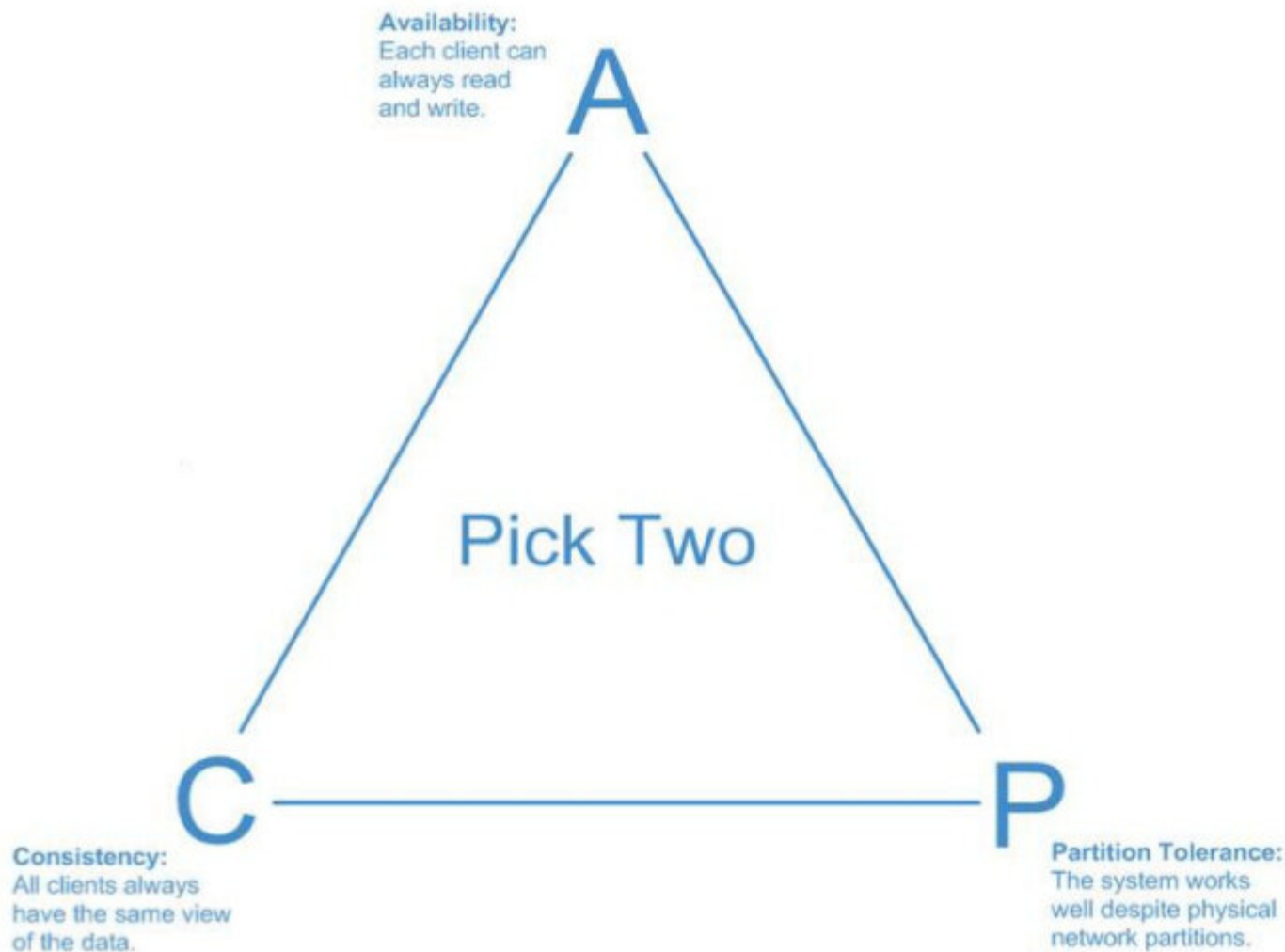


Начнём с основных понятий, прежде чем выбирать базу данных. Теорема CAP.

Теорема CAP - это фундаментальная концепция распределенных систем, которая объясняет компромисс между тремя ключевыми свойствами: согласованностью, доступностью и устойчивостью к разделам. Понимание теоремы CAP поможет разработчикам и архитекторам принимать обоснованные решения при проектировании и выборе технологий для своих распределенных систем.

Теорема CAP, сформулированная Эриком Брюэром в 2000 году, утверждает, что невозможно, чтобы распределенная система одновременно обеспечивала все три гарантии:

- 1. **Согласованность**(также называют **консистентность**) **(C)**: все клиенты видят одни и те же данные в одно и то же время.
- 2. **Доступность** **(A)**: каждый запрос получает ответ, при этом не гарантируется, что он содержит самые последние данные (согласованные).
- 3. **Устойчивость к разделениям** **(P)**: система продолжает работать, несмотря на разделение сети, которое может произойти, когда узлы (компоненты распределённой системы) теряют связь друг с другом из-за сбоев в сети (например два сервера потеряли связь между друг другом)



В любой распределенной системе разработчики должны сделать выбор между этими тремя гарантиями, поскольку одновременно могут быть удовлетворены только две. Это означает, что при проектировании распределенных систем архитекторы должны определять приоритеты свойств, которые наиболее важны для конкретного случая использования, и выбирать технологии и конструкции, соответствующие этим приоритетам.

Понимая теорему CAP и компромиссы, связанные с ней, разработчики могут принимать более обоснованные решения при проектировании своих распределенных систем, обеспечивая необходимый уровень согласованности, доступности и устойчивости к разбиению для конкретных требований приложения.

Давайте рассмотрим на примере:

Согласованность (C): каждая операция чтения возвращает последнюю запись или ошибку. Это означает, что все узлы в системе имеют одинаковые данные в любой момент времени. Это гарантирует, что пользователи всегда получают самую актуальную информацию. Пример: баланс банковского счета, который как сущность содержится на нескольких серверах(чтобы при падении одного из них отдать баланс). Если один сервер обновляет баланс, все остальные серверы должны немедленно обновить свой баланс, чтобы обеспечить согласованность. Это может привести к задержке обновлений или снижению доступности в случае разрыва сети, но при этом так поддерживается согласованность данных.

Доступность (A): каждый запрос (на чтение или запись) получает ответ без ошибок, даже если это несвежие (несогласованные) данные. Это означает, что система остается работоспособной и отзывчивой, даже если некоторые узлы не работают или отключены. Пример: платформа социальных сетей, которая позволяет пользователям публиковать сообщения, даже если некоторые серверы не работают. Пользователи по-прежнему могут видеть свои сообщения (хотя они могут не видеть самые последние данные), и система остается доступной даже в неблагоприятных условиях. Однако при этом жертвуется согласованность.

Устойчивость к разделениям (P): система продолжает функционировать даже при нарушении связи между ее компонентами. Это означает, что если сеть между узлами становится разделенной, система может продолжать работать. Пример: распределенная база данных, которая может справляться со сбоями связи между серверами путем репликации и согласования данных после восстановления соединения. Для этой системы приоритетом является устойчивость к разделениям, и она может сосредоточиться либо на согласованности, либо на доступности, но не на том и другом.