Fear No More:
Embrace Eventual Consistency

Sean Cribbs
@seancribbs

Distributed Systems Experts


riak


riakcs

FEAR

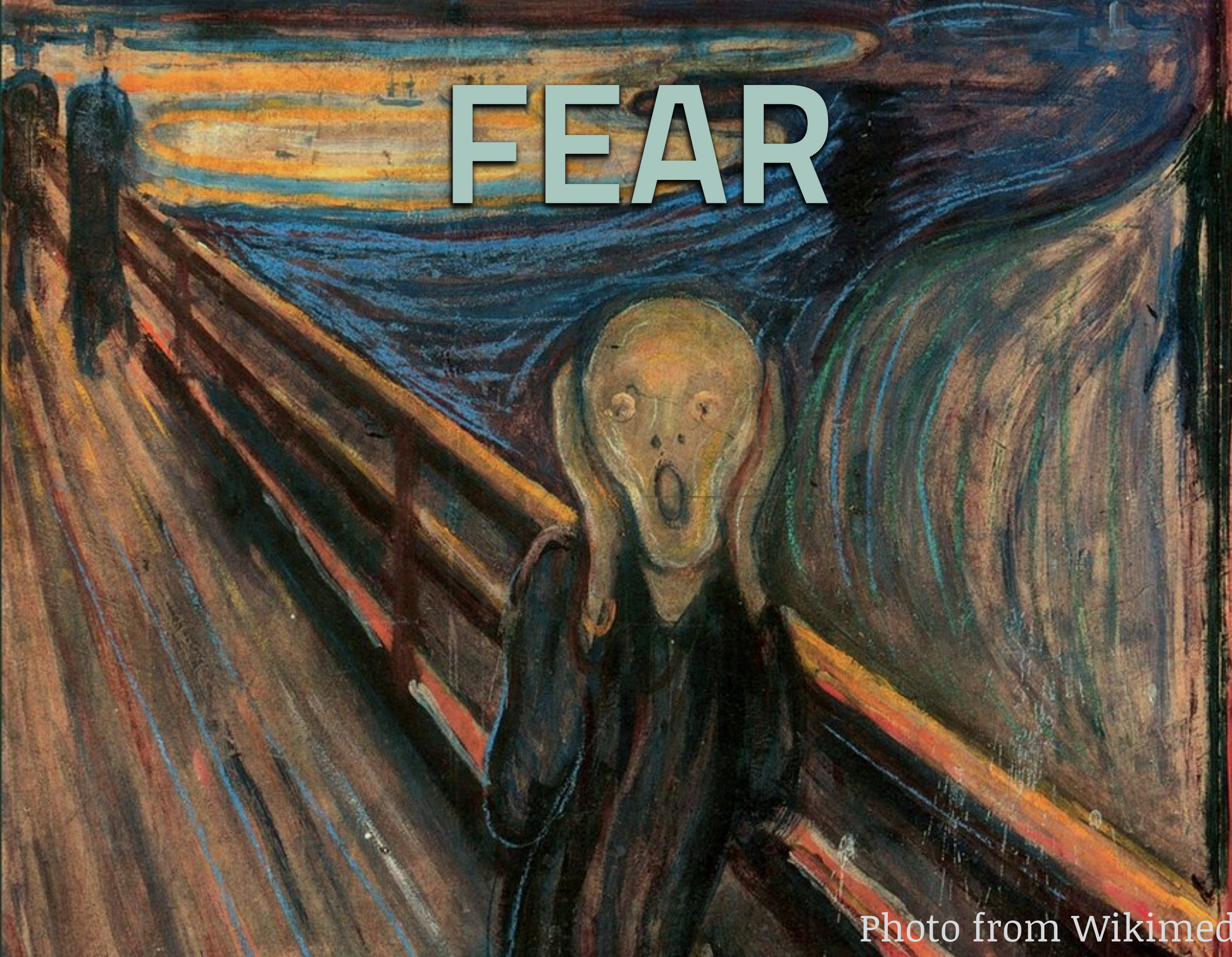


Photo from Wikimedia

ACID vs. BASE

ACID vs. BASE

- Strong consistency
- Isolation
- Focus on "commit"
- Nested transactions
- Conservative (pessimistic)

- Weak consistency
- Availability first
- Best effort
- Approximate answer
- Aggressive (optimistic)

Fox, Gribble, Chawathe, Brewer, Gauthier - *Cluster-Based Scalable Network Services* (SOSP97)

ACID vs. BASE

"*Inconsistency* is the worst thing that could happen."

"*Being unavailable* is the worst thing that could happen."

Why BASE / EC?

Why BASE / EC?

- "Omniscience" is **expensive** and **slow**.

Why BASE / EC?

- "Omniscience" is **expensive** and **slow**.
- **Availability** is often correlated to **revenue**.

Why BASE / EC?

- "Omniscience" is **expensive** and **slow**.
- **Availability** is often correlated to **revenue**.
- Failures happen **all the time**.

Why BASE / EC?

- "Omniscience" is **expensive** and **slow**.
- **Availability** is often correlated to **revenue**.

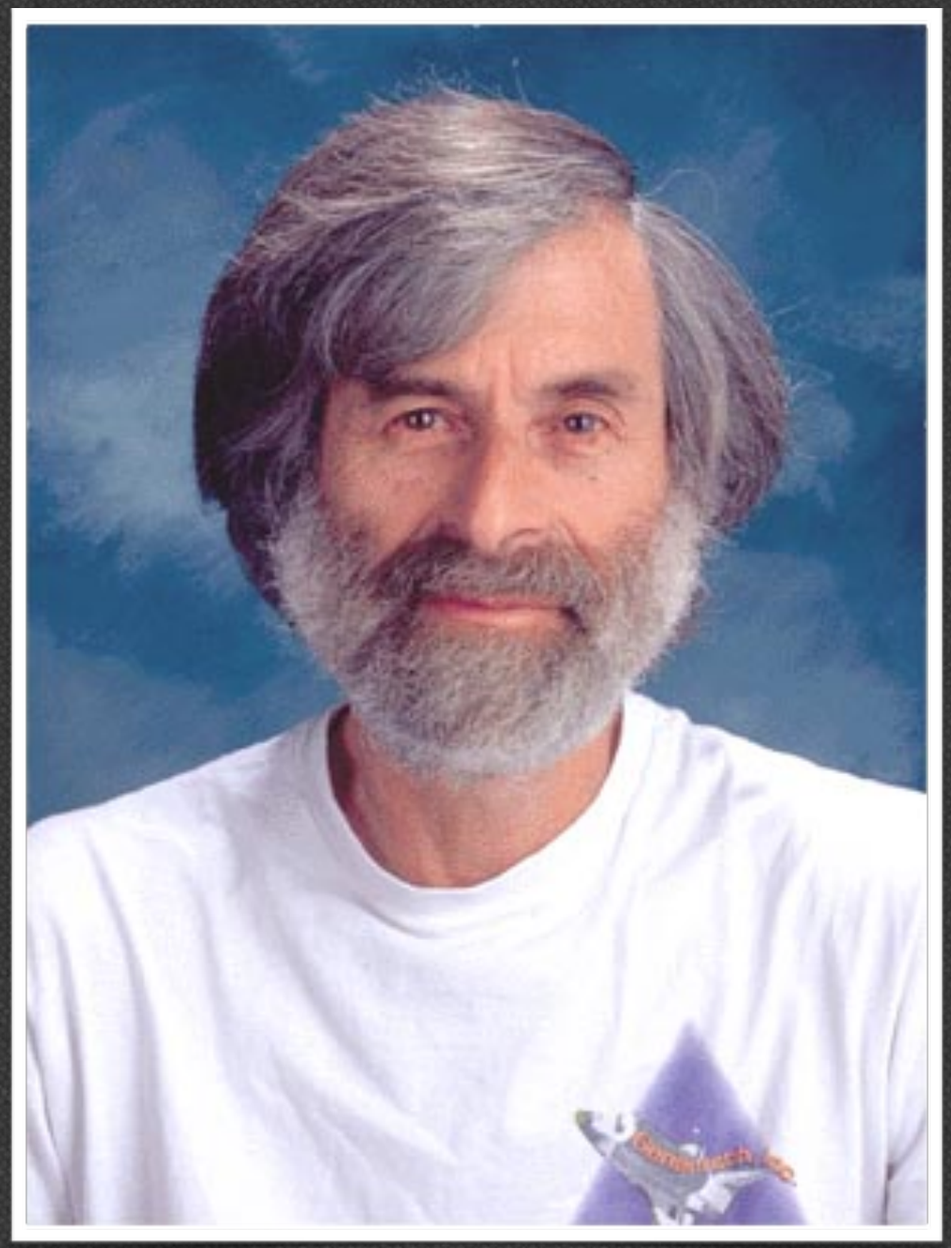
> *"Any sufficiently large system is in a constant state of partial failure."*
>
> *Justin Sheehy, Basho CTO*

Why BASE / EC?

- "Omniscience" is **expensive** and **slow**.
- **Availability** is often correlated to **revenue**.
- Failures happen **all the time**.
- You're probably **doing it already**.

Safety & Liveness

Leslie Lamport
1977



Safety

Safety

- "Bad things don't happen"
- Point-in-time identifiable

Safety

- "Bad things don't happen"
  - Point-in-time identifiable
- mutual exclusion
  - partial correctness
  - first-come, first-serve

Liveness

Liveness

- "Good things eventually happen"
- Always in future

Liveness

- "Good things eventually happen"
- Always in future

- starvation freedom
- termination
- guaranteed service

ACID vs. BASE

- Strong consistency
- Isolation
- Focus on "commit"
- Nested transactions
- Conservative (pessimistic)

- Weak consistency
- Availability first
- Best effort
- Approximate answer
- Aggressive (optimistic)

Fox, Gribble, Chawathe, Brewer, Gauthier - *Cluster-Based Scalable Network Services* (SOSP97)

Eventual consistency is not safe

"...it's easy to satisfy liveness ***without being useful****... If all replicas return the value 42 in response to every request,* ***the system is eventually consistent.****"*

Peter Bailis

http://www.bailis.org/blog/safety-and-liveness-eventual-consistency-is-not-safe/

Liveness of BASE

- Convergence - “eventual delivery“
- Responsiveness - “eventual service“
- Resilience - “eventual recovery“
- Consensus-free - “eventual progress“

Safety of BASE

- Durability - “accepted writes are not lost”
- Integrity - “data is not corrupted”
- Authenticity - “data is not forged”

Real BASE Systems

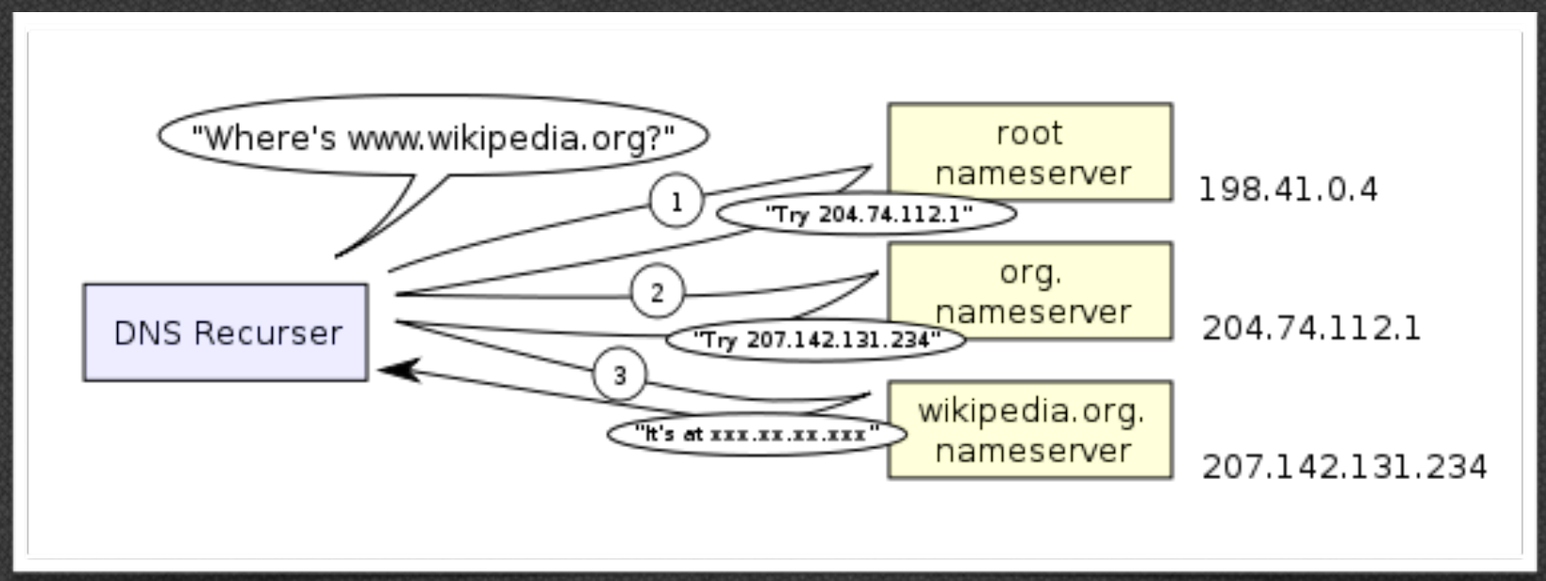
Photo from Wikimedia

Domain Name Service

- Federated, hierarchical database
  - How `qconsf.com` becomes `77.66.16.106`
- Layered system with caching

Domain Name Service

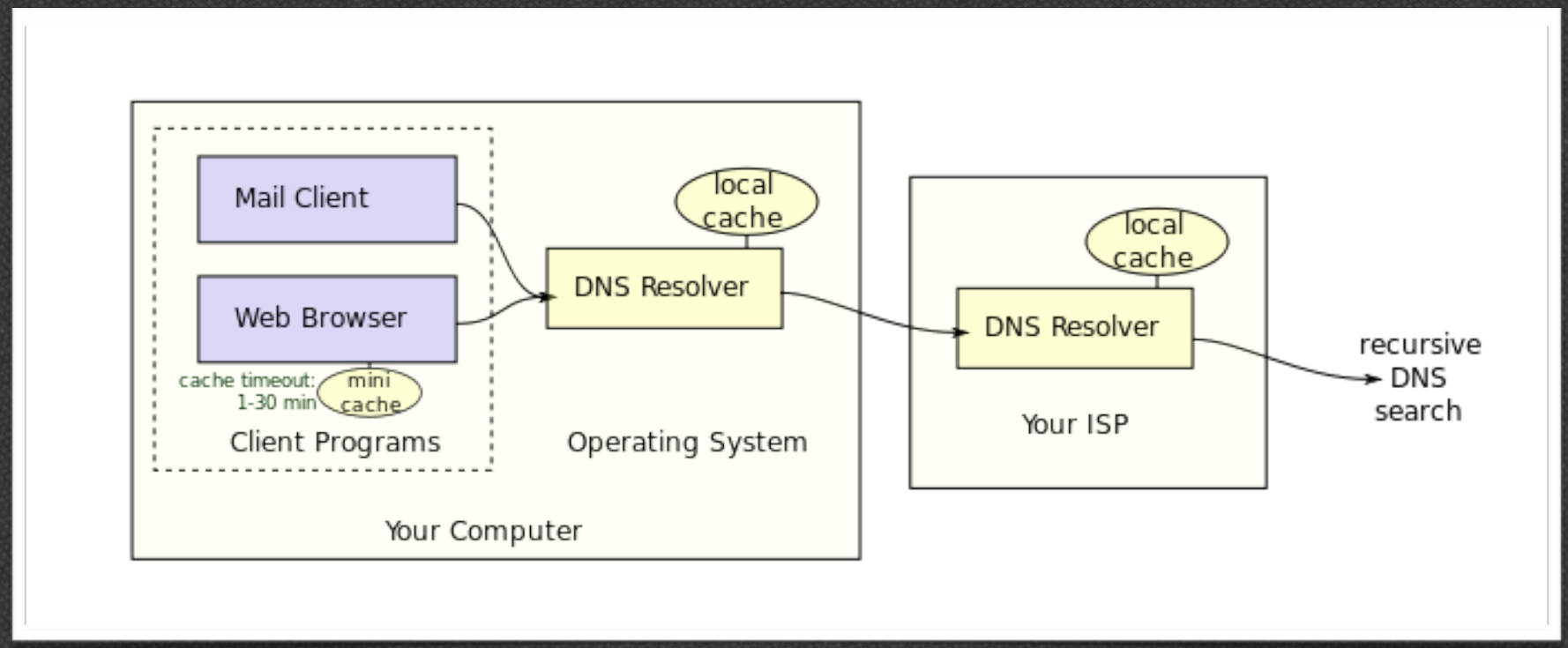
- Federated, hierarchical database
  - How `qconsf.com` becomes `77.66.16.106`
- Layered system with caching



Diagrams from Wikimedia

Domain Name Service

- Federated, hierarchical database
  - How `qconsf.com` becomes `77.66.16.106`
- Layered system with caching



Diagrams from Wikimedia

DNS Liveness

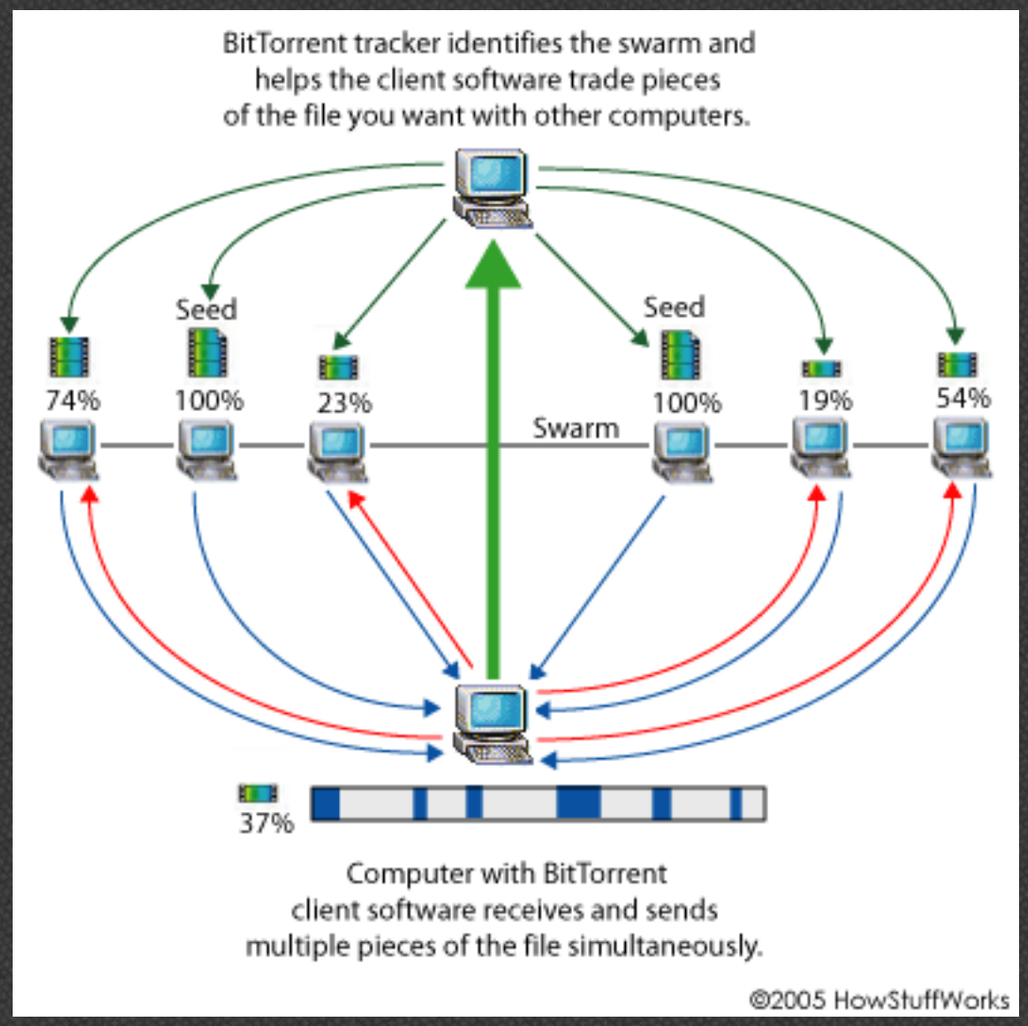
- Convergence - caches eventually expire
- Consensus-free - local authority over subtree updates
- Responsiveness - intermediaries can cache results and reply quicker
- Resilience - authority servers can be replicated/ load-balanced

DNS Safety

- Authenticity - forgery prevented by DNSSEC

BitTorrent

- Peer-to-peer cooperative large-file transfer
- Dynamic membership and block discovery through the “tracker” node
- Epidemic effect



http://computer.howstuffworks.com/bittorrent2.htm

BitTorrent Liveness

- Convergence - all peers that remain connected eventually become seeds
- Resilience - loss of one peer doesn't impede progress
- Responsiveness - closer, faster peers tend to be preferred

BitTorrent Safety

- Integrity - each block is checksummed to prevent corruption

The Web

- Sparsely-connected graph of hypertext documents identified by URIs
- Rich caching semantics: expiration, validation, control
- Fluid evolution through uniform interface
- Layered system (federated)

Web: Liveness

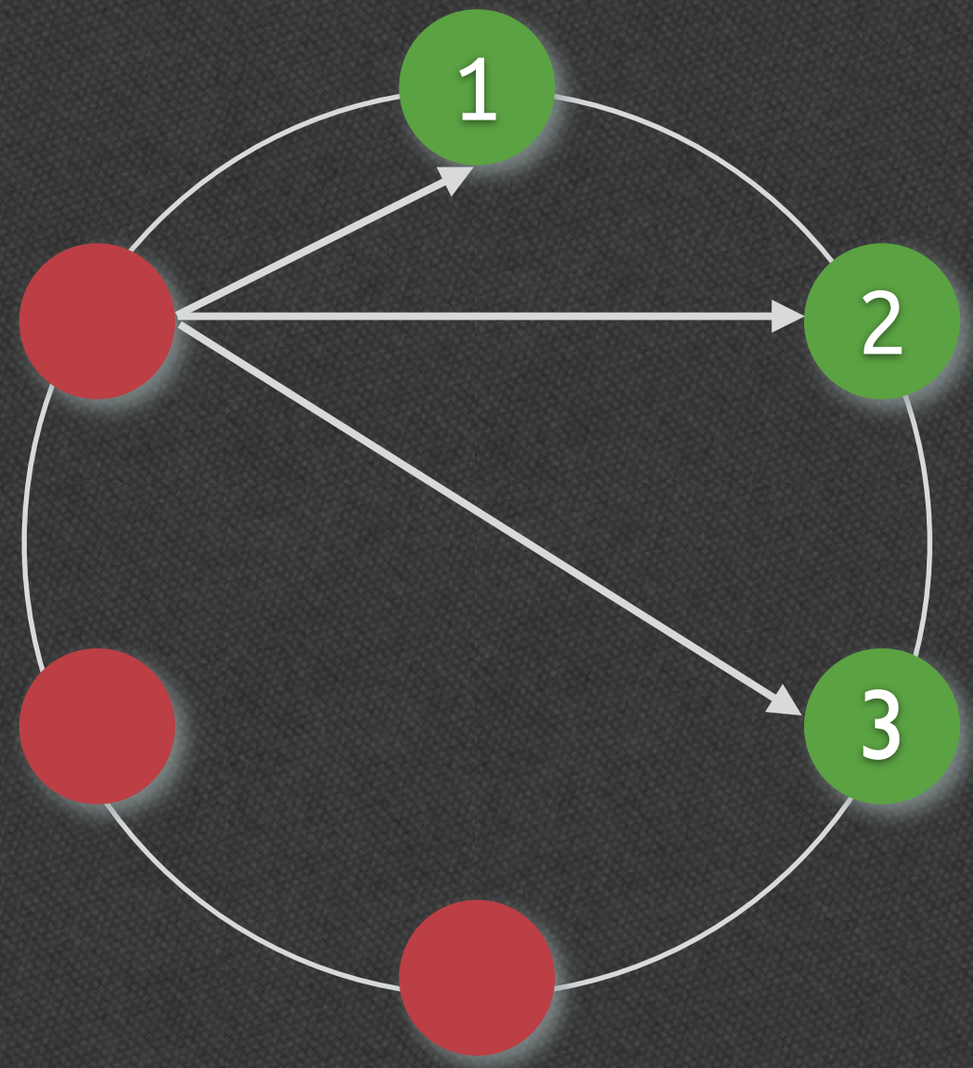
- Consensus-free - local documents can be changed, moved, removed without coordination
- Convergence - caching semantics prevent unbounded staleness, redirection
- Responsiveness - many parties can proxy, cache
- Resilience - failure of one server doesn't stop the system

Web: Safety

- Privacy & Authenticity - HTTPS/SSL/TLS
- Integrity - POST responses don’t pollute caches

Dynamo

- Key-value store: distributed, replicated, partitioned
- Client requests can go to any node
- Low-latency at high percentiles
- Many clones: Riak, Cassandra, Voldemort



Dynamo: Liveness

- Convergence - read-repair, hash-tree exchanges, vector-clocks
- Resilience - hinted-handoff, sloppy quorums
- Responsiveness - replication
- Consensus-free - loose coordination, concurrent updates

Dynamo: Safety

- Authenticity - won't serve data you didn't store
- Durability - confirmed writes are not lost

ACID vs. BASE

Photo by Associated Press

SPECTRUM

Photo by Associated Press

Embrace Eventual Consistency