CENTRAL UNIVERSITY Lecture

CAP, PACELC and Consistency Models

A story about real trade-offs in distributed systems

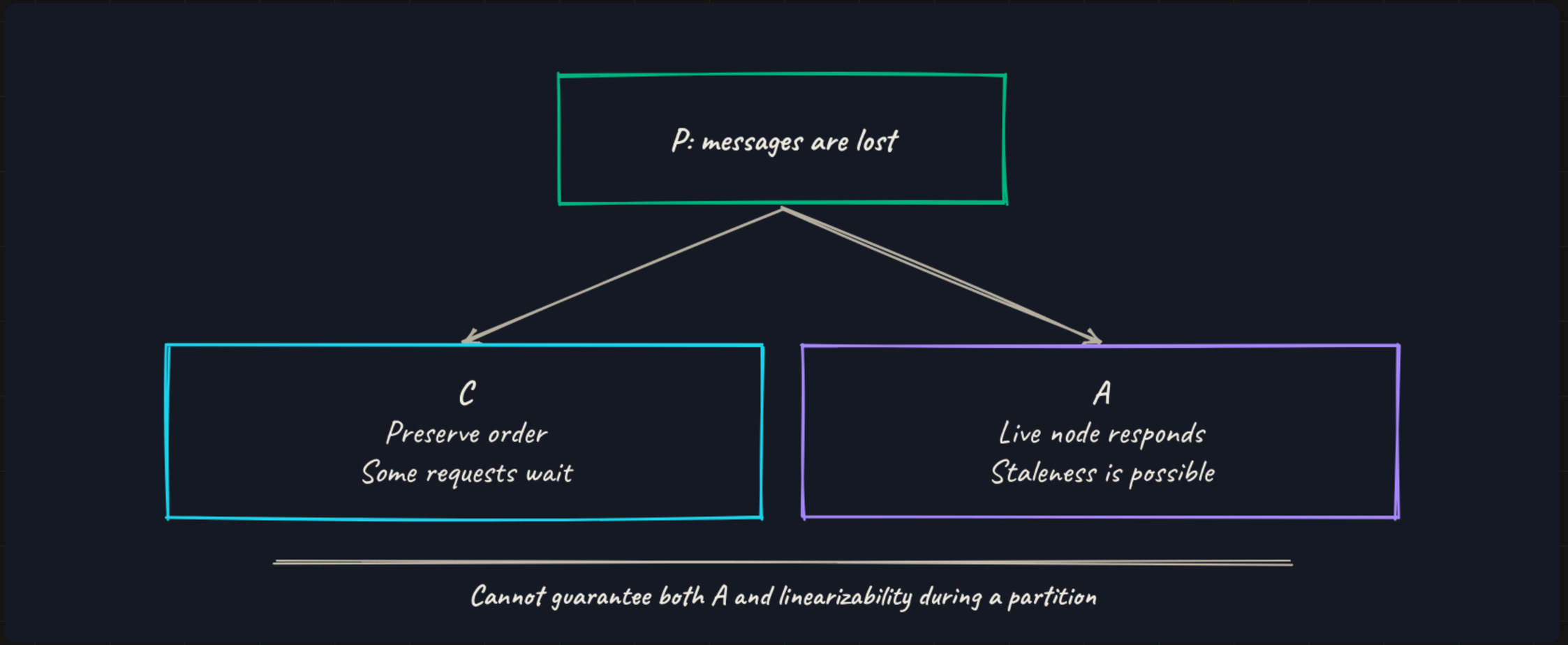
Lecture Map

Plan Navigation

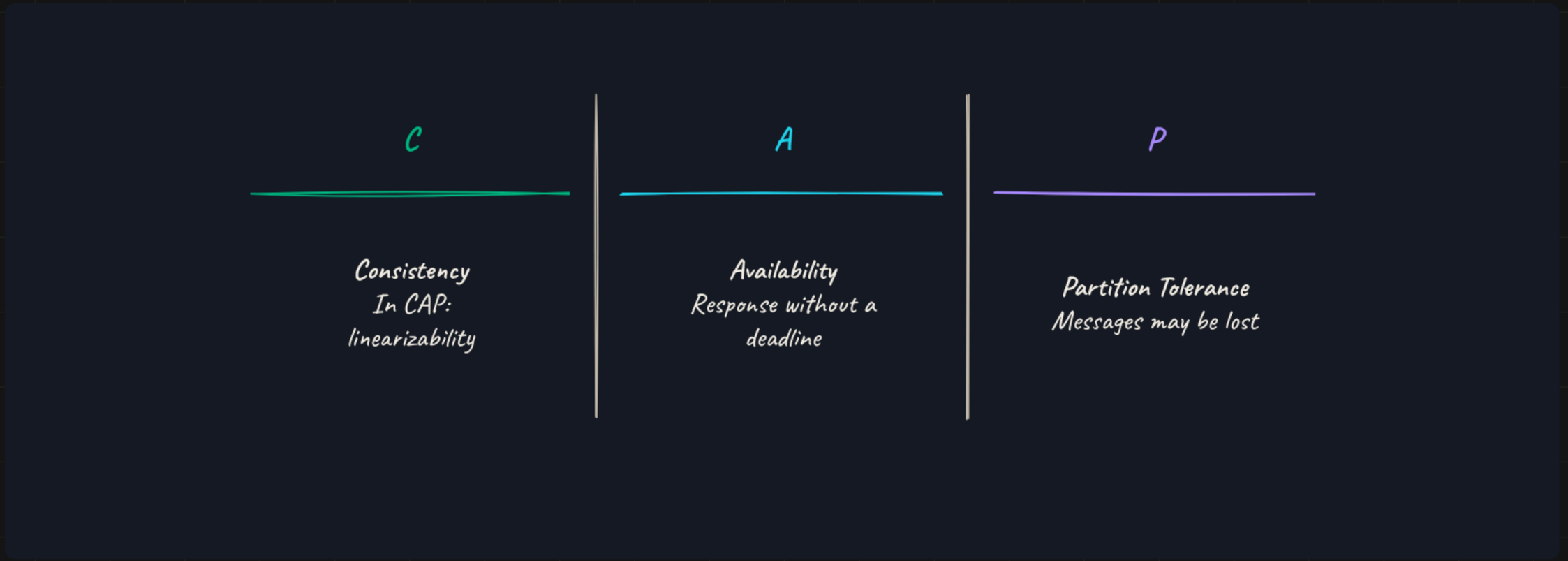


CAP Theorem in One Phrase

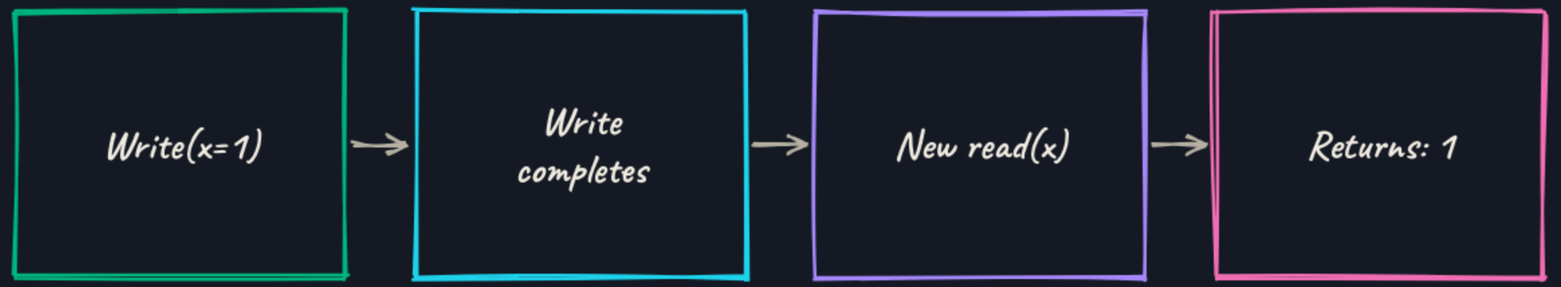
CAP Chapter Statement



C, A, and P: What's Inside



Consistency in CAP = Linearizability



The global order respects the real-time order of operations

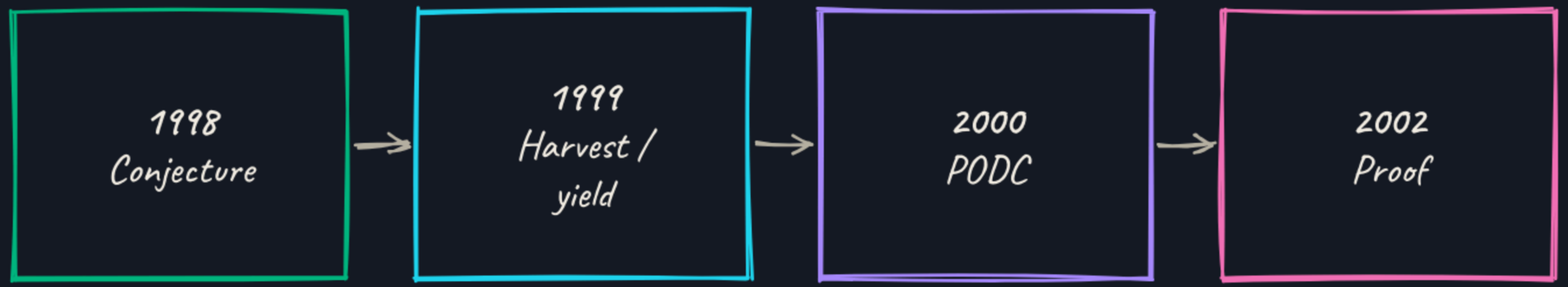
Availability in the Theorem



No fixed deadline; waiting forever violates A

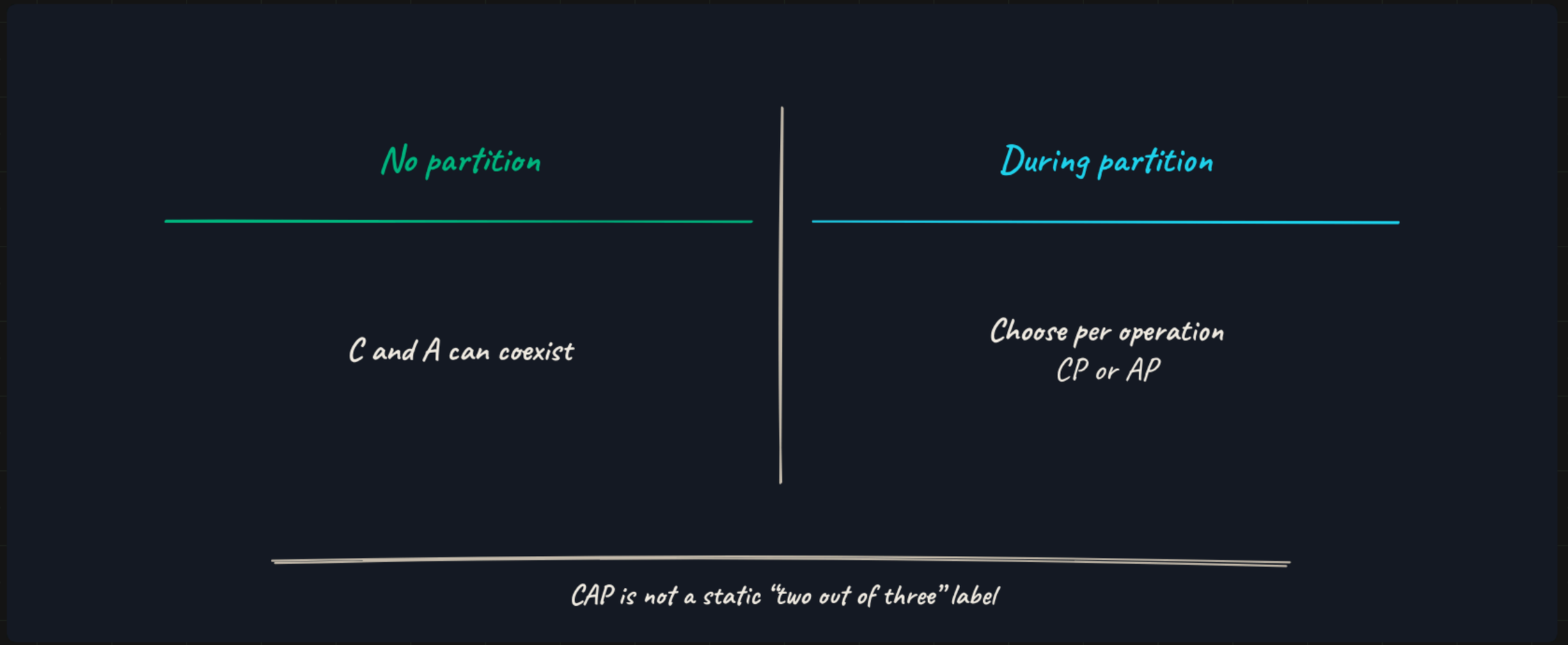
How CAP Emerged

CAP Chapter History



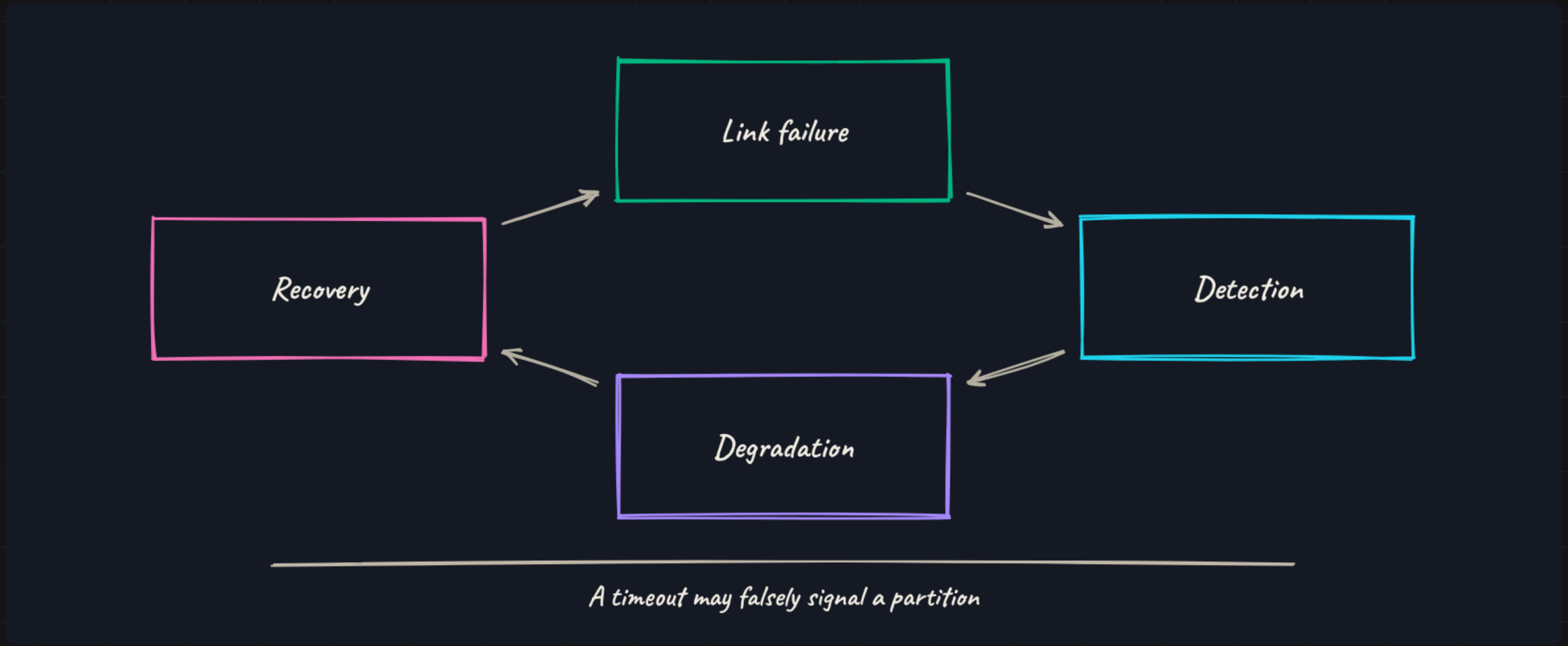
The ’Pick 2 of 3’ Myth

CAP Chapter Myths



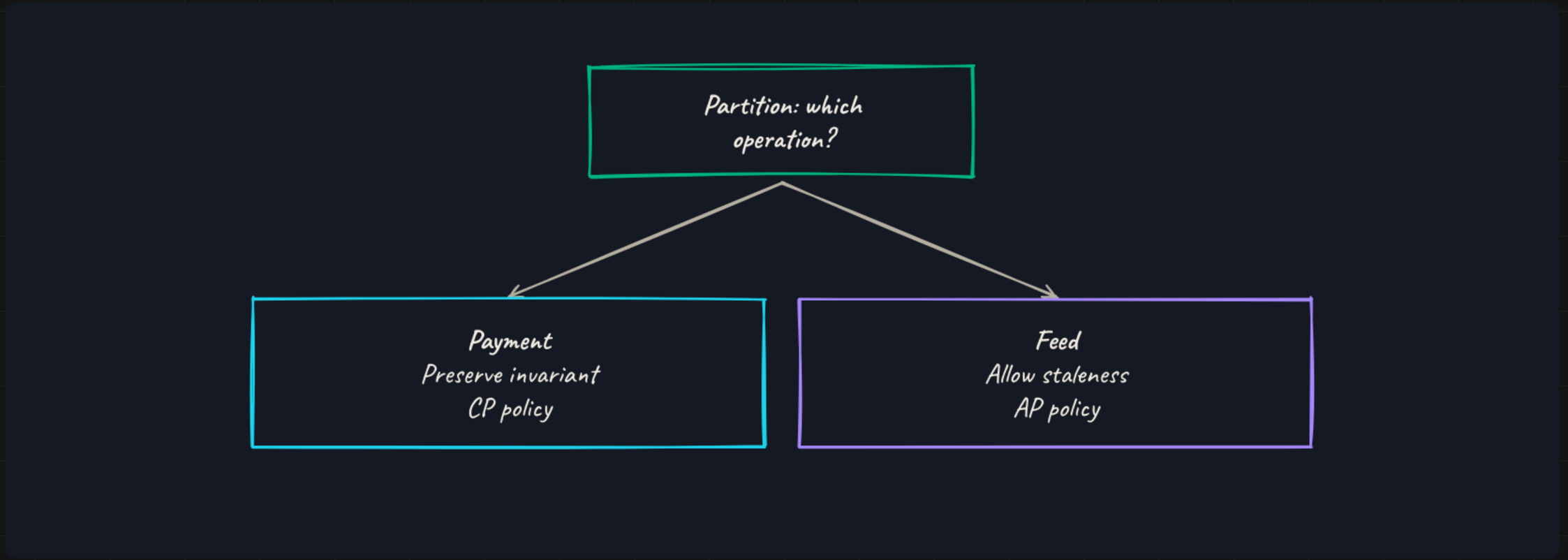
Partitions: Rare, Not Exotic

CAP Chapter Myth #1



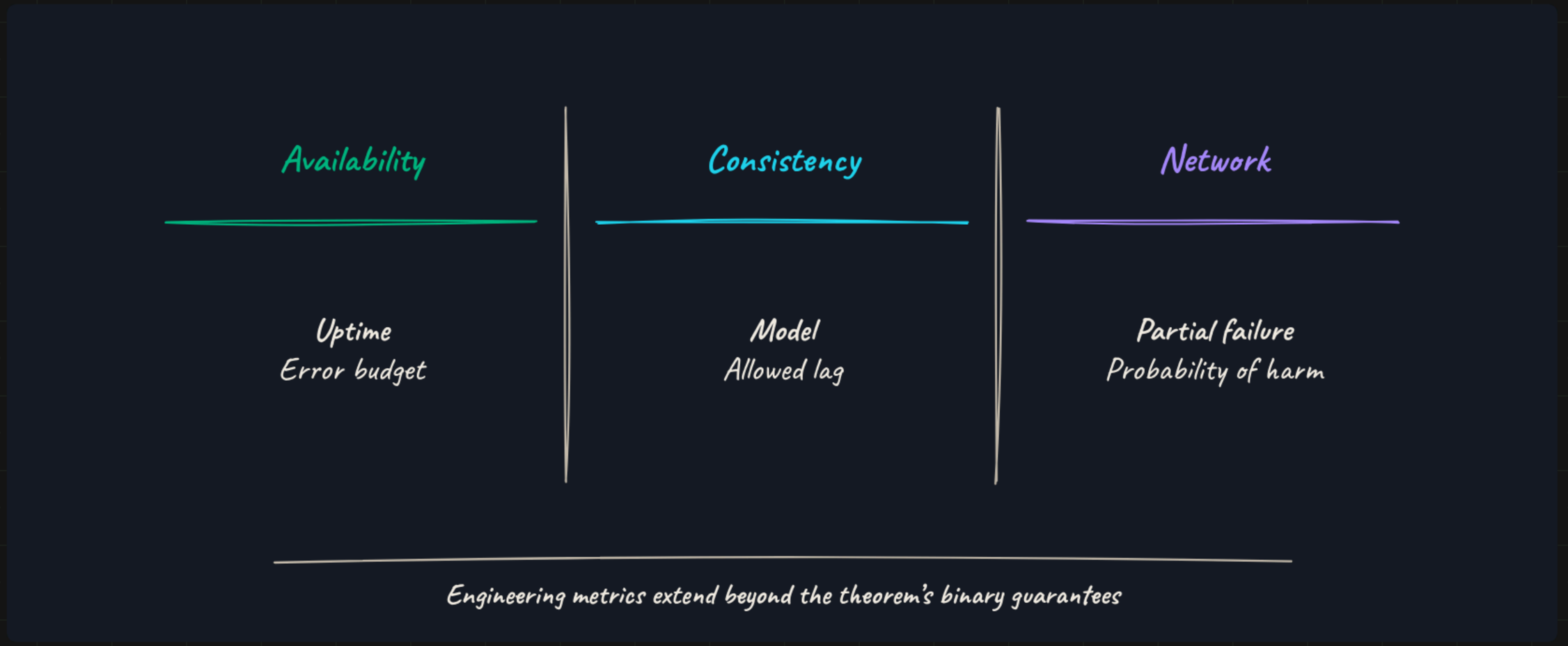
Decisions Are Made Per Operation

CAP Chapter Myth #2

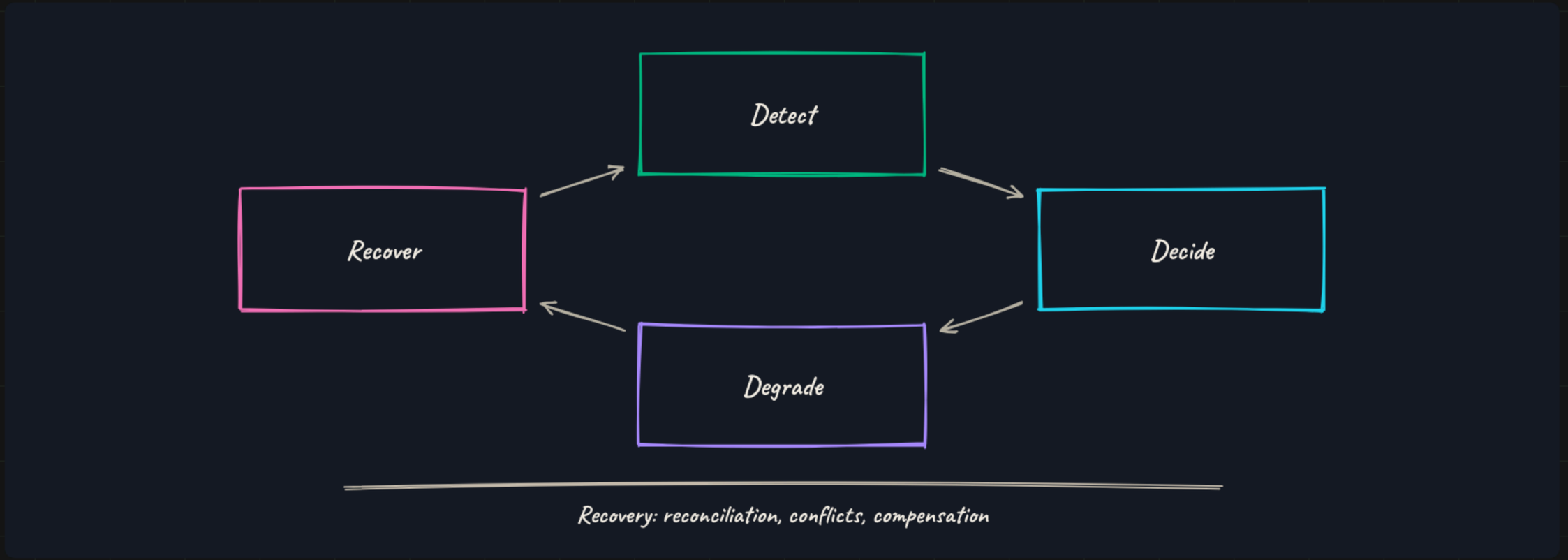


C/A/P Are Spectrums

CAP Chapter Myth #3

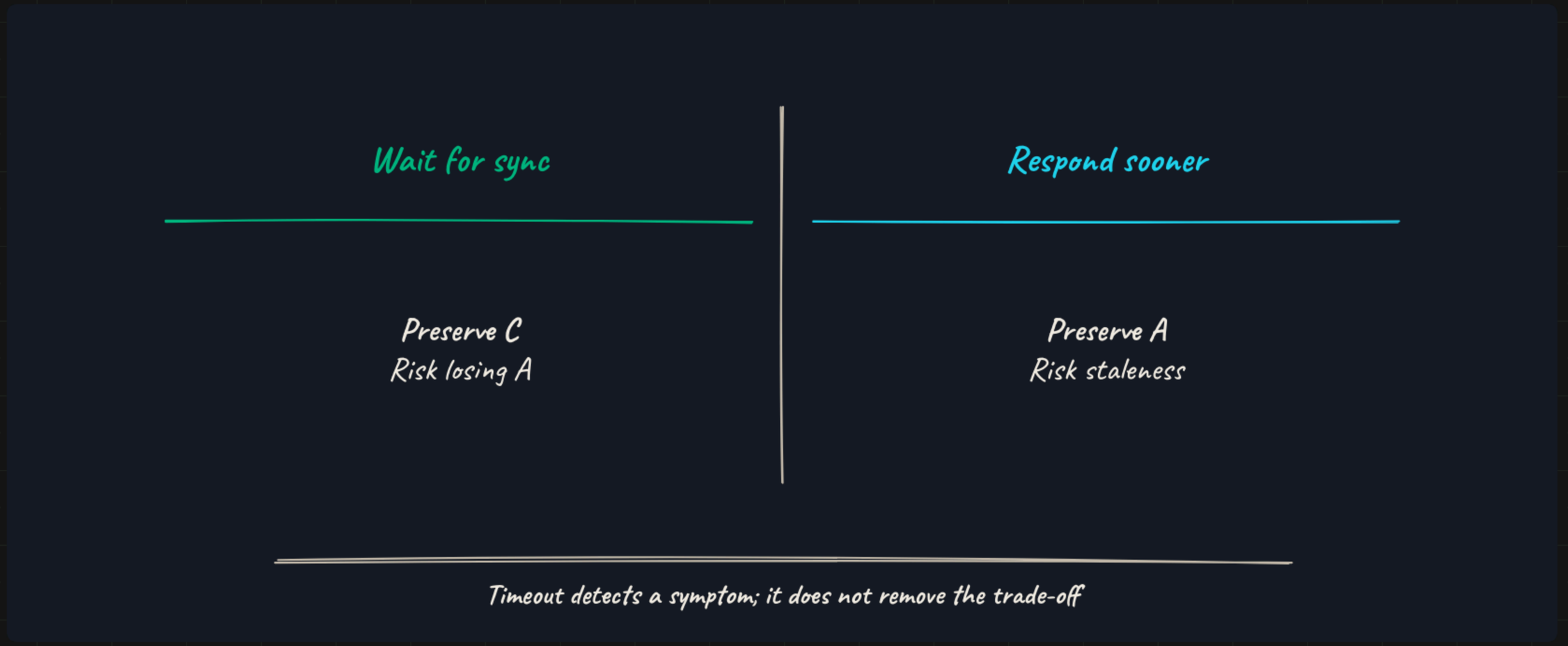


Partition Mode as State Machine

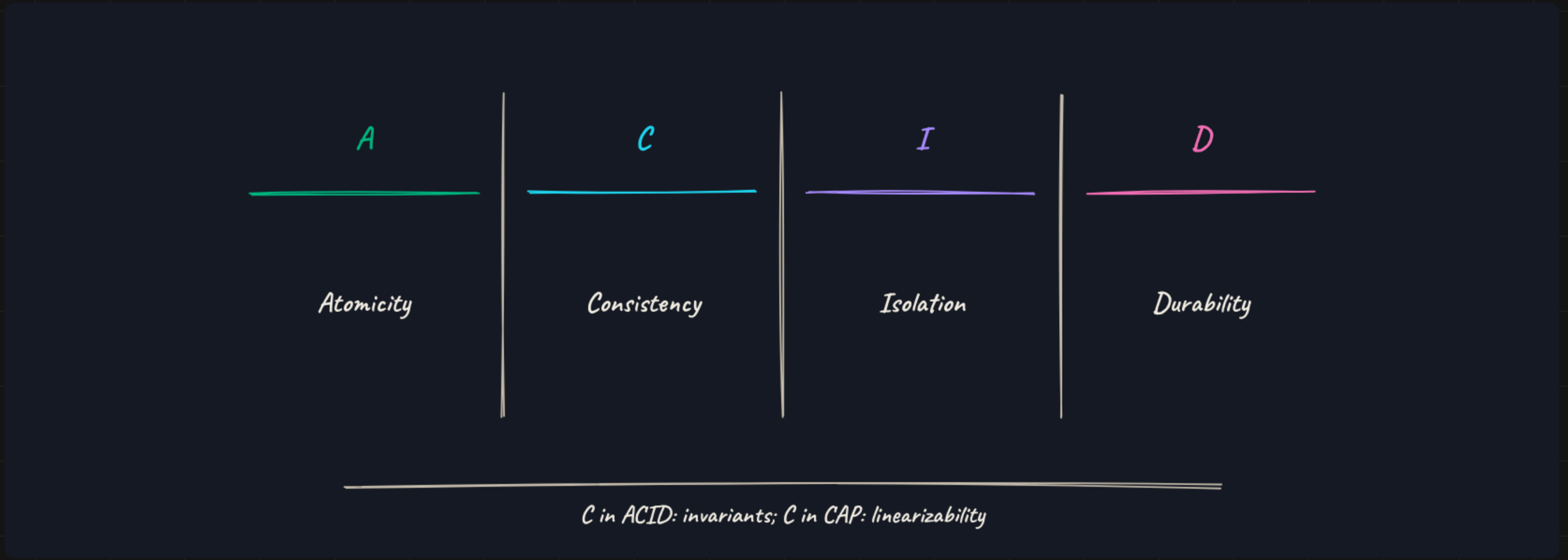


Latency Governs CAP

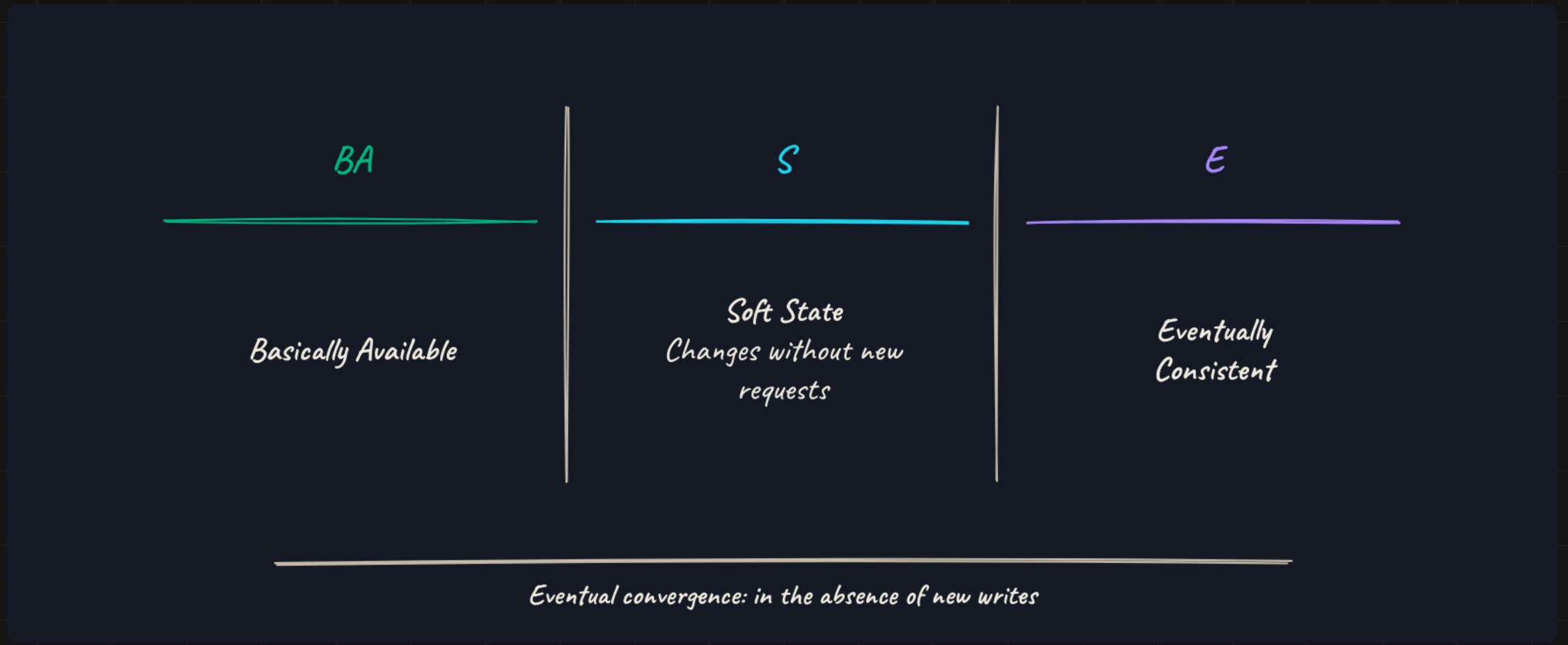
CAP Chapter Latency



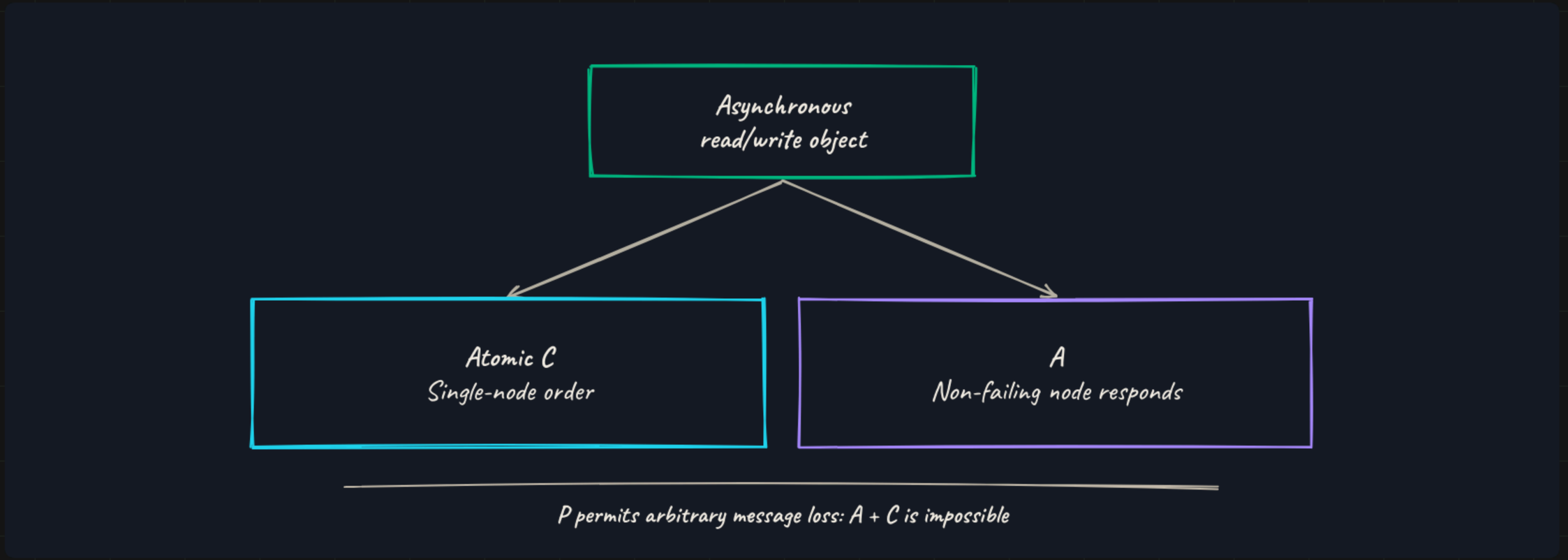
ACID vs CAP: Different C



BASE Answers Availability

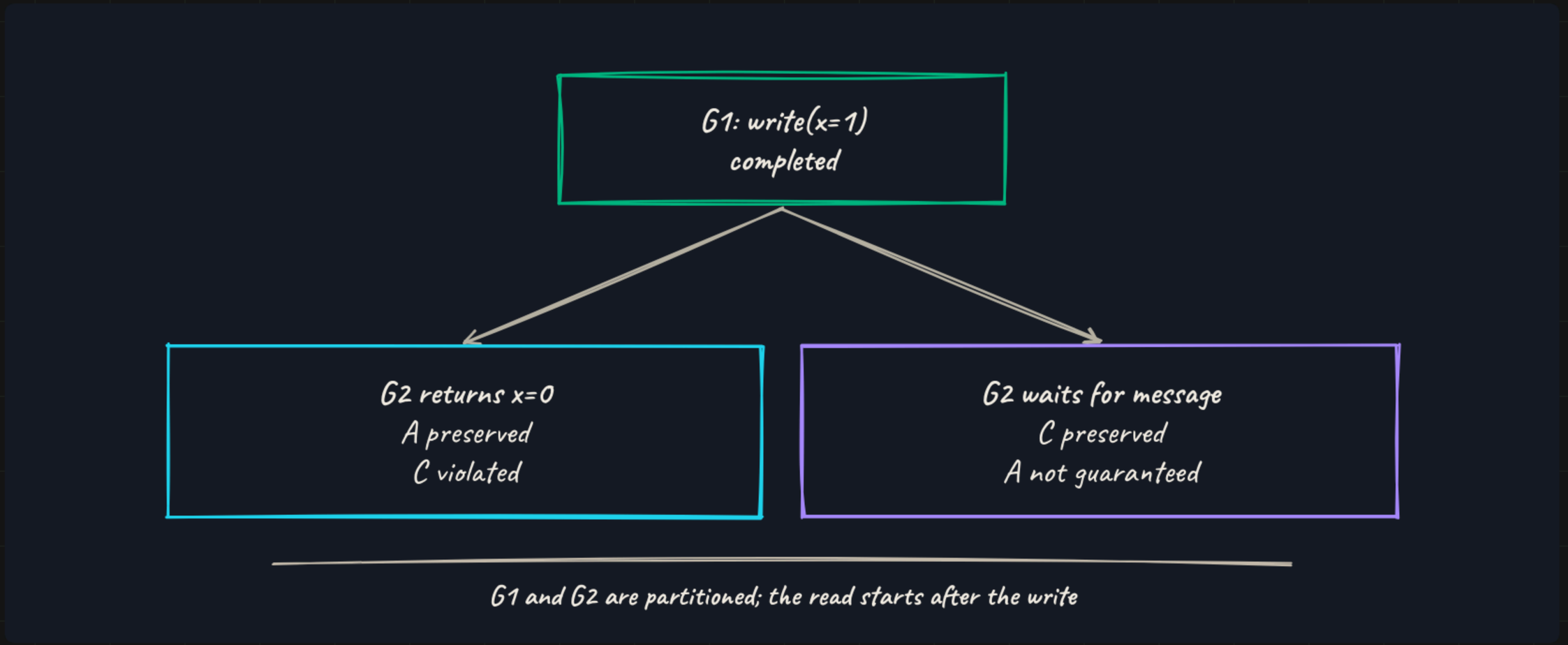


Gilbert–Lynch Formal Definitions



Asynchronous Impossibility

CAP Chapter | Theorem 1



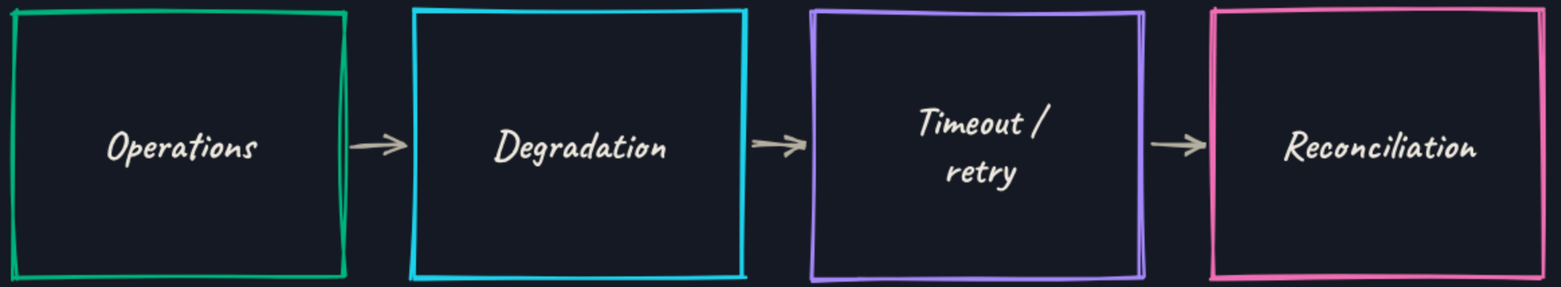
Partial Synchrony Doesn’t Help



Message loss preserves the conflict even with partial synchrony

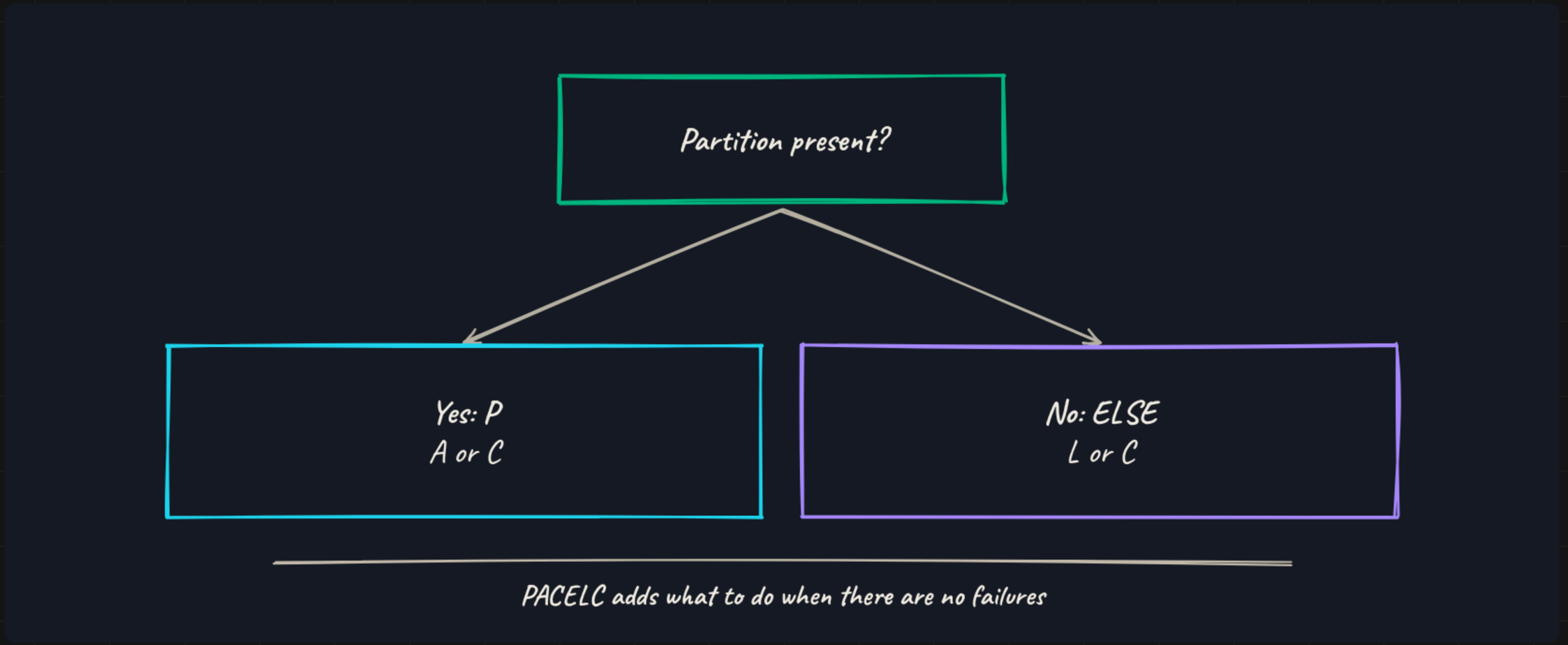
CAP Takeaways for Architecture

CAP Chapter Summary



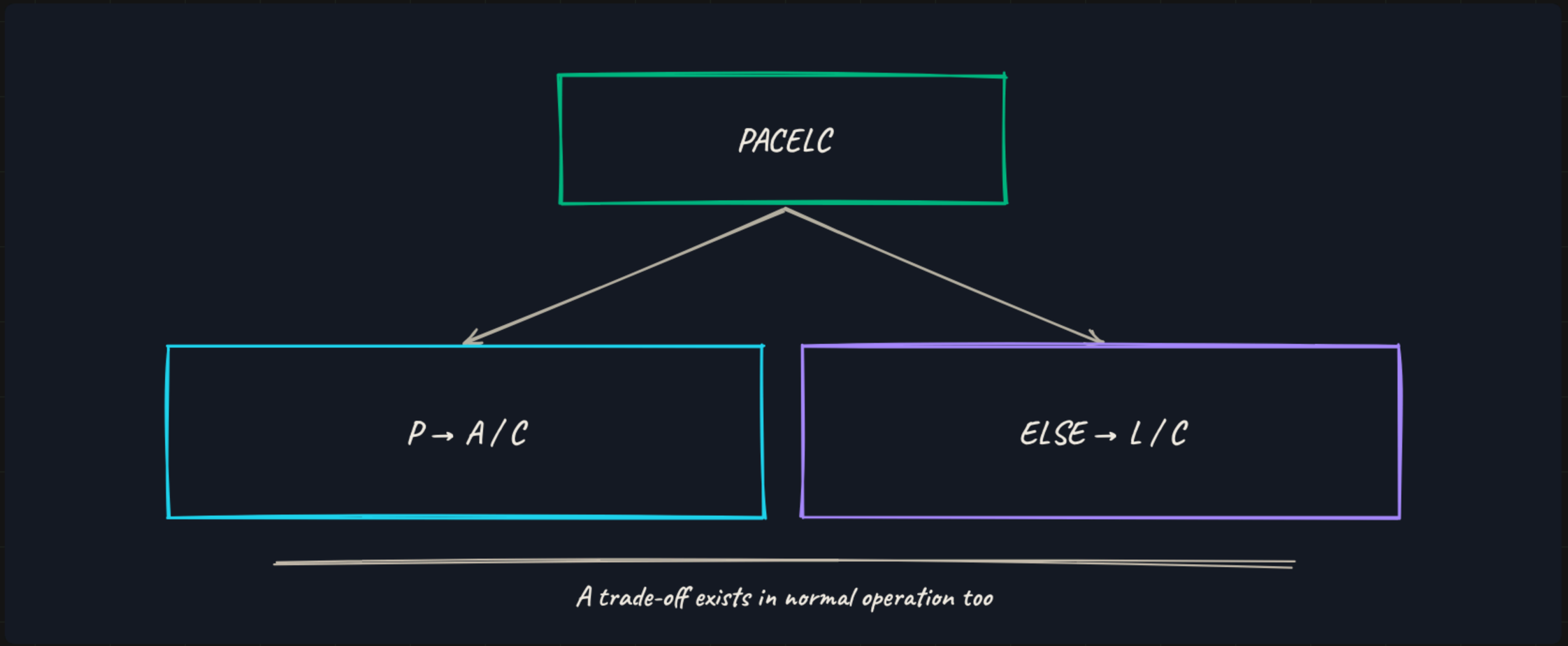
CAP: What to Do in Failure

Transition PACELC



PACELC Formula

PACELC Chapter Formula



Why PACELC Was Needed

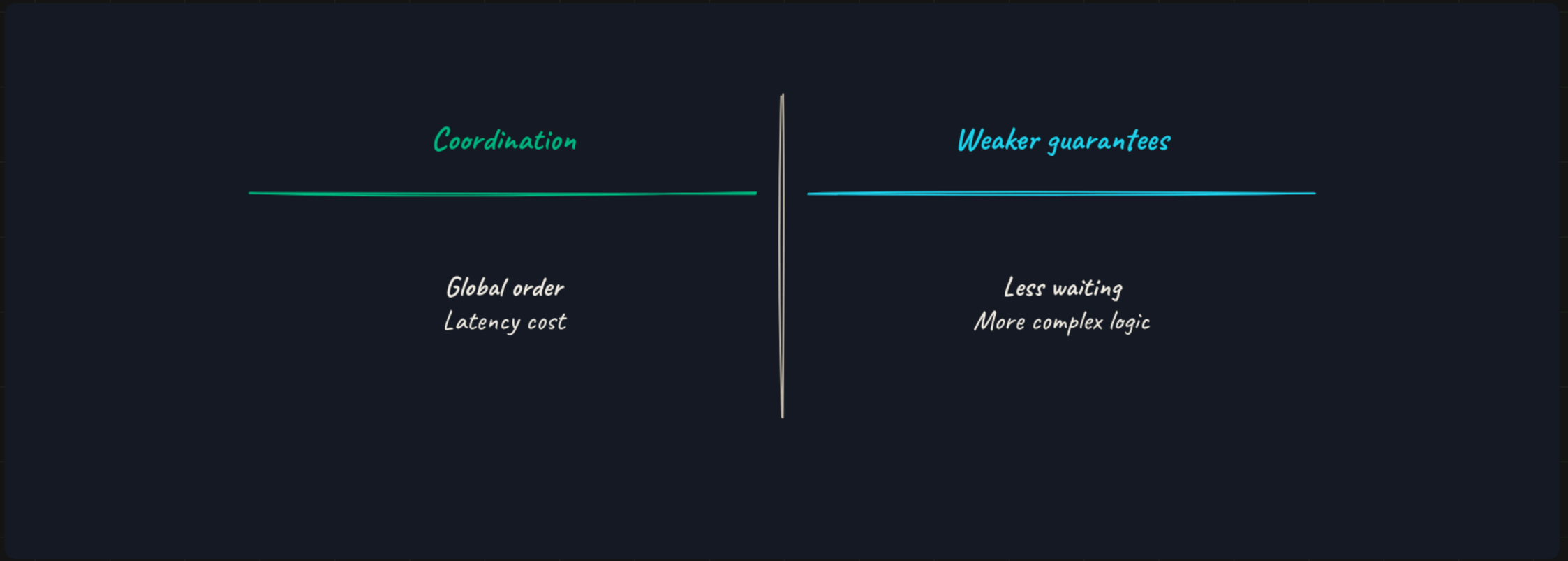
PACELC Chapter Motivation



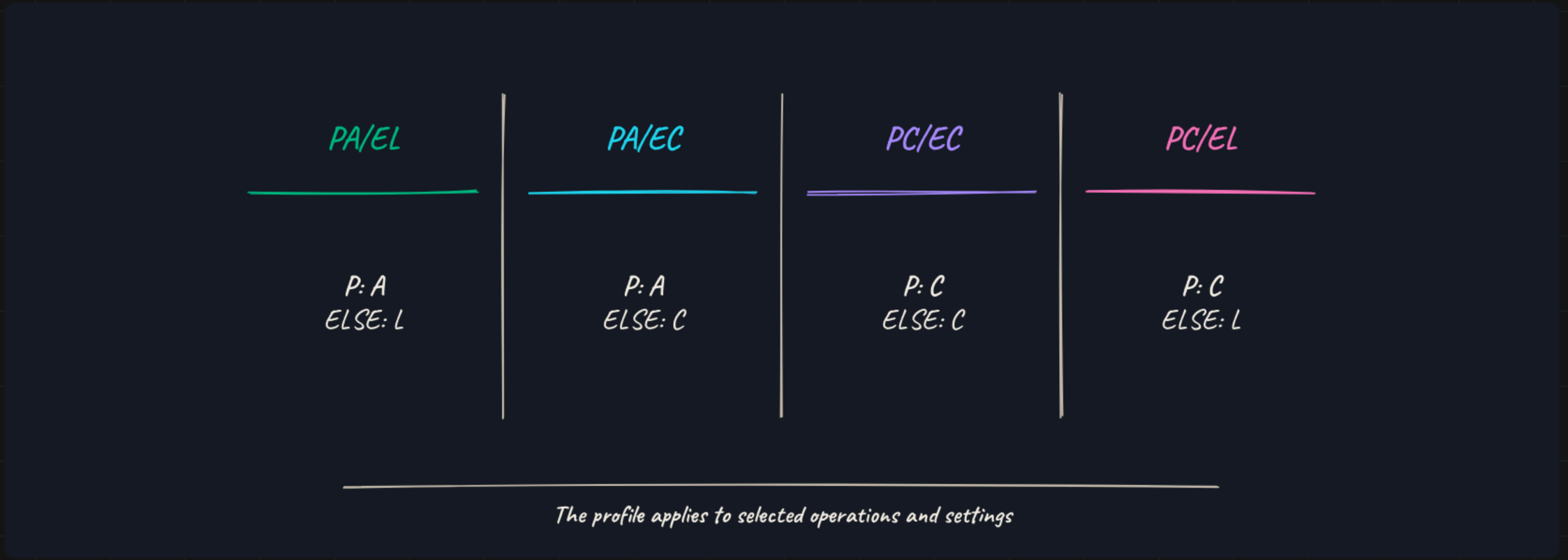
PACELC explains low-latency choices without a failure

Latency vs Consistency in Normal Mode

PACELC Chapter Trade-off

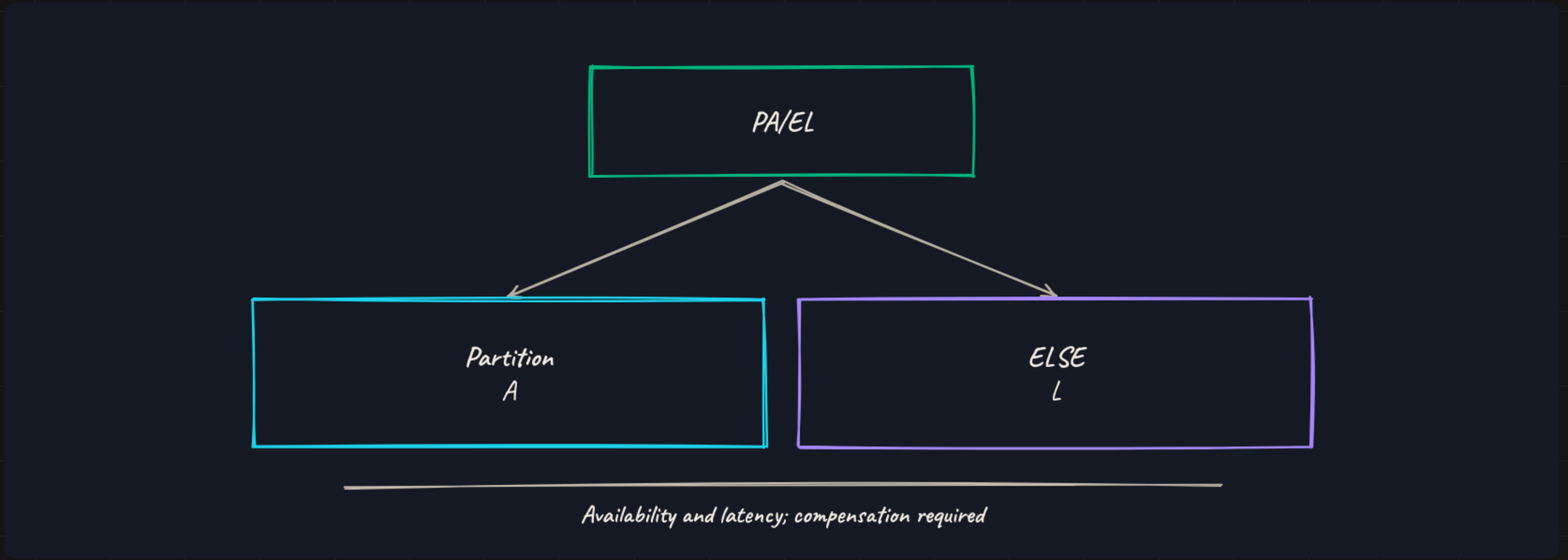


PACELC Categories



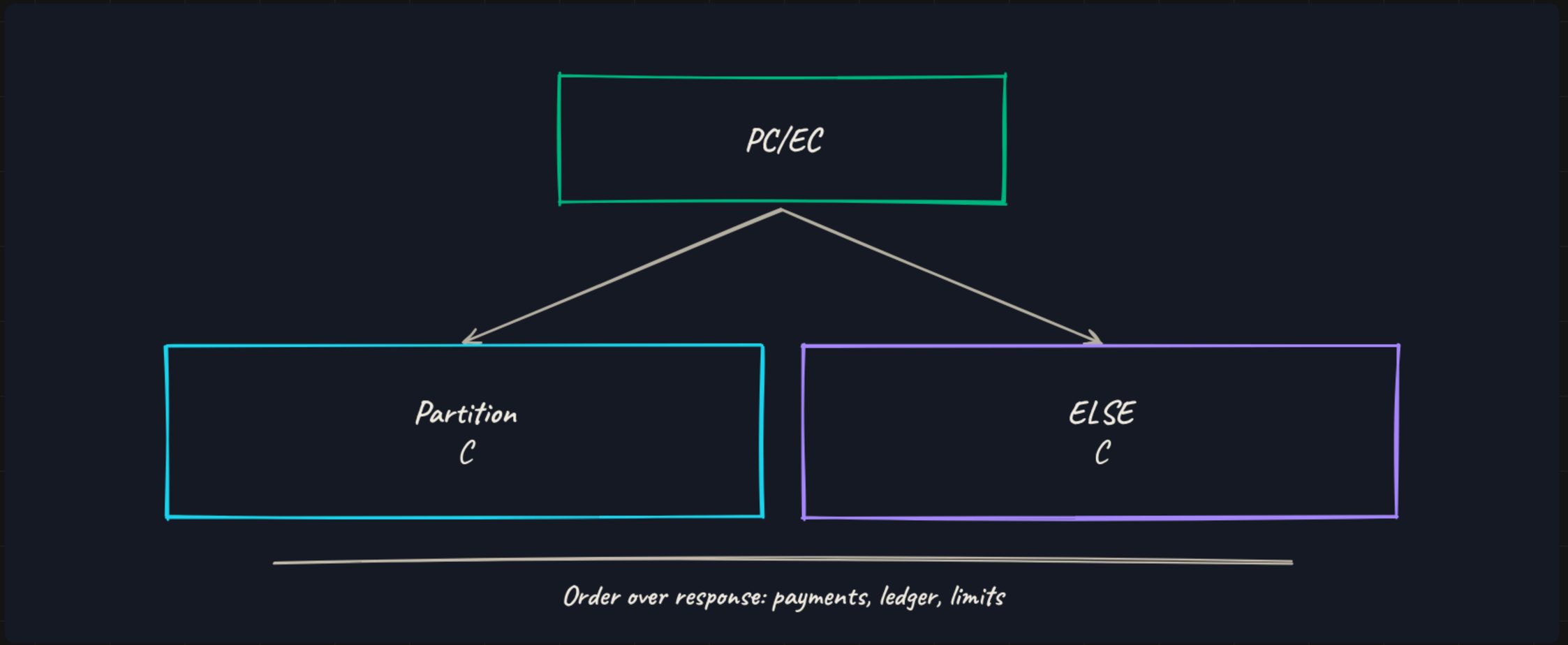
PA/EL: Speed and Availability First

PACELC Chapter PA/EL



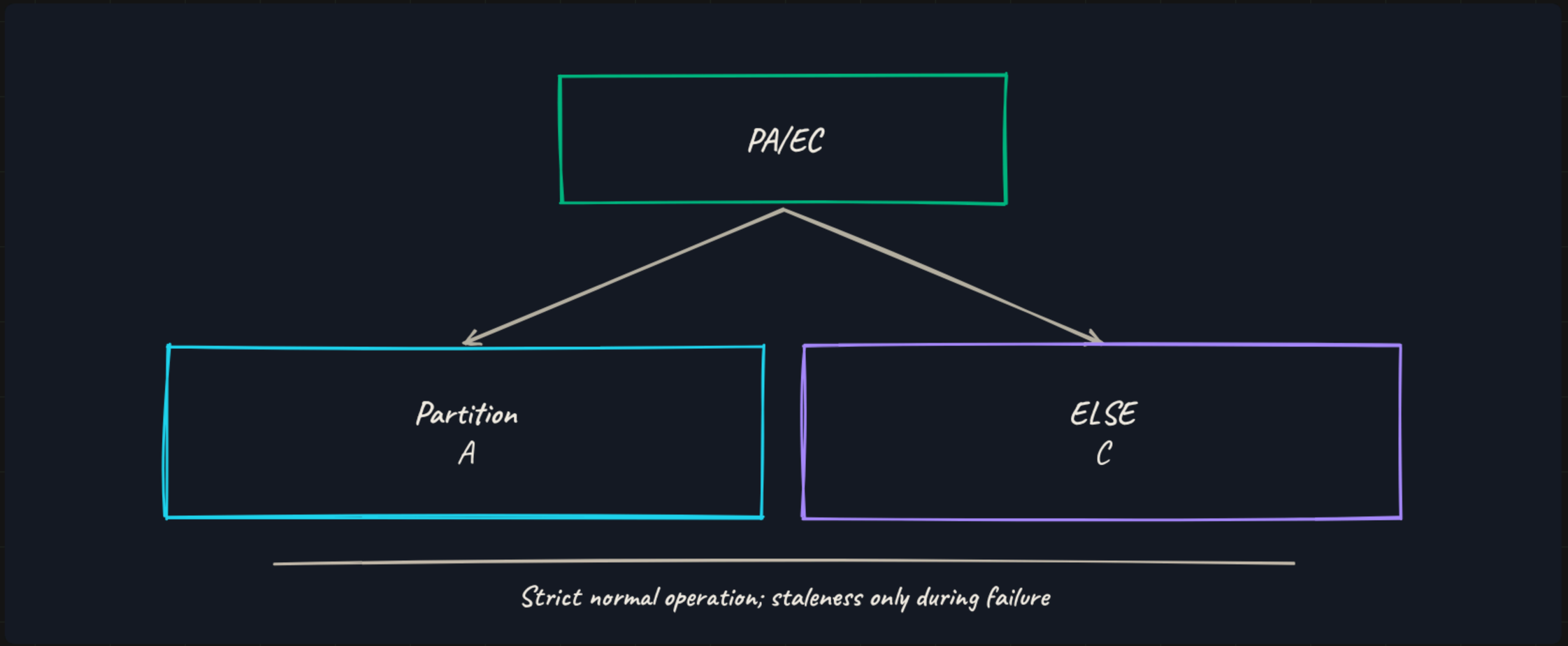
PC/EC: Consistency First

PACELC Chapter PC/EC



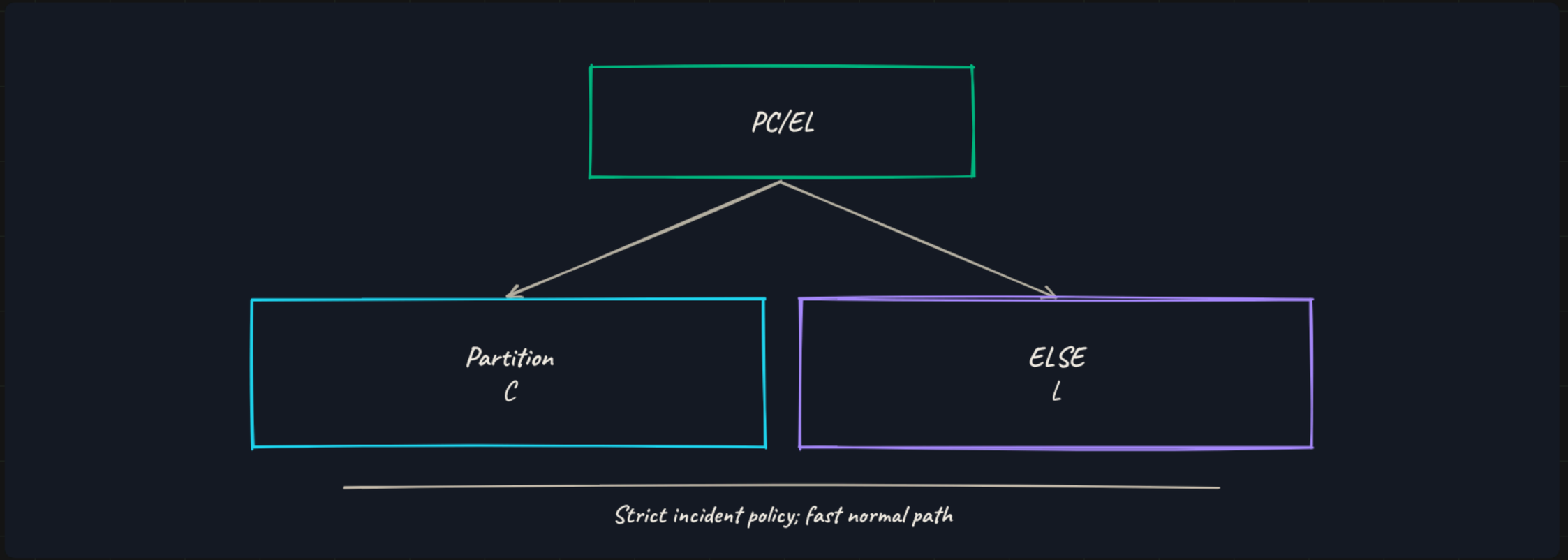
PA/EC: Reliable and Strict

PACELC Chapter PA/EC



PC/EL: Rare but Important Profile

PACELC Chapter PC/EL



Map PACELC to Product Needs

PACELC Chapter Business Mapping

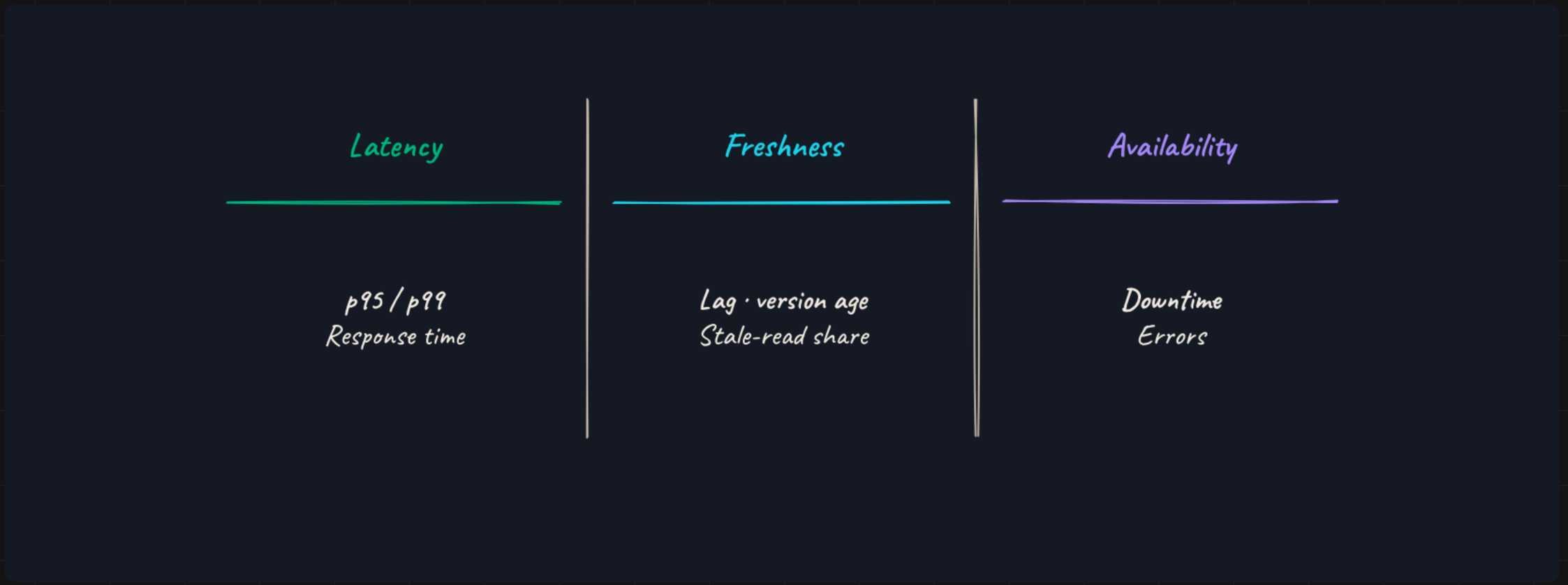
Prioritize C

Payments and limits
Legal ledger

Prioritize L / A

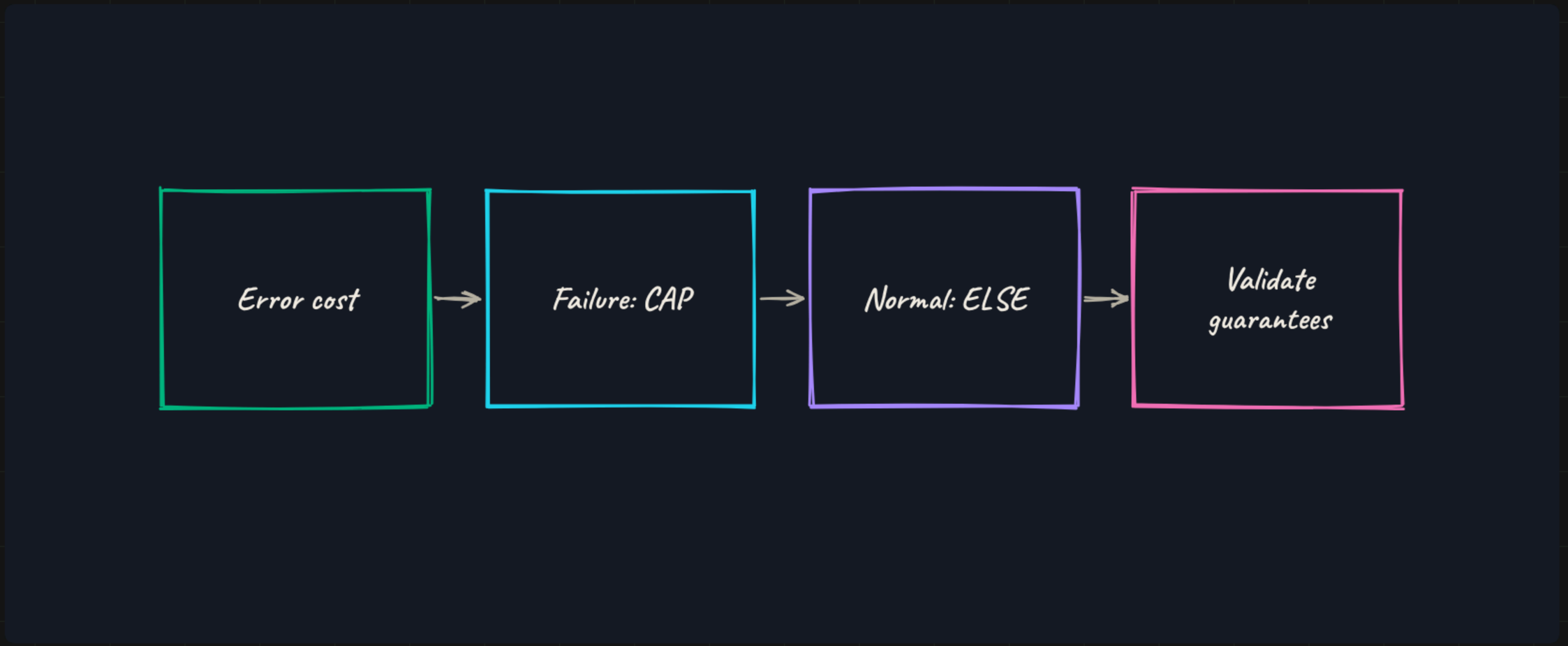
Feeds and search
Allowed drift

Calculate Latency/Consistency Budget



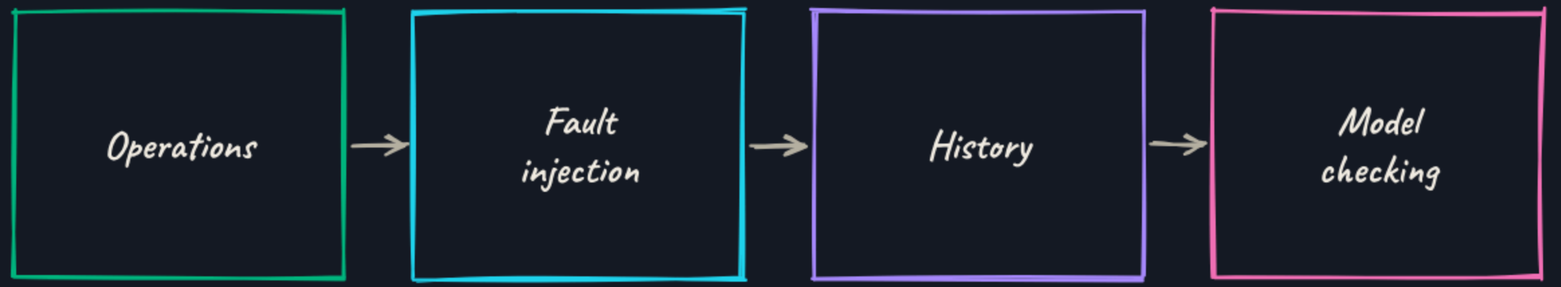
What PACELC Adds to CAP

PACELC Chapter Summary



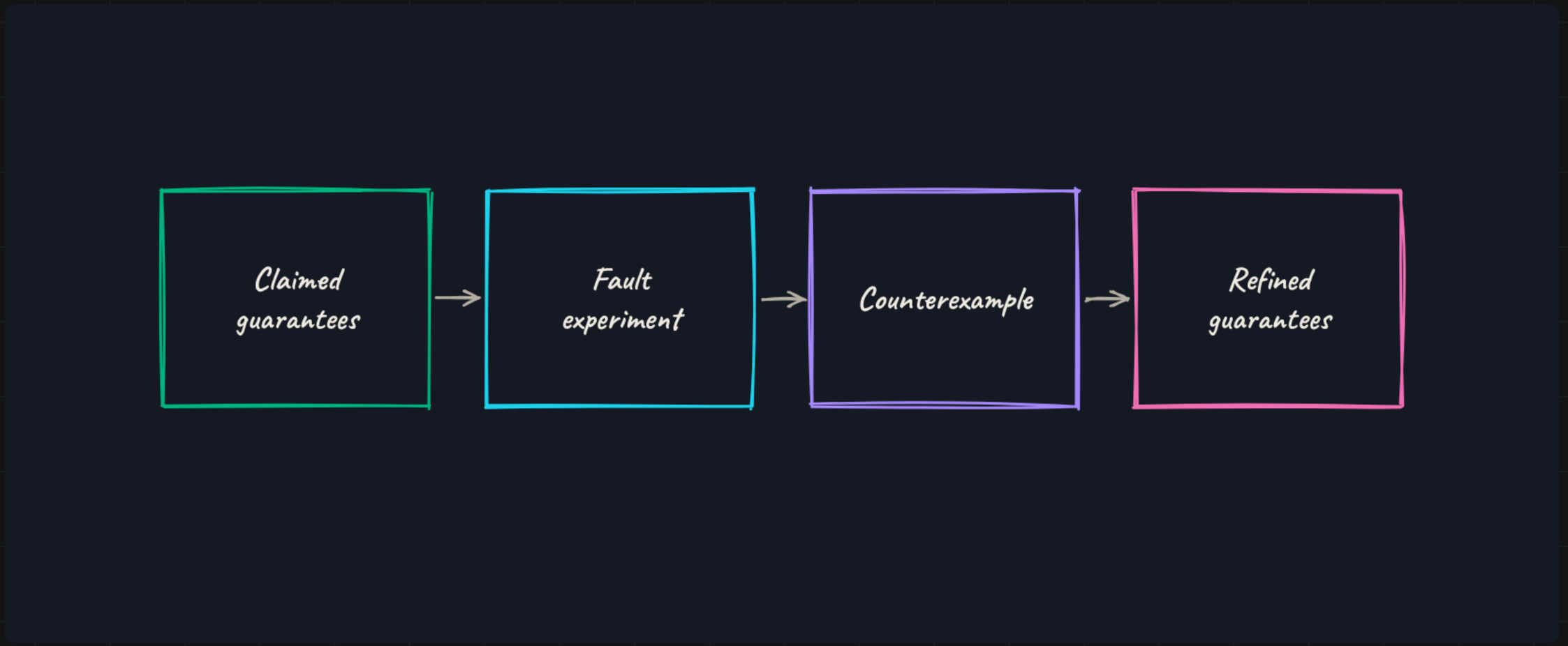
Jepsen: Stress-Testing Distributed Systems

Jepsen Chapter Practice



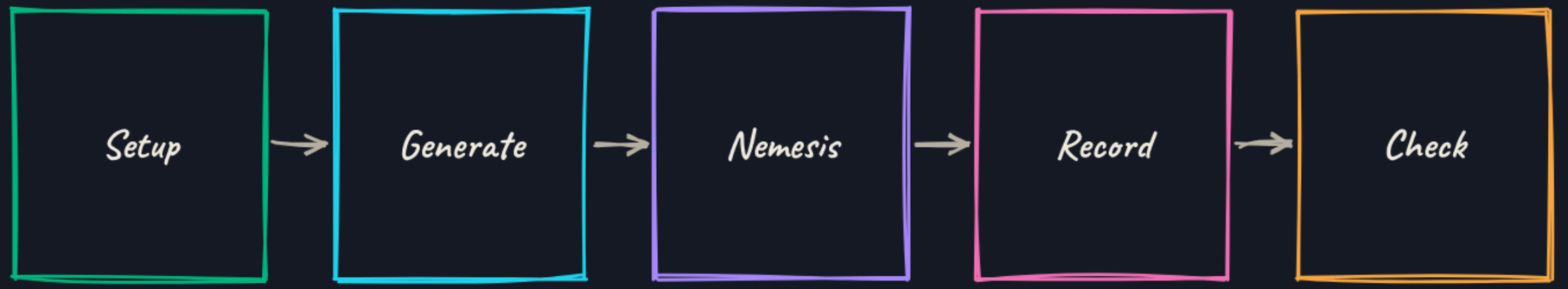
Why Architects Read Jepsen

Jepsen Chapter Value



Jepsen's Five-Step Cycle

Jepsen Chapter Method



Check the operation history against a formal model

Jepsen Failure Models

Jepsen Chapter Nemesis

Network

Partition

Asymmetry · jitter

Processes and time

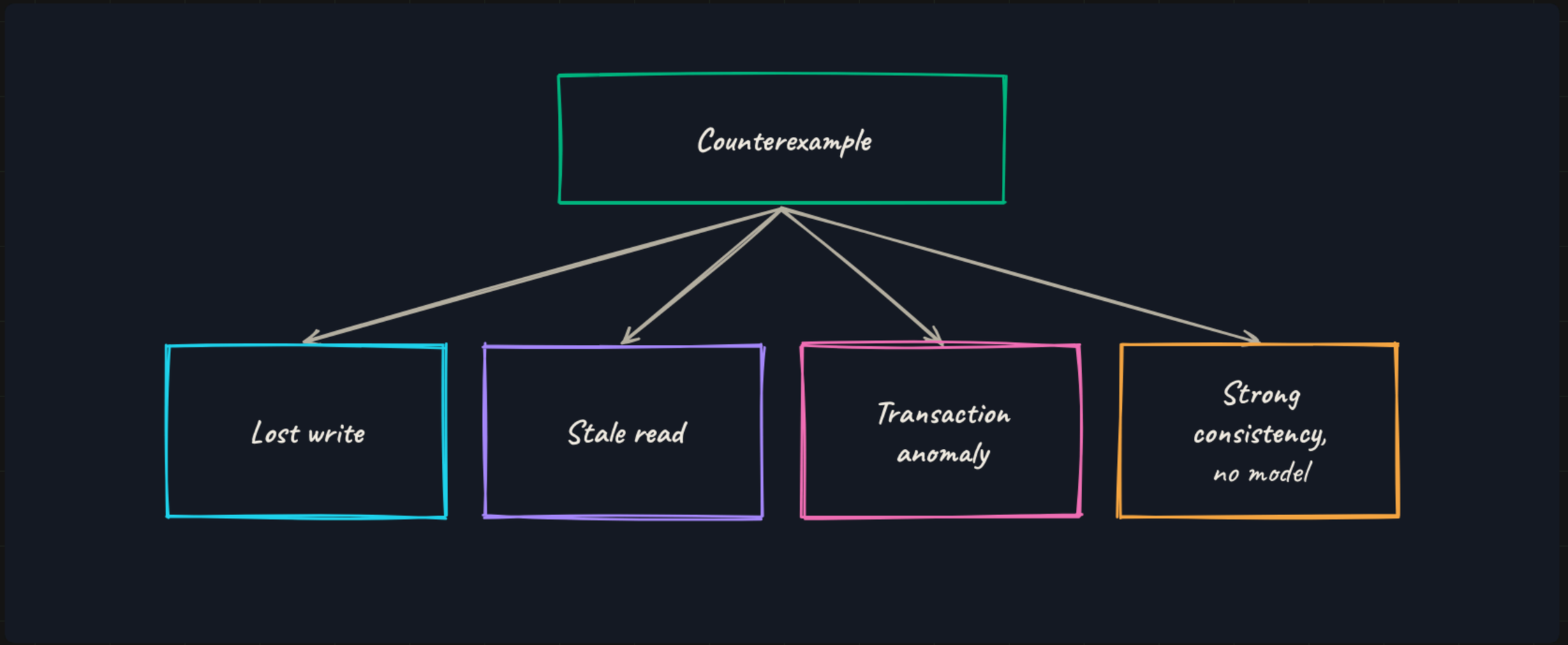
Crash / restart

Clock skew

Failure scenarios may overlap

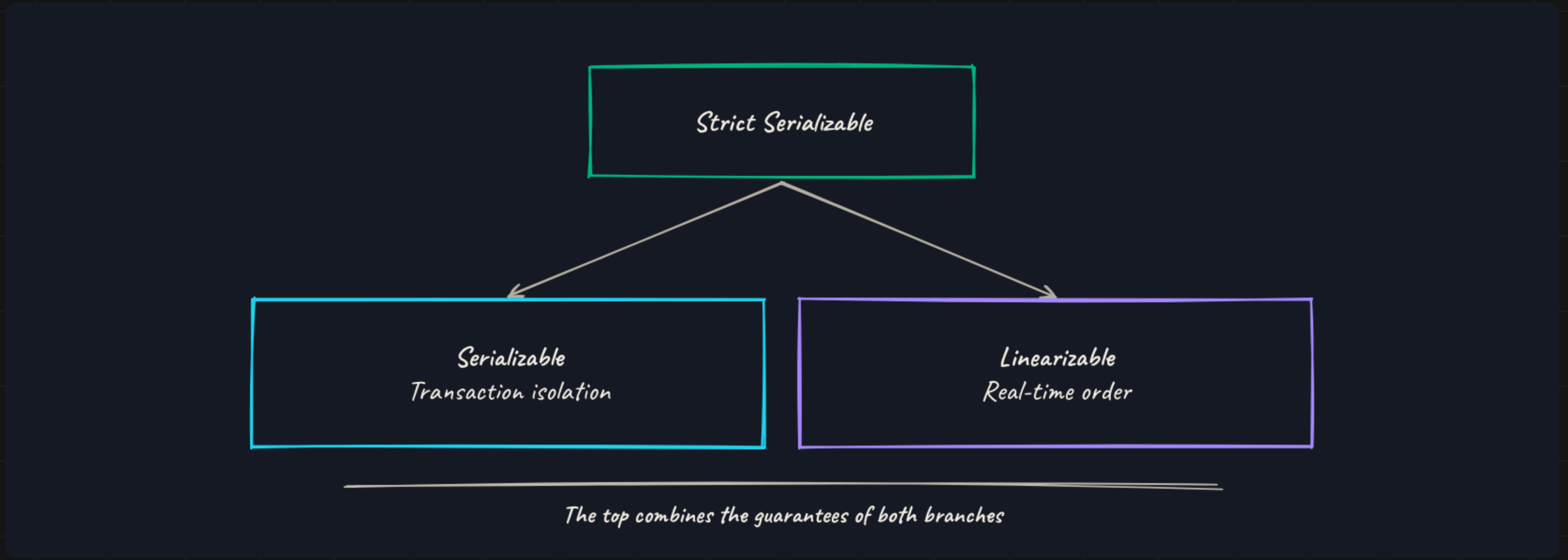
Typical Jepsen Findings

Jepsen Chapter Findings

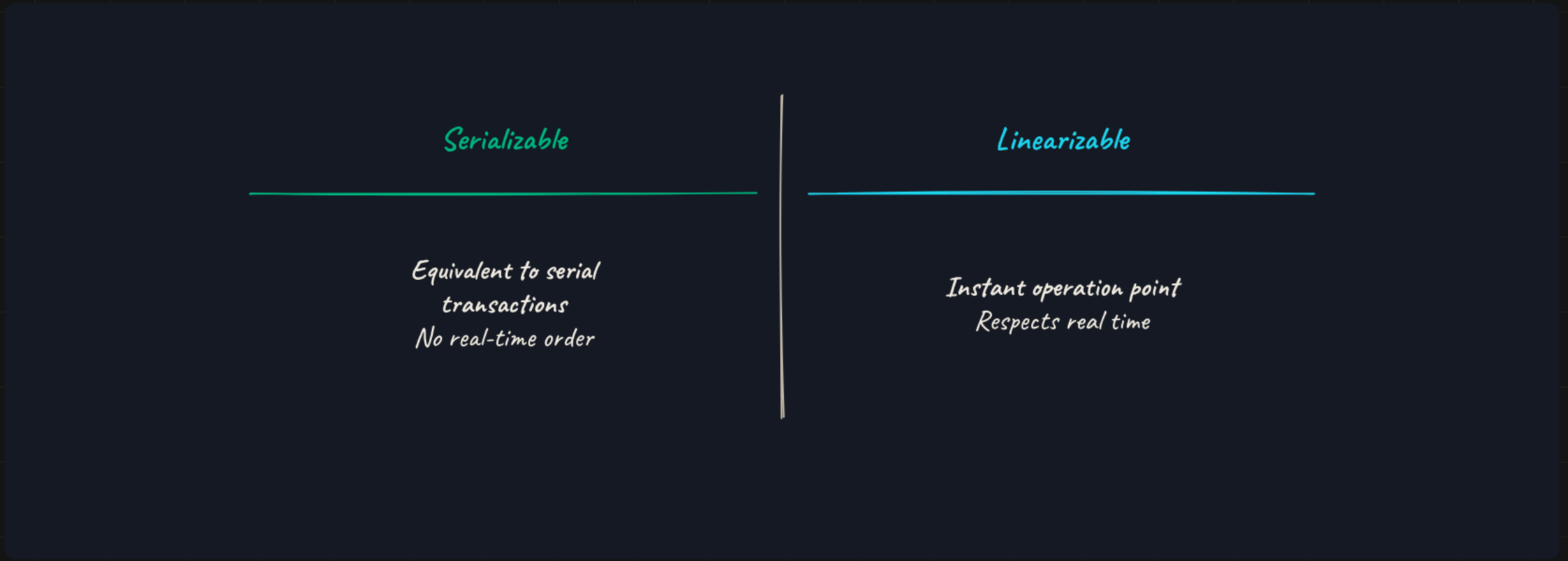


Jepsen Model Map: Two Branches, One Apex

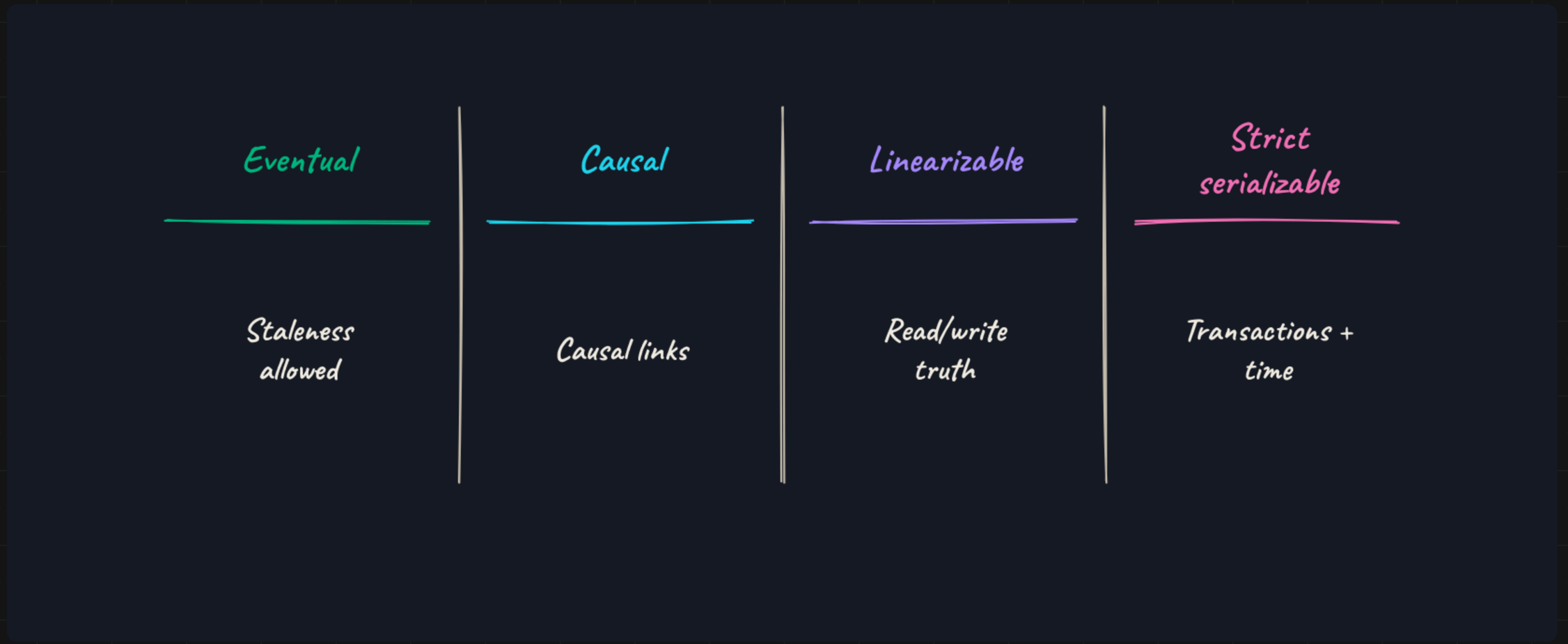
Consistency Models Hierarchy



Serializable vs Linearizable



Choose a Model for Product

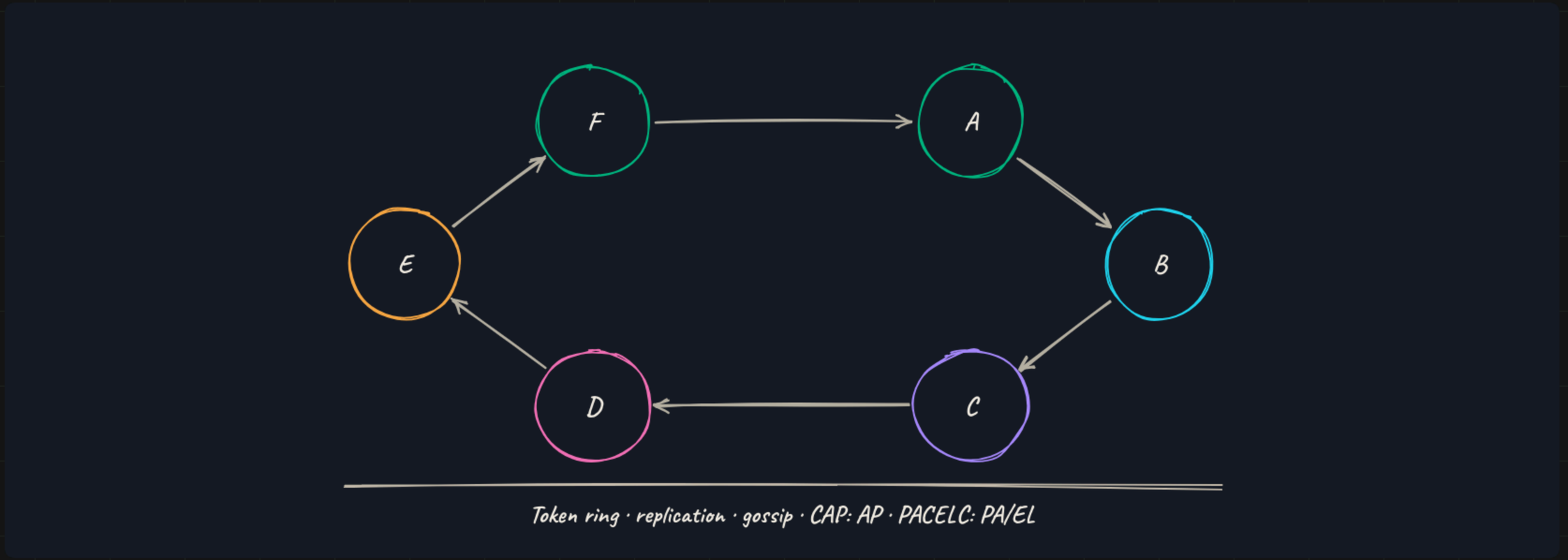


Overall Framework: CAP + PACELC + Jepsen



Cassandra Through CAP/PACELC

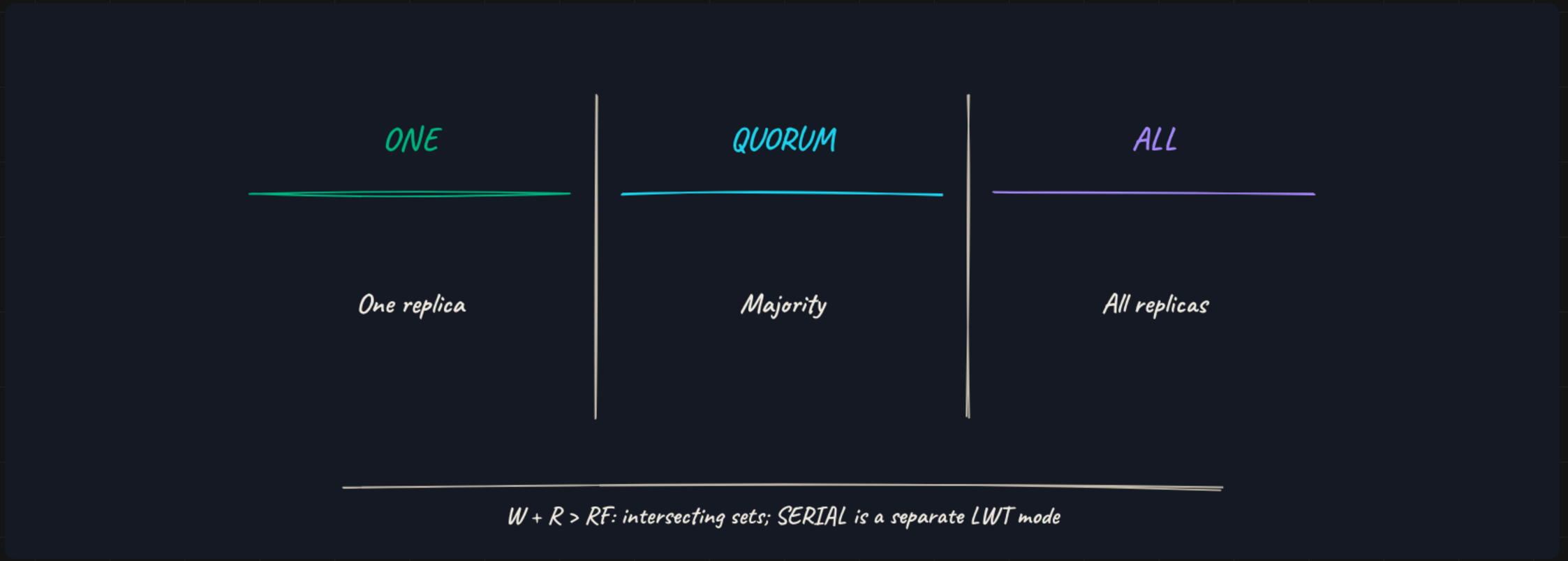
Bonus Case Apache Cassandra



Cassandra Tunable Consistency

Bonus Case · Cassandra

Consistency



Bonus Case · Cassandra

Architecture

How Cassandra Maintains Scale and Availability

Write path and storage

Commit Log → MemTable → SSTable
Compaction removes old versions
LSM optimizes writes

Distribution and resilience

Consistent hashing + vnodes
Gossip detector, no master
Repair drives convergence

References and Materials

- CAP: ↗system-design.space
- PACELC: ↗system-design.space
- Consistency Models (Jepsen): ↗system-design.space
- Cassandra: ↗system-design.space

- All 4 source chapters from ↗system-design.space in one place.