# Evolution of Web Application Architecture

PHP Unconference Hamburg

Kore Nordmann / @koredn / <kore@qafoo.com>

Spetember 19th, 2015

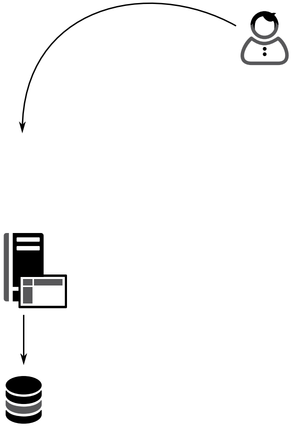
Qafoo
passion for software quality

## About Me

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

## Evolution



passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

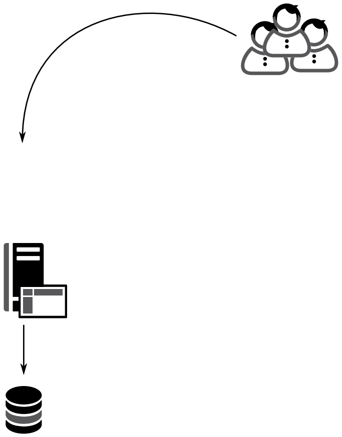
## Problem

Too many visitors

passion for software quality

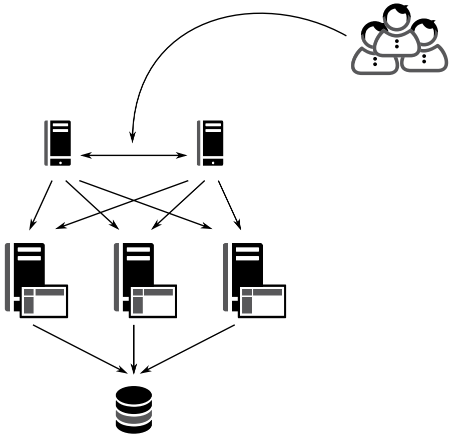
Copyright Qafoo GmbH; All Rights Reserved

# Evolution



Qafoo
passion for software quality

# Evolution



Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Load Balancing

- Works because of HTTP & PHP
  - HTTP is LCoDC$SS
  - PHP is build for shared-nothing
- Round Robin works best
  - Sticky sessions will overload certain servers

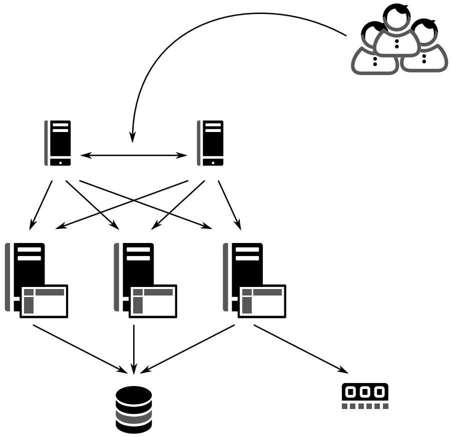
Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Problem

Non-sticky session – how?

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

talks.qafoo.com

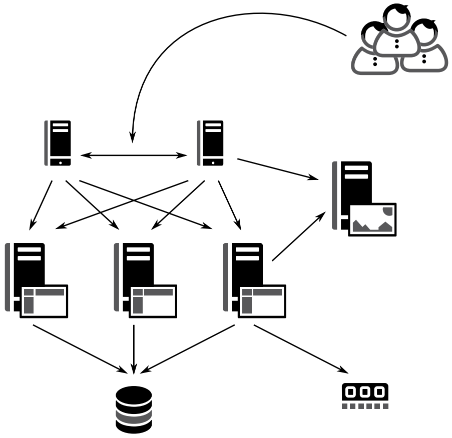
## Problem

Where to put the static data?

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

## Evolution



Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Static Files

- NFS will eventually lead to dead locks
  - . . . still seems the most popular solution around.
- Multiple domains can hurt performance (TCP slow start)
- Using dedicated CDN providers can help
  - Content locality

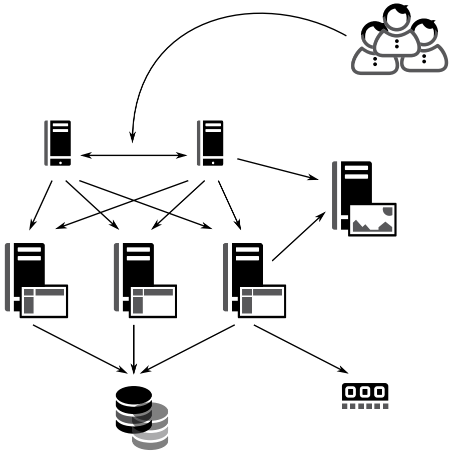
Copyright Qafoo GmbH; All Rights Reserved

## Problem

DB server too slow

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



## Lessons Learned: Replicate Database

- Master Slave Replication is fairly easy to set up
  - Obviously only scales READs
  - WRITEs are usually not your first problem

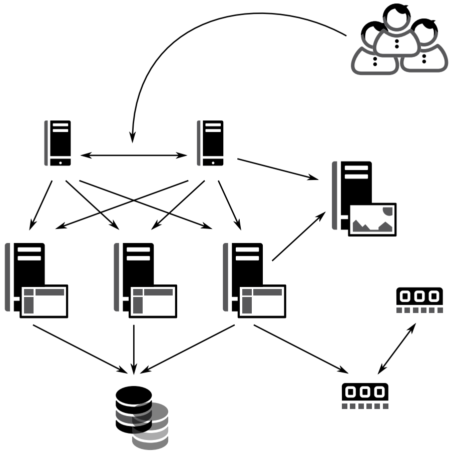
## Problem

DB servers are too expensive

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Cache With Memcache

- Cache all the things in *memory*
  - Cache entities
  - Cache collections
  - Full page cache
- Cache invalidation

  *There are three hard things in Computer Science: Cache invalidation and off by one errors.*

  - Cache dependency calculation
  - The paging problem

Qafoo
passion for software quality

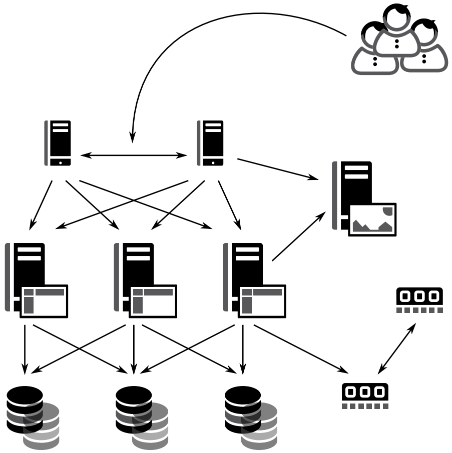
Copyright Qafoo GmbH; All Rights Reserved

## Problem

Too many writes

Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

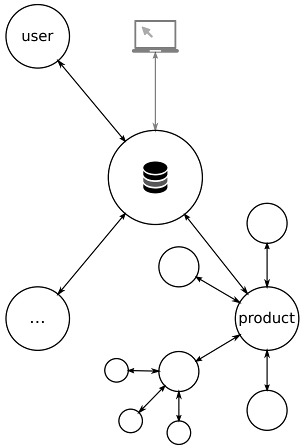
## Evolution



Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Sharding

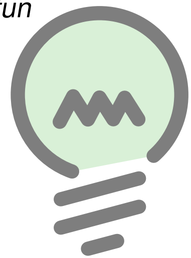
- Split tables across multiple nodes
- Shard by consistent hashing



Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Sharding

- Shard by table
  - … or even shard by consistent hash per entity
- No referential integrity checking
- Queries are limited to sharding solution
- Schema updates across multiple shards are *fun*



passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

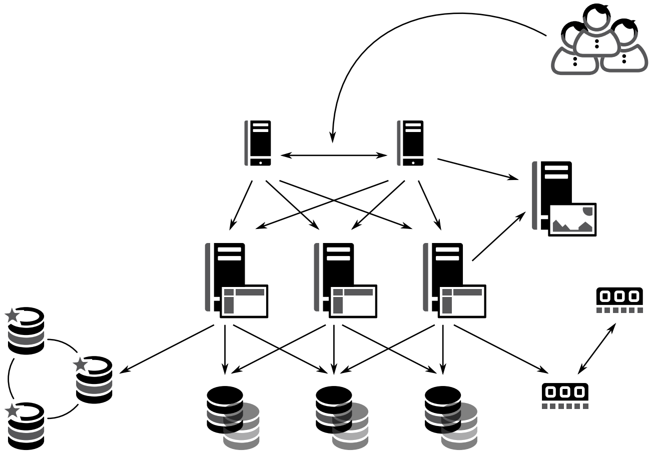
## Problem

Database setup too complex

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: NoSQL

- Usually solves one problem really well:
  - Sharding
  - Multi-Master-Replication
  - Cross-shard queries
- Usually omits:
  - SQL
  - Referential Integrity
- . . . we lost all relevant features from Relational Database Management Systems anyways. . .

Copyright Qafoo GmbH; All Rights Reserved

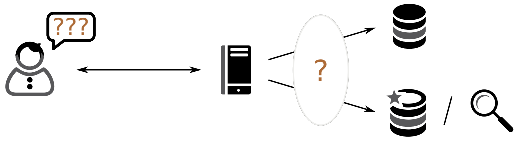
## Data Consistency

Keeping data consistent across multiple storages

passion for software quality

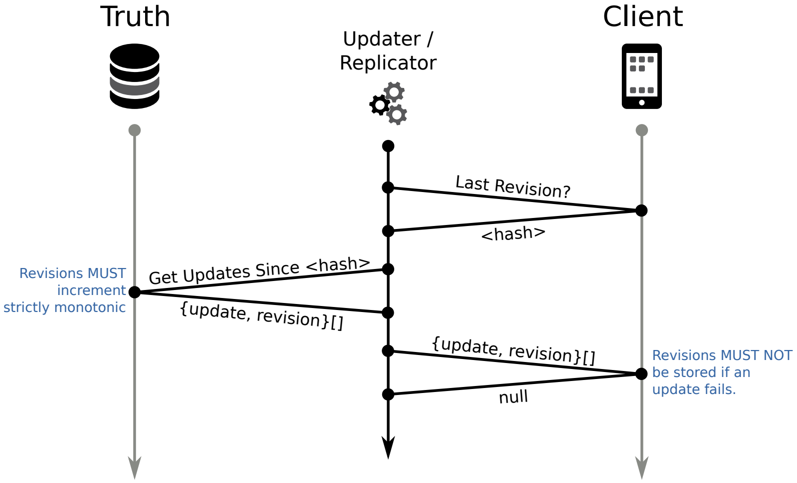
Copyright Qafoo GmbH; All Rights Reserved

## Data Consistency Across Nodes



passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

# Eventual Consistency



Qafoo passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Data Consistency

- Embrace Eventual Consistency
  - Compaction is hard
  - Data migrations are hard

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

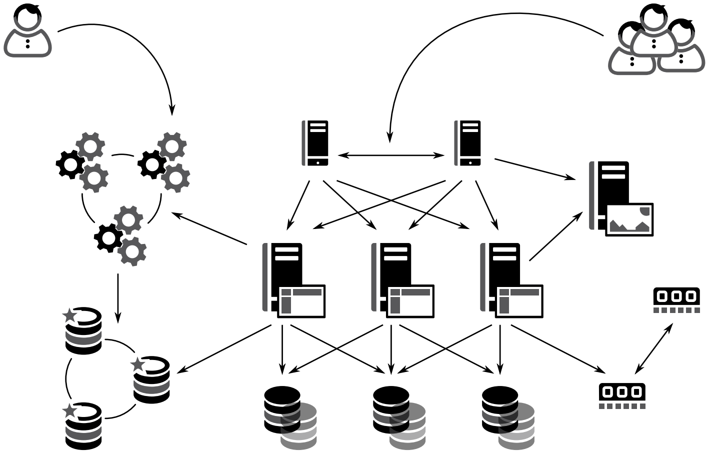
## Problem

Business wants to query data

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



Qafoo
passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Map-Reduce

- Execute queries on distributed databases
- New query language to learn
  - Your developers write analysis scripts, instead of the business analysts writing slow SQL queries

passion for software quality

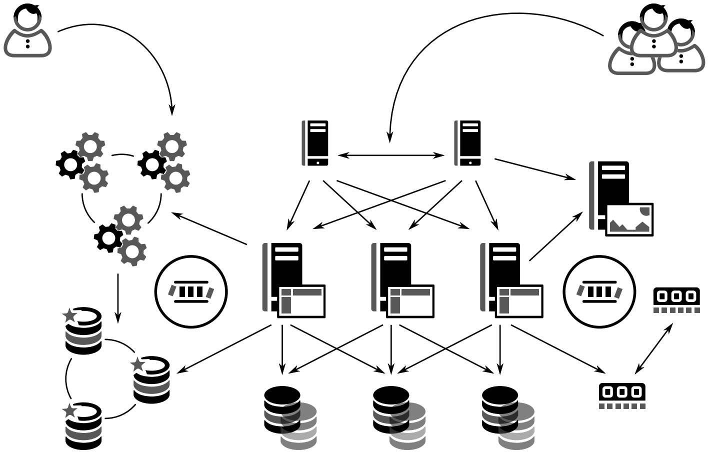
Copyright Qafoo GmbH; All Rights Reserved

## Problem

How to orchestrate?

Copyright Qafoo GmbH; All Rights Reserved

# Evolution



Qafoo
passion for software quality

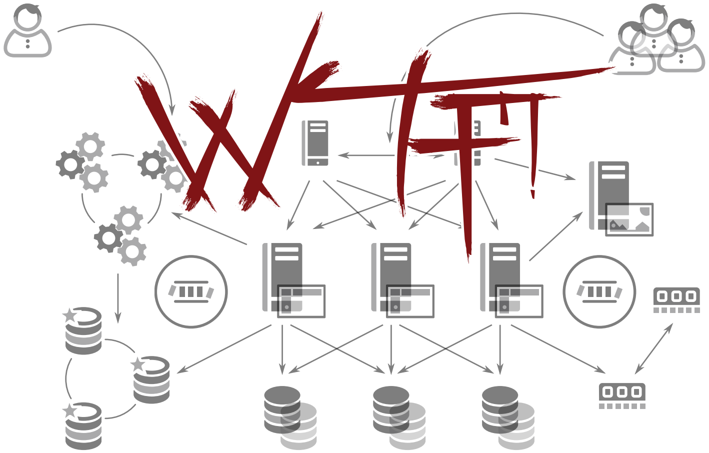
Copyright Qafoo GmbH; All Rights Reserved

## Lessons Learned: Queues

- Queues can ensure data is processed asynchronously
  - Data consistency must be ensured even when pushing into queues
  - Following the data flow of an action can be “tricky”
- Used to distribute data between systems

Qafoo
passion for software quality
Copyright Qafoo GmbH; All Rights Reserved

## Evolution



Qafoo
passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

## Microservices

*Apply **Seperation of Concerns** on service level to allow for seperate teams & technologies per concern.*

- Microservices **can** simplify things:
  - Choose optimal technology stack per team & concern
- Microservices **will** also complicate things:
  - Automated deployment is a must
  - Service orchestration is still a problem
  - Service downtimes and latency must be handled gracefully (Eventual Consistency)
- Big Data$^{TM}$ will stay a problem

# Lessons Learned (subjective)

- Boring technology choices will often work best
  - Just start & stay with LAMP?
- Only bring in shiny new technologies with care
  - There are enough reasons to eventually do that, though

Copyright Qafoo GmbH; All Rights Reserved

## Conclusion

There is no conclusion

Do not jump on every bandwagon – this includes microservices

passion for software quality

Copyright Qafoo GmbH; All Rights Reserved

passion for software quality

THANK YOU

Rent a quality expert
qafoo.com

Copyright Qafoo GmbH; All Rights Reserved