



Bringing Riak to the Mobile Platform

Kresten Krab Thorup
Hacker

@drkrab





Bringing Riak to the ~~Mobile~~ Platform Client

**Kresten Krab Thorup
Hacker**

@drkrab



About the Speaker

- **Language Geek** Emacs/TeX Hacker, Objective C, NeXT, GNU Compiled Java, Java Generics, Ph.D., Erlang/Erjang
- **Developer** J2EE AppServer, CORBA/RMI, XA-TM
- **Trifork CTO** QCon & GOTO Conferences
Technology Adoption; Riak; Erlang

Outline

- **Riak** and it's Data Model
- **Protocol** for Key/Value synchronization
- **BucketDB**, a Riak “client peer”

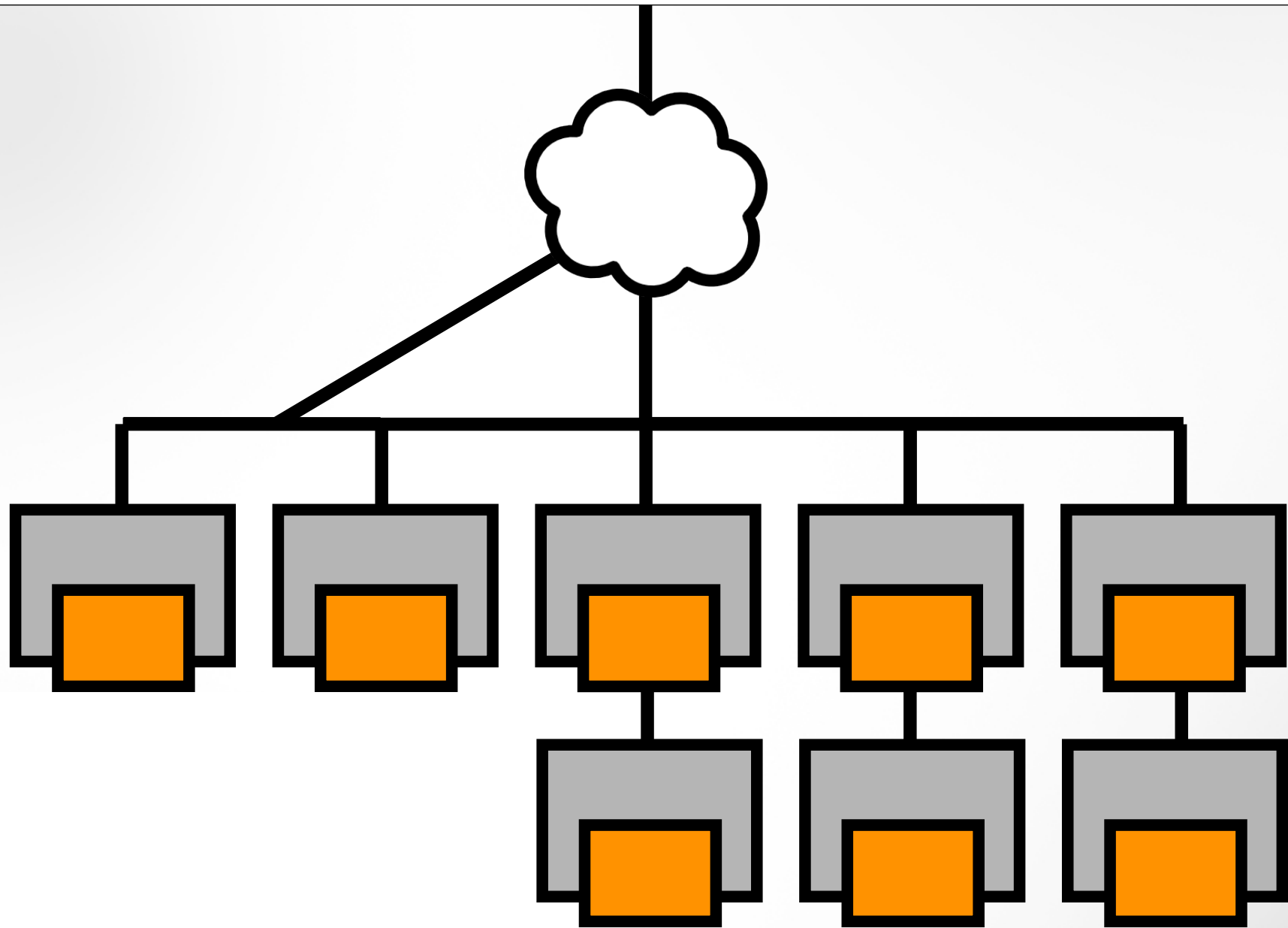
@drkrab

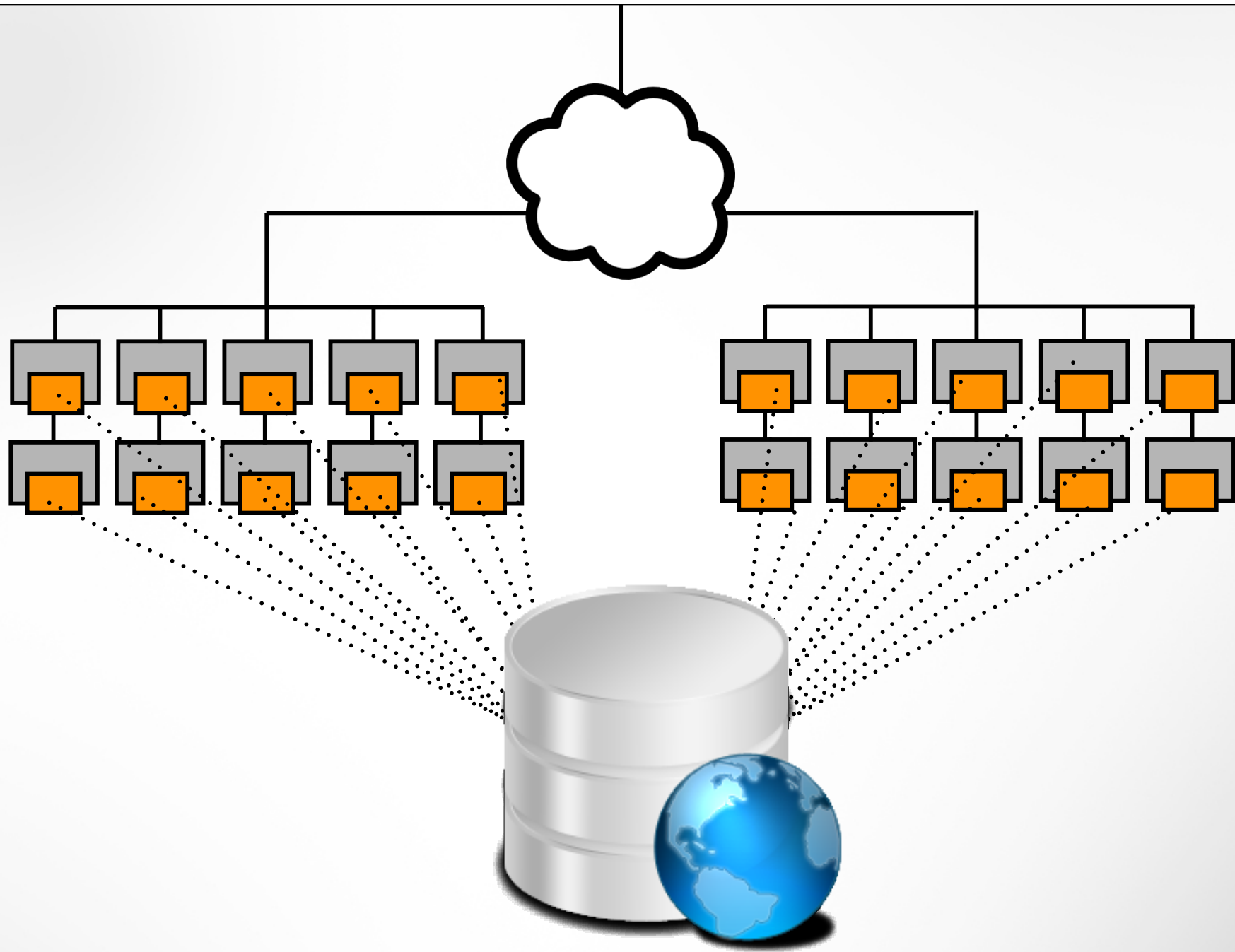


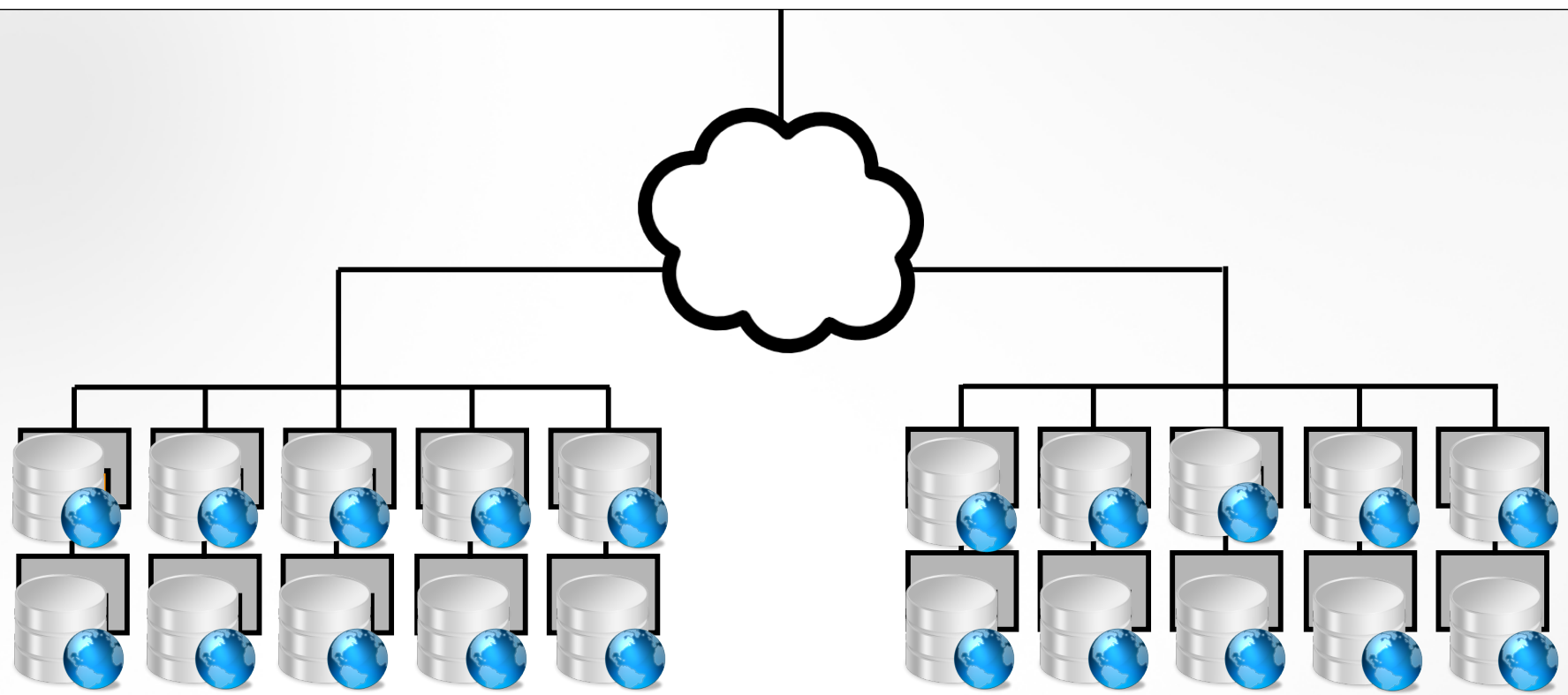
What is Riak, anyway?

[@drkrab](#)

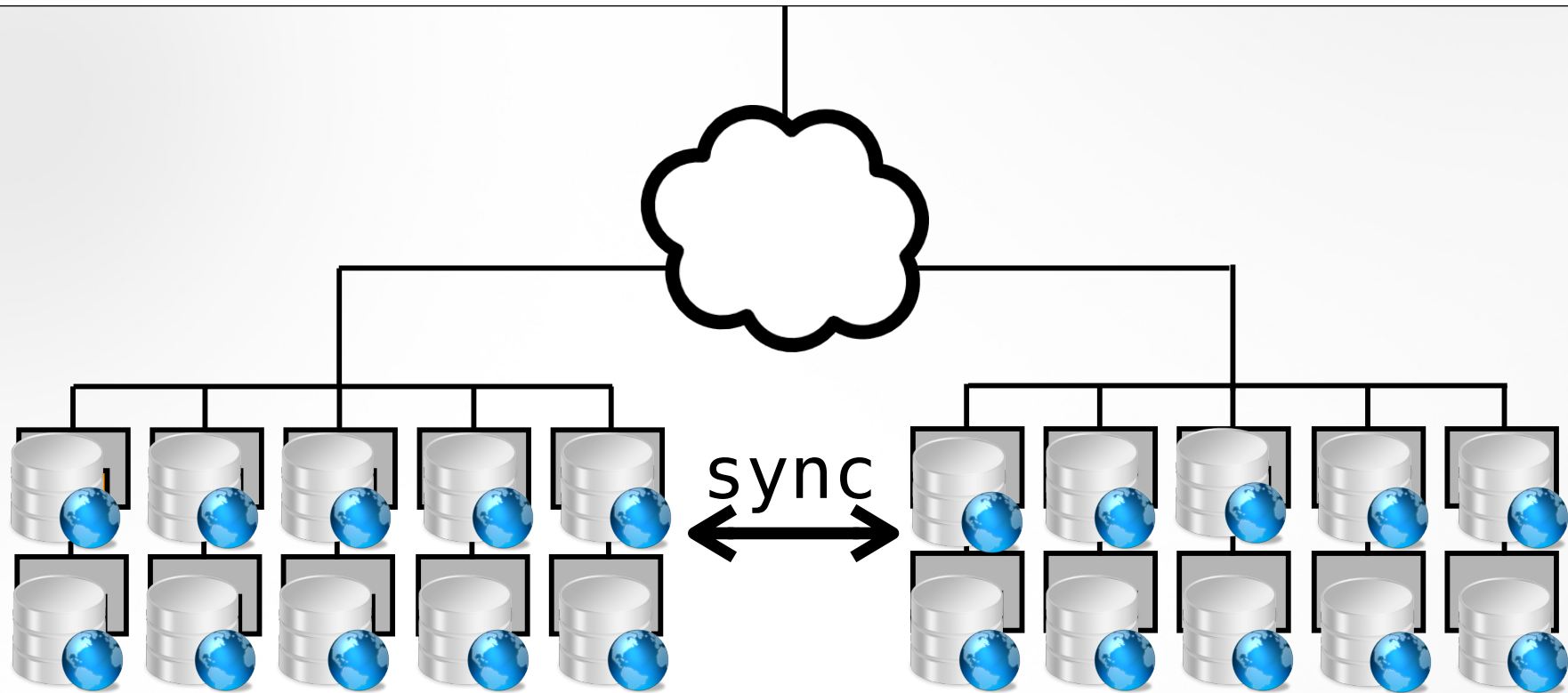


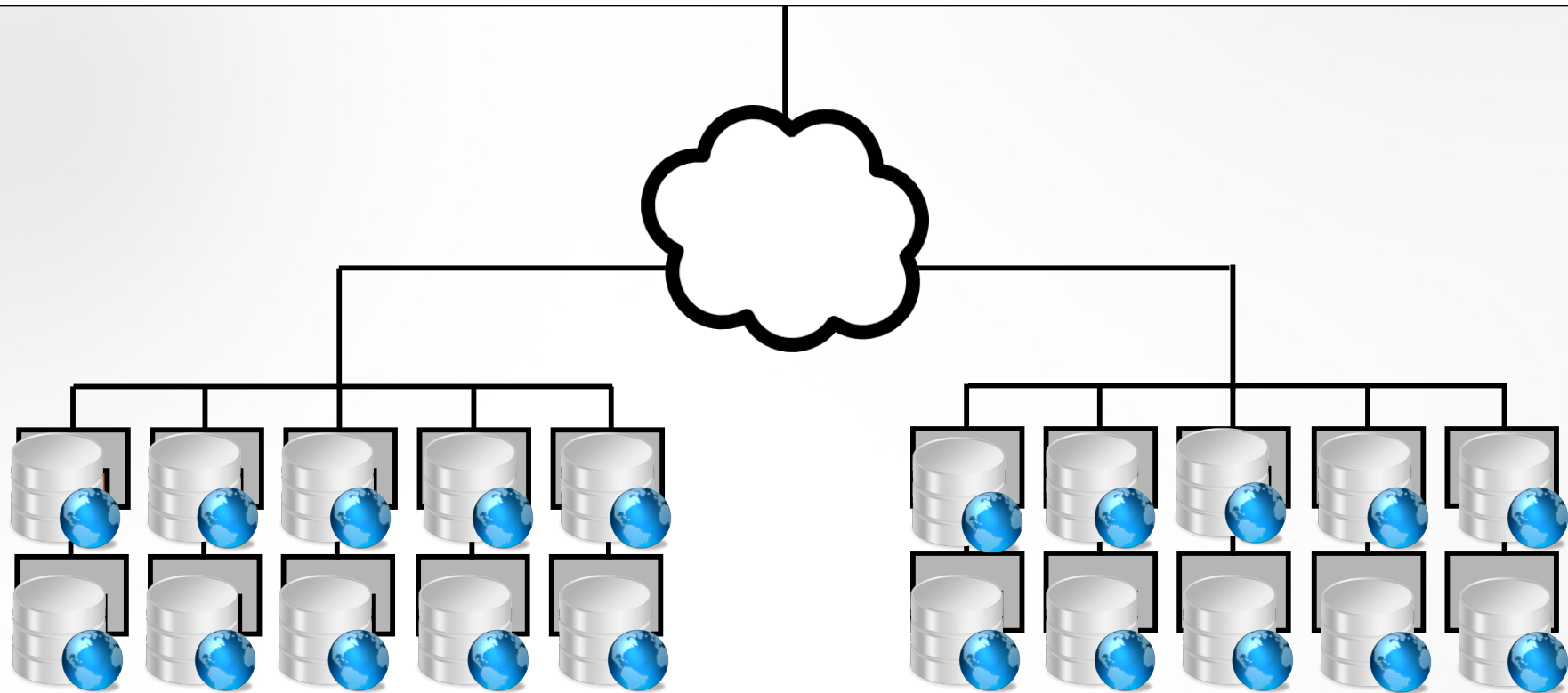






↖ ↗
coordinate





 **riak**

- scalable and available
- system captures write-write conflicts

Common Medicine Card



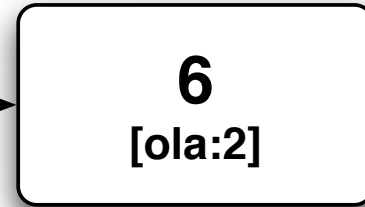
foo

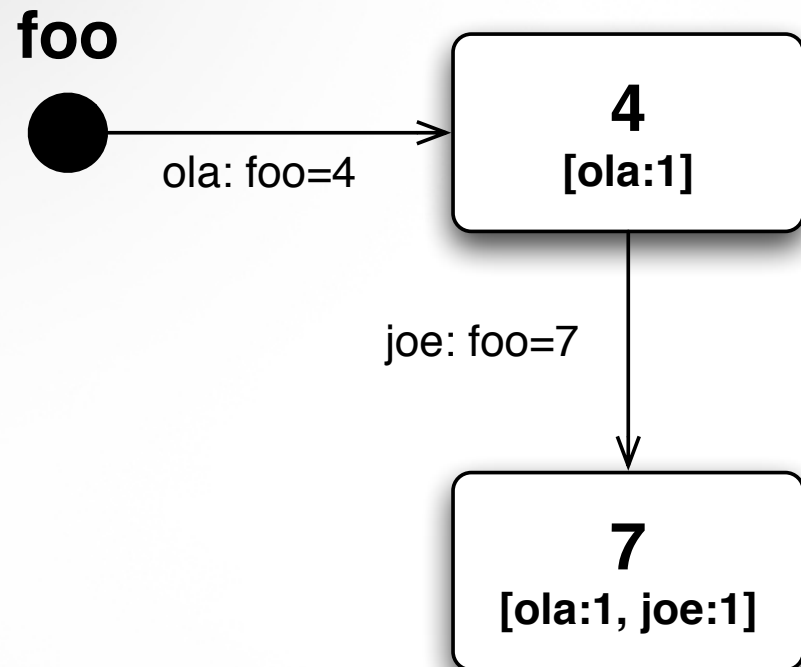


ola: foo=4



ola: foo=6





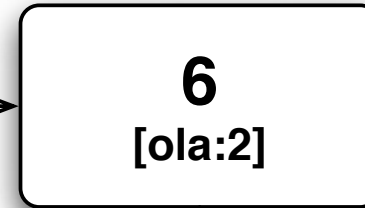
foo



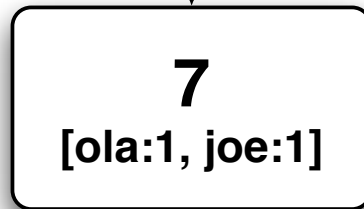
ola: foo=4



ola: foo=6

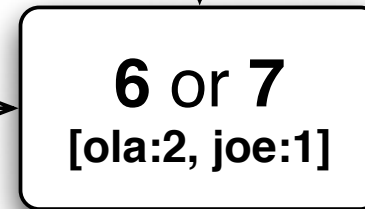


joe: foo=7



merge

merge



Riak API review

connect(Server) → Client

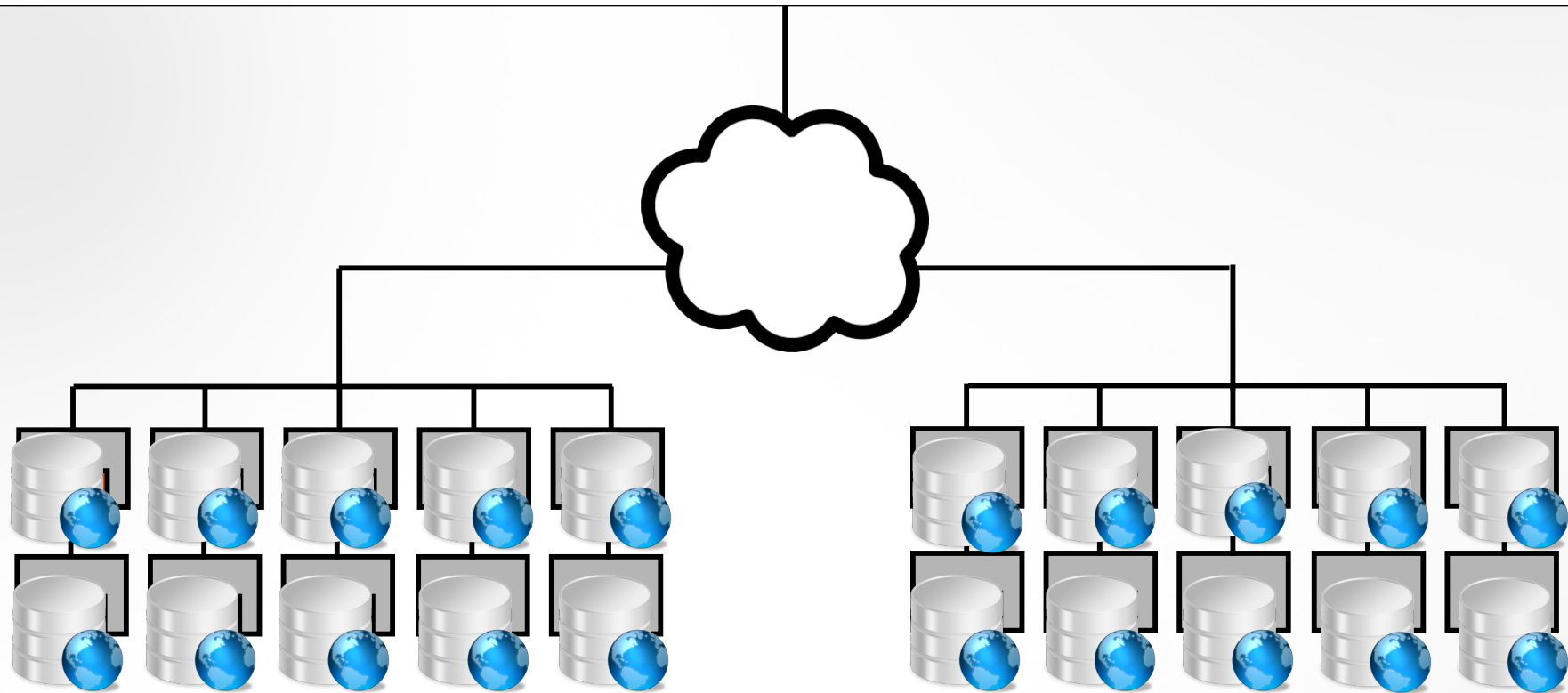
```
BKey  :: {Bucket, Key}
Datum :: {ContentType, RawData}
```

Client:**insert**(**BKey**, **Datum**) → {ok, VClock} | ...

Client:**lookup**(**BKey**) → {ok, VClock, [**Datum**] } | ...

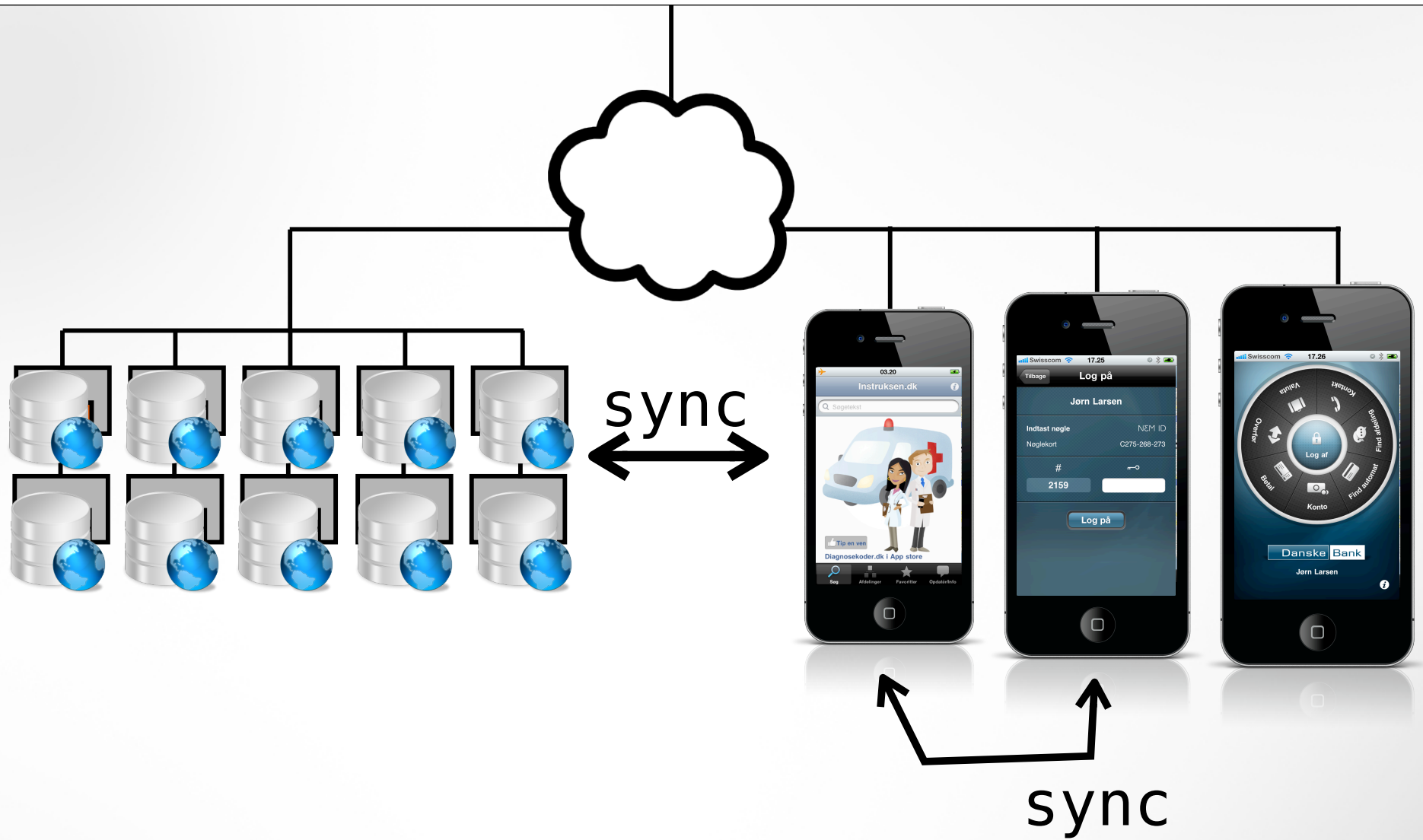
Client:**update**(**BKey**, VClock, **Datum**) → ok | ...

Client:**delete**(**BKey**, VClock) → ok | ...



 **riak**

- system captures write-write conflicts
- resolve lazily (read repair)



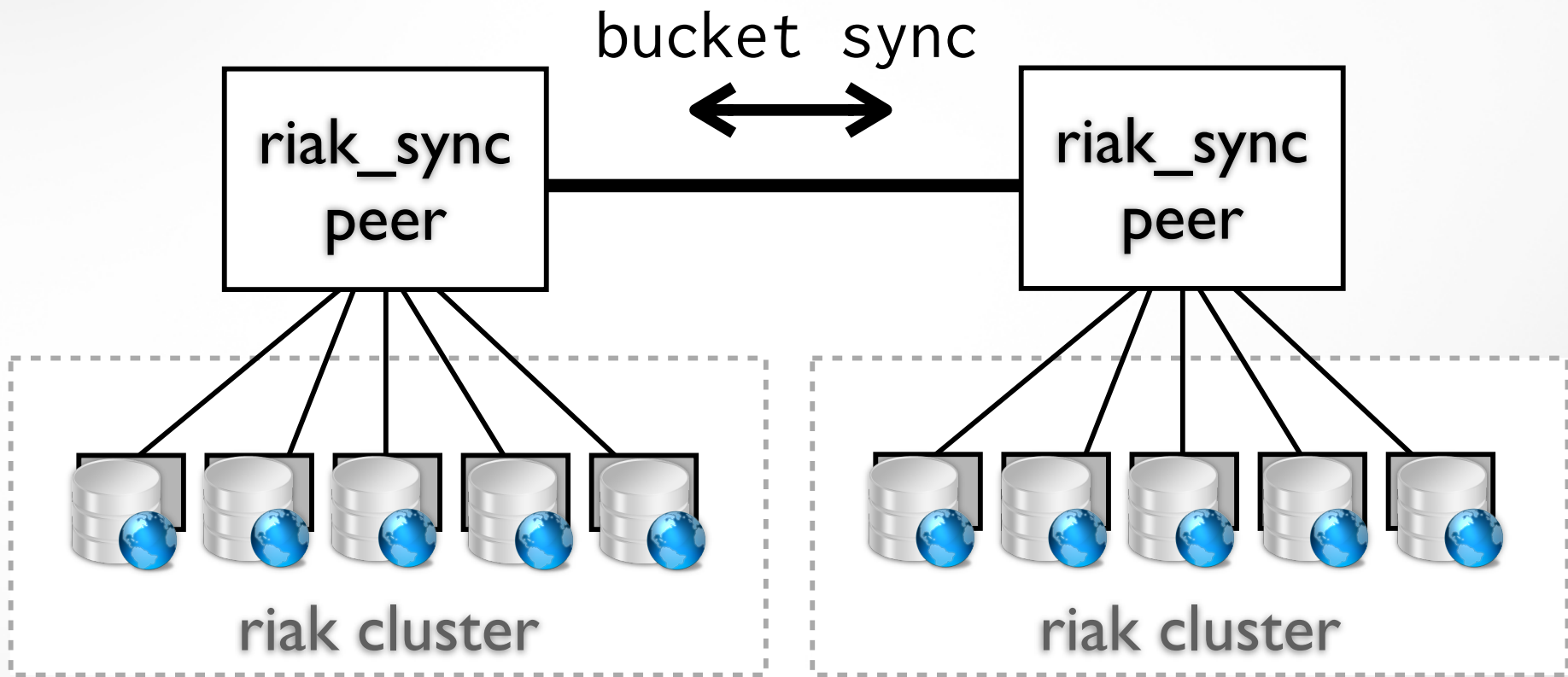
Outline

- ~~Riak and its Data Model~~
- **Protocol** for Key/Value synchronization
- **BucketDB**, a Riak “client peer”

@drkrab



Not to be confused with riak_rep1,
part of the commercial Riak EDS



```
%% “client” peer
```

```
> LocalStore = riak:client_connect('riak@127.0.0.1')
```

```
> riak_sync:sync(LocalStore, <<“bucket”>>,  
                  “172.1.35.204”, 8082)
```

```
%% Other site has riak_sync running...
```

```
(riak@172.1.35.204)1> riak_sync:start()
```

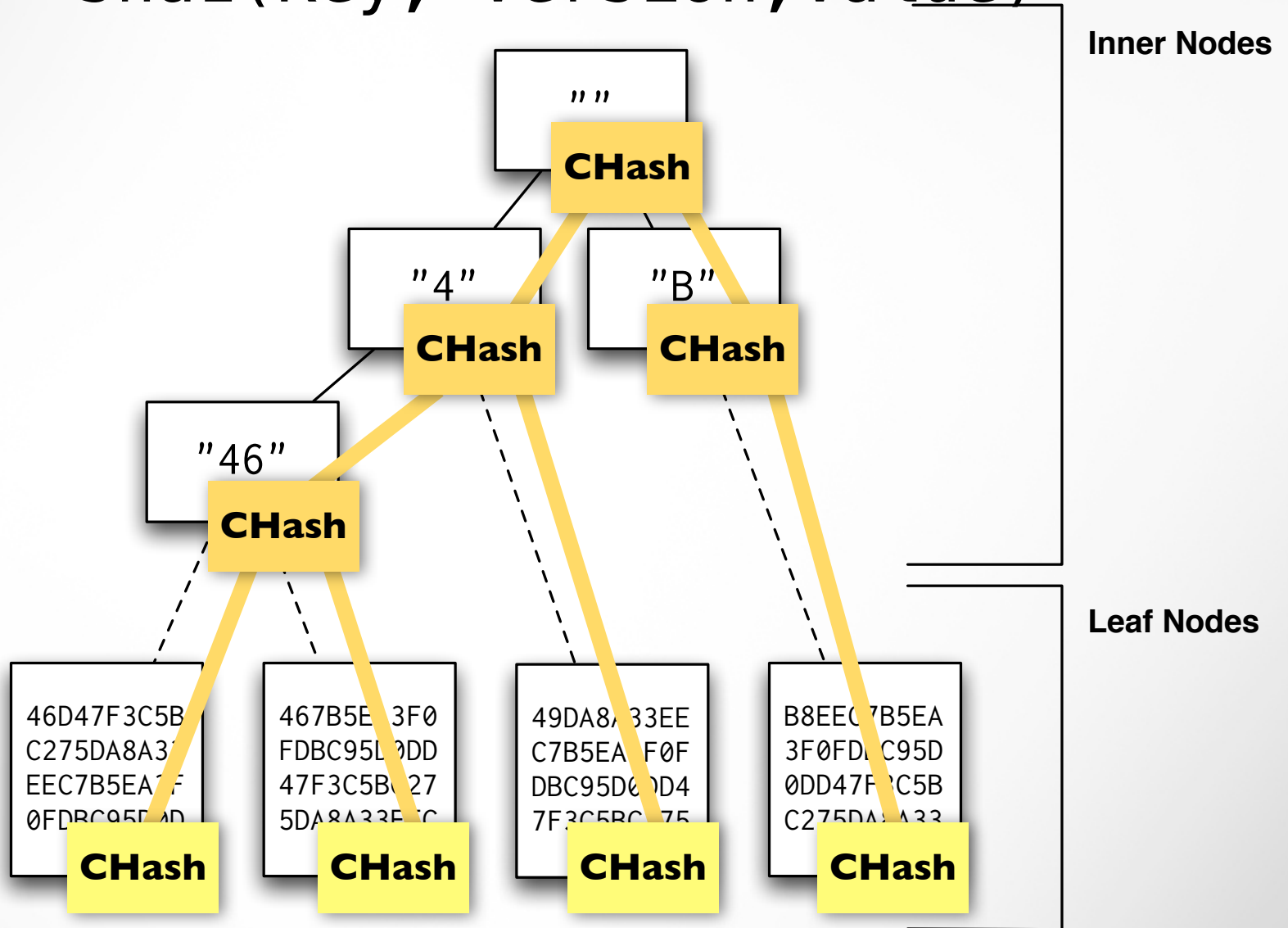
```
{riak_sync, [{pb_ip, “172.1.35.204”},  
             {pb_port, 8082},  
             {riak, ‘dev@127.0.0.1’}]]}
```

Riak Sync Protocol

- Riak bucket replication
- Asymmetric work load
- Designed for high-latency networks

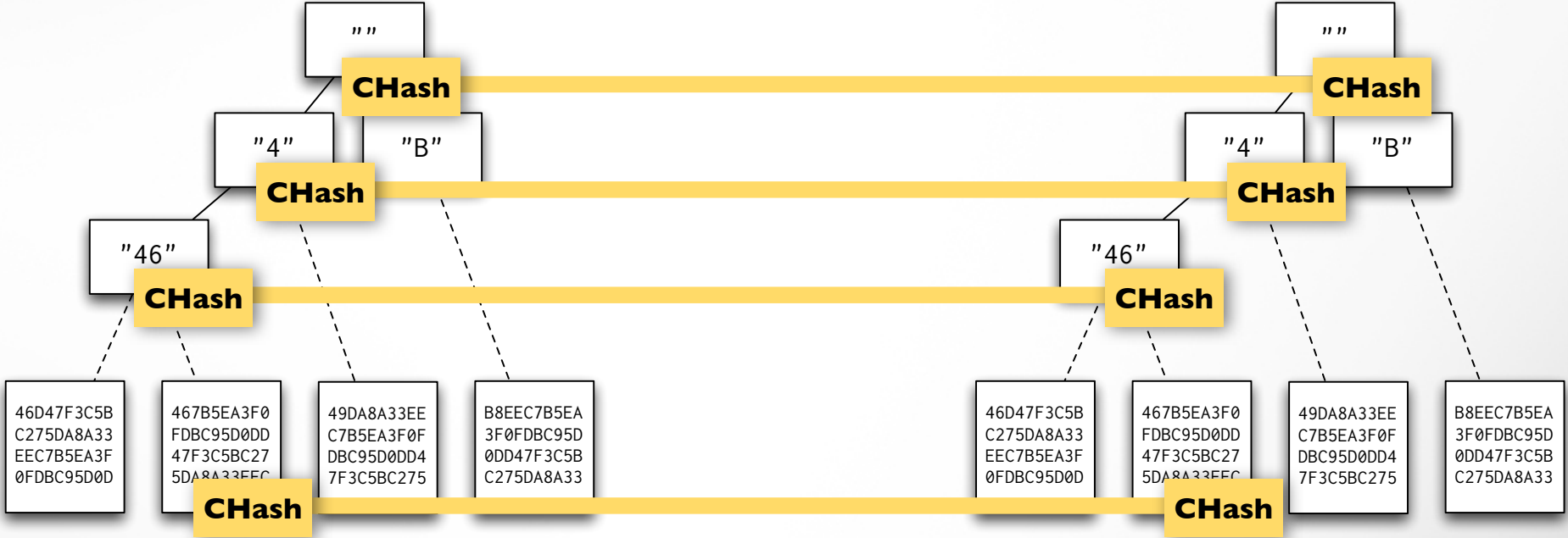
StorageKey = sha1(Key)

CHash = sha1(Key, Version, Value)



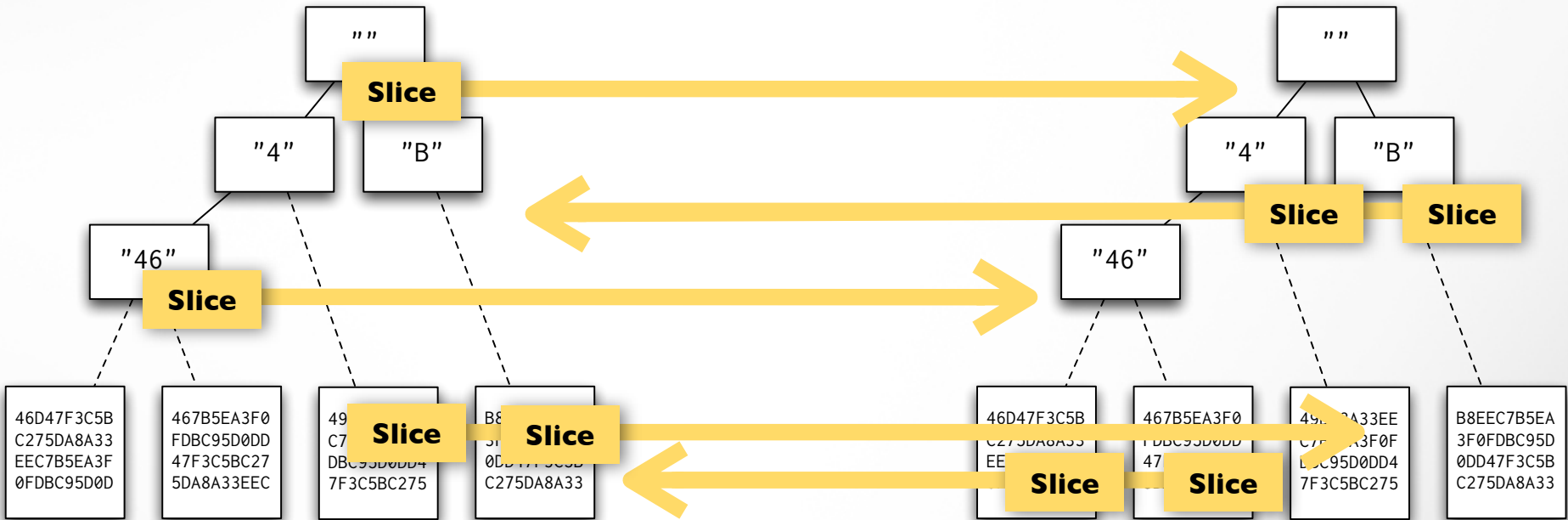
riksync peer

riksync peer



riksync peer

riksync peer



riaksync peer

riaksync peer

You need to restart your computer. Hold down the Power button for several seconds or press the Restart button.

Veillez redémarrer votre ordinateur. Maintenez la touche de démarrage enfoncée pendant plusieurs secondes ou bien appuyez sur le bouton de réinitialisation.

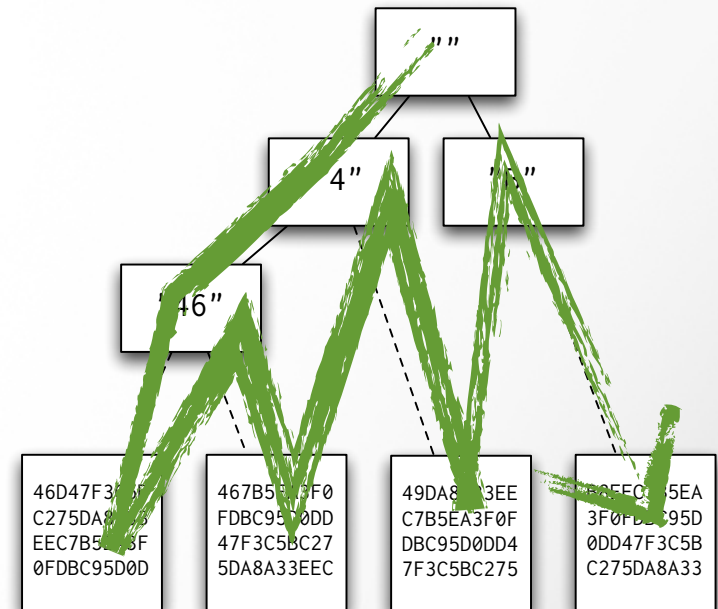
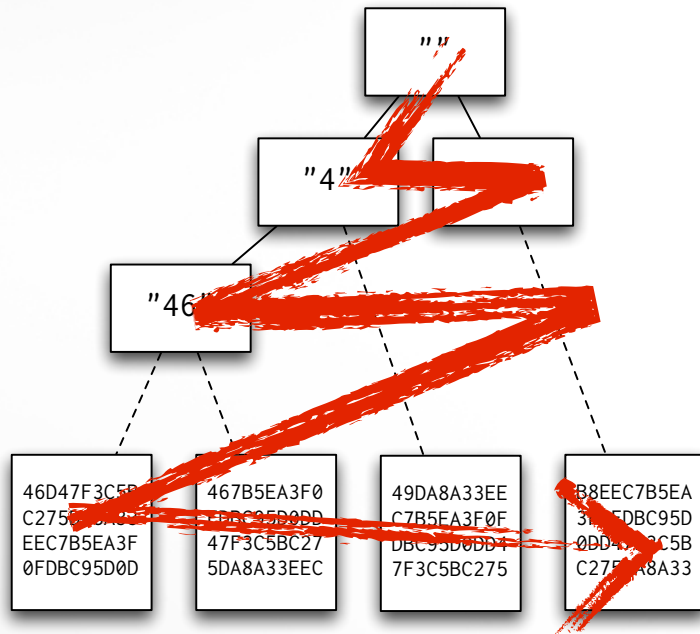
Sie müssen Ihren Computer neu starten. Halten Sie dazu die Einschalttaste einige Sekunden gedrückt oder drücken Sie die Neustart-Taste.

コンピュータを再起動する必要があります。パワーボタンを数秒間押し続けるか、リセットボタンを押してください。



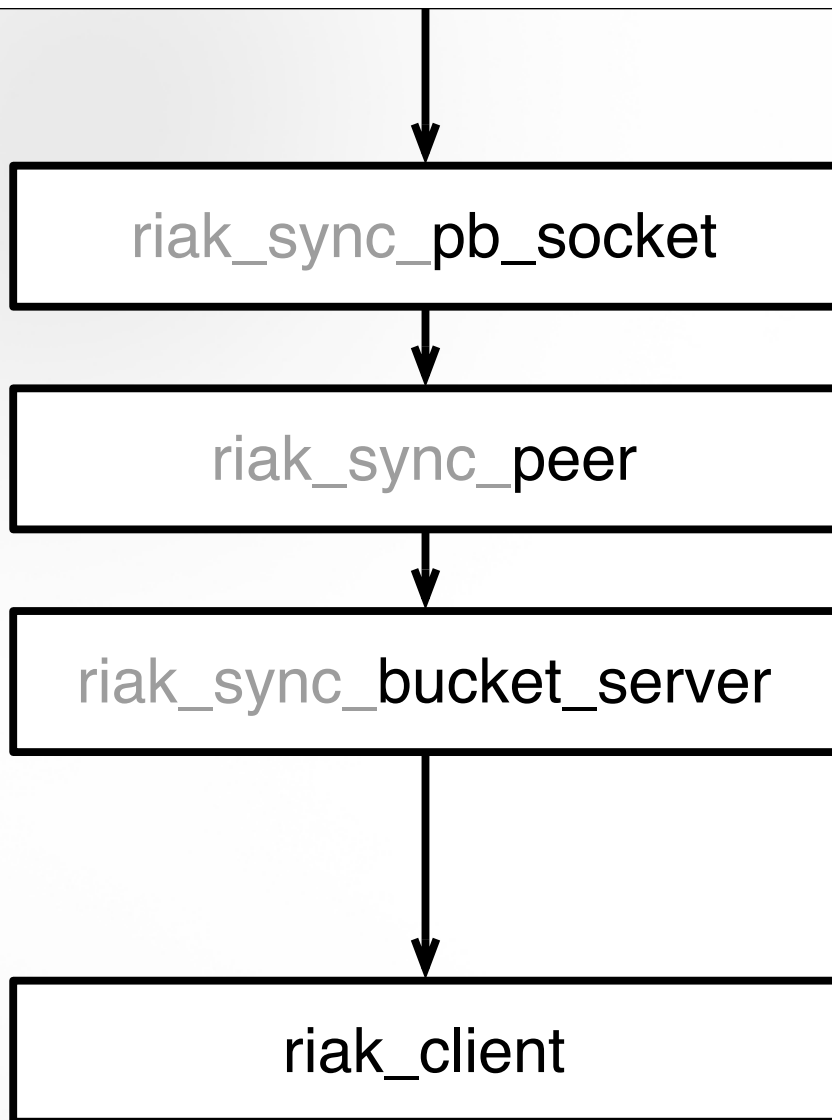
riksync peer

riksync peer



Sync Protocol

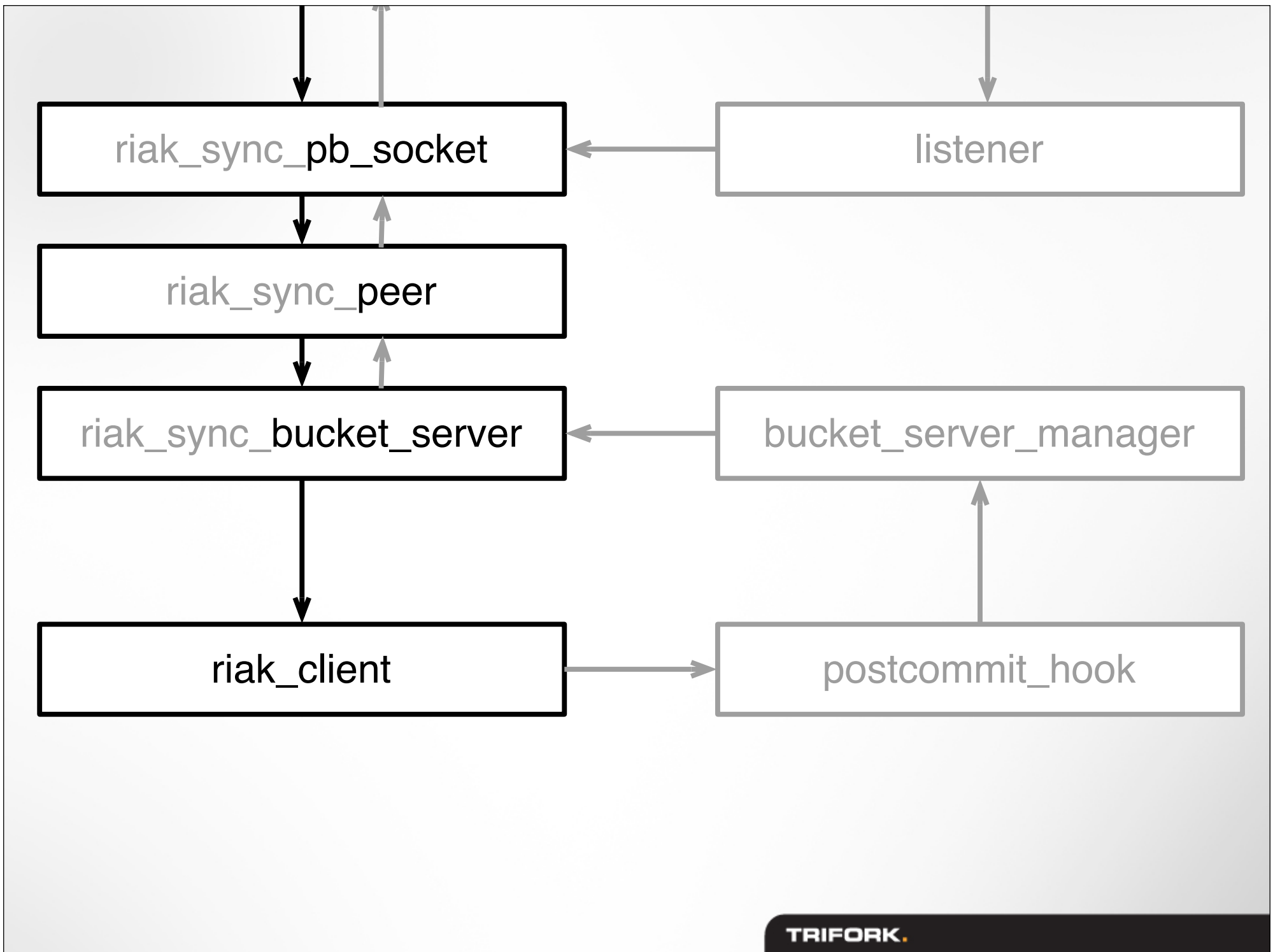
- Parallelizes quickly (client does breadth first)
- Many small messages (will benefit from Nagle's alg)
- Server peer reorders requests to reduce session/buffer state
- Client needs (almost) no session state



wire protocol handler

session management

merkle_tree server



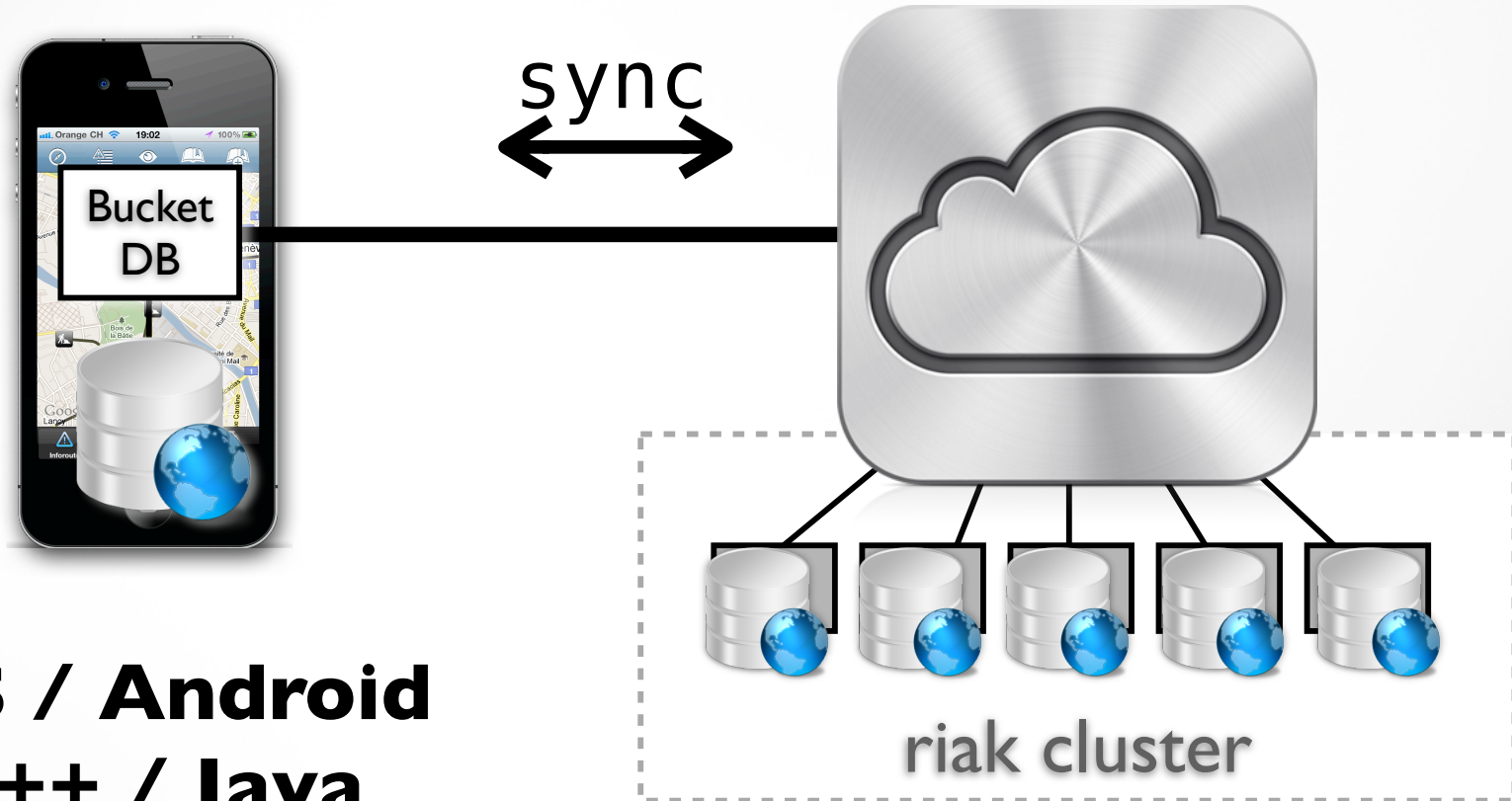
Outline

- ~~Riak and its Data Model~~
- ~~Protocol for Key/Value synchronization~~
- **BucketDB**, a Riak “client peer”

@drkrab

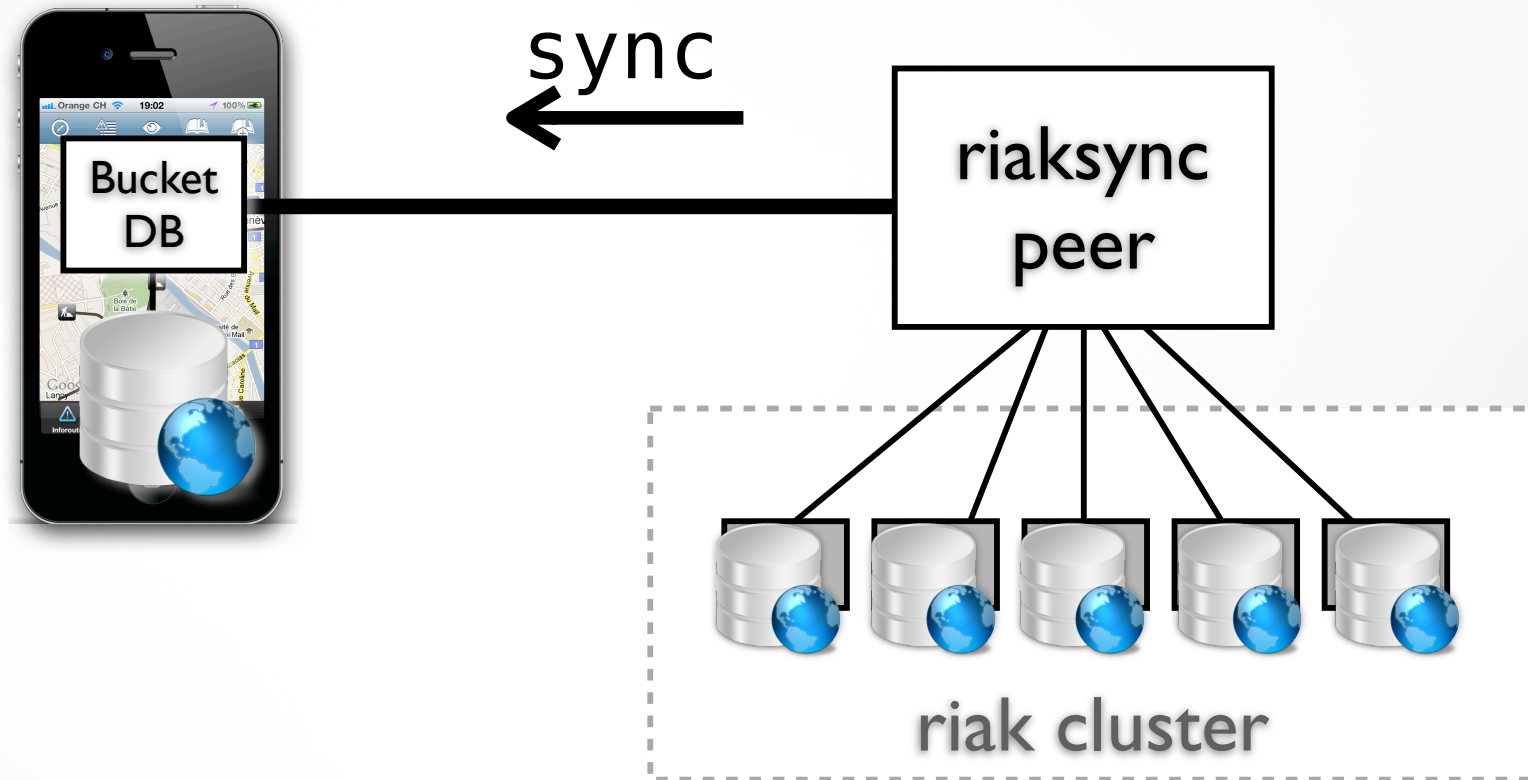


Client Riak: BucketDB

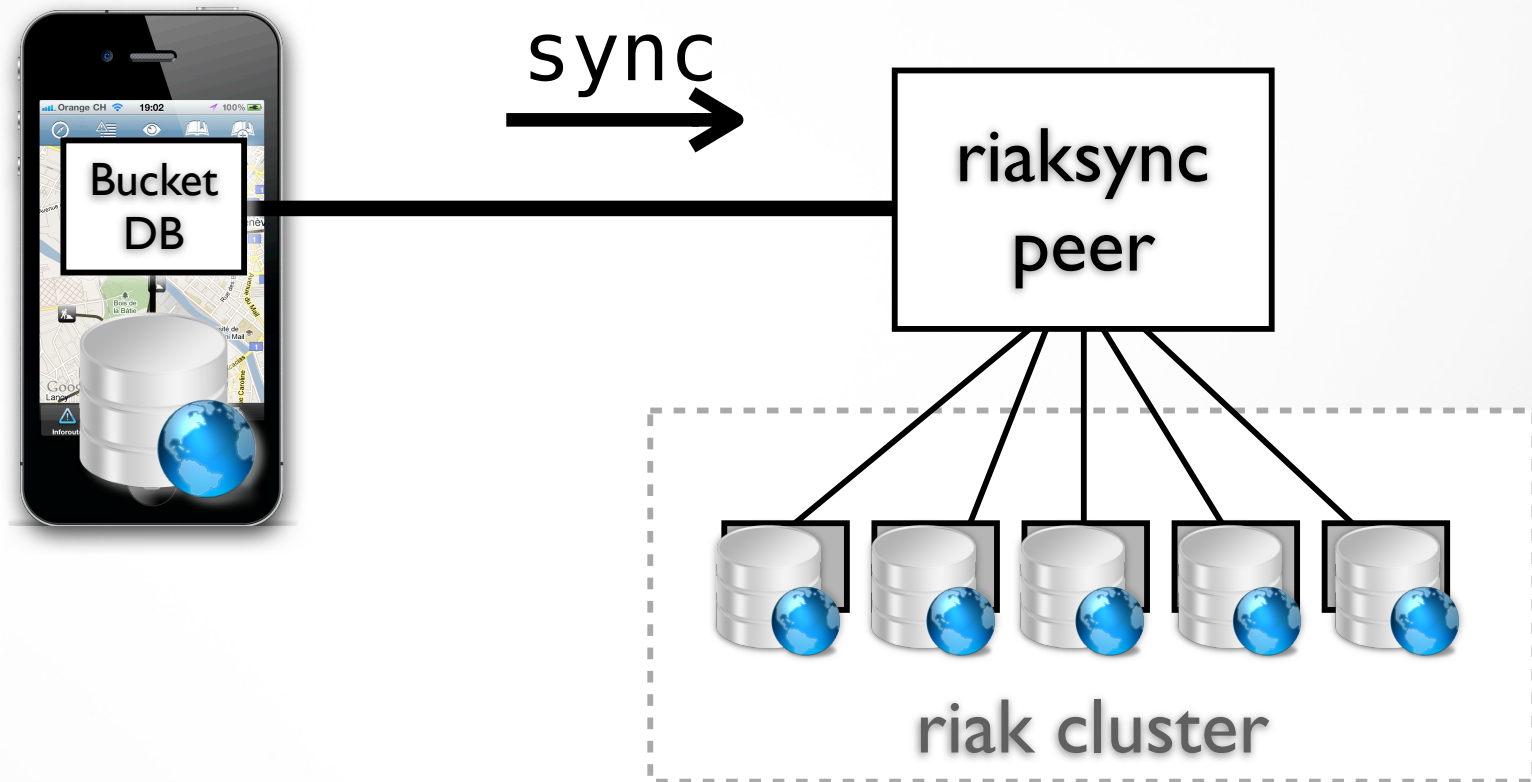


iOS / Android
C++ / Java
HTML / Desktop

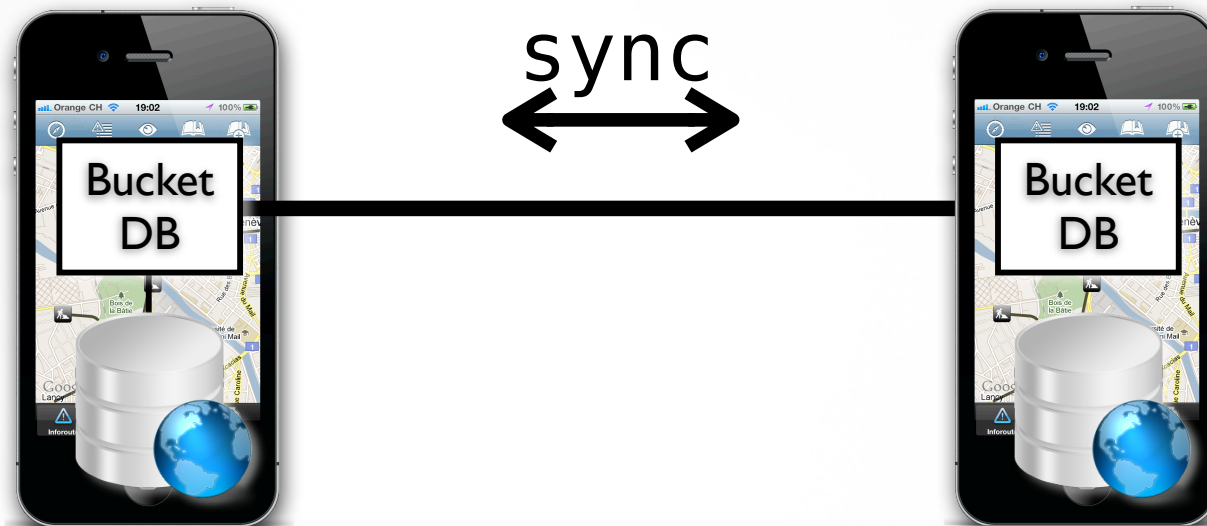
Content Distribution



Reliable Queueing



Peer Sync

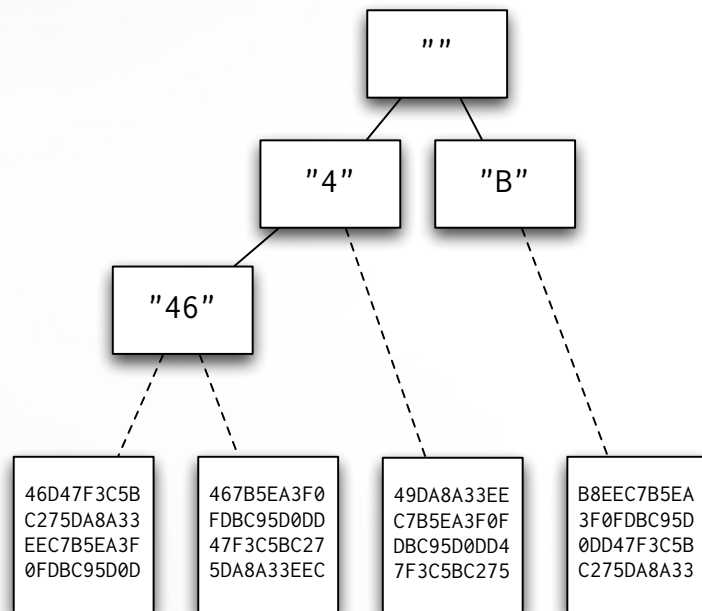


BucketDB

riak_sync peer



BucketDB



- Client stores data in local data store as a HASH TREE
- Always “ready to sync” (no session state, no cpu/memory intensive computation)

DATA

ROW_ID	KEY	CONTENT TYPE	DATA

TRIGGER



MERKLE_LEAF

HKEY	VCLOCK	CHASH	KEY

MERKLE_INNER

PATH	CHASH	CHILDREN

Other Items to Discuss

- Security models
 - Provide own “socket factory” / credentials
 - Document- or Record-oriented
- Handling Deleted data
- Future: Index/Search, local M/R

Summary

- **Riak** and it's Data Model
- **Protocol** for Key/Value synchronization
- **BucketDB**, a Riak “client peer”

@drkrab



Thank You

[@drkrab](#)



Thank You

[@drkrab](#)

