Megastore: Providing Scalable, Highly Available Storage for Interactive Services

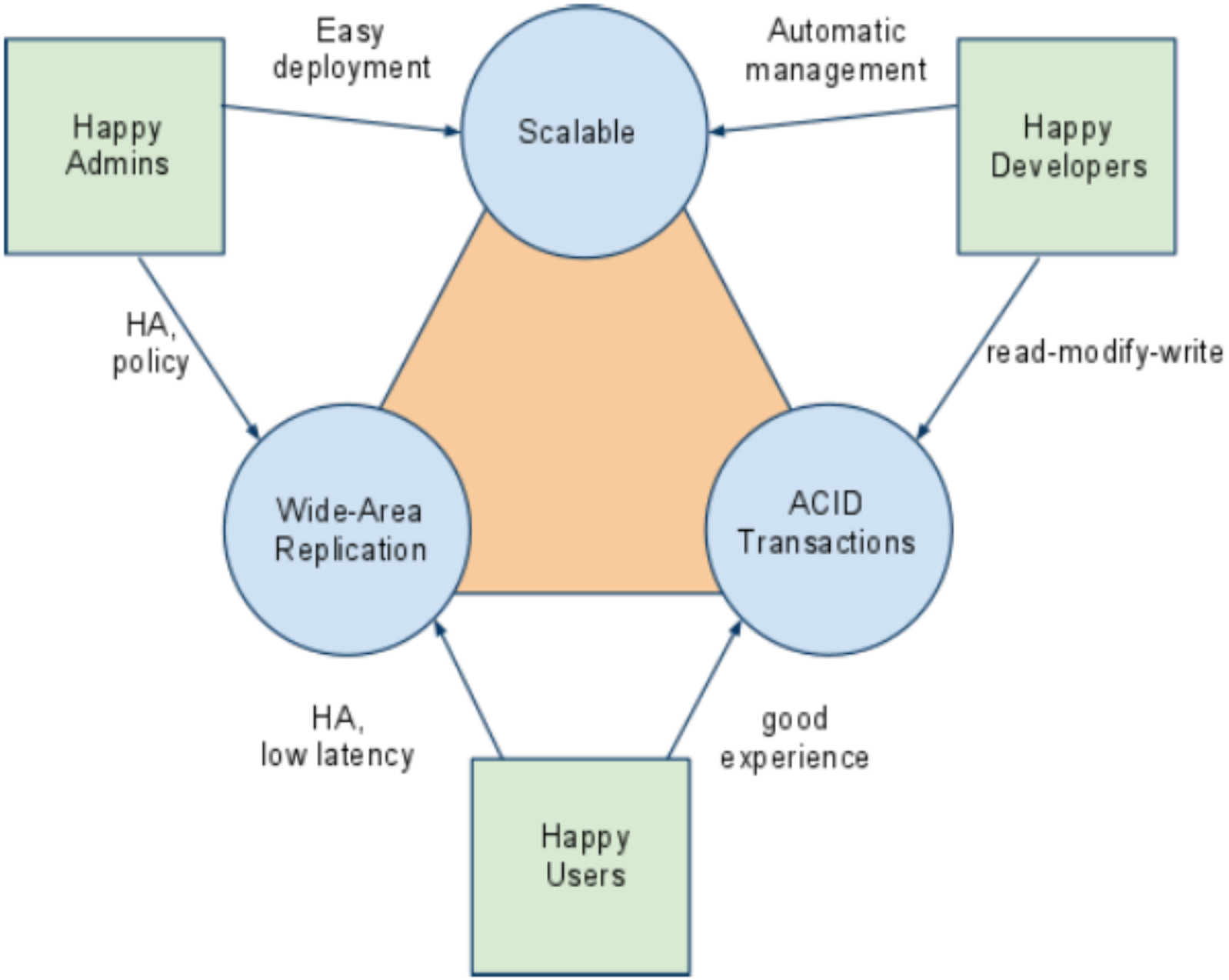
J. Baker, C. Bond, J.C. Corbett, JJ Furman, A. Khorlin, J. Larson, J-M Léon, Y. Li, A. Lloyd, V. Yushprakh

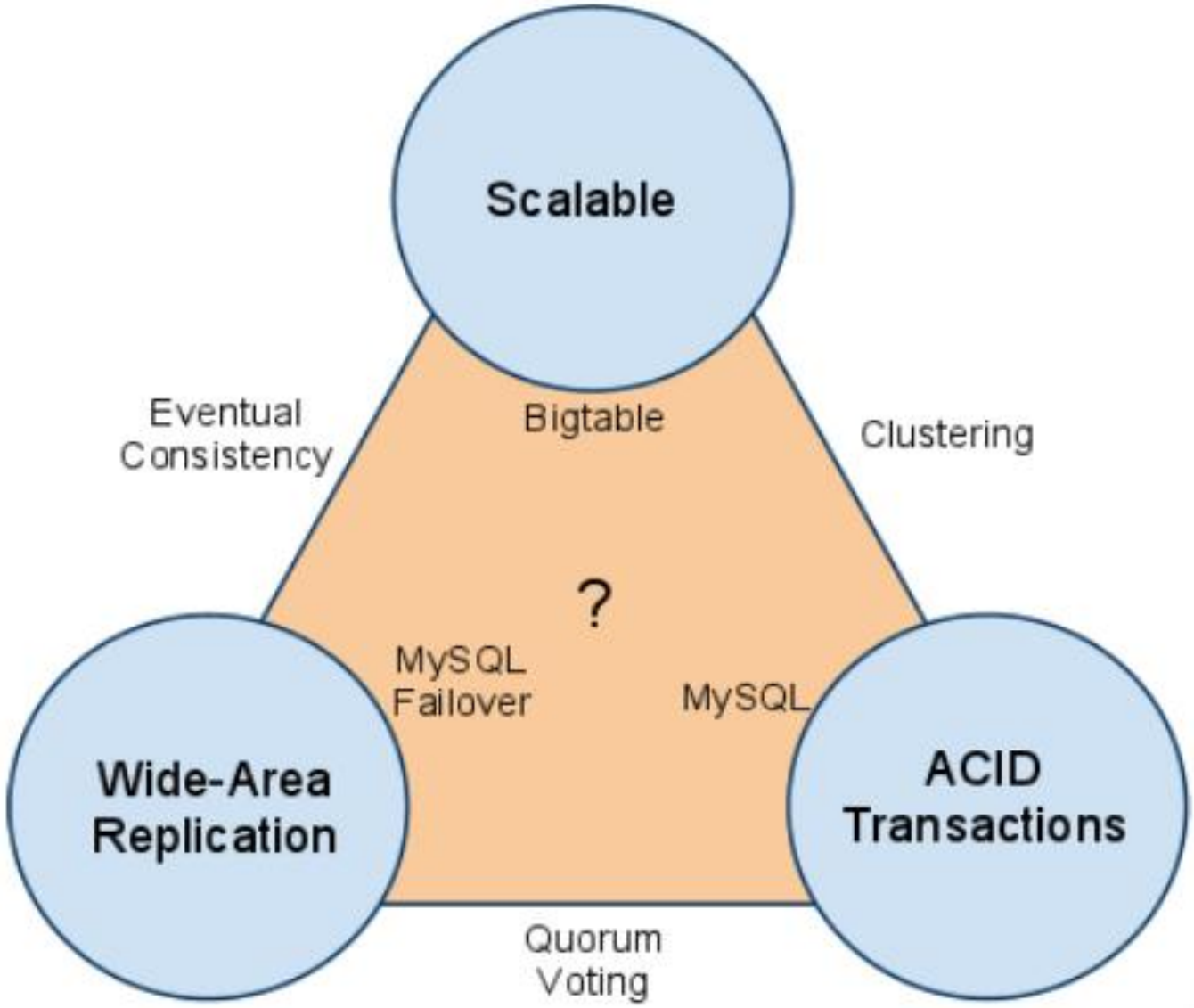
Google

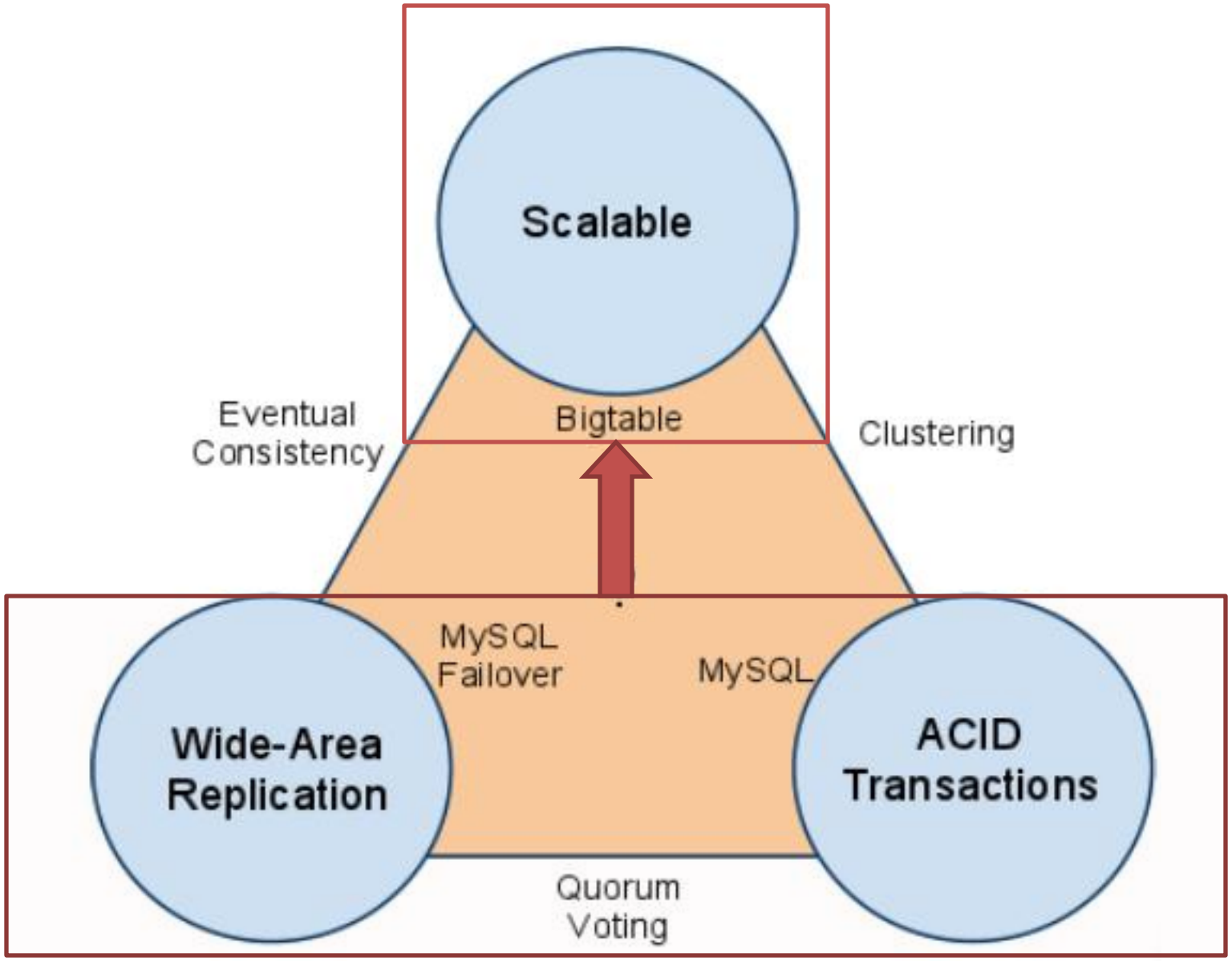
Presented by:
Sameer Agarwal
sameerag@cs.berkeley.edu

(some content in these slides is taken from the Megastore CIDR'11 Talk)

Megastore: Providing Scalable, Highly Available Storage for Interactive Services


Happy Admins
Easy deployment
Scalable
Automatic management
Happy Developers
HA, policy
read-modify-write
Wide-Area Replication
ACID Transactions
HA, low latency
good experience
Happy Users


Scalable
Eventual Consistency
Bigtable
Clustering
?
MySQL Failover
MySQL
Wide-Area Replication
ACID Transactions
Quorum Voting

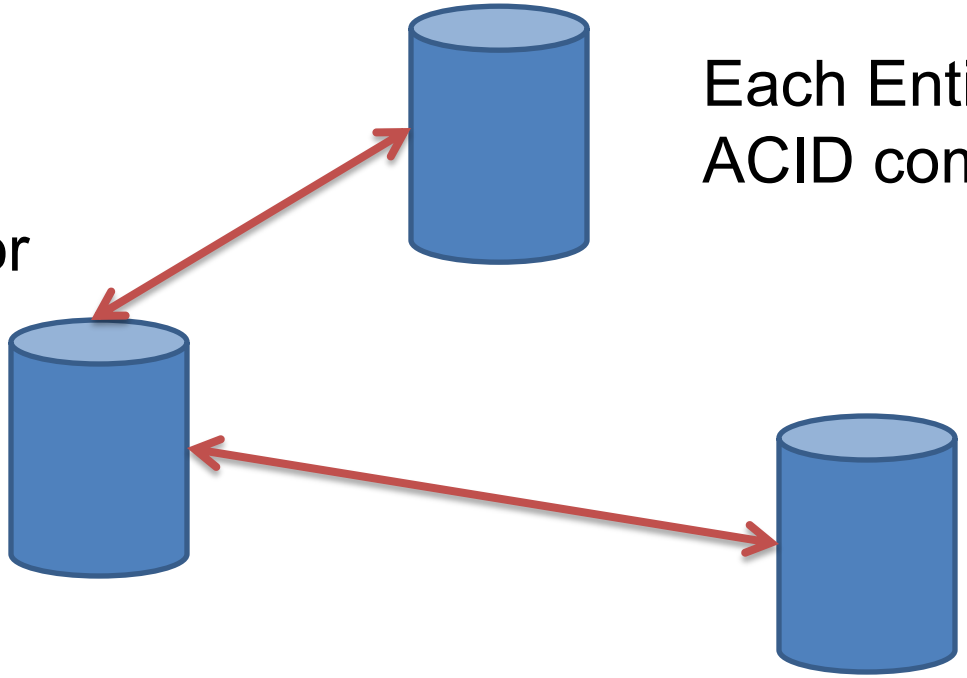

Scalable
Eventual Consistency
Bigtable
Clustering
MySQL Failover
MySQL
Wide-Area Replication
ACID Transactions
Quorum Voting

Megastore

Partition Data in Entity Groups [Helland '07]

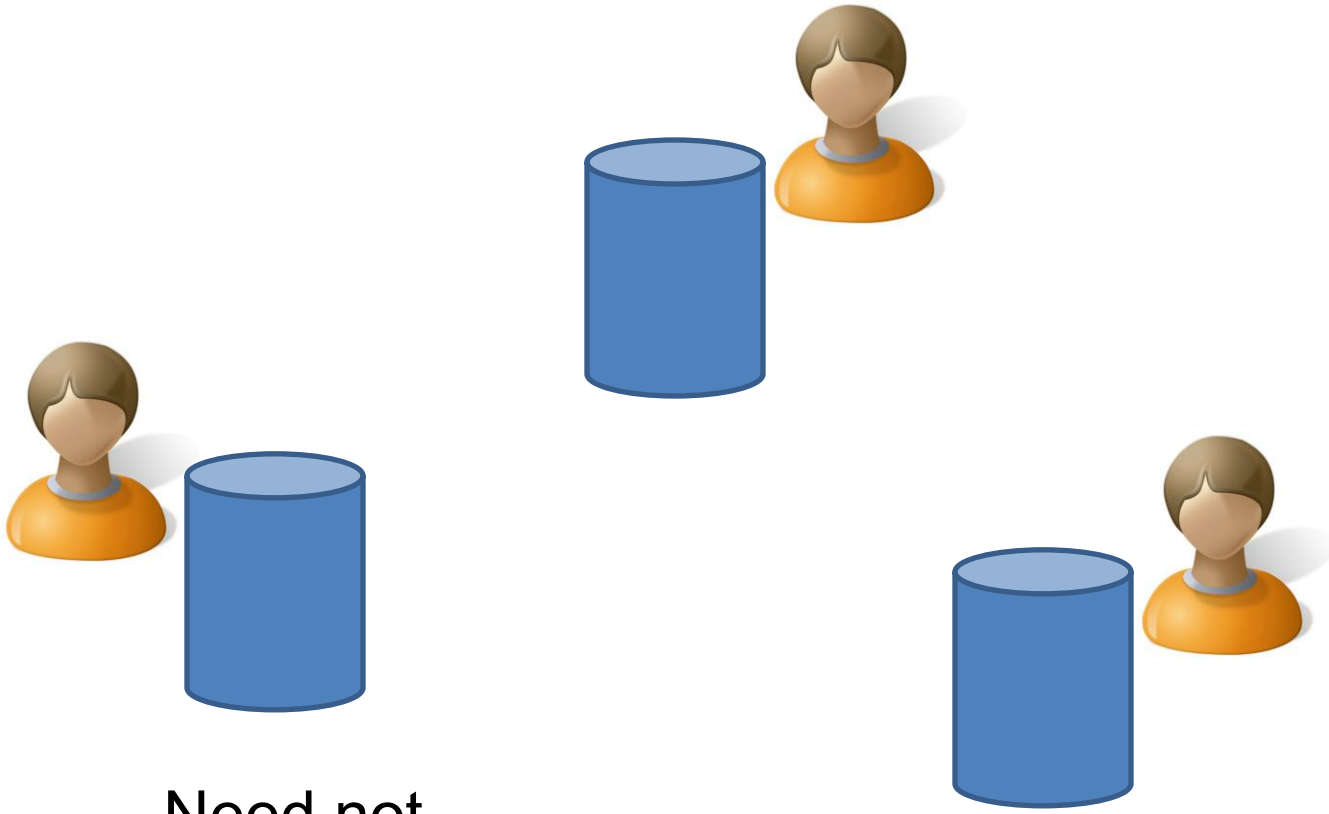
- Entity groups can be thought of as *small* databases.

Operations between Entity groups rely on expensive 2-PC or Asynchronous Messaging



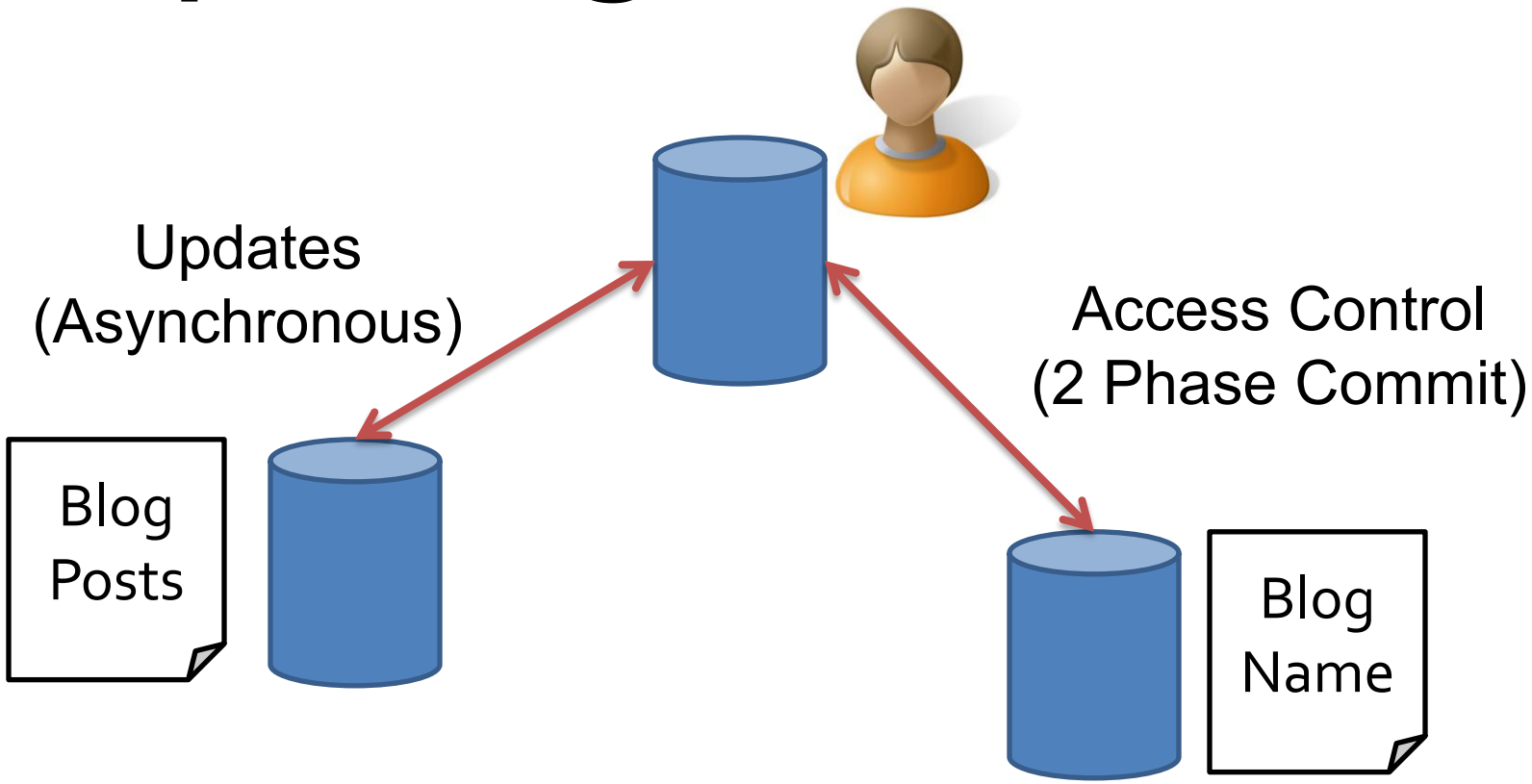
Each Entity group is ACID compliant

Examples of Entity Groups: Email

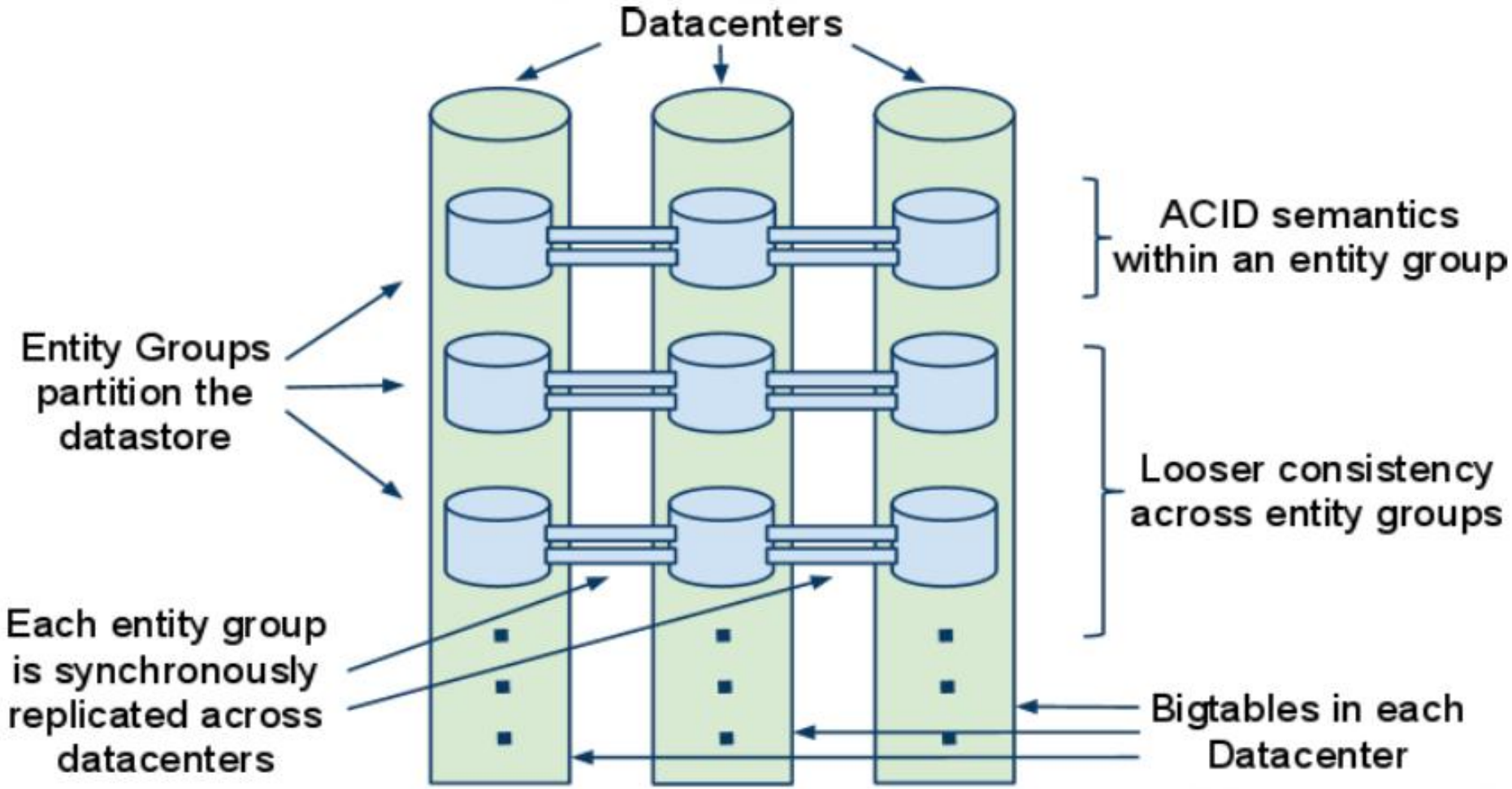


Need not Communicate Interactively

Examples of Entity Groups: Blogs



Architecture



Tweaking Paxos for Optimal Latency

- Replicates transaction logs on each write.
- **Writes**: Single WAN RTT on an average.
  - *Piggybacking prepares and accepts*.
- **Reads**: Zero WAN RTT on an average
  - *Coordinator server (a per-replica bitmap that is invalidated on faults.)*

Declarative Schema

- Applications have full control over the query plan (eg. indices, join implementations etc.)
- Applications have fine-grained control over physical data-placement (eg. STORING clause)

Summary

Scalability

- Partition data into entity groups and store them in Bigtable.

ACID Transactions

- Write-Ahead Log per entity group.
- 2-Phase Commits or Queues between entity groups.

Wide-Area Replication

- Paxos for replicating data.
- Tweaked for Optimal Latency

Comments/Critiques

Can All Data Be Partitioned into Entity Groups?

- **How about Complex Social Network Graphs?**
  - **The *new* Facebook Ticker:** Displays real time updates from your friends. User profile based entity groups might be an expensive deal!
  - **Twitter Feeds**: Displays real time messages from people you follow.

Does Megastore makes the right trade-offs?

- **Megastore favors consistency over performance**
  - average read latencies of *tens* of milliseconds
  - average write latencies of *100-400* milliseconds
  - a *few* writes per second per entity group

- Googlers find the latency tolerable but often have to hide write latency from users and choose entity groups carefully.

- Facebook application requires 4ms reads & 5ms writes[1]

[1]https://www.facebook.com/video/video.php?v=695491248045

Why not have lots of RDBMS's?

- **Google picked Bigtable because:**

  it provides Load Balancing, Fault Recovery, Monitoring and Operational backing.

- **What if you choose a MySQL server for each entity group?**

Thank You!