

Tuesday, June 4, 13

Jan Lehnardt

jan@apache.org

@janl



Tuesday, June 4, 13

Thanks to Julien for inviting me.

It is great to be hear. I learned a lot, met a lot of great people, hope to meet the rest of you too.

Thanks to Magnus & other folks who had trouble with the network.



The CouchDB Implementation

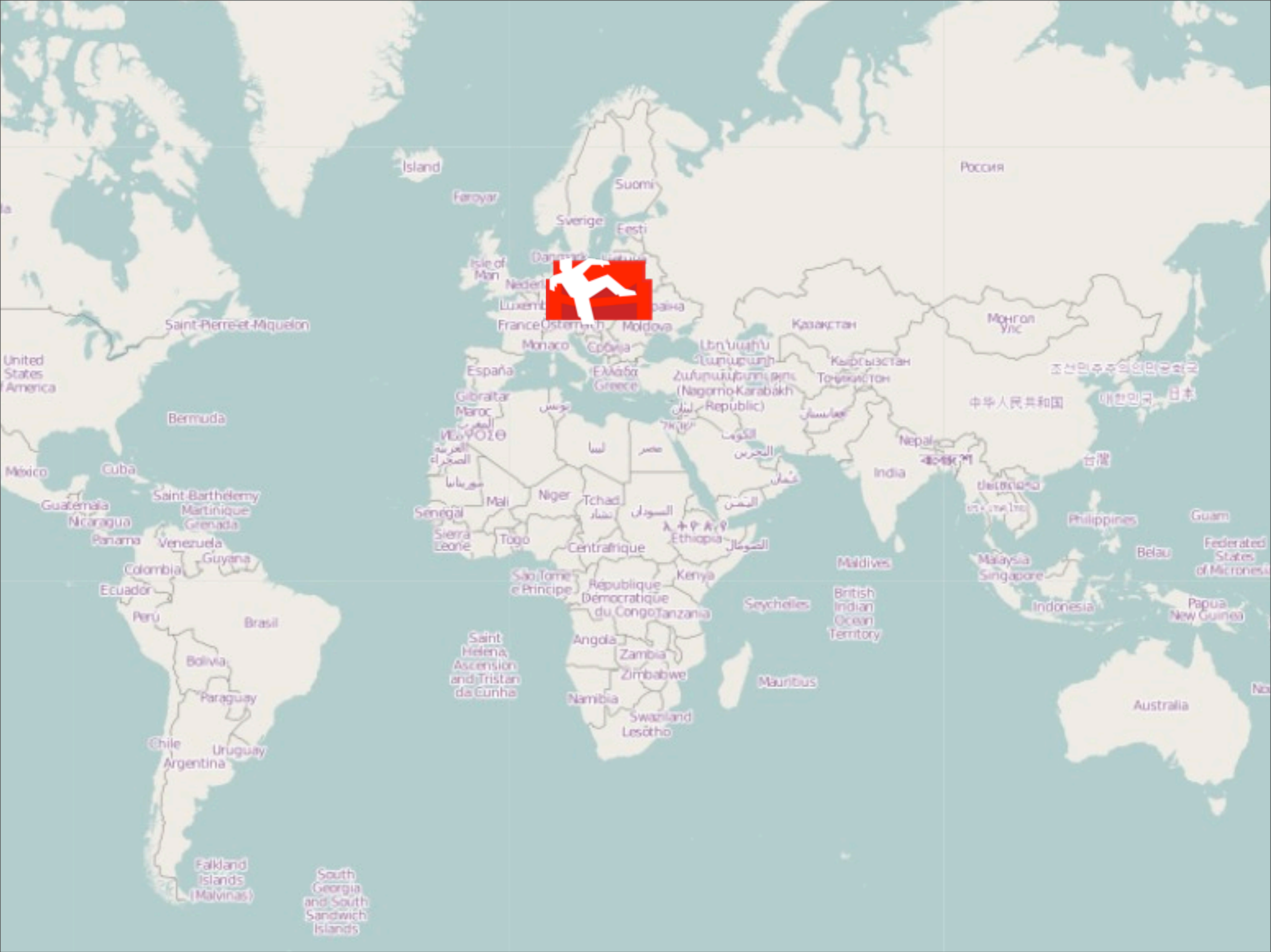
**CouchDB is a
database that
replicates.**

**«Think of CouchDB
as `git` for your
application data.»**

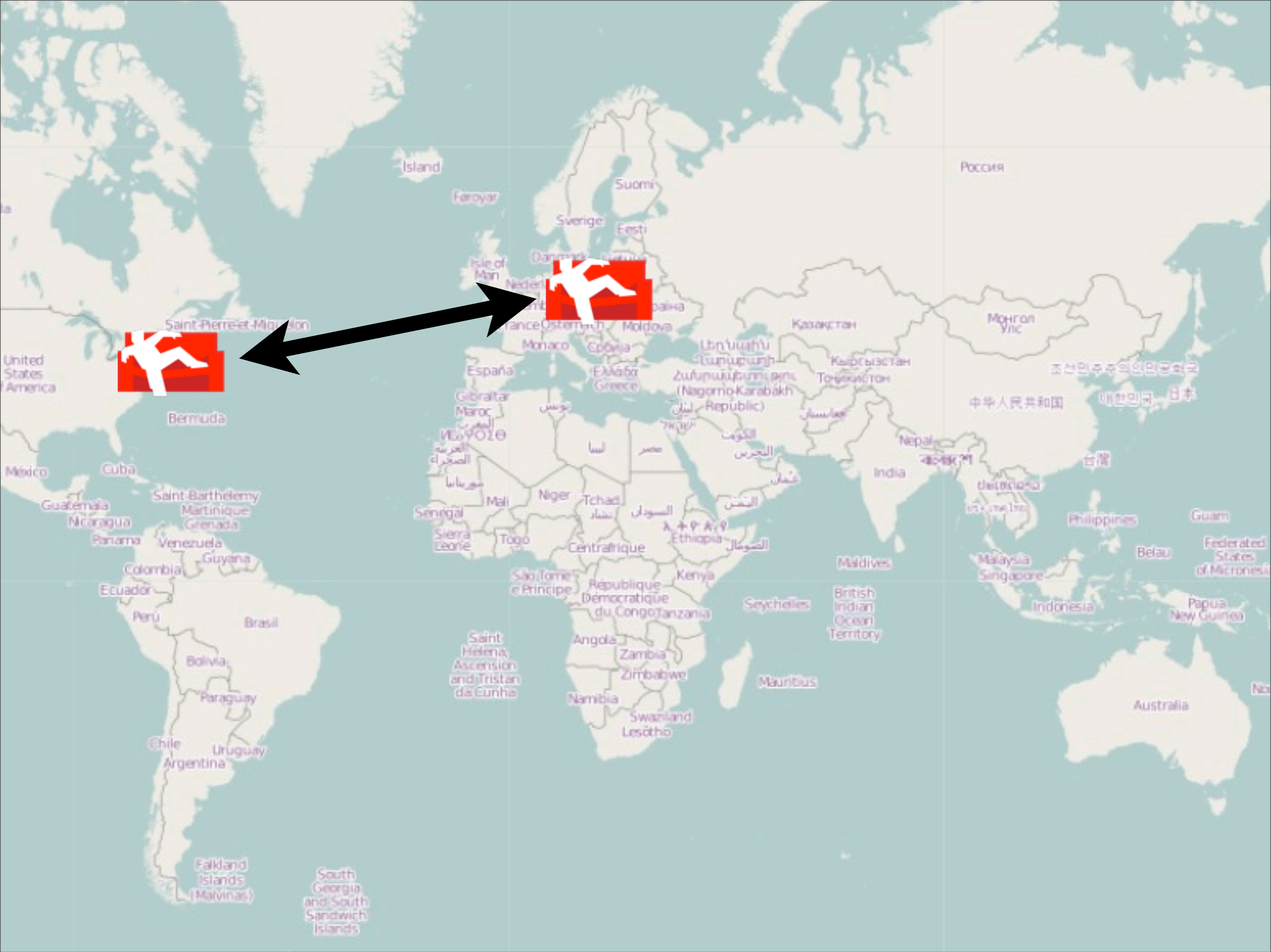
— Jan Lehnardt, Berlin Buzzwords

Tuesday, June 4, 13

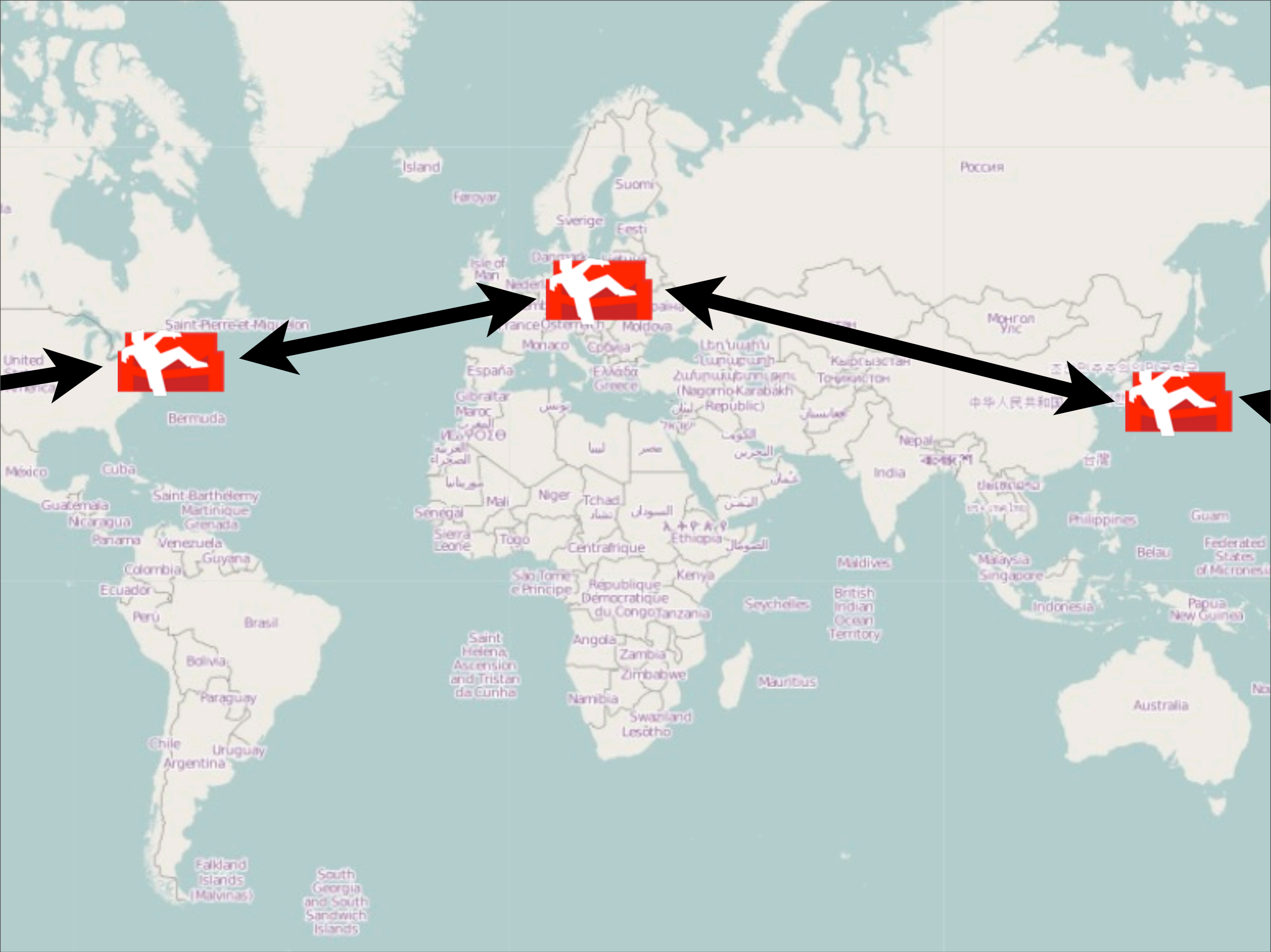
multiple locations / push / pull



Tuesday, June 4, 13



Tuesday, June 4, 13



Tuesday, June 4, 13



PouchDB

TouchDB



Any Database

Tuesday, June 4, 13



API

CORE FEATURES

CORE DATA STRUCTURES

FILE SYSTEM ACCESS

FILE SYSTEM



COUCH_HTTP*

COUCH_DOC COUCH_MR COUCH_REPL..

COUCH_BTREE

COUCH_FILE

FILE SYSTEM

Core Datastructures

Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top

**Behold the
b+tree**



Tuesday, June 4, 13



© M D HEPPLWHITE & WITKOPPEN WILDFLOWER NURSERY, 2011

Tuesday, June 4, 13

by-id

A													
B													
C													
D													
E													
F													
G													
H													
I													
J													
K													
L													
...													

A													
B													
C													
D													
E													
F													
DOC_G													
H													
I													
J													
K													
L													
...													

A														
B														
C														
DOC_D														
E														
F														
DOC_G														
H														
I														
J														
K														
L														
...														

A
B
C
DOC_D
E
F
DOC_G
H
I
J
DOC_K
L
...

DOC_D

DOC_G

DOC_K

by-sequence

“what

happened

since?”

1. DOC_G

2. DOC_D

3. DOC_K

The CouchDB File Format

Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top

HEADER



Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top

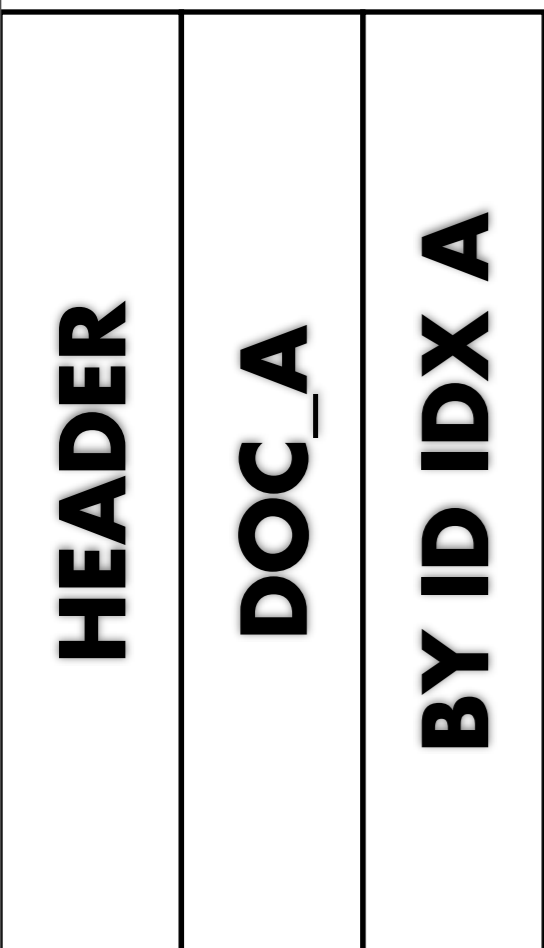
HEADER

DOC_A



Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control
- Can answer:
 - Data for \$key
 - What happened \$since
- Used for core data storage
- As well as indexes
- Everything else is built on top



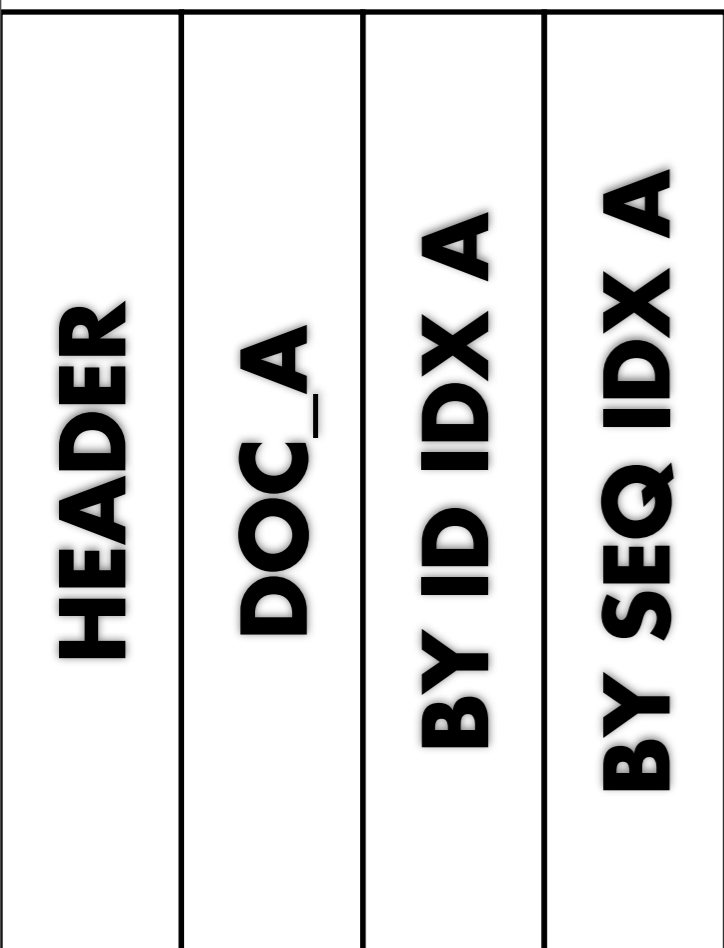
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



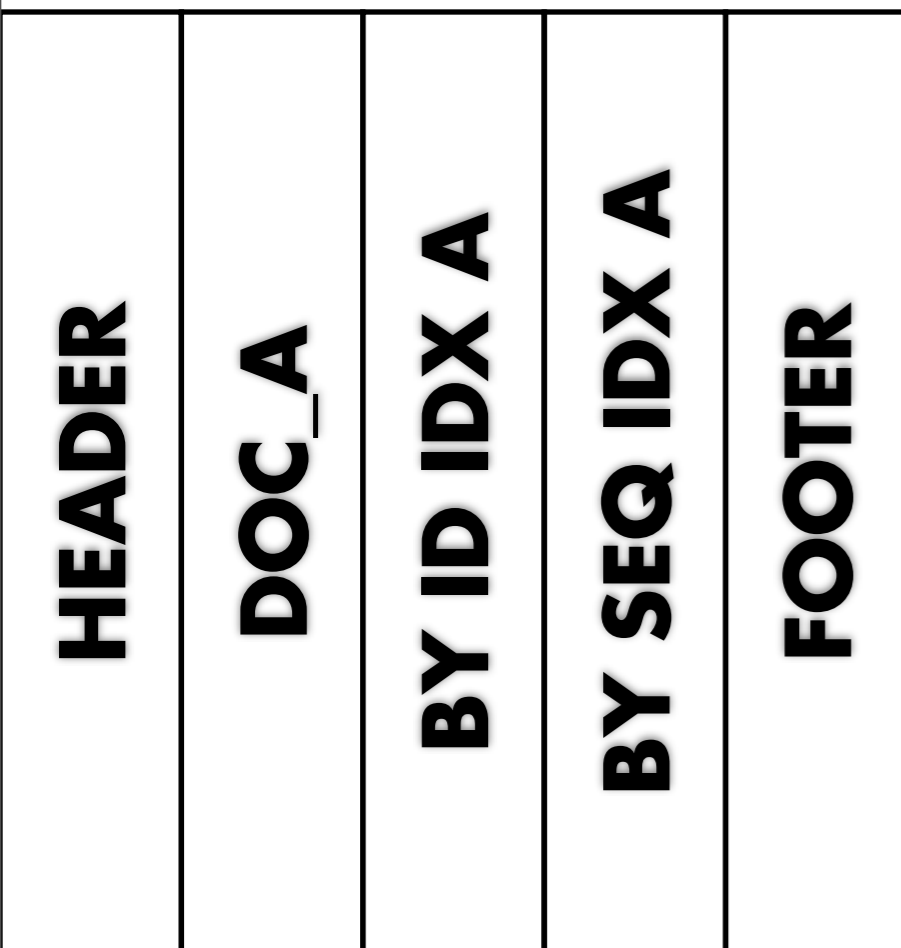
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



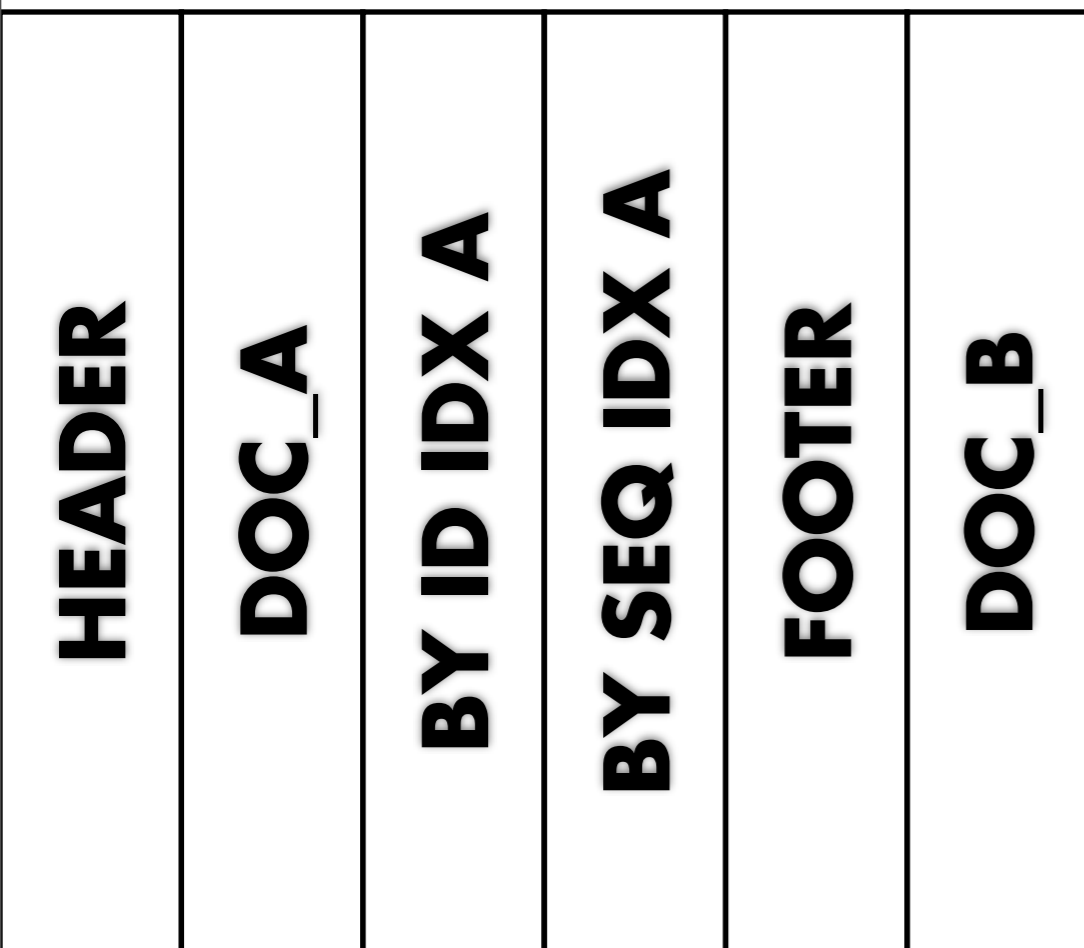
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



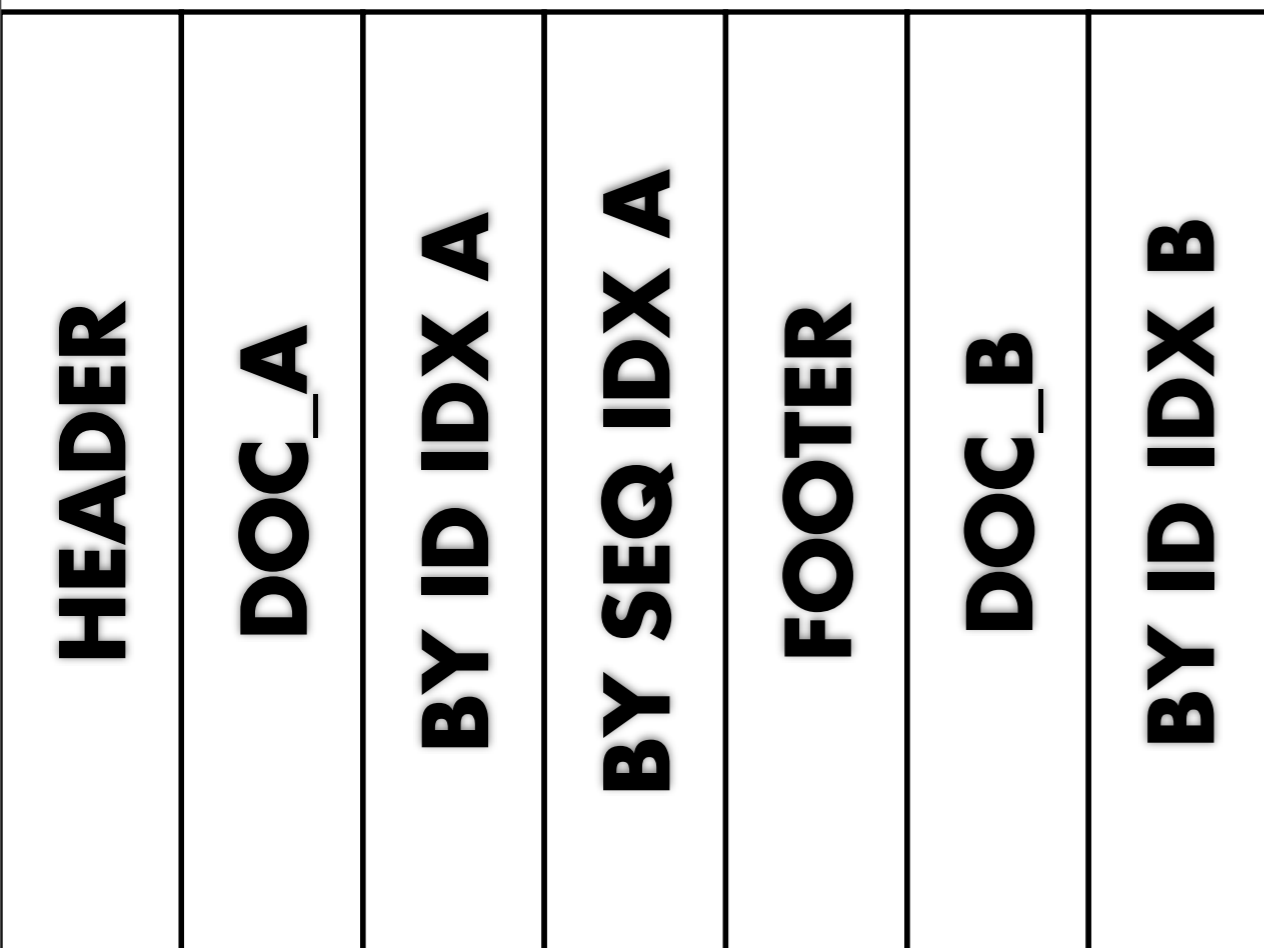
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



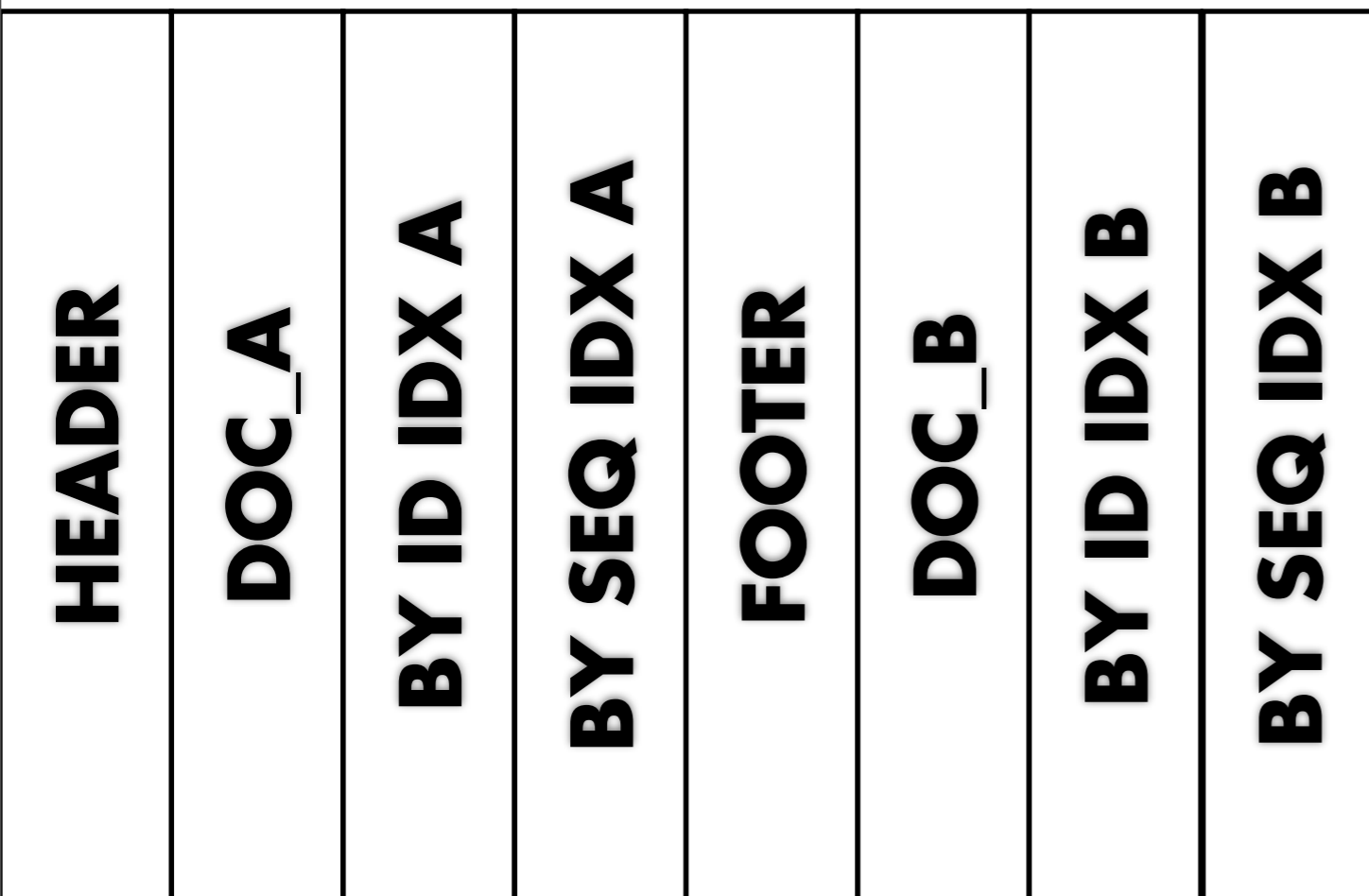
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



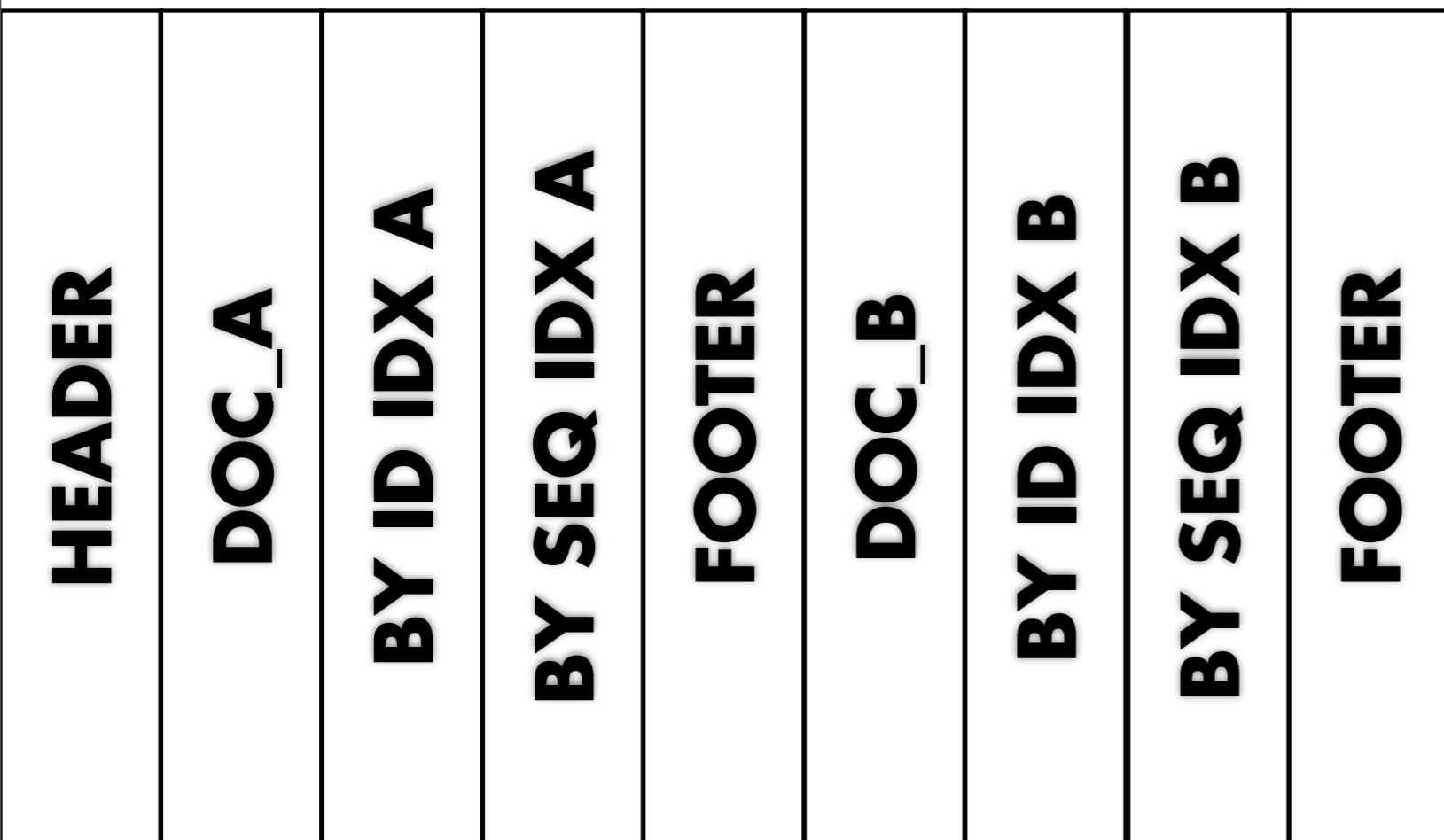
Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



Tuesday, June 4, 13

- 2 x b+tree & data interleaved
- append only, mvcc
- full fsync control

- Can answer:
 - Data for \$key
 - What happened \$since

- Used for core data storage
- As well as indexes

- Everything else is built on top



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B

BY ID IDX A

BY ID IDX B



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B

FOOTER



DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B

FOOTER

DEL DOC_A



DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B

FOOTER

DEL DOC_A

BY ID IDX A



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B

FOOTER

DEL DOC_A

BY ID IDX A

BY SEQ IDX A



Tuesday, June 4, 13

Bulk add + Delete

DOC_A

DOC_B

BY ID IDX A

BY ID IDX B

BY SEQ IDX A

BY SEQ IDX B

FOOTER

DEL DOC_A

BY ID IDX A

BY SEQ IDX A

FOOTER



Operational Consequences

Tuesday, June 4, 13

- efficient on spinning disk, “tape”
- btree = wide, upper layers in disk cache
- backup with `cp $a $b`
- crash safety/recovery

- compaction hurts

Efficient With Storage

Cache Friendly

Tuesday, June 4, 13

```
cp db.couch /mnt/backup
```


Safe

Tuesday, June 4, 13

Core Features (using by-req)

Tuesday, June 4, 13

- Replication
- Indexing / Views / GeoCouch / Lucene / ES etc.
- /_changes
- Compaction

Operational Consequences

Tuesday, June 4, 13

– compaction hurts

Tuesday, June 4, 13

Replication

DATABASE A

Tuesday, June 4, 13

Replication



Tuesday, June 4, 13
Replication

2
1
DATABASE A

Tuesday, June 4, 13
Replication

3
2
1
DATABASE A

Tuesday, June 4, 13
Replication

4
3
2
1
DATABASE A

4
3
2
1
DATABASE A

DATABASE B

4
3
2
1
DATABASE A

1
DATABASE B

Tuesday, June 4, 13
Replication

4
3
2
1
DATABASE A

2
1
DATABASE B

4
3
2
1
DATABASE A

3
2
1
DATABASE B

4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

5
4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

6
5
4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

7
6
5
4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

8
7
6
5
4
3
2
1
DATABASE A

4
3
2
1
DATABASE B

8
7
6
5
4
3
2
1
DATABASE A

5
4
3
2
1
DATABASE B

8
7
6
5
4
3
2
1
DATABASE A

6
5
4
3
2
1
DATABASE B

8
7
6
5
4
3
2
1
DATABASE A

7
6
5
4
3
2
1
DATABASE B

8
7
6
5
4
3
2
1
DATABASE A

8
7
6
5
4
3
2
1
DATABASE B

Tuesday, June 4, 13

Indexing

DATABASE A

Tuesday, June 4, 13

Indexing

1

DATABASE A

Tuesday, June 4, 13

Indexing

2
1
DATABASE A

Tuesday, June 4, 13

Indexing

3
2
1
DATABASE A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

INDEX A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

1
INDEX A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

2
1
INDEX A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

3
2
1
INDEX A

Tuesday, June 4, 13

Indexing

4
3
2
1
DATABASE A

4
3
2
1
INDEX A

Tuesday, June 4, 13

Indexing

5
4
3
2
1
DATABASE A

4
3
2
1
INDEX A

Tuesday, June 4, 13

Indexing

6
5
4
3
2
1
DATABASE A

4
3
2
1
INDEX A

Tuesday, June 4, 13

Indexing

7
6
5
4
3
2
1
DATABASE A

4
3
2
1
INDEX A

Tuesday, June 4, 13

Indexing

8
7
6
5
4
3
2
1
DATABASE A

4
3
2
1
INDEX A

8
7
6
5
4
3
2
1
DATABASE A

5
4
3
2
1
INDEX A

8
7
6
5
4
3
2
1
DATABASE A

6
5
4
3
2
1
INDEX A

8
7
6
5
4
3
2
1
DATABASE A

7
6
5
4
3
2
1
INDEX A

8
7
6
5
4
3
2
1
DATABASE A

8
7
6
5
4
3
2
1
INDEX A

Tuesday, June 4, 13

/_changes

DATABASE A

Tuesday, June 4, 13

/_changes

1

DATABASE A

Tuesday, June 4, 13

/_changes

2
1
DATABASE A

Tuesday, June 4, 13

/_changes

3
2
1
DATABASE A

Tuesday, June 4, 13
/_changes

4
3
2
1
DATABASE A

5
4
3
2
1
DATABASE A

Tuesday, June 4, 13
/_changes

6
5
4
3
2
1
DATABASE A

7
6
5
4
3
2
1
DATABASE A

8
7
6
5
4
3
2
1
DATABASE A

Tuesday, June 4, 13

Compaction

DATABASE A

Tuesday, June 4, 13

Compaction

1. DOC_A

DATABASE A

Tuesday, June 4, 13

Compaction

2. DOC_B

1. DOC_A

DATABASE A

Tuesday, June 4, 13

Compaction

3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

Tuesday, June 4, 13

Compaction

4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

Tuesday, June 4, 13

Compaction

5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

Tuesday, June 4, 13

Compaction

6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

Tuesday, June 4, 13

Compaction

7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

Tuesday, June 4, 13

Compaction

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

COMPACT A

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

3. DOC_C
COMPACT A

Tuesday, June 4, 13

Compaction

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

4. DOC_A
3. DOC_C
COMPACT A

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

5. DOC_D
4. DOC_A
3. DOC_C
COMPACT A

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
COMPACT A

Tuesday, June 4, 13

Compaction

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
COMPACT A

Tuesday, June 4, 13

Compaction

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
2. DOC_B
1. DOC_A
DATABASE A

8. DOC_G
7. DOC_F
6. DOC_B
5. DOC_D
4. DOC_A
3. DOC_C
COMPACT A

The Revision Tree

Tuesday, June 4, 13

Tuesday, June 4, 13

The Happy Path

DATABASE A

Tuesday, June 4, 13

The Happy Path

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

The Happy Path

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

The Happy Path

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

The Happy Path

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

The Happy Path

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DOC A [3-C,2-B,1-A]

DATABASE B

Tuesday, June 4, 13

The Happy Path

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DATABASE B

Tuesday, June 4, 13

The Happy Path

DOC A [4-D,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DATABASE A

Tuesday, June 4, 13

Conflicts!

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

Conflicts!

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

Conflicts!

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

Tuesday, June 4, 13

Conflicts!

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DOC A [3-C,2-B,1-A]

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [4-K,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [[4-K,3-C,2-B,1-A], [4-D, 3-C,2-B,1-A]]

DOC A [4-K,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [[4-K,4-D], 3-C,2-B,1-A]

DOC A [[4-K,3-C,2-B,1-A], [4-D, 3-C,2-B,1-A]]

DOC A [4-K,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [4-K,4-D,3-C,2-B,1-A]

DOC A [[4-K,4-D], 3-C,2-B,1-A]

DOC A [[4-K,3-C,2-B,1-A], [4-D, 3-C,2-B,1-A]]

DOC A [4-K,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [4-D,4-K,3-C,2-B,1-A]

DOC A [4-K,4-D,3-C,2-B,1-A]

DOC A [[4-K,4-D], 3-C,2-B,1-A]

DOC A [[4-K,3-C,2-B,1-A], [4-D, 3-C,2-B,1-A]]

DOC A [4-K,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

DOC A [5-L,4-D,4-K,3-C,2-B,1-A]

DOC A [4-D,4-K,3-C,2-B,1-A]

DOC A [4-K,4-D,3-C,2-B,1-A]

DOC A [[4-K,4-D], 3-C,2-B,1-A]

DOC A [[4-K,3-C,2-B,1-A], [4-D, 3-C,2-B,1-A]]

DOC A [4-K,3-C,2-B,1-A]

DOC A [4-D,3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [3-C,2-B,1-A]

DOC A [2-B,1-A]

DOC A [1-A]

DATABASE A

DATABASE B

Tuesday, June 4, 13

Conflicts!

Erlang

Tuesday, June 4, 13

- Small codebase
- Efficient in small teams

- Isolated processes
- Supervision tree
- Concurrency

- Portable runtime

- Hard to recruit for
- Steep ramp-on
- Bit of an operational black box (nine nines story)

Potential Improvements

Tuesday, June 4, 13

- Smarter compactor
- Smarter file-storage
- Less custom HTTP handling
- More indexers

The

End

Thanks!

Tuesday, June 4, 13