Databases a Brief-overview

Part 1 Comparison

Omg I can has tableZ

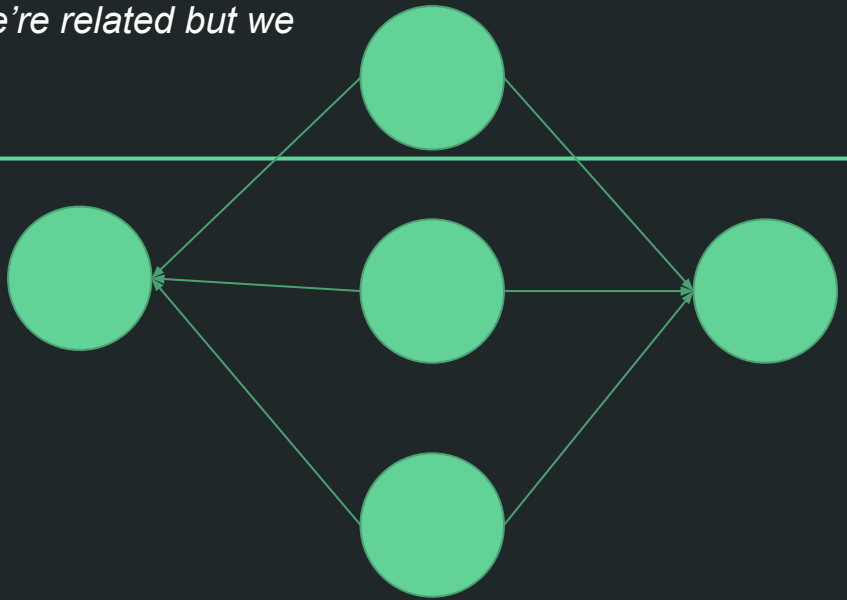
The SQL Table-Based

The usual suspects!

- MySQL
- Postgres
- Oracle
- MSSQL

Age	: 40+
Data-Structure	: Tables
Meaning	: Structured Query Language
Main Focus	: Consistency and Integrity

Well we're like cousins …. we're related but we don't really meet that often



Differences of the Engines

Postgres:

- pgSQL only has 1 engine
- since 9.3 you can call they support “NoSQL” via JSON
- Since 9.4 there are JSON aggregation methods
- …..
- Best open source SQL database to my knowledge
- A huge amount of data types
- DTL is transactional

MySQL

- MyISAM (No foreign keys, fast reads, fast writes, no transactions, fulltext indexes)
- InnoDB (FK, transactions [DDL])
- TokuDB (Fractal Indexes, for an high amount of inserts)
- Memory-Tables
- ……..

Oracle:

- All the same like MySQL and even more
- CSV tables
- Black Hole tables /dev/null
- Archive tables
- ……..

MSSQL

- Sucked in the past
- The new features sounded interesting
- In the end they are the click based solution which obviously can scale for enterprises
- JSON support

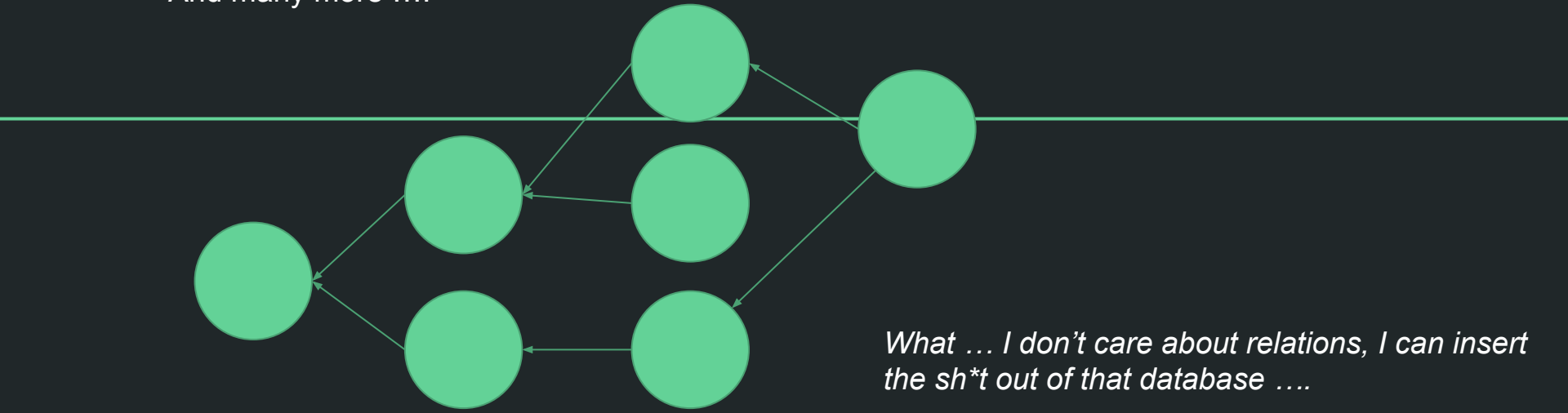
Omg I can has Documents

Document-based

A view examples

- CouchDB (HTTP API)
- RavenDB (.Net)
- MongoDB (the MySQL for Nodejs|MEAN)
- BaseX (XML based)
- Elasticsearch (Search-engine)
- And many more ….

Age	: 9+
Data-storage	: Document based
Main focus	: no schema, insert speed



*What … I don't care about relations, I can insert the sh*t out of that database ….*

CouchDB:

- HTTP API
- Erlang/C
- Database|Document|Views
- 1 Engine, a more functional than object oriented concept

MongoDB:

- Javascript/C/C++
- Database|Collections|Documents
- Classical Client API
- WiredTiger (Default since 3.2), MMAPv1 (Default Before), In-Memory-Storage (beta)

RavenDB

- .Net
- Windows compatible
- ESE, Voron
- Database|Collection|Documents

ElasticSearch:

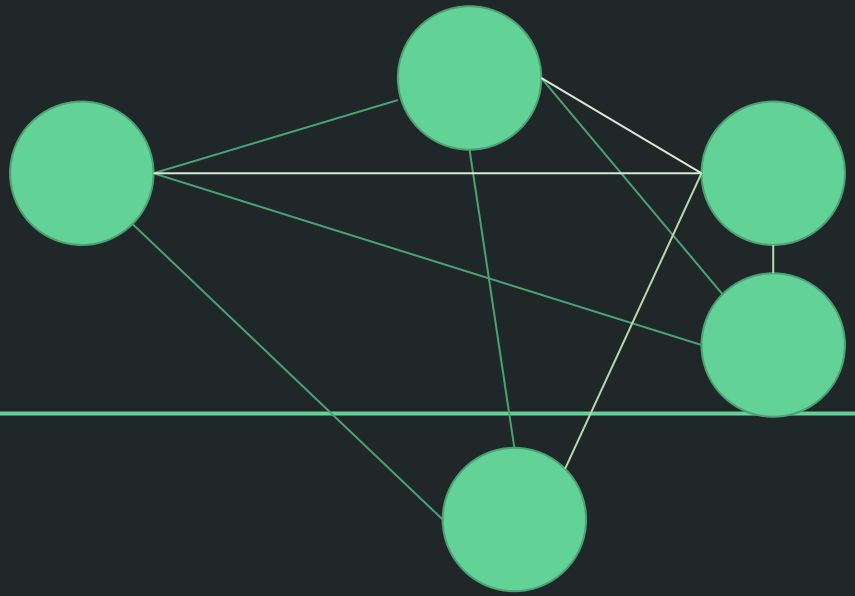
- Originates as a Search-Engine
- Now is listed as Database
- Java
- Database|Documents

Omg I can has GraphZ

Graph-Based

Wait a minute ….. I don’t need this ! I can solve this in SQL with only 50 tables !
I know my Joins ….

- Neo4j
- Allegrograph
- OrientDB



We are all related somehow!…. Don’t make this weird !

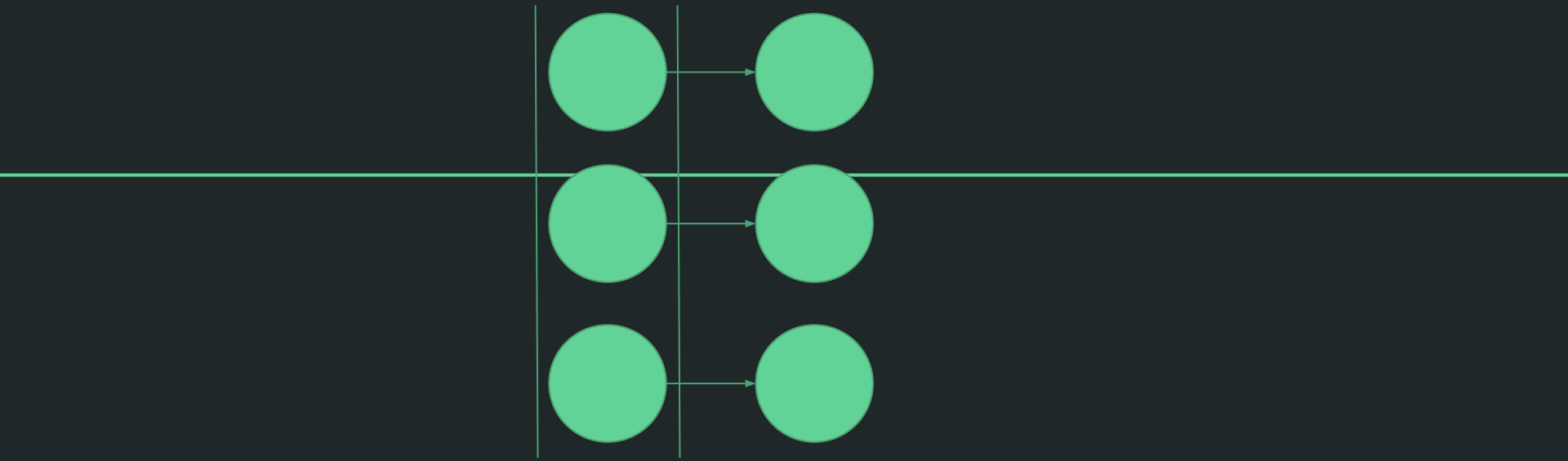
I can write hashmaps myself!

Key Value Based

Relations are for Suckers …..

- Redis
- BerkleyDB
- FoundationDB
- ……

I don’t care i have the same thing 1 million time in memory
I’ve got O(1) Average….

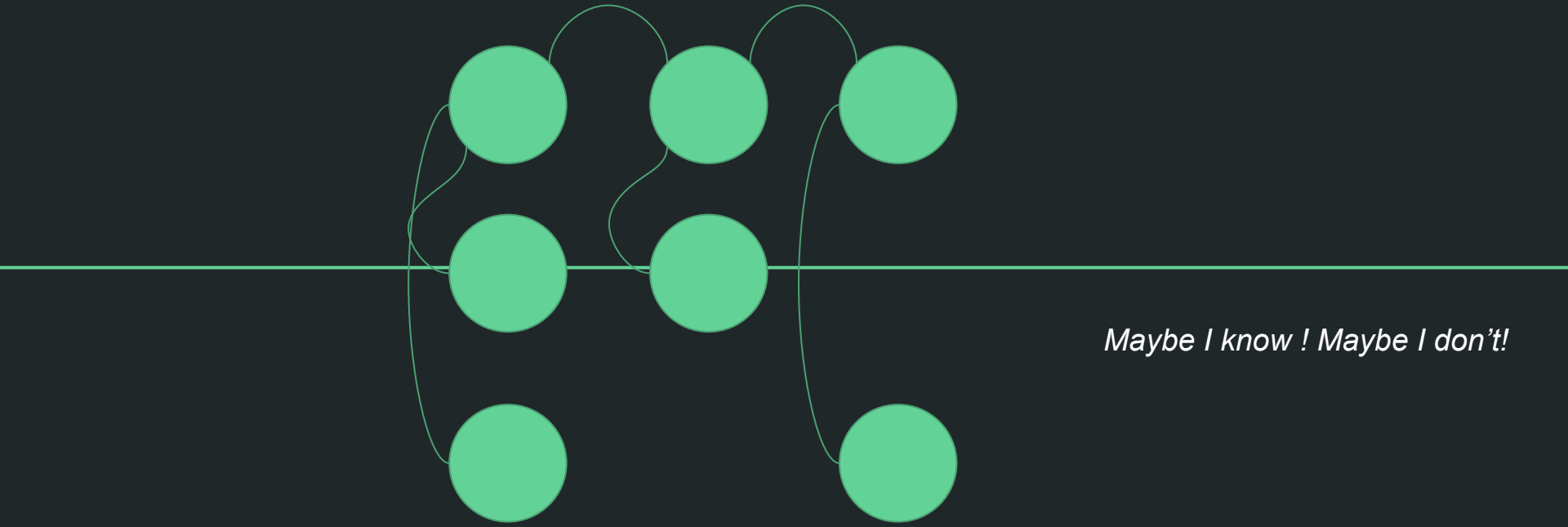


Omg I don’t know what that’s for!

Column-Based

Everything can be arbitrary …. And eventual consistent

- Cassandra
- Druid
- MonteDB



Maybe I know ! Maybe I don’t!