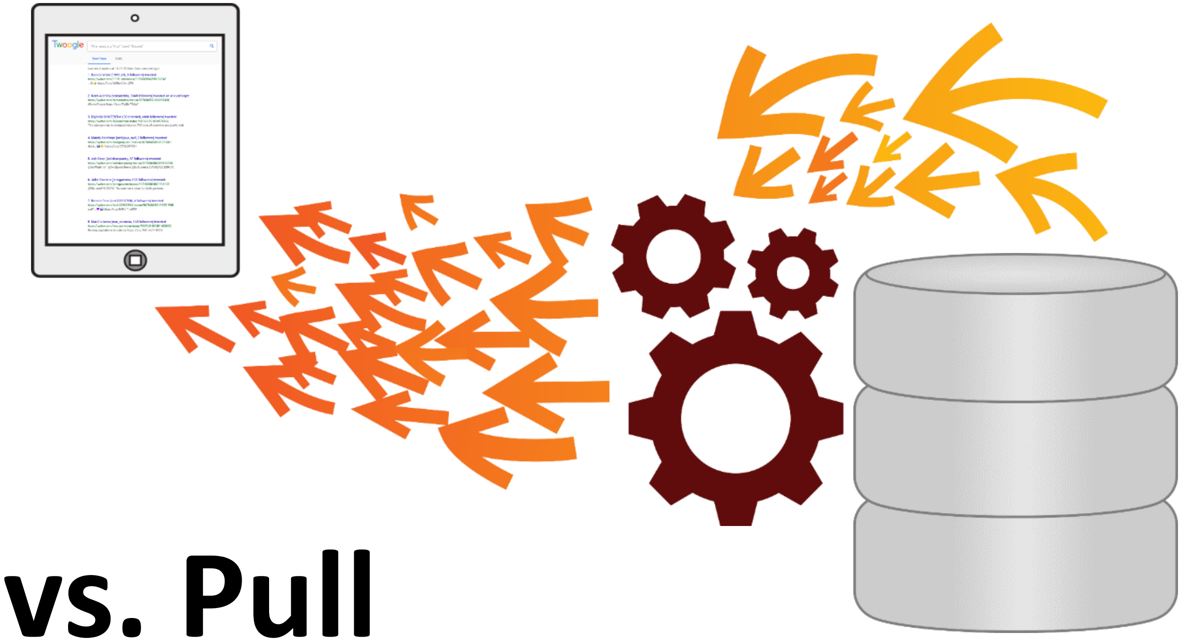


Push vs. Pull

The Future of Real-Time Databases in the Cloud

Wolfram Wingerath
ww@baqend.com

December 10, SCDM 2018, Seattle

UHH Universität Hamburg

BaQend
www.baqend.com

About me

Wolfram Wingerath

PhD Thesis & Research

Distributed Systems Engineer

Research:

- Real-Time Databases
- Stream Processing
- NoSQL & Cloud Databases
- …

Practice:

- Backend-as-a-Service
- Web Caching
- Real-Time Database
- …

UHH Universität Hamburg

BaQend

www.baqend.com

Outline

Push-Based Data Access
Why Real-Time Databases?

Real-Time Databases
System survey

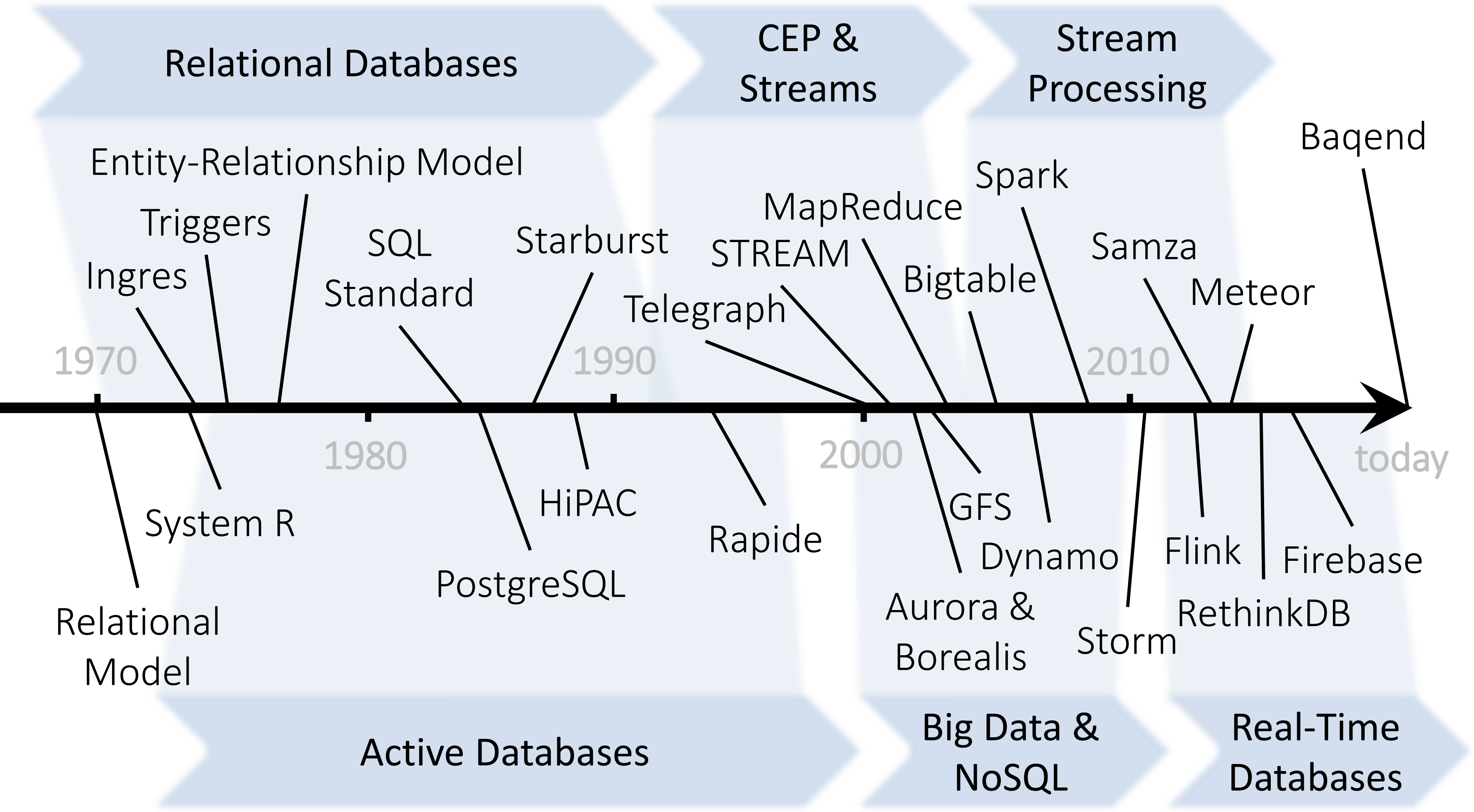
Discussion
What are the bottlenecks?

Future Directions
Scalability & Use Cases

- A Small History Lesson
- The Problem With Traditional Databases
- Real-Time Databases to the Rescue!

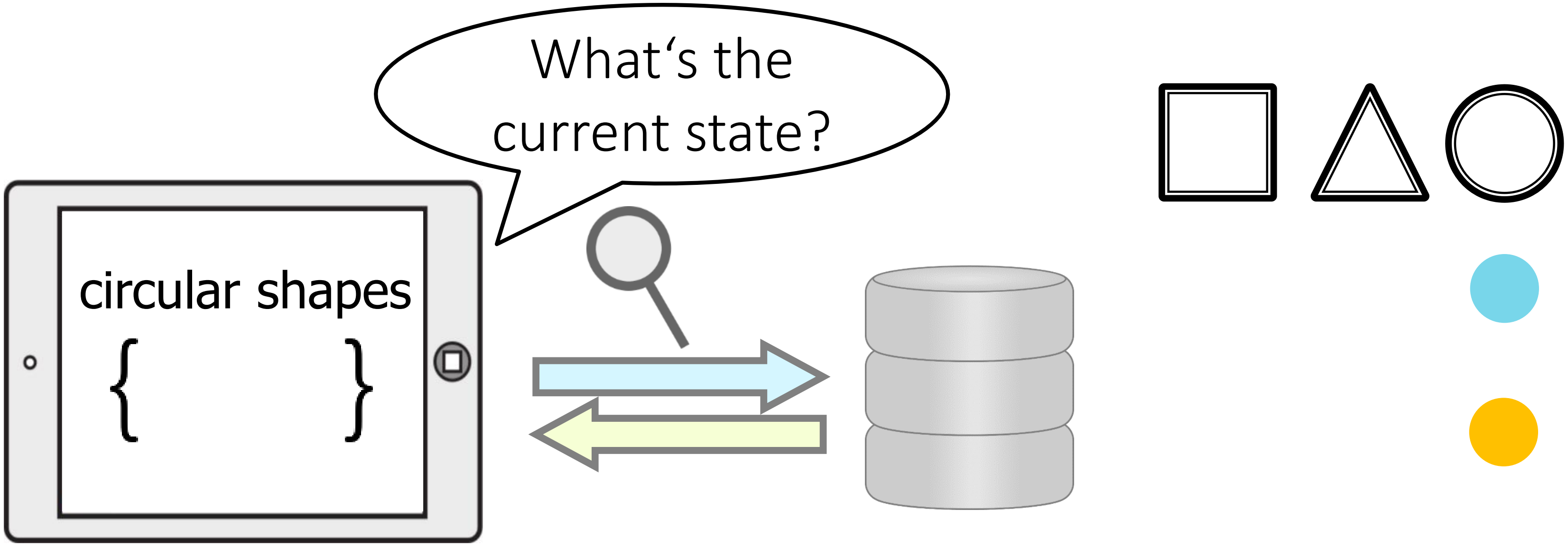
A Short History of Data Management

Hot Topics Through The Ages



Traditional Databases

The Problem: No Request – No Data!



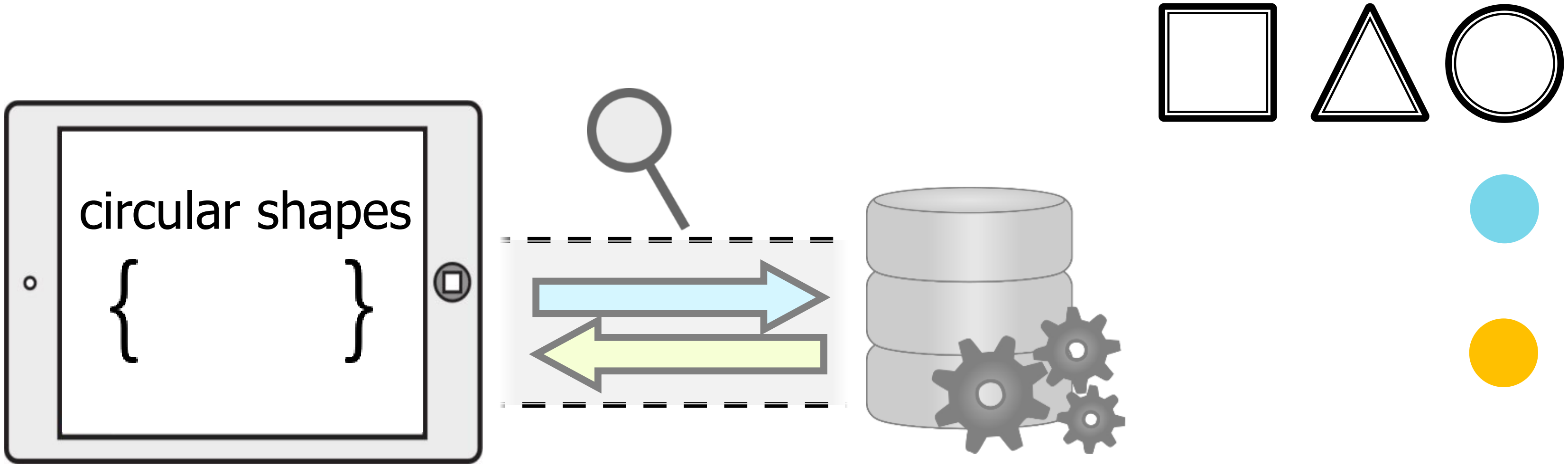
Periodic Polling for query result maintenance:

→ inefficient

→ slow

Real-time Databases

Always Up-to-Date With Database State



Real-Time Queries for query result maintenance:

→ efficient

→ fast

Real-Time Query Maintenance

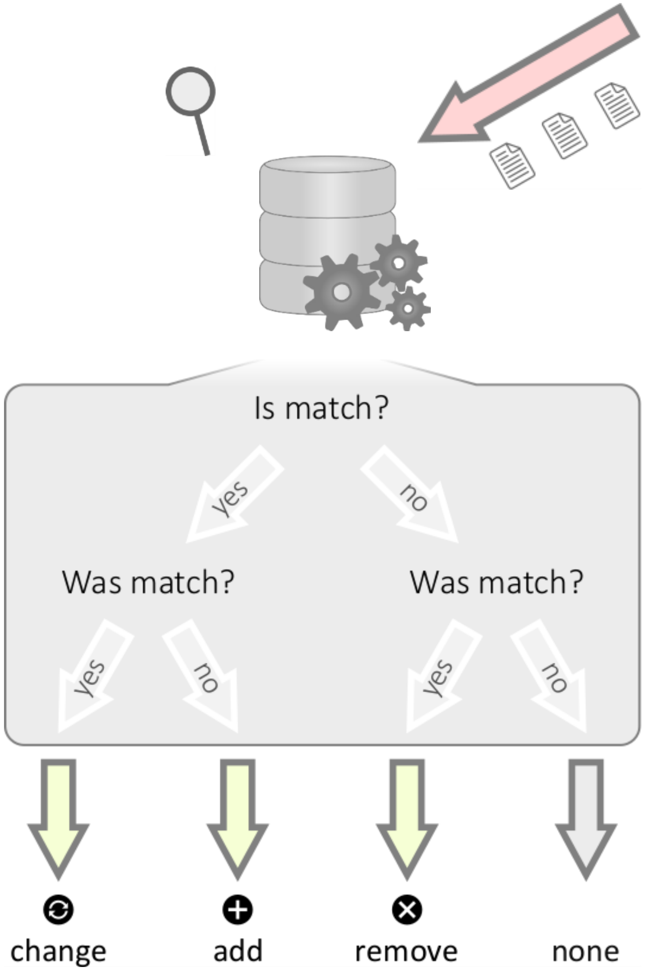
Matching Every Query Against Every Update

→ Potential *bottlenecks:*

- *Number of queries*
- *Write throughput*
- *Query complexity*

Similar processing for:

- Triggers
- ECA rules
- Materialized views



Outline

Push-Based Data Access
Why Real-Time Databases?

Real-Time Databases
System survey

- Meteor
- RethinkDB
- Parse
- Firebase
- Others

Discussion
What are the bottlenecks?

Future Directions
Scalability & Use Cases

Real-Time Databases

Meteor

Overview:

- **JavaScript Framework** for interactive apps and websites
  - **MongoDB** under the hood
  - **Real-time** result updates, full MongoDB expressiveness
- **Open-source**: MIT license
- **Managed service**: Galaxy (Platform-as-a-Service)

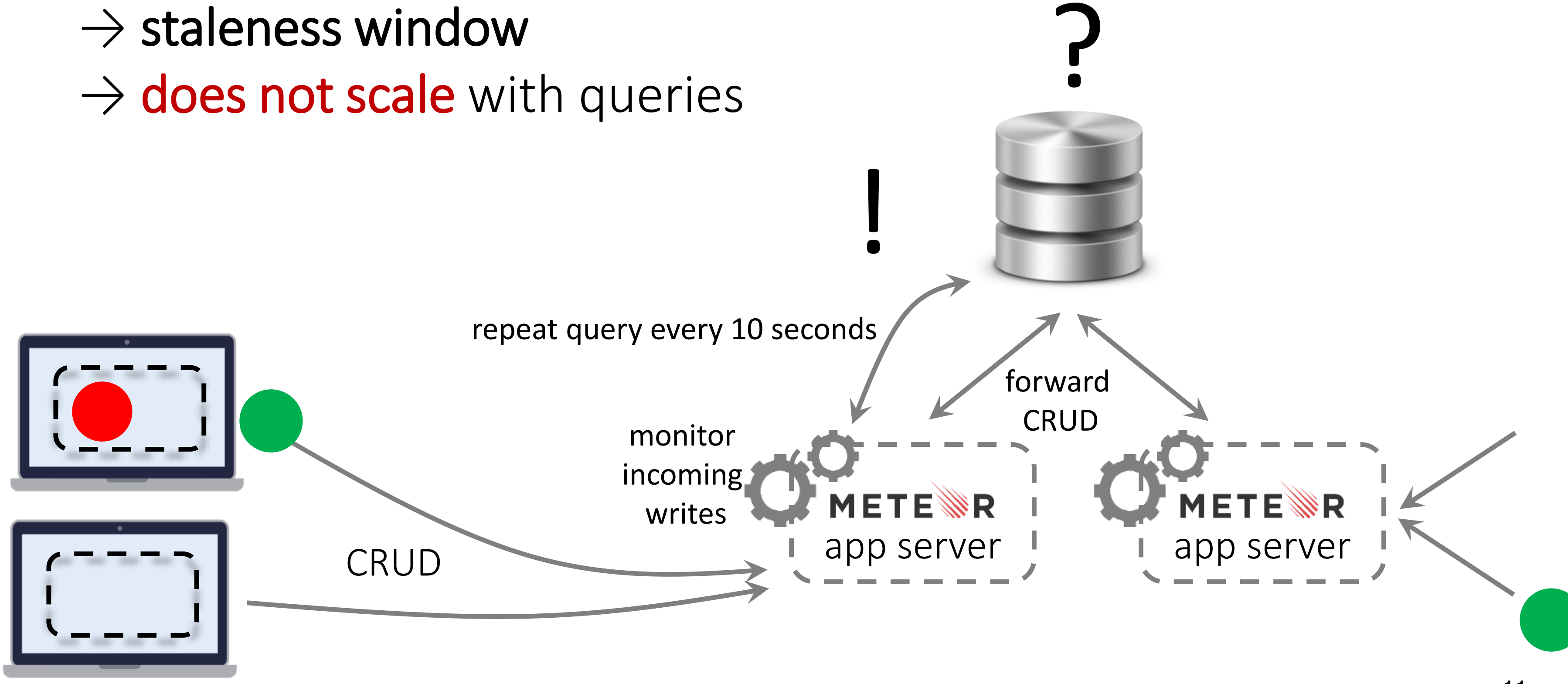
History:

- 2011: *Skybreak* is announced
- 2012: Skybreak is renamed to Meteor
- 2015: Managed hosting service Galaxy is announced

Live Queries

Poll-and-Diff

- **Change monitoring**: app servers detect relevant changes
  → *incomplete* in multi-server deployment
- **Poll-and-diff**: queries are re-executed periodically
  → **staleness window**
  → **does not scale** with queries

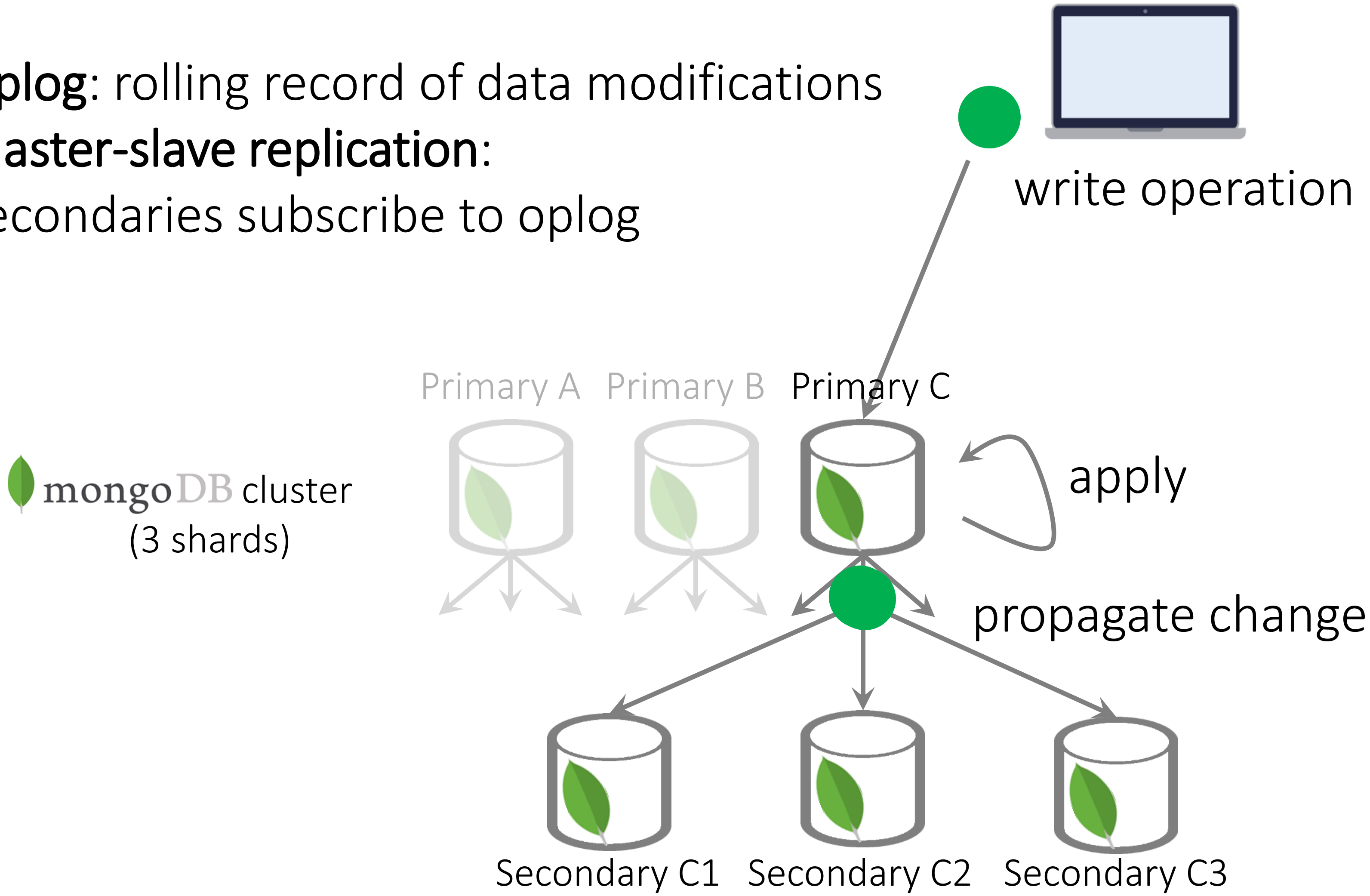


Oplog Tailing

Basics: MongoDB Replication

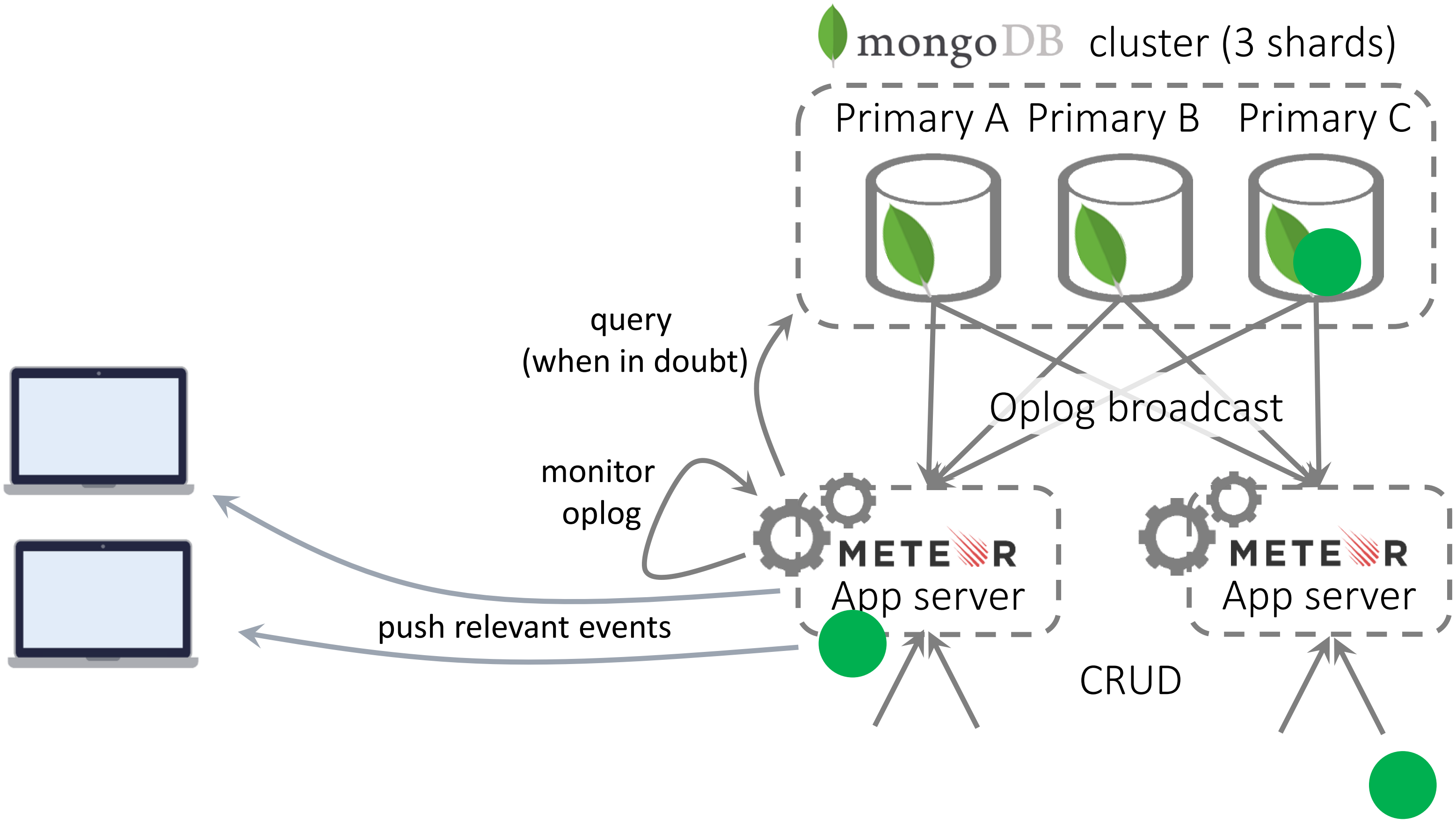
METEOR

- **Oplog**: rolling record of data modifications
- **Master-slave replication**:
  Secondaries subscribe to oplog



Oplog Tailing

Tapping into the Oplog



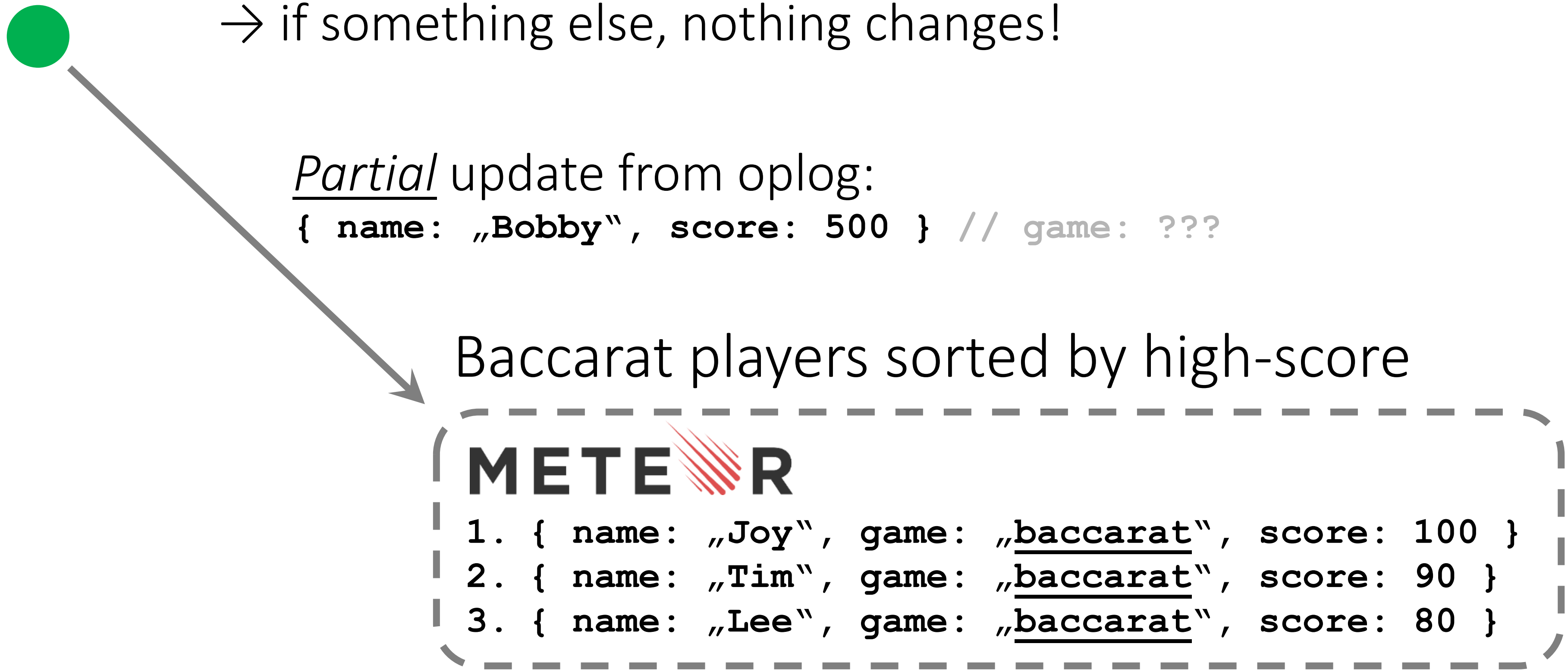
Oplog Tailing

Oplog Info is Incomplete

What game does Bobby play?

→ if baccarat, he takes first place!

→ if something else, nothing changes!



Partial update from oplog:

```
{ name: „Bobby“, score: 500 } // game: ???
```

Baccarat players sorted by high-score

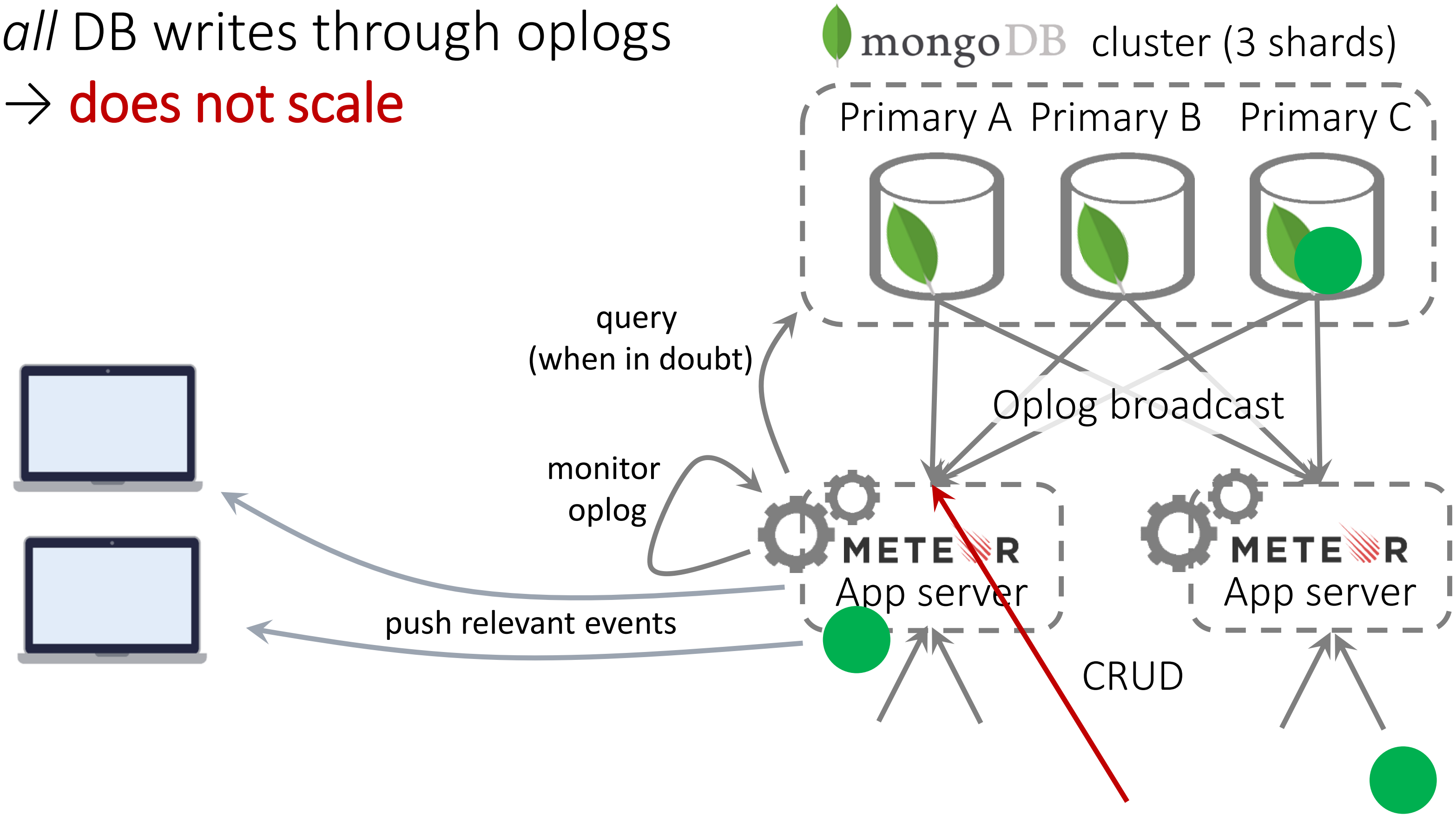
METEOR

```
1. { name: „Joy“, game: „baccarat“, score: 100 }
2. { name: „Tim“, game: „baccarat“, score: 90 }
3. { name: „Lee“, game: „baccarat“, score: 80 }
```

Oplog Tailing

Tapping into the Oplog

- *Every* Meteor server receives *all* DB writes through oplogs → does not scale



Bottleneck!

RethinkDB

Overview:

- **„MongoDB done right“**: comparable queries and data model, but also:
  - **Push-based queries** (filters only)
  - **Joins** (non-streaming)
  - **Strong consistency**: linearizability
- **JavaScript SDK** (*Horizon*): open-source, as managed service
- **Open-source**: Apache 2.0 license

History:

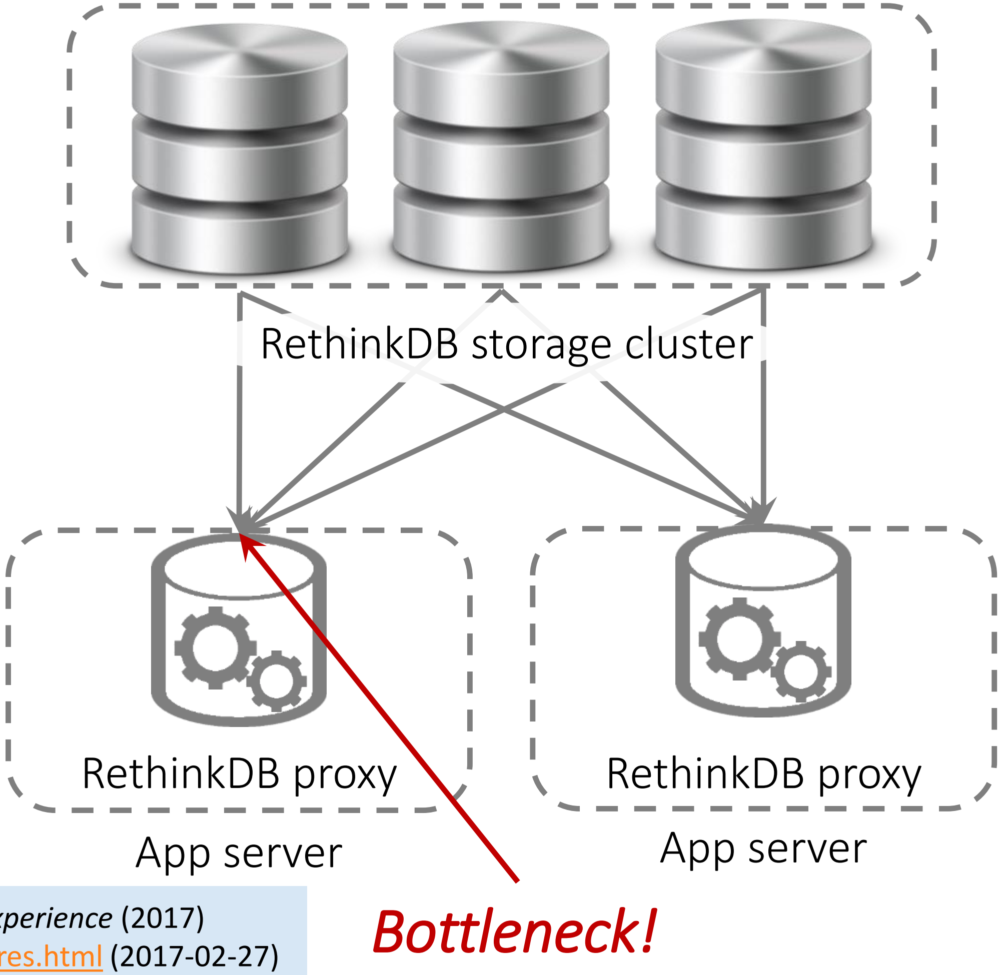
- 2009: RethinkDB is founded
- 2012: RethinkDB is open-sourced under AGPL
- 2016, May: first official release of Horizon (JavaScript SDK)
- 2016, October: RethinkDB announces shutdown
- 2017: RethinkDB is relicensed under Apache 2.0

RethinkDB

Changefeed Architecture

- Range-sharded data
- **RethinkDB proxy**: support node without data
  - Client communication
  - Request routing
  - Real-time query matching

- *Every* proxy receives *all* database writes → does not scale



William Stein, *RethinkDB versus PostgreSQL: my personal experience* (2017) http://blog.sagemath.com/2017/02/09/rethinkdb-vs-postgres.html (2017-02-27)

Daniel Mewes, *Comment on GitHub issue #962: Consider adding more docs on RethinkDB Proxy* (2016) https://github.com/rethinkdb/docs/issues/962 (2017-02-27)

Parse

Parse

Overview:

- **Backend-as-a-Service** for mobile apps
  - **MongoDB:** largest deployment world-wide
  - **Easy development**: great docs, push notifications, authentication, …
  - **Real-time** updates for most MongoDB queries
- **Open-source**: BSD license
- **Managed service**: discontinued

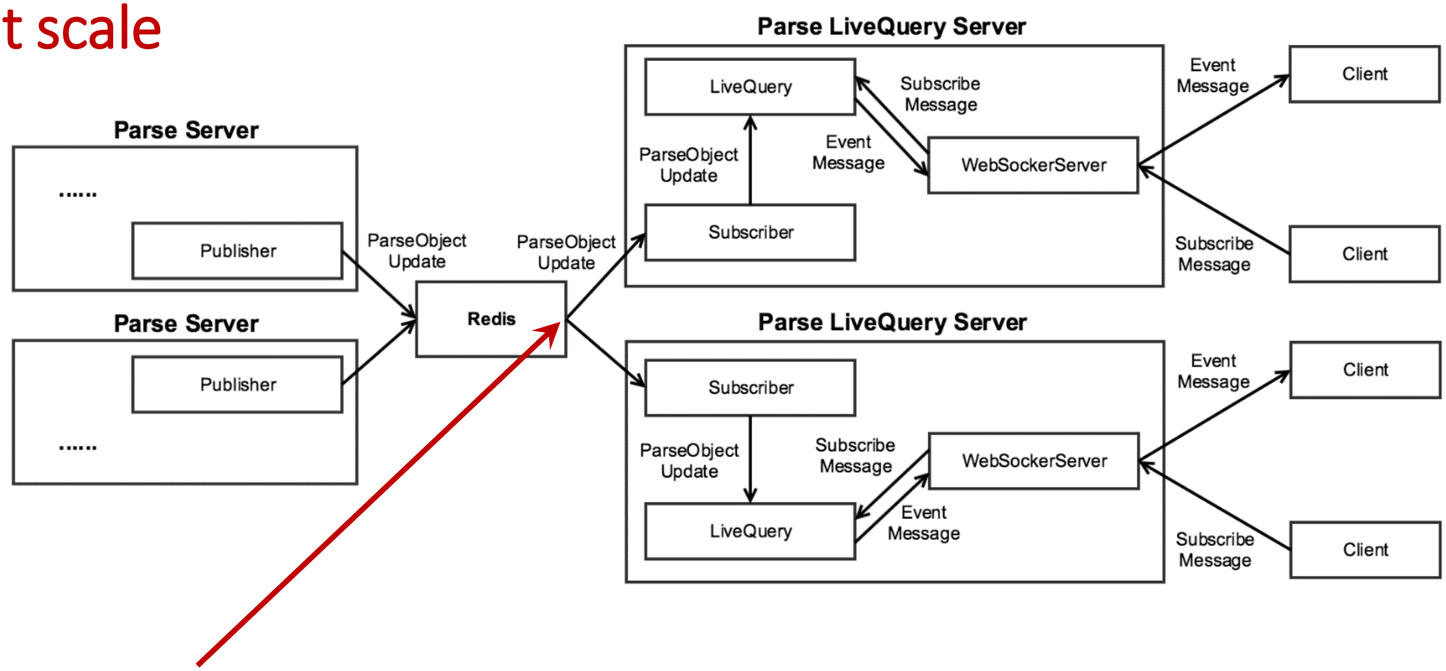
History:

- 2011: Parse is founded
- 2013: Parse is acquired by Facebook
- 2015: more than 500,000 mobile apps reported on Parse
- 2016, January: Parse shutdown is announced
- 2016, March: **Live Queries** are announced
- 2017: Parse shutdown is finalized

Parse

LiveQuery Architecture

- **LiveQuery Server**: no data, real-time query matching
- *Every* LiveQuery Server receives *all* database writes
  → does not scale



Bottleneck!

Illustration taken from:
http://parseplatform.github.io/docs/parse-server/guide/#live-queries (2017-02-22)

Firebase

Overview:

- **Real-time state synchronization** across devices
- **Simplistic data model:** nested hierarchy of lists and objects
- **Simplistic queries**: mostly navigation/filtering
- **Fully managed**, proprietary
- **App SDK** for App development, mobile-first
- **Google services integration**: analytics, hosting, authorization, ...

History:

- 2011: chat service startup Envolve is founded
→ was often used for cross-device state synchronization
→ state synchronization is separated (Firebase)
- 2012: Firebase is founded
- 2013: Firebase is acquired by Google

Firebase

Real-Time State Synchronization

Firebase

- **Tree data model**: application state ~JSON object
- **Subtree synching**: push notifications for specific keys only
  → Flat structure for fine granularity

→ *Limited expressiveness!*

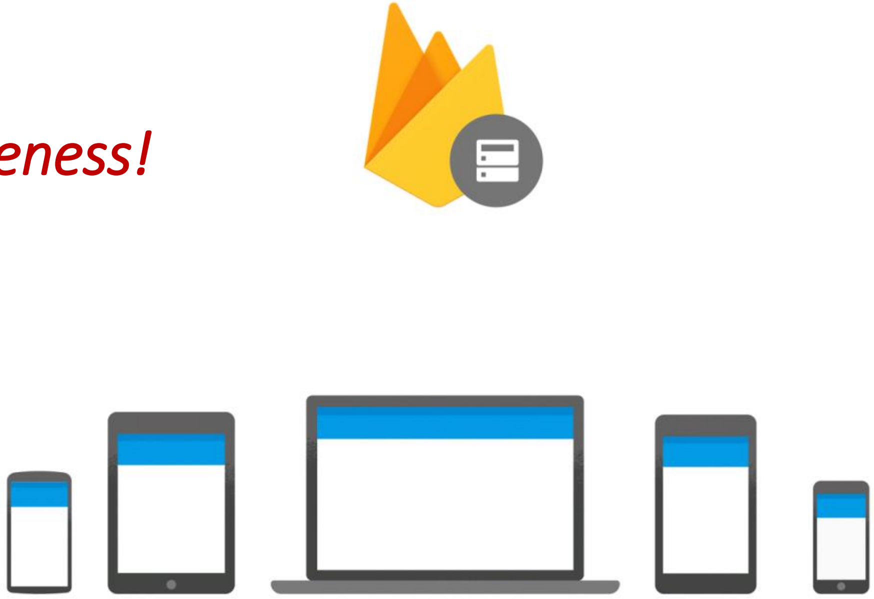


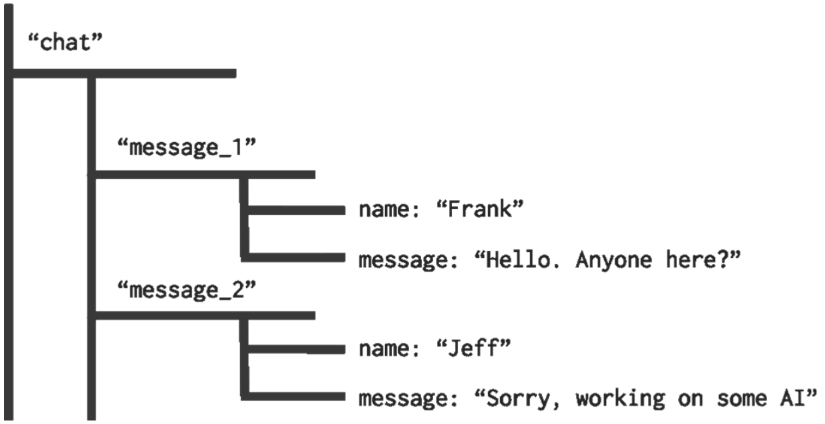
Illustration taken from: Frank van Puffelen, *Have you met the Realtime Database? (2016)*
https://firebase.googleblog.com/2016/07/have-you-met-realtime-database.html (2017-02-27)

Firebase

Query Processing in the Client

- Push notifications for **specific keys** only
  - Order by a **single attribute**
  - Apply a **single filter** on that attribute

- Non-trivial query processing in client
  → does not scale!


"chat"
"message_1"
name: "Frank"
message: "Hello. Anyone here?"
"message_2"
name: "Jeff"
message: "Sorry, working on some AI"

Jacob Wenger, on the Firebase Google Group *(2015)*
https://groups.google.com/forum/#!topic/firebase-talk/d-XjaBVL2Ko (2017-02-27)

Illustration taken from: Frank van Puffelen, *Have you met the Realtime Database? (2016)*
https://firebase.googleblog.com/2016/07/have-you-met-realtime-database.html (2017-02-27)

Firebase

Hard Scaling Limits

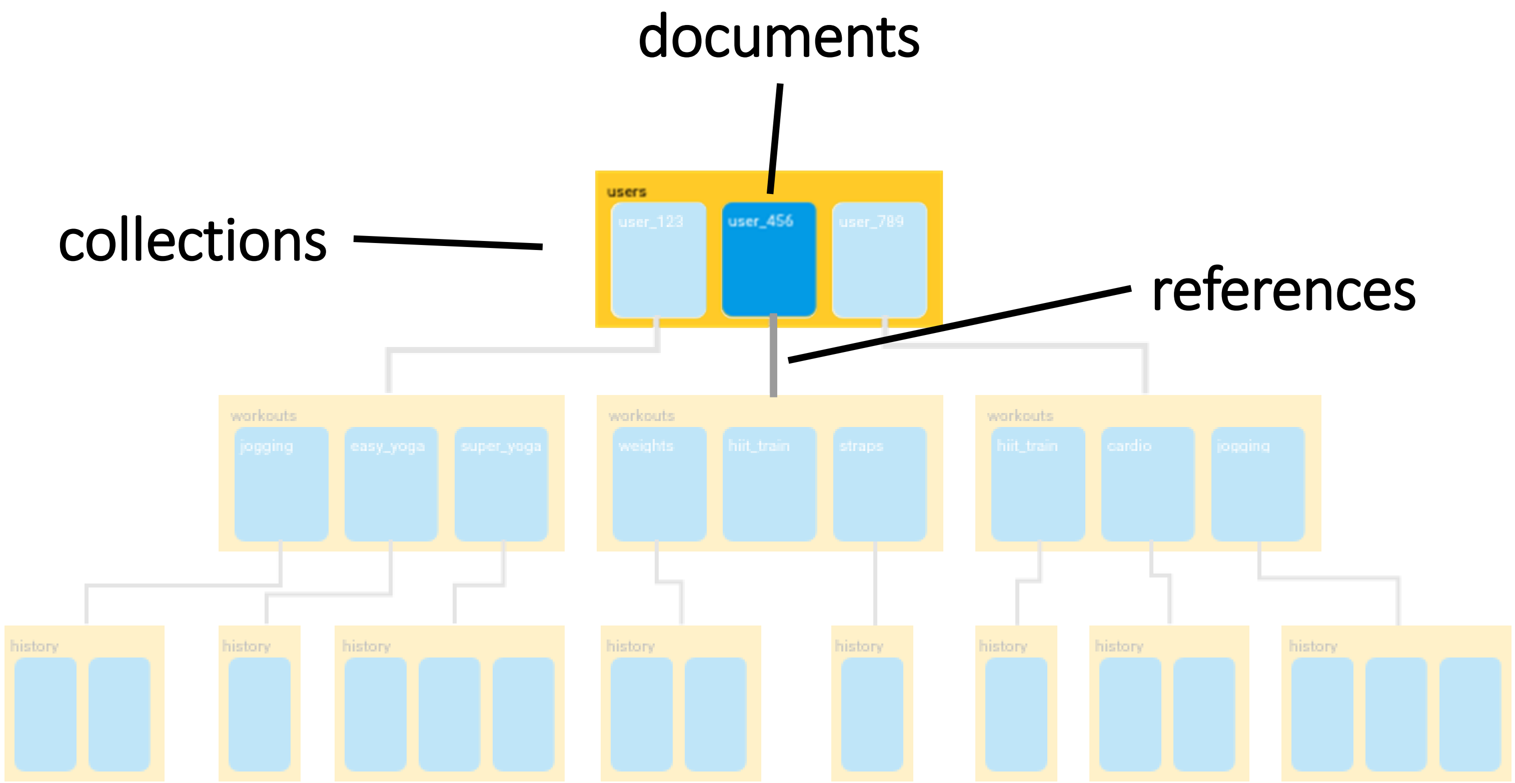
“Scale to around **100,000 concurrent connections** and **1,000 writes/second** in a single database. Scaling beyond that requires sharding your data across multiple databases.”

Bottleneck!

Firebase, *Choose a Database: Cloud Firestore or Realtime Database (2018)*
https://firebase.google.com/docs/database/rtdb-vs-firestore (2018-03-10)

Firebase

Firestore: New Model



documents

collections

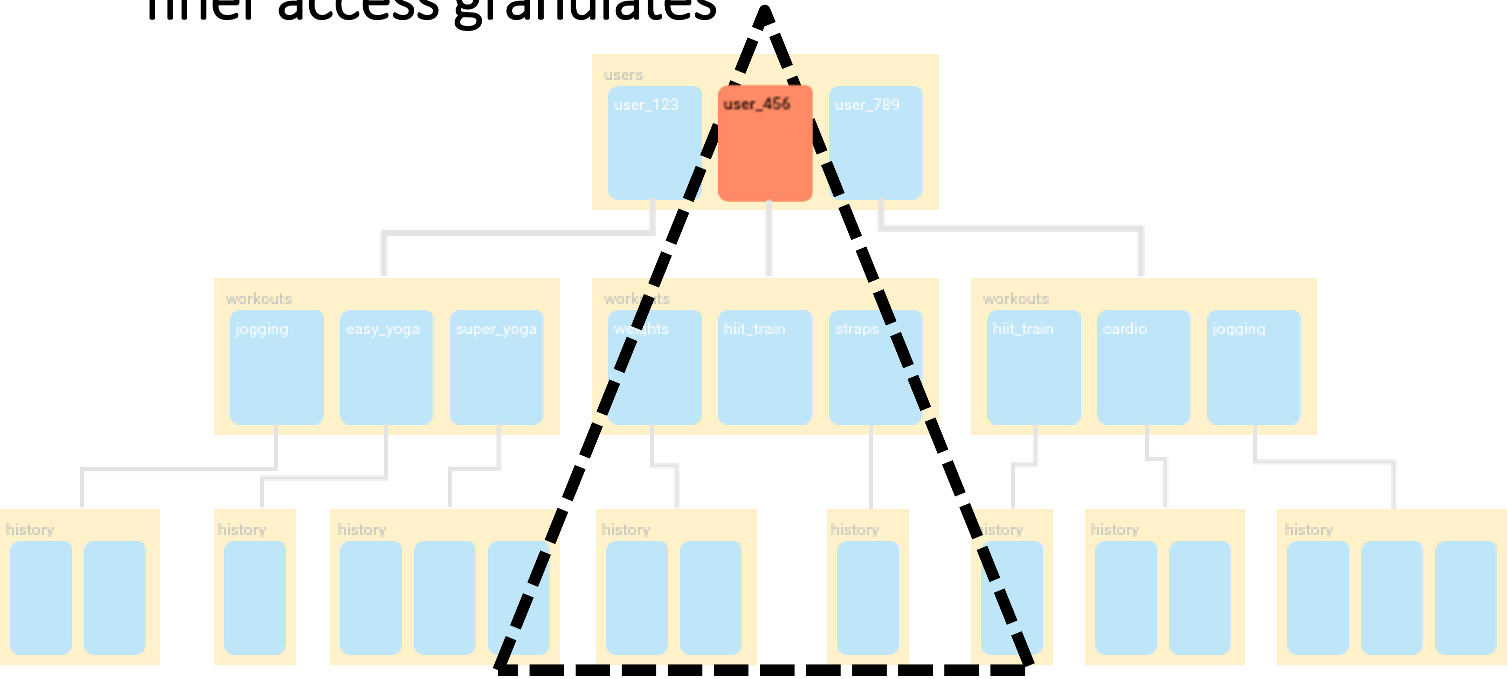
references

Illustration taken from: Todd Kerpelman, *Cloud Firestore for Realtime Database Developers (2017)*
https://firebase.googleblog.com/2017/10/cloud-firestore-for-rtdb-developers.html (2018-03-10)

Firebase

Firestore: New Model

finer access granulates



tree-like structure

Illustration taken from: Todd Kerpelman, *Cloud Firestore for Realtime Database Developers (2017)*
https://firebase.googleblog.com/2017/10/cloud-firestore-for-rtdb-developers.html (2018-03-10)

Firebase

Firestore: Summary

- More specific data selection
- Logical AND for some filter combinations

… But:

- Still **Limited Expressiveness**
  - No logical OR
  - No logical AND for many filter combinations
  - No content-based search (regex, full-text search)
- Still **Limited Write Throughput**:
  - *500 writes/s* per collection
  - *1 writes/s* per document

Firebase, *Firestore: Quotas and Limits (2018)*
https://firebase.google.com/docs/firestore/quotas (2018-03-10)

Honorable Mentions

Other Systems With Real-Time Features

realm

GRAPHCOOL

rapid.io
BETA

Apache CouchDB relax

OrientDB®

mongoDB®

Outline

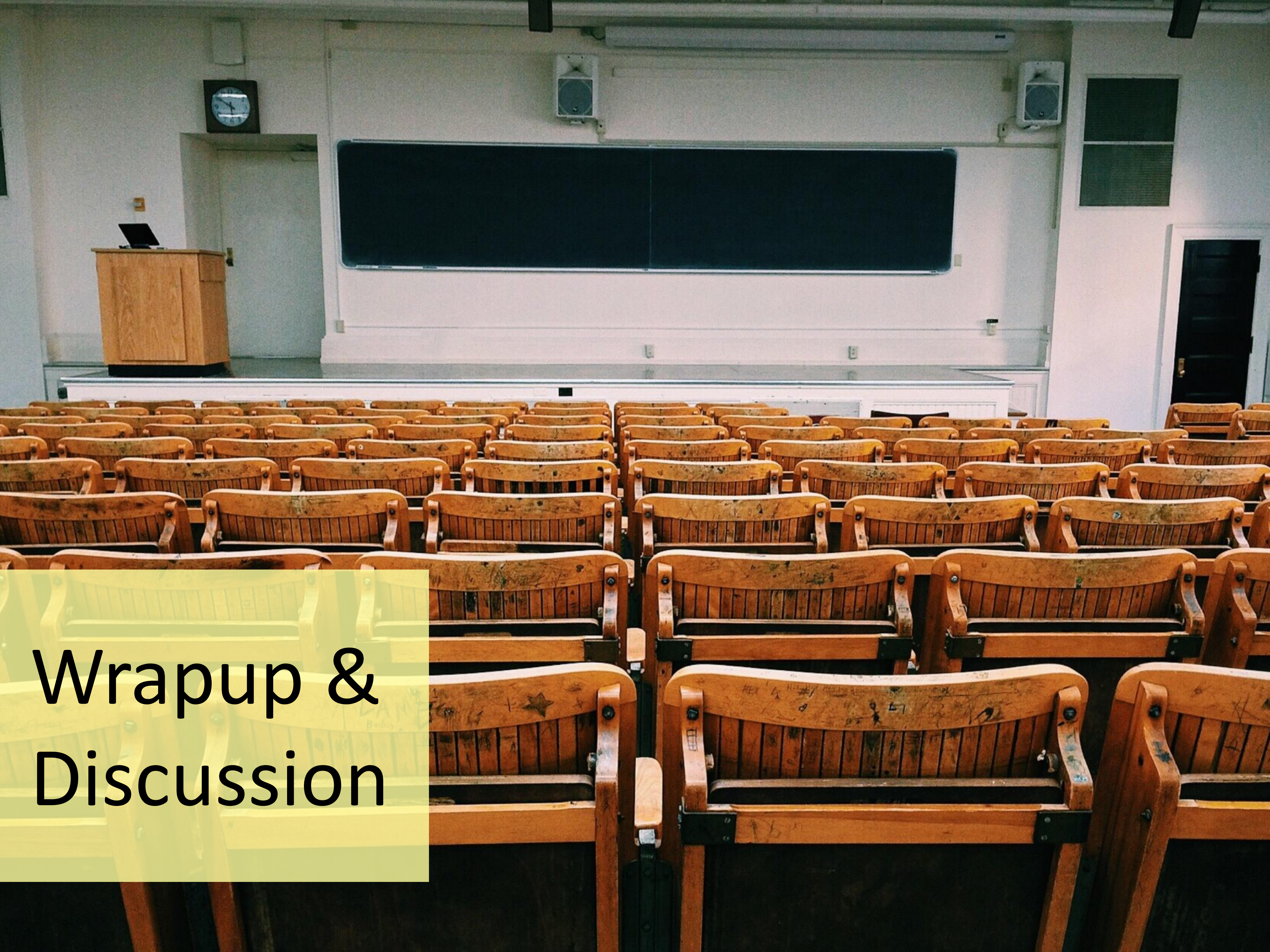
Push-Based Data Access
Why Real-Time Databases?

Real-Time Databases
System survey

Discussion
What are the bottlenecks?

Future Directions
Scalability & Use Cases

- System Classification:
  - Databases
  - Real-Time Databases
  - Stream Management
  - Stream Processing
- Side-by-Side Comparison



Wrapup & Discussion

Data Management Overview

DBMS vs. Real-Time DB vs. Stream Management

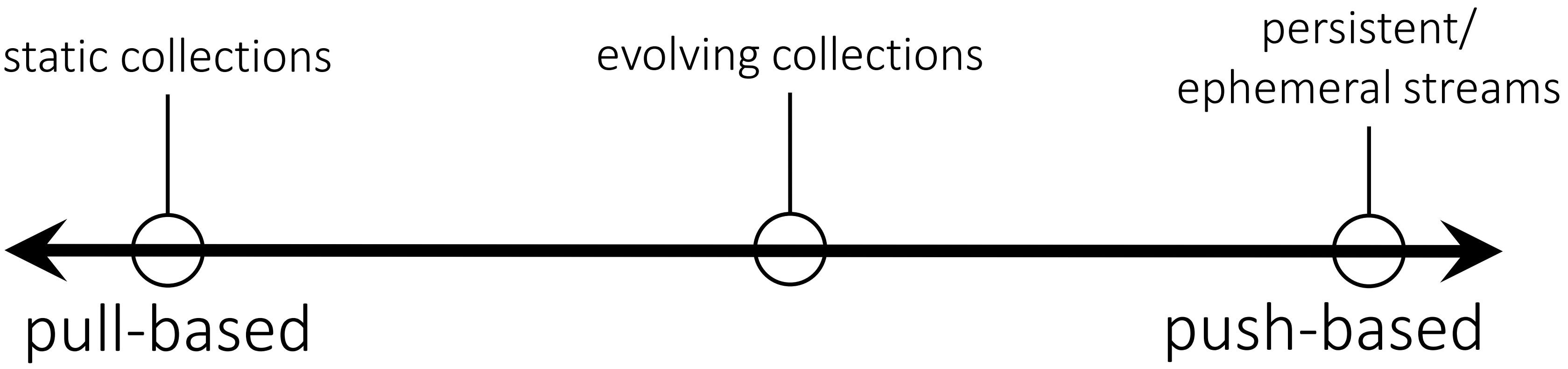
ORACLE
PostgreSQL
MySQL
IBM DB2

PIPELINEDB
EsperTech
sqlstream
influxdata

Database Management

Real-Time Databases

Data Stream Management



static collections

evolving collections

persistent/
ephemeral streams

pull-based

push-based

Real-Time Database Comparison

	METEOR		RethinkDB	Parse	Firebase	Baqend
	Poll-and-Diff	Log Tailing			Unknown	2-D Partitioning
Write Scalability	✓	✗	✗	✗	✗	✓
Read Scalability	✗	✓	✓	✓	? (100k connections)	✓
Composite Filters (AND/OR)	✓	✓	✓	✓	○ (AND In Firestore)	✓
Sorted Queries	✓	✓	✓	✗	○ (single attribute)	✓
Limit	✓	✓	✓	✗	✓	✓
Offset	✓	✓	✗	✗	○ (value-based)	✓
Self-Maintaining Queries	✓	✓	✗	✗	✗	✓
Event Stream Queries	✓	✓	✓	✓	✓	✓

Outline

Push-Based Data Access
Why Real-Time Databases?

Real-Time Databases
System survey

Discussion
What are the bottlenecks?

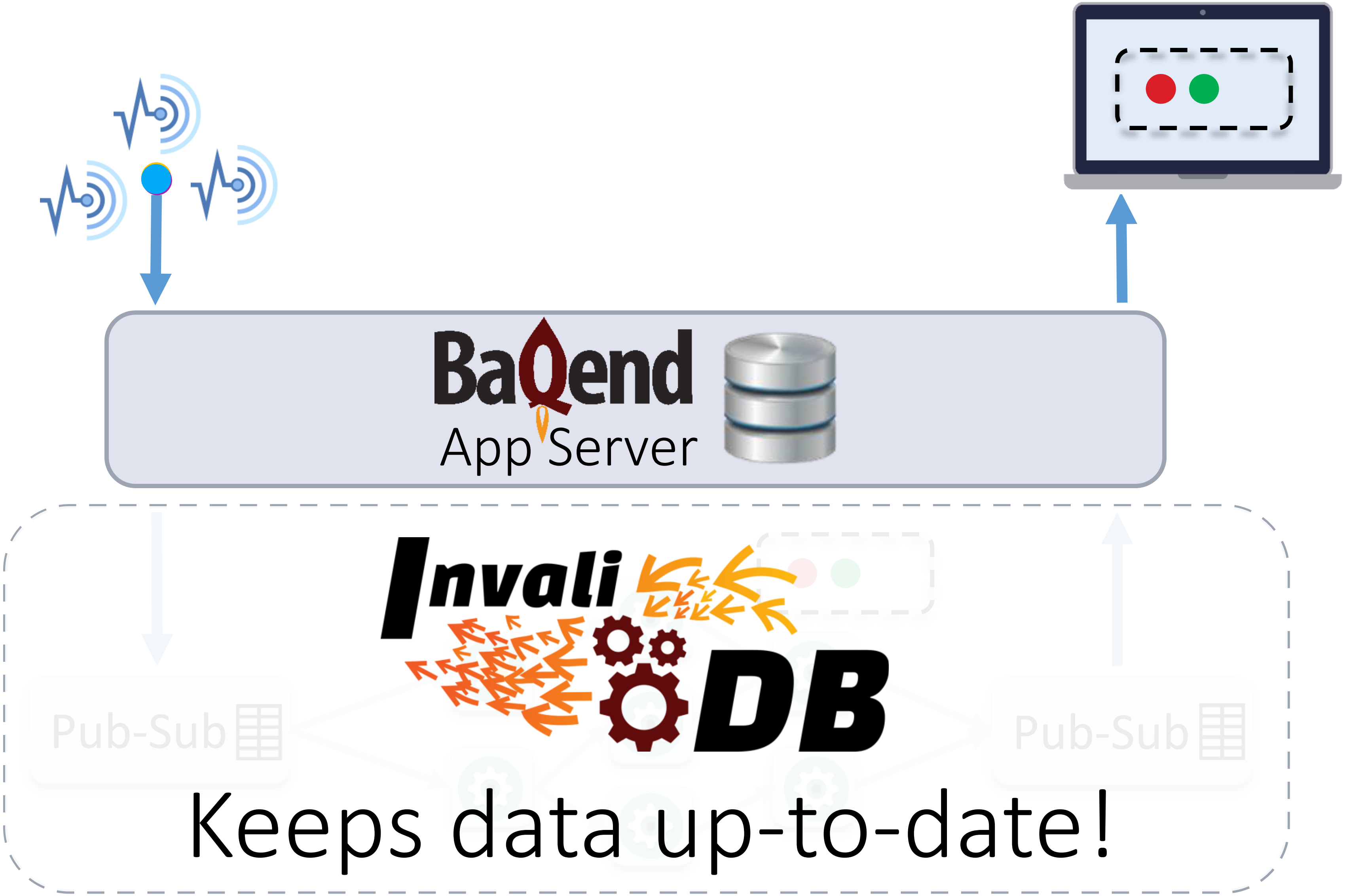
Future Directions
Scalability & Use Cases

- Performance & Scalability
- Query Expressiveness
- Use Cases
  - Real-Time Apps
  - Query Caching
- Summary

Making Real-Time Databases Scale

Baqend Real-Time Queries

Real-Time Decoupled



Keeps data up-to-date!

Baqend Real-Time Queries

Filter Queries: Distributed Query Matching

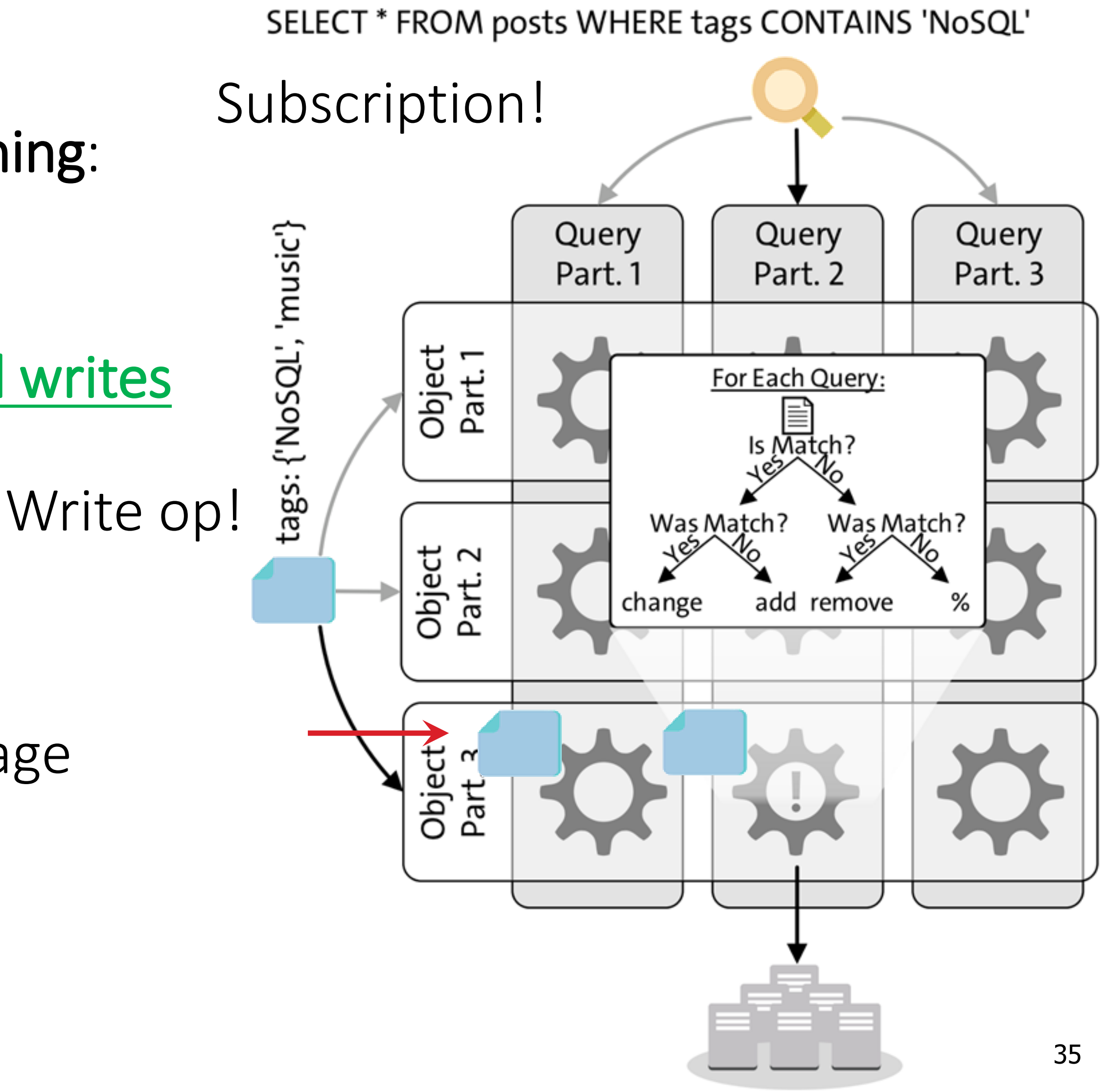
Two-dimensional partitioning:

- *by Query*
- *by Object*

→ scales with queries and writes

Implementation:

- Apache Storm
- Topology in Java
- MongoDB query language
- **Pluggable query engine**

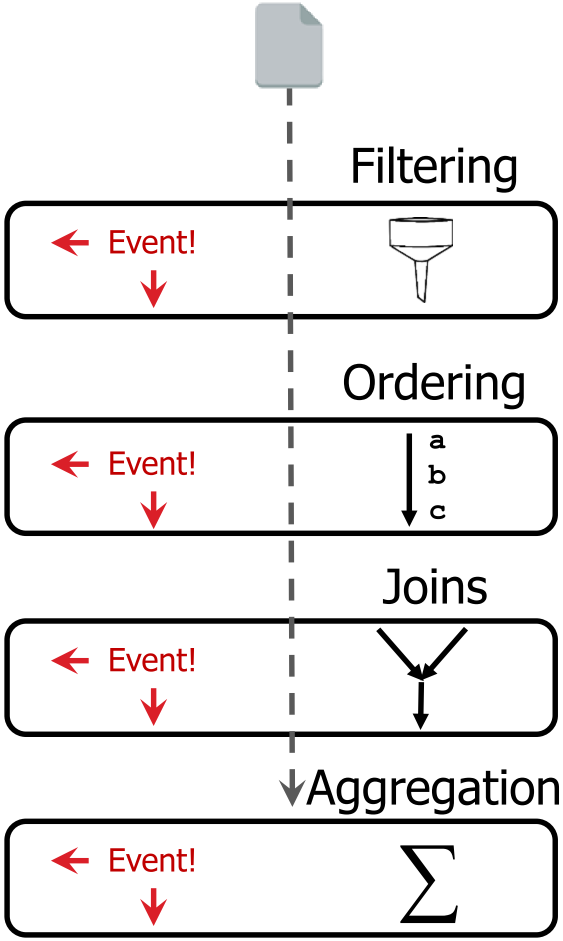


Baqend Real-Time Queries

Staged Real-Time Query Processing

Change notifications go through up to 4 query processing stages:

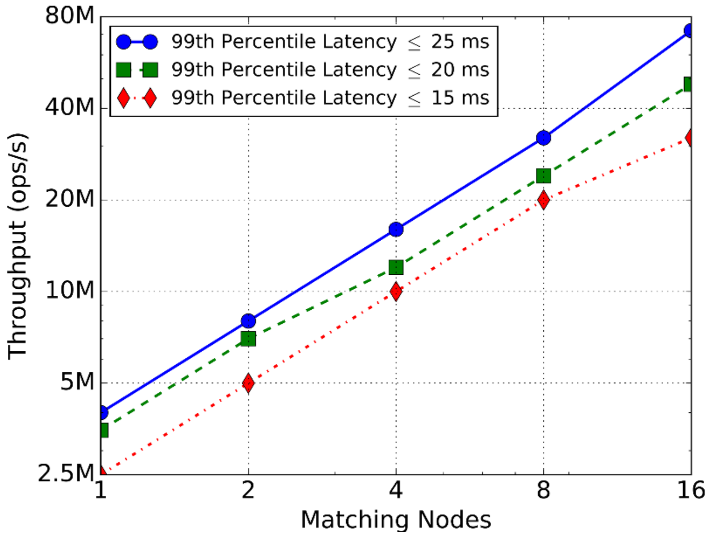
1. **Filter queries**: track matching status
   → *before*- and after-images
2. **Sorted queries**: maintain result order
3. **Joins**: combine maintained results
4. **Aggregations**: maintain aggregations



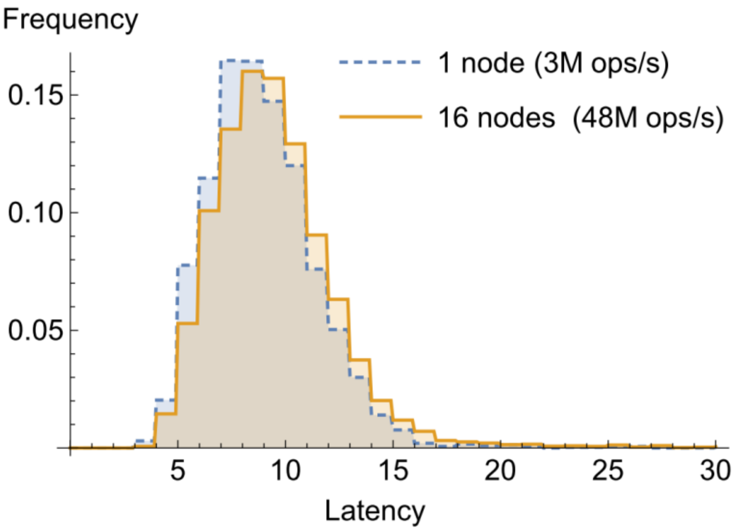
Baqend Real-Time Queries

Low Latency + Linear Scalability

Linear Scalability



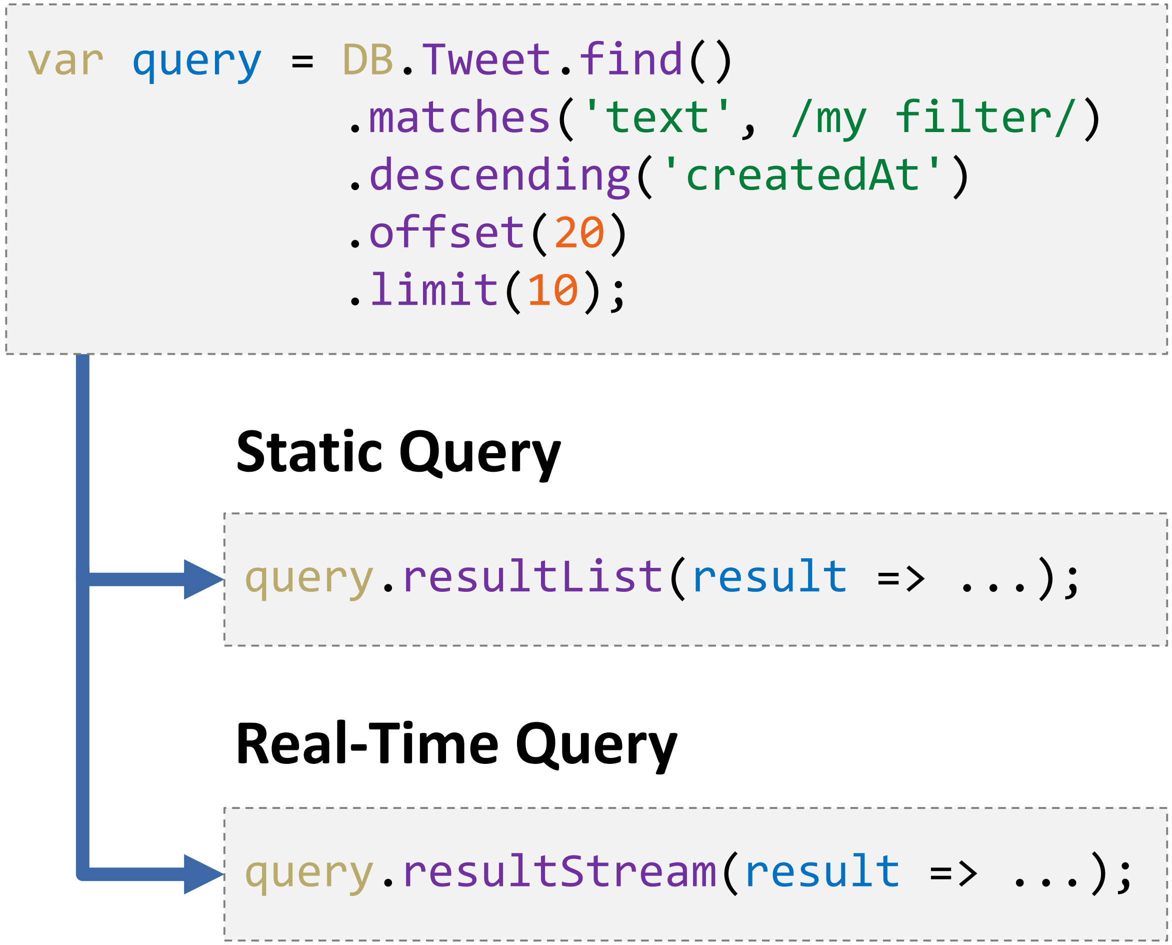
Stable Latency Distribution



Quaestor: Query Web Caching for Database-as-a-Service Providers
VLDB ‘17

Programming Real-Time Queries

JavaScript API



```
var query = DB.Tweet.find()
                .matches('text', /my filter/)
                .descending('createdAt')
                .offset(20)
                .limit(10);
```

Static Query

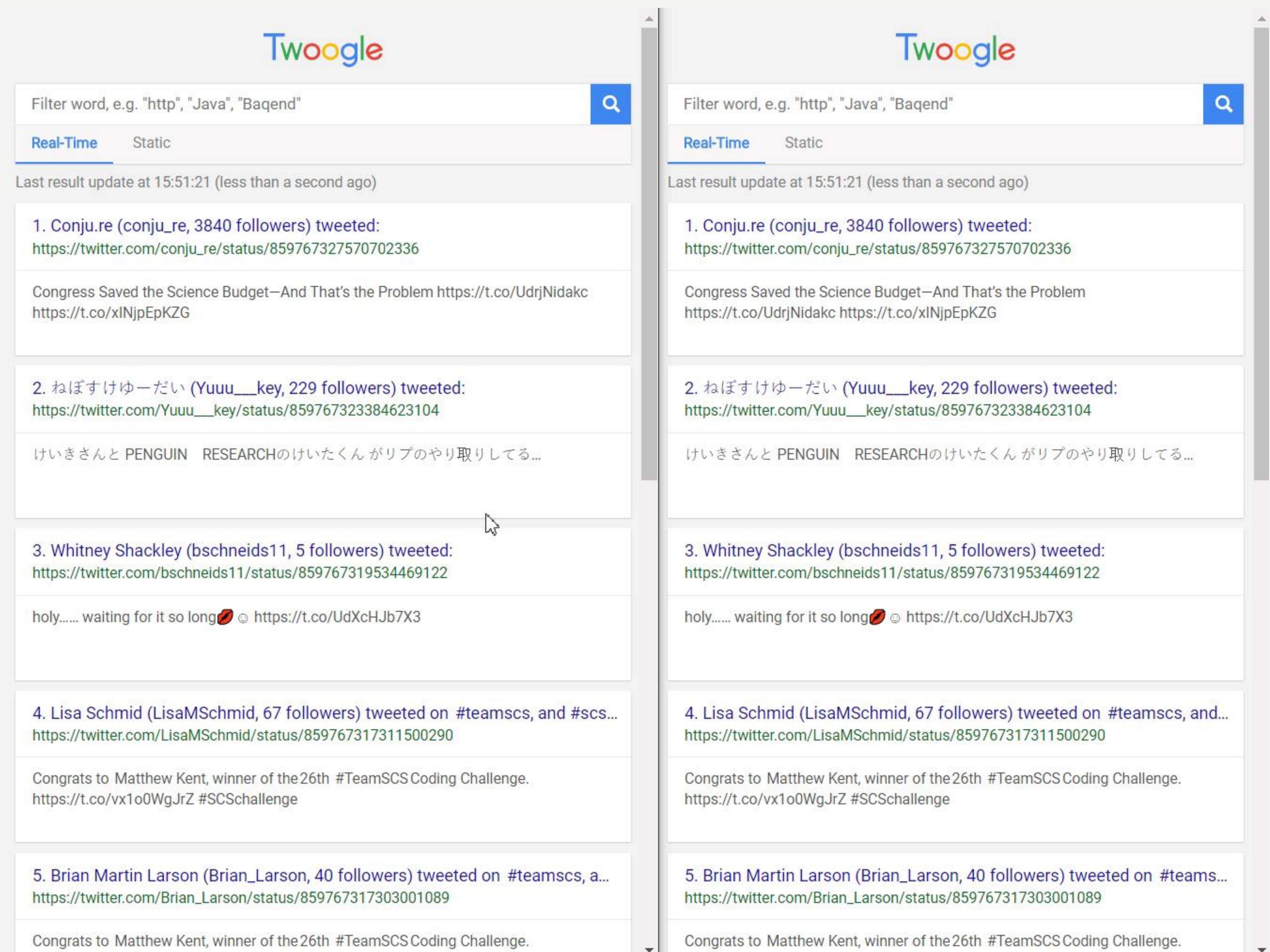
```
query.resultList(result => ...);
```

Google

Real-Time Query

```
query.resultStream(result => ...);
```

Twoogle



Last result update at 15:51:21 (less than a second ago)

1. Conju.re (conju_re, 3840 followers) tweeted:
https://twitter.com/conju_re/status/859767327570702336

Congress Saved the Science Budget—And That's the Problem https://t.co/UdrjNidakc https://t.co/xlNjpEpKZG

2. ねぼすけゆーだい (Yuuu__key, 229 followers) tweeted:
https://twitter.com/Yuuu__key/status/859767323384623104

けいきさんと PENGUIN　RESEARCHのけいたくん がリプのやり取りしてる...

3. Whitney Shackley (bschneids11, 5 followers) tweeted:
https://twitter.com/bschneids11/status/859767319534469122

holy...... waiting for it so long💋☺ https://t.co/UdXcHJb7X3

4. Lisa Schmid (LisaMSchmid, 67 followers) tweeted on #teamscs, and #scs...
https://twitter.com/LisaMSchmid/status/859767317311500290

Congrats to Matthew Kent, winner of the 26th #TeamSCS Coding Challenge. https://t.co/vx1o0WgJrZ #SCSchallenge

5. Brian Martin Larson (Brian_Larson, 40 followers) tweeted on #teamscs, a...
https://twitter.com/Brian_Larson/status/859767317303001089

Congrats to Matthew Kent, winner of the 26th #TeamSCS Coding Challenge.

Last result update at 15:51:21 (less than a second ago)

1. Conju.re (conju_re, 3840 followers) tweeted:
https://twitter.com/conju_re/status/859767327570702336

Congress Saved the Science Budget—And That's the Problem https://t.co/UdrjNidakc https://t.co/xlNjpEpKZG

2. ねぼすけゆーだい (Yuuu__key, 229 followers) tweeted:
https://twitter.com/Yuuu__key/status/859767323384623104

けいきさんと PENGUIN　RESEARCHのけいたくん がリプのやり取りしてる...

3. Whitney Shackley (bschneids11, 5 followers) tweeted:
https://twitter.com/bschneids11/status/859767319534469122

holy...... waiting for it so long💋☺ https://t.co/UdXcHJb7X3

4. Lisa Schmid (LisaMSchmid, 67 followers) tweeted on #teamscs, and...
https://twitter.com/LisaMSchmid/status/859767317311500290

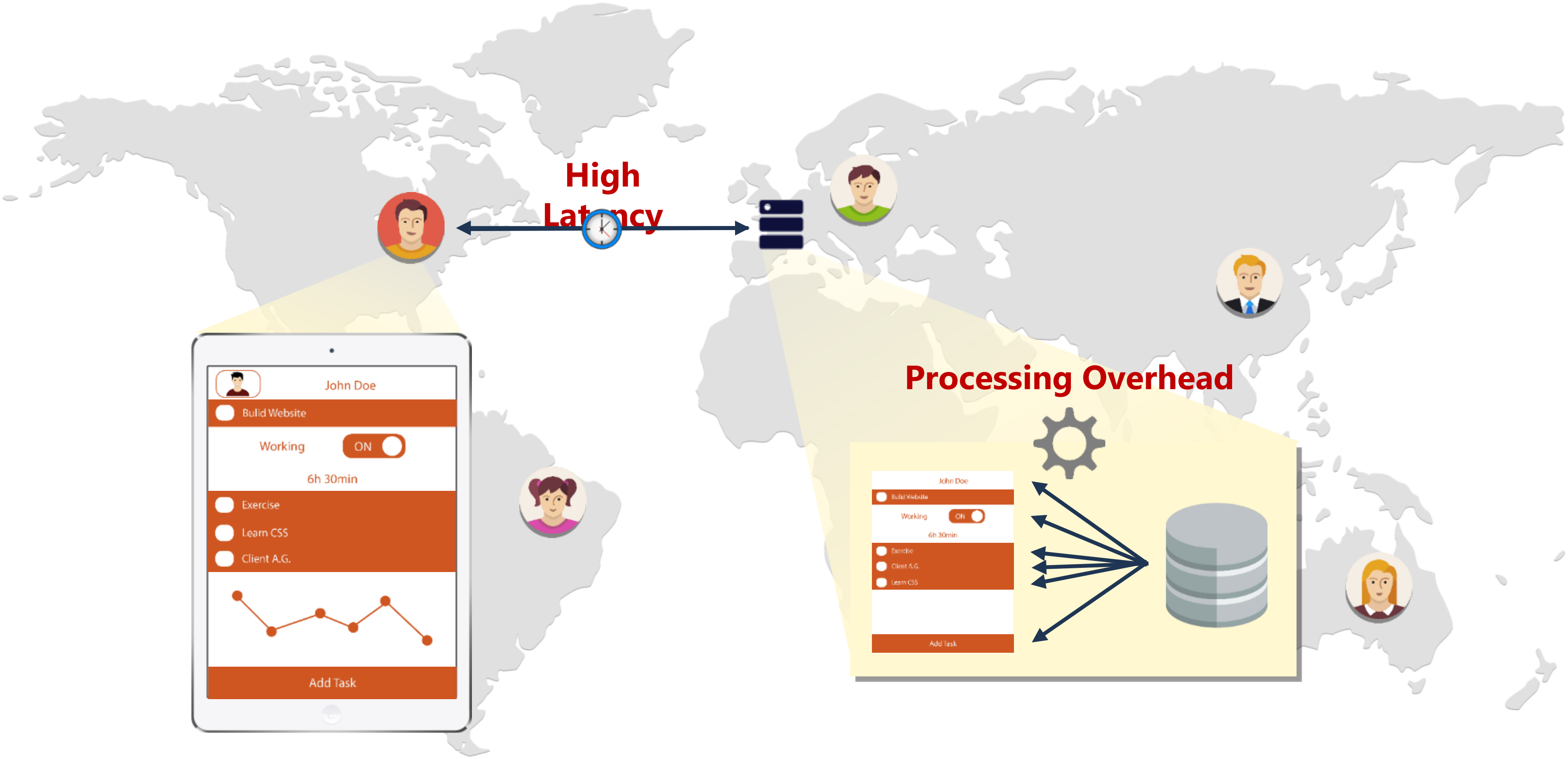
Congrats to Matthew Kent, winner of the 26th #TeamSCS Coding Challenge. https://t.co/vx1o0WgJrZ #SCSchallenge

5. Brian Martin Larson (Brian_Larson, 40 followers) tweeted on #teams...
https://twitter.com/Brian_Larson/status/859767317303001089

Congrats to Matthew Kent, winner of the 26th #TeamSCS Coding Challenge.

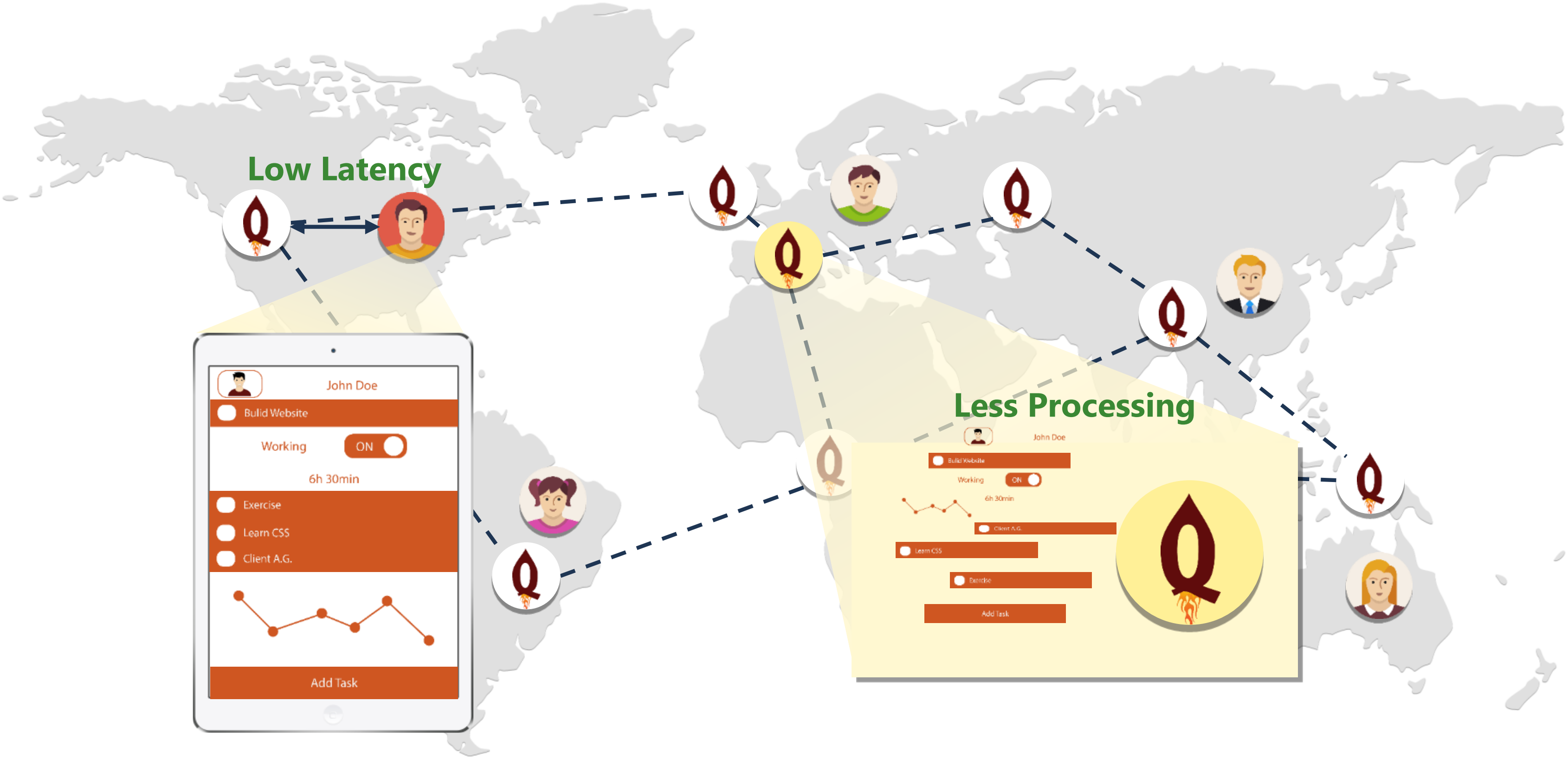
Problem: Slow Websites

Two Bottlenecks: Latency and Processing



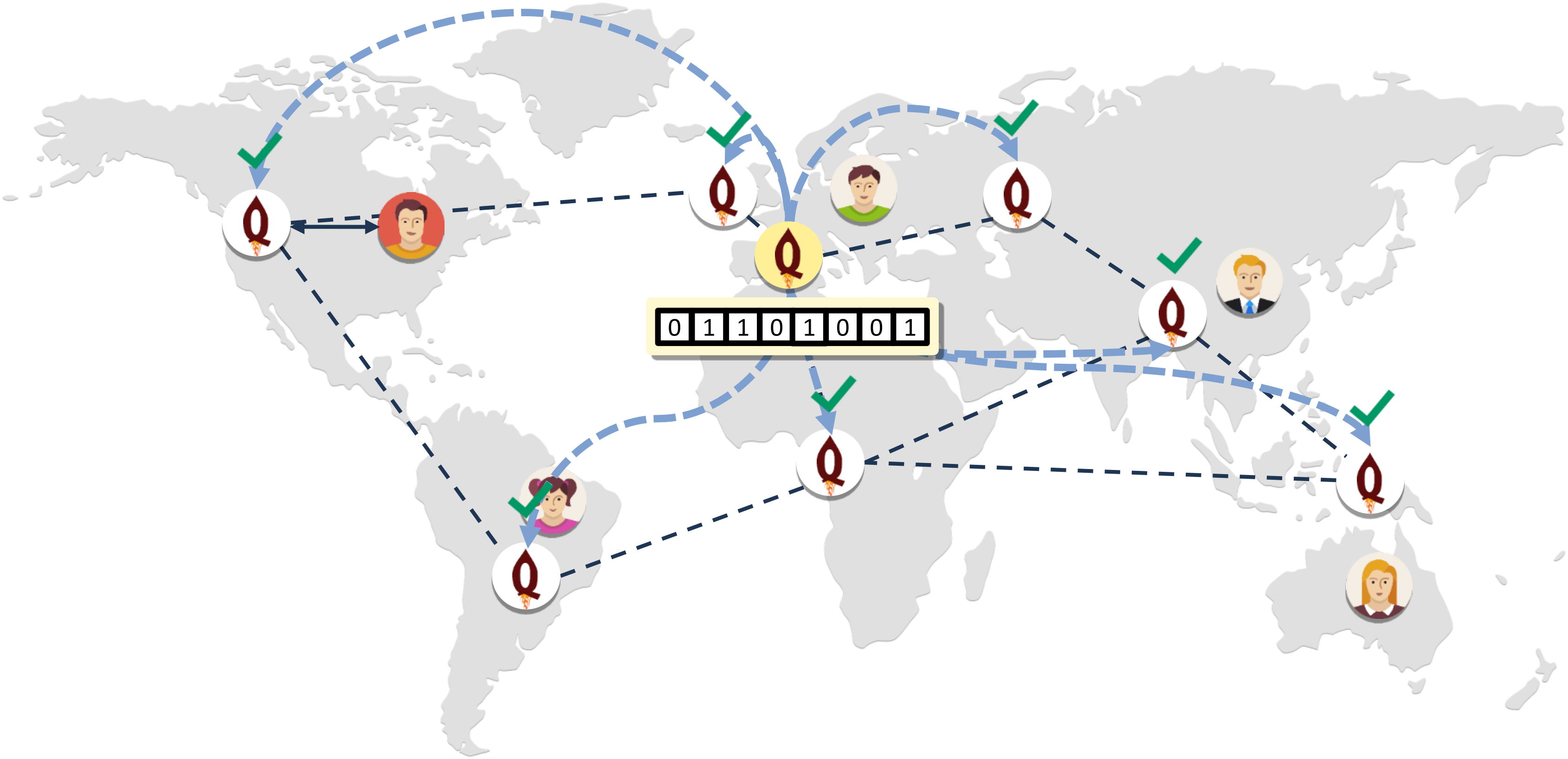
Solution: Global Caching

Fresh Data From Distributed Web Caches



New Caching Algorithms

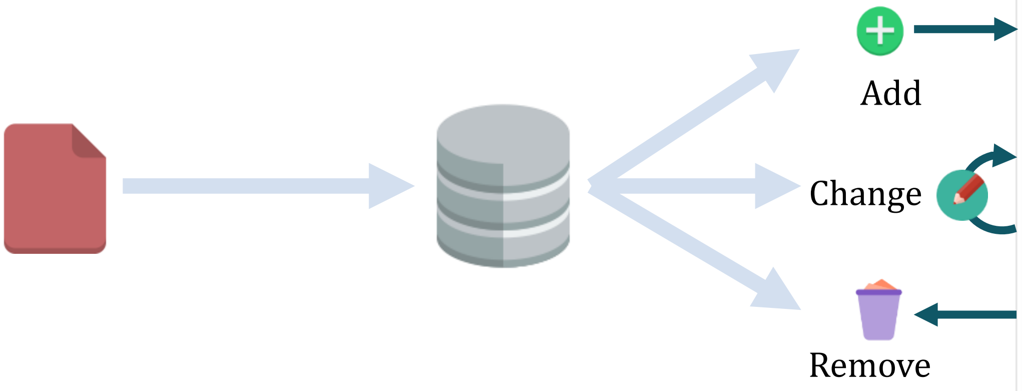
Solve Consistency Problem

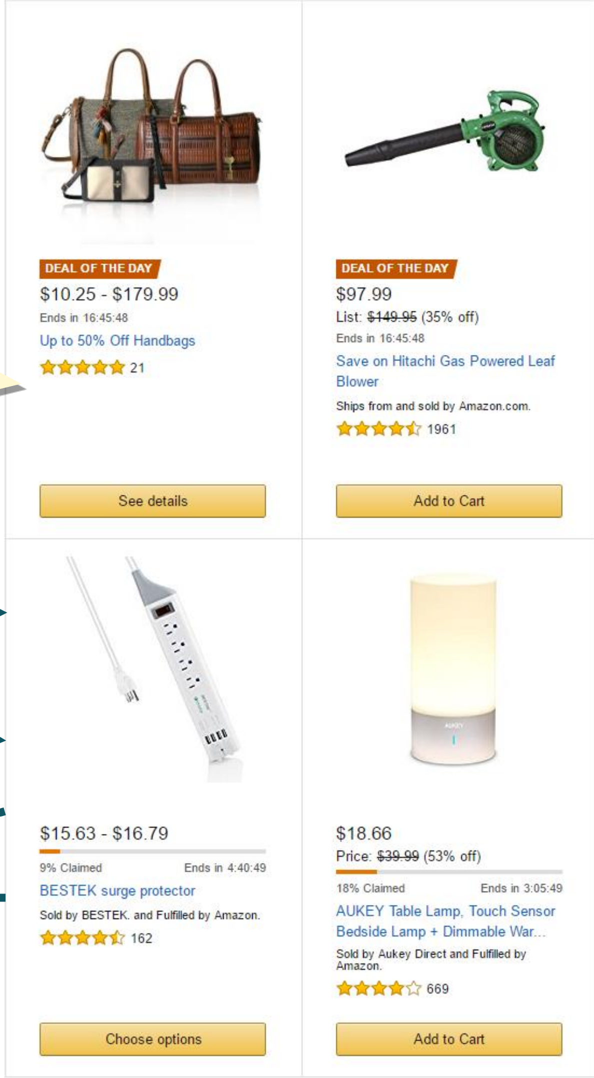


InvaliDB

Invalidating DB Queries

How to **detect changes to query results**:
„Give me the most popular products that are in stock.“





DEAL OF THE DAY
$10.25 - $179.99
Ends in 16:45:48
Up to 50% Off Handbags
★★★★★ 21

DEAL OF THE DAY
$97.99
List: ~~$149.95~~ (35% off)
Ends in 16:45:48
Save on Hitachi Gas Powered Leaf Blower
Ships from and sold by Amazon.com.
★★★★☆ 1961

$15.63 - $16.79
9% Claimed Ends in 4:40:49
BESTEK surge protector
Sold by BESTEK. and Fulfilled by Amazon.
★★★★☆ 162

$18.66
Price: ~~$39.99~~ (53% off)
18% Claimed Ends in 3:05:49
AUKEY Table Lamp, Touch Sensor Bedside Lamp + Dimmable War...
Sold by Aukey Direct and Fulfilled by Amazon.
★★★★☆ 669

Summary

Real-Time Databases: Major challenges

Scalability:

- Handle increasing throughput
- Handle additional queries

Expressiveness:

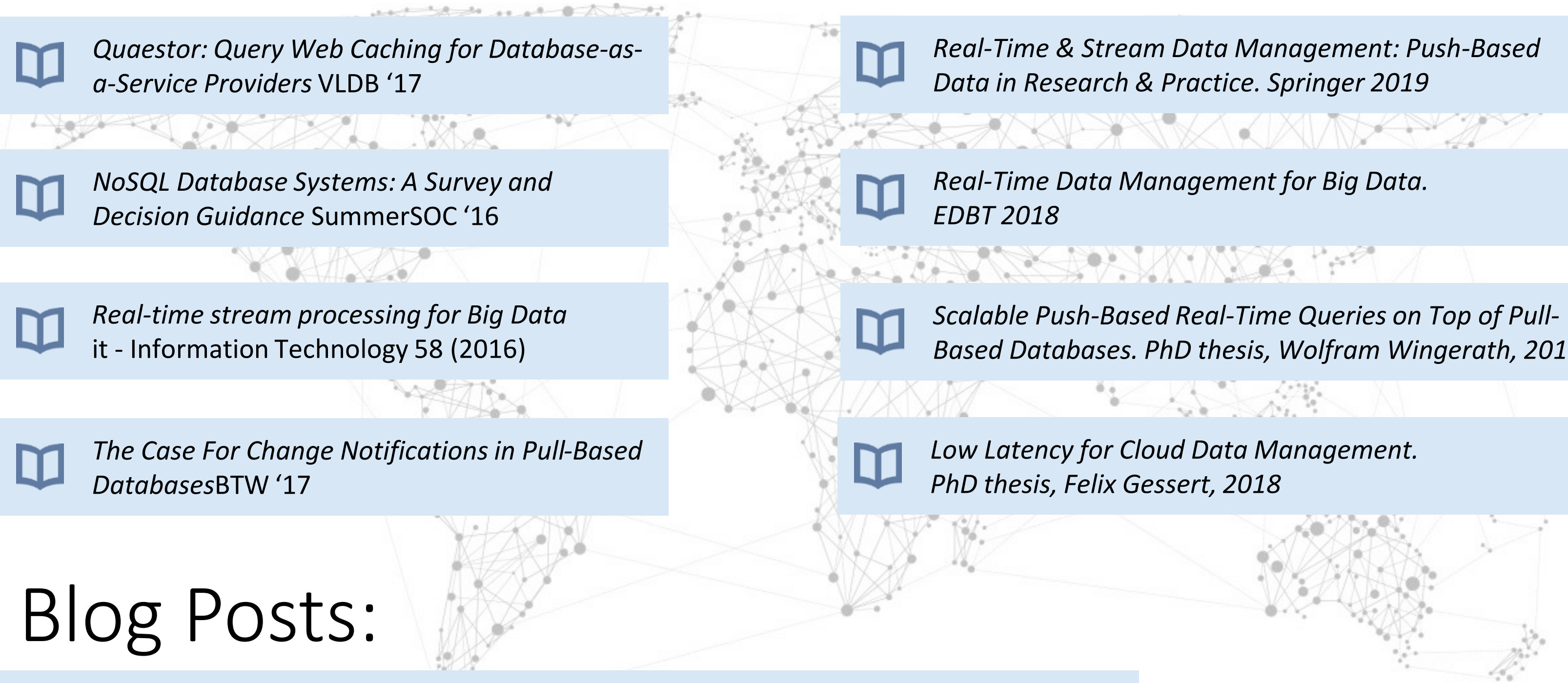
- Content-based search? Composite filters?
- Ordering? Limit? Offset?

Legacy Support:

- Real-time queries for *existing databases*?
- *Decouple* OLTP from real-time workloads?

Our Related Publications

Book, Papers, Articles & Tutorials:



- *Quaestor: Query Web Caching for Database-as-a-Service Providers* VLDB ‘17
- *NoSQL Database Systems: A Survey and Decision Guidance* SummerSOC ‘16
- *Real-time stream processing for Big Data* it - Information Technology 58 (2016)
- *The Case For Change Notifications in Pull-Based Databases*BTW ‘17
- *Real-Time & Stream Data Management: Push-Based Data in Research & Practice. Springer 2019*
- *Real-Time Data Management for Big Data. EDBT 2018*
- *Scalable Push-Based Real-Time Queries on Top of Pull-Based Databases. PhD thesis, Wolfram Wingerath, 2018*
- *Low Latency for Cloud Data Management. PhD thesis, Felix Gessert, 2018*

Blog Posts:

- *Real-Time Databases Explained: Why Meteor, RethinkDB, Parse and Firebase Don't Scale* Baqend Tech Blog (2017): https://medium.com/p/822ff87d2f87

Learn more at blog.baqend.com!

Thank you

wingerath@informatik.uni-hamburg.de

Blog: blog.baqend.com
Slides: slides.baqend.com

@baqendcom