goto; conference aarhus

RIAK ON DRUGS (AND THE OTHER WAY AROUND)

Kresten Krab Thorup
CTO, Trifork

INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

gotocon.com

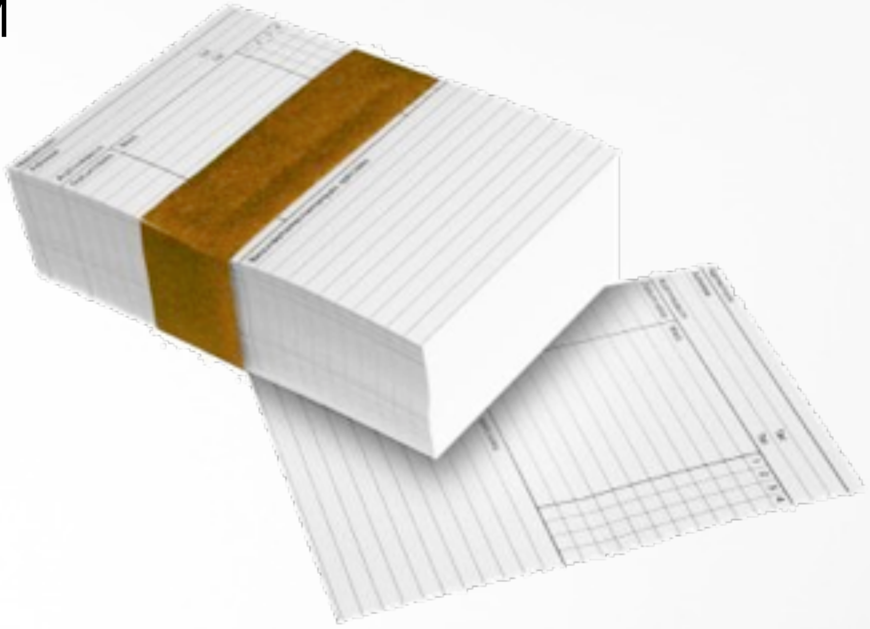
About the Speaker

- **Language Geek** Emacs/TeX Hacker, Objective C, NeXT, GNU Compiled Java, Java Generics, Ph.D.
- **Developer** J2EE AppServer, CORBA/RMI, XA-TM, Java Firefighter
- **Trifork CTO** Conference “Editor”, Technology Adoption

In this talk...

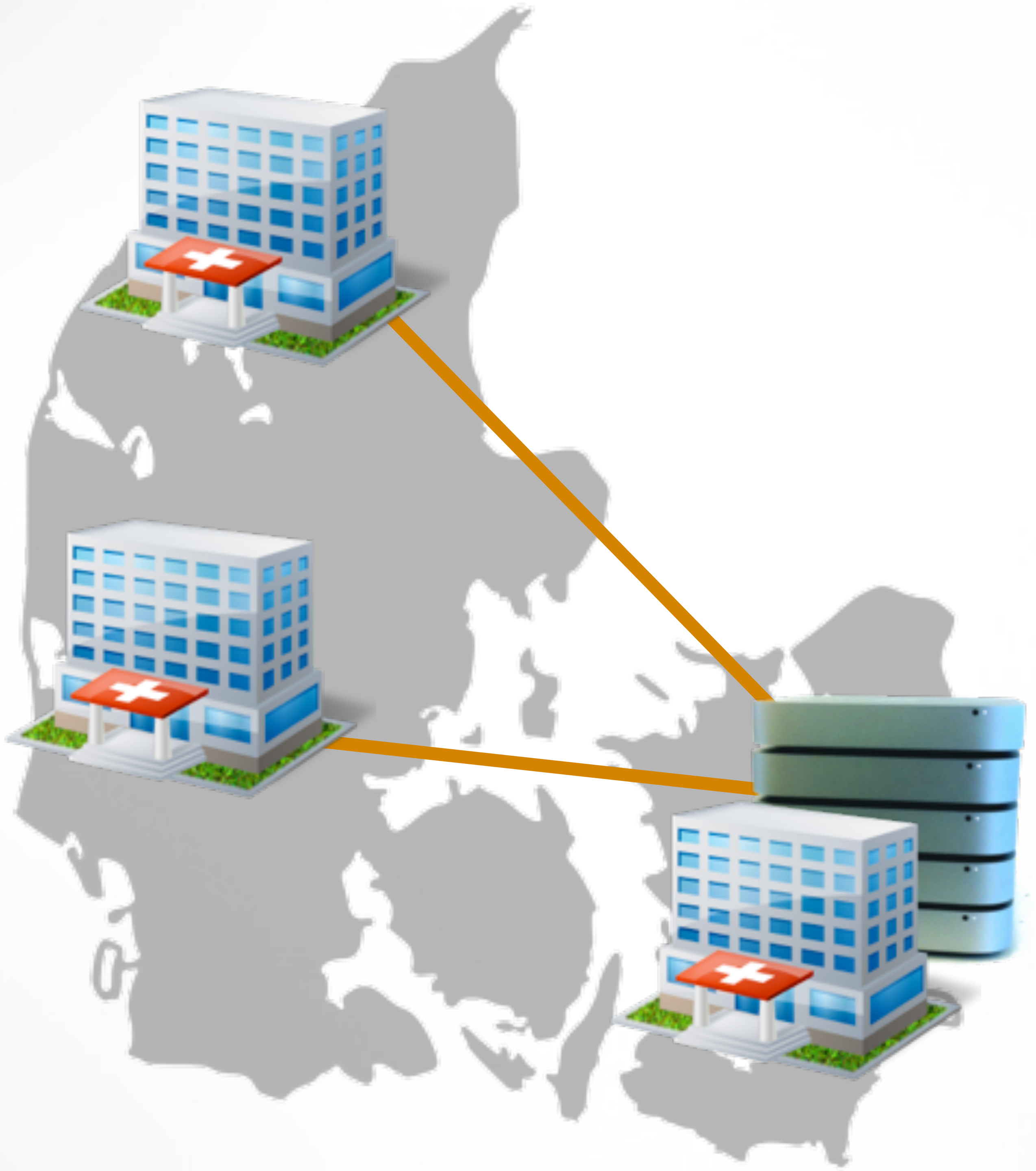
- About Common Medicine Card
- Building a Decentralized Architecture
- 4 different "shapes of data" and how to map this to a Key/Value store
- Hints, tips and tricks for Riak along the way

A Medicine Card



- For a person
- List of current drug treatments
- With prescriptions and related events

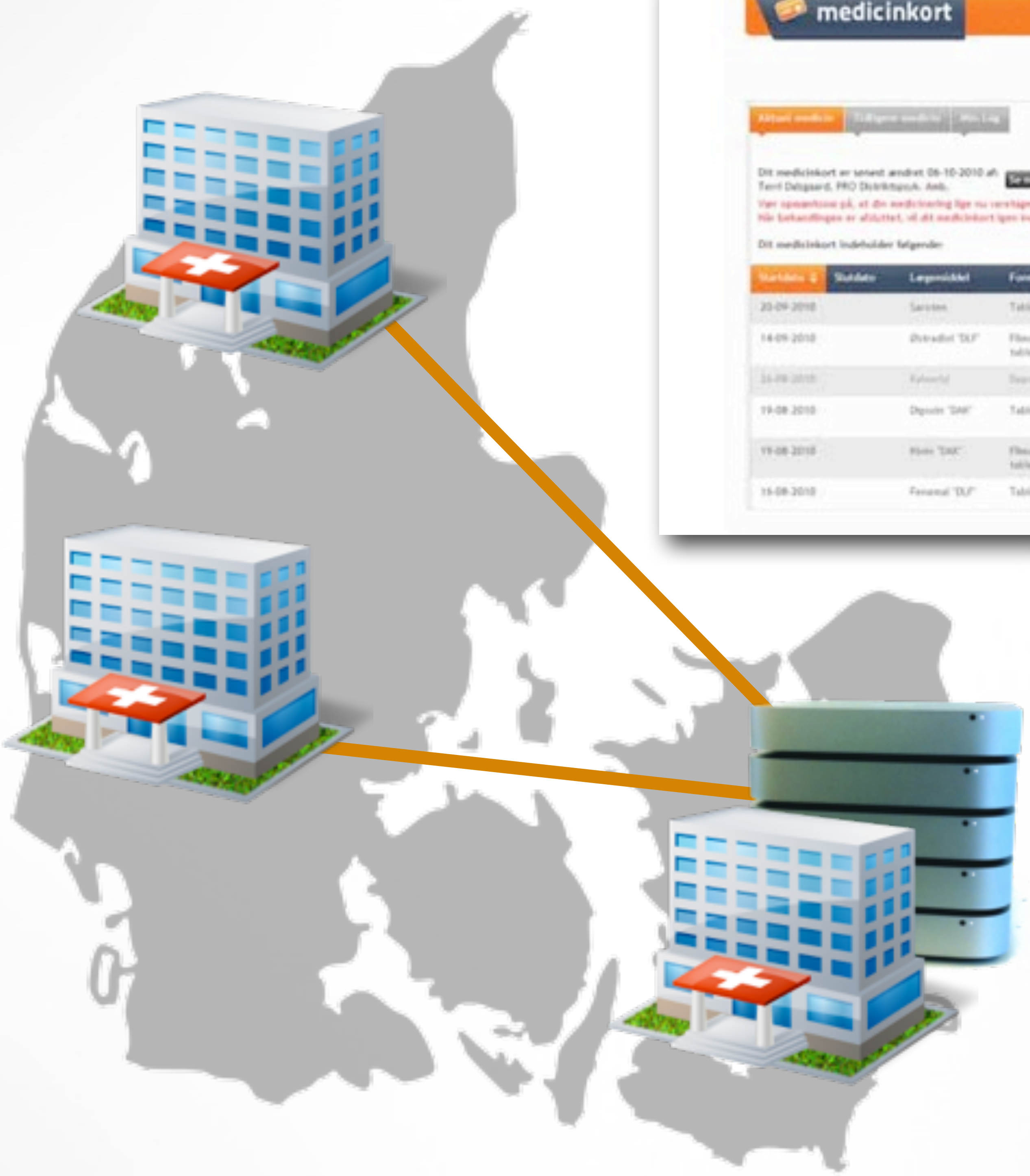
Common Medicine Card

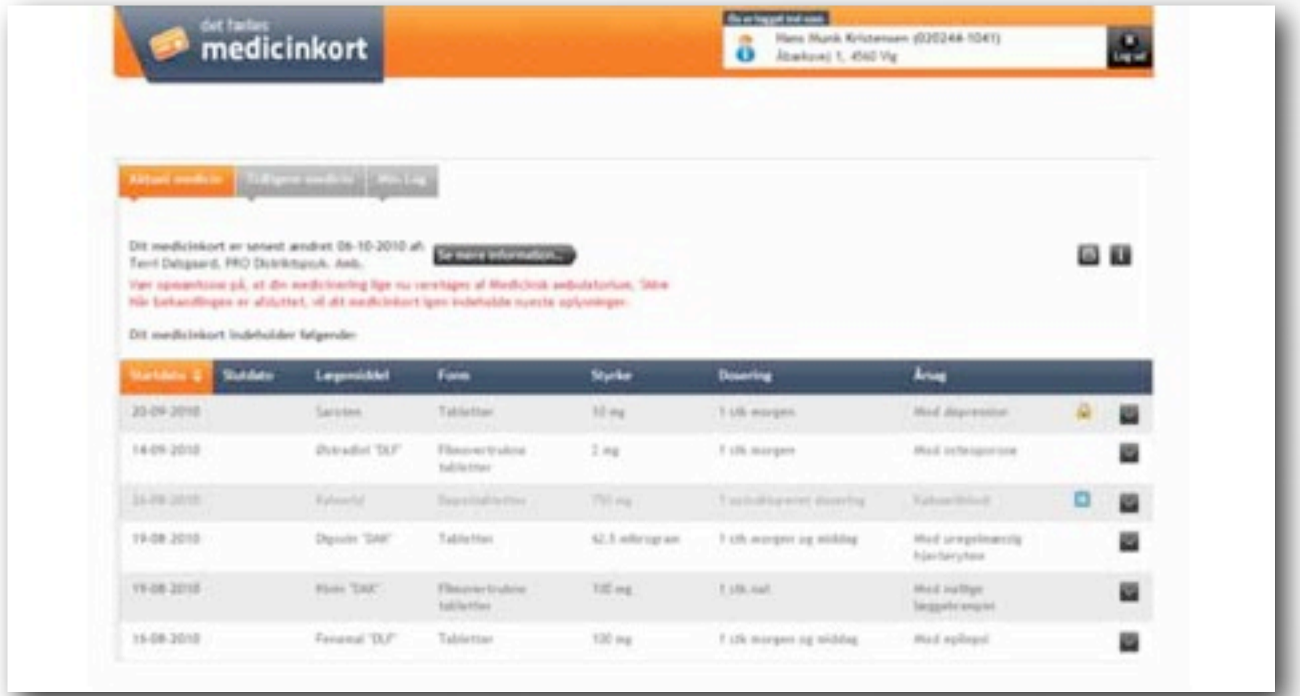


TRIFORK.

Tuesday, October 11, 11

Common Medicine Card





Hans Munk Kristensen (020244-1041)
Åbækvej 1, 4560 Vig

Dit medicinkort er senest ændret 06-10-2010 af:
Terri Dalsgaard, PRO Distriktspsyk. Amb.

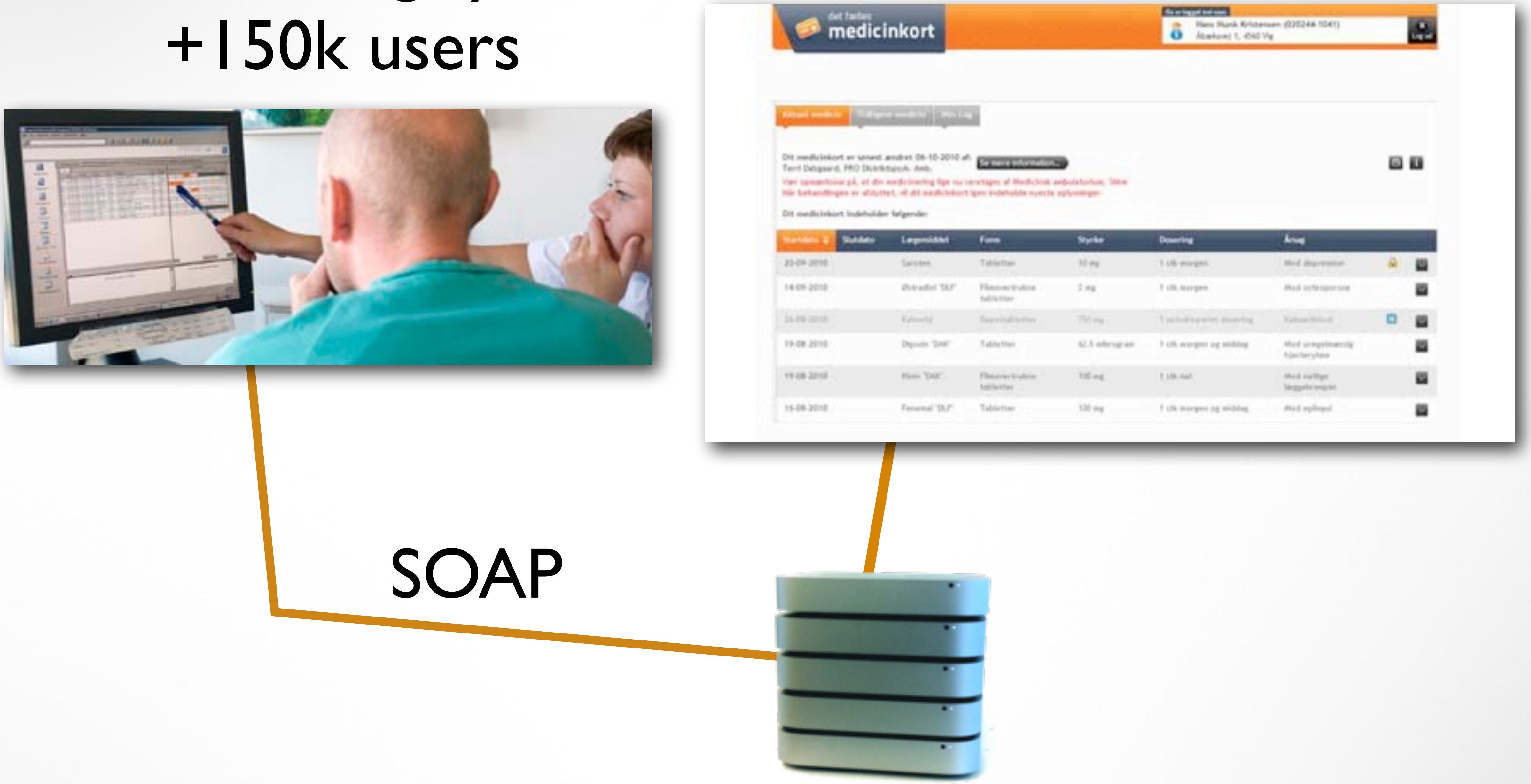
Vær opmærksom på, at din medicinering lige nu varetages af Medicinsk ambulatorium, Skive.
Når behandlingen er afsluttet, vil dit medicinkort igen indeholde nyeste oplysninger.

Dit medicinkort indeholder følgende:

Startdato	Slutdato	Lægemiddel	Form	Styrke	Dosering	Årsag
20-09-2010		Saroten	Tabletter	10 mg	1 stk morgen	Mod depression
14-09-2010		Østradiol "DLF"	Filmovertrukne tabletter	2 mg	1 stk morgen	Mod osteoporose
26-08-2010		Kalcevita	Depottabletter	750 mg	1 individuelt doseret dosering	Kalciumtilskud
19-08-2010		Digoxin "DAK"	Tabletter	62,5 mikrogram	1 stk morgen og middag	Mod uregelmæssig hjerterytme
19-08-2010		Kinin "DAK"	Filmovertrukne tabletter	100 mg	1 stk. nat	Mod natlige læggekramper
16-08-2010		Fenemal "DLF"	Tabletter	100 mg	1 stk morgen og middag	Mod epilepsi

TRIFORK.

15-30 existing systems
+150k users



det fælles medicinkort

Du er logget ind som: Hans Munk Kristensen (020244-1041) Åbækvej 1, 4560 Vig

Log ud

Aktuel medicin | Tidligere medicin | Min Log

Dit medicinkort er senest ændret 06-10-2010 af: Terri Dalsgaard, PRO Distriktspsyk. Amb.

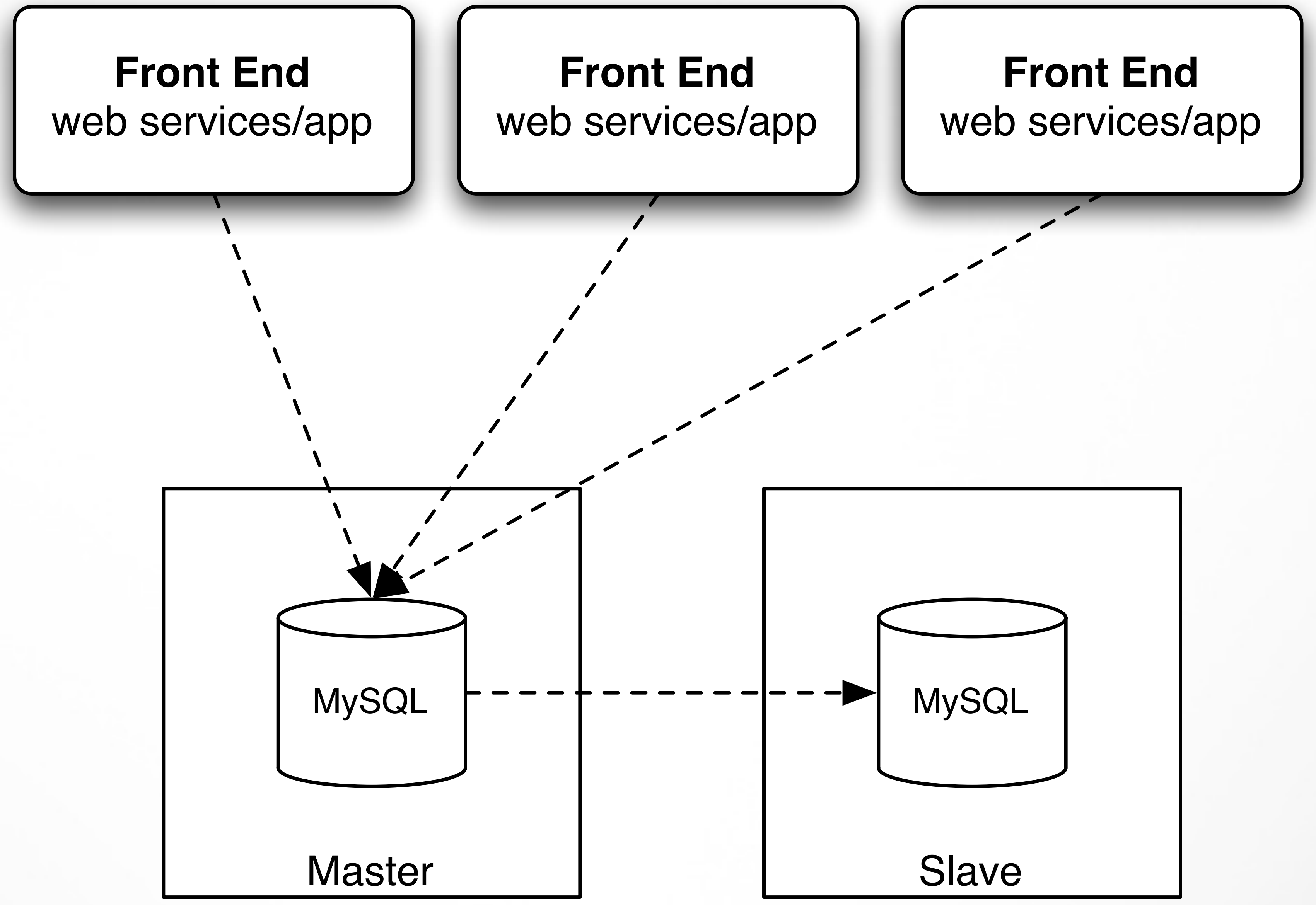
Se mere information...

Vær opmærksom på, at din medicinering lige nu varetages af Medicinsk ambulatorium, Skive. Når behandlingen er afsluttet, vil dit medicinkort igen indeholde nyeste oplysninger.

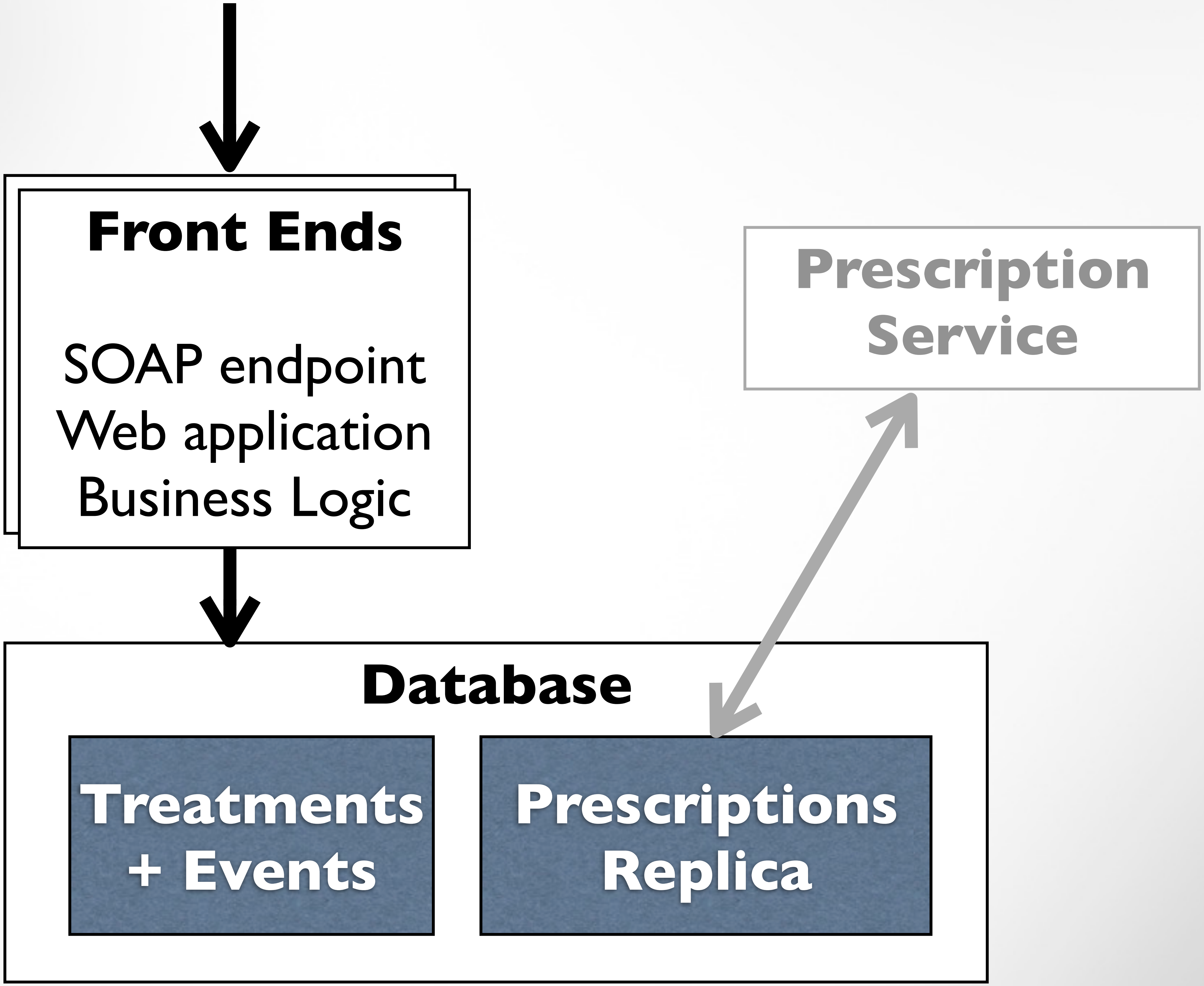
Dit medicinkort indeholder følgende:

Startdato	Slutdato	Lægemiddel	Form	Styrke	Dosering	Årsag
20-09-2010		Sarotex	Tabletter	10 mg	1 stk morgen	Mod depression
14-09-2010		Østradiol "DLF"	Filmovertrukne tabletter	2 mg	1 stk morgen	Mod osteoporose
26-08-2010		Kaleorid	Depottabletter	750 mg	1 [illegible] dosering	Kaliumtilskud
19-08-2010		Digoxin "DAK"	Tabletter	62,5 mikrogram	1 stk morgen og middag	Mod uregelmæssig hjerterytme
19-08-2010		Kinin "DAK"	Filmovertrukne tabletter	100 mg	1 stk nat	Mod natlige læggekramper
15-08-2010		Fenemal "DLF"	Tabletter	100 mg	1 stk morgen og middag	Mod epilepsi

"Old" Architecture

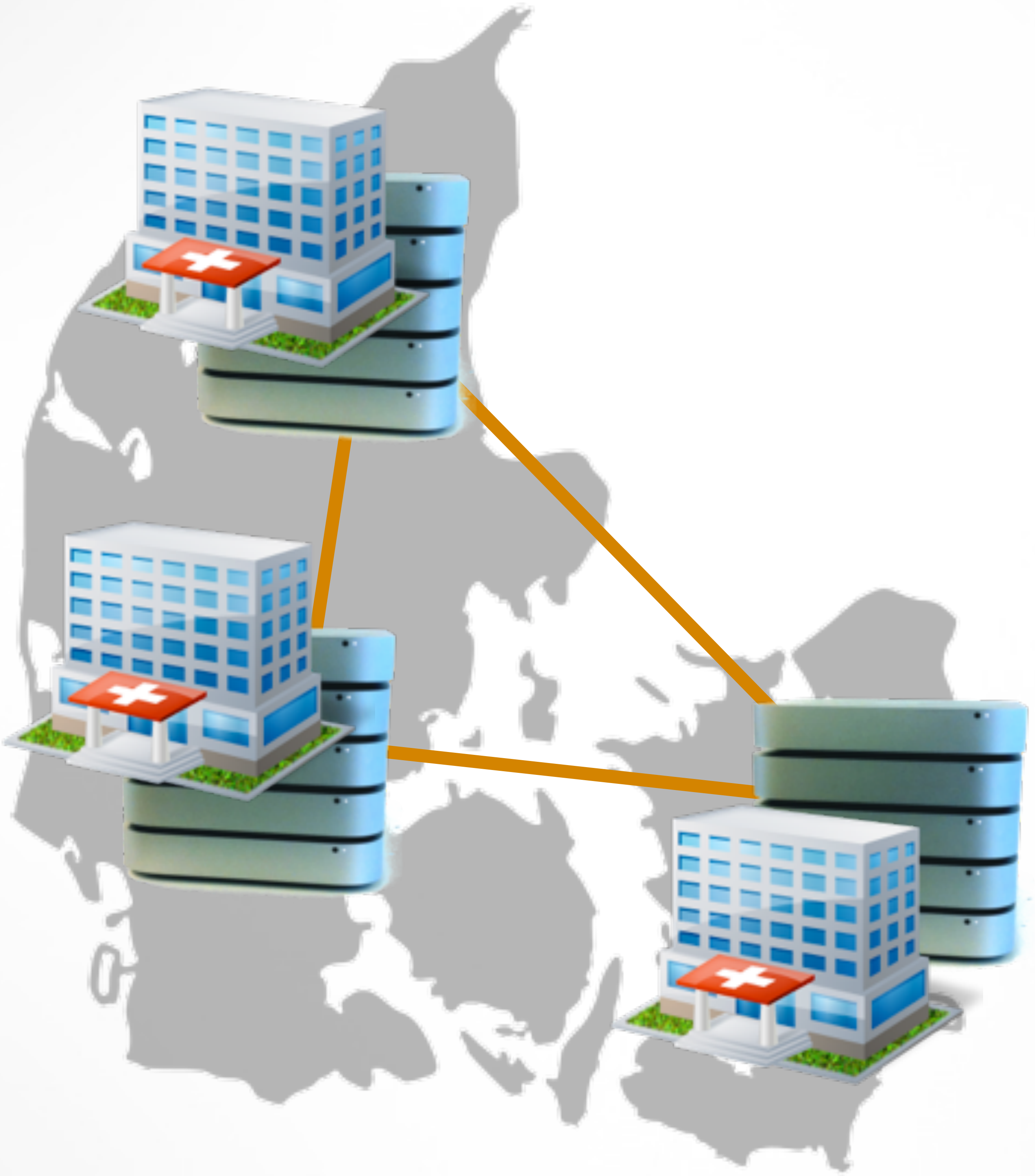


Overload


Front Ends
SOAP endpoint
Web application
Business Logic
Prescription Service
Database
Treatments + Events
Prescriptions Replica

TRIFORK.

Distributed Architecture

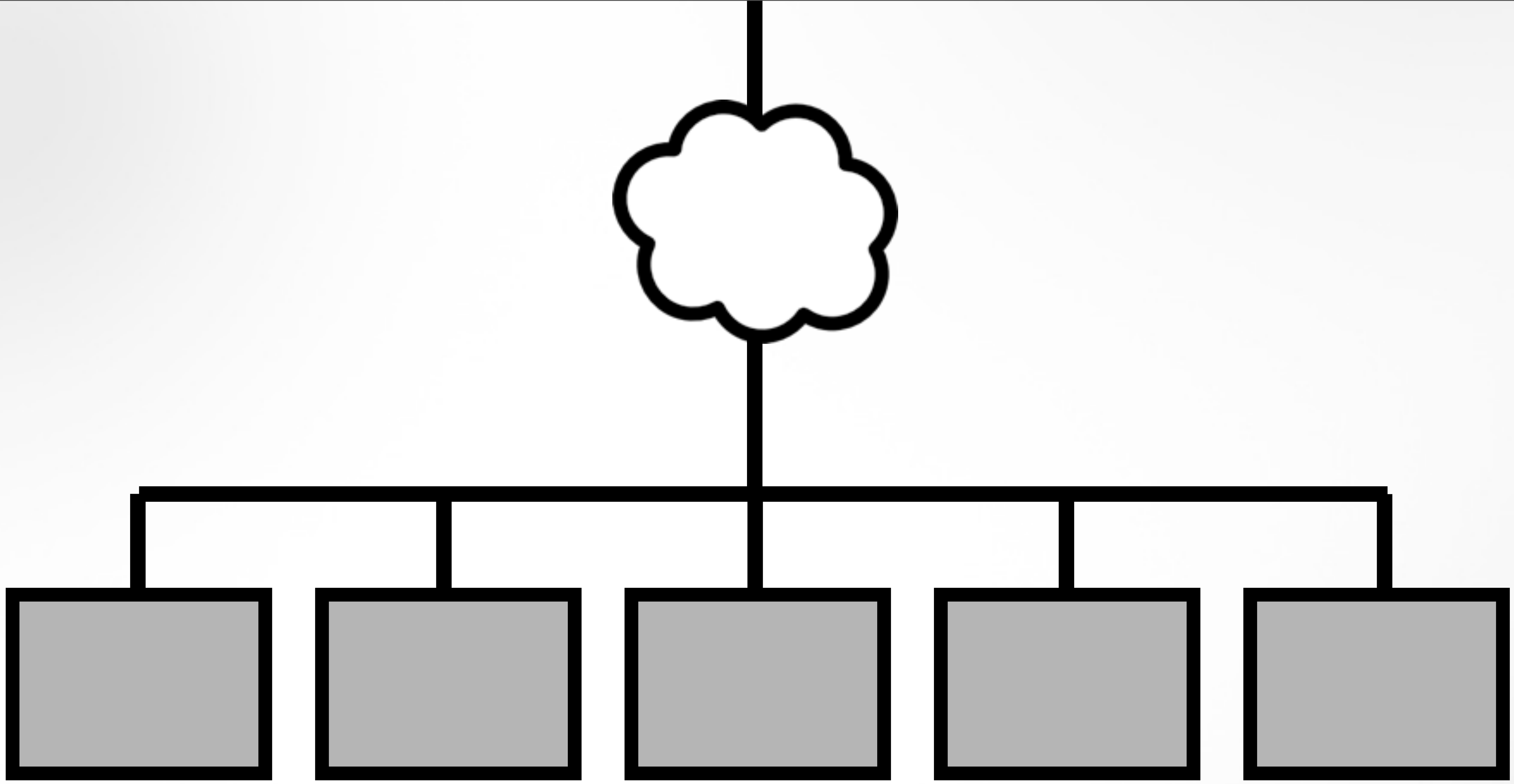


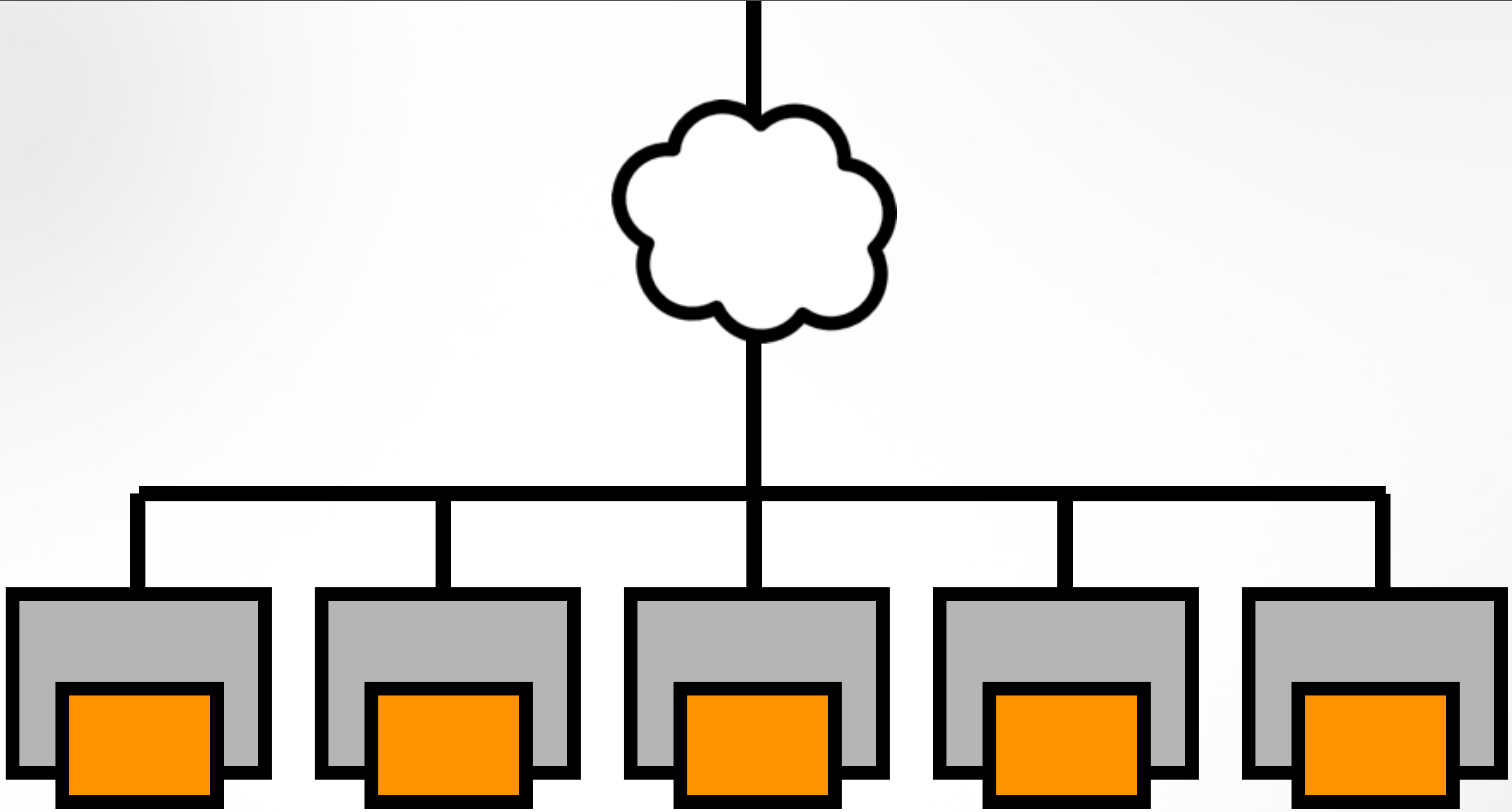
- Availability: Run in multiple data centers
- Scalability: Prepare the system for expected growth

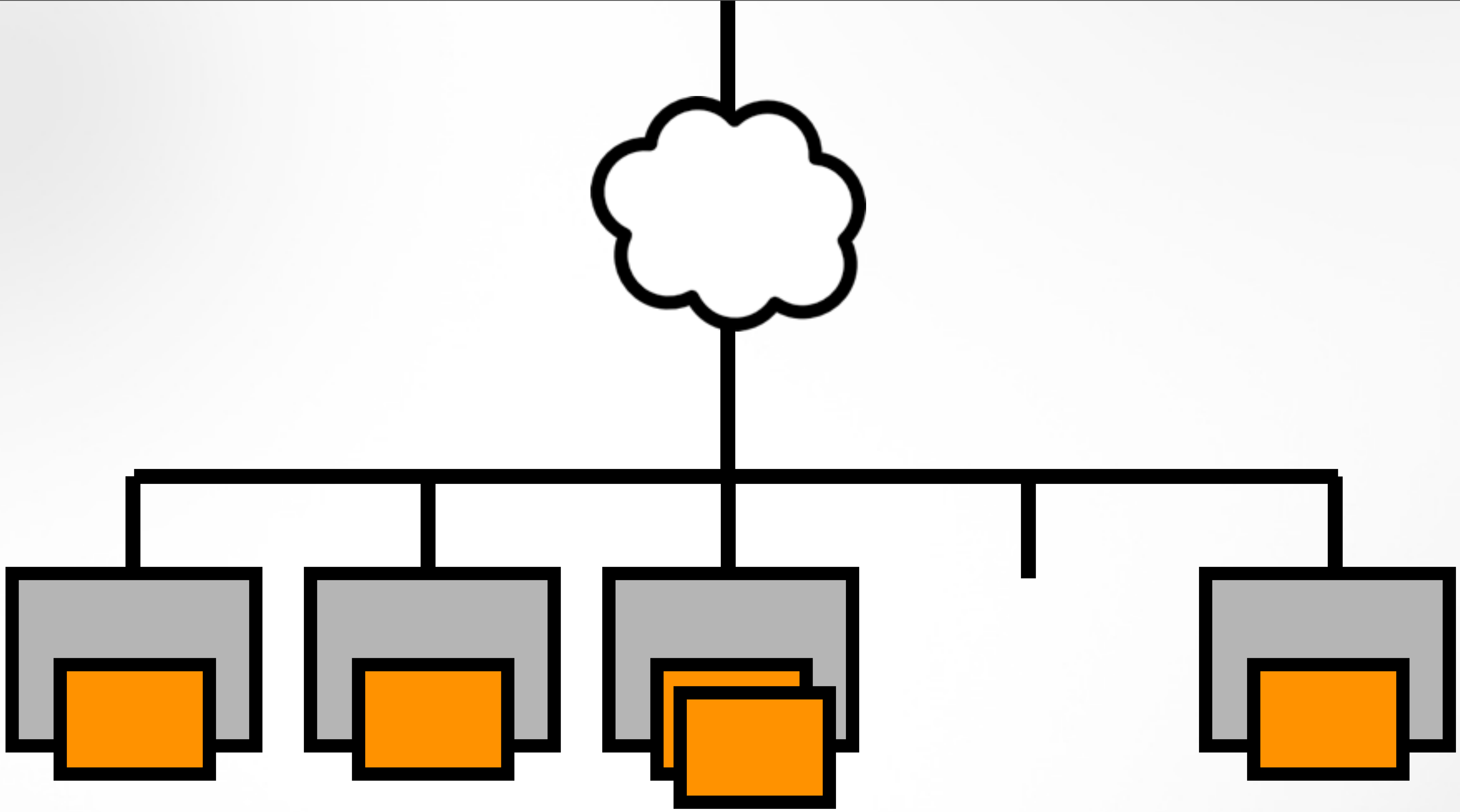
TRIFORK.

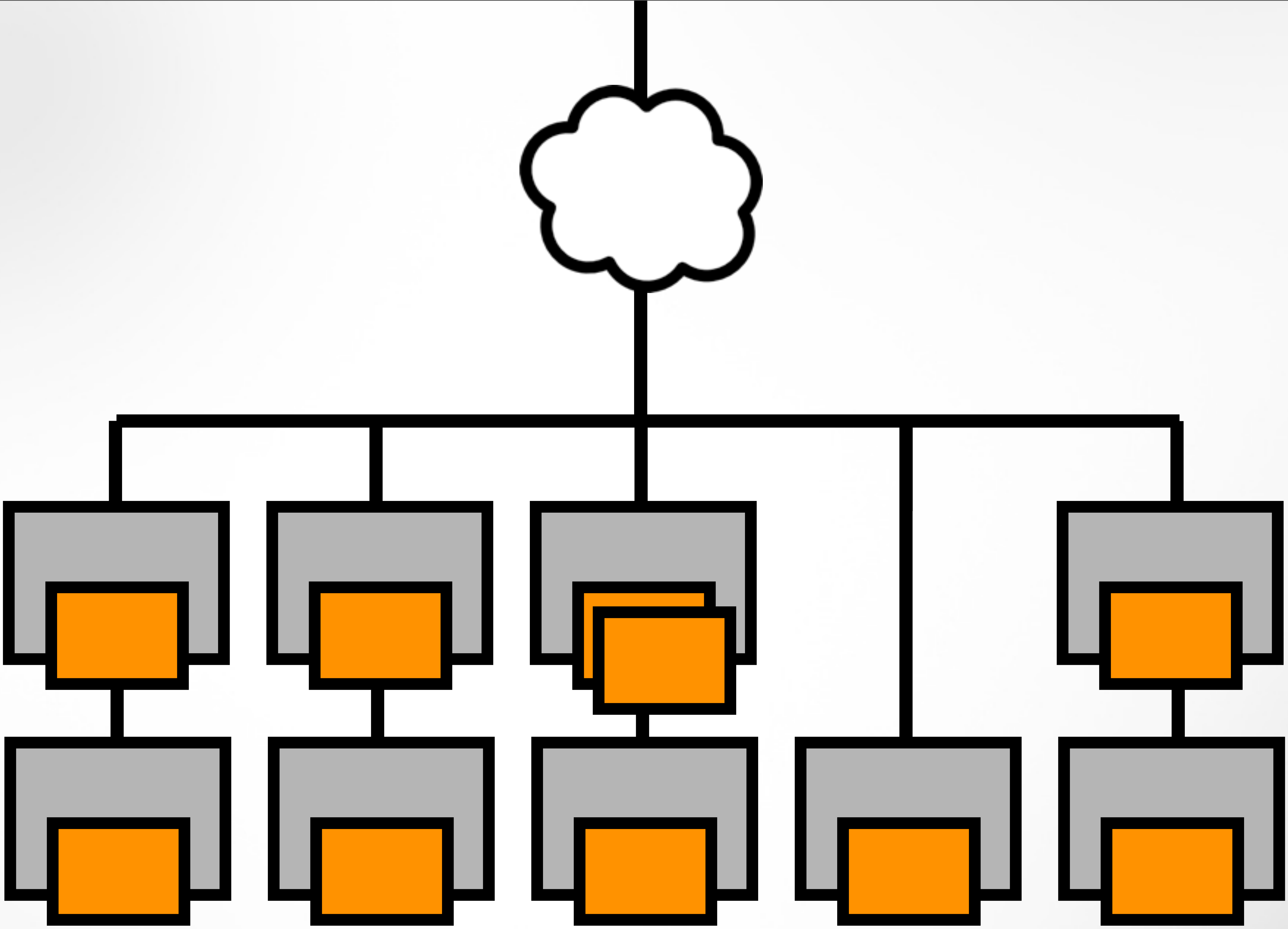
Riak Data Store riak

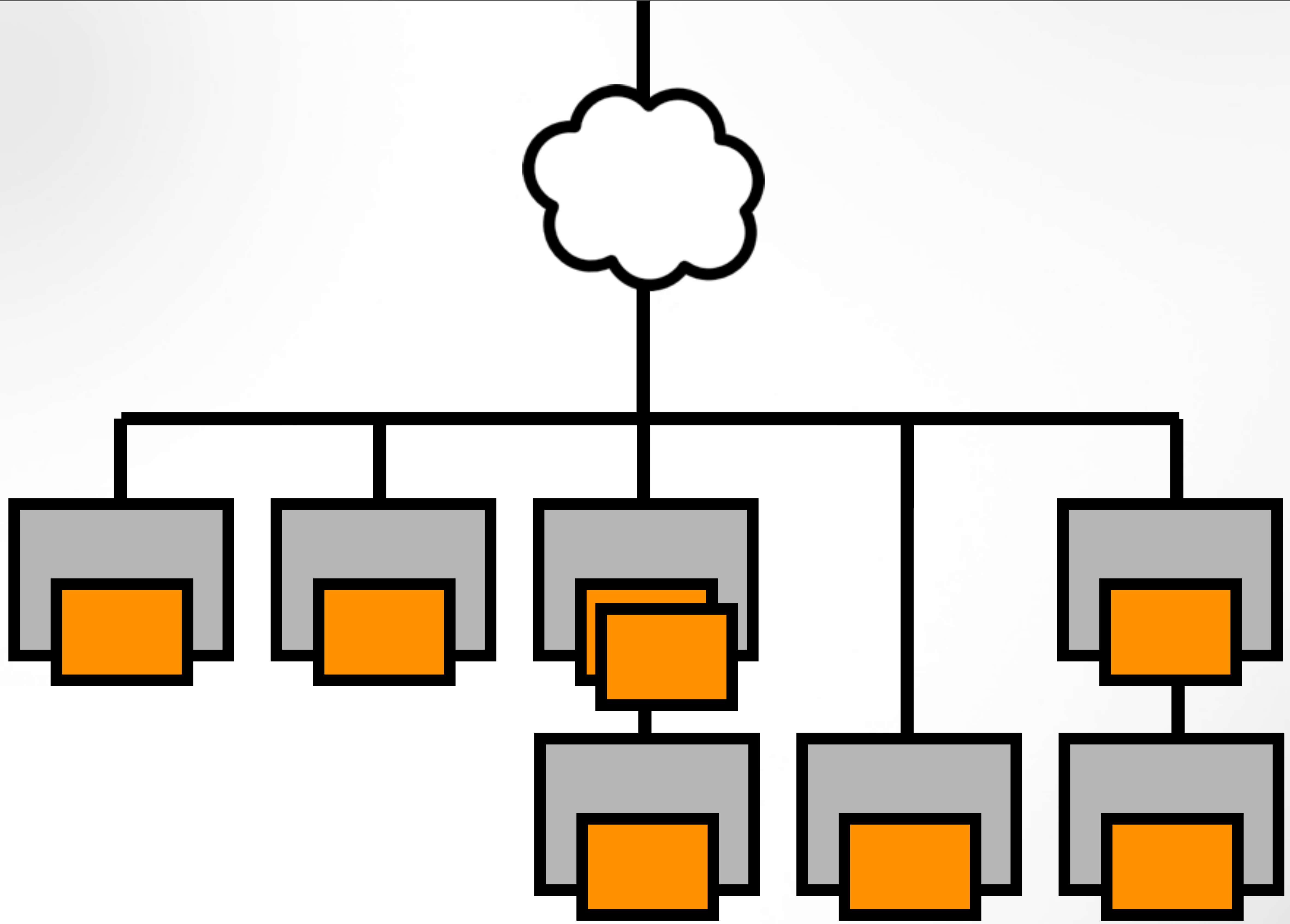
- Fit the general requirements
  - Availability + Scalability
  - Operational improvements
- Challenges
  - Key/Value Store, vs Relational Model
  - New technology, many unknowns

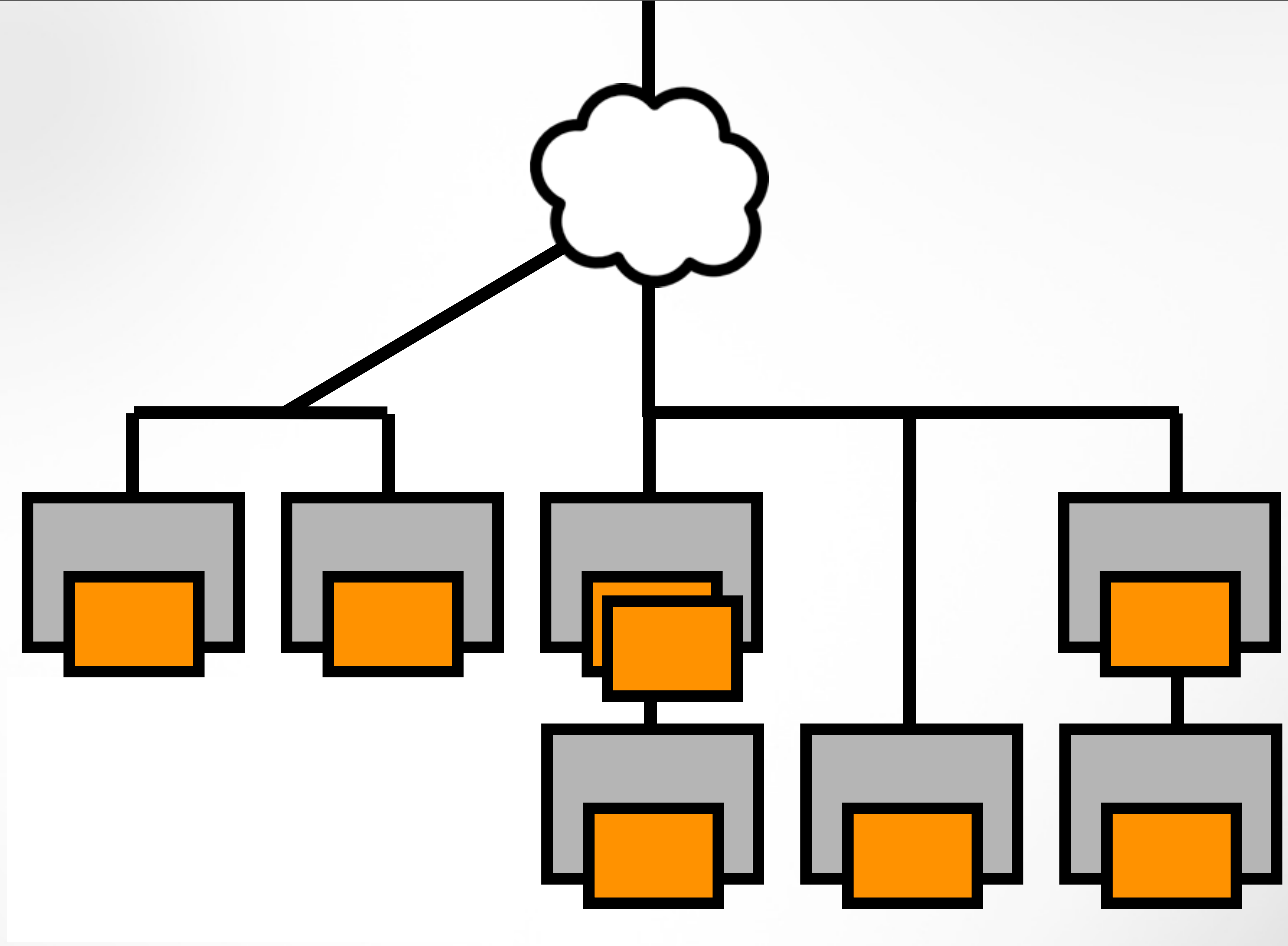


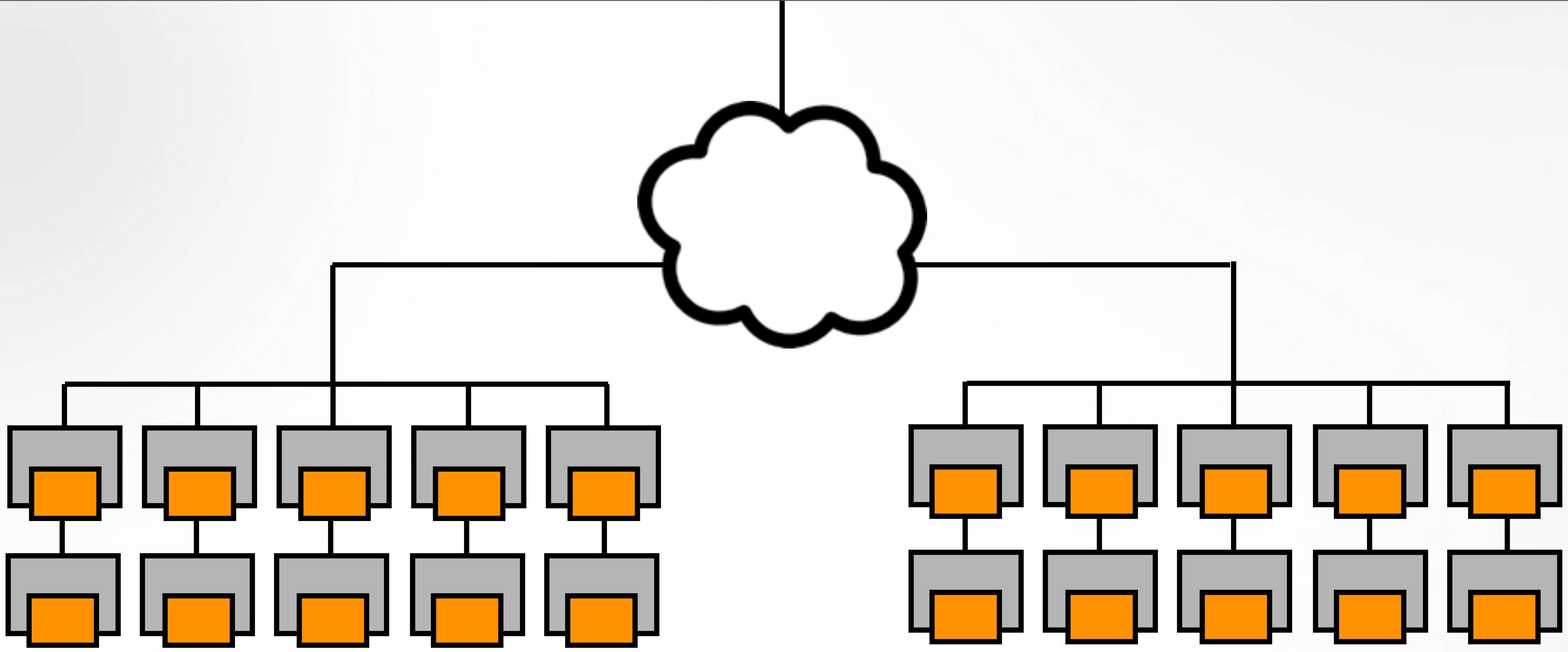


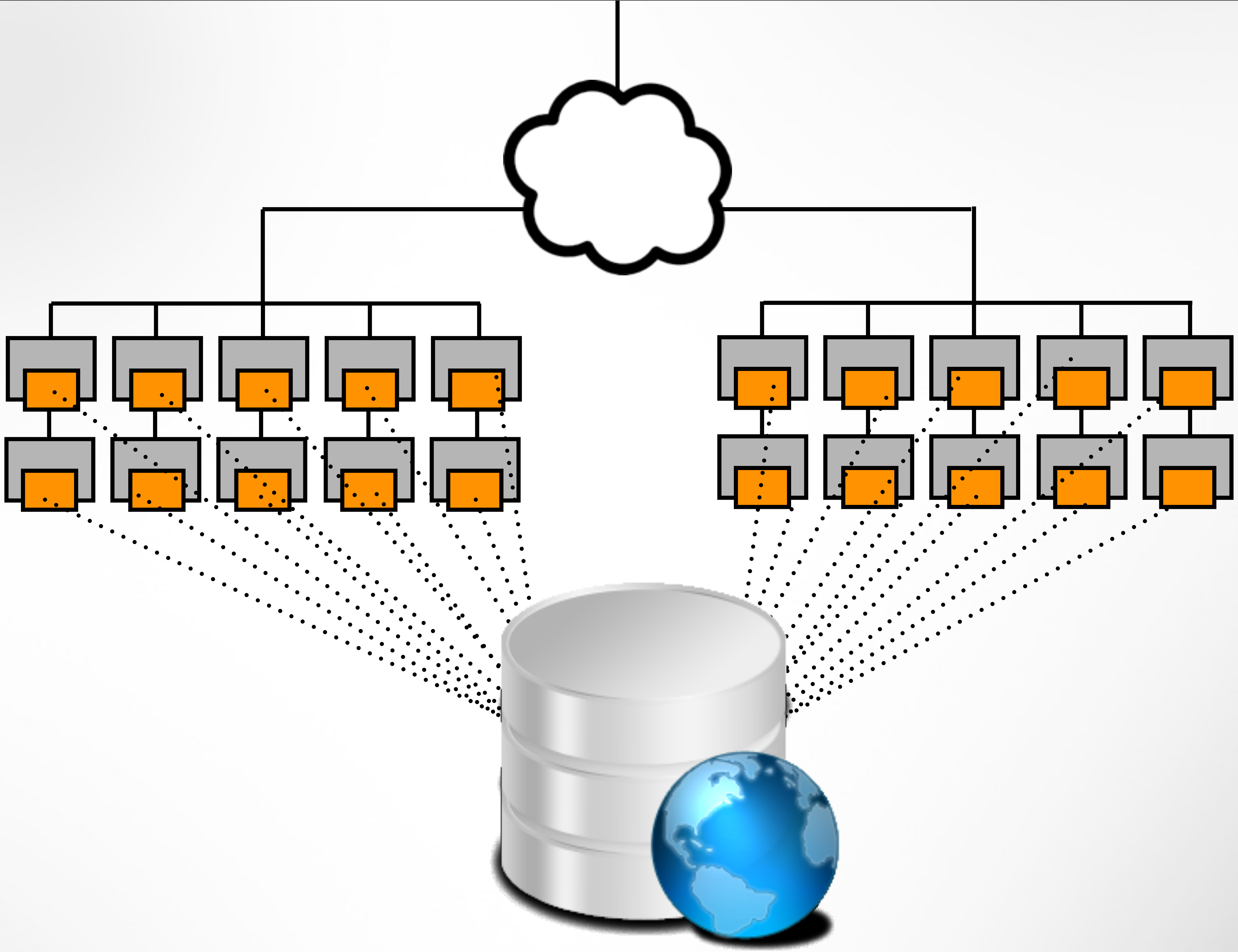


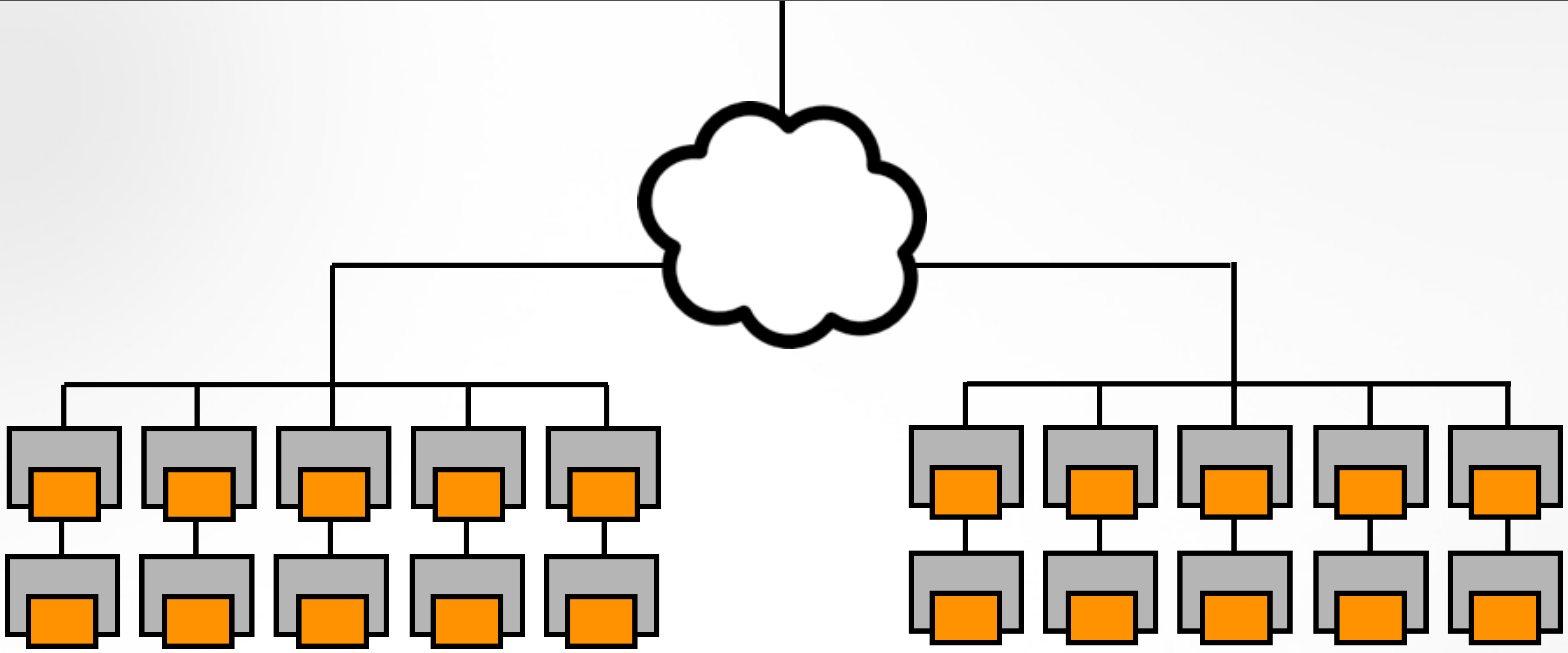




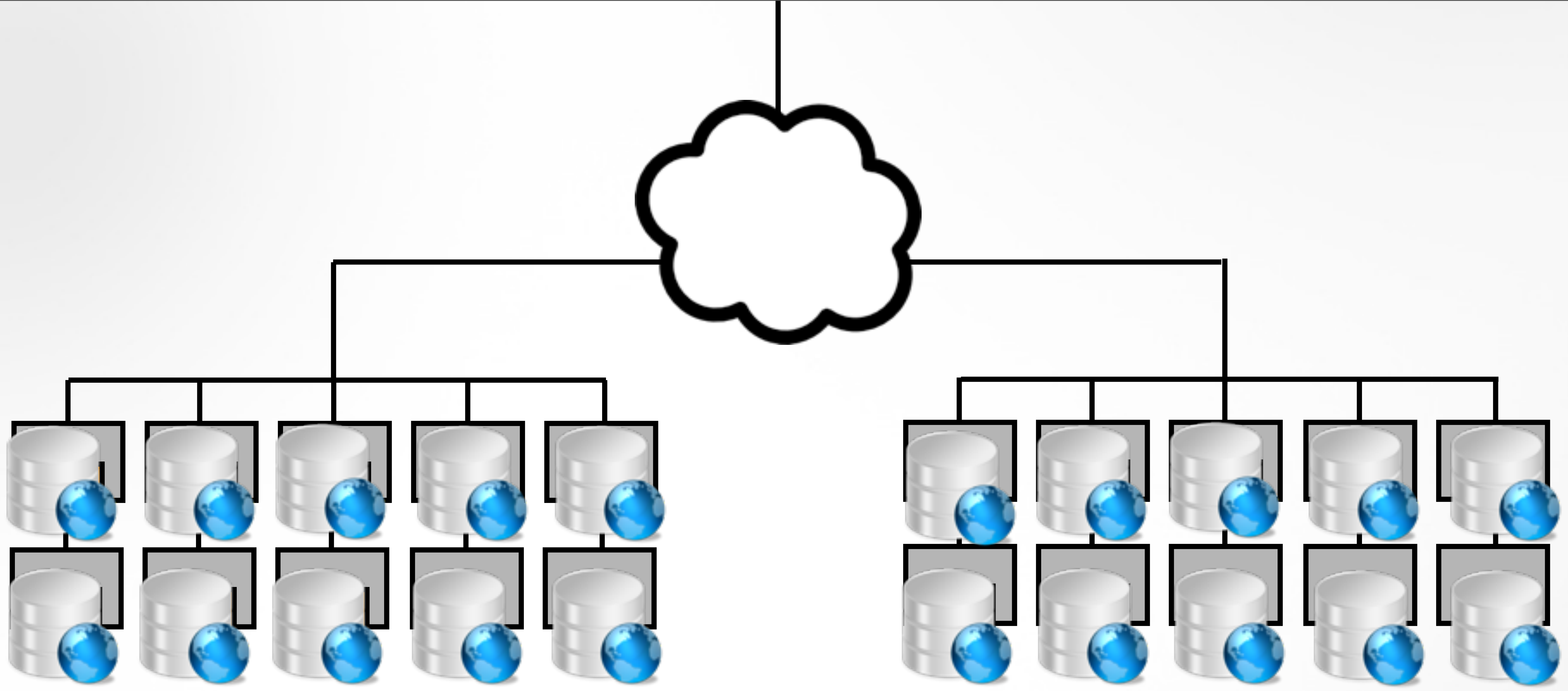


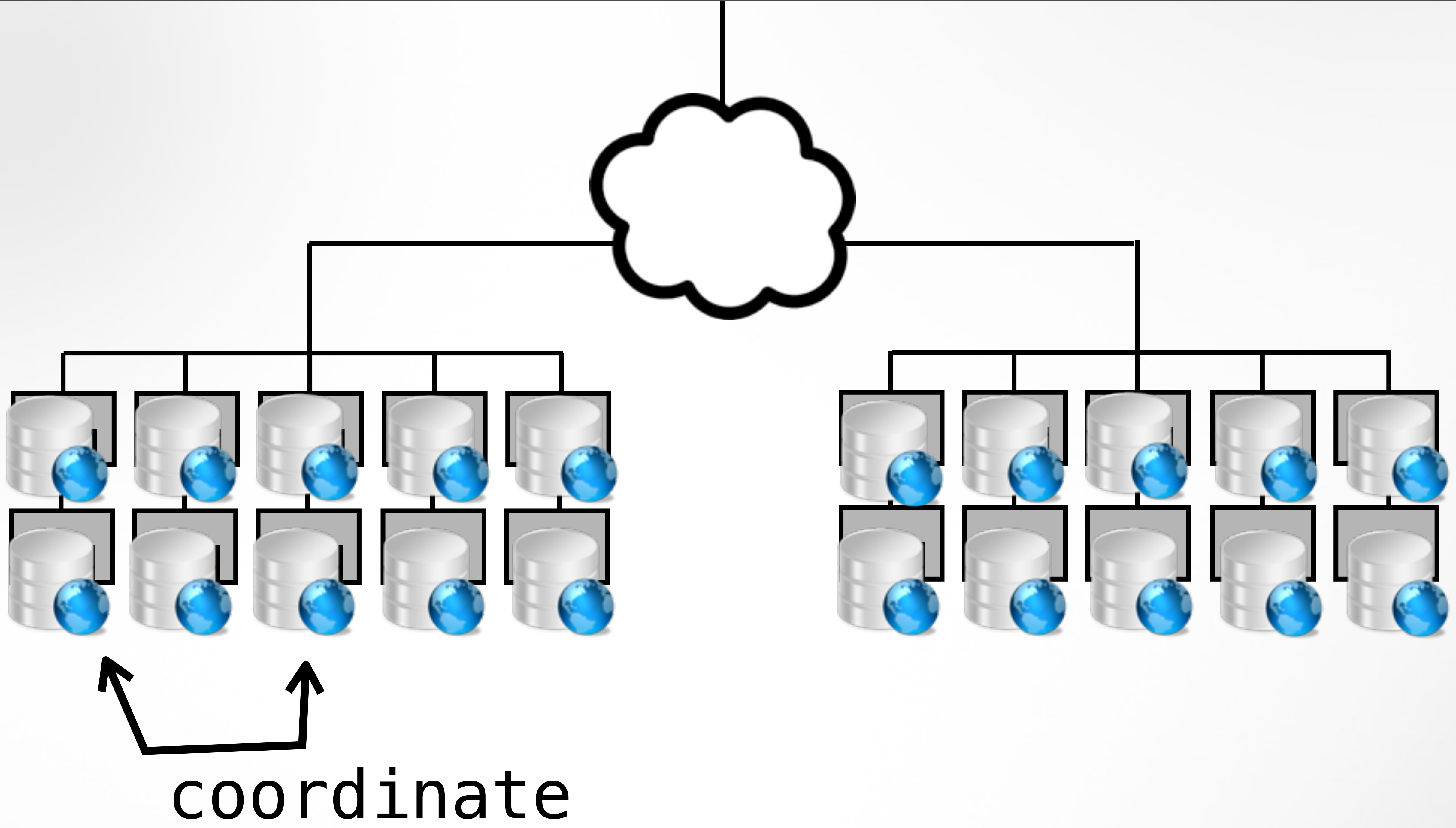


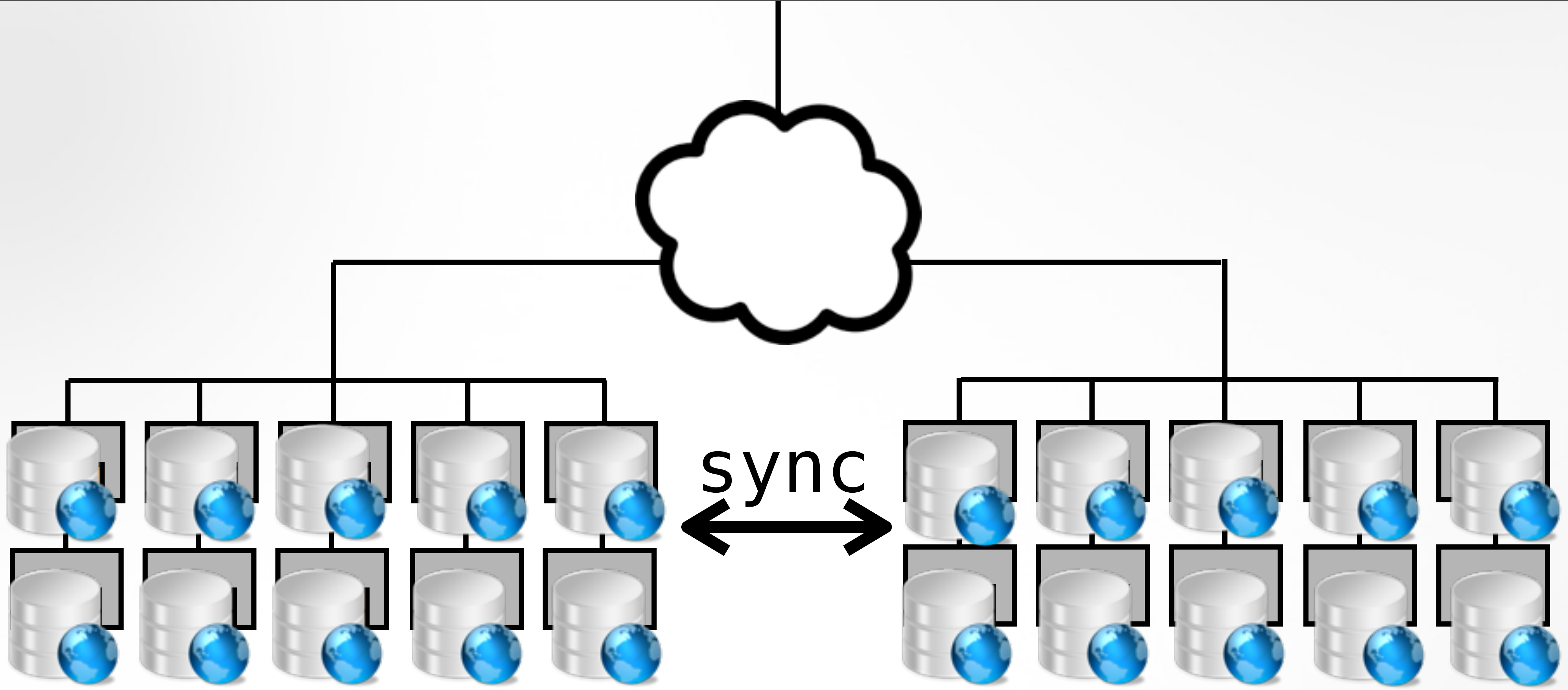




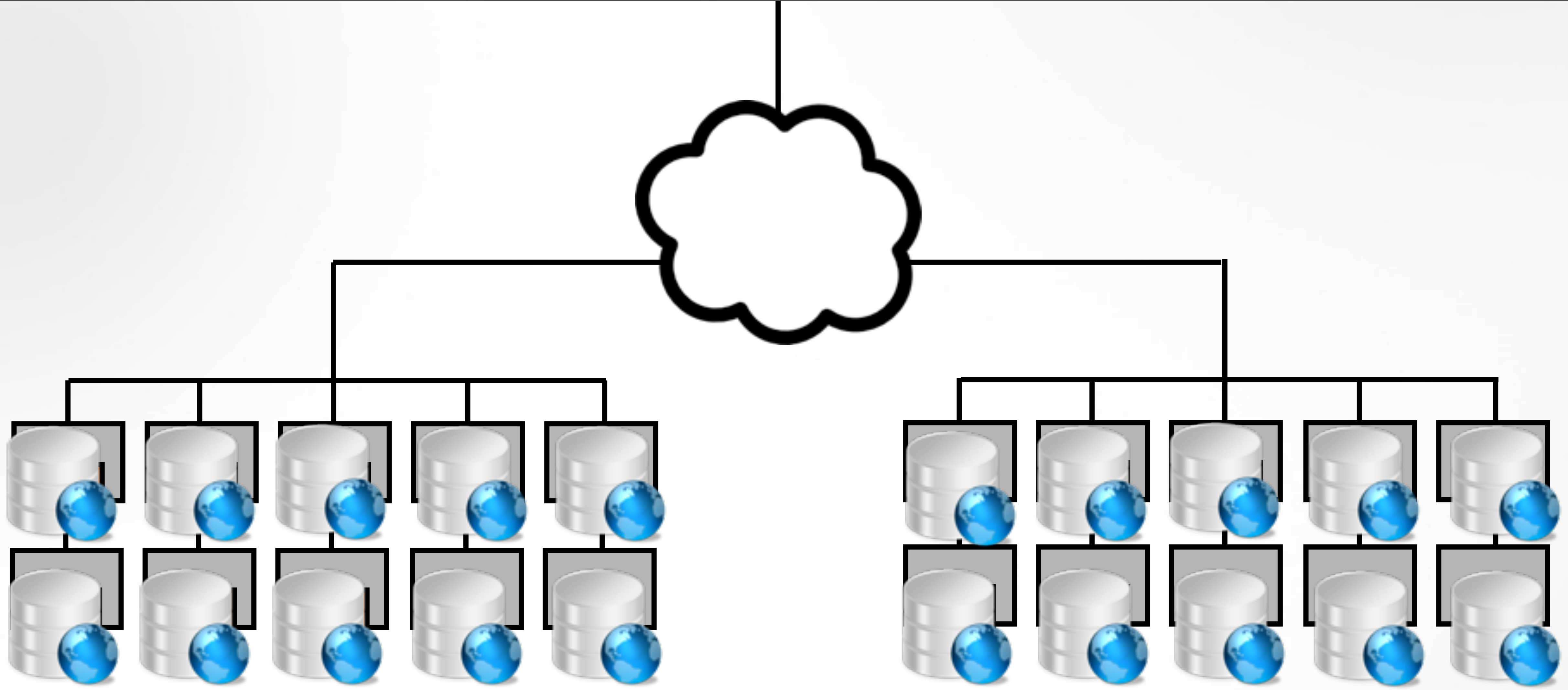
TRIFORK.




coordinate

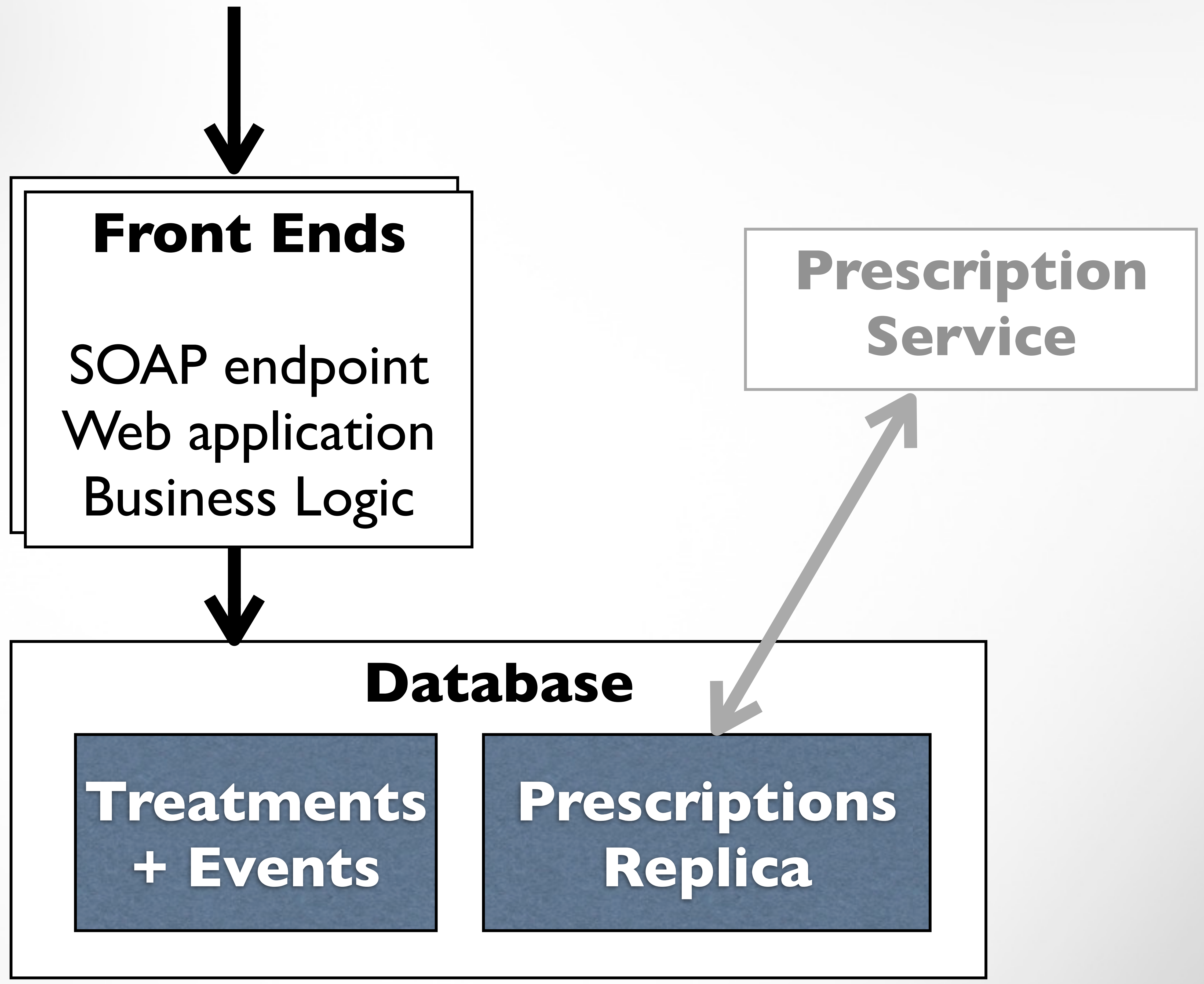


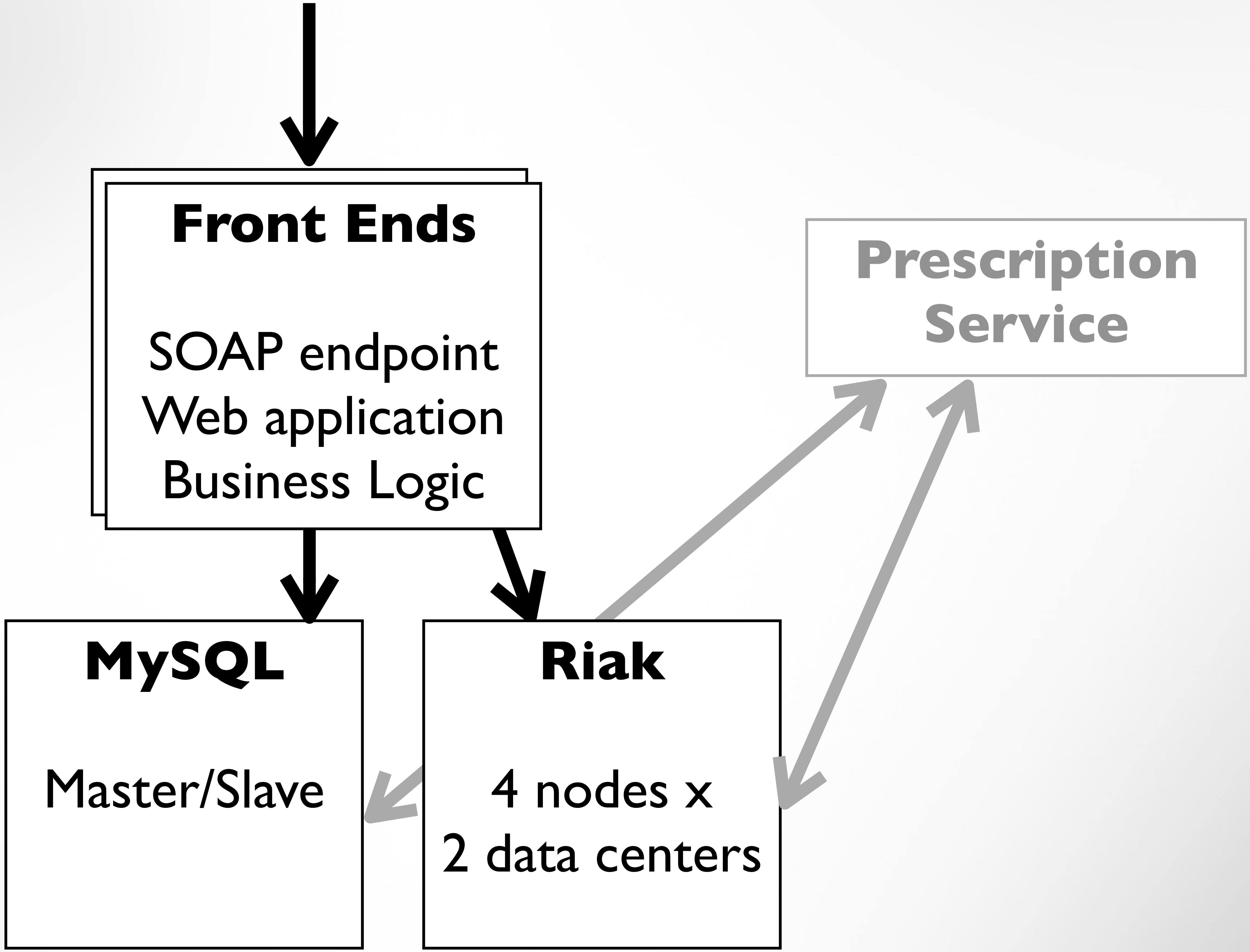
TRIFORK.



riak

- scalable and available
- system captures write conflicts
- resolve lazily (read repair)


Front Ends
SOAP endpoint
Web application
Business Logic
Prescription Service
Database
Treatments + Events
Prescriptions Replica


Front Ends
SOAP endpoint
Web application
Business Logic
Prescription Service
MySQL
Master/Slave
Riak
4 nodes x
2 data centers

Challenges

- Data model: how to go from Relational model to Key/Value model
- Experiences with Riak's backends
- How to keep version history
- A true war story

TRIFORK.

Data Model

- Integrity without ACID transactions
- Riak's default storage keeps *all* keys in memory
- Dealing with Write Conflicts

Phase I

- To validate the architecture, we built a system where these are kept in Riak:
  - Prescription Replicas
  - Audit-log
  - Request cache

First Attempt: Using Links

~5 million

Person
Key: Person-ID
Links: Prescription-ID*

~200 million

Prescription
Key: Prescription-ID
Content: Protobuf+GZip

- Allows reading of **N** record in one roundtrip
- Performance suffered: **1+N** disk access
- Too many keys in memory

First Attempt: Using Links

~5 million

Person
Key: Person-ID
Links: Prescription-ID*

~200 million

Prescription
Key: Prescription-ID
Content: Protobuf+GZip

- Ran poorly on Virtual Hardware
- Trying to figure out how to handle conflicts

Second Take

~5 million

Prescriptions
Key: Person-ID
Content: Protobuf+GZip

- Very simple: read - resolve - modify - write
- Integrity: 1 person ↔ 1 record
- Performance good: **1** disk access
- All keys fit in memory

Read Repair

- On every read, we handle write conflicts
- If so, auto-merge[*], store and re-read
- Resolve: Merging is *business logic*; some merge actions need user attention, others don’t.
- Forward: This is also the hook for schema evolution

TRIFORK.

The Audit Log

- ~1 billions log entries per year
- Stores generic JSON documents
- Need some search capability
- Bitcask backend was not an option

The Audit Log

- InnoDB backend [basically MySQL]
- Increasing keys for B-tree backend “YYYYMMDDhhmmss:<random-bits>”
- Indexing in SQL store
  - New version of Riak has a new backend with secondary indexing capability, which we’ll try out

Request Cache

- Makes SOAP-endpoints idempotent
- Keep Request/Response for 14 days
- Perfect fit for default Bitcask backend

A Real War Story...

- First production launch with Riak
- Strange data corruption started to appear
- Also spontaneous I/O errors sometimes
- Does not exactly make you comfortable...

TRIFORK.

A Real War Story

- We installed commit hooks in Riak (MD5 validation)
- TCP data was being corrupted **in transit**!
- Spottet **IP-Headers** in the middle of data
- Operations folks were still suspicious...

A Real War Story

- The problem was a buggy network driver
- TCP checksumming is very simple
- $1/2^{16}$ packets was let thru - MD5 caught it
- Also the reason for I/O dropped connections

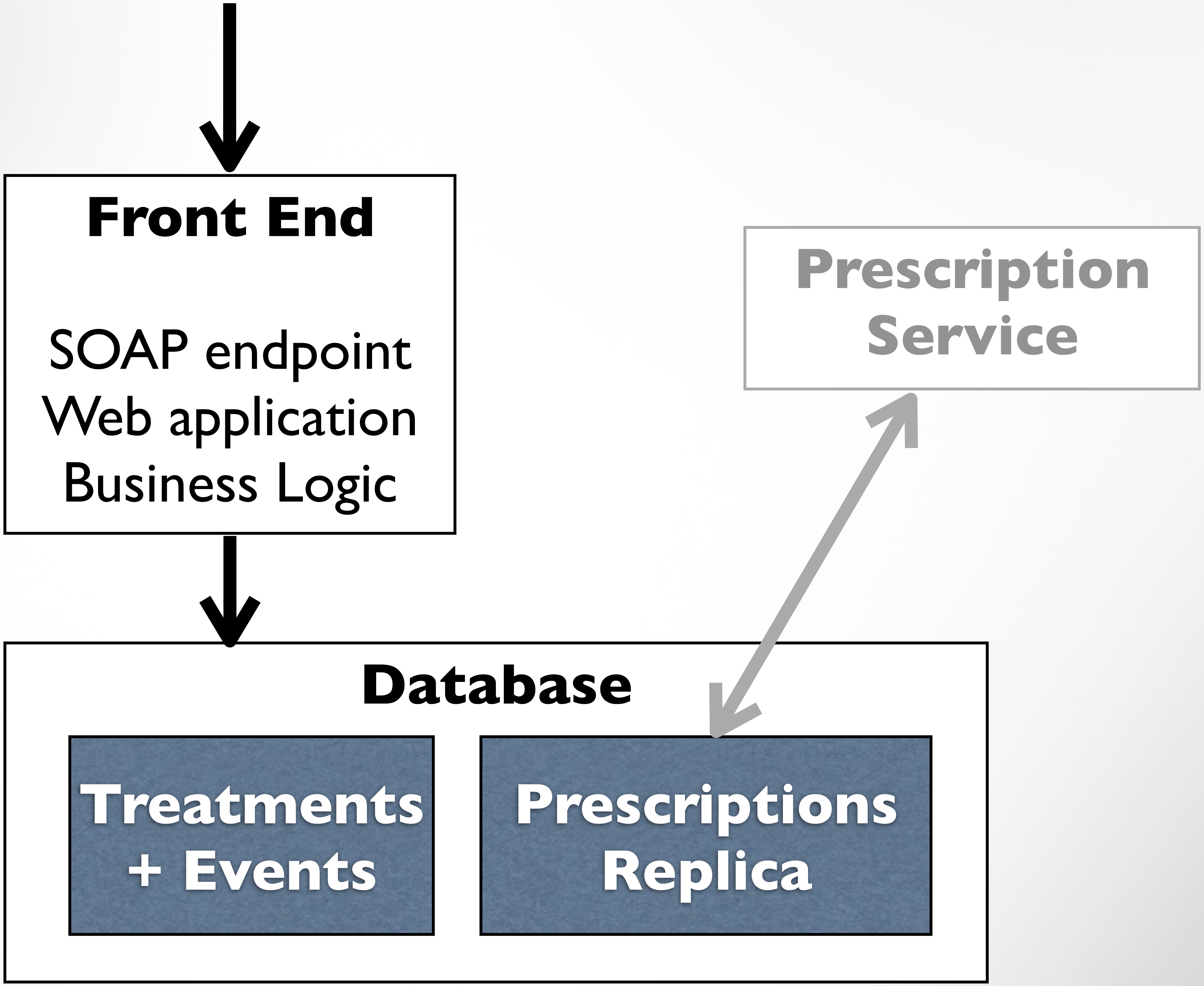
Phase I: Conclusions

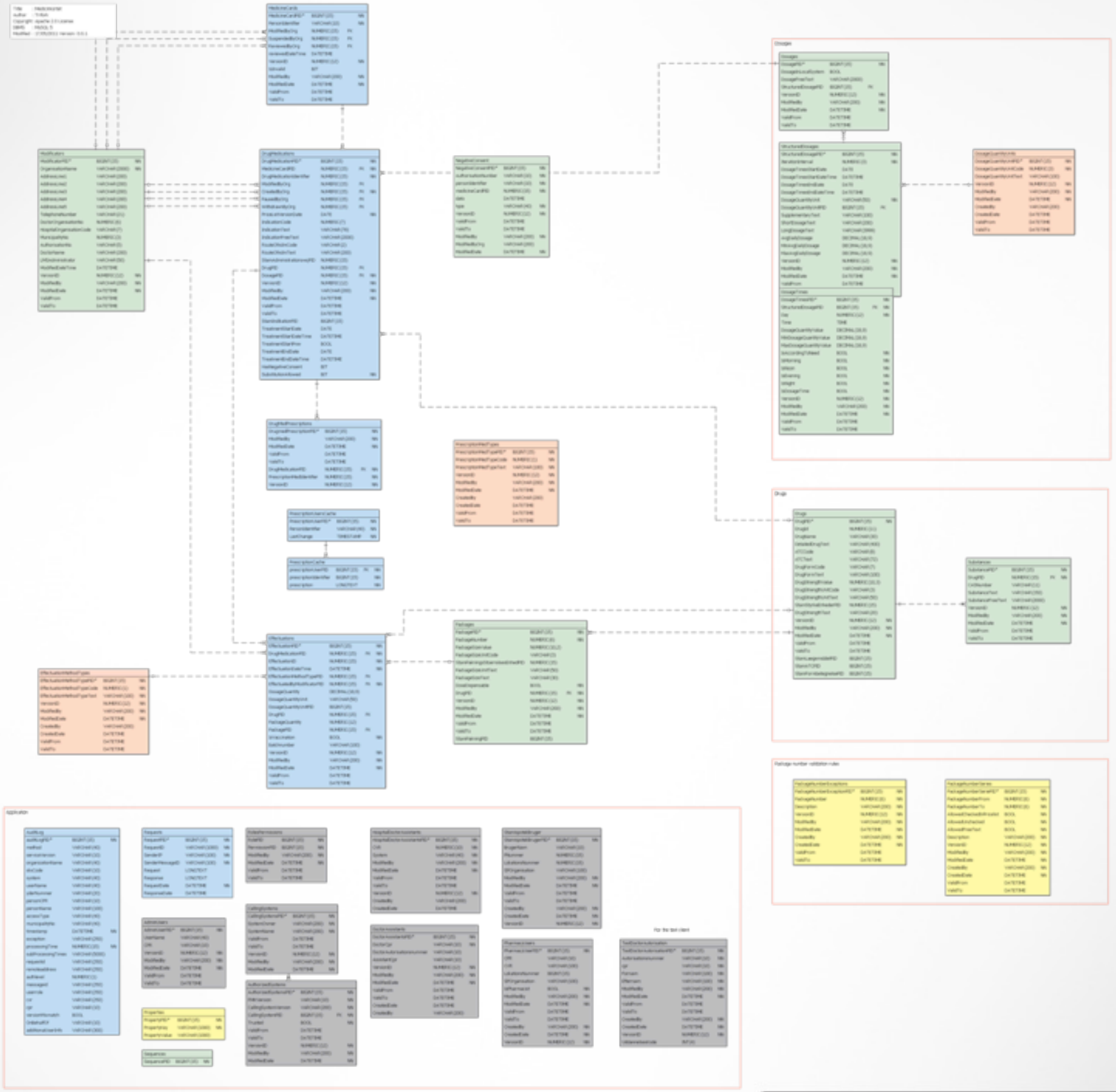
- 3 data sets - 3 different solutions
- Availability & Scalability
- Response times are better and more predictable
- Before: Locked at max # ops/sec
- Now: 4 x ops/sec ... and can scale more

TRIFORK.

Phase II

- Move “the rest” of the application into Riak
- Building this on new Riak 1.0
  - Secondary indexing
  - Version/history
  - Ad-hoc querying


Front End
SOAP endpoint
Web application
Business Logic
Prescription Service
Database
Treatments + Events
Prescriptions Replica



~5 million

MedicineCard
Key: Person-ID
Content: Protobuf+GZip

~5 million

Prescriptions
Key: Person-ID
Content: Protobuf+GZip

- We'll use the same simple data model
- Storing version history using "DeltaZip"
- Provide ad-hoc querying using XPath/protobuf

DeltaZip

V_1	V_2	V_3	...	V_N

- Store all versions of data in one record
- Compress **data**$_M$ using **data**$_{M+1}$ as compression dictionary.
- Works amazingly well for our kind of data, since we just update some of an object

Querying Riak w/ XPath

- We’ve built an xpath evaluator for JSON and protobuf data - simple Map/Reduce
- For protobuf encoded record we store it’s schema in a header.
- Avoids using javascript or erlang for map/reduce querying

Consider this...

- How much scalability/availability do you need?
- Multi-version (update data)
  - Read-repair of write conflicts
- Last write wins (caching, logging, ...)
  - No need to handle conflicts
- Store complex data by natural keys

Conclusions

- “Eventual consistent” may be better match for your business problem than ACID
- Data Modeling involving large datasets is very different [have to consider physics]
- The system runs faster [throughput + response time]
- We sleep better at night

Thank You.

@drkrab

TRIFORK.