Riak

a distributed, web-inspired database

NoSQLBerlin'09
Martin Scholl <ms@diskware.net>
@zeit_geist

Historical Notes

- Riak is Basho Inc's brainchild
- Apache 2.0 licensed
- first public release 09/08/07
  - http://riak.basho.com/
  - http://bitbucket.org/justin/riak
  - http://github.com/zeitgeist/riak

1. Overture

What is Riak?

- a lot of Twitter fame recently
- uses a bunch of buzzword technology
  - its so NoSQL, MapReduce and that stuff
  - written in Erlang
  - even your mother-in-law loves Riak
- obvious question: how awesome is it really?

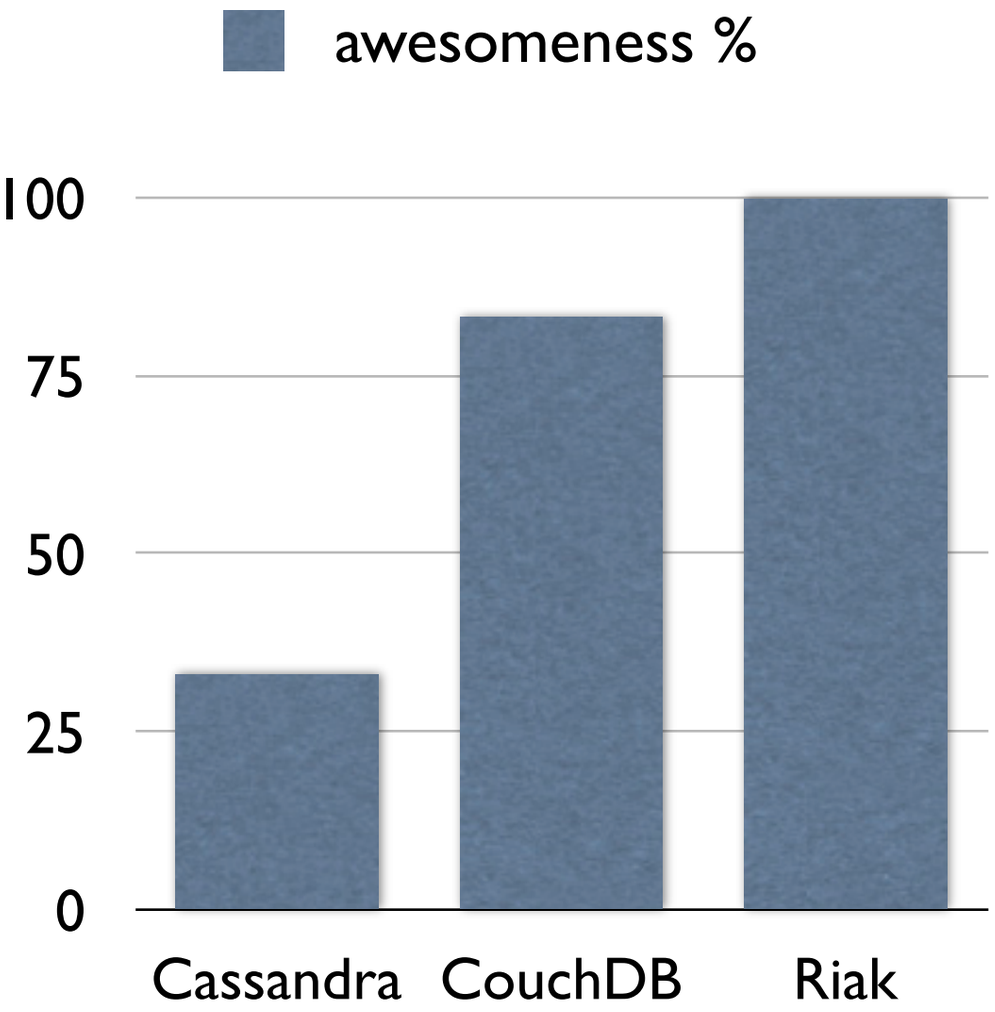
Scientific Model of Awesomeness

	Cassandra	CouchDB	Riak
cool?	✓	✓	✓
distributed	✓		✓
HTTP/REST		✓	✓
JSON		✓	✓
Erlang		✓	✓
M/R		✓	✓

We have a winner

- result of a fair and objective competition:

Riak is 100% awesome



2. The Serious Part

(caffeine will be served in 42 minutes)

What Riak really is

- Distributed Data Storage System (DDSS)
- BASE
- Dynamo inspired
- Erlang implemented
- MapReduce'ing
- Textbook style DDSS implementation

Data Model

- Data-Sphere: *Bucket x Key x Document*
- *Bucket*: a named scope of keys and values
  - created implicitly, on demand
  - has constraints
- *Key*: choose freely

Document Model

- Documents hold the actual data
- actual data can be virtually anything
- internal data format: Erlang-Tuple
- current gold-standard: JSON objects
- model the Web's nature
  - *embedded doc-links!*

2.1 A tour through Riak

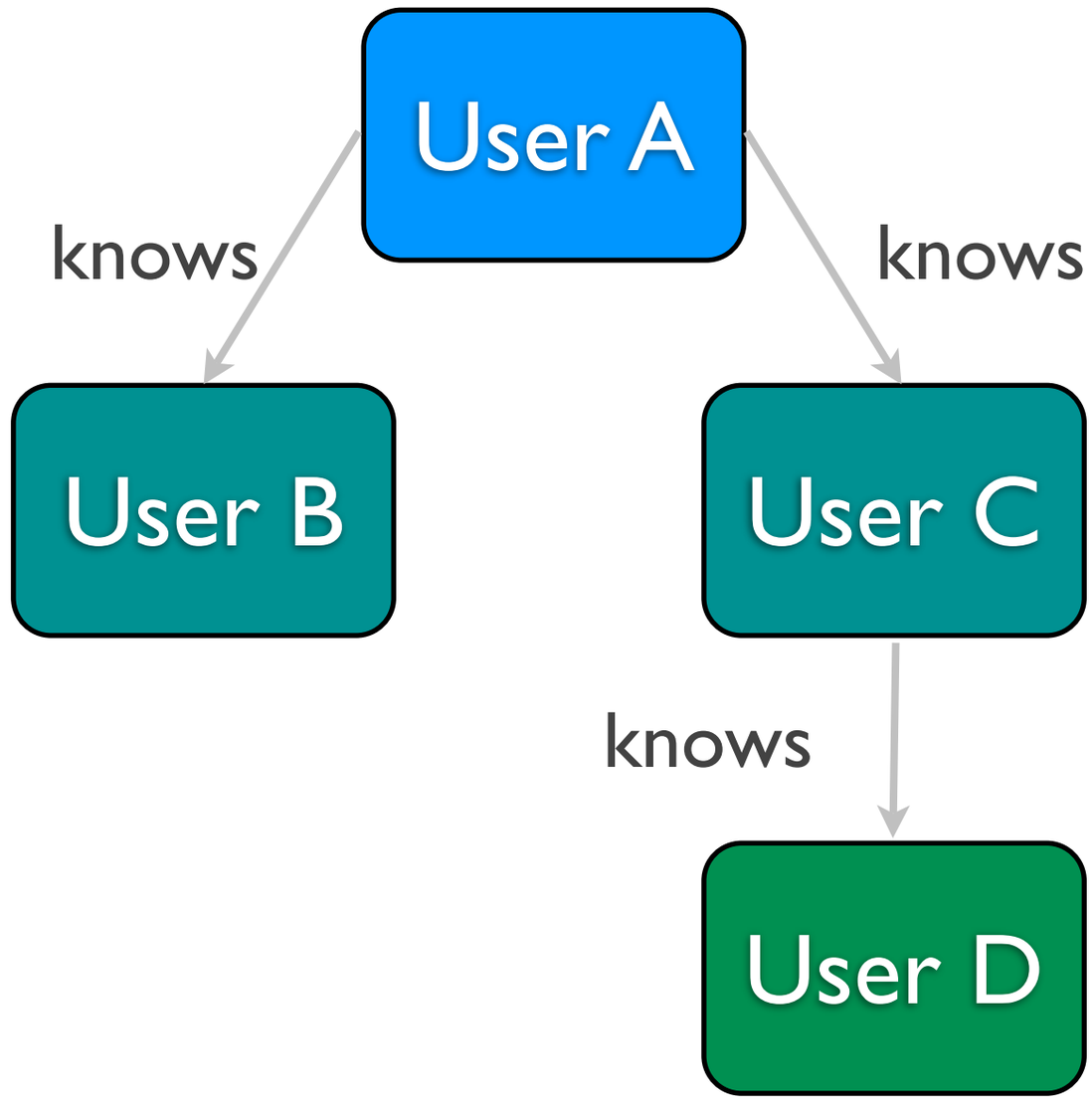
We jump off cliff HTTP/REST and land in Riak's guts

HTTP/REST JSON-API

- `GET /jiak/<bucket>/<key>`
  - fetch a document
- `POST /jiak/<bucket>`
  - create a new entry, key gets generated
- `PUT /jiak/<bucket>/<key>`
  - create / update a doc

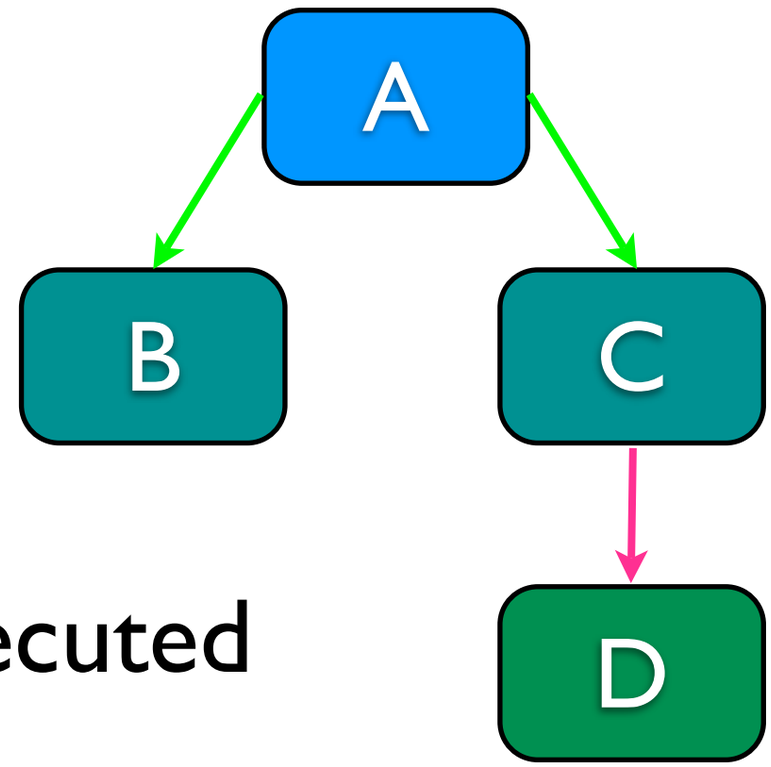
JSON Documents

```
{
 bucket:"users",
 key:"A"
 object: {
  name:...
 }
 links:[
  ["users","B","B"],
  ["users","C","C"]
 ]
}
```



MapReduce Links

- query Documents via M/R
- model Graph Structure
  - *chain* M/R stages
- Map and Reduce: parallel executed
- M/R via HTTP/REST:
  - `GET /jiak/<Bucket>/<Key>[/<MR>]+`



M/R Example

- Link: `[<B>,<K>,<T>]`
- M/R: `<B>,<K>,<T>`
- get A's friends

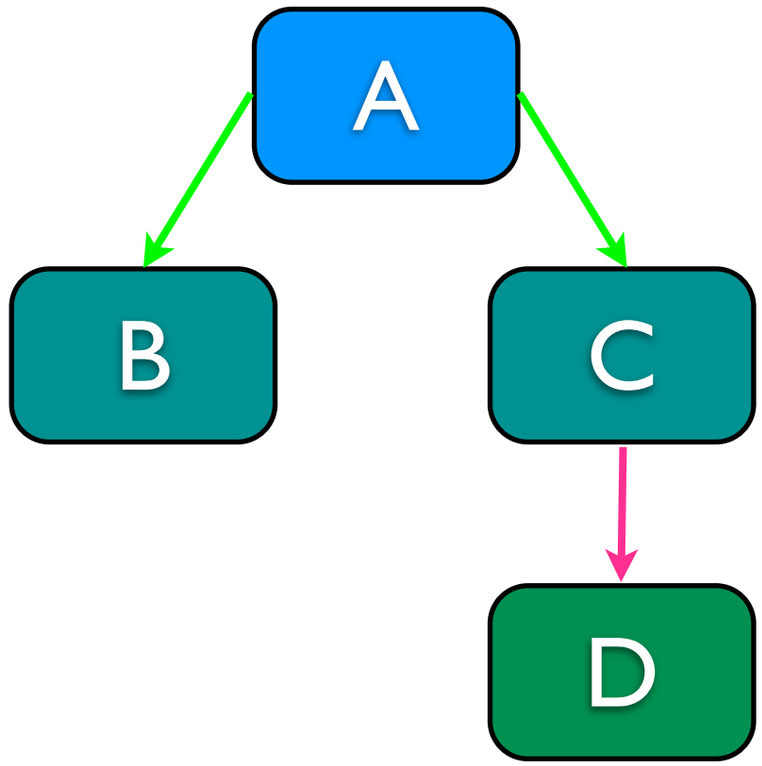
  `GET /jiak/users/A/`

  `users,_,_`

- get A's friends' friends

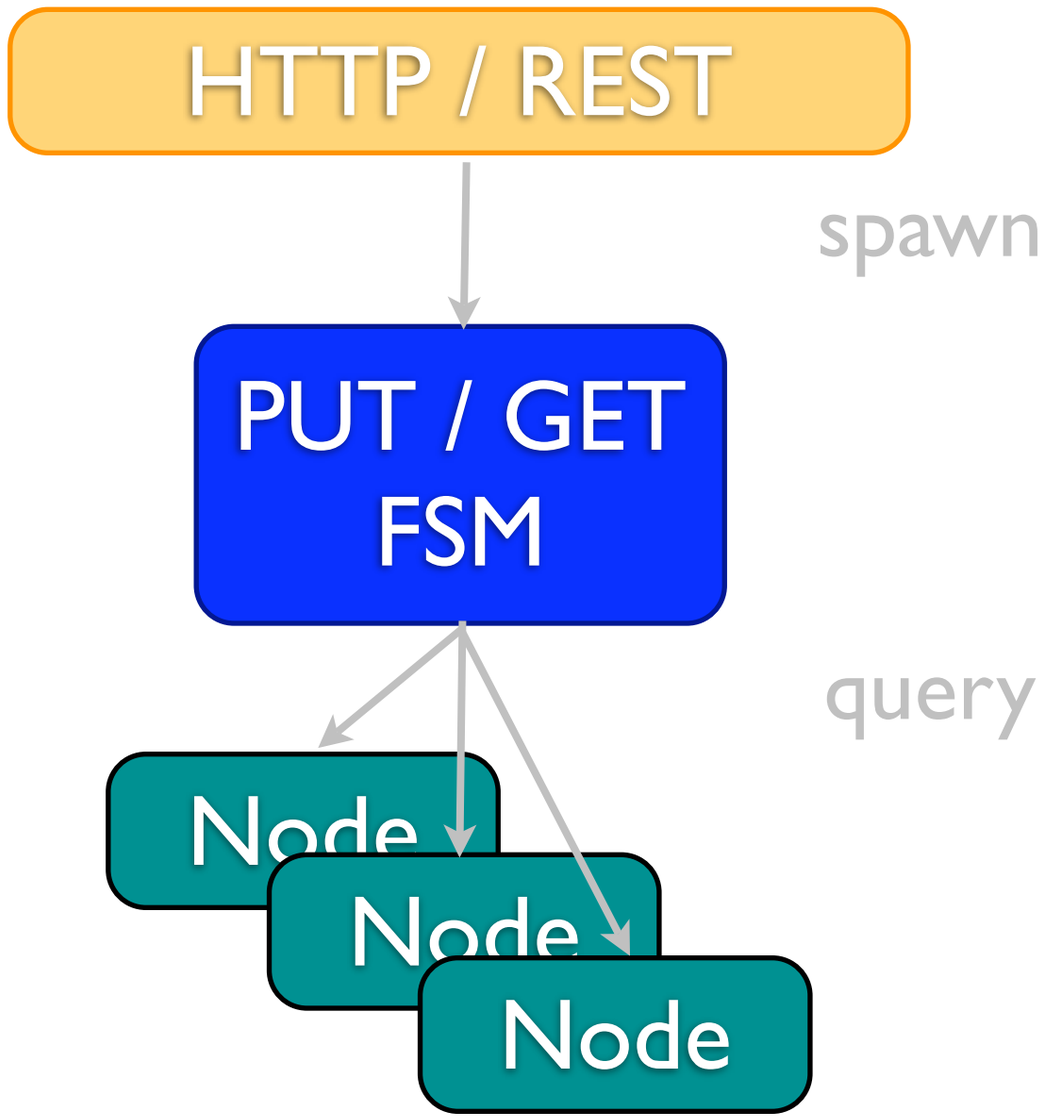
  `GET /jiak/users/A/`

  `users,_,_/users,_,_`



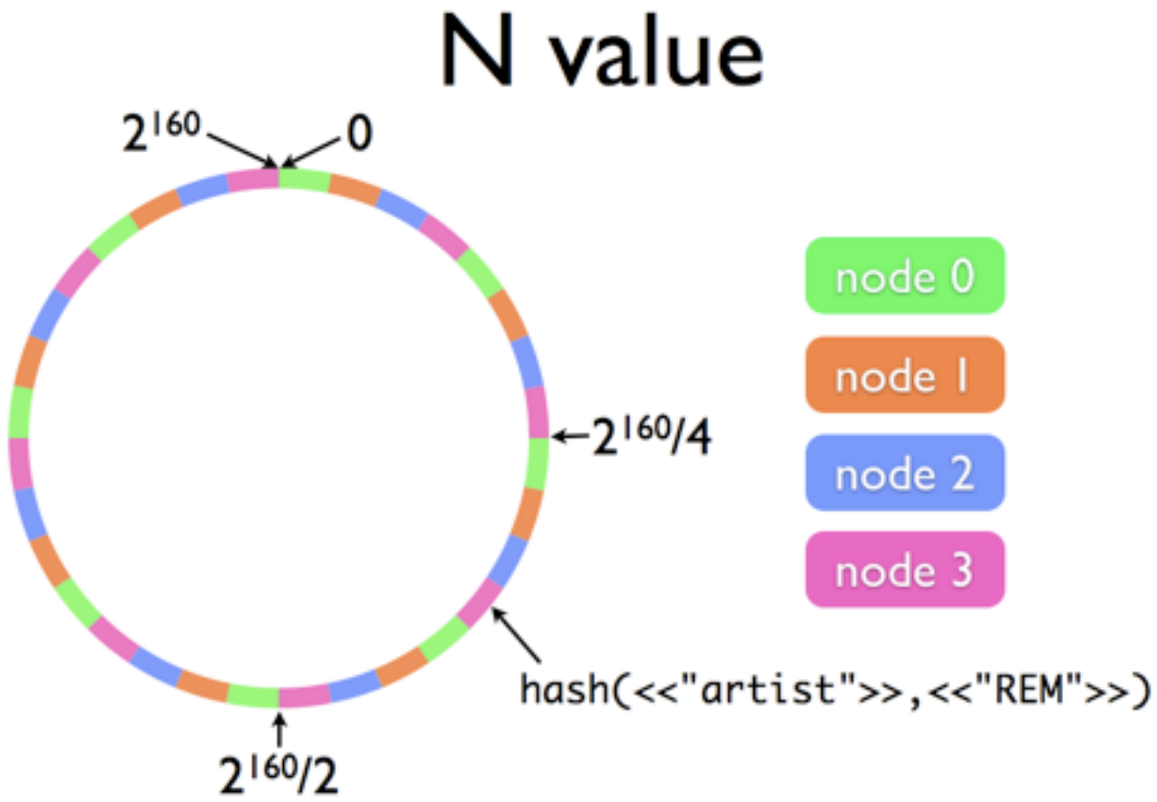
Request processing

- REST API is transparent
- Each Request is modelled as an Erlang process
- different FSMs for Put, Get, Map and Reduce operations.



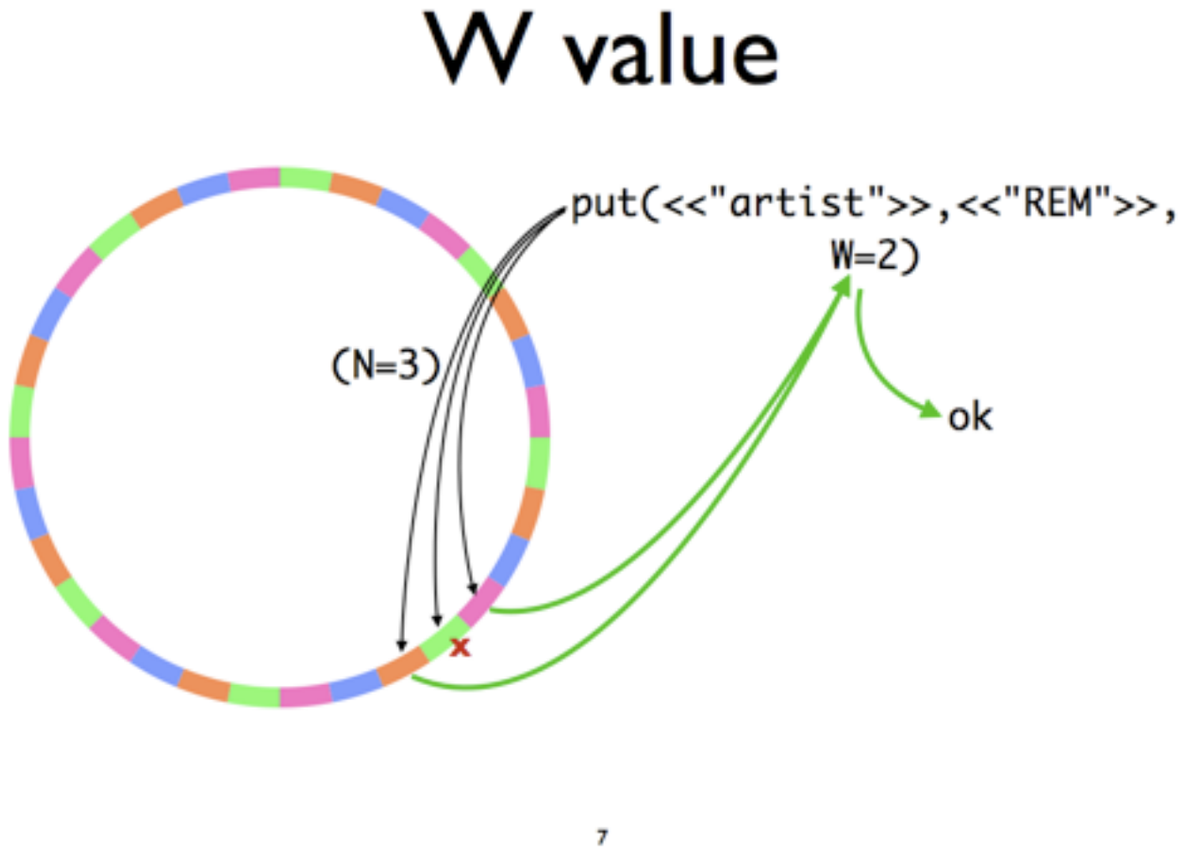
The Ring

- Ring: a fixed-size distribution map
- data-base for determining nodes responsible for a key
- `hash: (B x K) -> 160b`
- `filtered_preflist: (Ring x 160b)->Node`

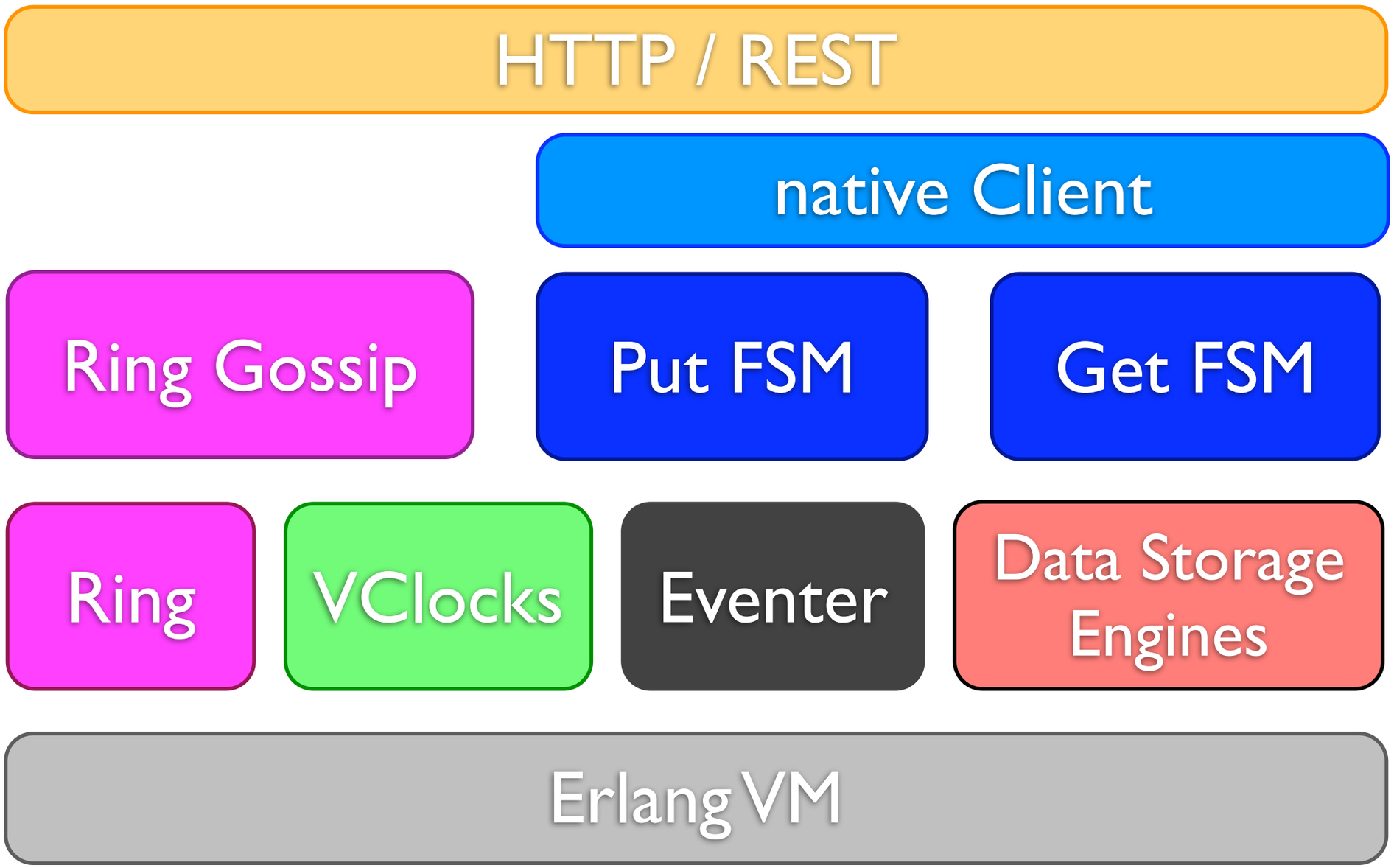


Request Distribution

- eventual consistency
- `N` or `n_val`: # replicas
- `R`: min `get()`s
- `W`: min `put()`s
- implemented as Erlang `gen_fsm` processes



The Big Picture



Riak is a DDSS Minix

- Riak's kernel: ~3.5k LOC!
- Riak is more than a Document DB
- clean and self-documenting codebase
- extensible in many ways
- *Riak is a perfect fit for building reliable and scalable custom data storage systems!*

Thank you

Riak is more:
http://riak.basho.com/

don't hesitate to contact me
[to talk about e.g. Riak,
Distributed systems, Erlang, etc.]

Martin Scholl
<ms (at) globalinfinity.de>
global infinity GmbH