ▶ Basiert auf einer Sammlung von Dokumenten
- ▶ Kein festes Datenbankschema
- ▶ NoSQL
- ▶ Programmiert in C++
- ▶ Nutzung von ReplicaSets → hohe Skalierbarkeit
- ▶ Als SaaS oder OnPrem verfügbar



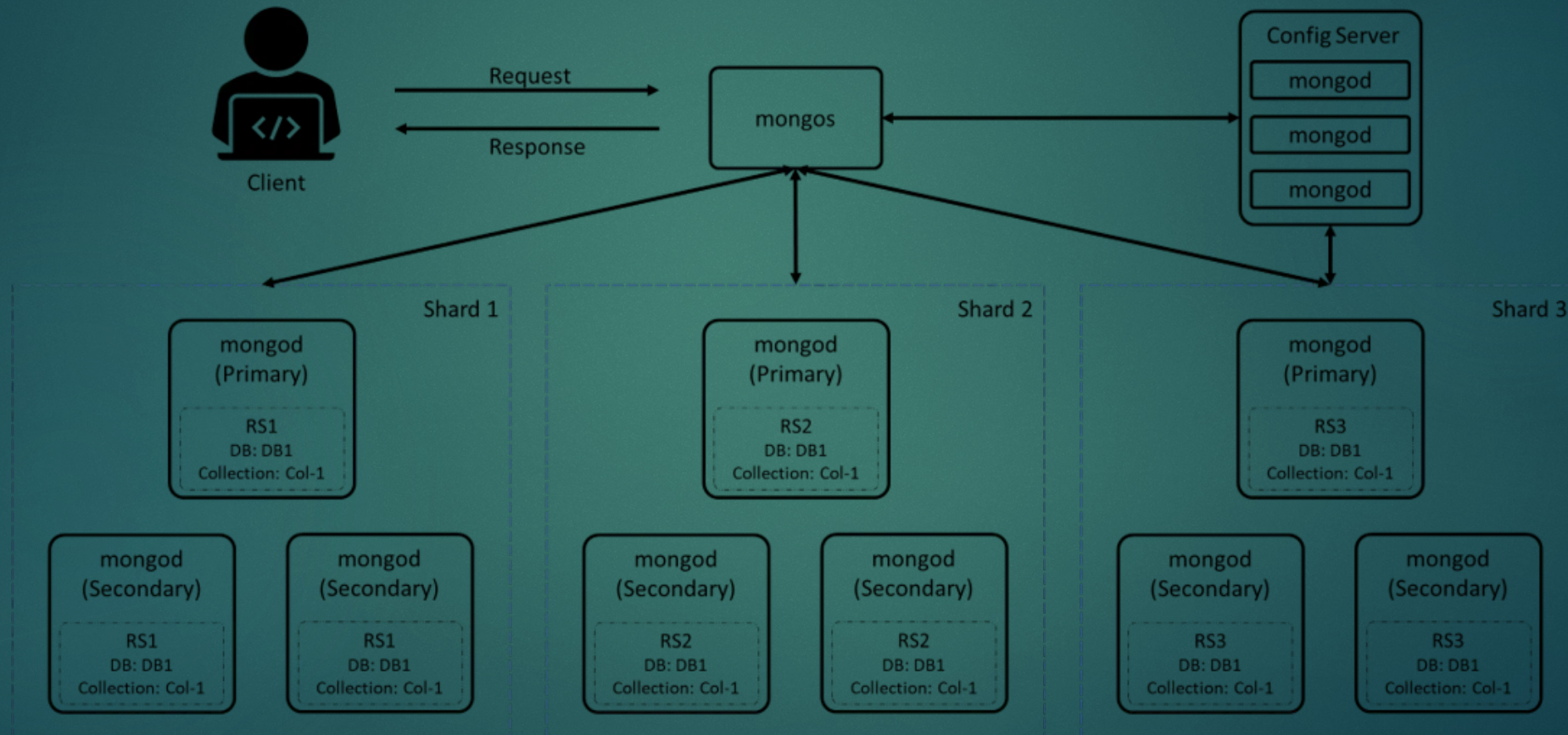
# Aufbau der Dokumente (JSON)

```
[  
  {  
    "departureAirport": "MUC",  
    "arrivalAirport": "SFO",  
    "aircraft": "Airbus A380",  
    "distance": 12000,  
    "intercontinental": true  
  },  
  {  
    "departureAirport": "LHR",  
    "arrivalAirport": "TXL",  
    "aircraft": "Airbus A320",  
    "distance": 950,  
    "intercontinental": false  
  }  
]
```



# Aufbau von MongoDB

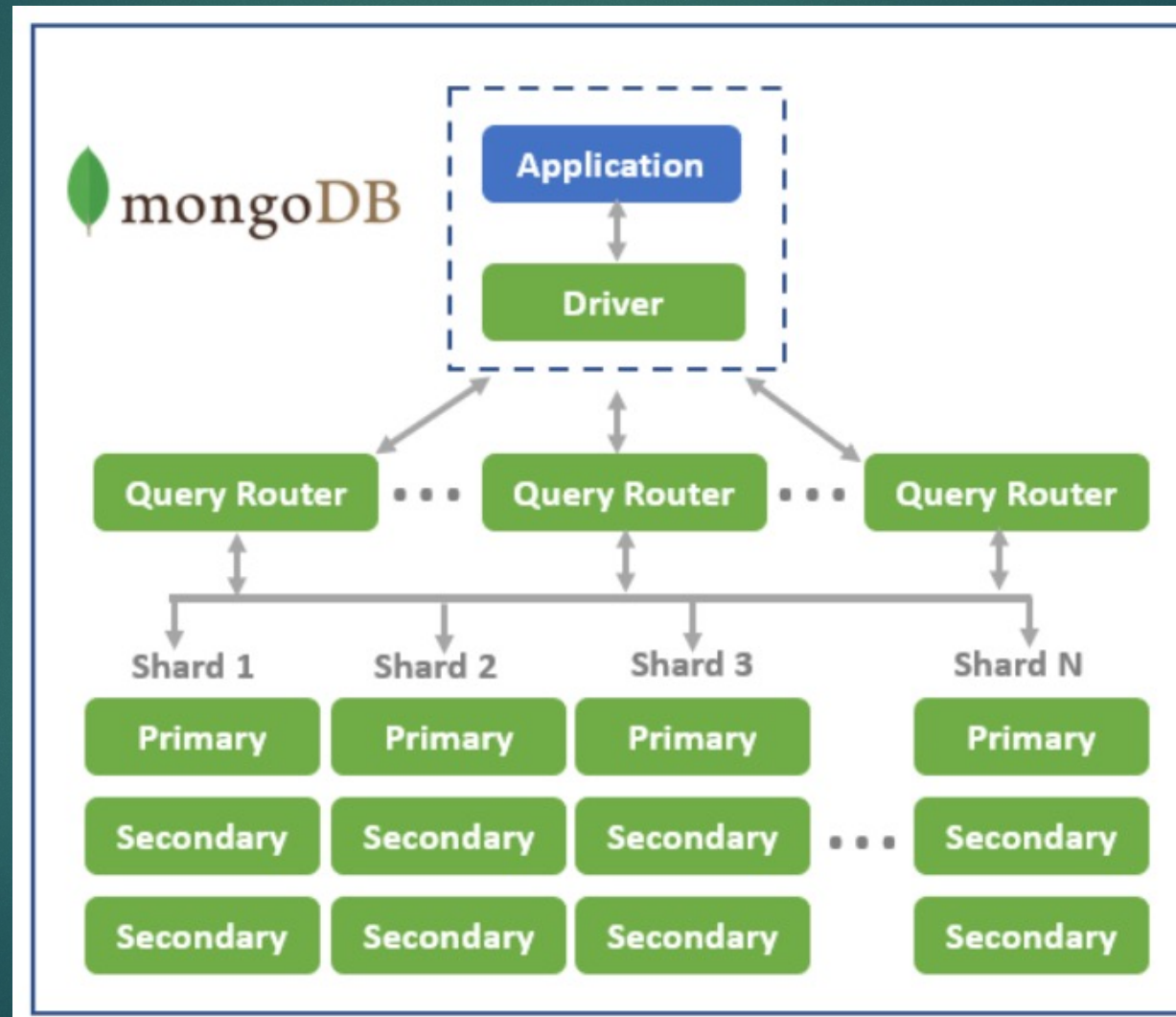
5





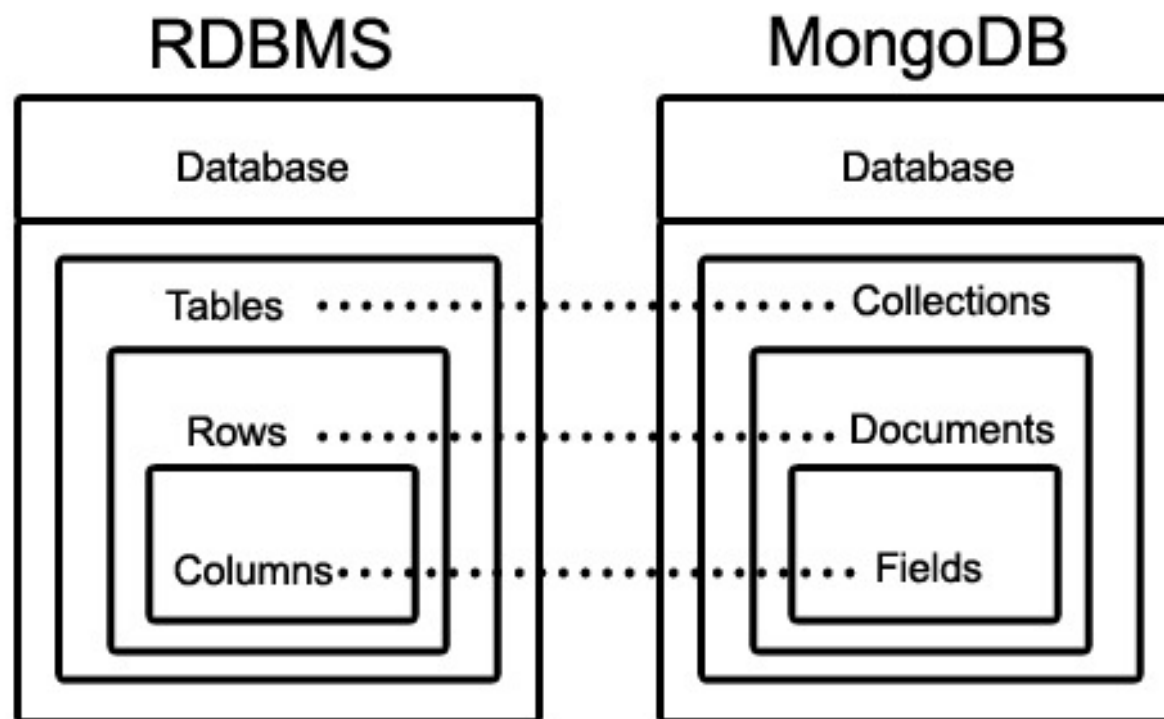
# Aufbau von MongoDB

6



# MySQL(RDBMS) vs. MongoDB

7





# MySQL vs. MongoDB

8

MySQL	MongoDB
<code>SELECT * FROM customer</code>	<code>db.customer.find()</code>
<code>INSERT INTO customer (cust_id, branch, status) VALUES ('appl01', 'main', 'A')</code>	<code>db.customer.insert({ cust_id: 'appl01', branch: 'main', status: 'A' })</code>



# Andwendungszwecke

9

- ▶ App Entwicklung → wächst und verändert sich mit der App
- ▶ Echtzeitauswertung und Analyse
- ▶ Big Data
- ▶ IoT Bereich
- ▶ Tracking Systeme
- ▶ Hochverfügbare Anwendungen
- ▶ Hohe Schreib/Lese Geschwindigkeiten benötigt werden



# MongoDB Server

10

- ▶ Über MongoDB Atlas → SaaS Service oder eigene Installation

Zugang zur Cloud DB über MongoDB Compass:

```
mongodb+srv://<username>:<password>@cluster0.5dj9j.mongodb.net/<
databasename>
```

Zugang zur Cloud DB über MongoDB Shell:

```
mongo "mongodb+srv://cluster0.5dj9j.mongodb.net/<databasename>"
--username <username>
```



# Server Demo



# CRUD Befehle

12

- ▶ Create:

`insertOne(data, options)`

`insertMany(data, options)`

`insert(data, options)`

- ▶ Read:

`find(filter, options)`

`findOne(filter, options)`

- ▶ Update:

`updateOne(filter, data, options)`

`updateMany(filter, data, options)`

`replaceOne(filter, data, options)`

- ▶ Delete:

`deleteOne(filter, options)`

`deleteMany(filter, options)`



# DEMO