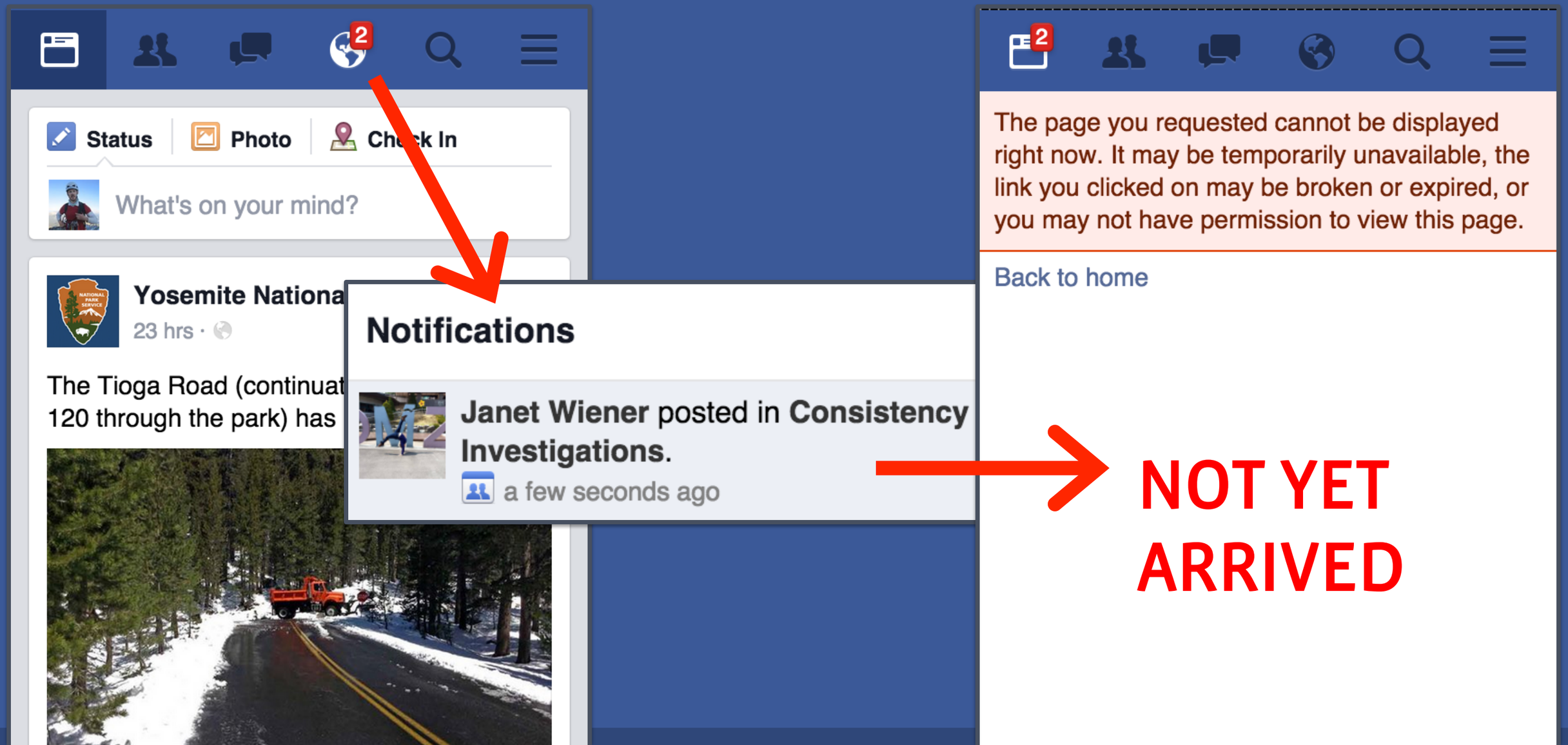
facebook

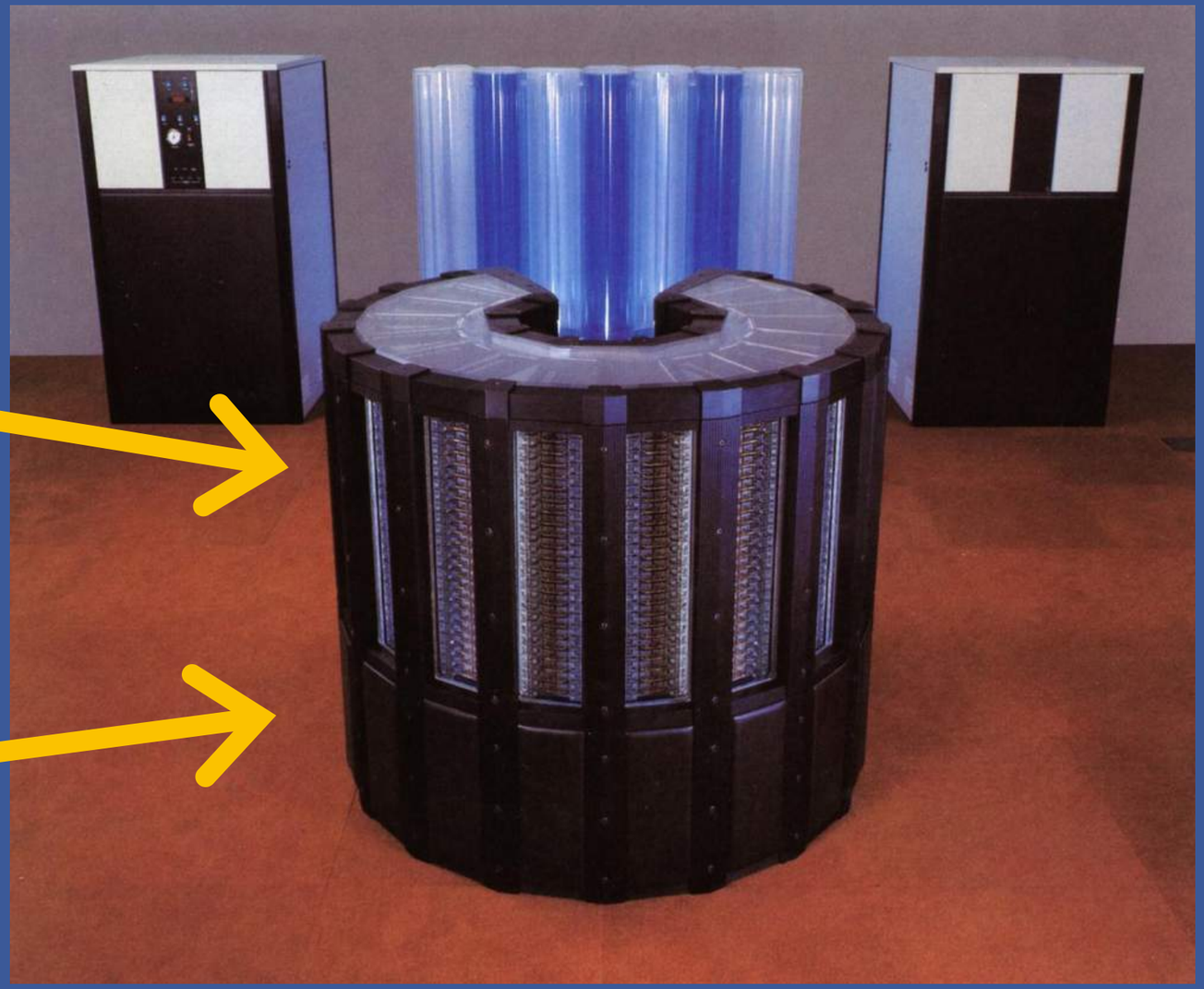
Challenges to Adopting Stronger Consistency at Scale

HotOS 2015
19 May

Phillipe Ajoux, *Nathan Bronson*, Sanjeev Kumar, Wyatt Lloyd, Kaushik Veeraraghavan

A user-visible inconsistency

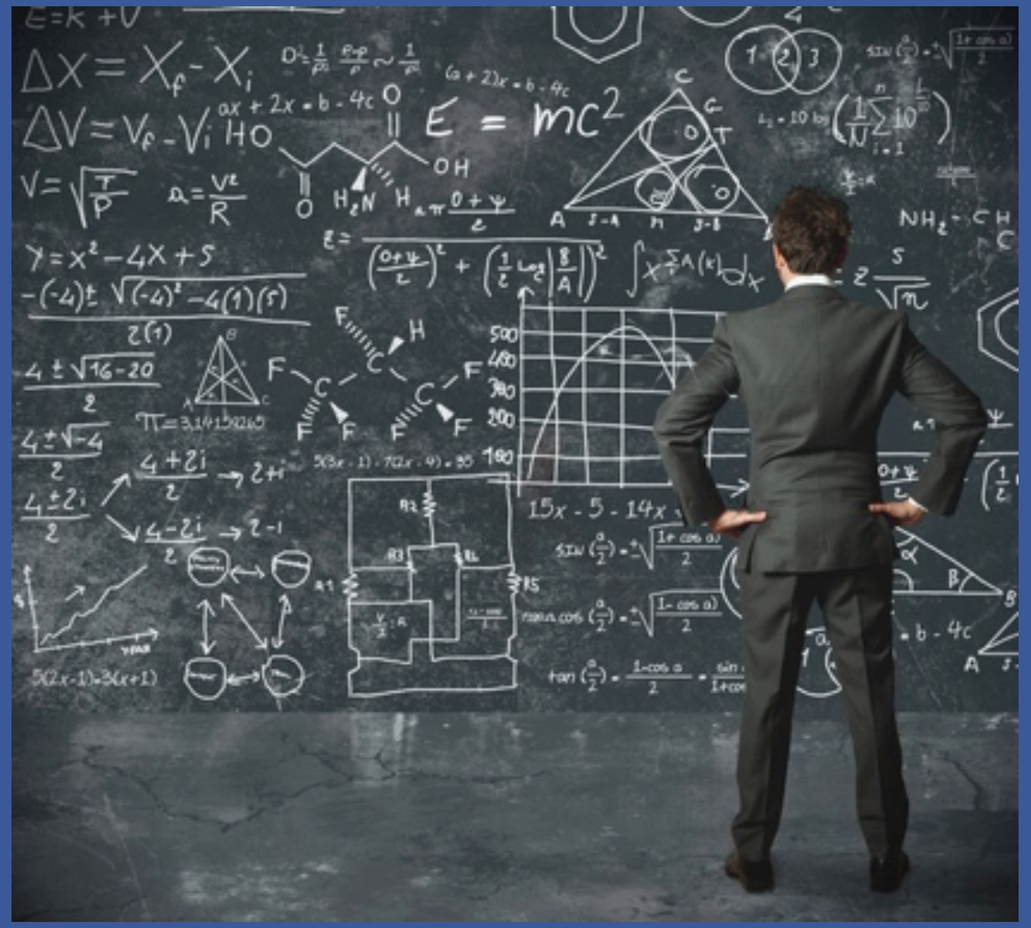




People should think FB runs on one computer

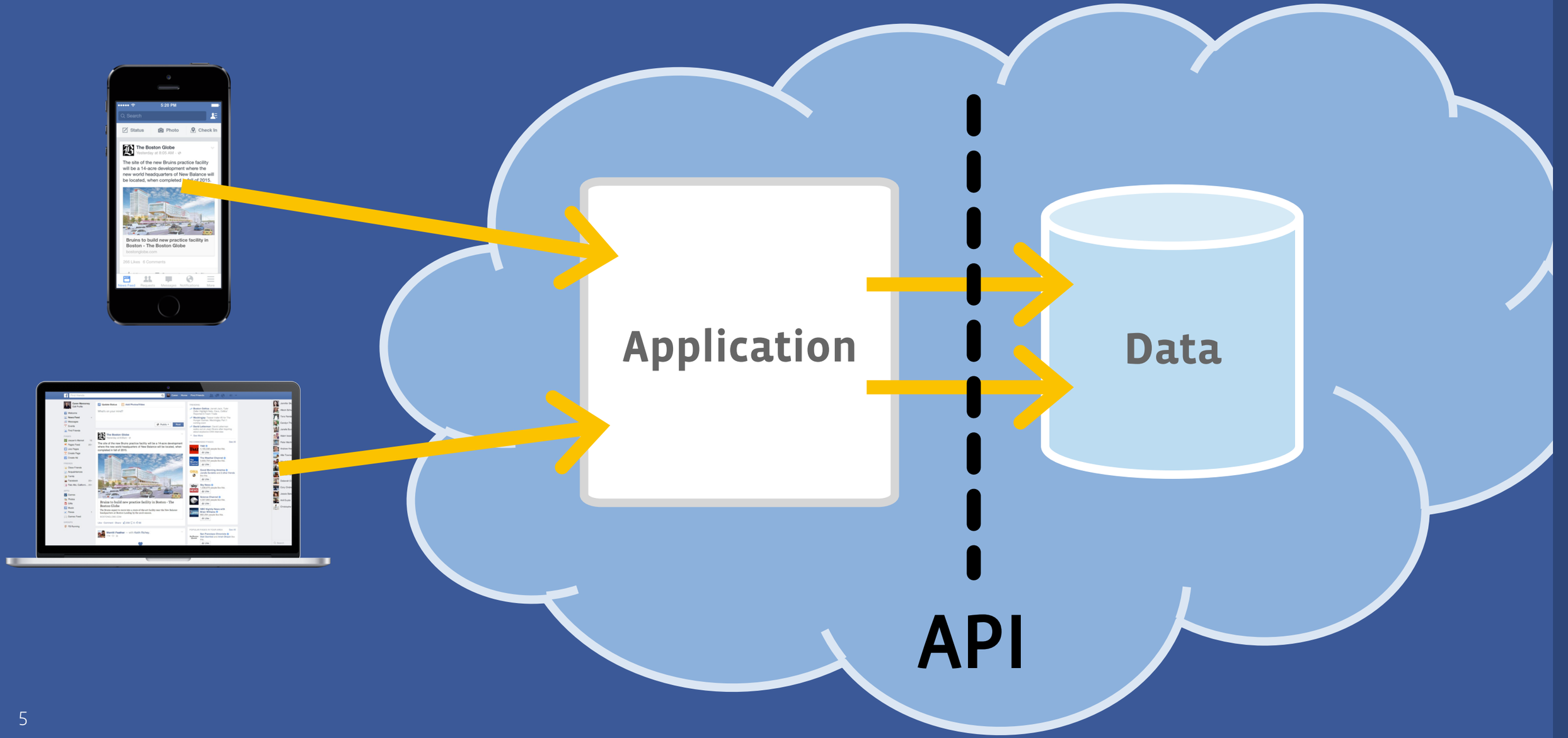
How can we fix inconsistencies?

or

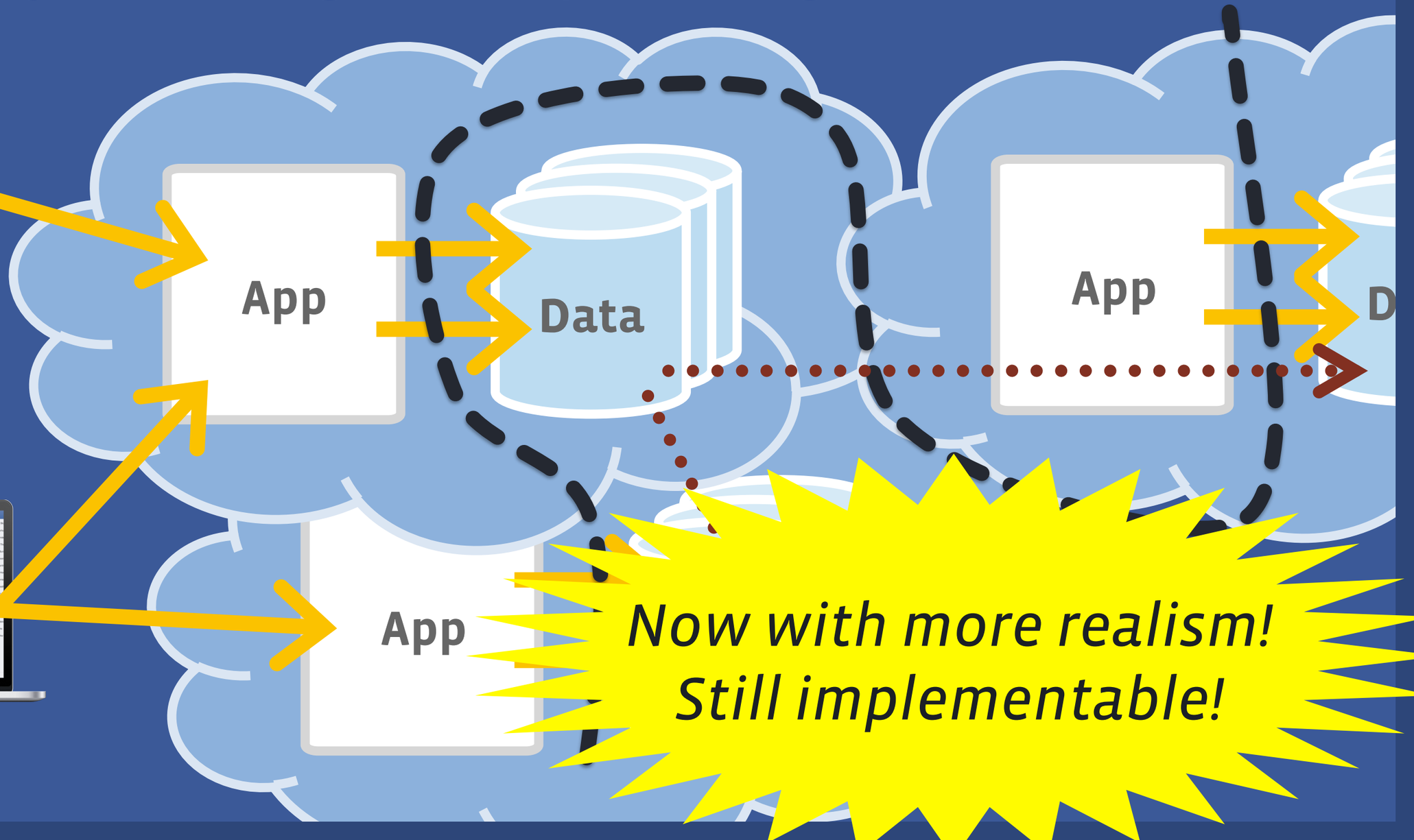


(or both)

Building a site (software engineering)



Scaling by sharding and replicating



Now with more realism!
Still implementable!

Why not strengthen the data store?

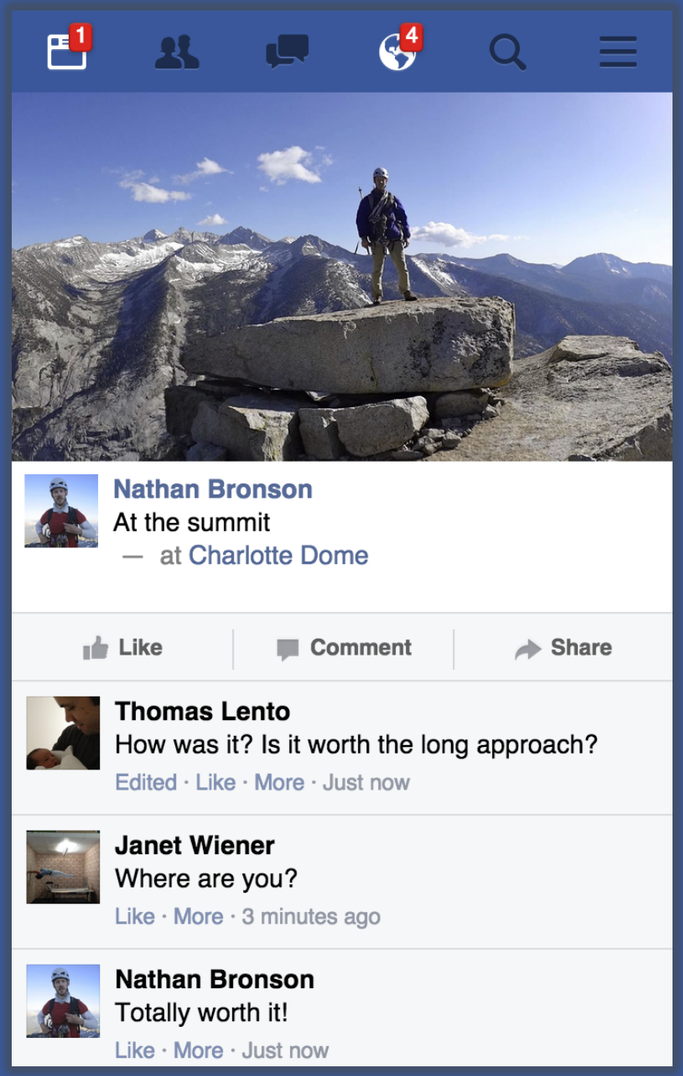
- Will it meet our requirements?
  - Outlier sensitivity – latency & availability
  - Pathological data access patterns
  - Low average latency needed for efficiency

} see the paper

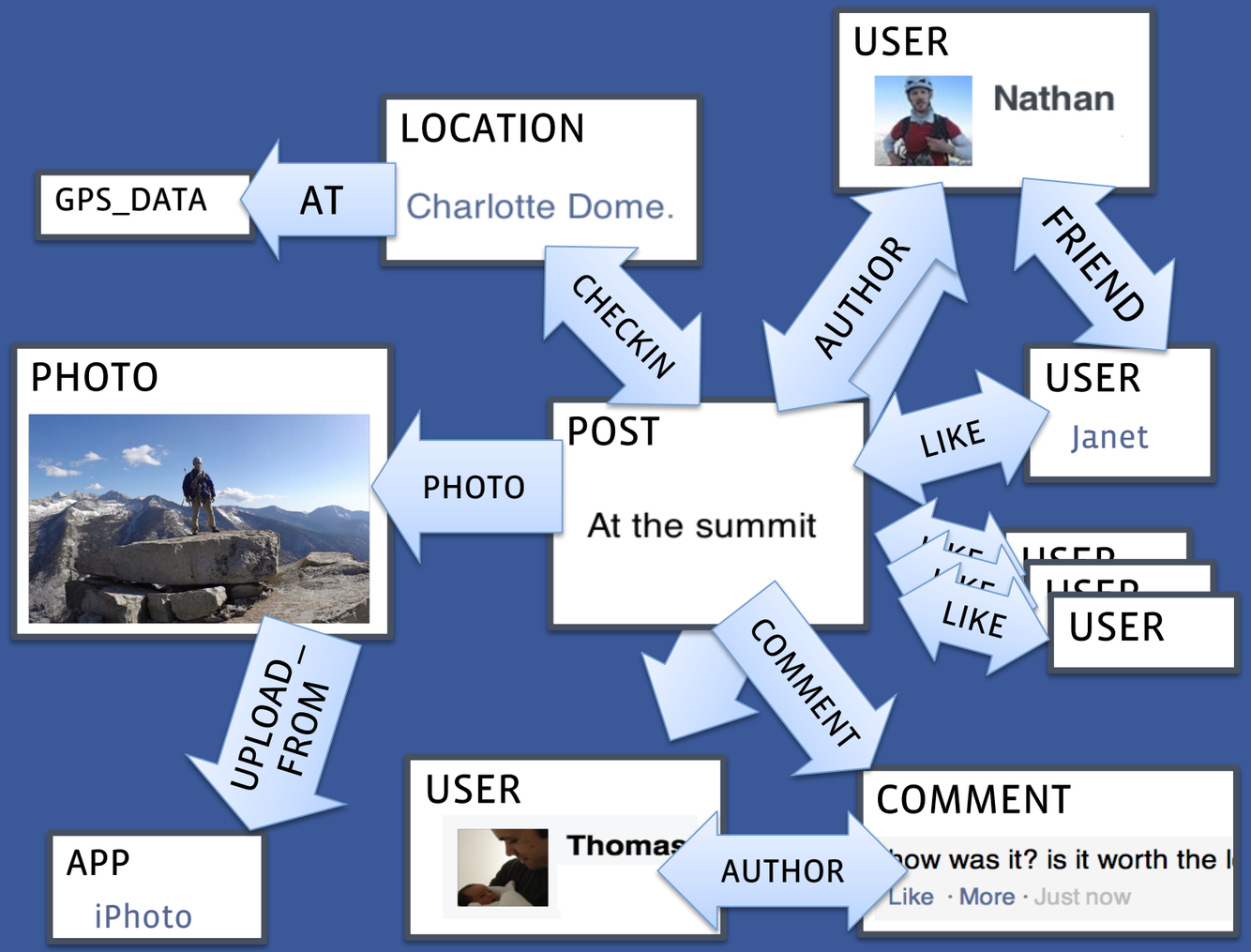
- What about data copies?
  - Lots of systems store ad-hoc data copies
  - Those systems are loosely coupled

} more slides

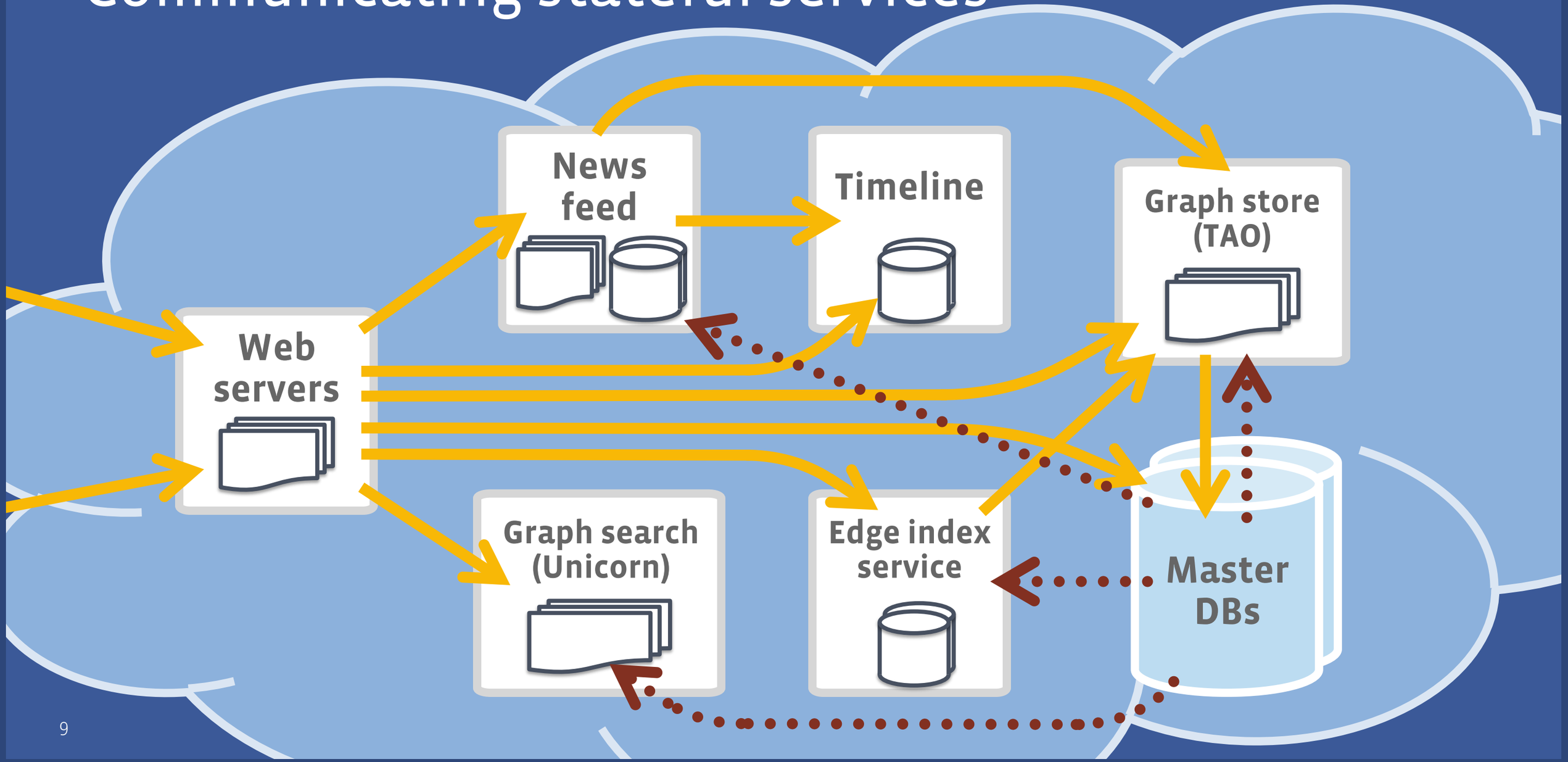
Social graph = types nodes + edges



=

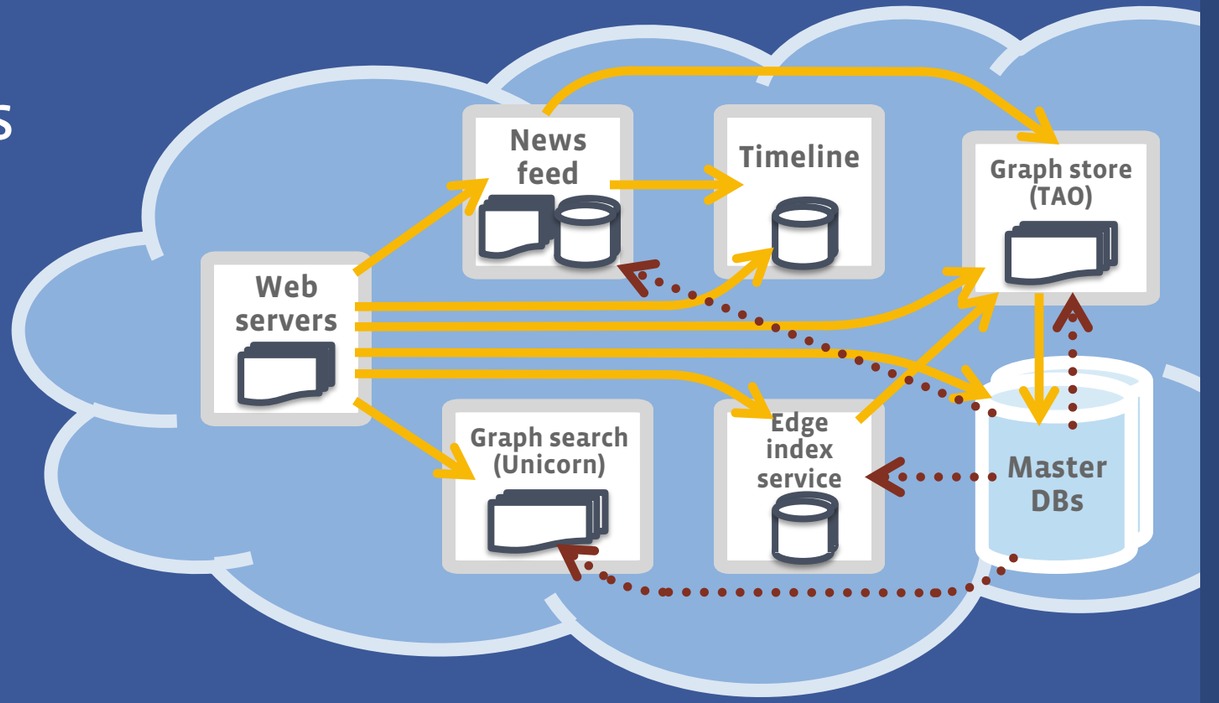


Communicating stateful services



Consistency glue challenges

- Multiple copies of the data
  - Copies are materialized query results
- Ad-hoc query languages
  - Service building block is RPC call
  - Didn’t design API as a language
- Loose coupling
  - Separate teams, different languages
  - Different deployment schedules, reliability goals



Techniques from tightly-coupled databases

Locality

- ✓ Denormalization
- ✓ Caching
- ✓ Materialized join views
- ✓ Secondary indices
- ✓ Covering indices
- ✓ Partial indices
- ✓ Stored procedures

Consistency glue

- ✗ Unified analyzable query language
- ✗ Two phase locking
- ✗ Range locks, table locks
- ✗ Predicate locks
- ✗ Totally-ordered sequence numbers
- ✗ Foreign key constraints
- ✗ Linear durable log

What do I hope?

- Add tools for locality optimization to existing systems
- Make the problem topology more realistic
- Are there tools for end-to-end consistency?


facebook