**Klassifikation von Key-Value Datenbanken**

Unter Skalierbarkeit von Datenbanken versteht man, die Fähigkeit der Datenbank mit hoher Anzahl von Zugriffen und großen Datenmengen umgehen zu können. Dabei unterscheidet man grundsätzlich die Skalierbarkeit in zwei Typen:

- *Horizontale Skalierung*

Horizontale Skalierung von Datenbanken ist sinnvoll, wenn viele Zugriffe auf die Datenbank erfolgen, diese jedoch beschäftigt ist und daher sich die Performance reduziert. Aus diesem Grund können Cluster von identischen Datenbank Instanzen verwendet werden, um die Last zu balancieren, wodurch sich der Durchsatz erhöht.

*- Vertikale Skalierung*

Vertikale Skalierung kann in Betracht gezogen werden, wenn die Datenbank die große Menge an Daten nicht mehr verwalten kann. Zum Beispiel wir verwalten einen Datenbankserver mit 2GB Speicher, welcher nun durch die große Datenmenge an Performanz verliert. Um die Performanz zu erhöhen, kaufen wir einen neuen Datenbank Server mit 2TB Speicher.

Dabei klassifizieren wir die Key-Value Datenspeicher in drei Kategorien:

* **Bibliotheken:**

In dieser Kategorie sind die Key-Value Datenspeicher enthalten, die sich einbinden lassen und danach genutzt werden können. Als Beispiel kann hier MapDB herangezogen werden, da durch Ergänzen der dependency im Maven Projekt diese direkt genutzt werden kann.

* **Prozesse, die skalierbar sind:**

In diese Kategorie fallen die Key-Value Datenspeicher, die als ein eigener Prozess ausgeführt werden und sich dynamisch bei steigender oder sinkender Last an den Workload adaptieren. Ein Beispiel wäre hierfür Riak bei der man einen eigenen Prozess starten kann und die Daten auf mehrere Cluster („Buckets“) verteilen kann.

* **Prozesse, die nicht skalierbar sind:**

In diese Kategorie fallen die Key-Value Datenspeicher, die zwar als eigener Prozess ausgeführt werden, jedoch sich nicht dynamisch anpassen können.