ЦЕНТРАЛЬНЫЙ УНИВЕРСИТЕТ Лекция

CAP, PACELC и модели консистентности

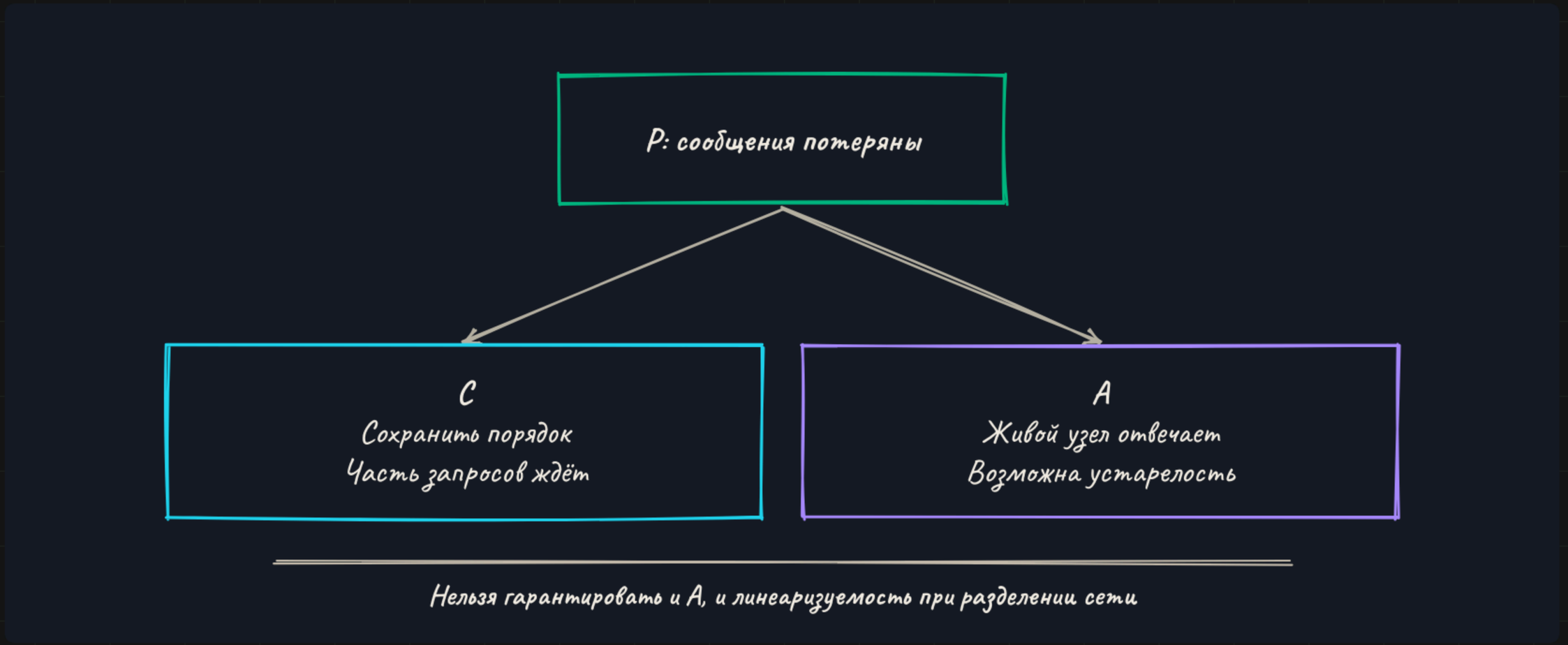
История про реальные компромиссы распределенных систем

Карта лекции

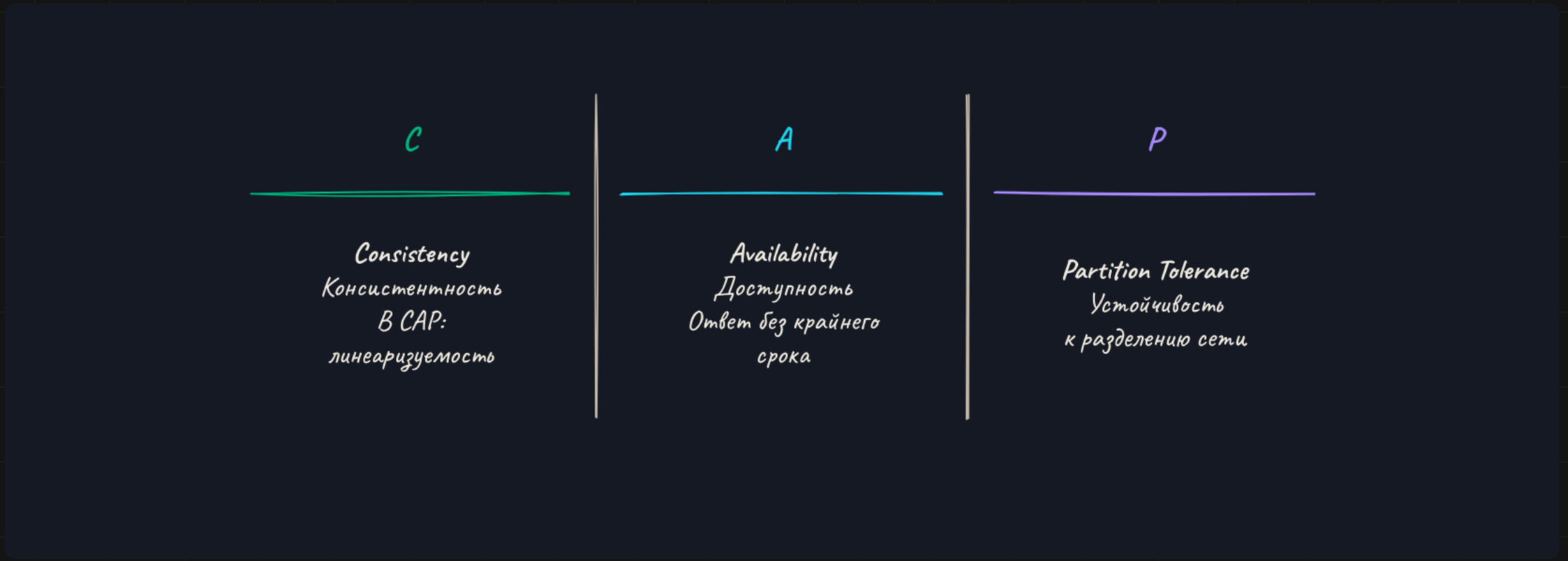
План Навигация



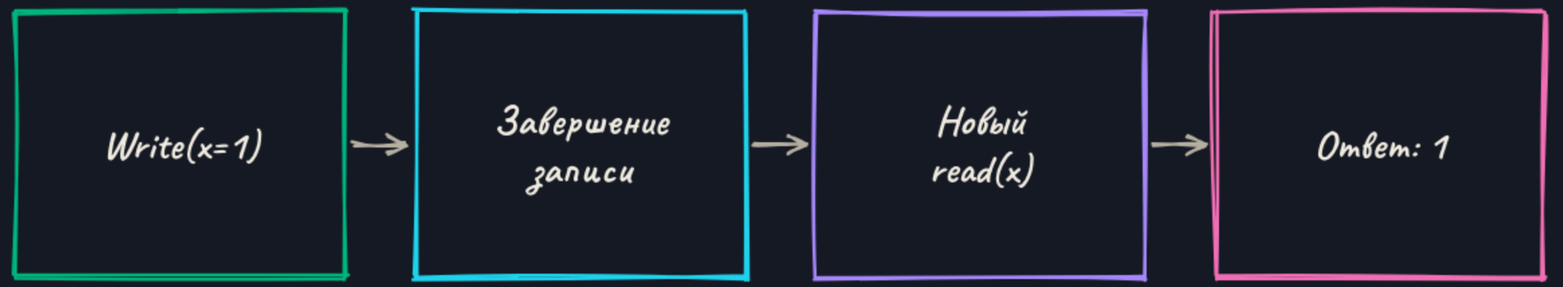
CAP-теорема в одной фразе



C, A и P: что внутри



Консистентность в CAP = линеаризуемость



Глобальный порядок уважает реальное время операций

Доступность в теореме



Срок ответа не фиксирован; бесконечное ожидание нарушает A

Как появилась CAP



Миф «выбери 2 из 3»

Без разделения сети

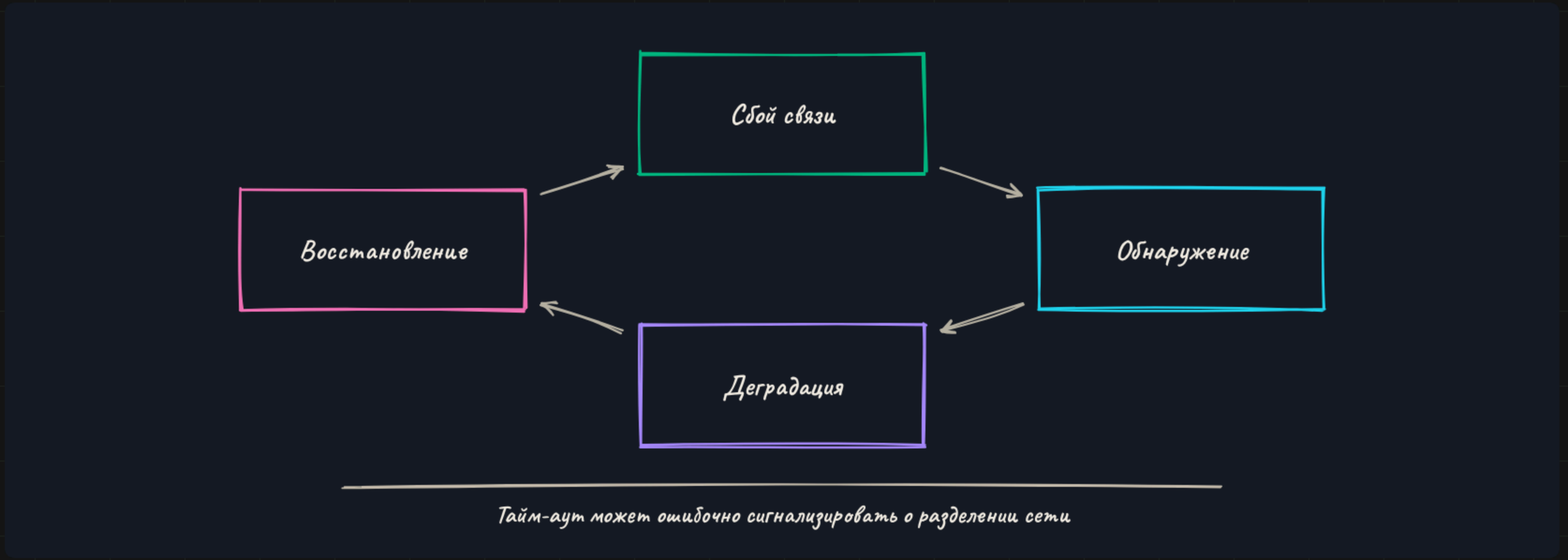
C и A совместимы

При разделении сети

Выбор по операциям
CP либо AP

CAP — не статичная этикетка «два из трёх»

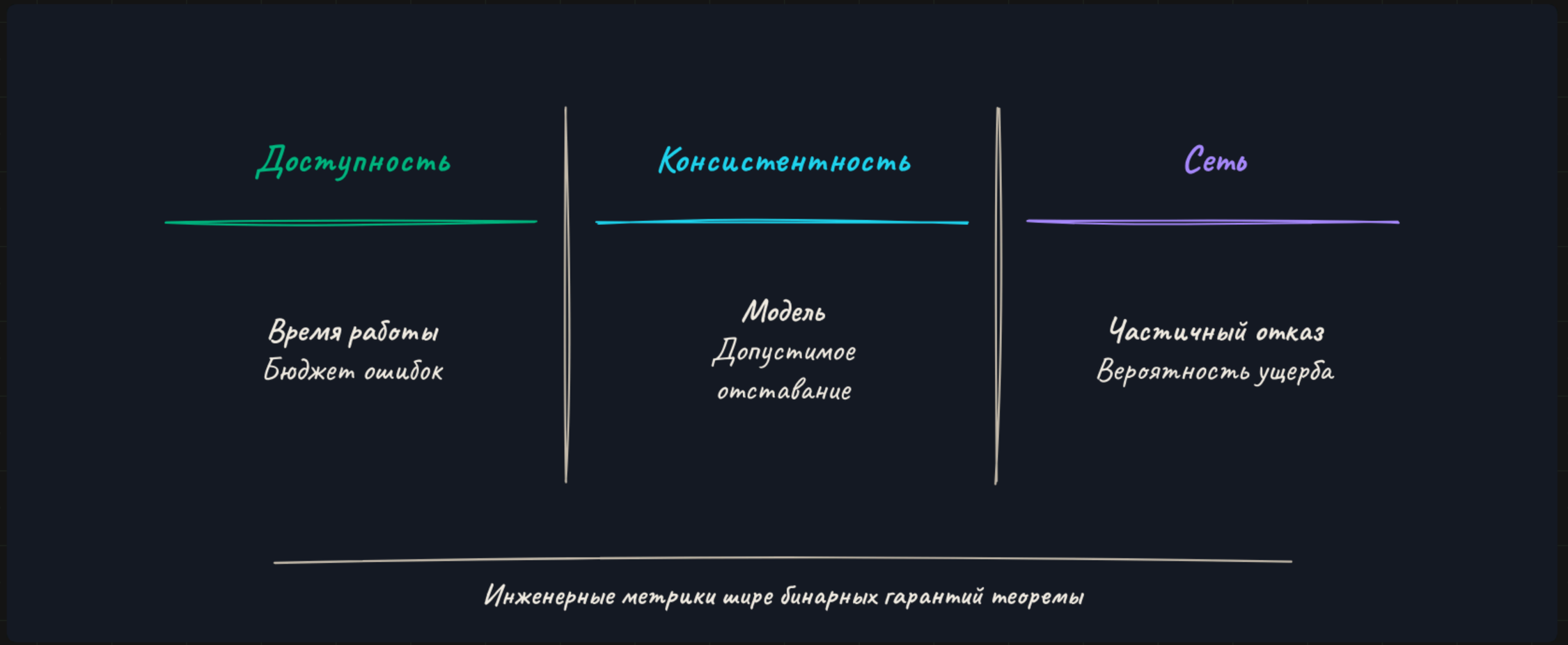
Разделение сети: редкость, не экзотика



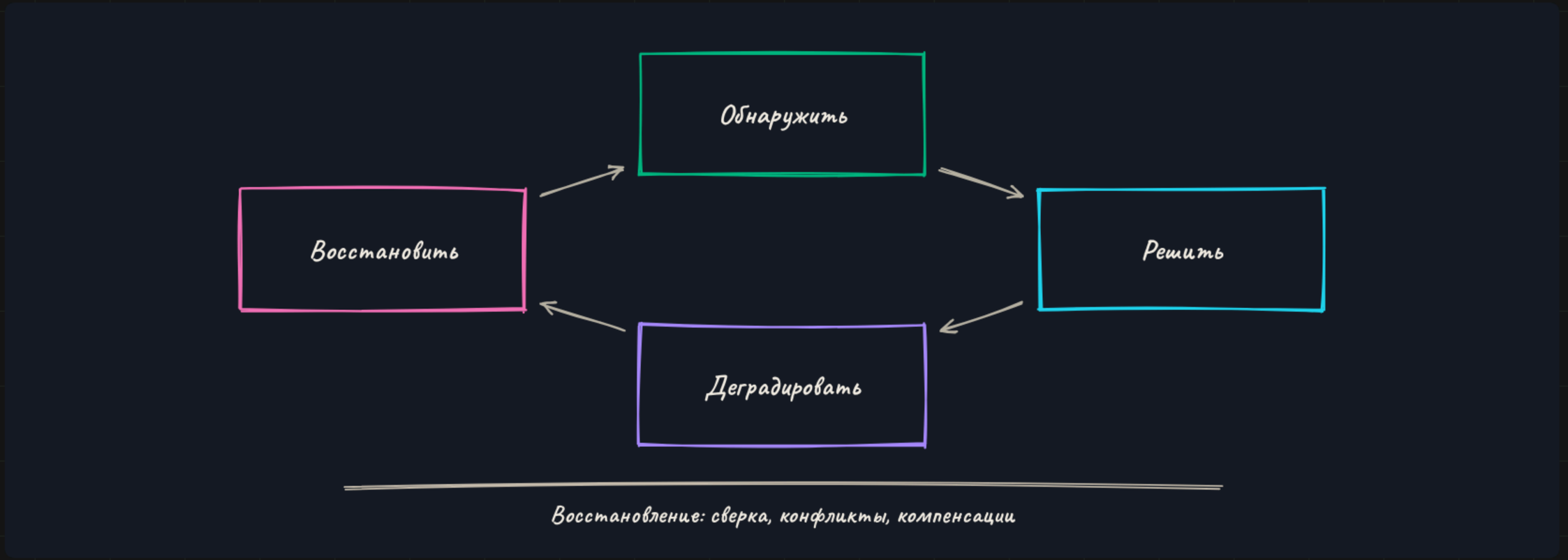
Решение принимается по операциям



C/A/P — спектры


Доступность
Время работы
Бюджет ошибок
Консистентность
Модель
Допустимое отставание
Сеть
Частичный отказ
Вероятность ущерба
Инженерные метрики шире бинарных гарантий теоремы

Режим разделения сети как конечный автомат

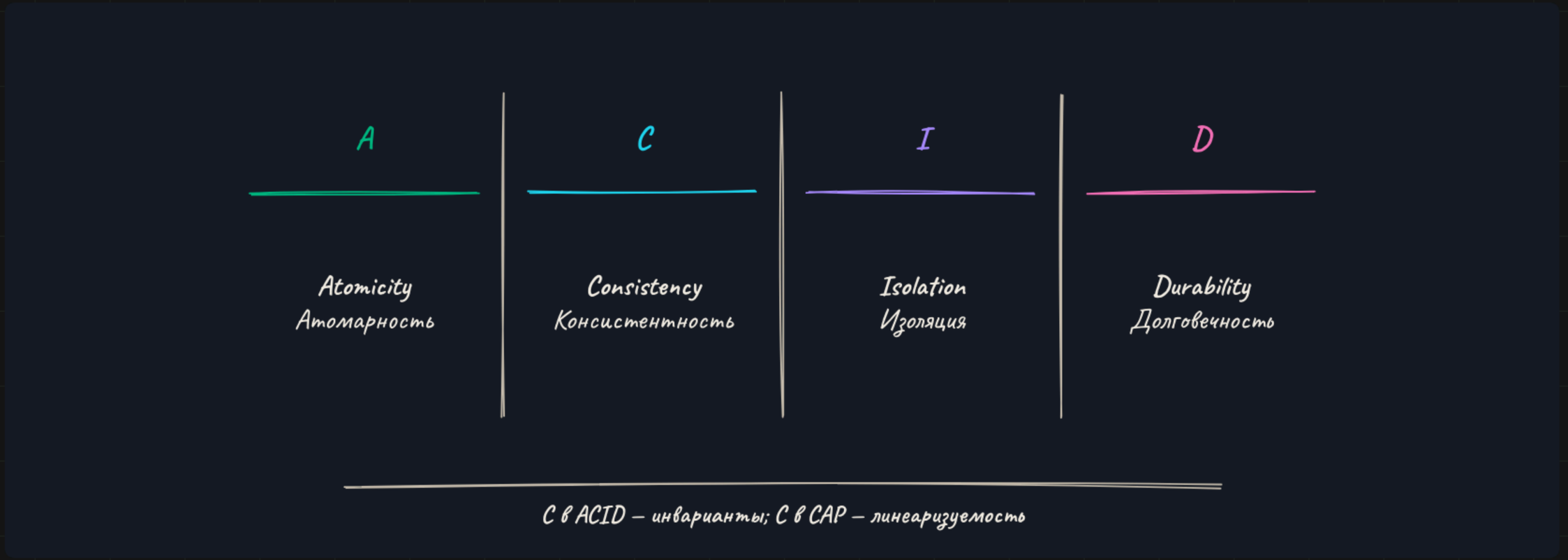


Задержка управляет CAP

Ждать синхронизации	Ответить раньше
Сохранять C Риск потерять A	Сохранять A Риск устарелости

Тайм-аут обнаруживает симптом, но не устраняет компромисс

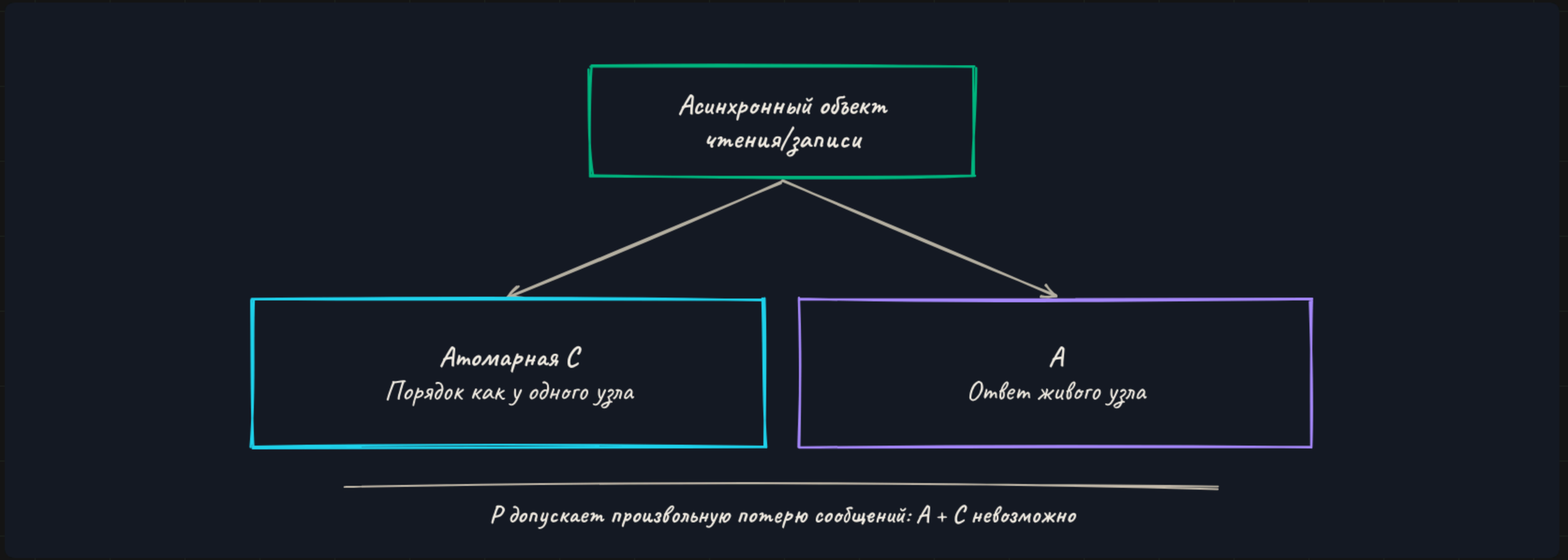
ACID vs CAP: разная буква C



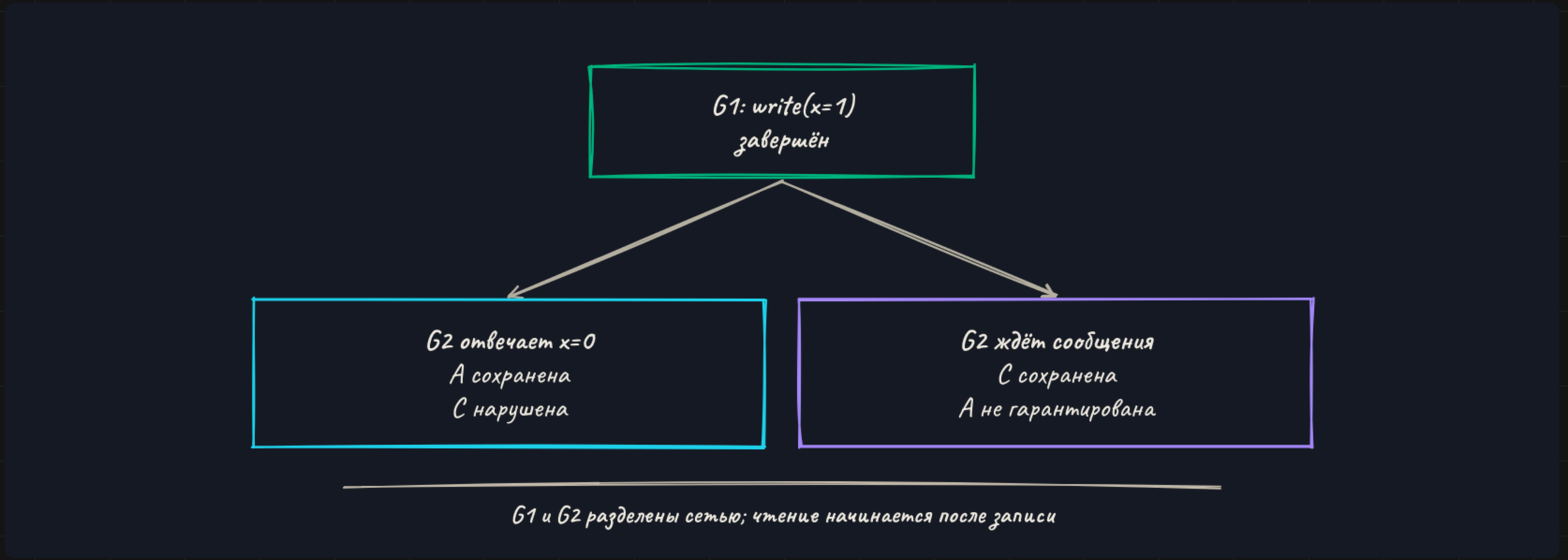
BASE: ответ на доступность



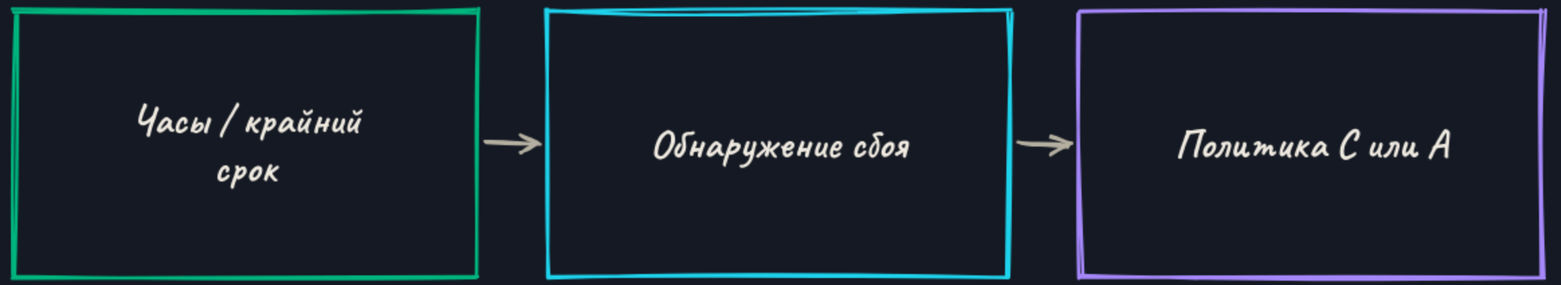
Gilbert–Lynch: формальные определения



Асинхронная невозможность



Частичная синхронность не спасает



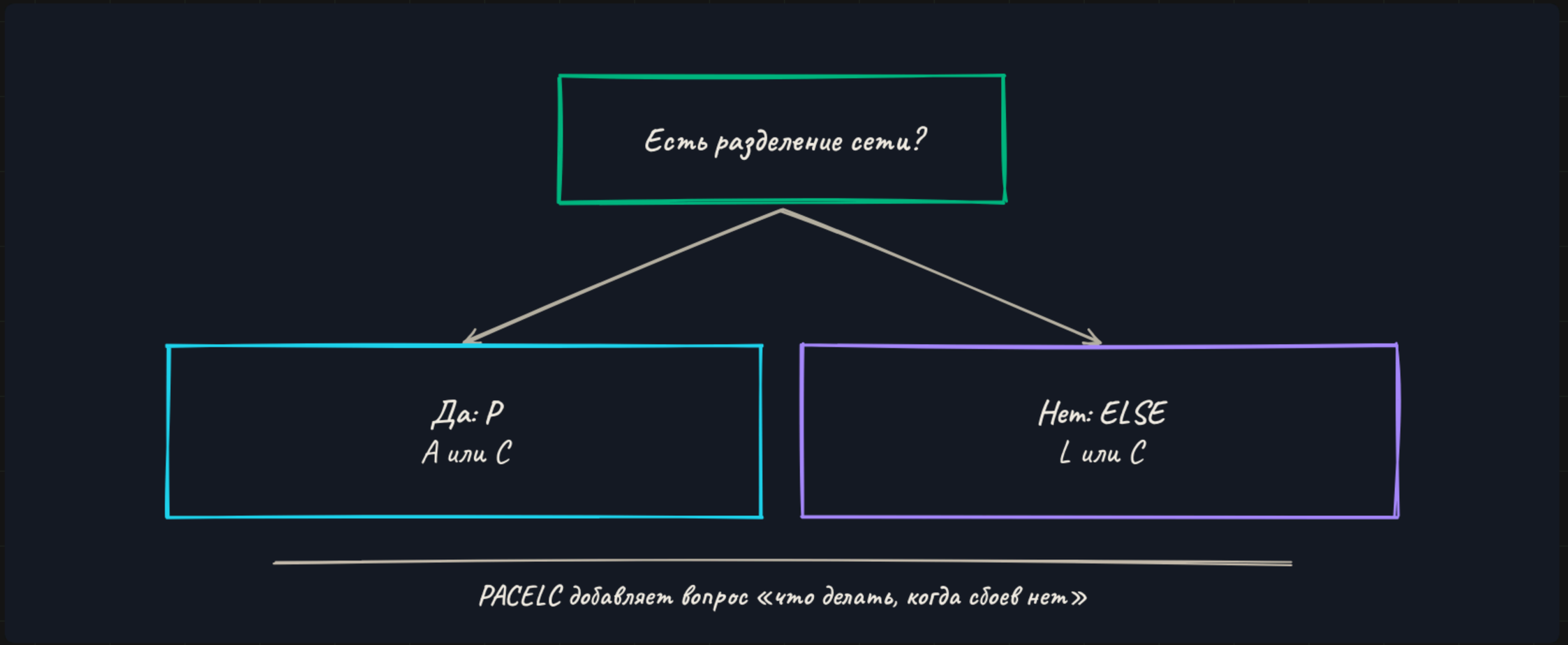
Потеря сообщений сохраняет конфликт даже при частичной синхронности

Что берем из CAP в реальную архитектуру

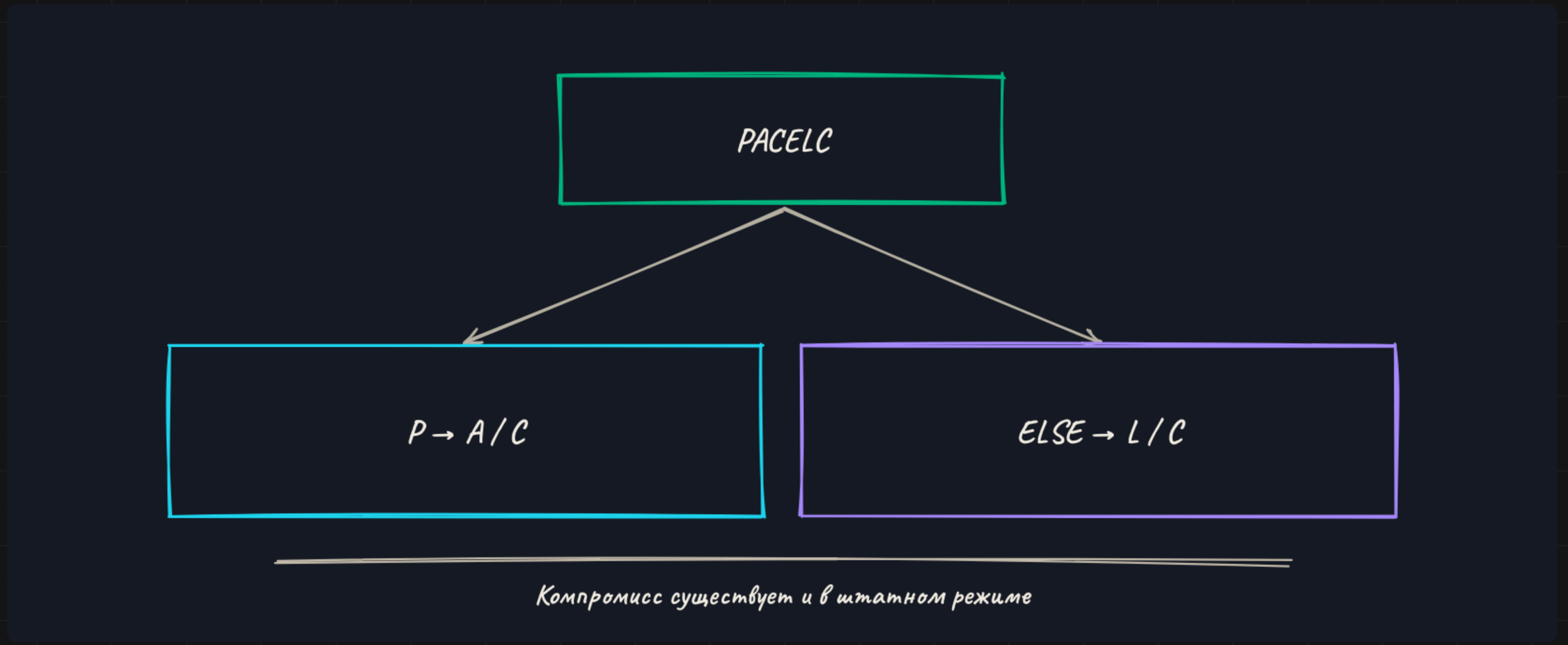


CAP: что делать при сбое

Переход PACELC



Формула PACELC



Зачем появилась PACELC



PACELC объясняет выбор низкой задержки без аварии

Задержка vs консистентность в штатном режиме

Глава PACELC

Компромисс

Координация

Глобальный порядок

Цена задержки

Слабее гарантии

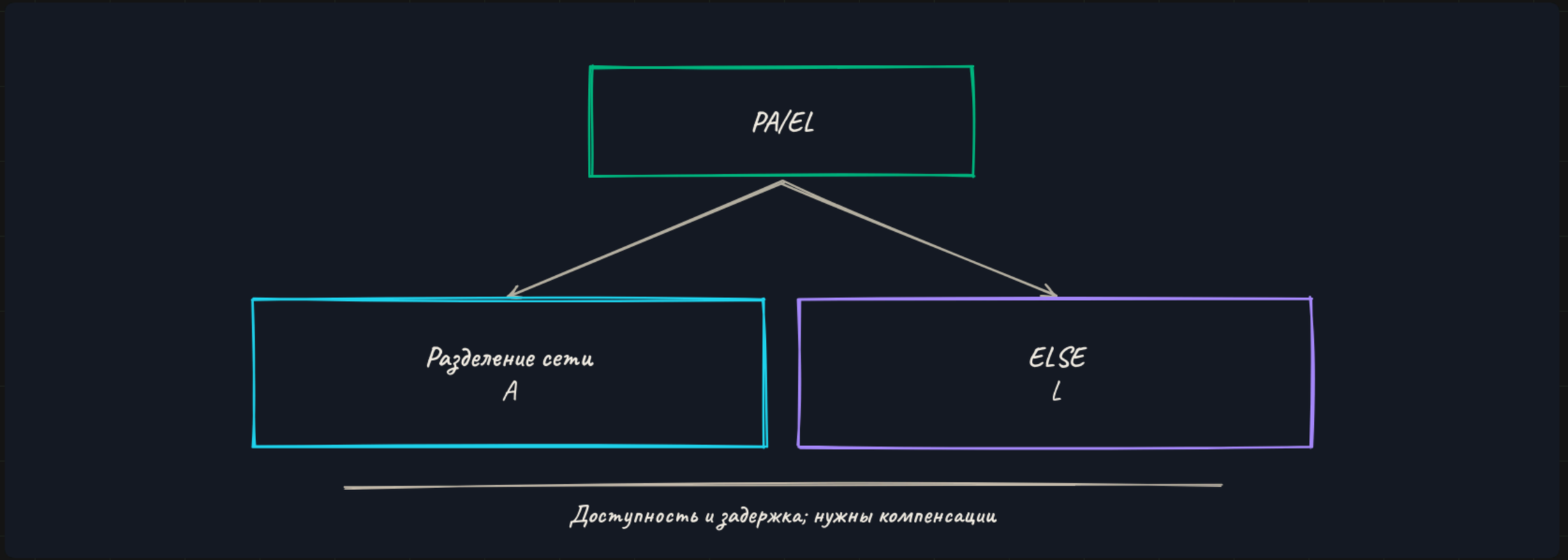
Меньше ожидания

Сложнее логика

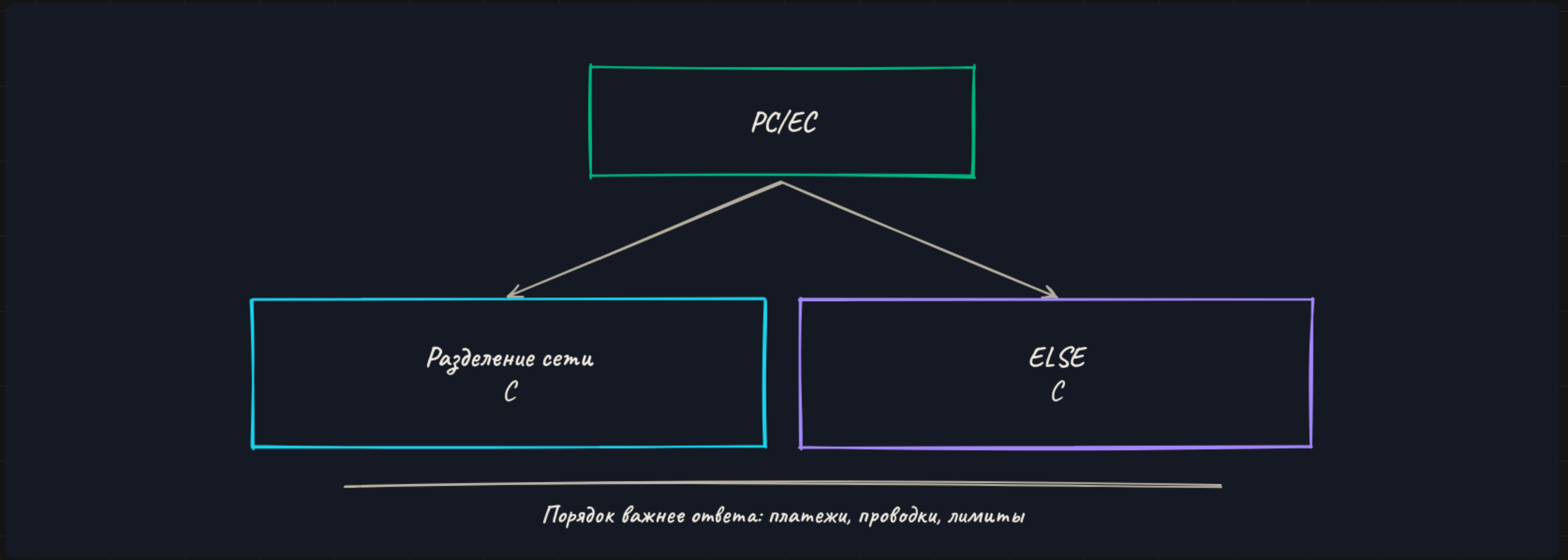
Четыре категории PACELC



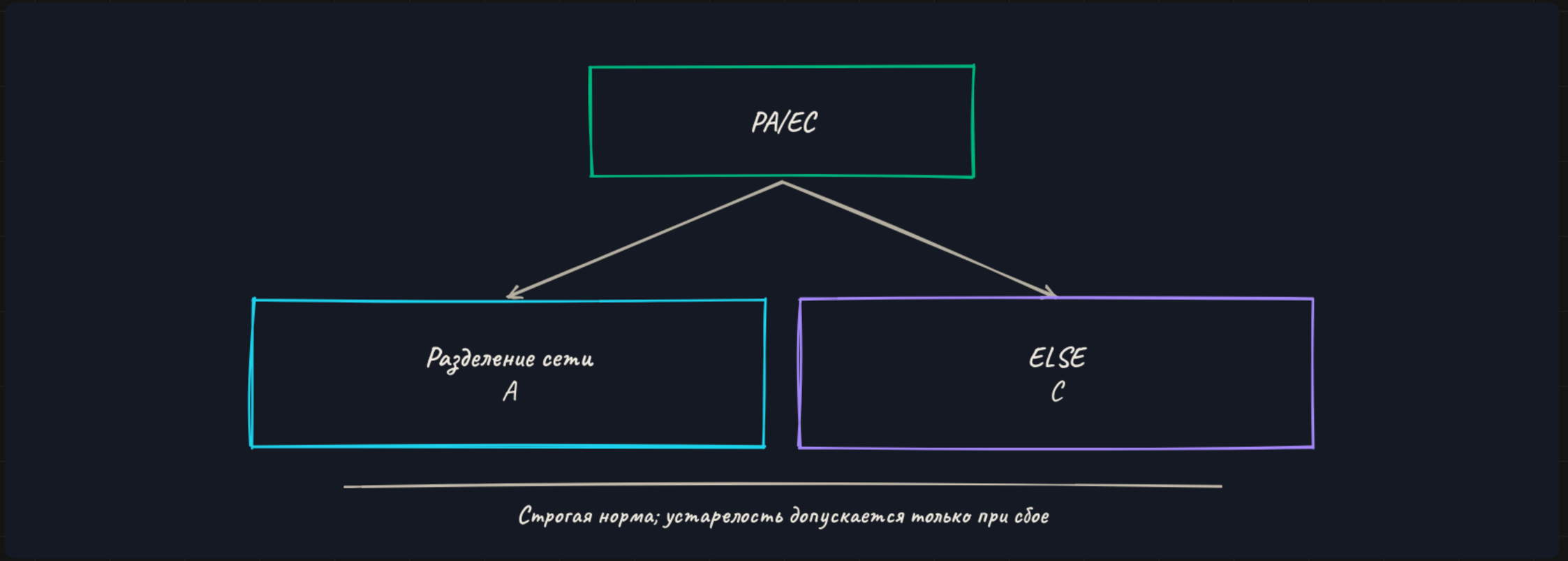
PA/EL: скорость и доступность в приоритете



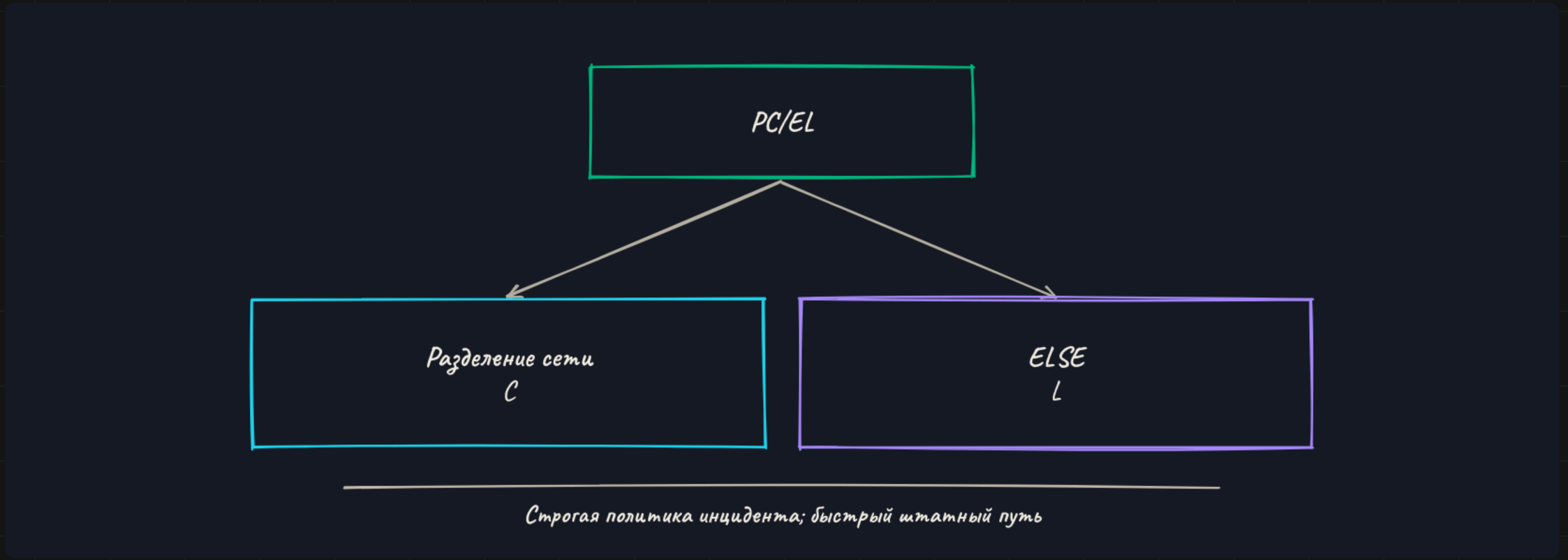
PC/EC: консистентность прежде всего



PA/EC: надежность и строгая норма



PC/EL: редкий, но важный профиль



Как связать PACELC-профиль с продуктовой задачей

Приоритет C

Платежи и лимиты

Юридический журнал

Приоритет L / A

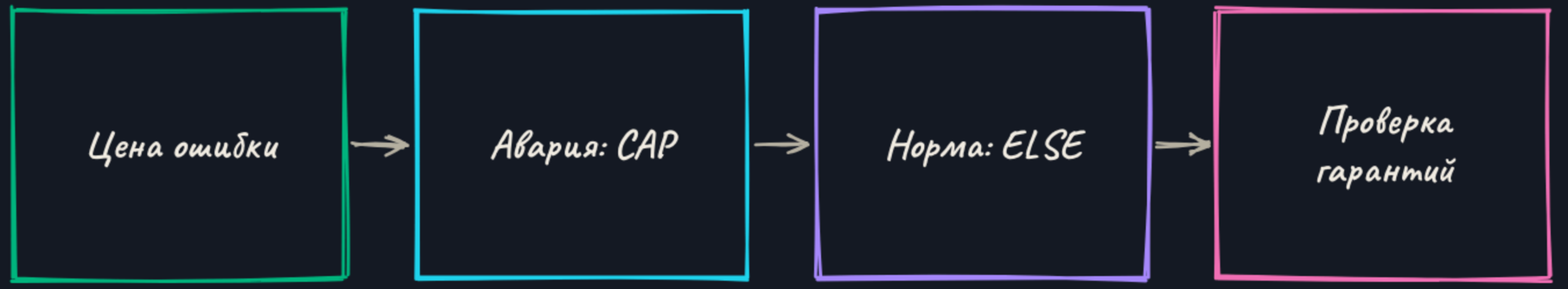
Ленты и поиск

Допустимое расхождение

Как считать бюджет задержки и консистентности



Что PACELC добавляет к CAP



Глава Jepsen

Практика

Jepsen: проверка распределенных систем



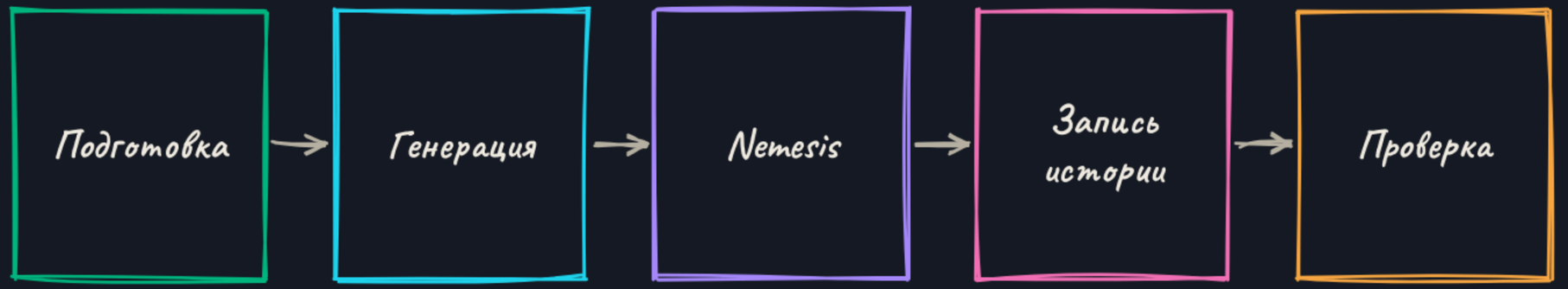
Зачем архитекторам Jepsen

Глава Jepsen Ценность



Пятишаговый цикл Jepsen

Глава Jepsen Метод



История операций проверяется против формальной модели

Какие сбои обычно моделирует Jepsen

Глава Jepsen Nemesis

Сеть

Разделение

Асимметрия · джиттер

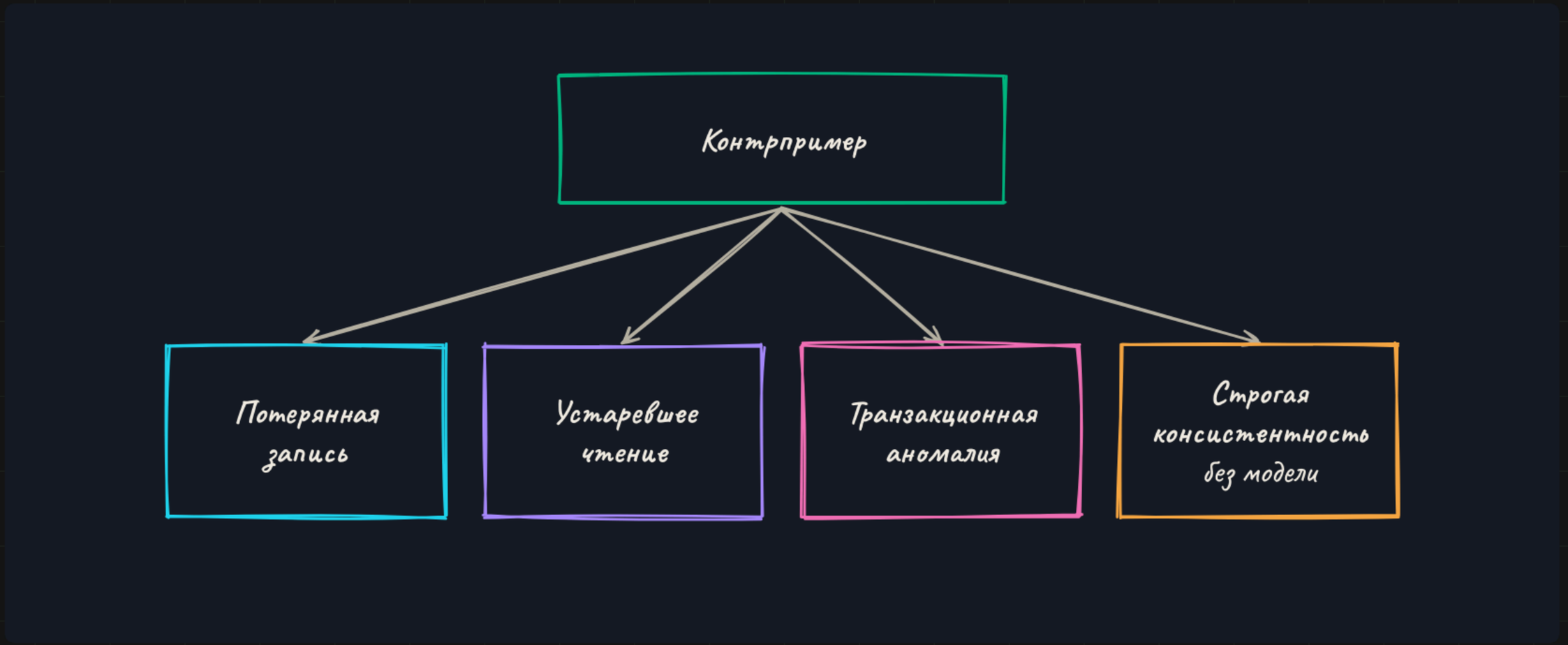
Процессы и время

Падение / перезапуск

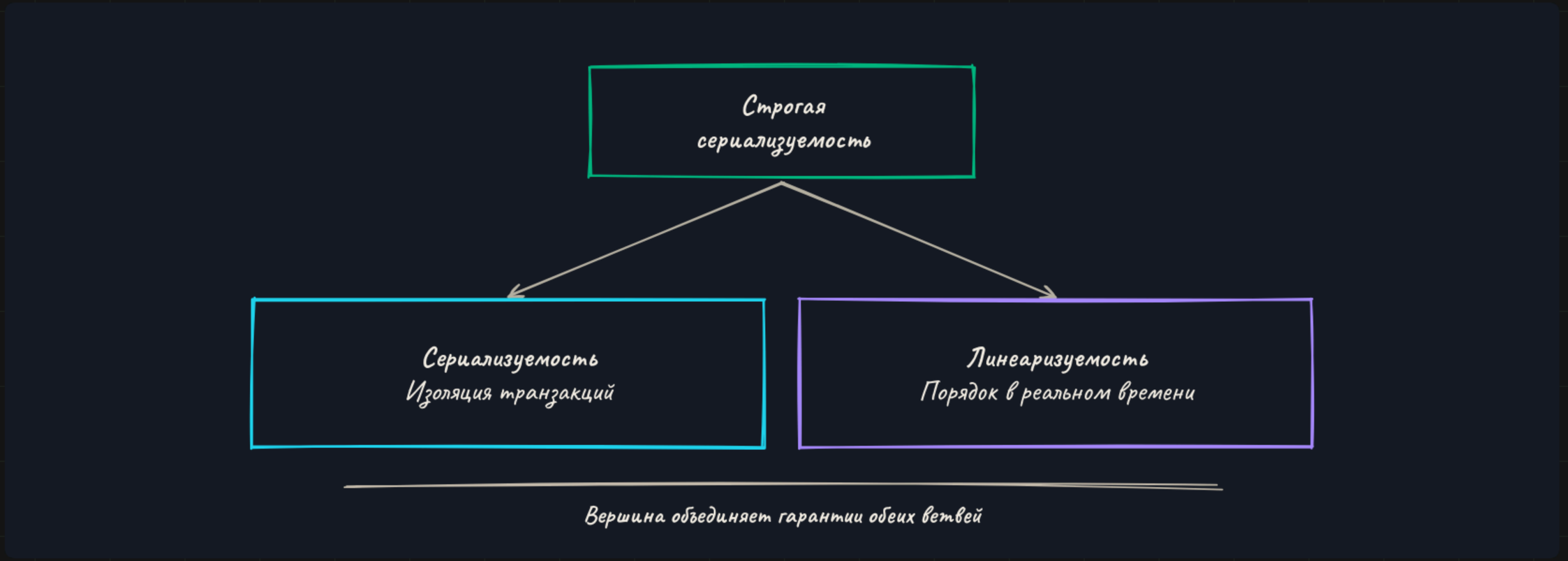
Рассинхрон часов

Сценарии могут накладываться друг на друга

Типичные находки Jepsen



Jepsen-карта моделей: две ветки, одна вершина



Сериализуемость vs линеаризуемость

Сериализуемость

Как последовательные транзакции
Без порядка реального времени

Линеаризуемость

Мгновенная точка операции
Учитывает реальное время

Выбор модели под продукт

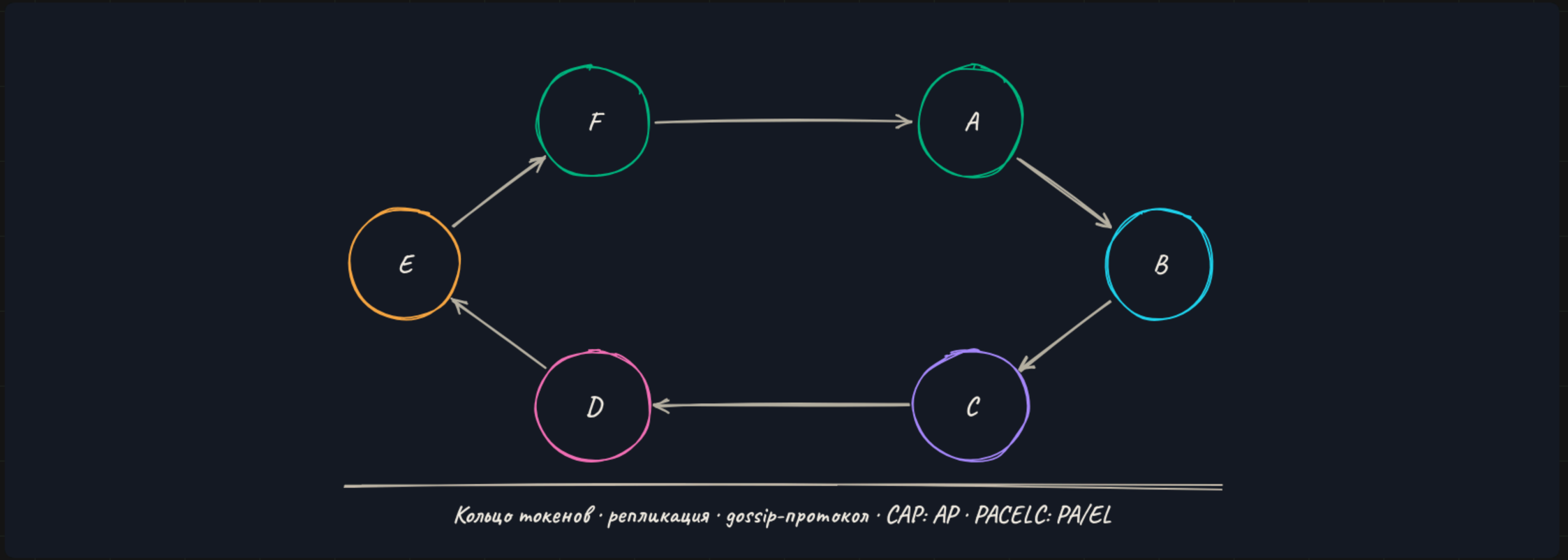
В конечном счёте	Причинная	Линеаризуемая	Строго сериализуемая
Устаревание допустимо	Причина раньше следствия	Единая истина чтения/записи	Транзакции + время

Общая схема: CAP + PACELC + Jepsen



Реальная БД: Cassandra через призму CAP/PACELC

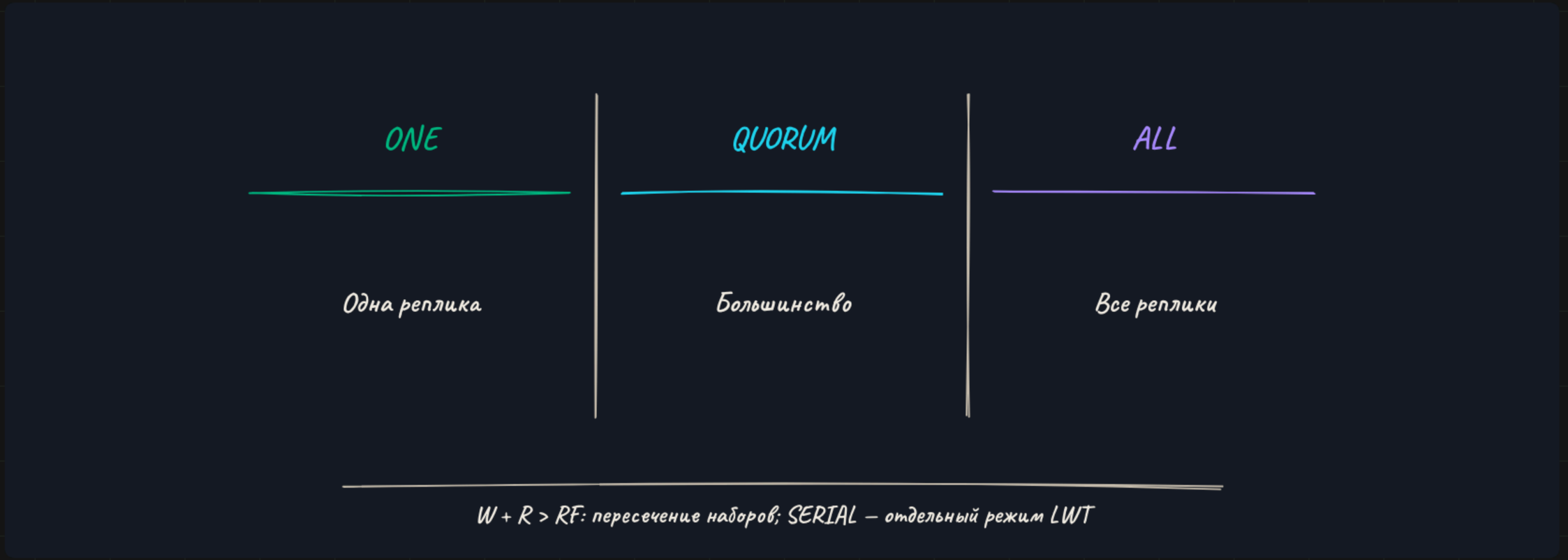
Бонус-кейс Apache Cassandra



Бонус-кейс · Cassandra

Консистентность

Cassandra: настраиваемая консистентность



Бонус-кейс · Cassandra

Архитектура

Как Cassandra держит масштаб и доступность

Путь записи и хранения

Commit Log → MemTable → SSTable

Уплотнение чистит версии

LSM оптимизирован под запись

Распределение и устойчивость

Согласованное хеширование + vnodes

Gossip-протокол без мастера

Восстановление реплик для сходимости

Ссылки и материалы

- CAP: ↗system-design.space
- PACELC: ↗system-design.space
- Модели консистентности (Jepsen): ↗system-design.space
- Cassandra: ↗system-design.space

- Все 4 исходные главы ↗system-design.space в одном месте.