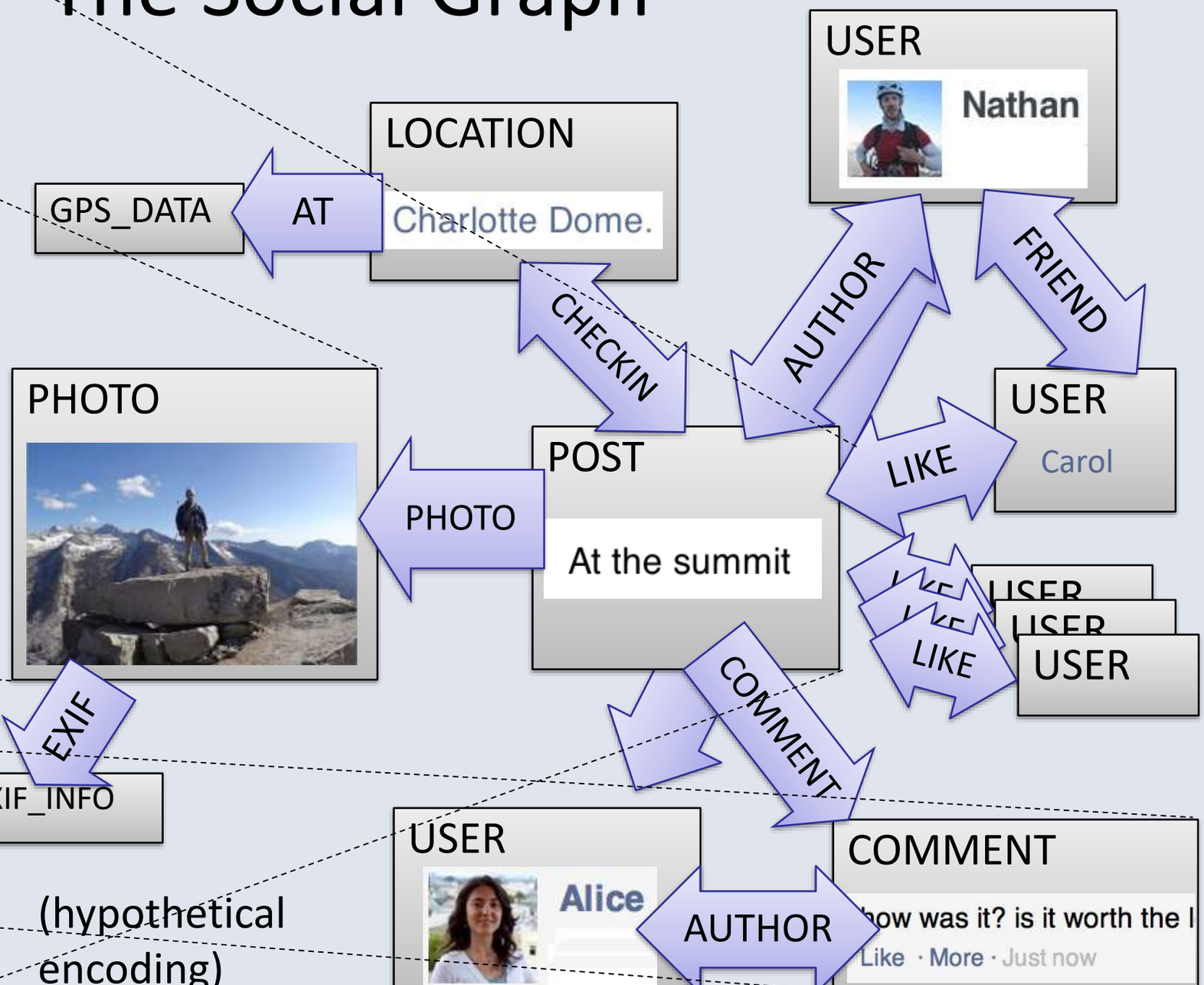
facebook

TAO
Facebook's Distributed Data Store for the Social Graph

Nathan Bronson, Zach Amsden, George Cabrera, Prasad Chakka, Peter Dimov, Hui Ding, Jack Ferris, Anthony Giardullo, Sachin Kulkarni, Harry Li, Mark Marchukov, Dmitri Petrov, Lovro Puzar, Yee Jiun Song, Venkat Venkataramani

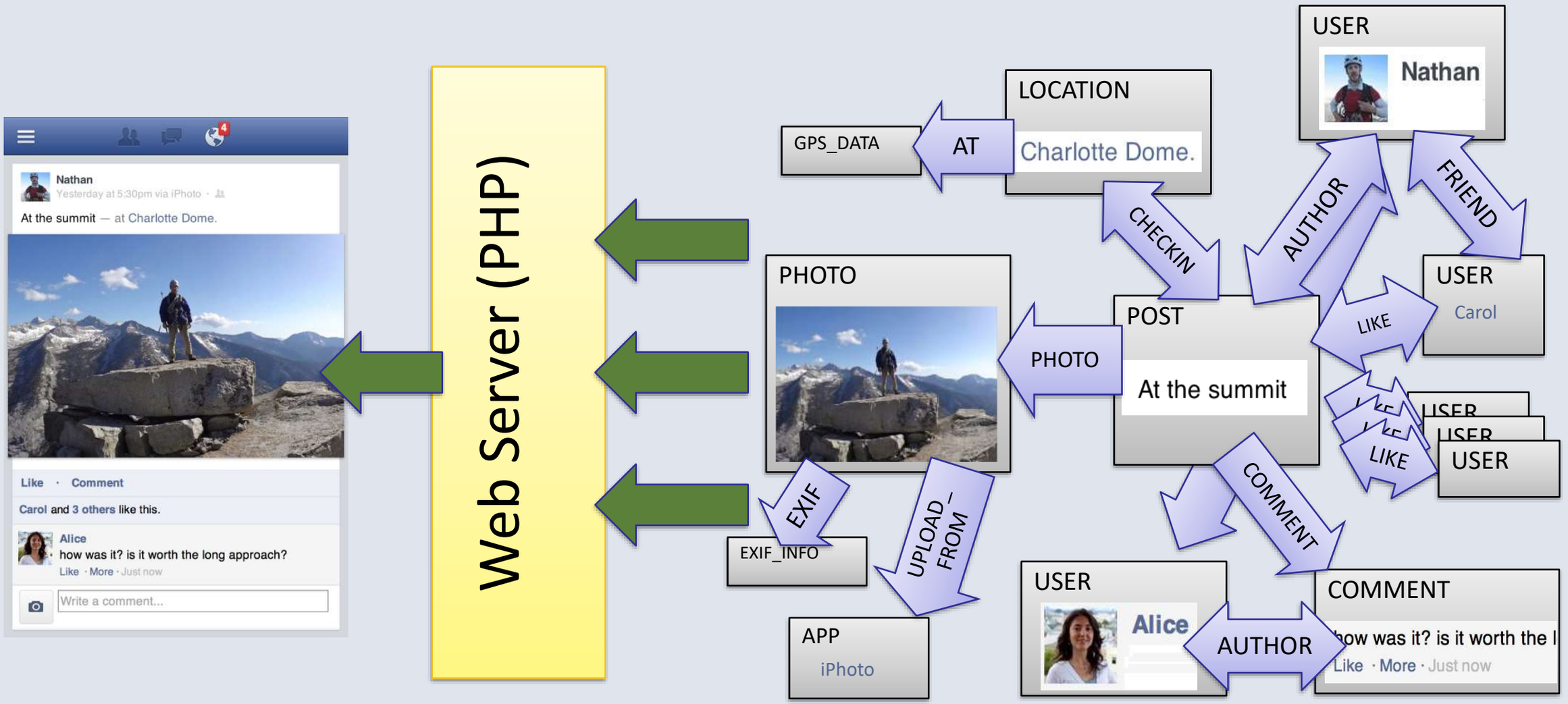
Presented at USENIX ATC – June 26, 2013

The Social Graph

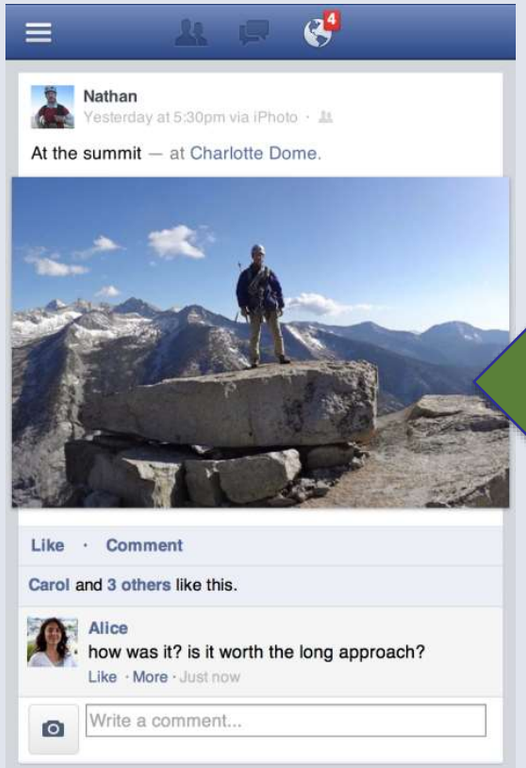


(hypothetical encoding)

Dynamically Rendering the Graph

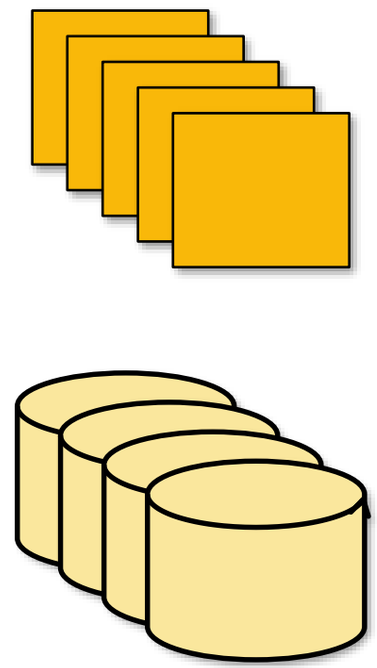


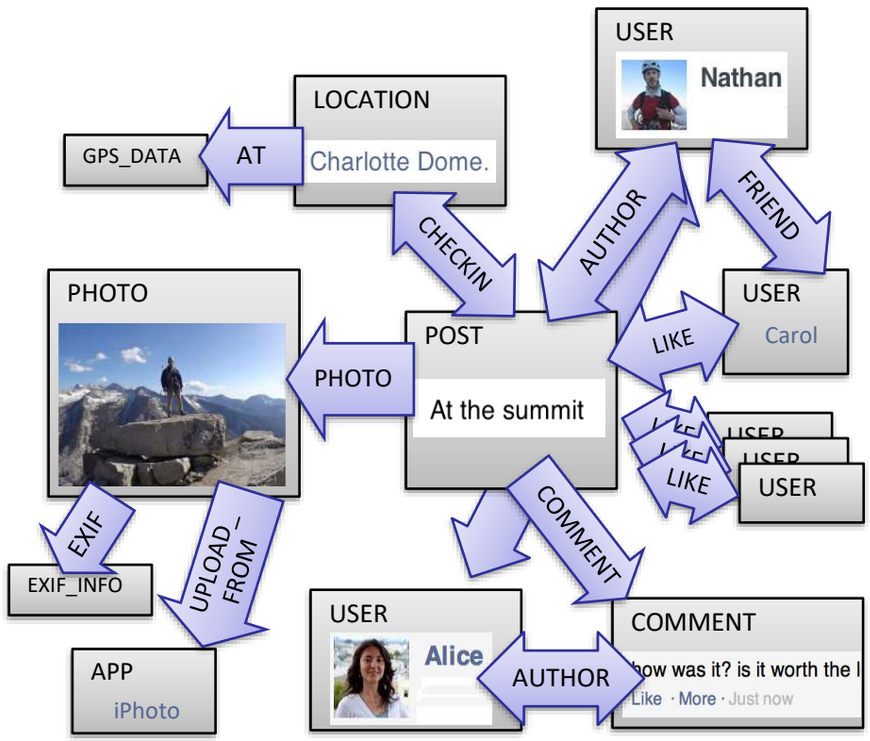
Dynamically Rendering the Graph



Web Server (PHP)

TAO





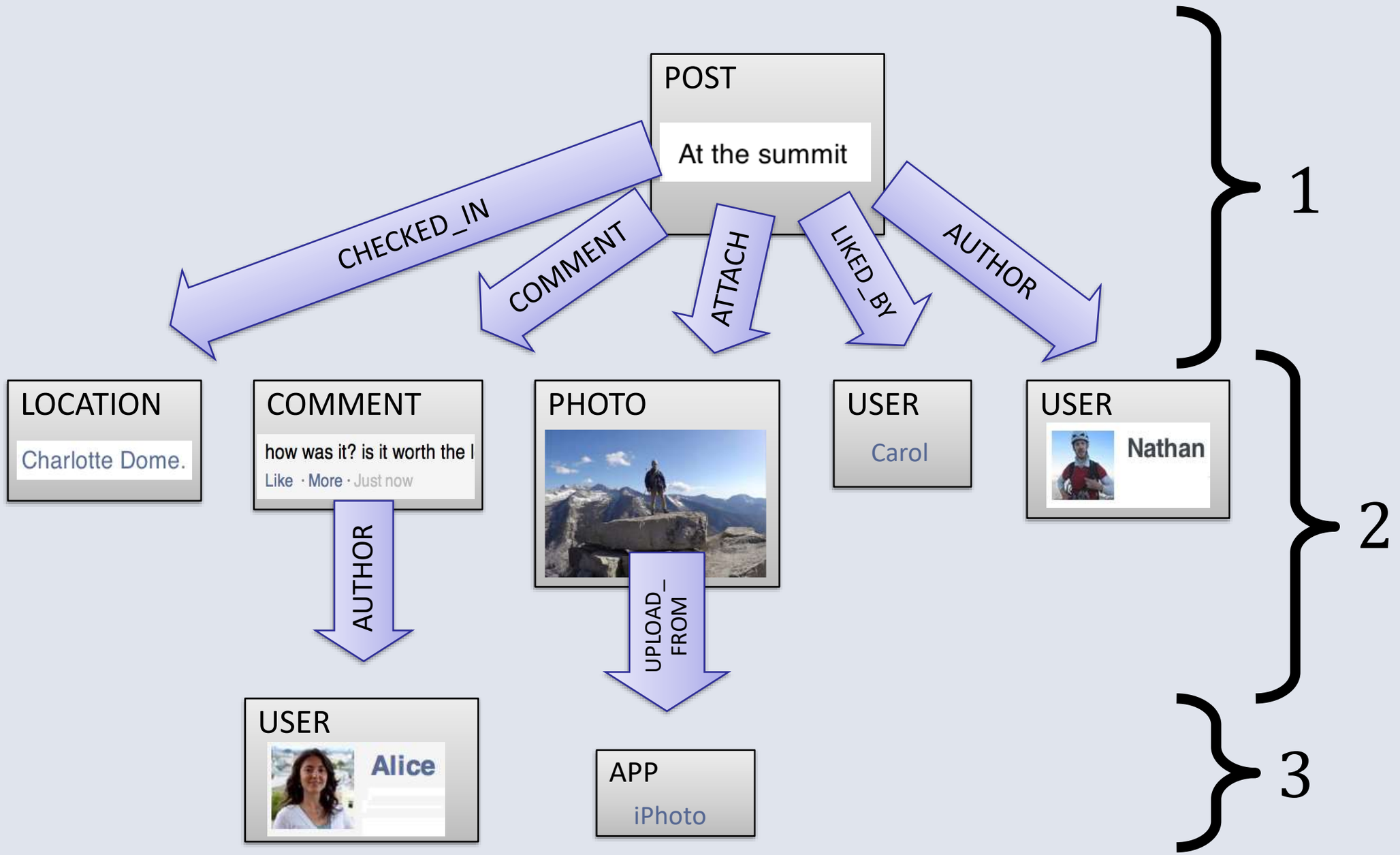
- 1 billion queries/second
- many petabytes of data

facebook data centers

What Are TAO's Goals/Challenges?

- Efficiency at scale

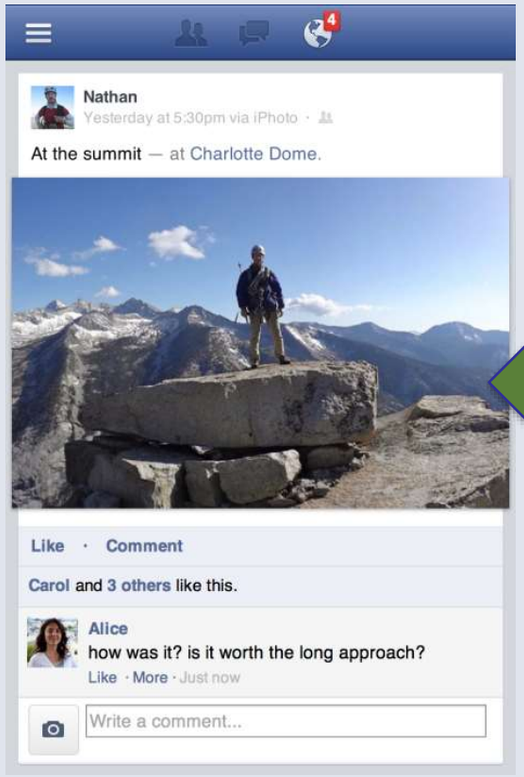
Dynamic Resolution of Data Dependencies



What Are TAO's Goals/Challenges?

- Efficiency at scale
- Low read latency
- Timeliness of writes
- High Read Availability

Graph in Memcache

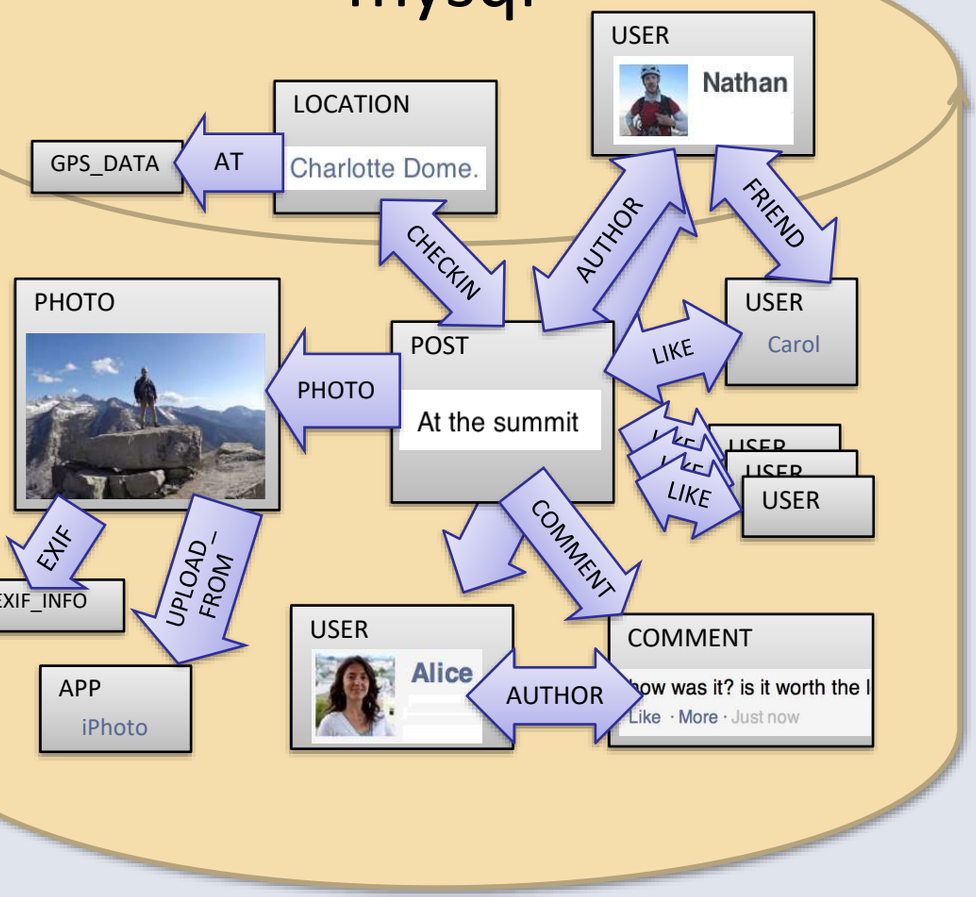


Web Server (PHP)

Obj & Assoc API

memcache
(nodes, edges, edge lists)

mysql

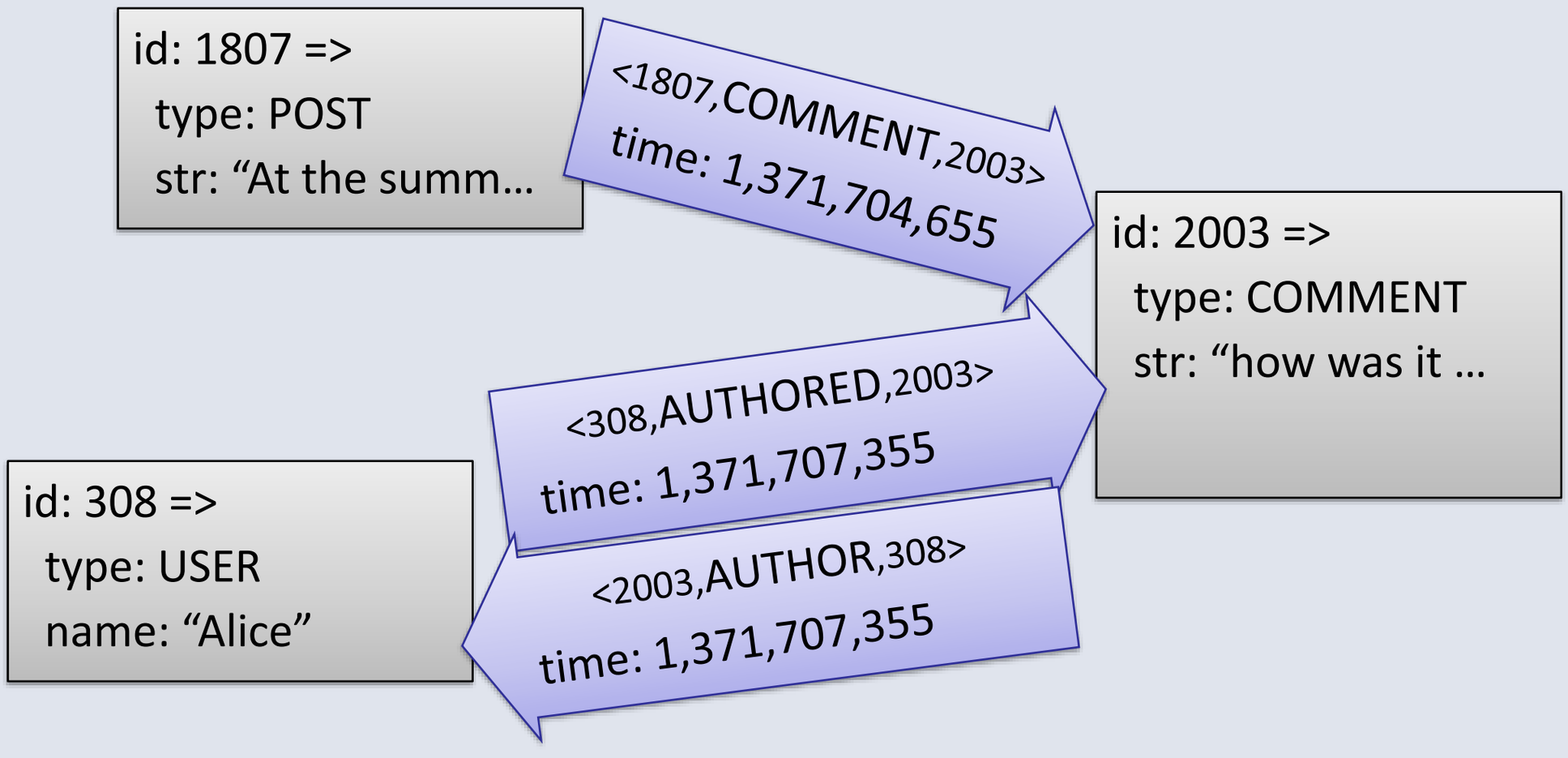


Objects = Nodes

- Identified by unique 64-bit IDs
- Typed, with a schema for fields

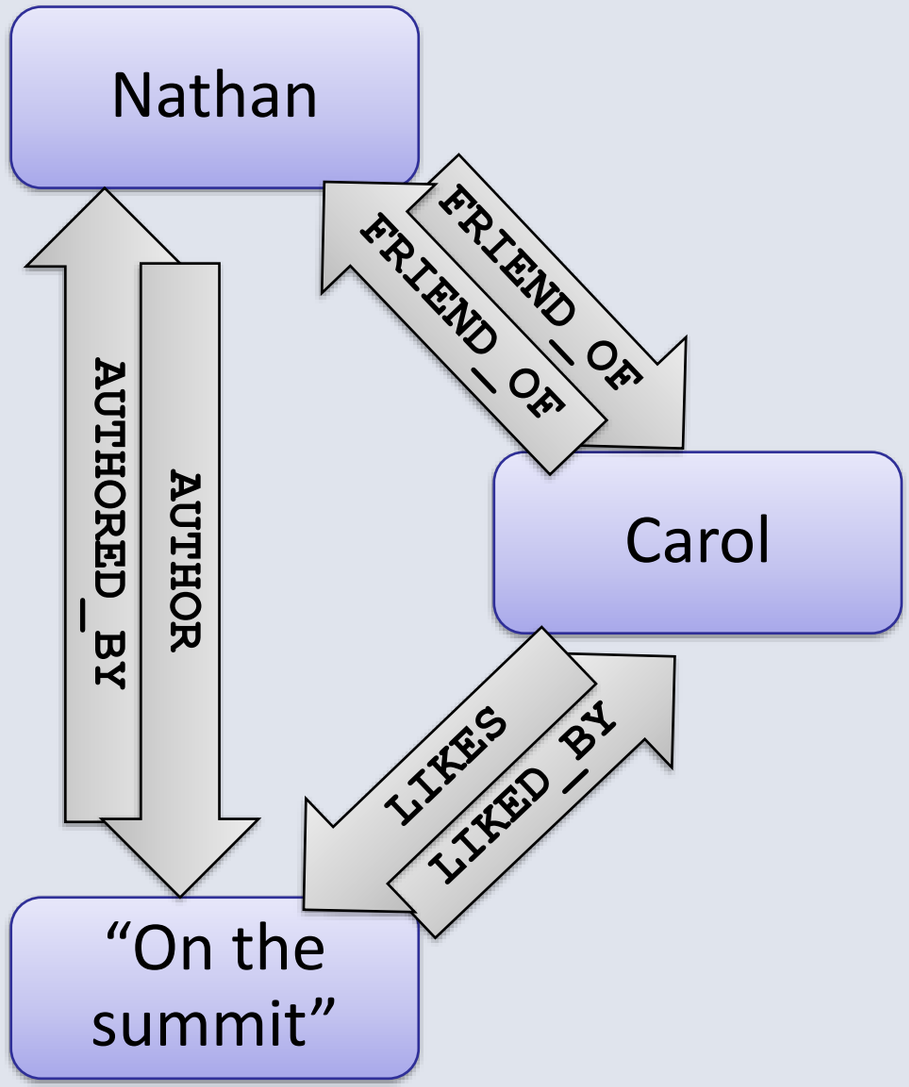
Associations = Edges

- Identified by <id1, type, id2>
- Bidirectional associations are two edges, same or different type



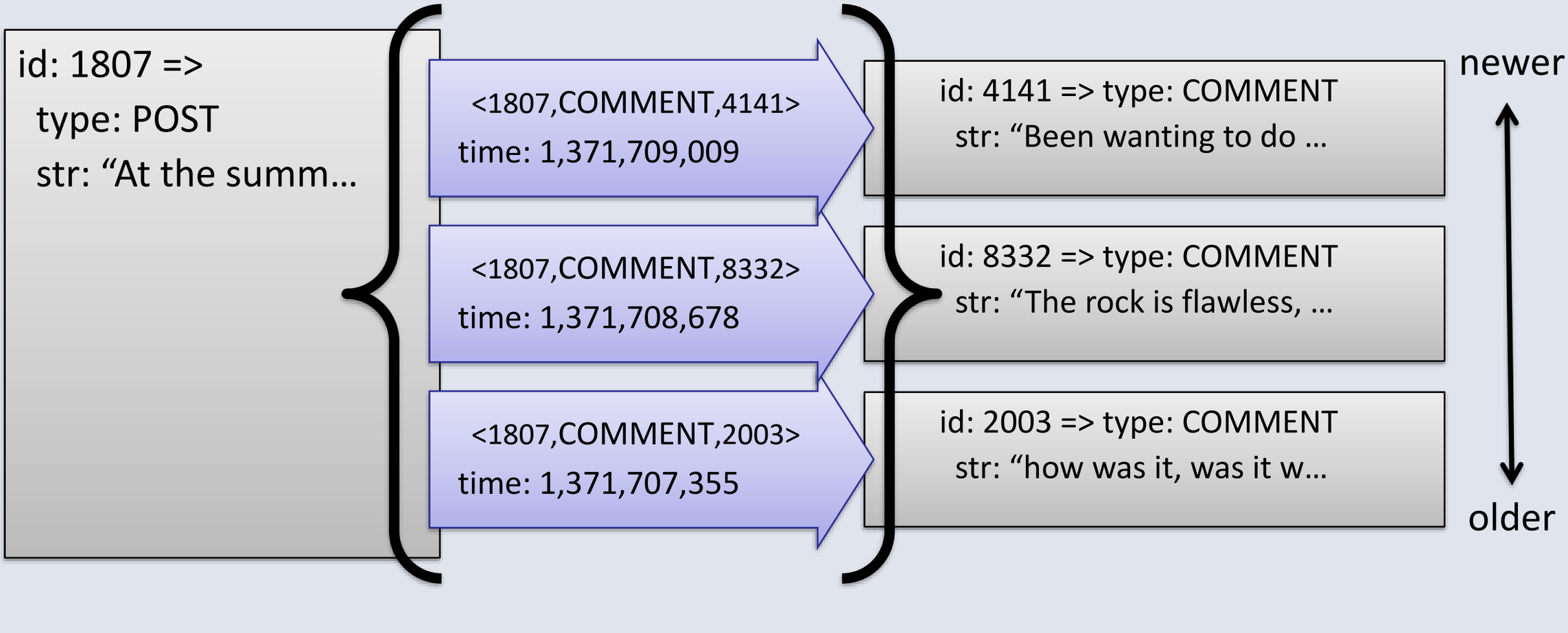
Inverse associations

- Bidirectional relationships have separate *a*→*b* and *b*→*a* edges
  - inv_type(**`LIKES`**) = **`LIKED_BY`**
  - inv_type(**`FRIEND_OF`**) = **`FRIEND_OF`**
- Forward and inverse types linked only during write
  - TAO **`assoc_add`** will update both
  - Not atomic, but failures are logged and repaired



Association Lists

- <id1, type, *>
- Descending order by time
- Query sublist by position or time
- Query size of entire list



Objects and Associations API

Reads – 99.8%

- Point queries
  - **obj_get** 28.9%
  - **assoc_get** 15.7%
- Range queries
  - **assoc_range** 40.9%
  - **assoc_time_range** 2.8%
- Count queries
  - **assoc_count** 11.7%

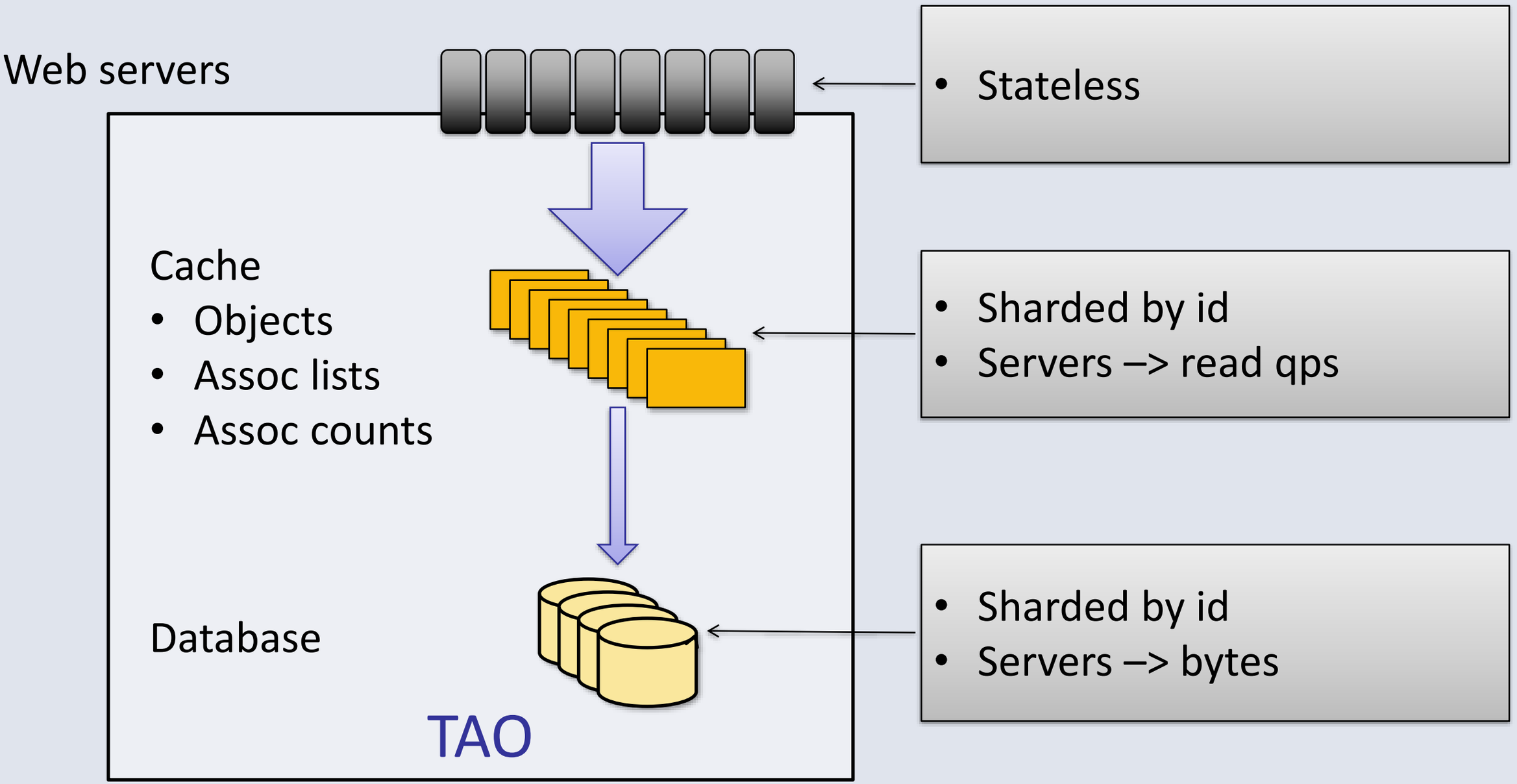
Writes – 0.2%

- Create, update, delete for objects
  - **obj_add** 16.5%
  - **obj_update** 20.7%
  - **obj_del** 2.0%
- Set and delete for associations
  - **assoc_add** 52.5%
  - **assoc_del** 8.3%

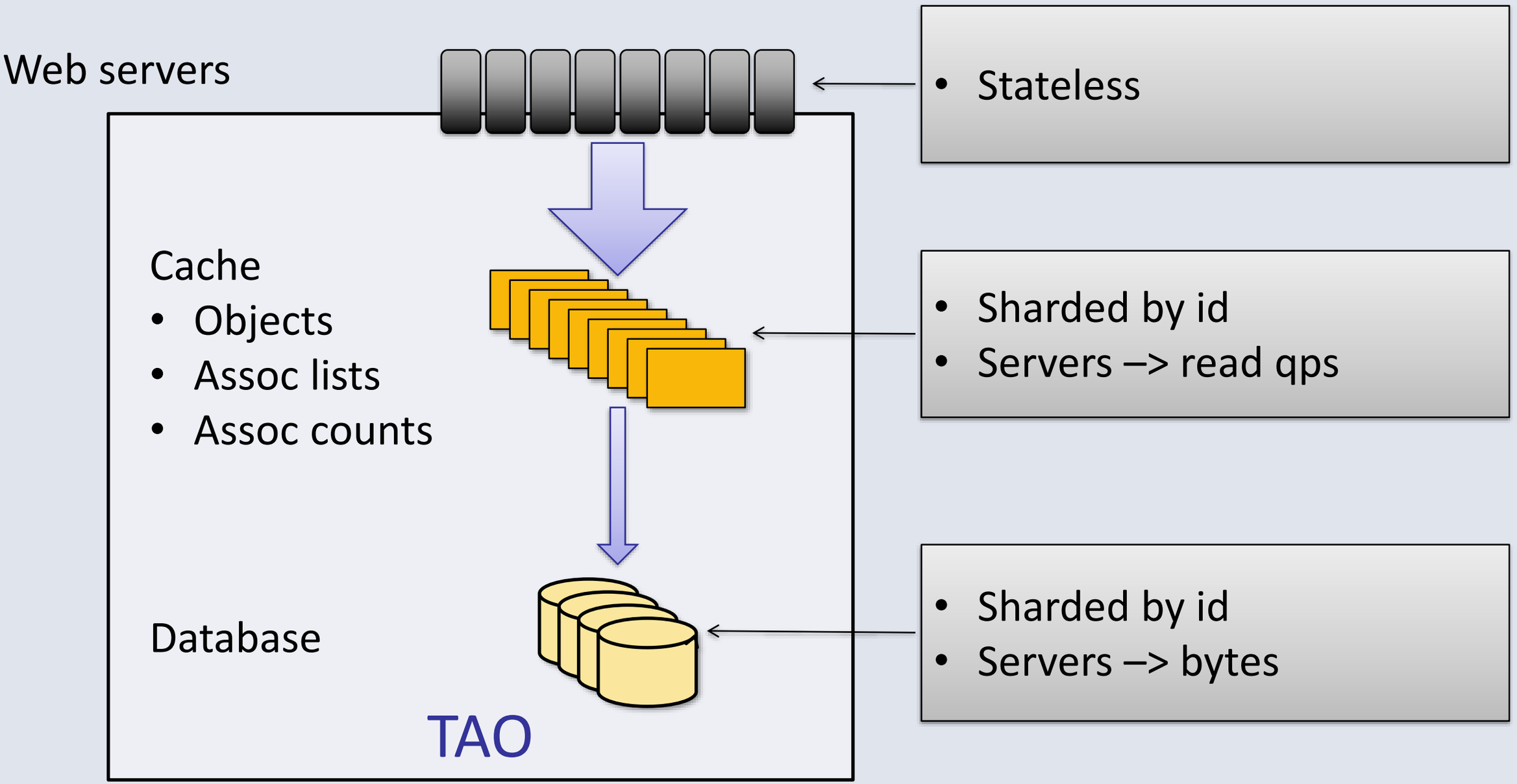
What Are TAO's Goals/Challenges?

- Efficiency at scale
- Low read latency
- Timeliness of writes
- High Read Availability

Independent Scaling by Separating Roles



Independent Scaling by Separating Roles



Object ID -> Shard -> Cache node

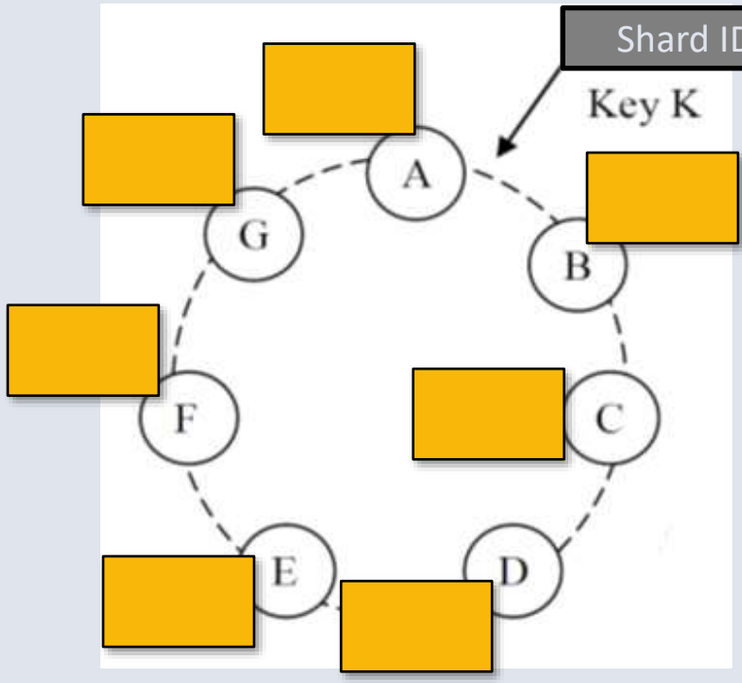
ObjID O_i

Assoc. O_i, type, O_j

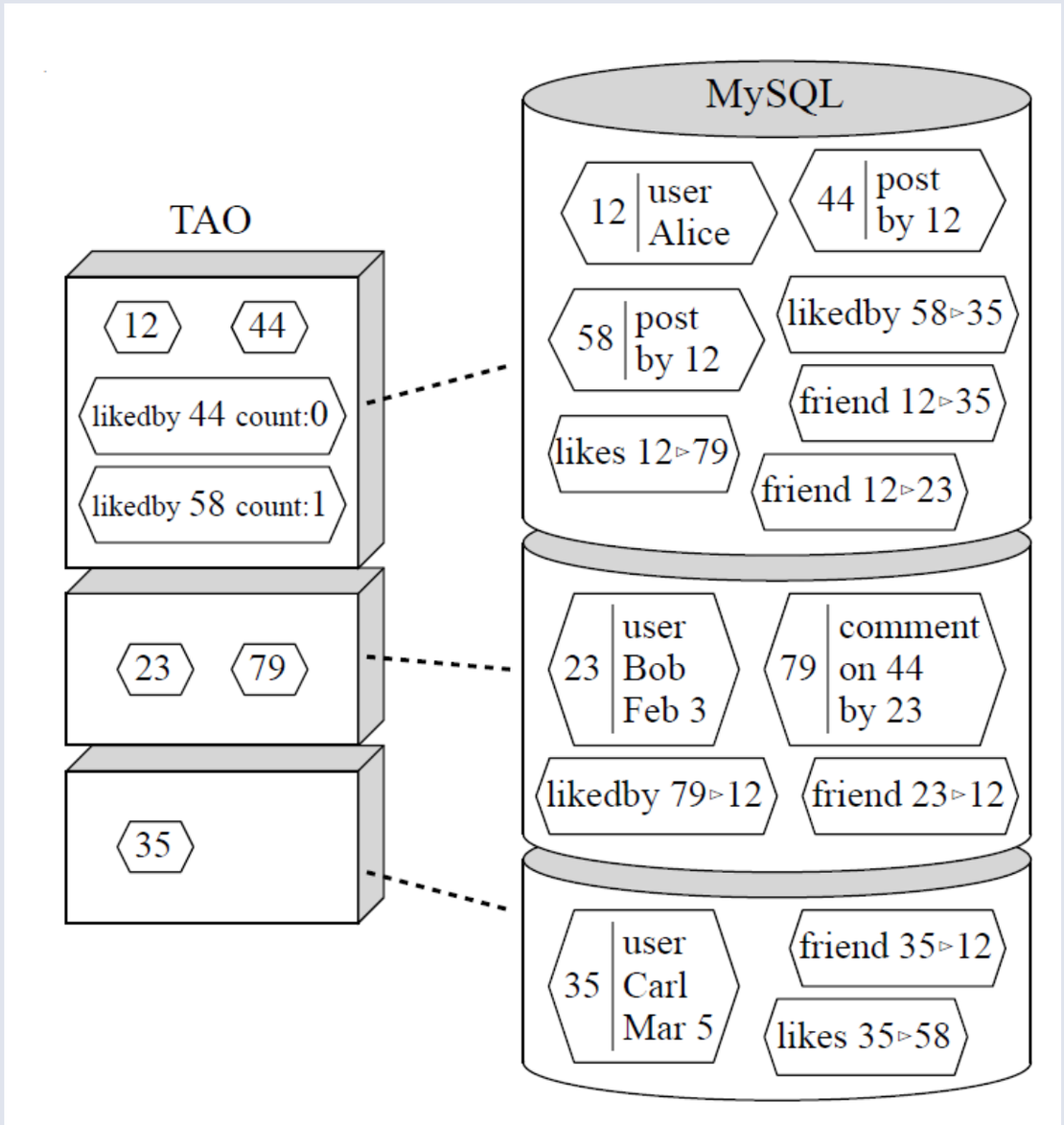
Static assignment, at creation time

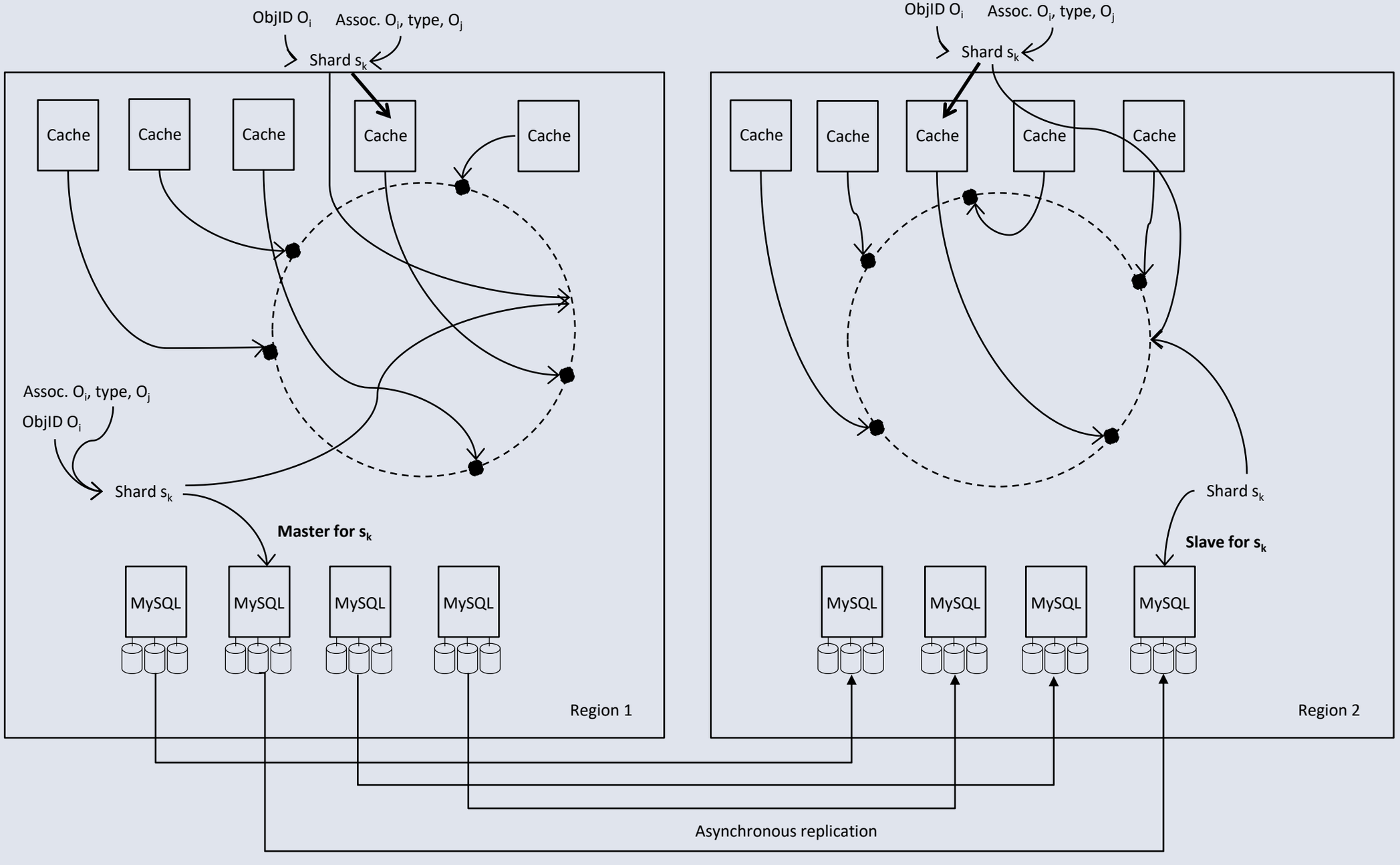
Shard ID

Consistent hashing



Sharding in Tao



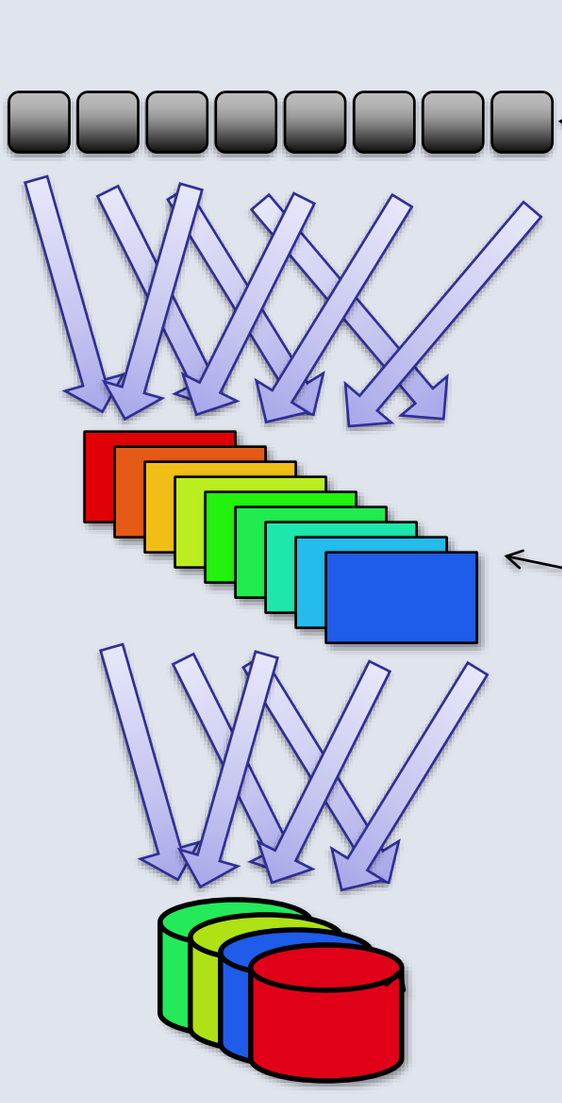

ObjID O_i
Assoc. O_i, type, O_j
Shard s_k
Cache
Cache
Cache
Cache
Cache
Assoc. O_i, type, O_j
ObjID O_i
Shard s_k
Master for s_k
MySQL
MySQL
MySQL
MySQL
Region 1
ObjID O_i
Assoc. O_i, type, O_j
Shard s_k
Cache
Cache
Cache
Cache
Cache
Shard s_k
Slave for s_k
MySQL
MySQL
MySQL
MySQL
Region 2
Asynchronous replication

Subdividing the Data Center

Web servers

Cache

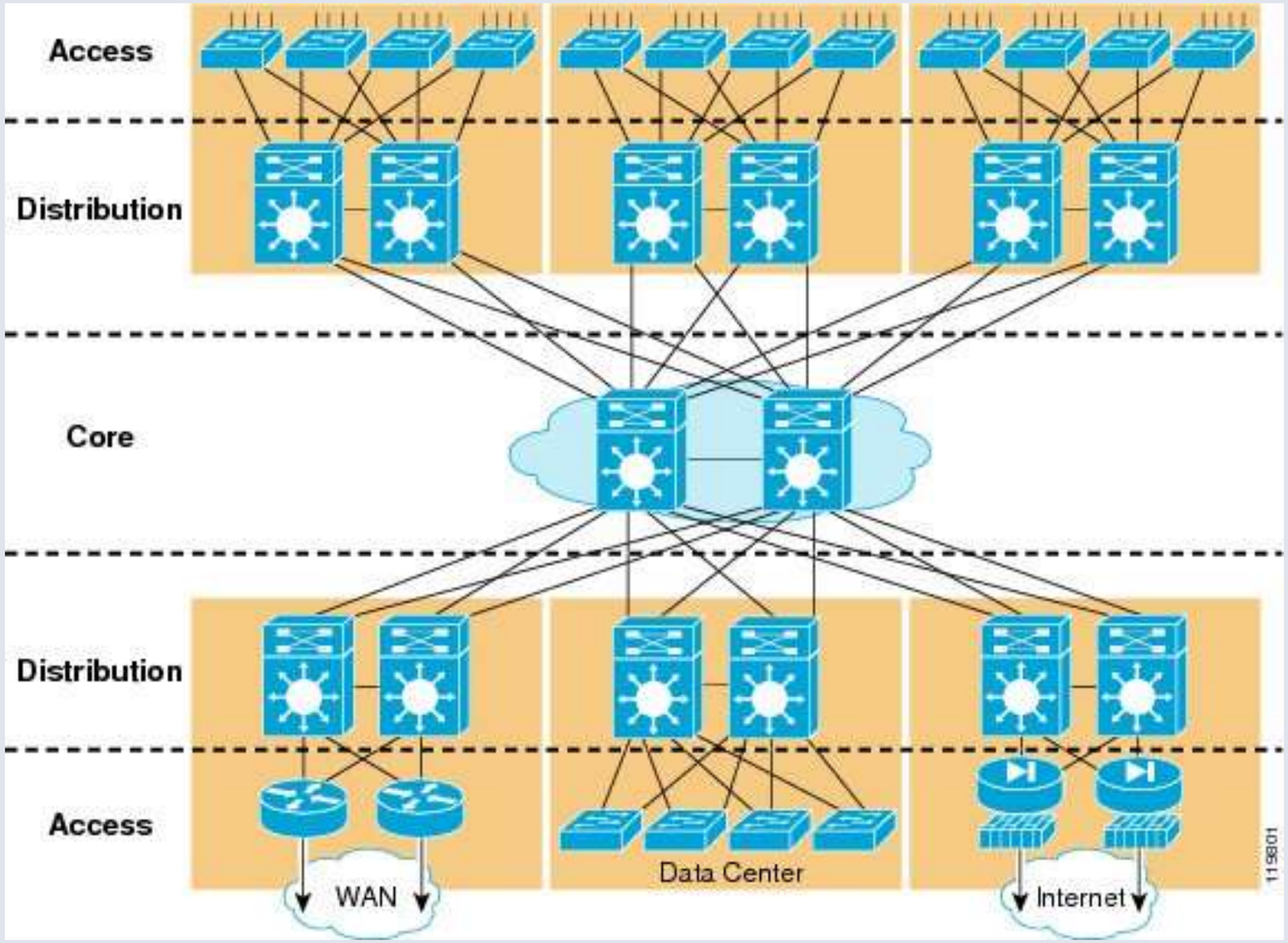
Database



- Inefficient failure detection
- Many switch traversals

- Many open sockets
- Lots of hot spots

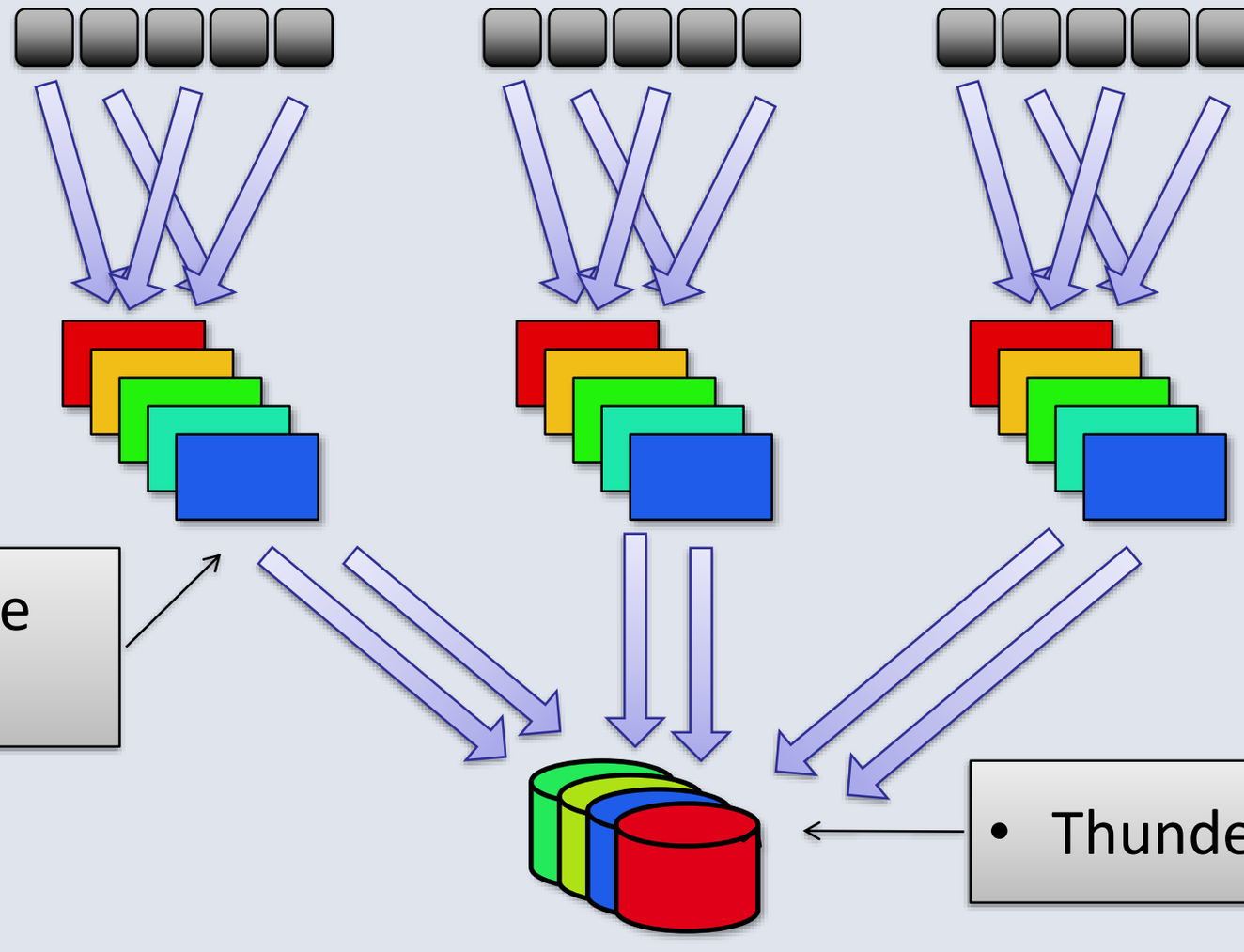
Network design



Source: http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html

Subdividing the Data Center

Web servers



Cache

- Distributed write control logic

Database

- Thundering herds

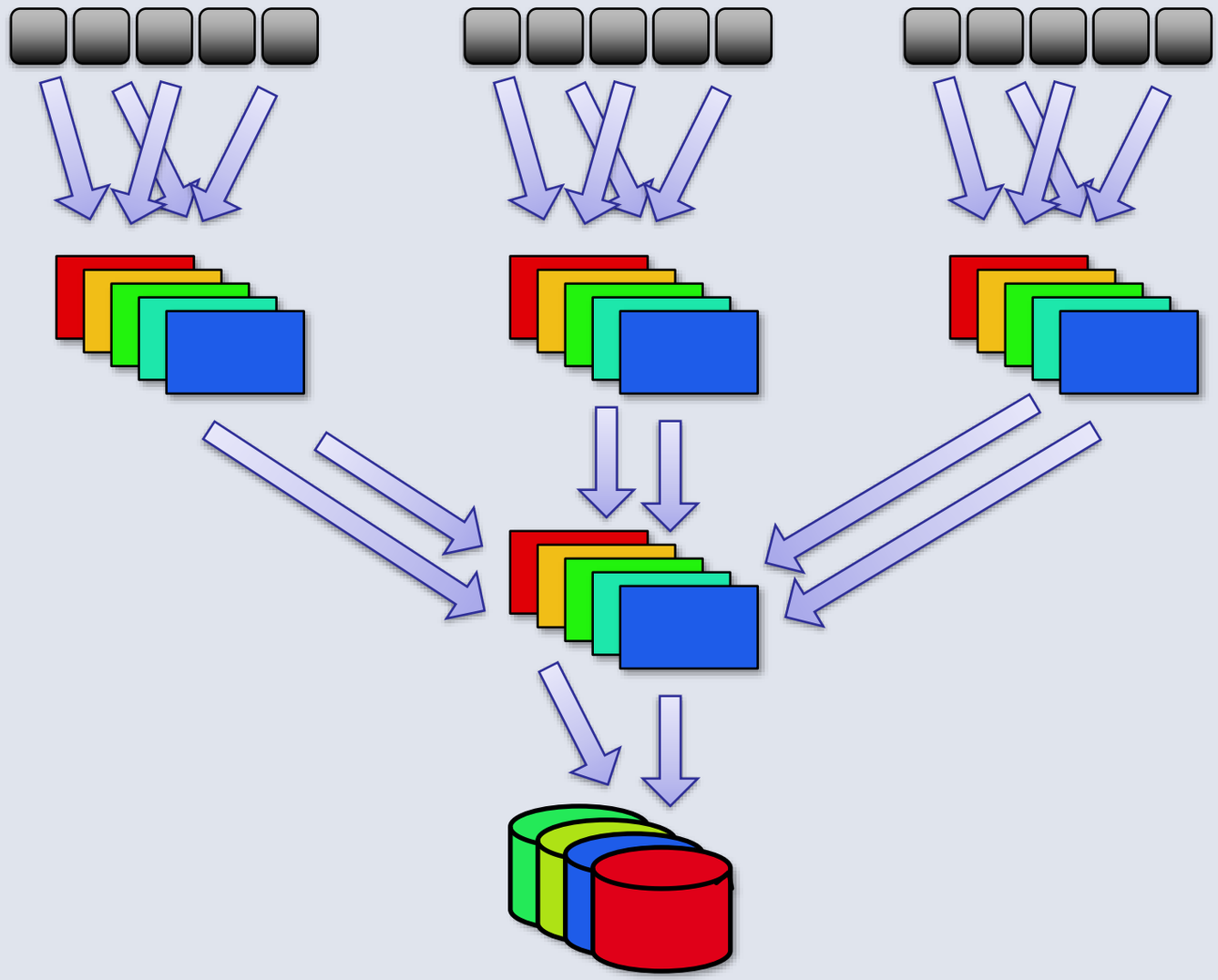
Follower and Leader Caches

Web servers

Follower cache

Leader cache

Database



What Are TAO's Goals/Challenges?

- Efficiency at scale
- Low read latency
- Timeliness of writes
- High Read Availability

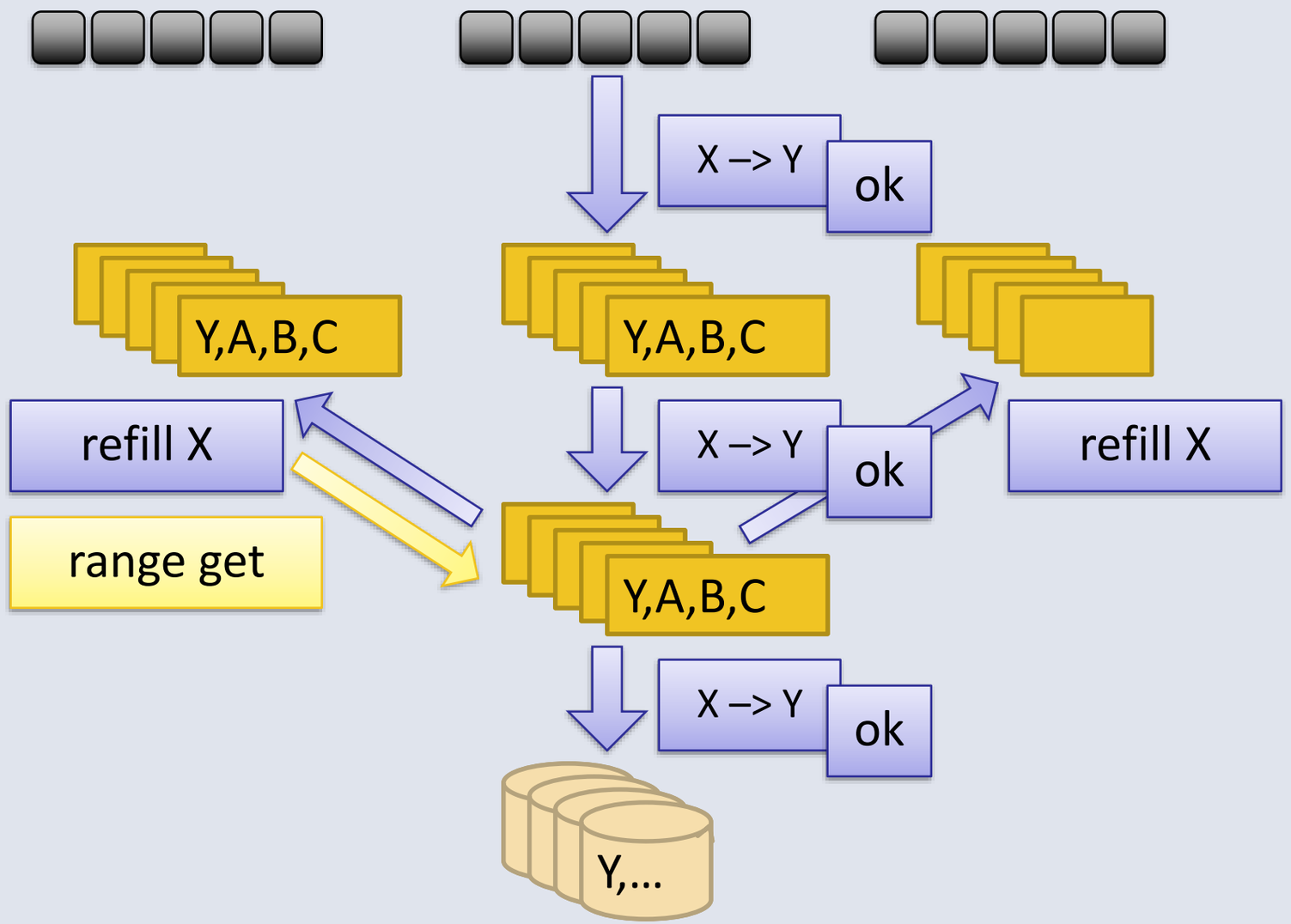
Write-through Caching – Association Lists

Web servers

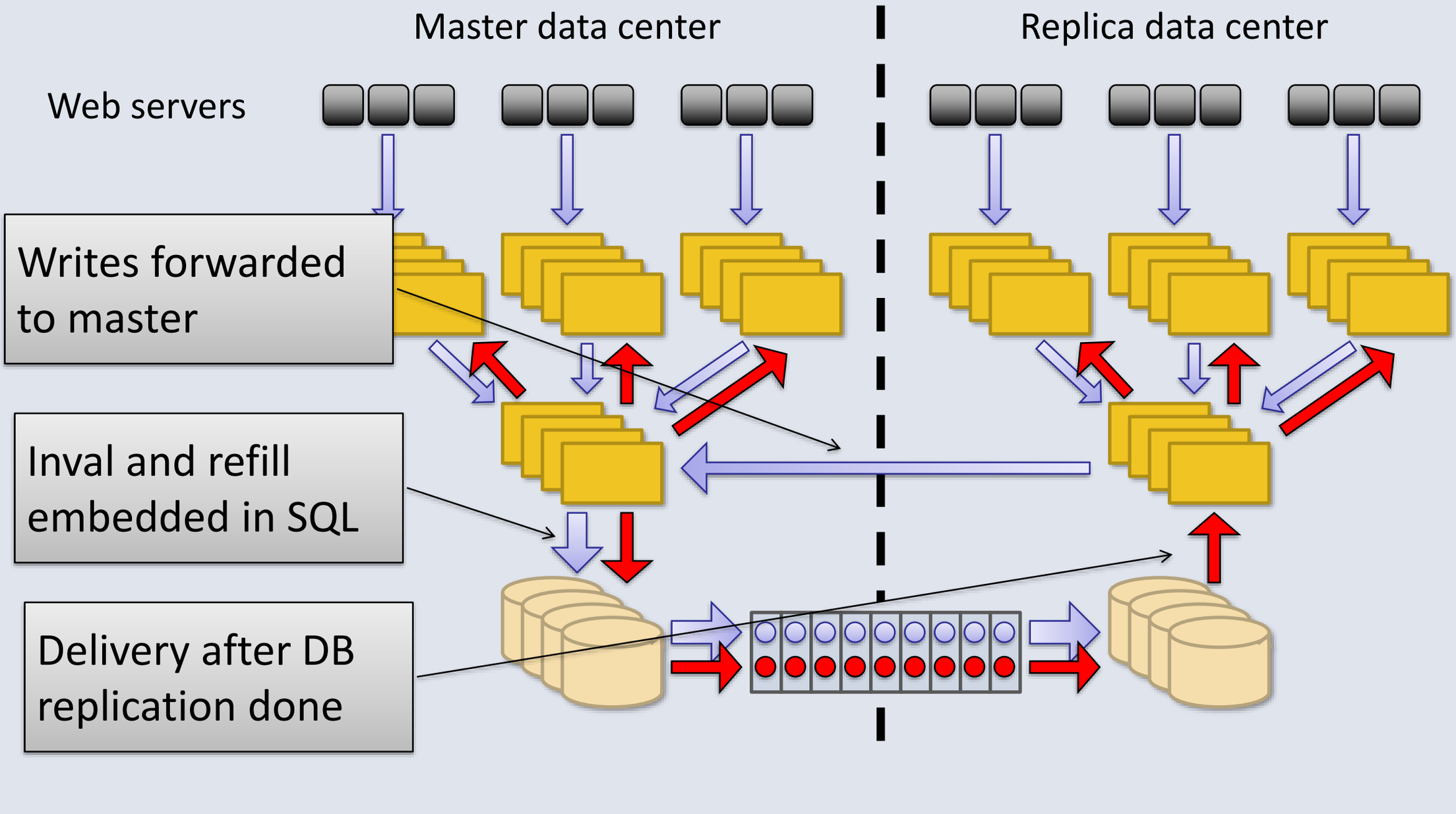
Follower cache

Leader cache

Database



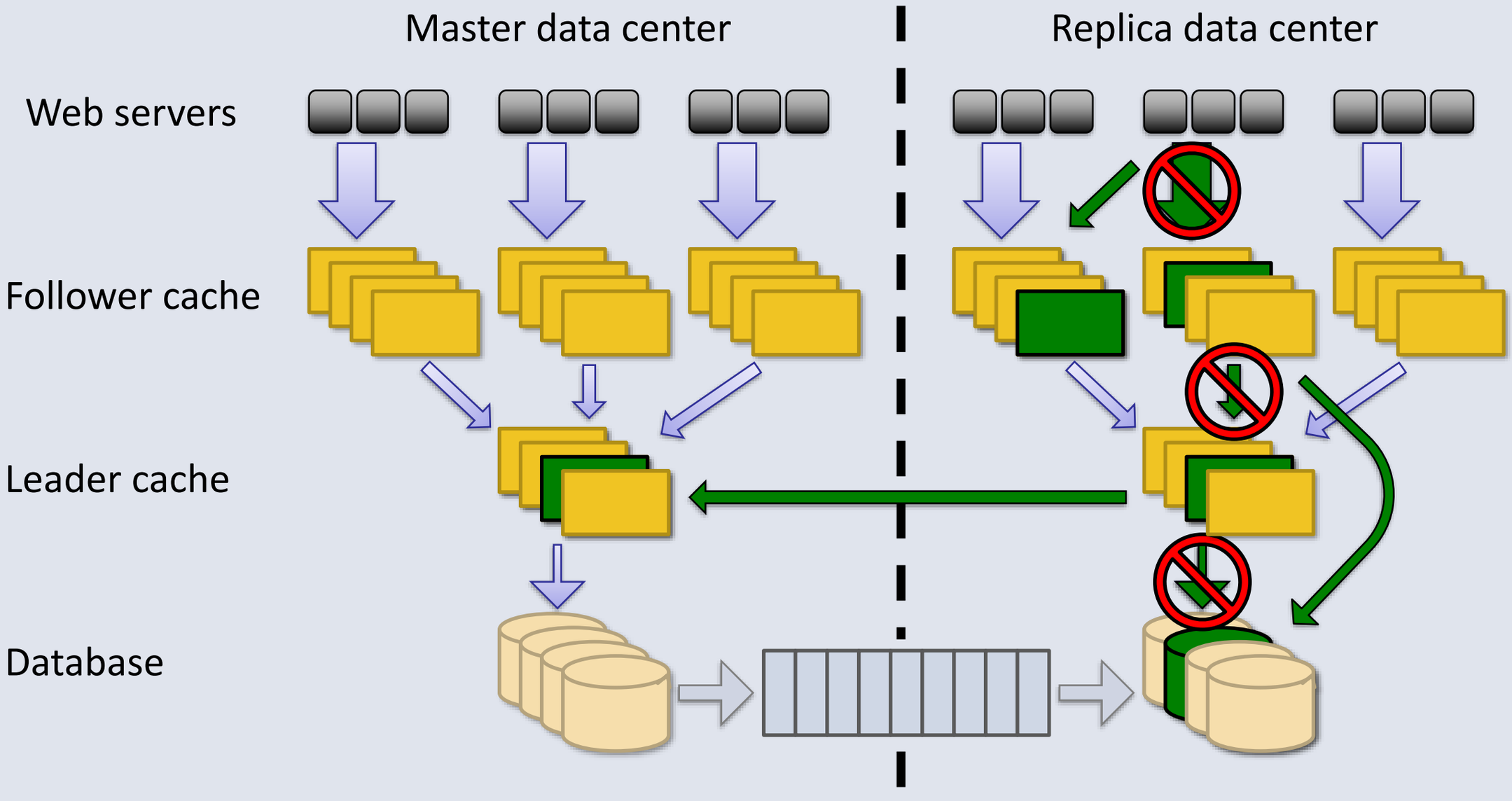
Asynchronous DB Replication



What Are TAO's Goals/Challenges?

- Efficiency at scale
- Low read latency
- Timeliness of writes
- High Read Availability

Improving Availability: Read Failover

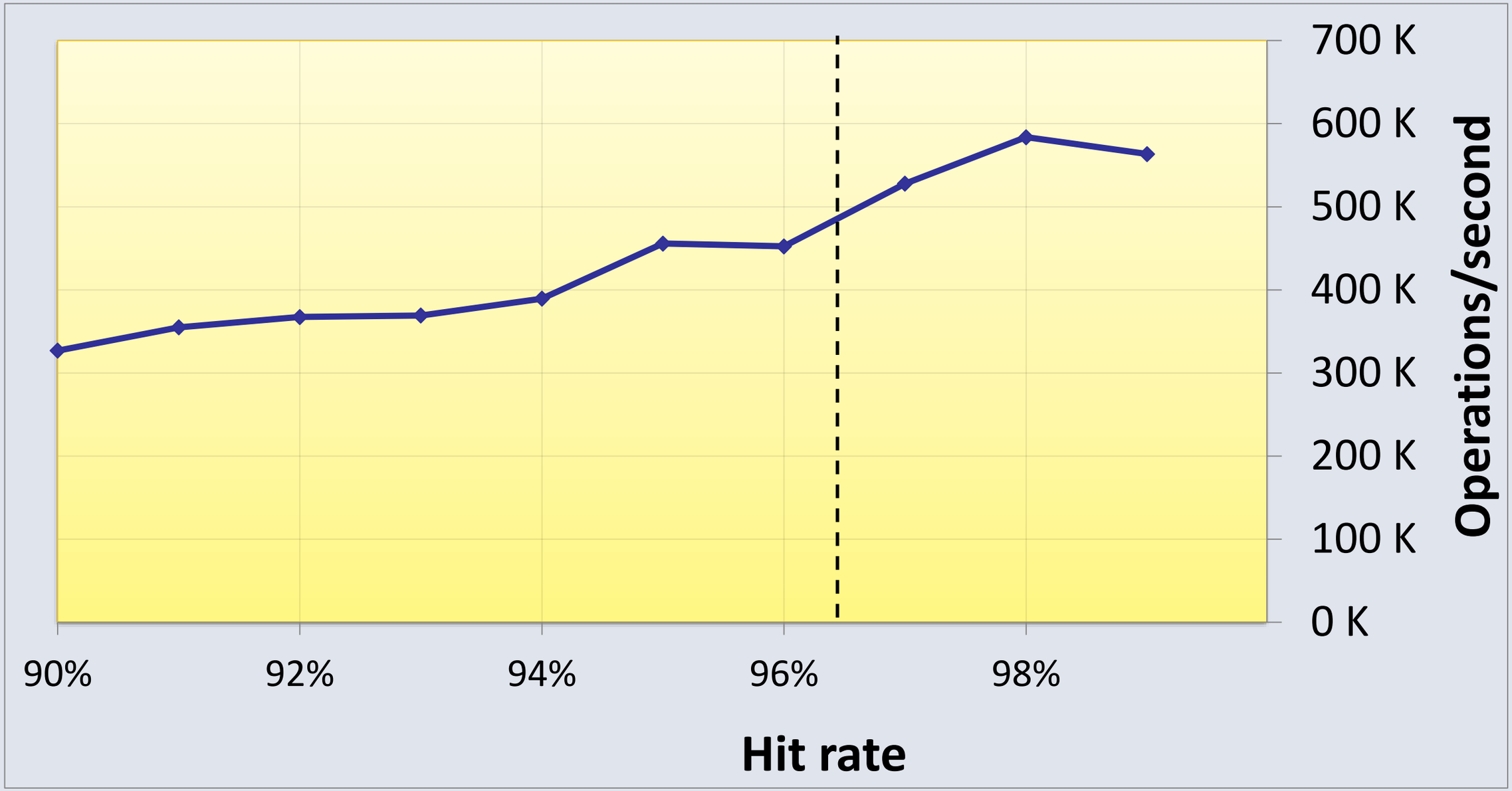


TAO Summary

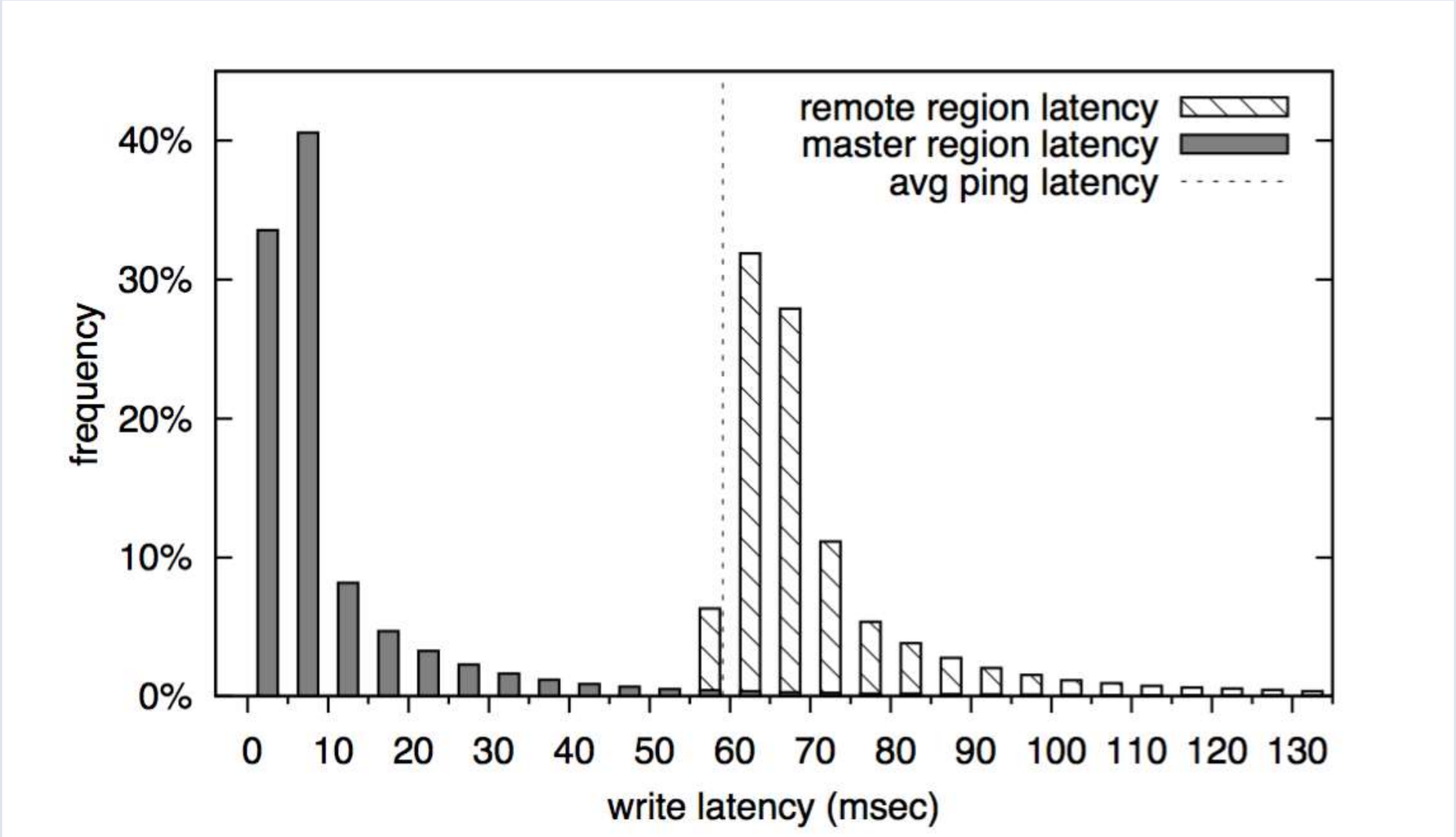
Efficiency at scale Read latency	• Separate cache and DB • Graph-specific caching • Subdivide data centers
Write timeliness	• Write-through cache • Asynchronous replication
Read availability	• Alternate data sources

facebook

Single-server Peak Observed Capacity

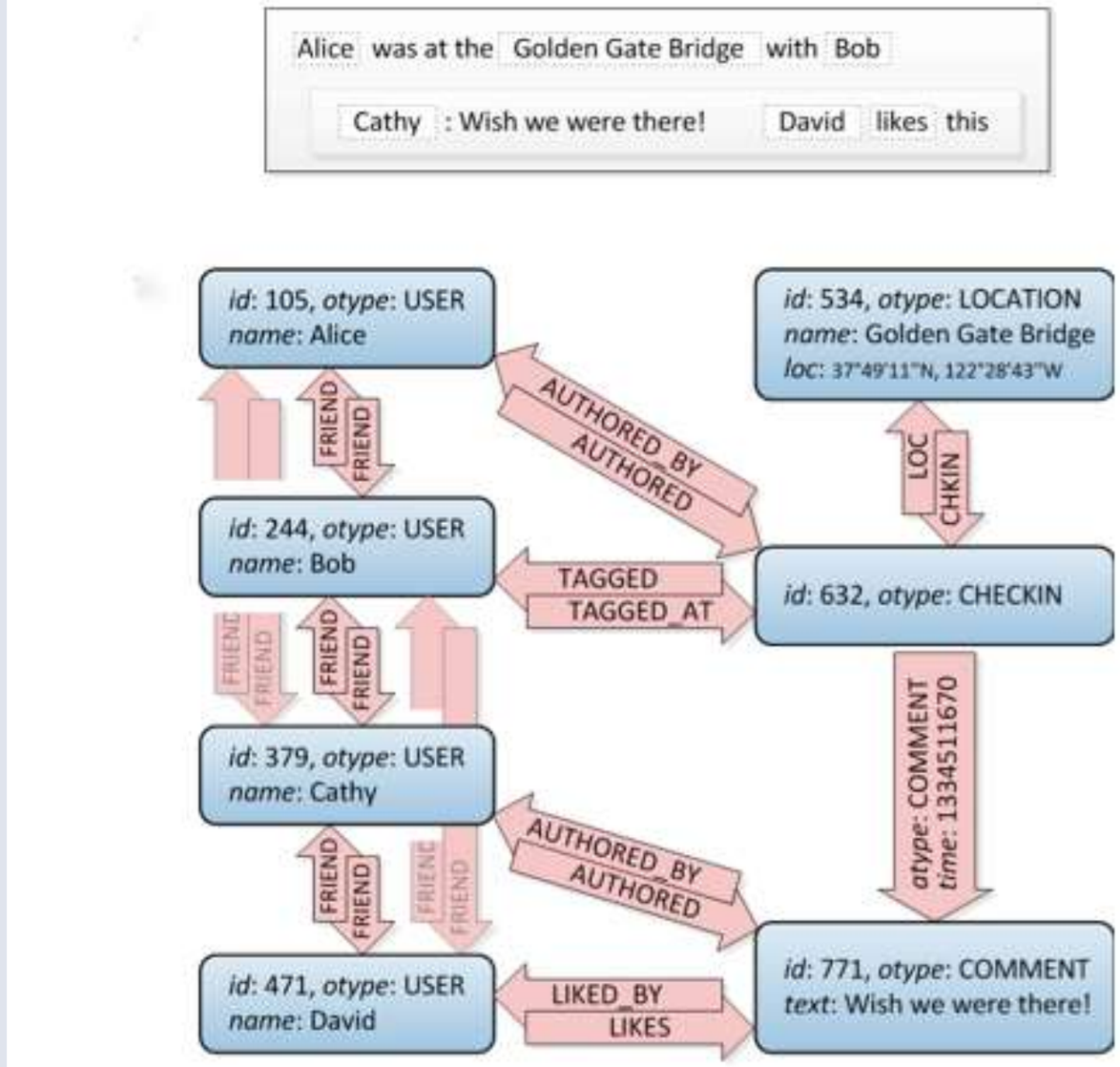


Write latency



More In the Paper

- The role of association time in optimizing cache hit rates
- Optimized graph-specific data structures
- Write failover
- Failure recovery
- Workload characterization


Alice was at the Golden Gate Bridge with Bob
Cathy : Wish we were there! David likes this
id: 105, otype: USER
name: Alice
id: 534, otype: LOCATION
name: Golden Gate Bridge
loc: 37°49'11"N, 122°28'43"W
FRIEND
FRIEND
AUTHORED_BY
AUTHORED
LOC
CHKIN
id: 244, otype: USER
name: Bob
TAGGED
TAGGED_AT
id: 632, otype: CHECKIN
FRIEND
FRIEND
FRIEND
FRIEND
id: 379, otype: USER
name: Cathy
atype: COMMENT
time: 1334511670
AUTHORED_BY
AUTHORED
FRIEND
FRIEND
FRIEND
FRIEND
id: 471, otype: USER
name: David
LIKED_BY
LIKES
id: 771, otype: COMMENT
text: Wish we were there!