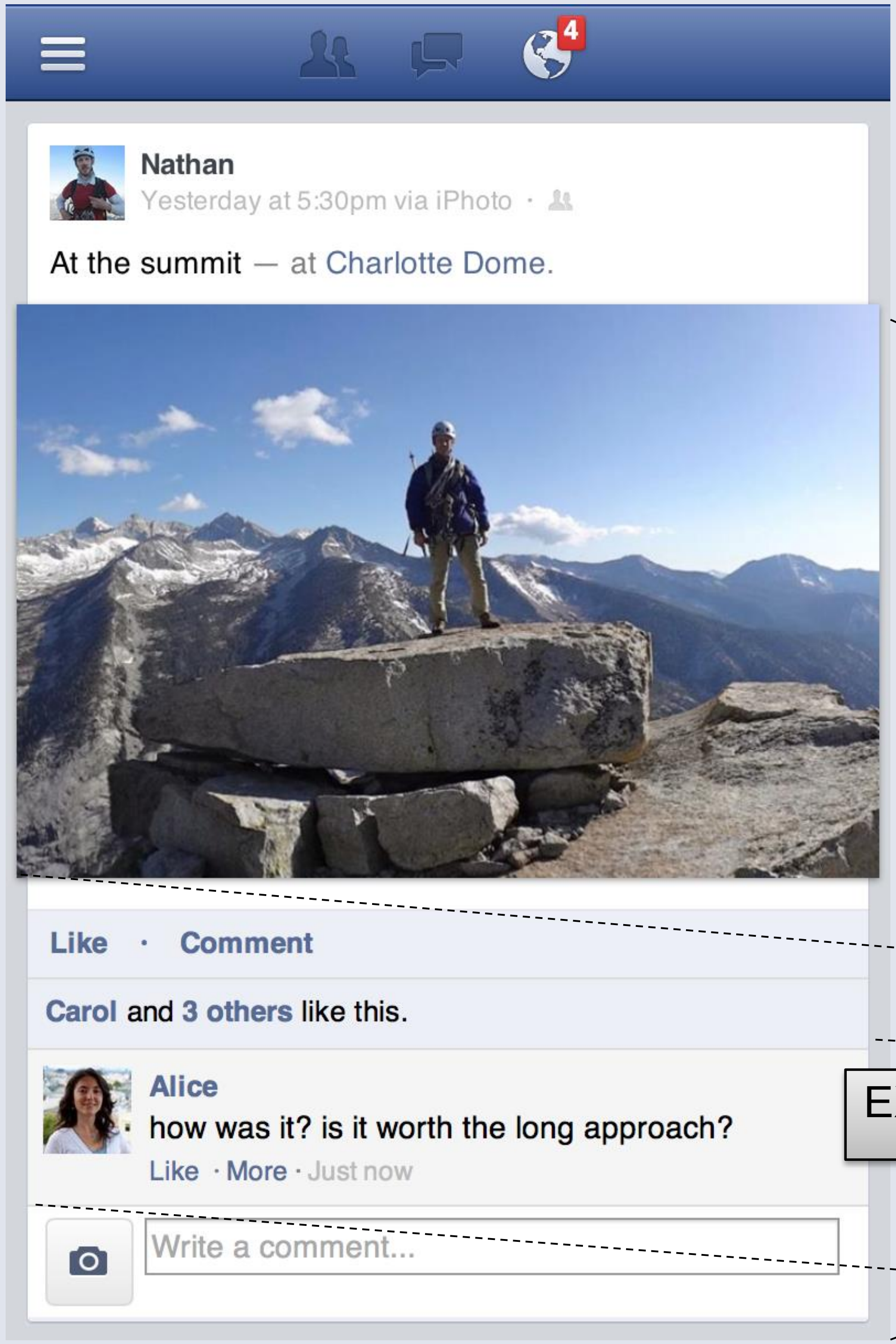
facebook

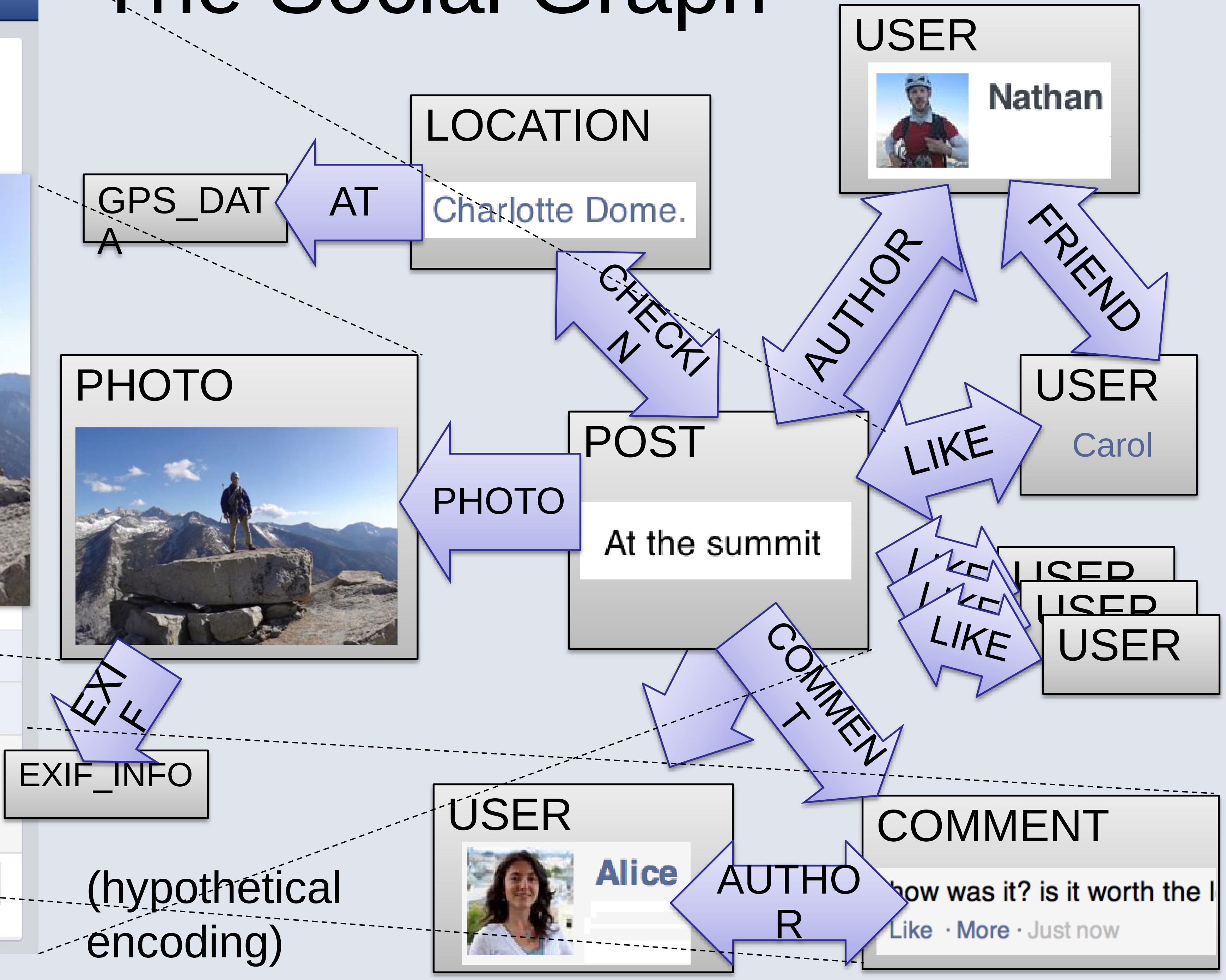
TAO
Facebook's Distributed Data Store for the Social Graph

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

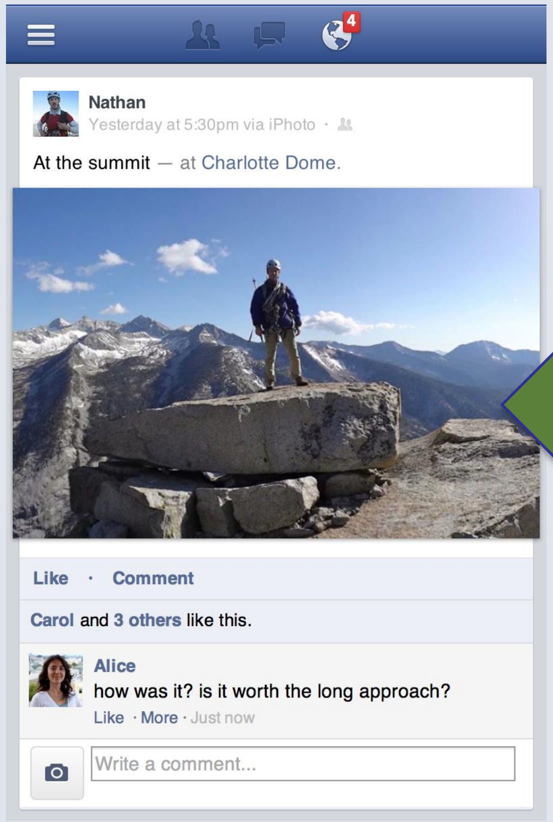
The Social Graph

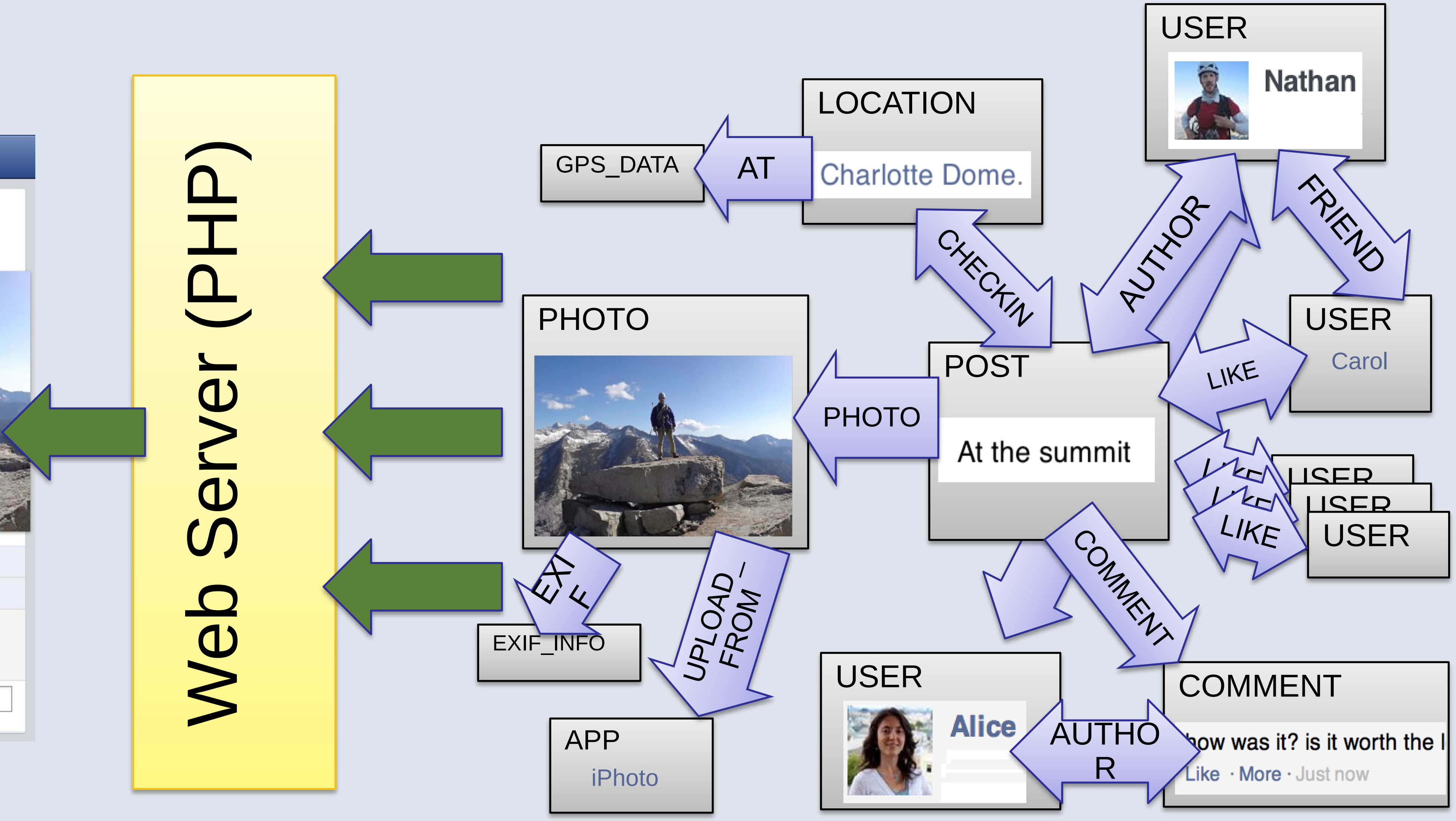

Nathan
Yesterday at 5:30pm via iPhoto ·
At the summit — at Charlotte Dome.
Like · Comment
Carol and 3 others like this.
Alice
how was it? is it worth the long approach?
Like · More · Just now
Write a comment...



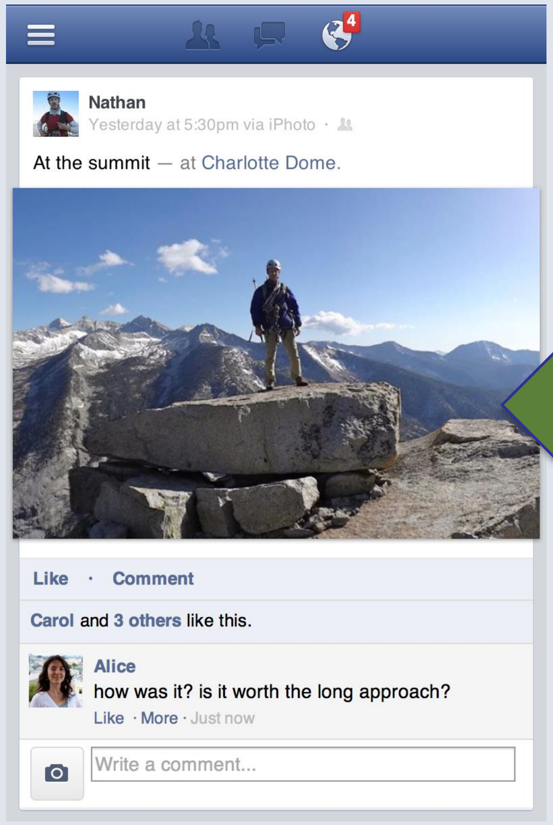
(hypothetical encoding)

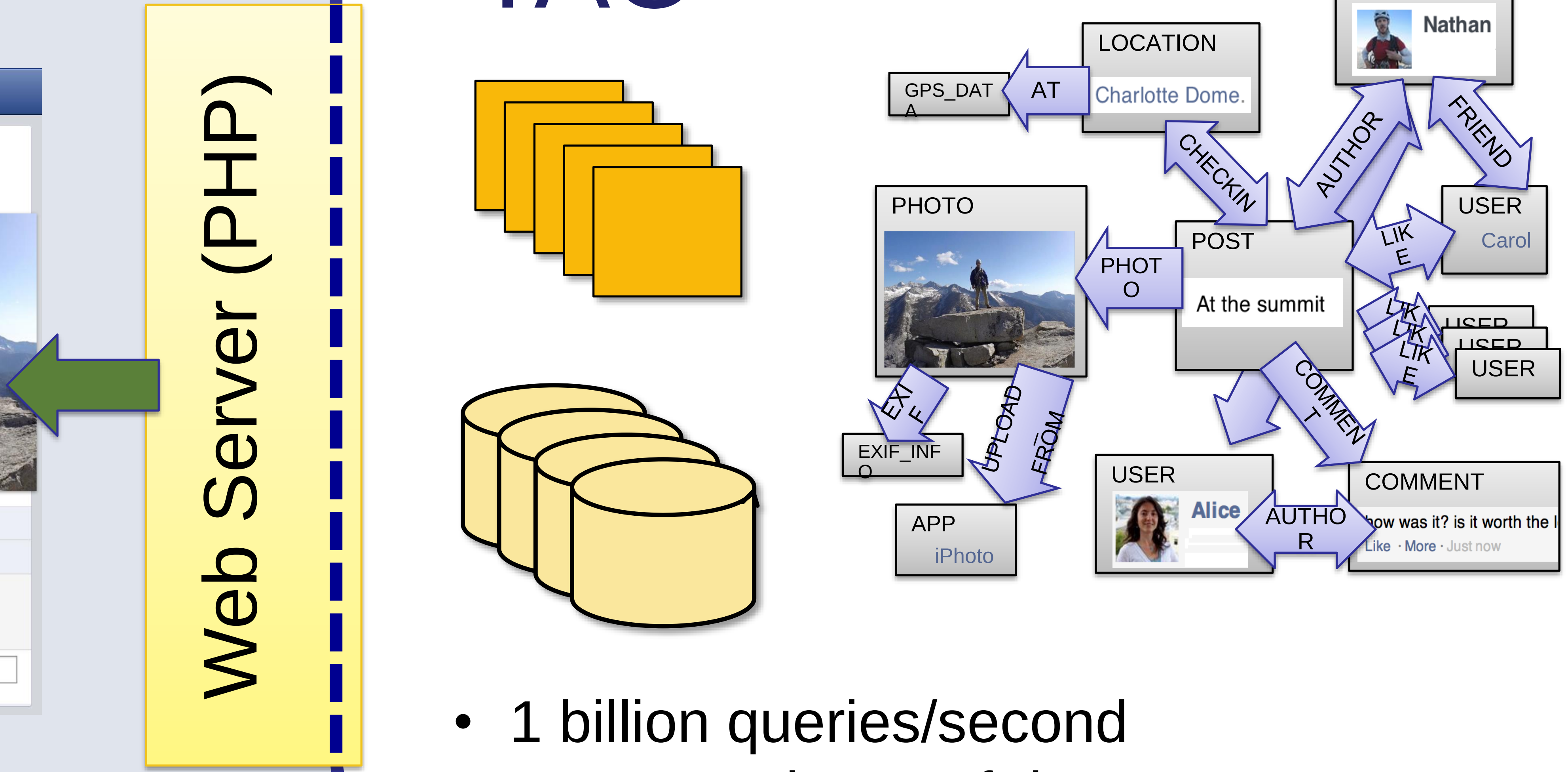
Dynamically Rendering the Graph


Nathan
Yesterday at 5:30pm via iPhoto ·
At the summit — at Charlotte Dome.
Like · Comment
Carol and 3 others like this.
Alice
how was it? is it worth the long approach?
Like · More · Just now
Write a comment...



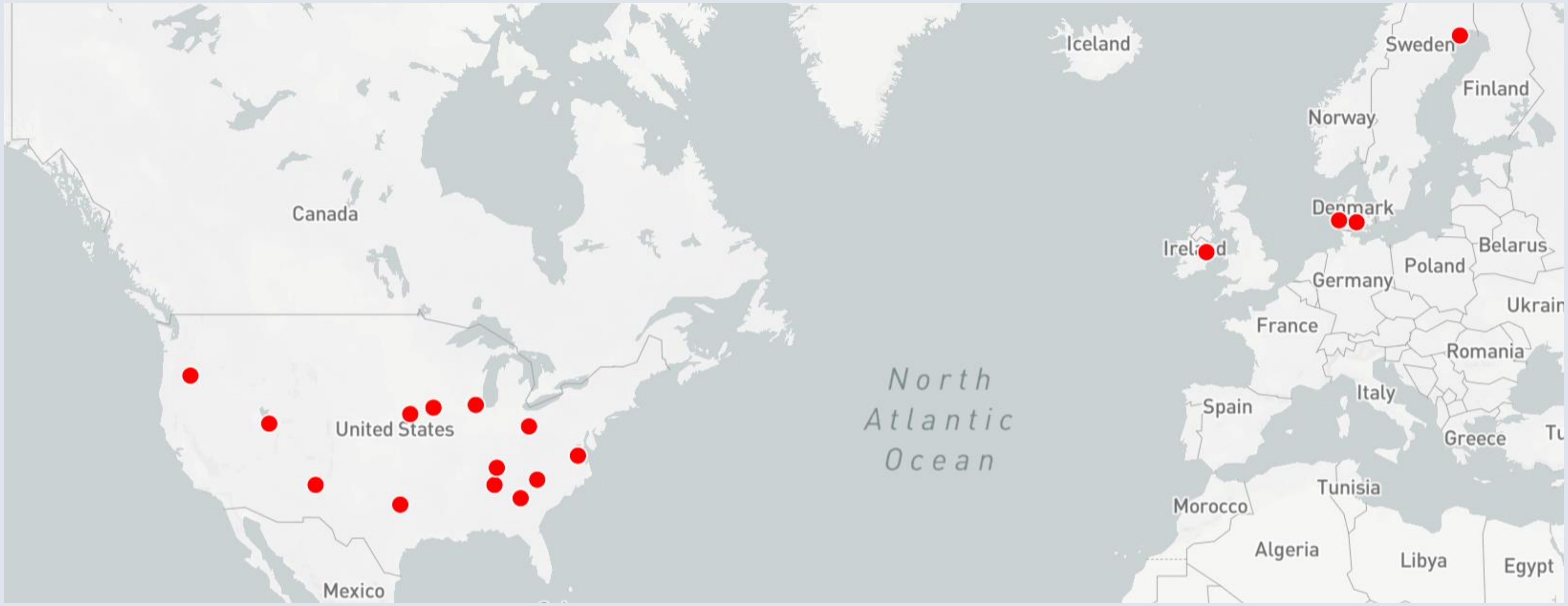
Dynamically Rendering the Graph


Nathan
Yesterday at 5:30pm via iPhoto
At the summit — at Charlotte Dome.
Like · Comment
Carol and 3 others like this.
Alice
how was it? is it worth the long approach?
Like · More · Just now
Write a comment...



- 1 billion queries/second
- many petabytes of data

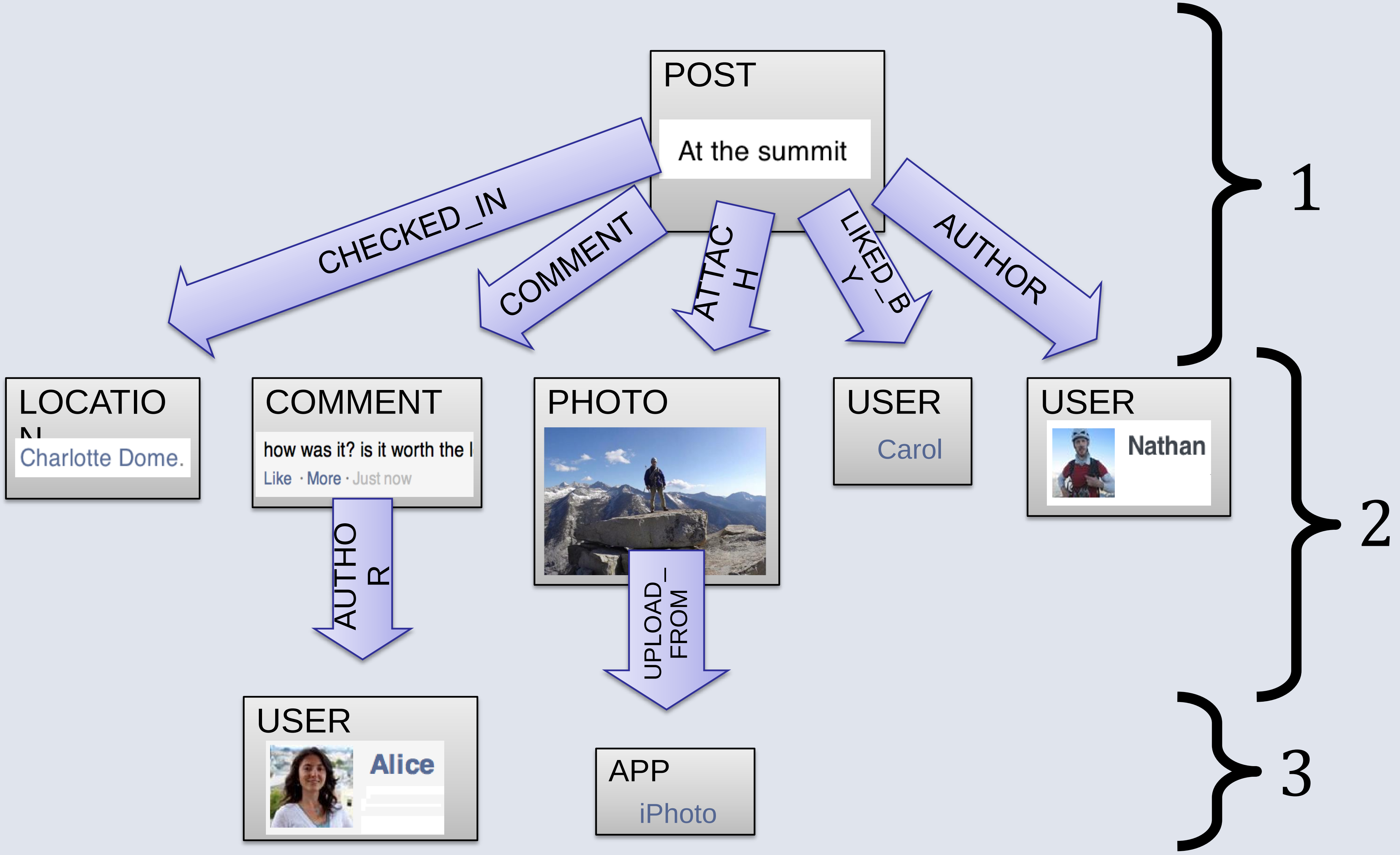
facebook data centers



What Are TAO's Goals/Challenges?

- Efficiency at scale

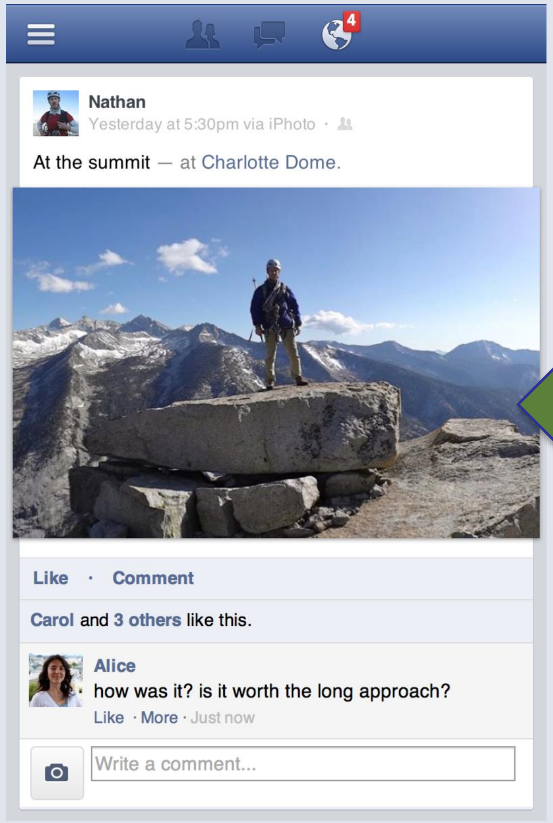
Dynamic Resolution of Data Dependencies



What Are TAO’s Goals/Challenges?

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

Graph in Memcache



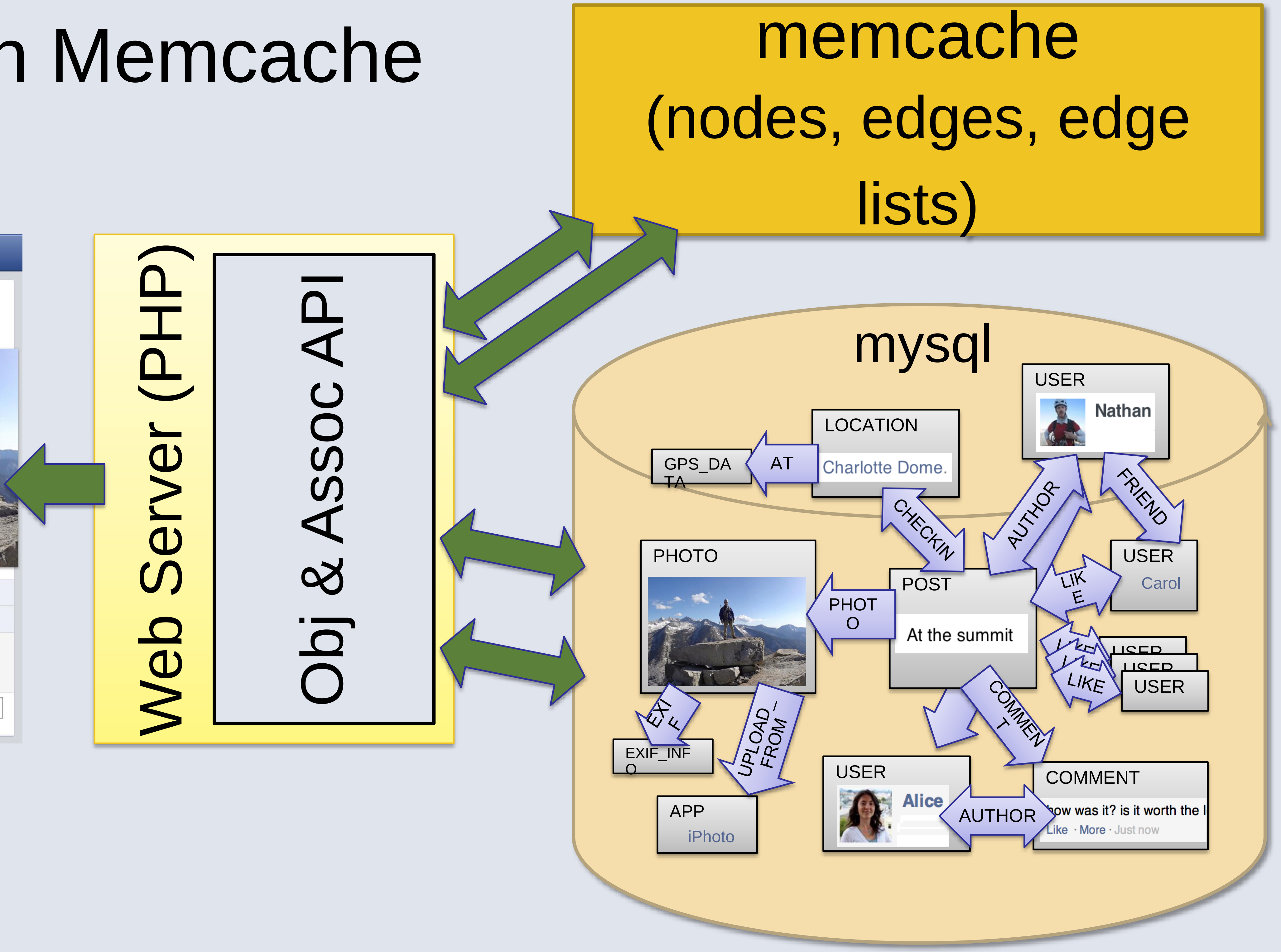
Nathan
Yesterday at 5:30pm via iPhoto ·
At the summit — at Charlotte Dome.

Like · Comment

Carol and 3 others like this.

Alice
how was it? is it worth the long approach?
Like · More · Just now

Write a comment...

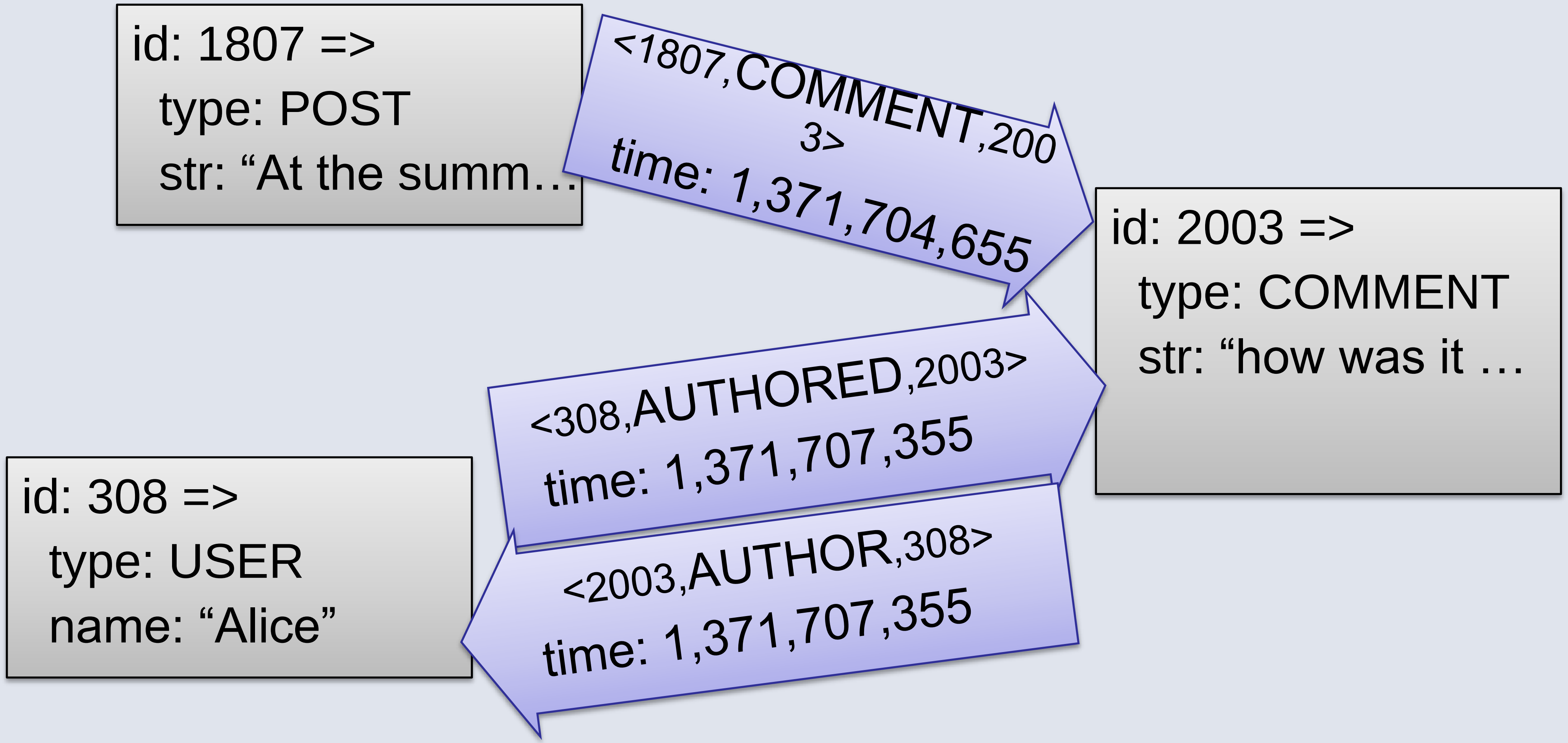


Objects = Nodes

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

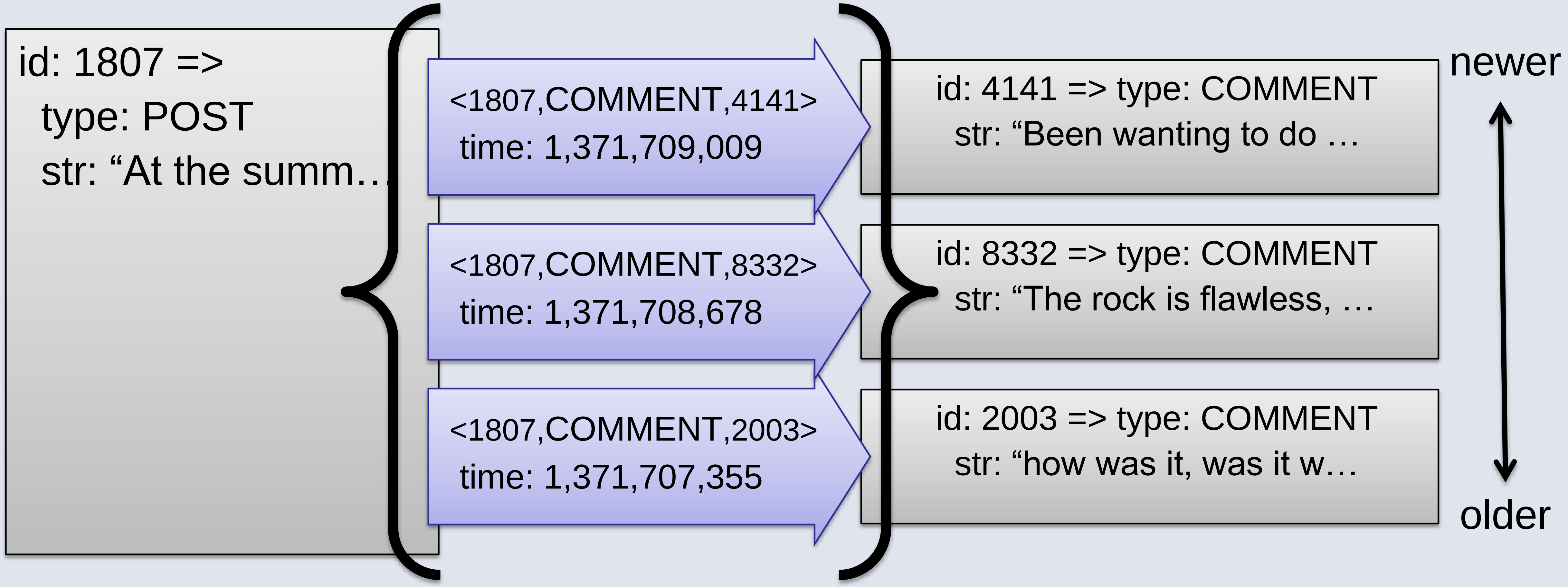
Associations = Edges

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



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%

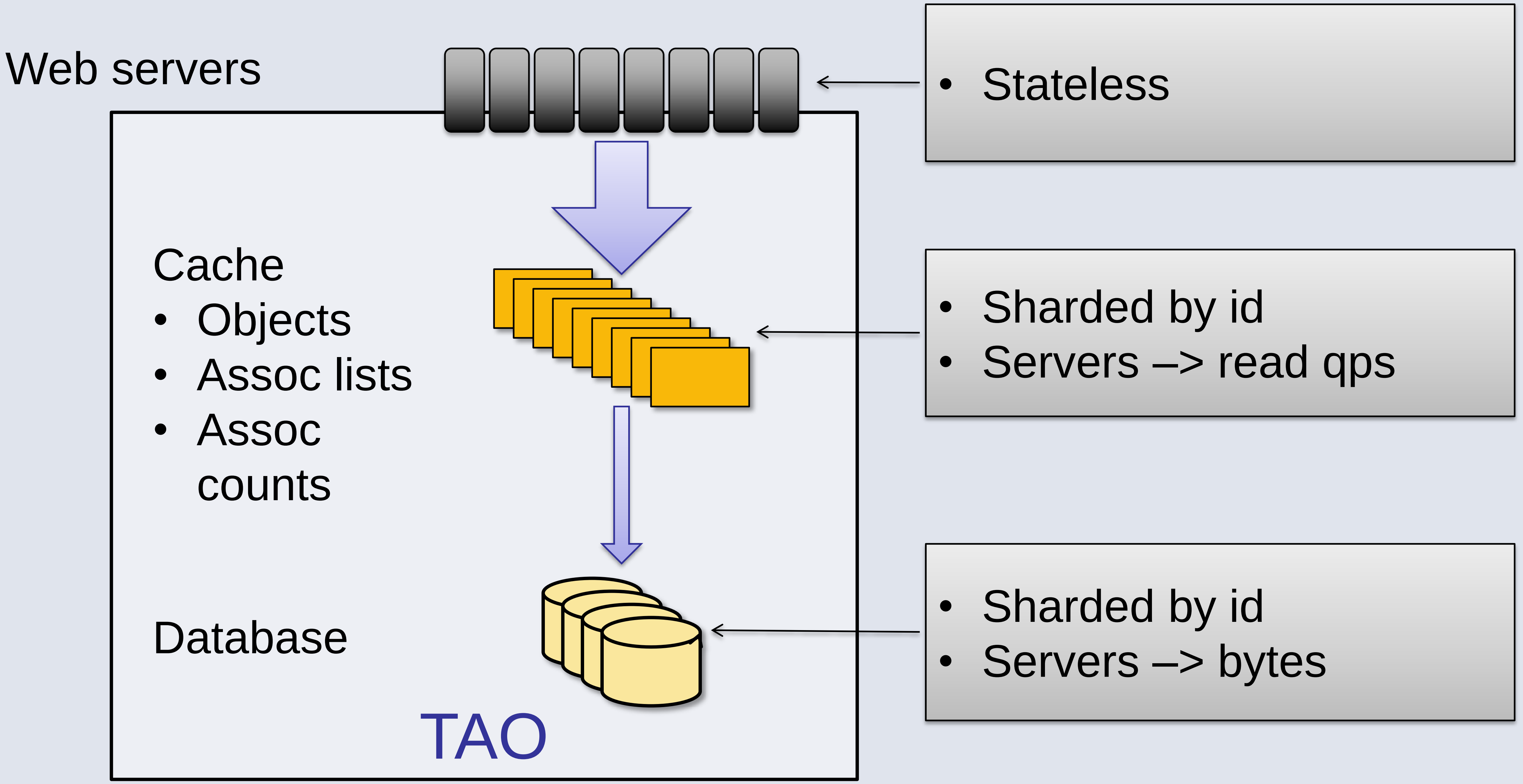
- Set and delete for associations

assoc_add	52.5%
assoc_del	8.3%

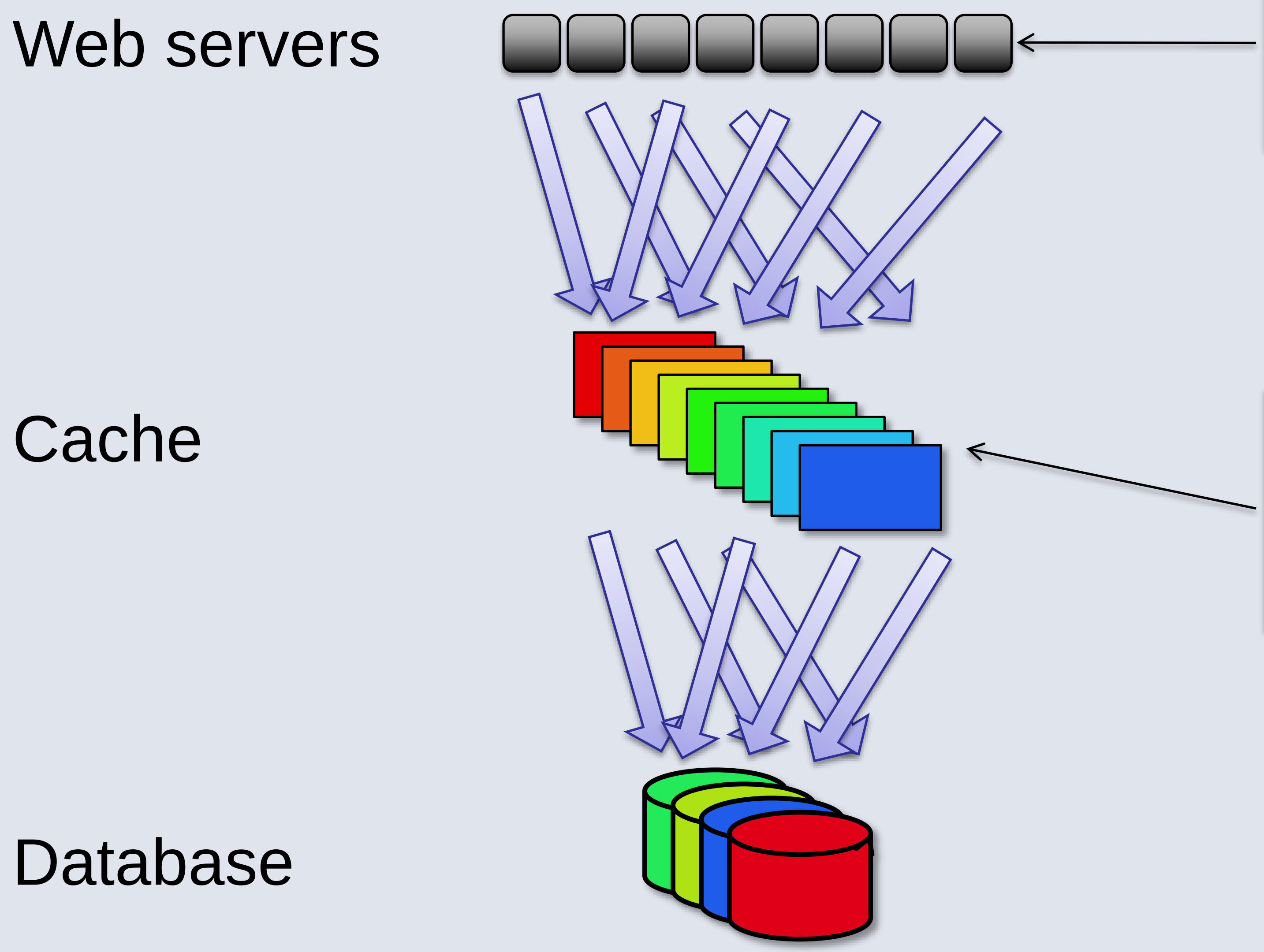
What Are TAO's Goals/Challenges?

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

Independent Scaling by Separating Roles



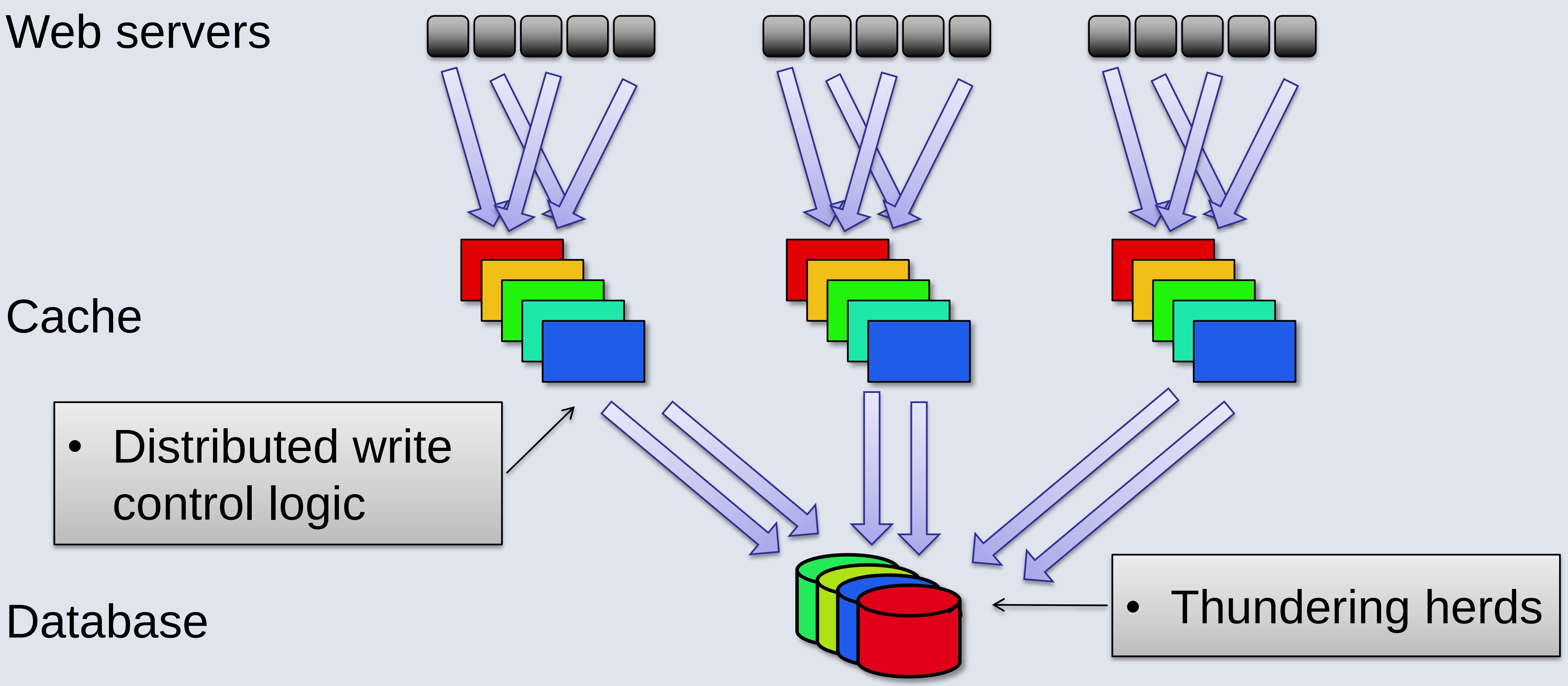
Subdividing the Data Center



- Inefficient failure detection
- Many switch traversals

- Many open sockets
- Lots of hot spots

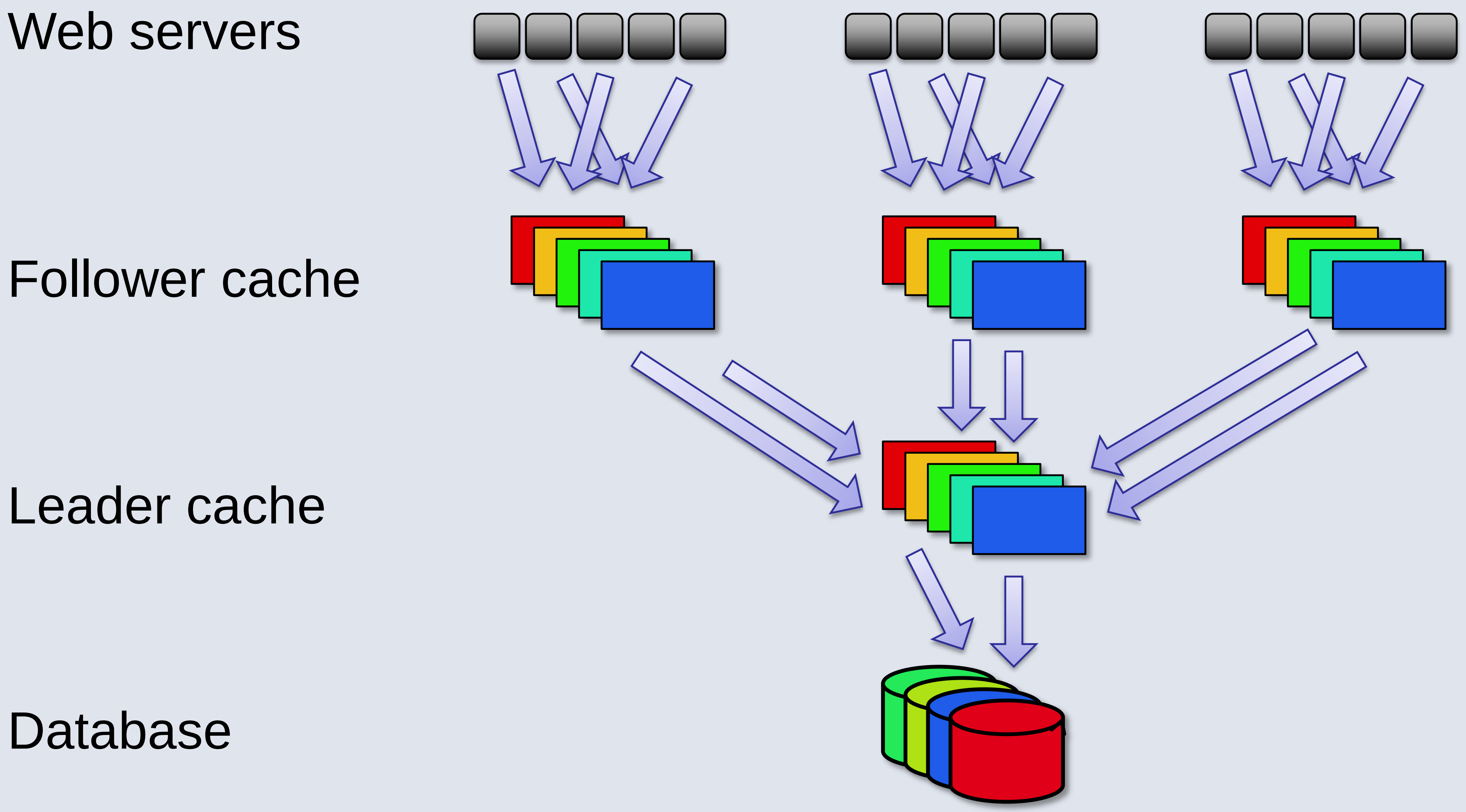
Subdividing the Data Center



- Distributed write control logic

- Thundering herds

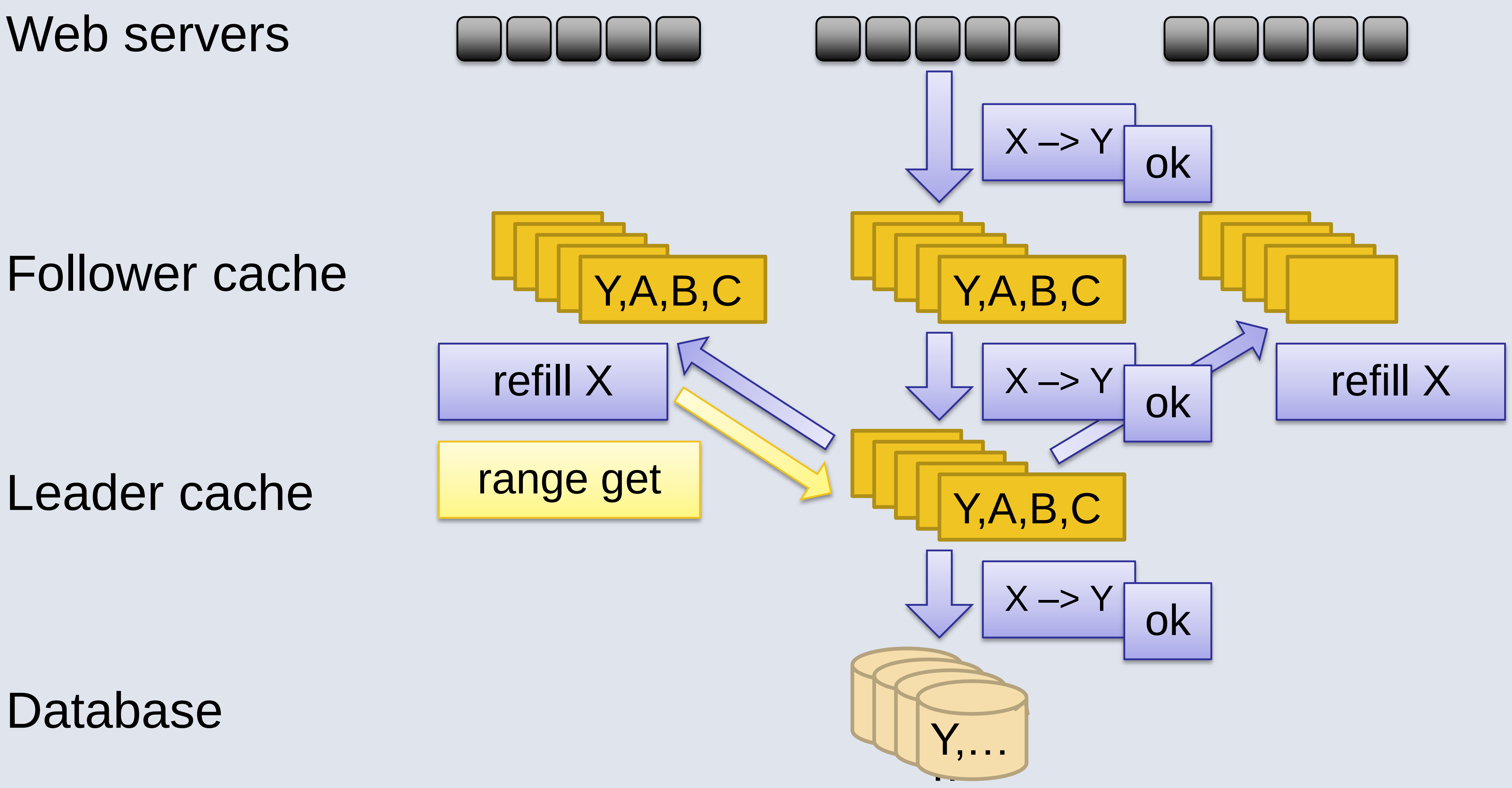
Follower and Leader Caches



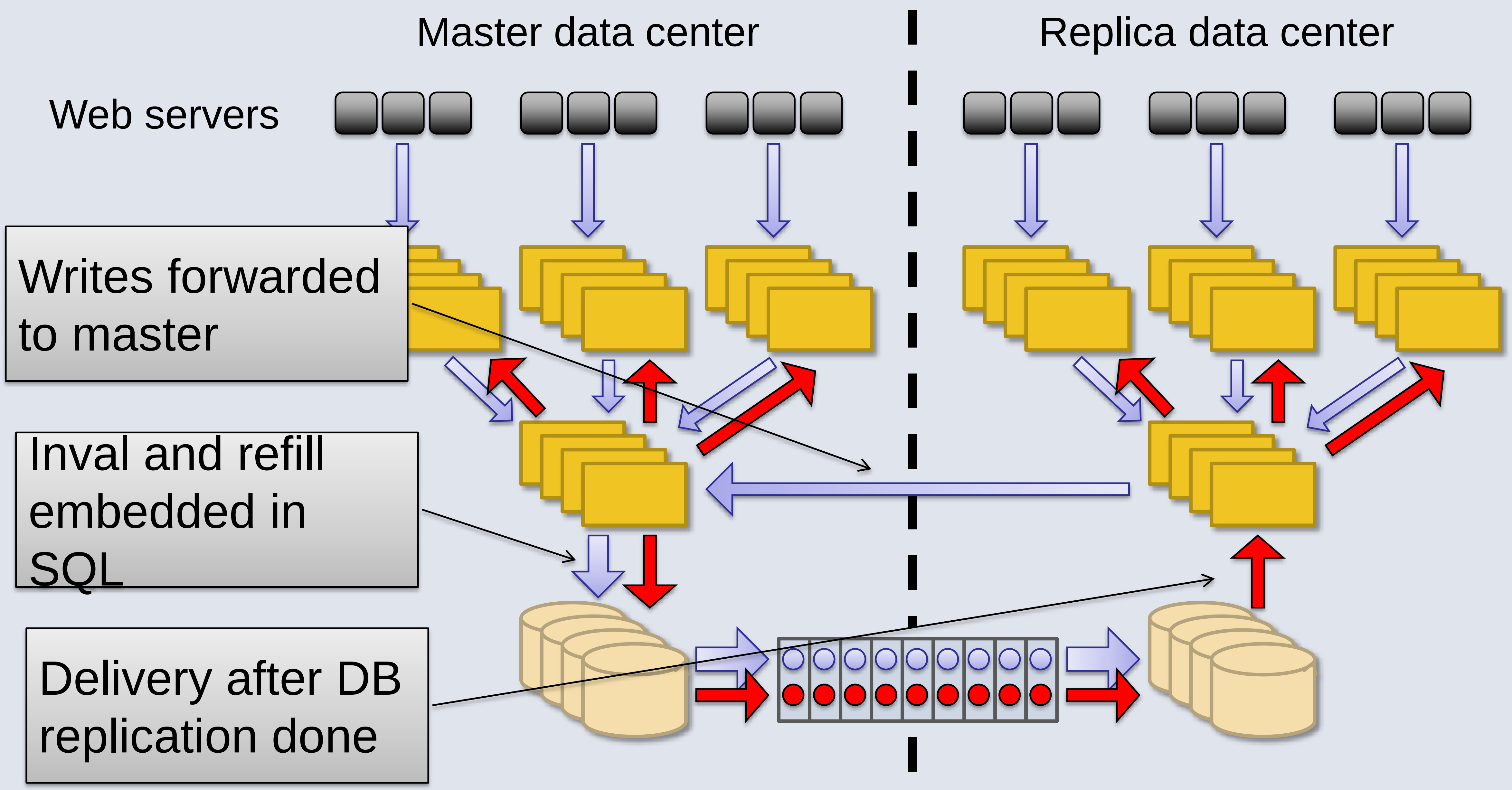
What Are TAO’s Goals/Challenges?

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

Write-through Caching – Association Lists



Asynchronous DB Replication



Writes forwarded to master

Inval and refill embedded in SQL

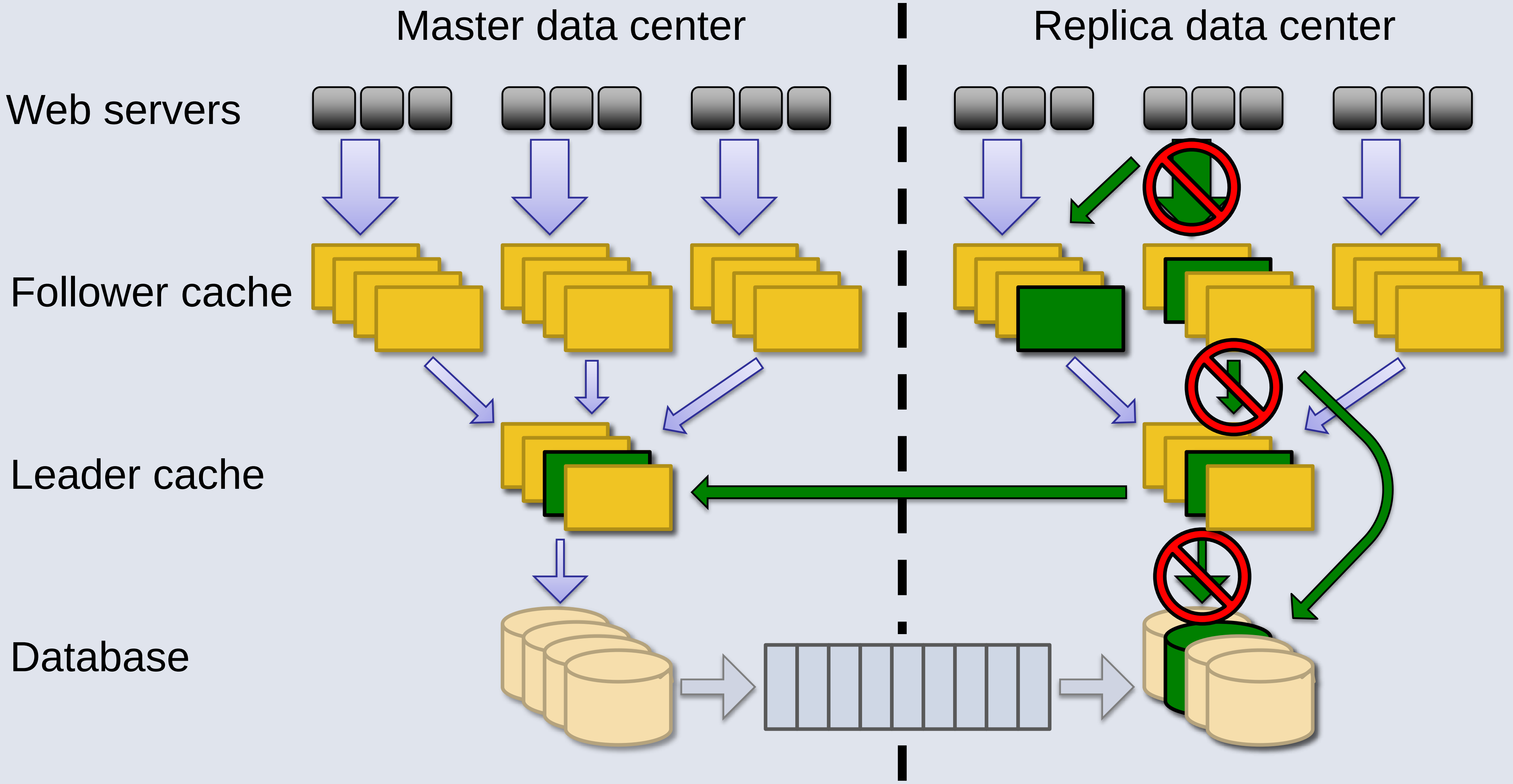
Delivery after DB replication done

What Are TAO’s Goals/Challenges?

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

facebook

Improving Availability: Read Failover



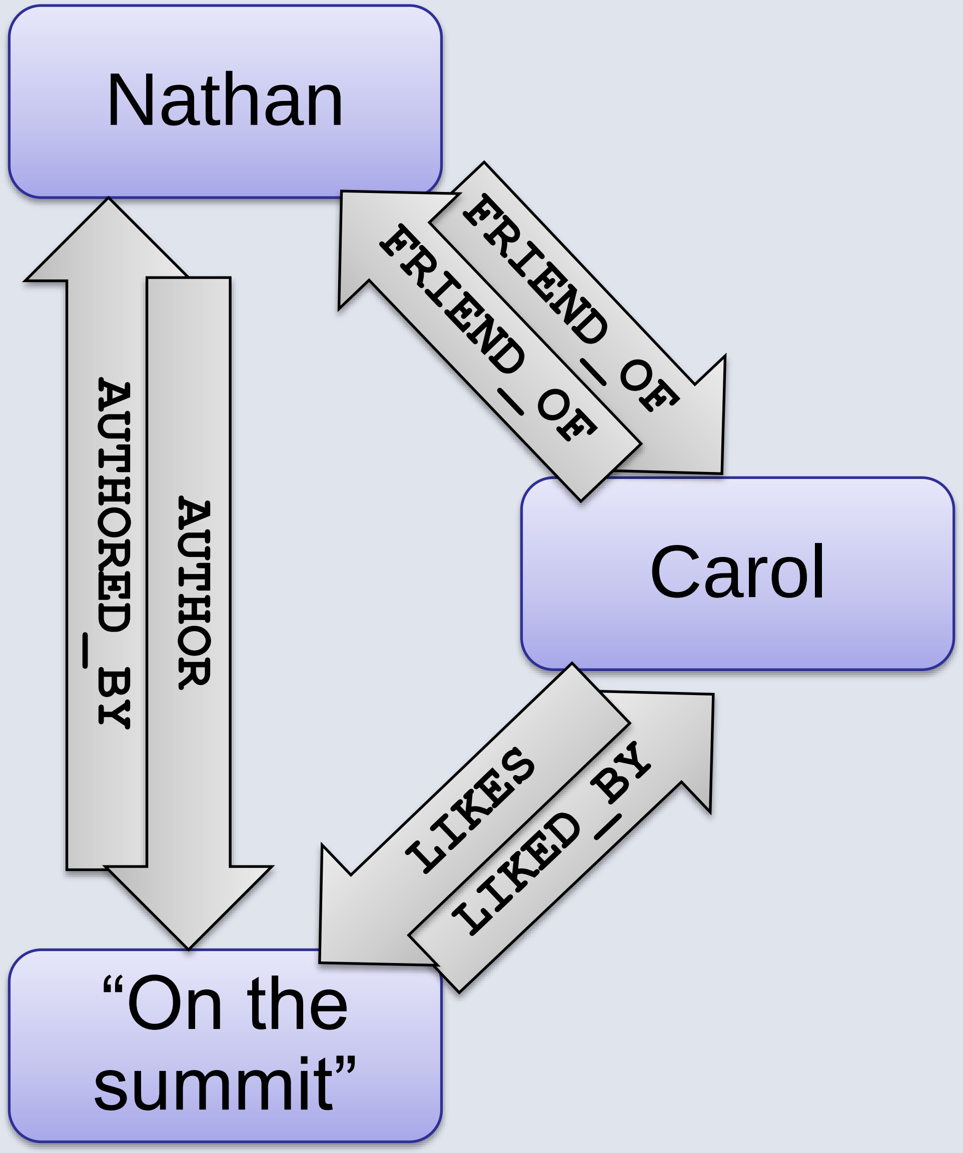
TAO Summary

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

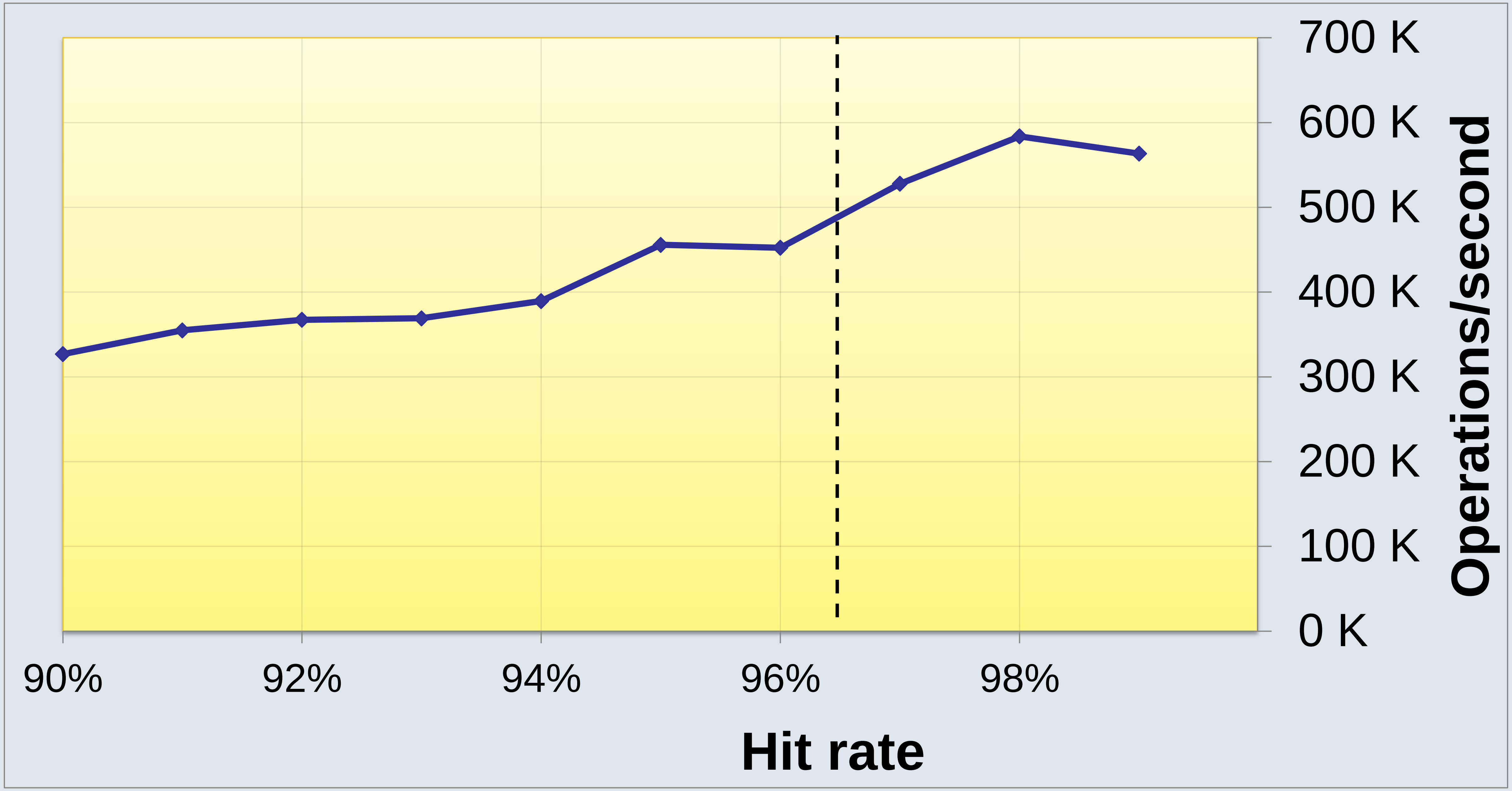
facebook

Inverse associations

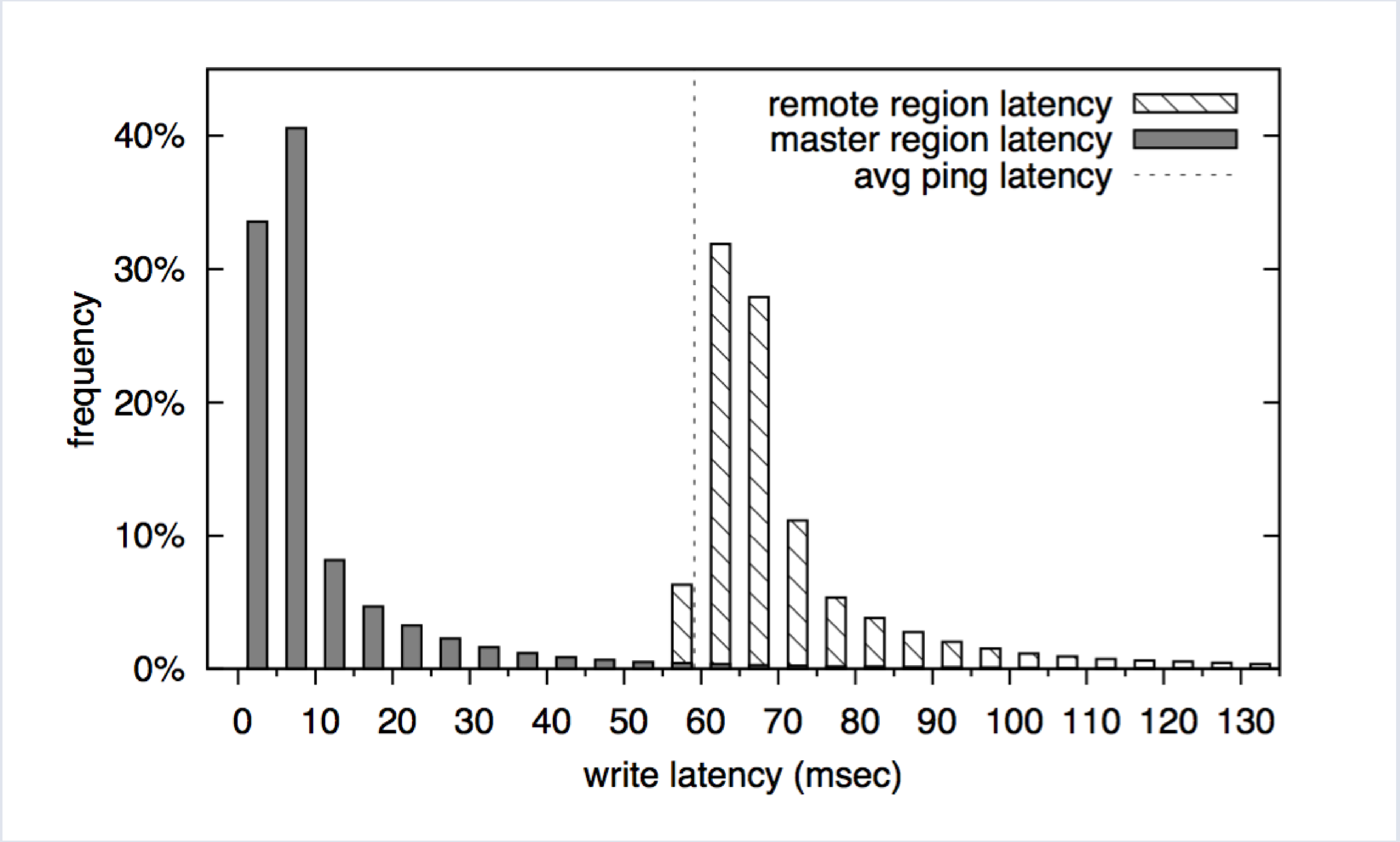
- Bidirectional relationships have separate *a*→*b* and *b*→*a* edges
  - inv_type(**`LIKES`**) = **`LIKED_BY`**
  - inv_type(**`FRIEND_OF`**) = **`FRIEND_OF`**
- Forward and inverse types linked only during write
  - TAO **`assoc_add`** will update both
  - Not atomic, but failures are logged and repaired



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