JBoss Community

Eventual Consistency in Infinispan

Manik Surtani
msurtani@redhat.com
Founder and Project Lead, Infinispan

Who is Manik?

- R&D Engineer at JBoss by Red Hat
- Founder and Project Lead, Infinispan
- Spec lead, JSR 347
  - Data Grids for Java
- EG representative, JSR 107
  - Temporary Caching for Java

- http://blog.infinispan.org
- http://twitter.com/maniksurtani

JBoss Community

Agenda

- Eric Brewer's CAP Theorem
- Consistency models
- NoSQL, Data Grids and CAP
- Eventual Consistency and you
- Consistency in Infinispan
- JSR 347 and consistency

Prof. Eric Brewer

CAP Theorem

Three desirable characteristics of a distributed system:

Consistency

Availability

Partition-tolerance

CAP Theorem

Consistency

all nodes see the same data at the same time

CAP Theorem

Availability

a guarantee that every request receives a response about whether it was successful or failed

CAP Theorem

Partition-tolerance

the system continues to operate despite arbitrary message loss

JBoss Community

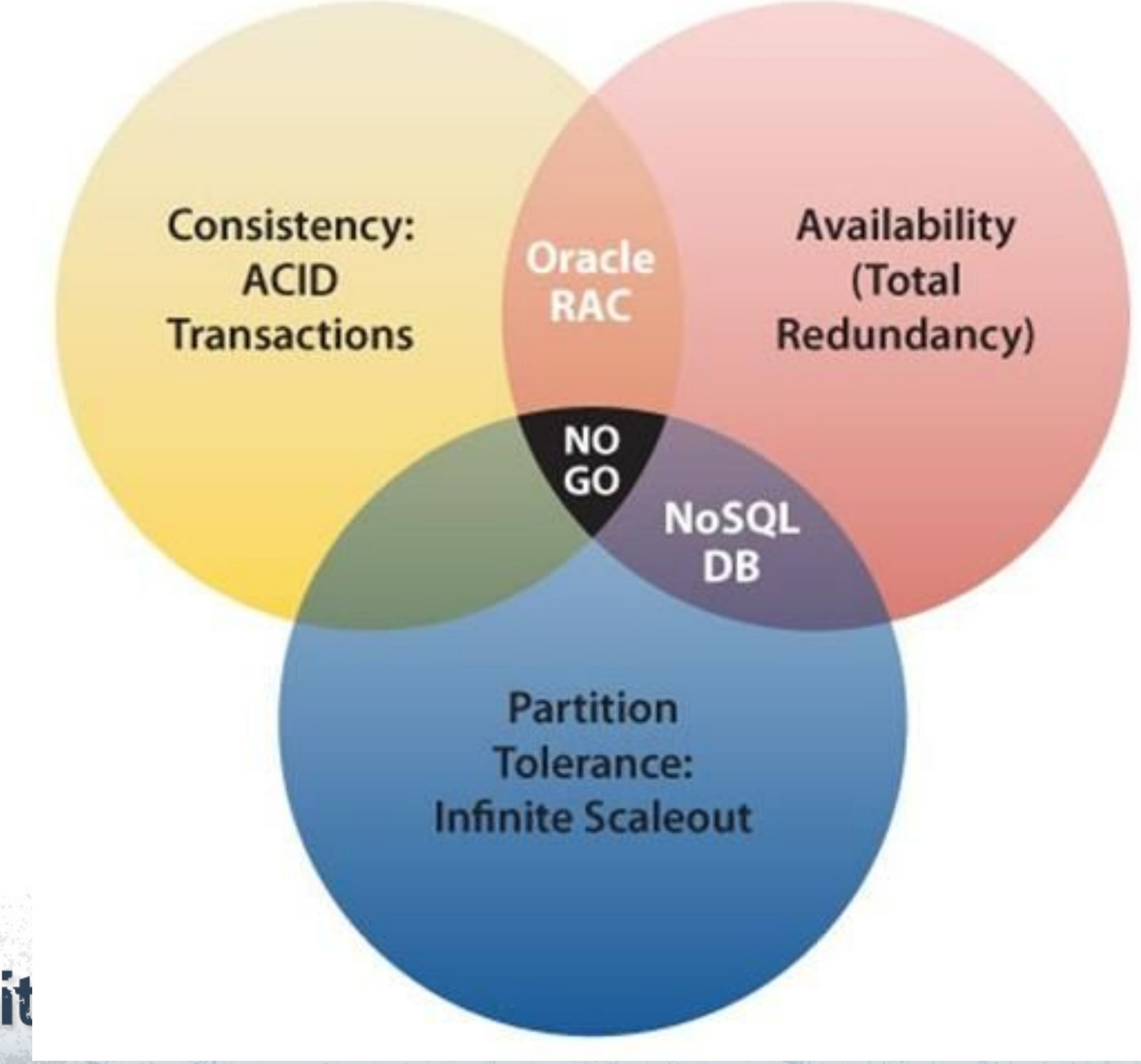
Networks are unreliable!

Split Brains

Prof. Eric Brewer

JBoss Community

CAP Theorem


Consistency: ACID Transactions
Oracle RAC
Availability (Total Redundancy)
NO GO
NoSQL DB
Partition Tolerance: Infinite Scaleout

CONSISTENCY
It's Only a Virtue if You're Not a Screwup.
www.despair.com

Eventual Consistency

Not That Scary

JBoss Community

Eventual Consistency

Not That Scary

- The real world is eventually consistent and works (mostly) fine
- “Eventual” doesn’t mean minutes, days, or even seconds in non-failure cases
- DNS, HTTP with Expires: header
- How you model the real world matters

ACID

Atomicity
Consistency
Isolation
Durability

vs.

BASE

Basically **A**vailable
Soft state
Eventually consistent

JBoss Community

Typically ACID

Relational Databases

Oracle DB
MySQL
PostgreSQL
...

Data Grids

Coherence
Gigaspaces
Infinispan
...

Typically ACID

Relat

Oracl

MySQ

Postg

…

ids

ce

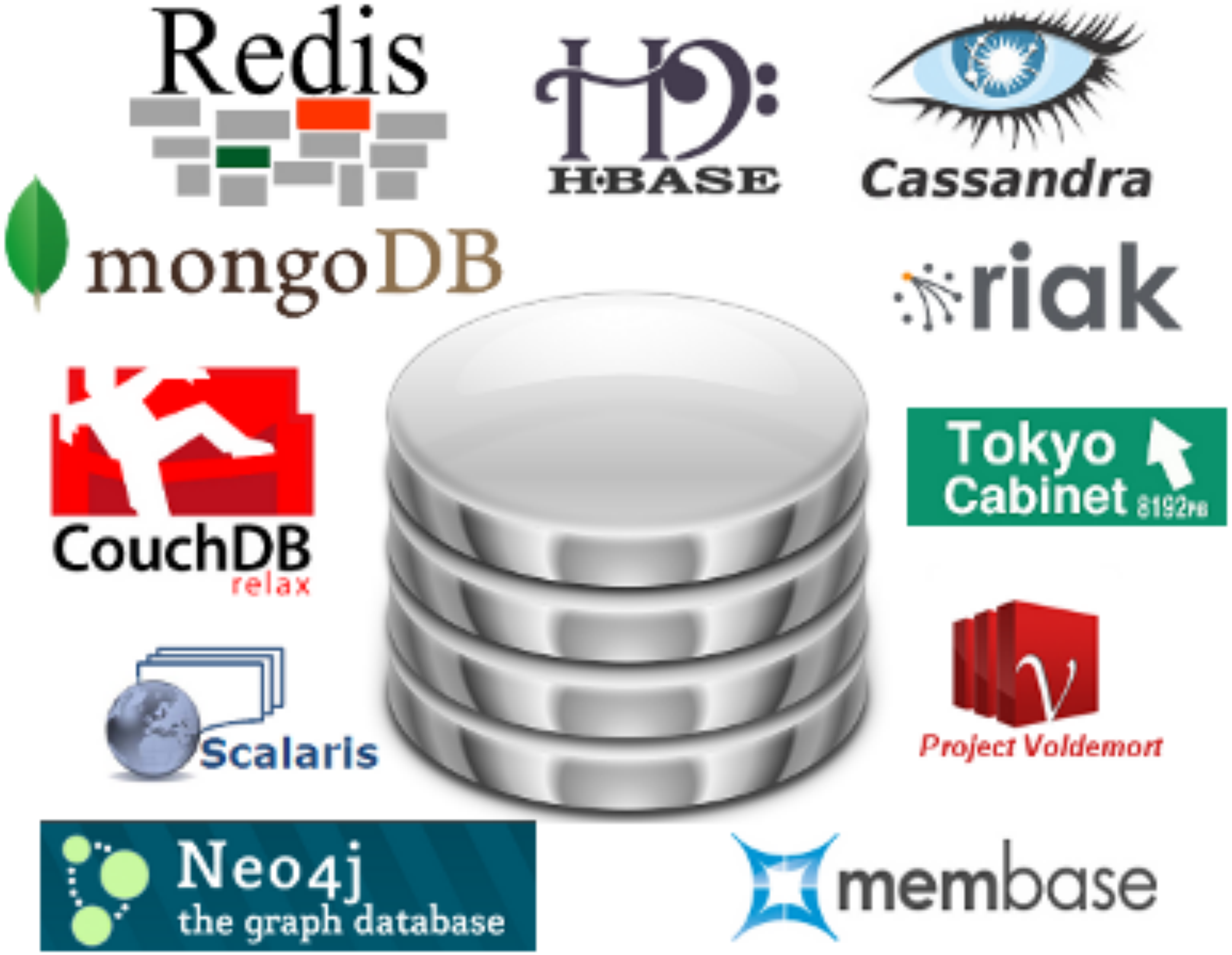
ces

an

DON’T COPE WELL WITH NETWORK PARTITIONS!

JBoss Community

SQL


Redis
HBASE
Cassandra
mongoDB
riak
CouchDB
relax
Tokyo Cabinet 8192PB
Scalaris
Project Voldemort
Neo4j
the graph database
membase

Dealing with Eventual Consistency

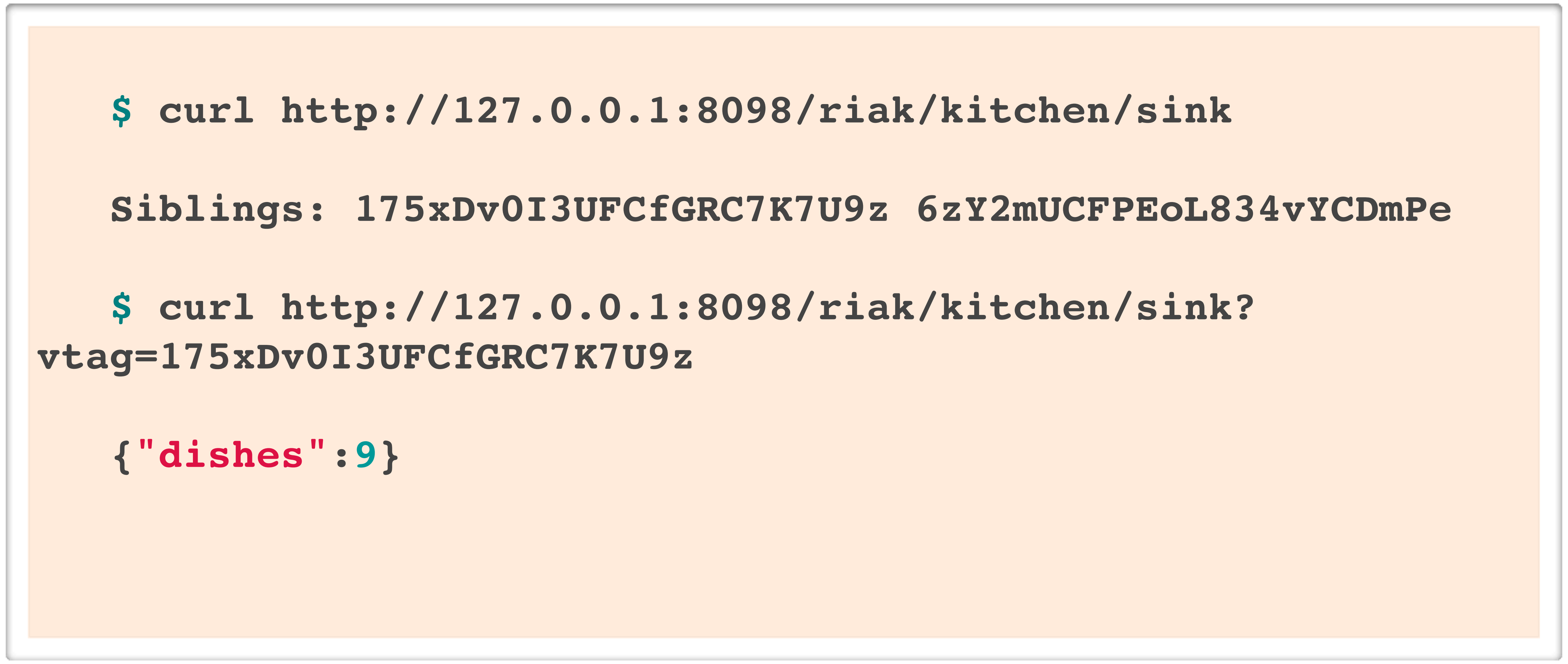
Challenge #1

Your app must deal with multiple versions of data

Dealing with
Eventual Consistency

Challenge #1

Your app must deal with multiple versions of data

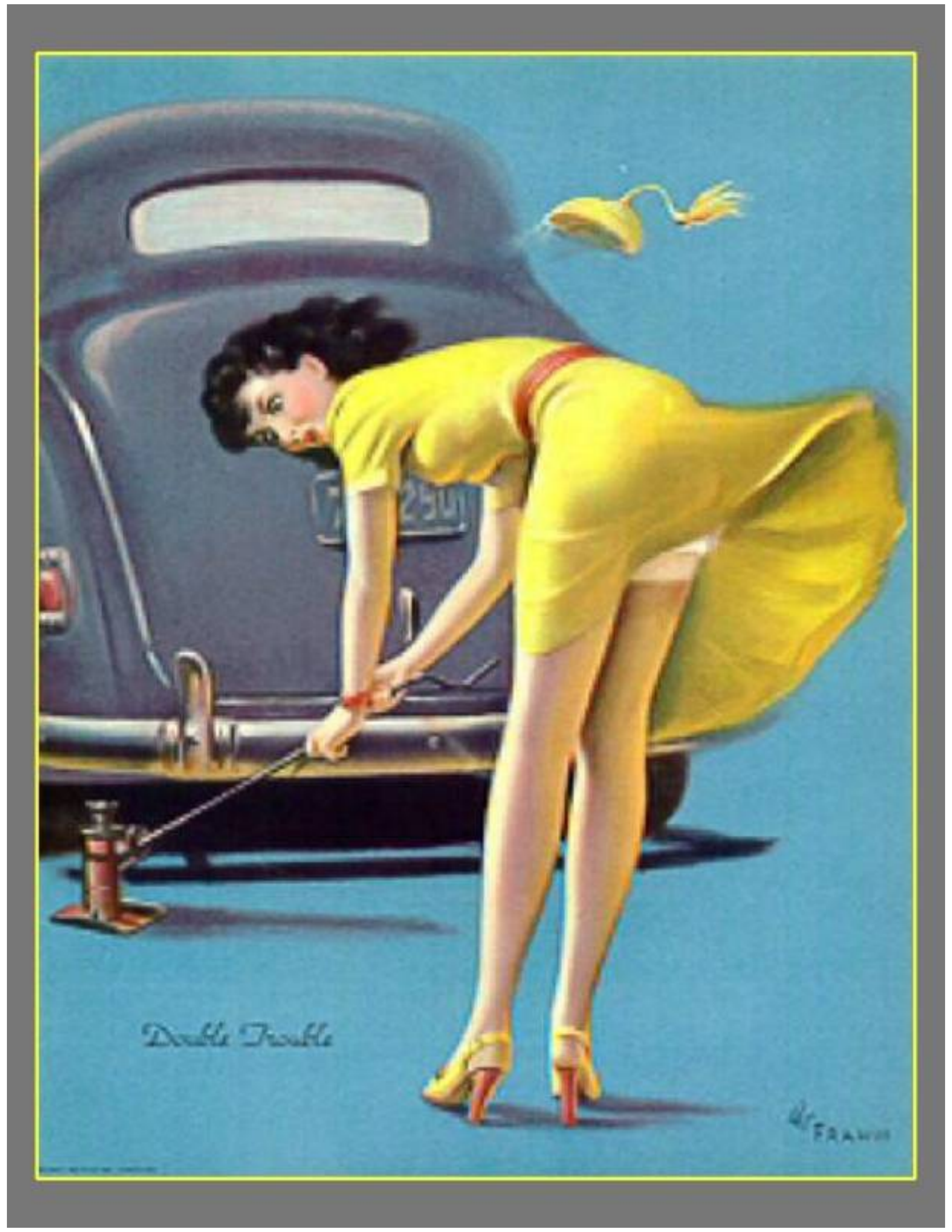


```
   $ curl http://127.0.0.1:8098/riak/kitchen/sink

   Siblings: 175xDv0I3UFCfGRC7K7U9z 6zY2mUCFPEoL834vYCDmPe

   $ curl http://127.0.0.1:8098/riak/kitchen/sink?
vtag=175xDv0I3UFCfGRC7K7U9z

   {"dishes":9}
```

Dealing with Eventual Consistency

Challenge #2

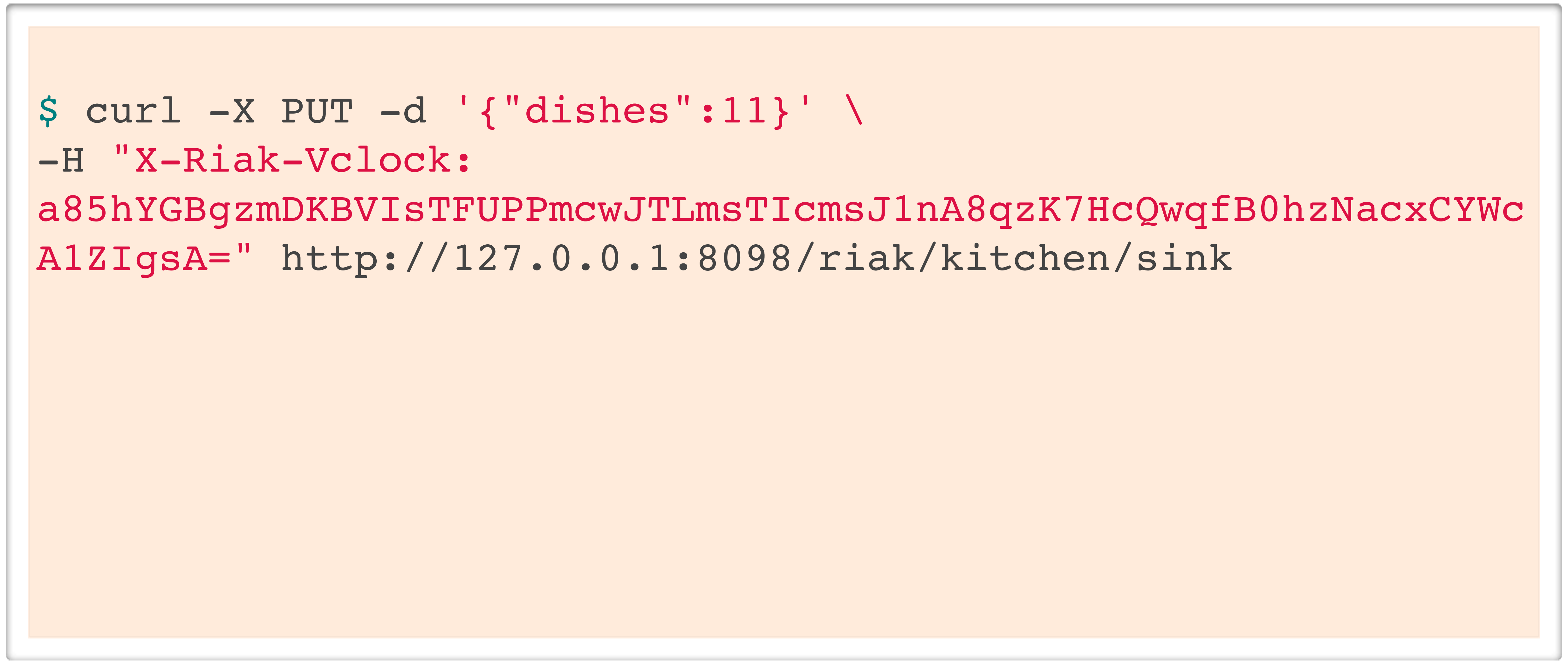
Your app must be able to fix inconsistent state



Dealing with Eventual Consistency

Challenge #2

Your app must be able to fix inconsistent state



```
$ curl -X PUT -d '{"dishes":11}' \
-H "X-Riak-Vclock:
a85hYGBgzmDKBVIsTFUPPmcwJTLmsTIcmsJ1nA8qzK7HcQwqfB0hzNacxCYWc
A1ZIgsA=" http://127.0.0.1:8098/riak/kitchen/sink
```

Consistency in Infinispan

- Strongly consistent
  - Support for XA transactions
- Weak on partition tolerance
  - Healing network partitions are reliably detected
  - But not much can be done about it
  - Data inconsistent!
- Same as most other data grids

JBoss Community

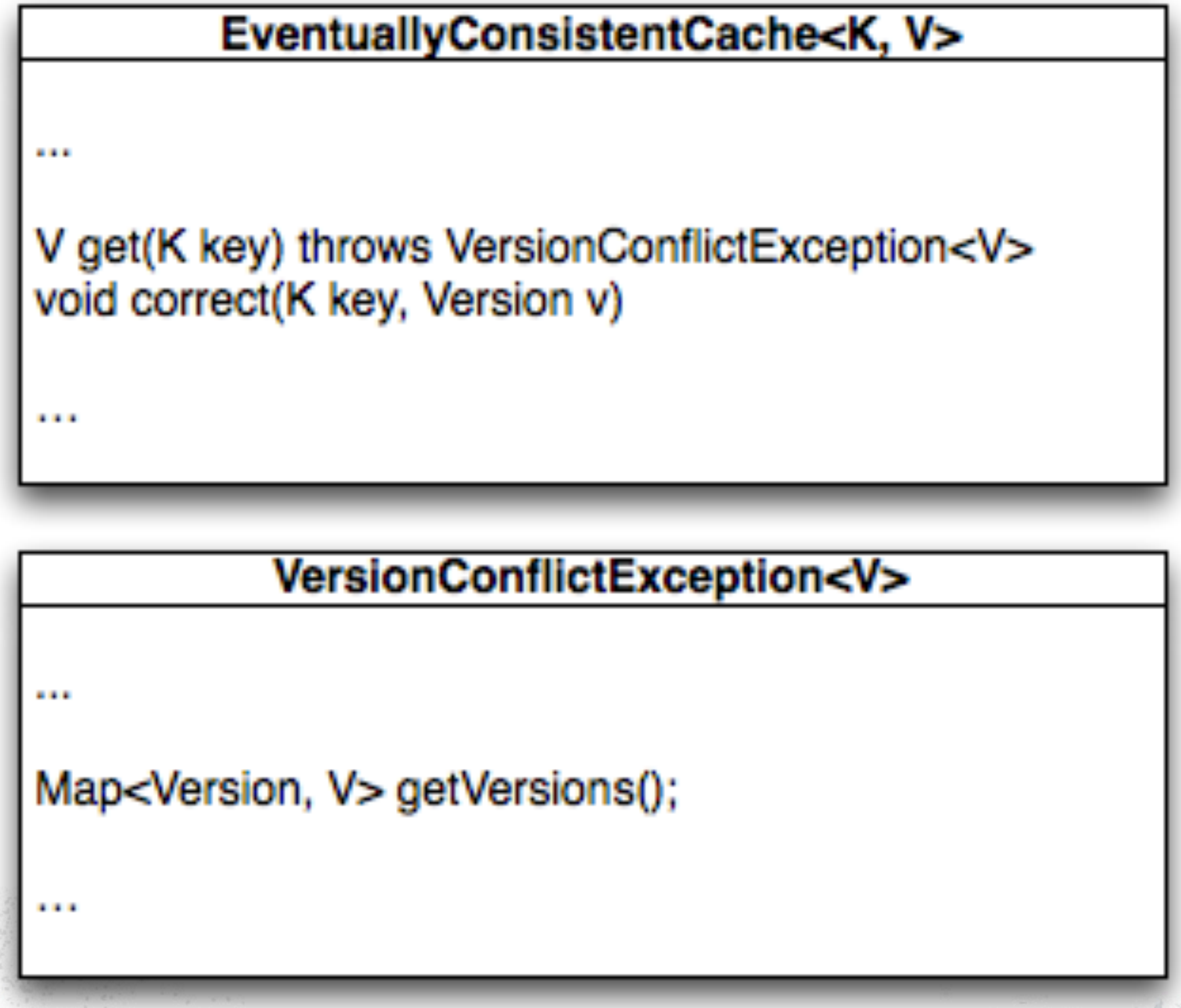
Time for a change

JBoss Community

Eventually Consistent Infinispan

- All entries will be versioned
  - Vector clocks to detect causal order during a partition
- Expose API to deal with multiple return values
- Expose API to allow applications to correct
- Quorums

Eventually Consistent Infinispan



```
EventuallyConsistentCache<K, V>

...

V get(K key) throws VersionConflictException<V>
void correct(K key, Version v)

...
```

```
VersionConflictException<V>

...

Map<Version, V> getVersions();

...
```

JSR 347

- A new standard for data grids for Java
- Built on top of JSR 107
- Additional features:
  - Async/non-blocking API
  - Grouping API
  - Map/Reduce API
  - Eventually Consistent API

Want to learn more about JSR 347?

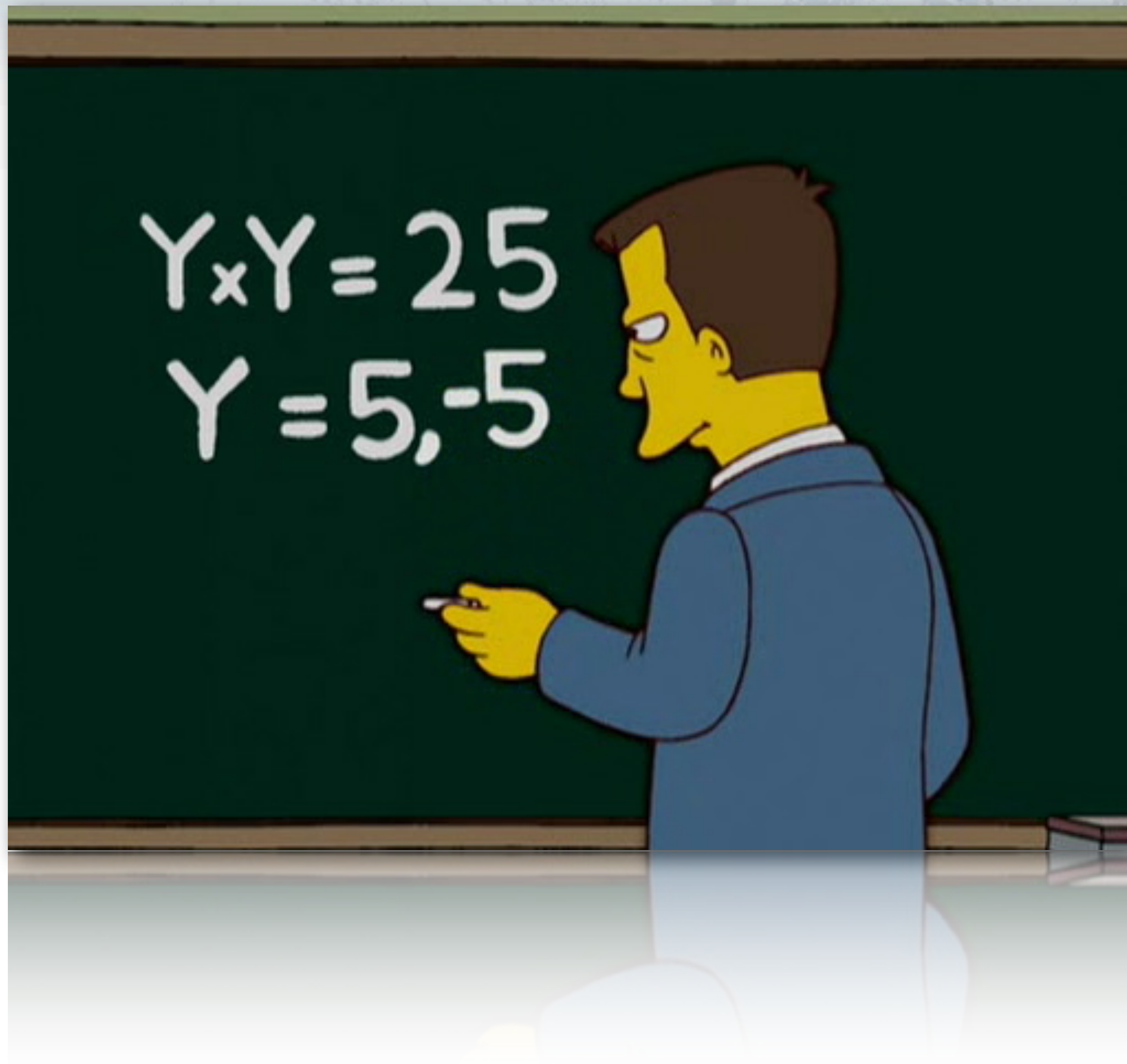
JavaOne Talk

http://db.tt/uRIu12IU

GitHub

http://github.com/datagrids/spec/wiki

To Summarize



- Discussed the CAP theorem
- Consistency models
- Consistency in RDBMSs, NoSQL and data grids
- Applications dealing with eventual consistency
- Infinispan, JSR 347 and eventual consistency

Questions & More Info

- http://www.infinispan.org
- http://twitter.com/infinispan

- http://github.com/datagrids/spec/wiki
- http://groups.google.com/group/jsr347
- http://twitter.com/jsr347