

CAP Twelve Years Later: How the “Rules” Have Changed

Eric Brewer

Computer, February 2012, IEEE Computer Society

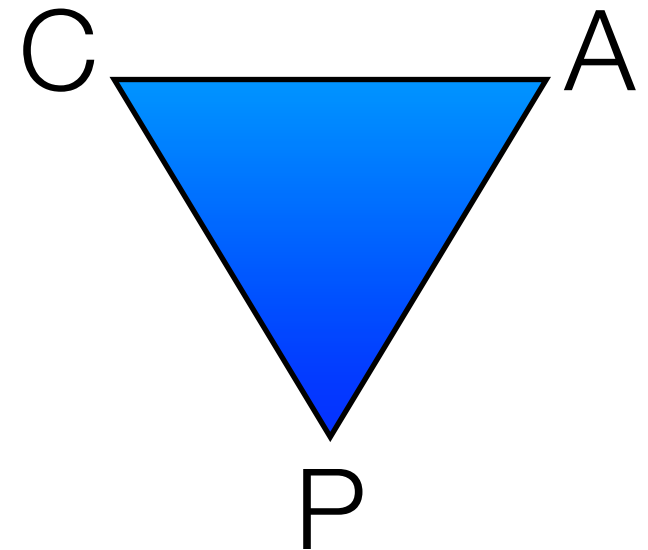
Presentation by Kevin Urban

On the Publication

- February 2012 – hot and new!
- By Eric Brewer (Mr. CAP) himself
- Reflects on the past 12 years and points in new directions

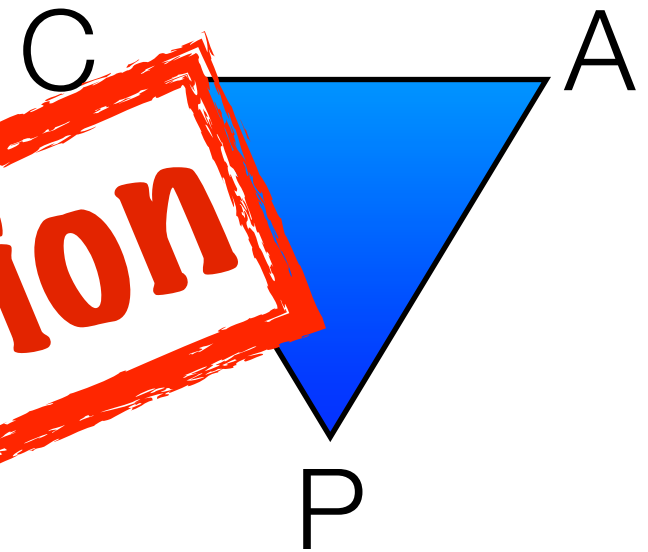
Original CAP “Rules”

- Original formulation: “2 of 3” CAP properties
- Initial perception:
 - **P**artition tolerance is needed
 - Trade off between **C**onsistency (ACID Databases) and **A**vailability (NoSQL)



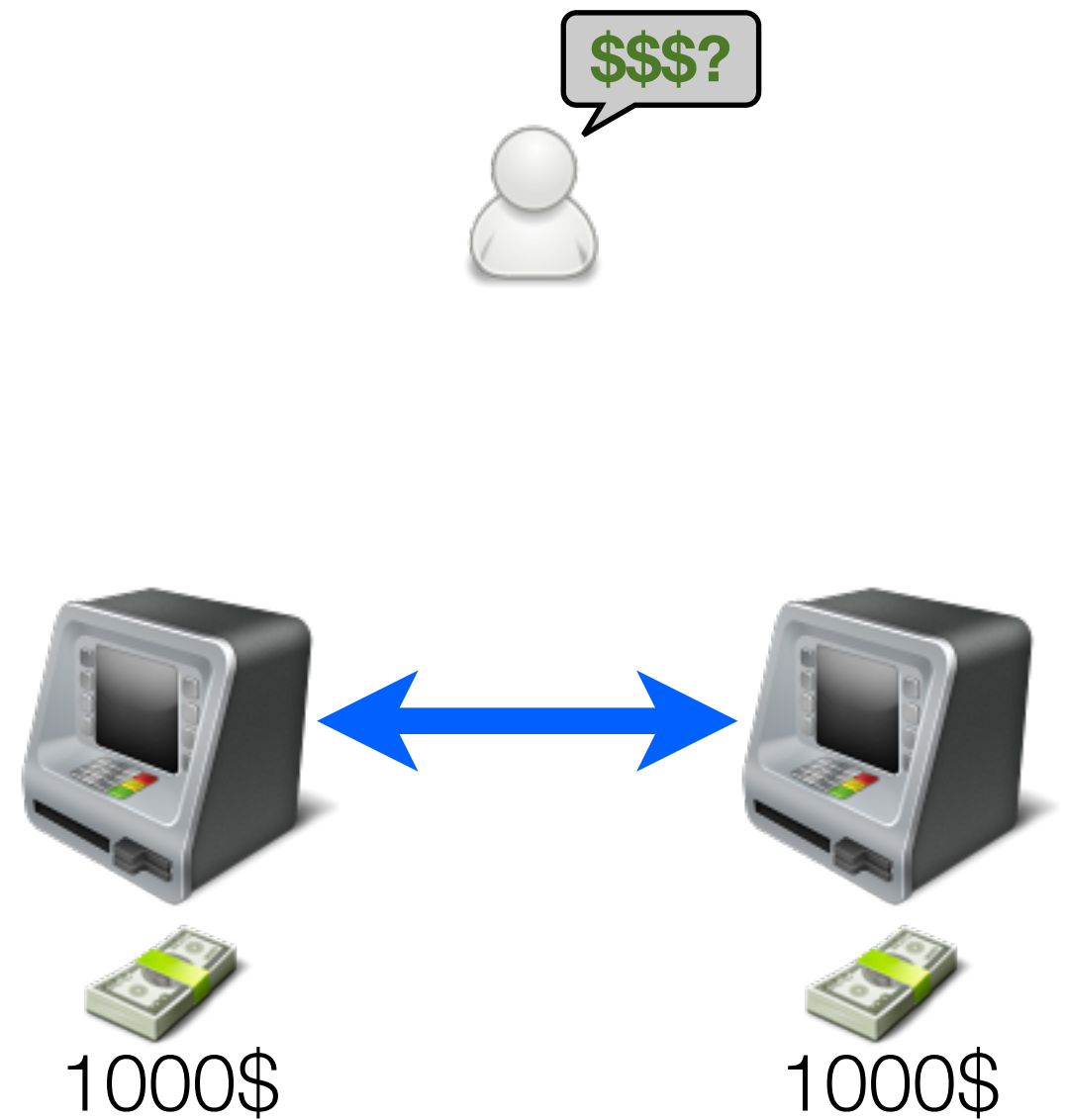
Original CAP “Rules”

- Original formulation: “2 of 3” CAP properties
- Initial perception:
 - **P**artition tolerance is needed
 - Trade off between **C**onsistency (ACID Databases) and **A**vailability (NoSQL)



over simplification

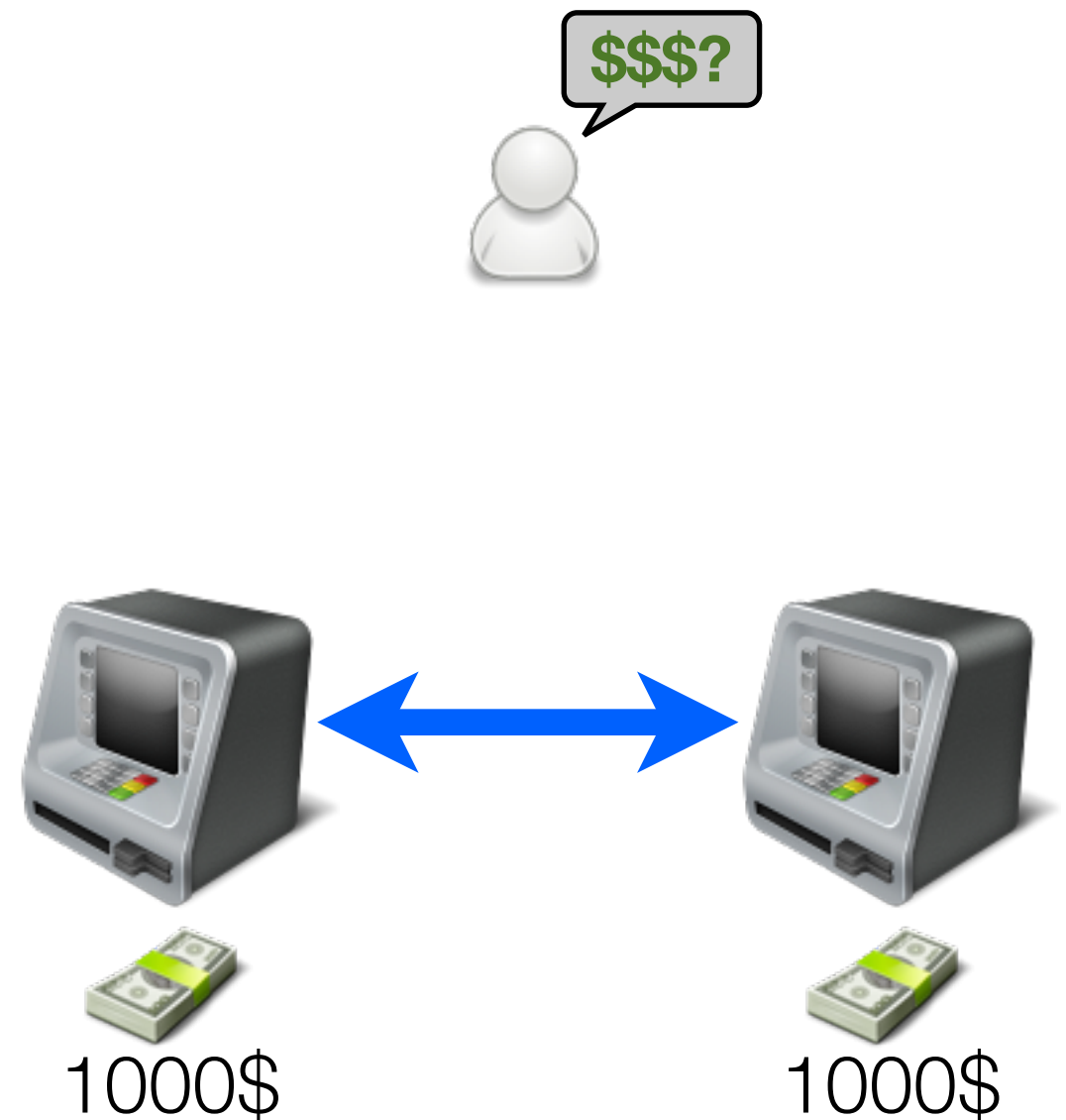
Partition Decision – Example



Partition Decision – Example

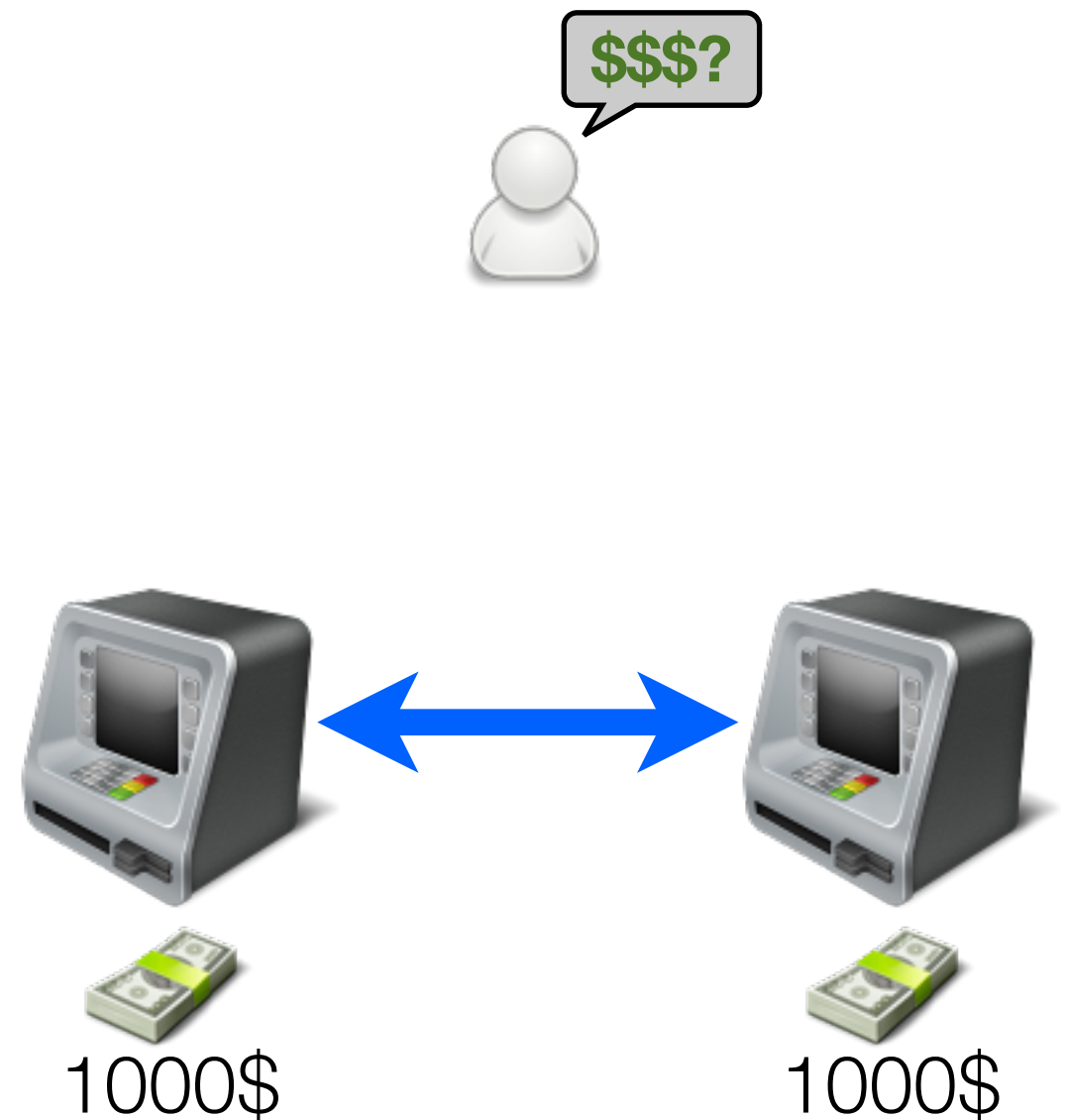
- Partition Decision: Favor...

- **A:** Availability (NoSQL)
- **C:** Consistency (e.g. 2PC)



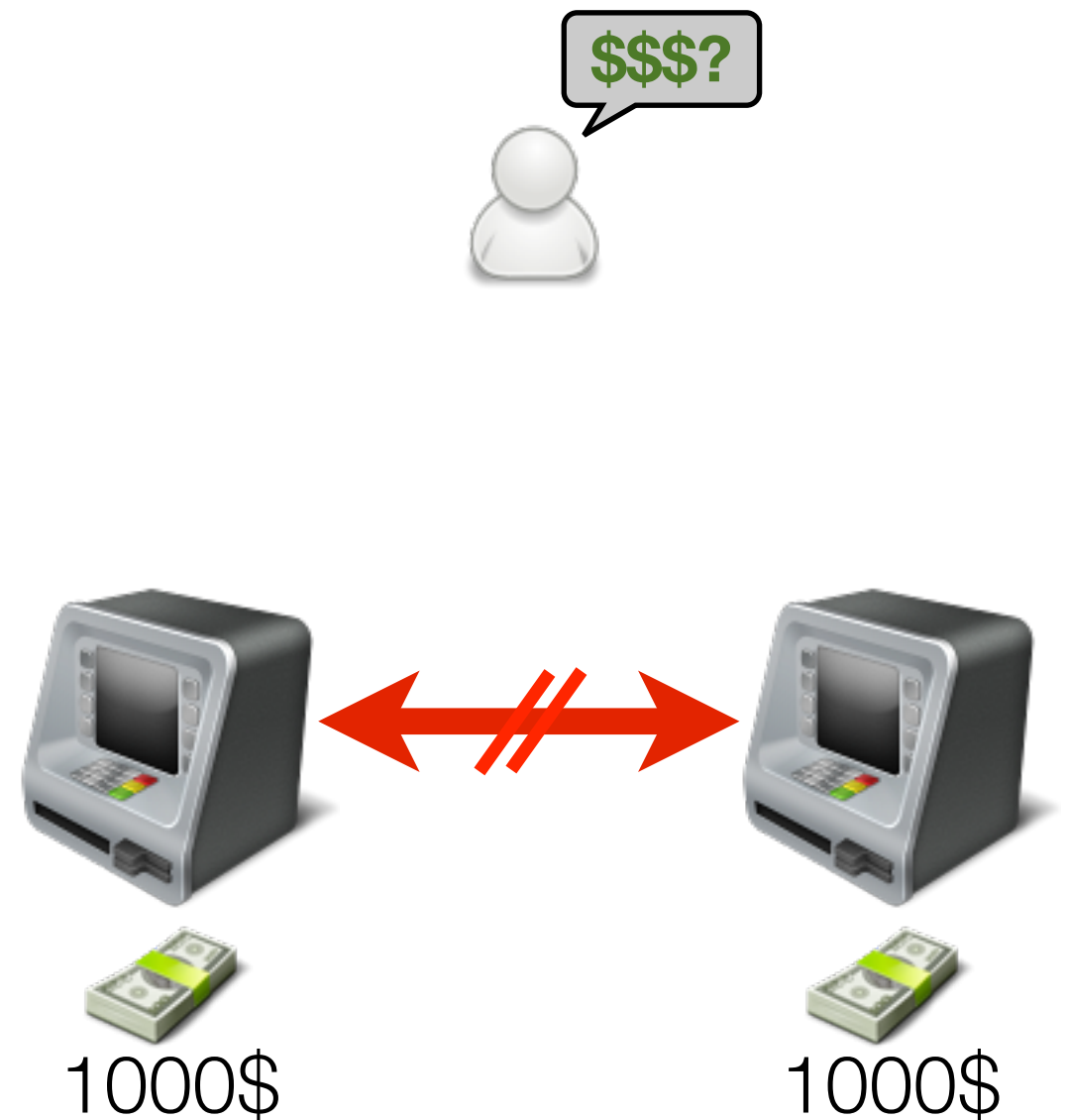
Partition Decision – Example

- Partition Decision: Favor...
 - **A:** Availability (NoSQL)
 - **C:** Consistency (e.g. 2PC)
- Can be postponed



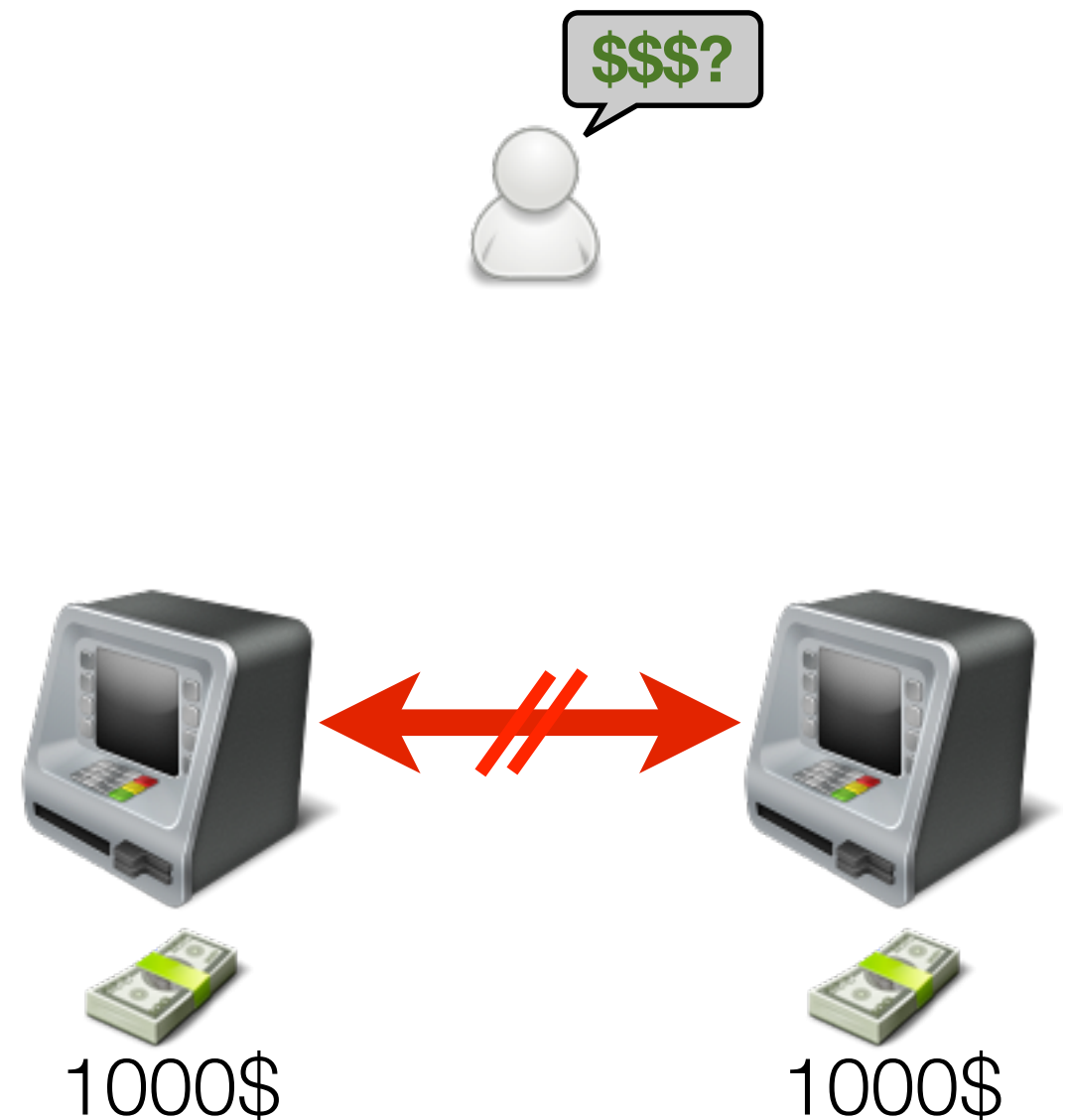
Partition Decision – Example

- Partition Decision: Favor...
 - **A:** Availability (NoSQL)
 - **C:** Consistency (e.g. 2PC)
- Can be postponed



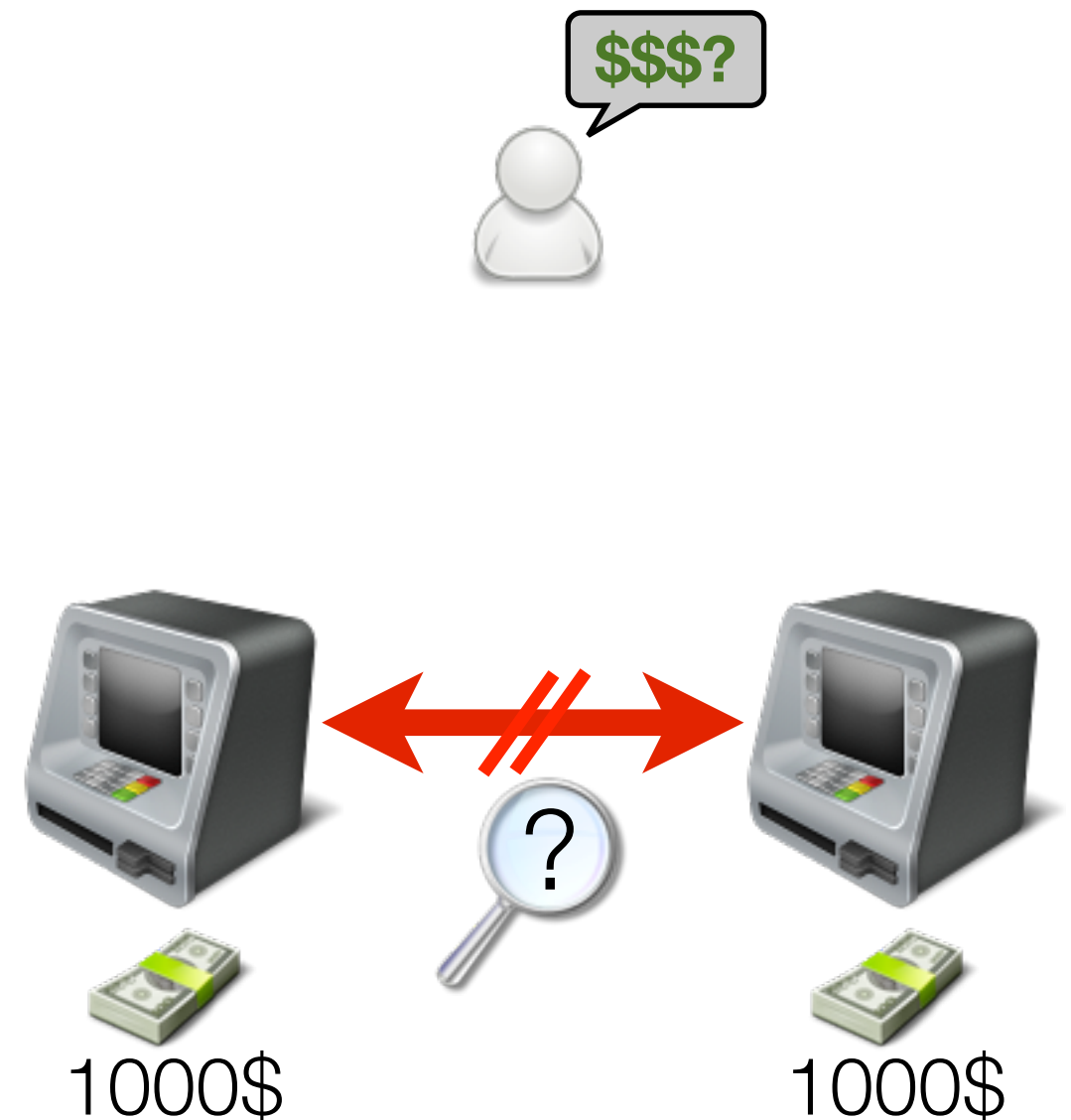
Partition Decision – Example

- Partition Decision: Favor...
 - **A:** Availability (NoSQL)
 - **C:** Consistency (e.g. 2PC)
- Can be postponed
- No trade off needed while connected!



Partition Decision – Example

- Partition Decision: Favor...
 - **A:** Availability (NoSQL)
 - **C:** Consistency (e.g. 2PC)
- Can be postponed
- No trade off needed while connected!
- Partition Detection

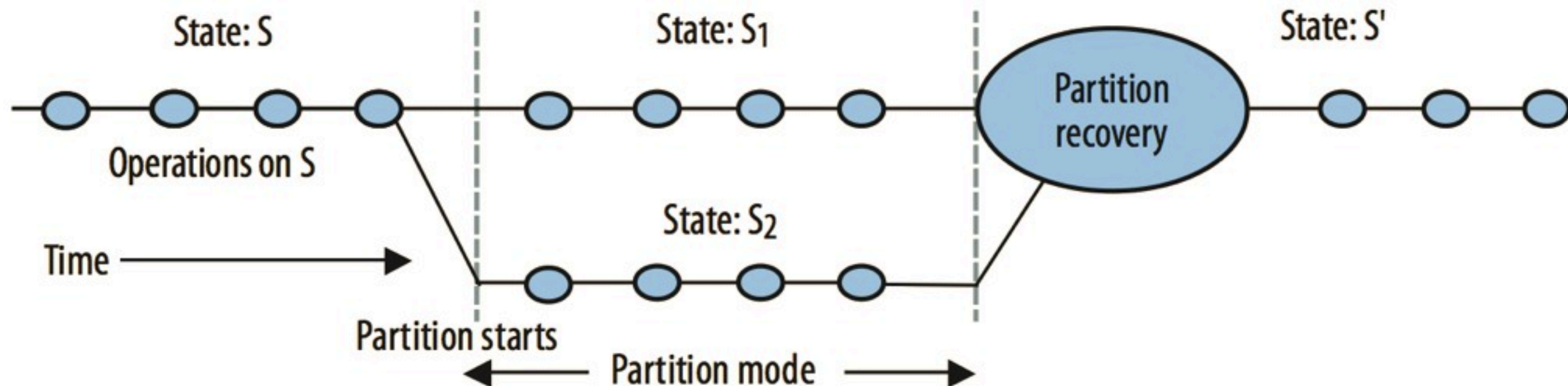


CAP is not binary

- Consistency vs. Availability
 - Facebook / HTML5 Local store
 - There are nuances in partition, too!
 - Disagreements about whether there is a partition
 - Delay / Partition discussion
- ➡ Explore freedom in this space!

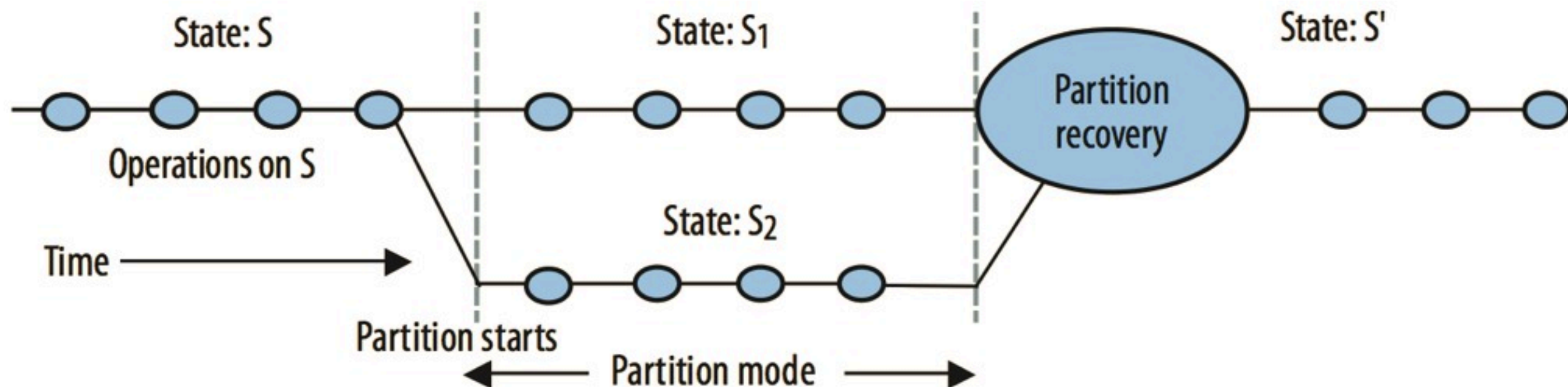
Partition Managing

- Detect start of a partition
- Enter an explicit partition mode that may limit some operations
- Initiate partition recovery when communication is restored

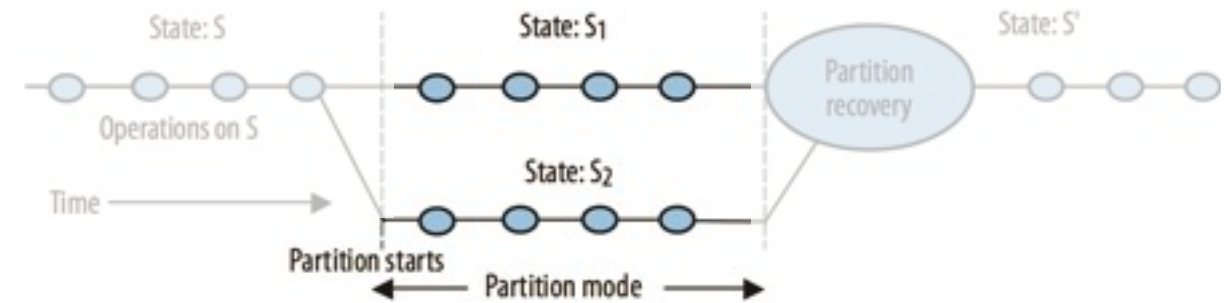


Partition Managing (cont.)

- Two options in partition mode:
 - Record extra information (→ more availability)
 - Limit operations (→ more consistency)
- CAP Decisions at fine granularity!

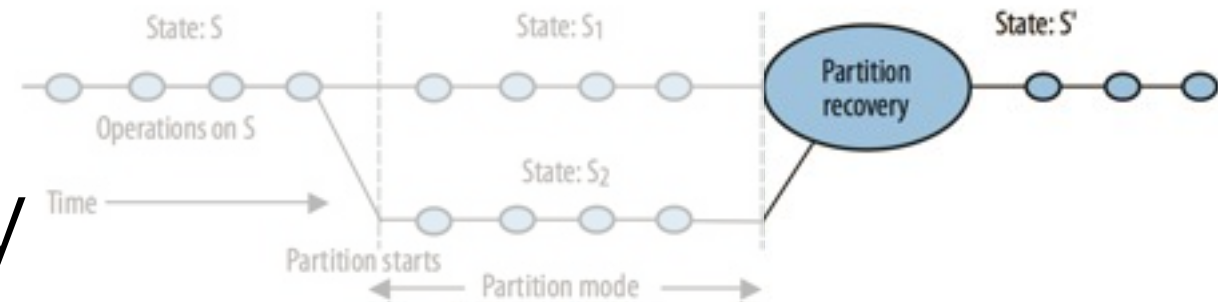


Partition Mode: Invariants



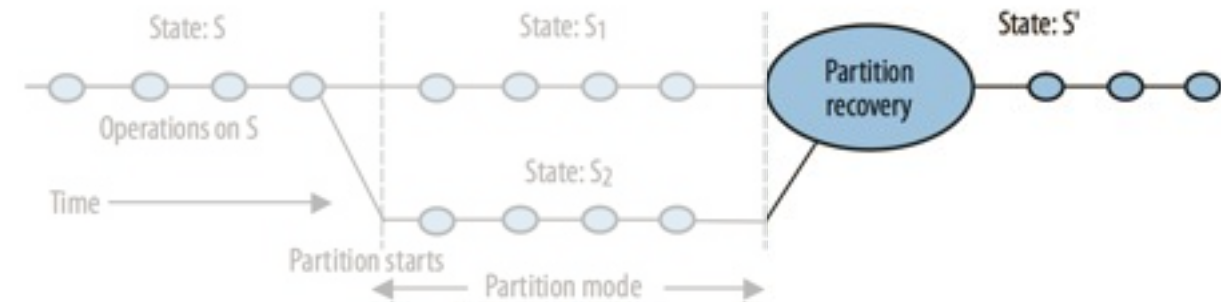
- Invariants: “Constraints” in a system
 - Example: keys in a table are unique
- System designer has to look at each invariant and action and...
 - prohibit an action (maintain consistency);
 - delay an action (forfeit availability, but hide it); or
 - modify it (→ more towards availability, risk consistency)

Recovery – Re-establishing consistency



- State must become consistent
 - Use log information
 - Automatic merging (think SVN / git / Mercurial)
 - Commutative operations (Google Docs, Bounded addition)
 - Commutative Replicated Data Types (CRDT)

Recovery – Compensating Errors



- History required (Version Vectors, limited Operations)
- Externalized operations need to be known
- Human Escalation
 - Overbooking an airplane (partitioned system)
 - Partition Recovery: boarding the airplane
- Restore consistency again

Conclusion

- Original formulation misleading
- Much development has been done since CAPs inception
- Reduces different approaches to a common formulation
- Pay attention to optimization details
- The best services depend on such details

“*System Designers should not blindly sacrifice consistency or availability [...].*

They can optimize both properties through careful management of invariants during partitions.”

Thank you

Questions?