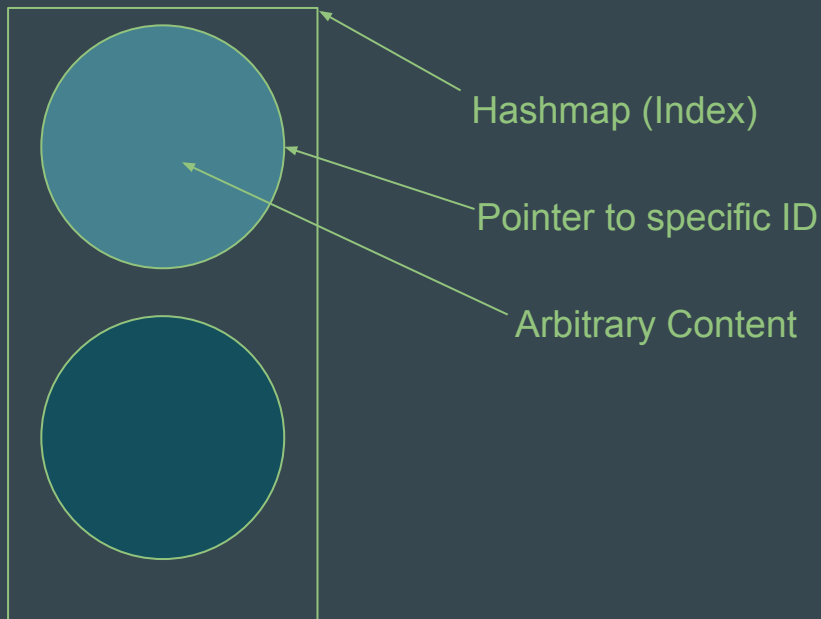# About Me
Jakob Oberhummer

- Quit school with 17 (HTL spengergasse)
- Started an apprenticeship as a Bookbinder .. realized getting up at 7 is not my thing as well as 8 hours in front of a glueing machine kills the romantic idea ….
- Did a lot of shit jobs (handing out flyers, call-center agent, ….)
- Spent 1,5 Years in a monastery doing construction work
- Started an IT education (Telekommunikationstechnik) got bored and quit
- Started in an Agency as a Volunteer, exhaustion hurray …. HTML4, CSS2, Templates..
- Got a job in parallel as a web-developer …. PHP, HTML, CSS2, Product photography
- Quit the job started as Support and Template engineer at emarsys
- Got 2nd level support, automated testing, started programming some tools for the support
- Quit the job …. CEO didn't wanna let me go … got a job as Dev
- Started learning large-scale applications, virtualizations
- Got Team-lead
- Gave up Team-lead (to much planning too little programming)
- Quit Job
- Started as DevOps @ impossible project
- … started my own "company" (EPU)
- Got a job in parallel ….. Started a Startup ….. Left the startup … started giving talks
- …… blabla bla "I can haz code"

NoSQL-Document based

•••

Because Semi is still better than no structure
Key-Value example …. Because i can ;) …

Key Value storages

Hashmap (Index)

Pointer to specific ID

Arbitrary Content

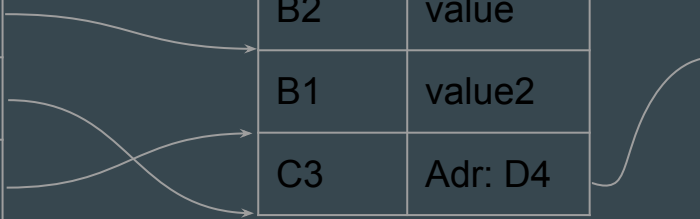| Attribute	Value
Access time	O(1) in general this is excluding the overhead Like: ● socket connection / access time ● Buffer write / initialization time ● IO-reads ● Page cache handling [L1, L2, L3] ● The specific implementation
Use Cases	Caching, Sessions, basically anything where the state and the content can be arbitrary
Types	Eventual Consistent, Ordered, Ram, ...
Candidates	Redis, Cassandra, Dynamo, Riak, Berkley DB

Hashmap Pseudo in Memory Representation

Key Store (typed String)

Buckets

Key	Ptr
Key1	B2
Key2	C3
Key3	B1

Adr	Val
B2	value
B1	value2
C3	Adr: D4

Adr	Ref
D4	ptr1

Documents

Documents Container

A Document is semi-structured data-set with an unique Identifier

Id   Property A   Property B

Id   Property B

Attribute	Value
Access time	Depends obviously on Query ● index O(1)
Use Cases	Logs, Customer-data, Session data, formdata. Where the structure of the sets can be changing.
Candidates	MongoDB, CouchDB, Elasticsearch,....

Collections

Collection
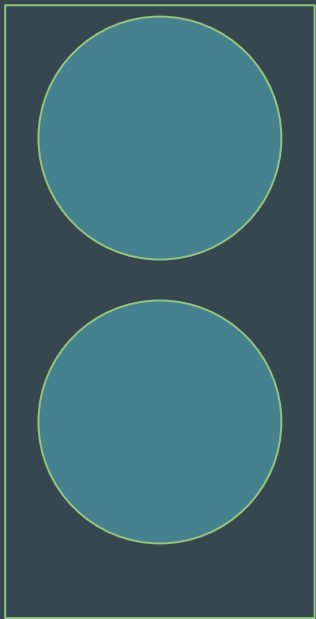
*"In <u>computer science</u>, a **collection** or **container** is a grouping of some variable number of data items (possibly zero) that have some shared significance to the problem being solved and need to be operated upon together in some controlled fashion."*

(https://en.wikipedia.org/wiki/Collection_%28abstract_data_type%29)

Lets see some Code

Thank you