

**Prof. DI Dr. Erich Gams**

Einführung und Anwendung

# MongoDB

informationssysteme htl-wels

# Übersicht Was lernen wir?



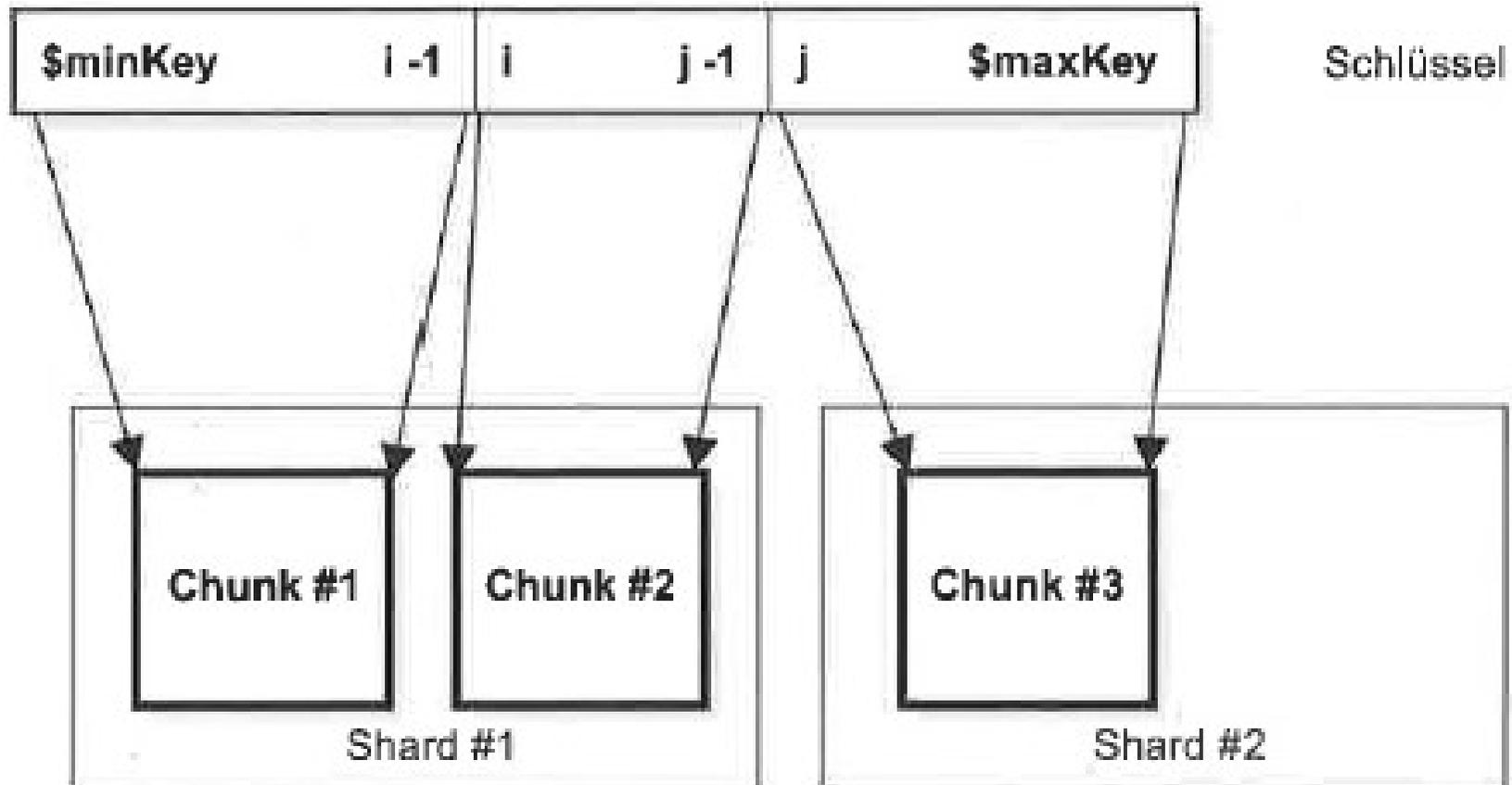
- › Partitionierung oder Sharding
- › Hands-on -> Tutorial



# Partitionierung oder Sharding

- › Ist die Verteilung der Daten auf mehrere Netzwerkknoten.
- › Collections werden in kleinere Einheiten, sogenannte Chunks aufgeteilt, von denen jeder auf genau einem Sharding-Knoten (sogenannte shards) liegt.
- › Sie werden redundanzfrei persistiert!

## Verteilung der Daten anhand des ShardKeys



## Brauche ich Sharding?

- + Plattenplatz eines Knotens reicht nicht aus.
- + Durchsatz an Schreiboperationen kann von einer Server-Instanz nicht mehr bewältigt werden.
- Keine besonderen Anforderungen an Schreib- und Lesezugriffe.
- Nicht mehr als ein paar Terrabyte an Daten.

# Bereichspartitionierung (range partitioning) - Beispiel

- › Shard 1: Personen [A-F]
- › Shard 2: Personen [G-N]
- › Shard 3: Personen [O-Z]
- › Abfrage der Person Franka -> Shard 1
- › Einfügen der Person Paul -> Shard 3
- › Verteilung der Last!
- › Wird ein Chunk zu groß, wird er auf zwei aufgesplittet und einer auf einen anderen Knoten verlegt.

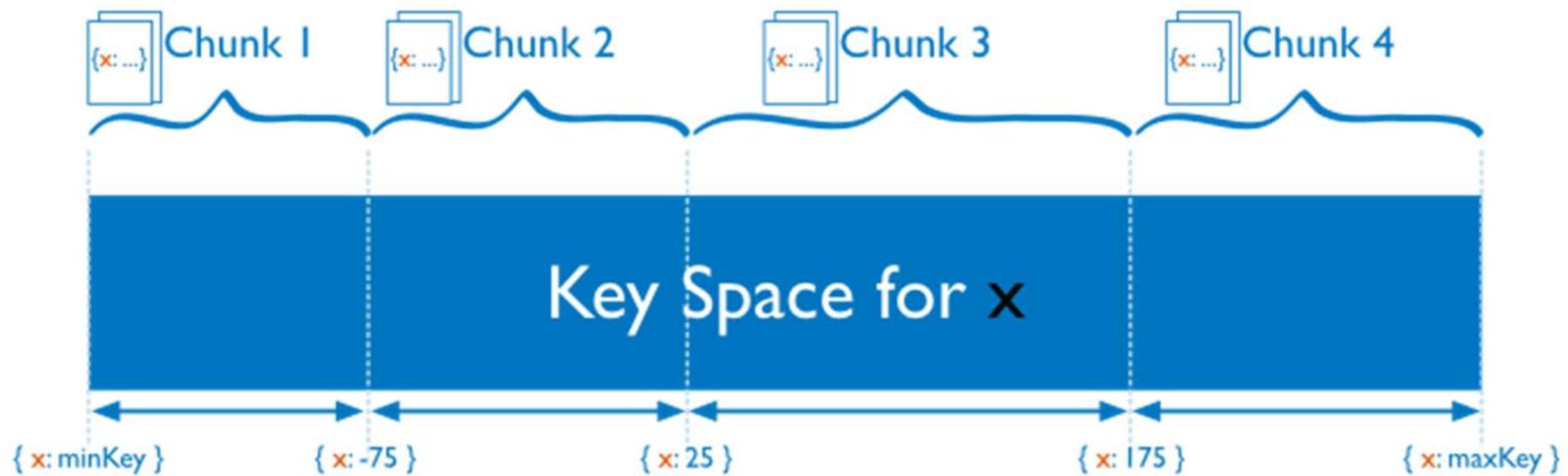
# Schlüssel zur Verteilung der Daten

- › Strings oder Ganzzahlwerte
  - sofern große Teile des Wertebereichs ausgeschöpft werden
  - Bei großer Variabilität der Werte (Zuordnung zu vielen verschiedenen Chunks)
  - Boolesche Werte?
  
- › Verteilung der Schlüsselwerte
  - Monoton steigend bzw. fallende Werte
  - Hashwerte
  - Mischformen
  
- › Abhängig von der Relation von Schreib- und Lesewerten

# Monoton steigende Werte

- › z.B.:Felder mit Datum
- › Es wird immer nur auf den aktuellen letzten Chunk geschrieben.
- › Nachteil beim Einfügen -> Bottleneck
- › Bereichsabfragen performant

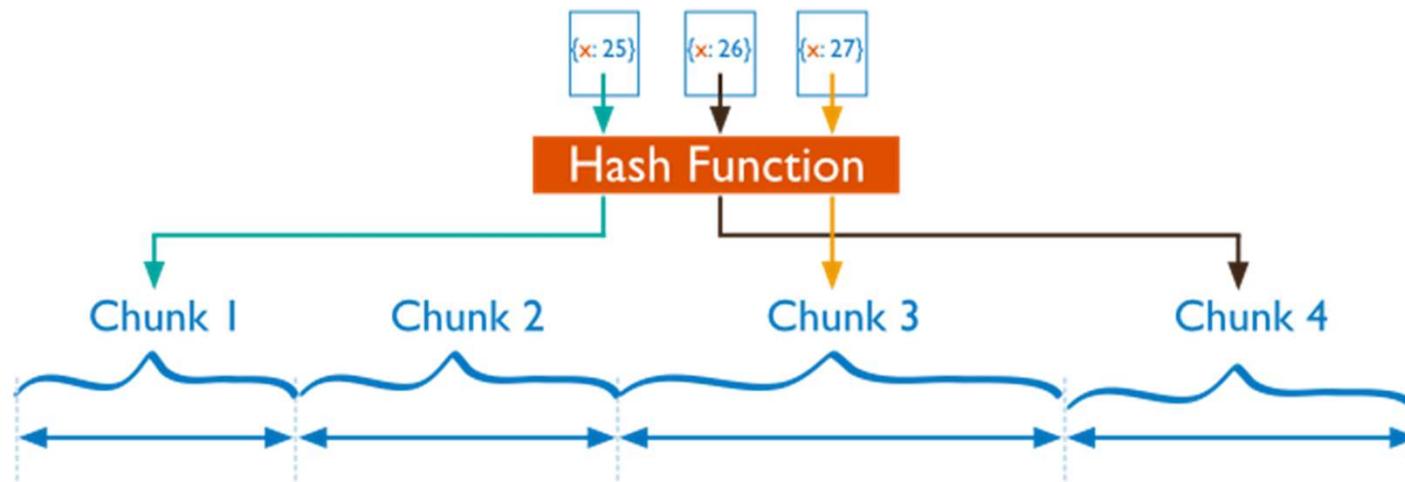
# Monoton steigende Werte



## Hash partitioning - Hashwerte

- › Können von der Anwendung oder automatisch vergeben werden. (Indextyp: {vnr: "hashed"})
- › Einfügen performant (Verteilung)
- › Bereichsabfragen gehen (meist)immer über alle Shards

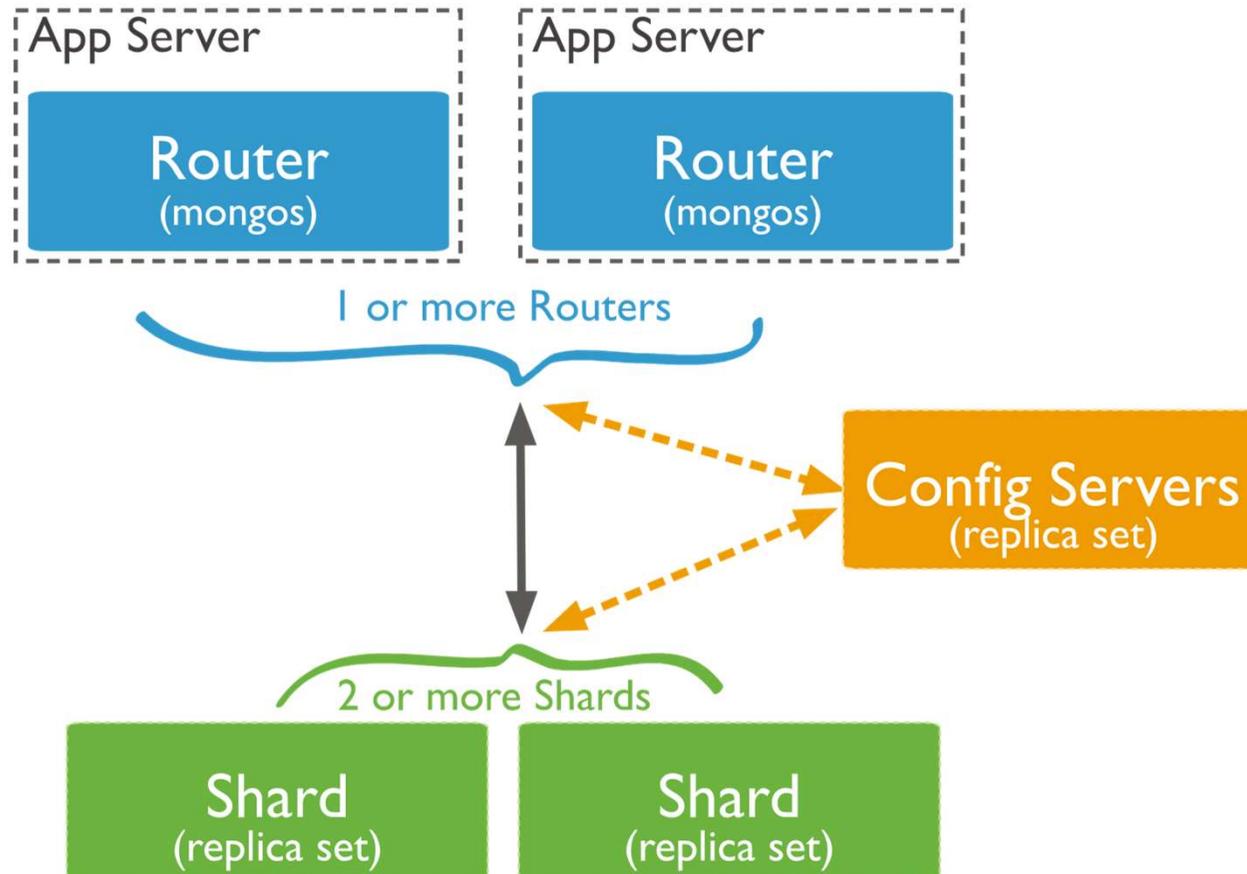
# Hashwerte



# Mischform

- › Zusammengesetzter Shard Key
- › Beispiel: E-Mail System
- › Erster Teil
  - sorgt für grobgranulare Einteilung und gleichmäßige Verteilung auf die Shards
  - z.B.: Username
- › Zweiter Teil
  - Zeitstempel: Erzeugungszeit
  - Variabilität auf Ebene des Dokuments

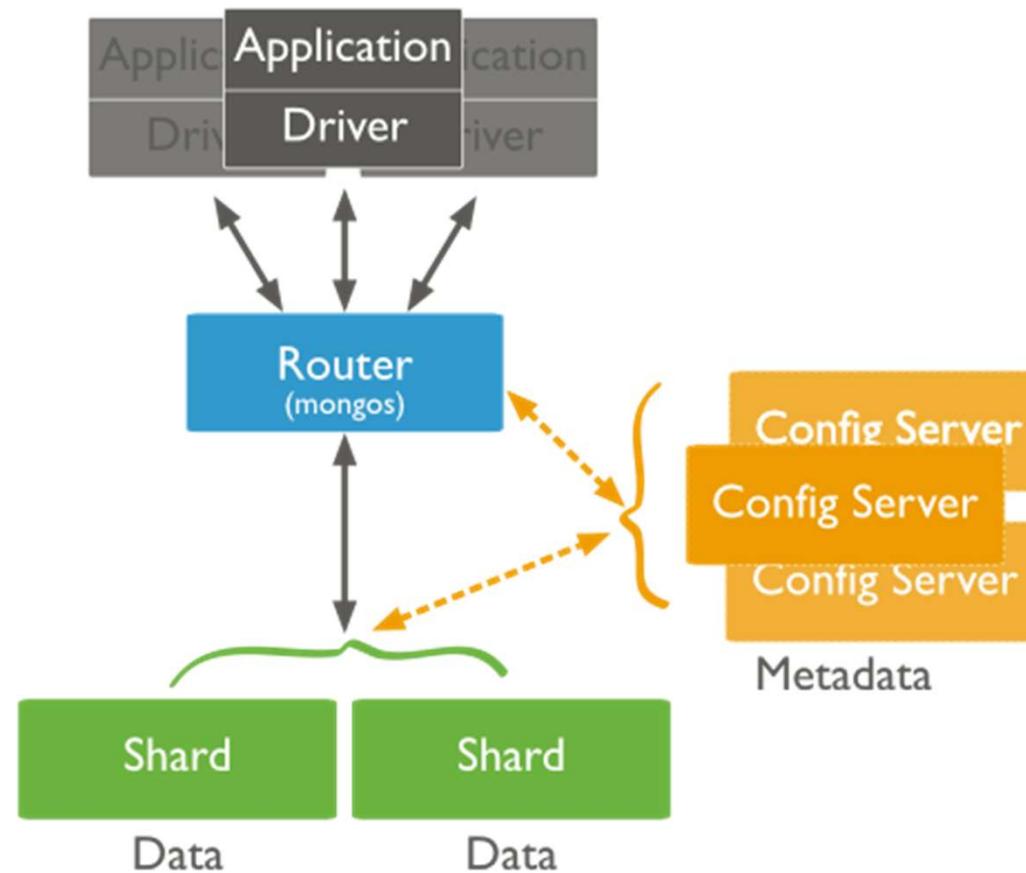
# Ablauf und Rollen beim Sharding



# Ablauf und Rollen beim Sharding

- › Die Anwendung wendet sich mit einer Anfrage an einen der Router.
- › Der Router befragt einen Konfig-Server welchen Shard er auswählen soll. Konfig Server enthalten sämtliche Metadaten (Welche Rechner nehmen am Sharding teil, wo findet man welche Dokumente usw.).
- › Die Anfrage wird an den Knoten gereicht und der Router fasst die Antwort zusammen und verweist an die Anwendung.

# Ablauf und Rollen beim Sharding



# Ablauf und Rollen beim Sharding

## > Shards

- **mongod** Instanz oder **replica set**

## > Config servers

- **mongod** Instanz (In produktiven System werden laut Doku 3 empfohlen). Kann am Shards Server laufen.

## > Router

- Jeder Router ist eine **mongos** Instanz. Können am Applikationsserver laufen.



# Sharding

› Übung!

Auf los geht's los ;-)

