iDelta
advanced solutions

A lightning Introduction to noSQL

Iain MacLeod

iDelta Ltd

JBUG Scotland 30/01/14

iDelta
advanced solutions

Outline

1. What are noSQL Databases.
2. Types of noSQL Databases.
3. Use Cases.
4. Summary.

iDelta
advanced solutions

1. What are noSQL Databases?

iDelta
advanced solutions

What are noSQL Databases?

Trivia:

#noSQL was a hashtag used when organising a 2009 meetup to discuss non RDBMS /clustered databases in San Francisco.

iDelta
advanced solutions

What are noSQL Databases?

What's "wrong" with RDBMS /SQL ?

iDelta
advanced solutions

What are noSQL Databases?

What's "wrong" with RDBMS /SQL ?

- Big Data - horizontal scaling is an issue.

iDelta
advanced solutions

What are noSQL Databases?

What's "wrong" with RDBMS /SQL ?

- Big Data - horizontal scaling is an issue.
- Object-Relational Impedance mismatch - Developer productivity

iDelta
advanced solutions

What are noSQL Databases?

What's "wrong" with RDBMS /SQL ?

- Big Data - horizontal scaling is an issue.

Main driver behind the development of noSQL Databases:

Google - BigTable

Amazon - Dynamo

iDelta
advanced solutions

What are noSQL Databases?

Characteristics of noSQL DB's:

- Non relational
- No Schema
- Can scale horizontally
- Tend to be open source

iDelta
advanced solutions

2. Types of noSQL Database

iDelta
advanced solutions

Types of noSQL Database

mongoDB

Infinispan

riak

TERRACOTTA

Neo4j
the graph database

ORACLE
Coherence

apache CouchDB
relax

Red Hat JBoss Data Grid

elasticsearch.

Cassandra

Apache Jackrabbit™

APACHE HBASE

InfiniteGraph

redis

Amazon DynamoDB

MarkLogic

APACHE RIVER

MEMCACHED

HYPERTABLE INC

WebSphere eXtreme Scale

iDelta
advanced solutions

Types of noSQL DB

1. Key-Value Database
2. Document Database
3. Column Family Database
4. Graph Database

iDelta
advanced solutions

Types of noSQL DB

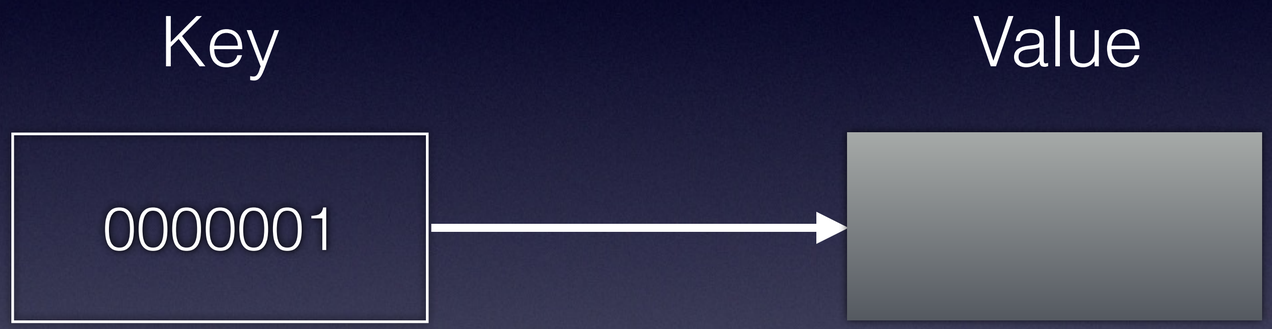
“Aggregate-Orientated”

1. Key-Value Database
2. Document Database
3. Column FamilyDatabase
4. Graph Database

iDelta
advanced solutions

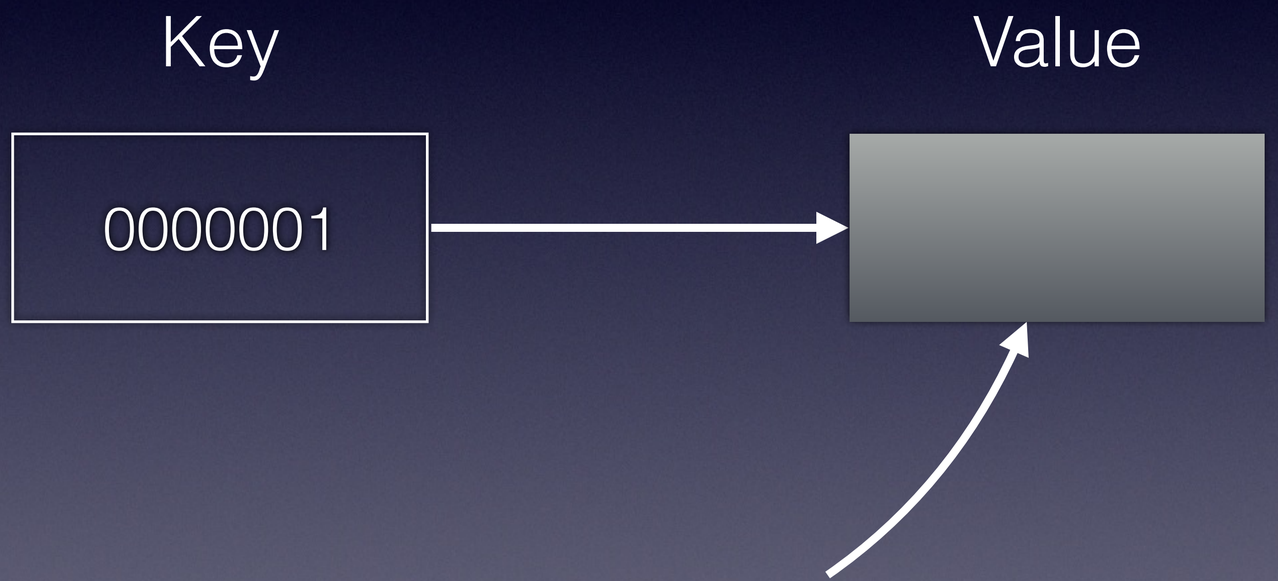
Key - Value Database

Essentially a distributed, persisted, hash table



Key - Value Database

Essentially a distributed, persisted, hash table



K-V Stores have no awareness of the internal structure of the Value
- can’t index on fields.

iDelta
advanced solutions

Key - Value Database

TERRACOTTA

ORACLE
Coherence

Infinispan

Red Hat JBoss Data Grid

WebSphere eXtreme Scale

Document Database

Value (typically JSON document)

```
{
“id“: 0000001,
“jBugName: "Scotland" ,
“members”: “500”,
“location”: SopraHQ,
“nextMeeting”: “300114
}
```

Document Stores CAN see the internals of the document being stored - can index on document fields.

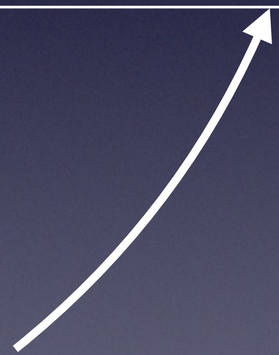
Document Database

Value (typically JSON document)

Example MongoDB query:

```
db.jbugs.find( { jBugName: "Scotland" } )
```

```
{
“id“: 0000001,
“jBugName: "Scotland" ,
“members”: “500”,
“location”: SopraHQ,
“nextMeeting”: “300114
}
```

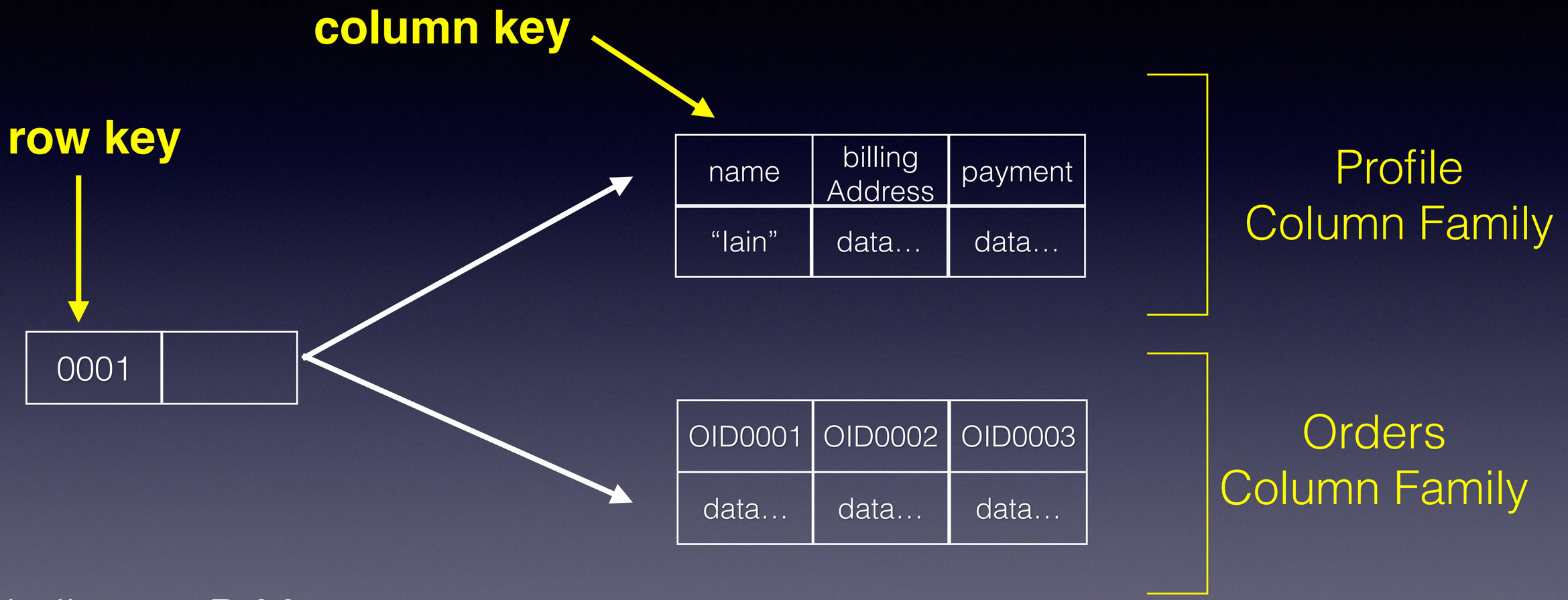


Document Stores CAN see the internals of the document being stored - can index on document fields.

iDelta
advanced solutions

Document Database

Column Family Databases



Similar to 2D Map:

```
get(001,name)
```

iDelta
advanced solutions

Column Family Databases

Example: Cassandra CQL

```
CREATE COLUMN FAMILY Jbugs
AND column_metadata = [
{column_name: jbugname, validation_class: UTF8Type}
{column_name: location, validation_class: UTF8Type}
{column_name: members, validation_class:UTF8Type}
];
```

```
SET Jbugs['scotland']['jbugname']='JBUG Scotland';
SET Jbugs['scotland']['location']='Edinburgh';
SET Jbugs['scotland']['members']='500';
```

iDelta
advanced solutions

Column Family Databases

iDelta
advanced solutions

Aggregate-Orientated DB's

"An Aggregate is a cluster of associated objects that we treat as a unit for the purpose of data change."

(*Evans, Domain Driven Design*)

iDelta
advanced solutions

Aggregate-Orientated DB's

Key-Value Database

- The aggregate is the value
- The aggregate is opaque
- Can only do key lookup for the whole aggregate value

iDelta
advanced solutions

Aggregate-Orientated DB's

Document Database

- The aggregate is the document
- The aggregate is transparent
- The aggregate is unstructured
- Can query inside the aggregate & retrieve parts of it.

iDelta
advanced solutions

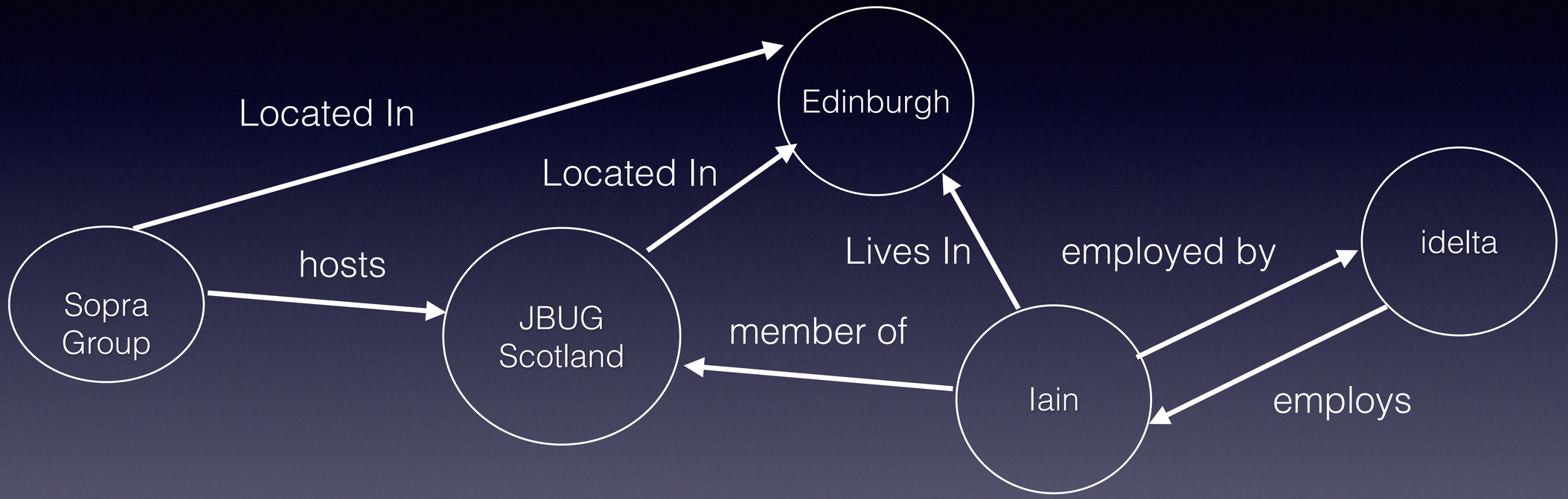
Aggregate-Orientated DB's

Column Family Database

- The aggregate is the set of column family'
- The aggregate is transparent
- The aggregate has a structure imposed
- The structure improves performance

Graph Databases

Allow you to persist entities and relationships between entities.



Node - represents an entity

Edge - represents relationships between nodes

iDelta
advanced solutions

Use Cases

Storing Session Data Key-Value Database

iDelta
advanced solutions

Use Cases

Storing Session Data	Key-Value Database
Logging Event Data	Document Database

iDelta
advanced solutions

Use Cases

Storing Session Data	Key-Value Database
Logging Event Data	Document Database
	Column Family Database

iDelta
advanced solutions

Use Cases

Storing Session Data	Key-Value Database
Logging Event Data	Document Database
	Column Family Database
Recommendation Engine	Graph Database

iDelta
advanced solutions

noSQL or RDMS ?

- Exercise caution in selecting a noSQL over RDMS.
- RDMS technology is mature
- Organisational investment in RDMS technology and expertise
- Require complex transactional logic / JOINS - RDMS
- Improve developer productivity - noSQL
- Deal with very large data volumes, reduce access times, increase throughput - noSQL

iDelta
advanced solutions

Summary

- 4 Categories of noSQL Database

  K-V,
  Document,
  Column Family,
  Graph

- Aggregate-Orientated Databases
- Column Family optimised for access, provides richest data model

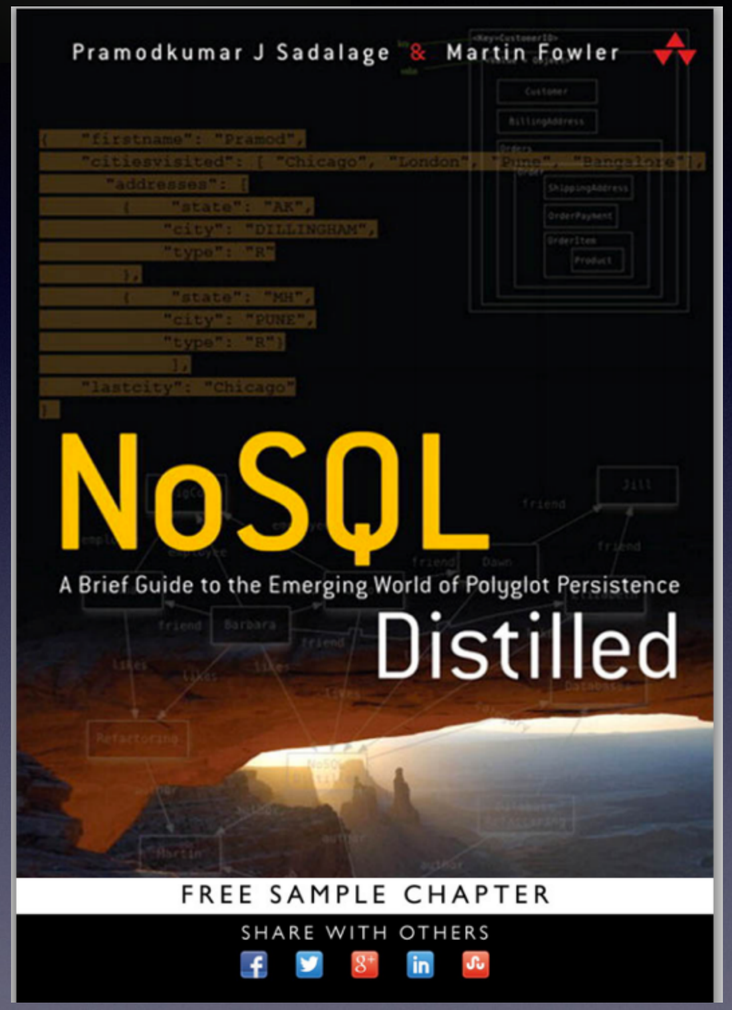
iDelta
advanced solutions

Summary

- 4 Categories of noSQL Database
  - K-V,
  - Document,
  - Column Family,
  - Graph
- Aggregate-Orientated Databases
- Column Family optimised for access, provides richest data model
- **Didn't cover:** Transactions, ACID, CAP Theorem, etc

iDelta
advanced solutions

References



NoSQL Distilled
A Brief Guide to the Emerging
World of Polyglot Persistence
Pramod J. Sadalage
Martin Fowler

A readable introduction to noSQL technology, covering it's history, implementations, use cases, and place in an enterprise environment. A very useful source of introductory material about noSQL.

- **Google** - lot's of good blog posts, articles, technotes, documentation, etc, out there.