Skillbox

Работа с событиями. Особенности CQRS и ES

Андрей Гордиенков

Solution Architect

ABAX

В прошлом уроке

- Что такое CQRS, какую проблему решает?
- Связь с Event Sourcing

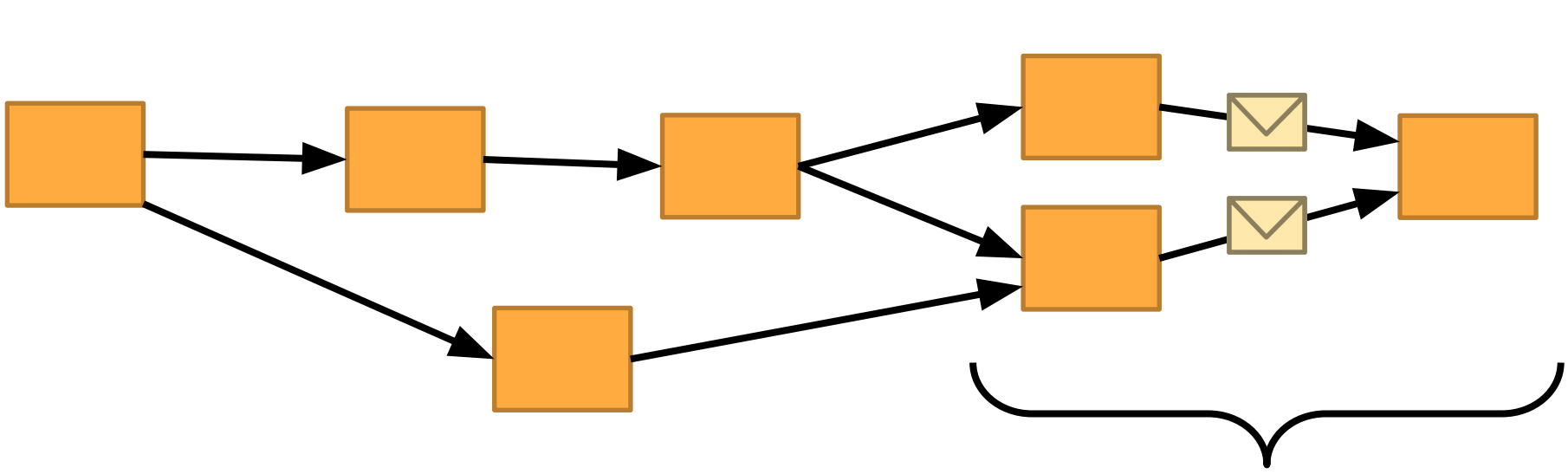
В этом уроке

- Применимость в разных доменах
- Нюансы реализации
- Возможные заблуждения о работе Event-Driven-систем

Трейсинг

Событие может быть сгенерировано разными сервисами, которые по-разному написаны, и каждый из них может быть с ошибками.

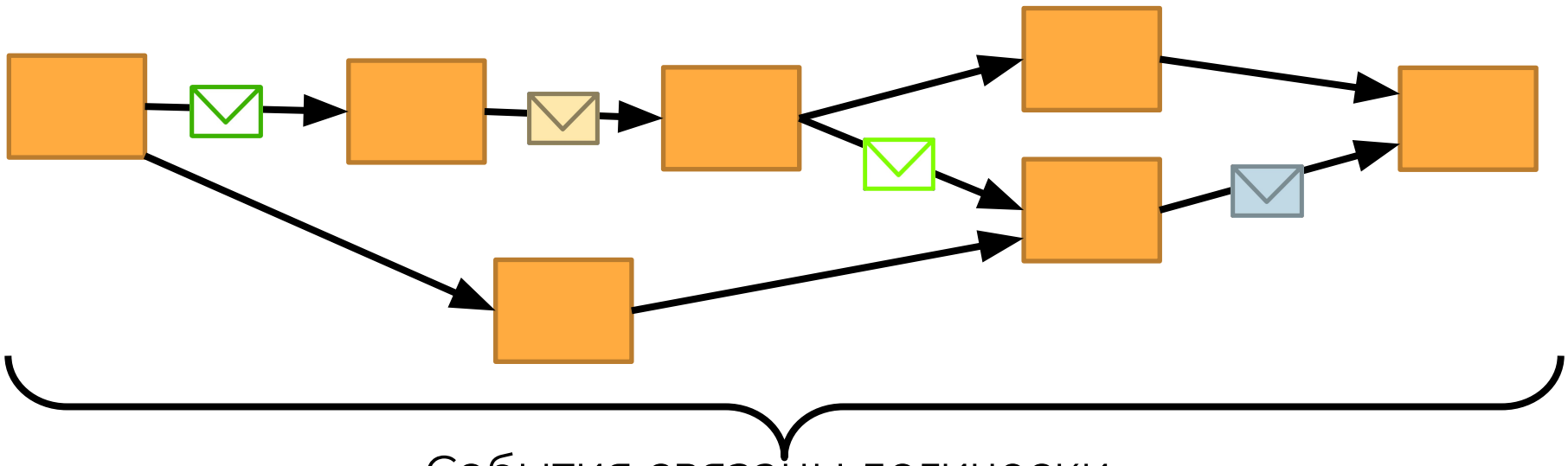
Необходимо использовать ID инициатора.



Skillbox

Корреляция событий

В слабосвязанной системе с преимущественно асинхронным взаимодействием тяжело находить полный путь выполнения запроса без дополнительных усилий.



События связаны логически.
Это один бизнес-процесс.

Необходимо использовать в ответе корреляционный ID, чтобы можно было собрать воедино все связанные события, относящиеся к бизнес-процессу.

Идентификаторы

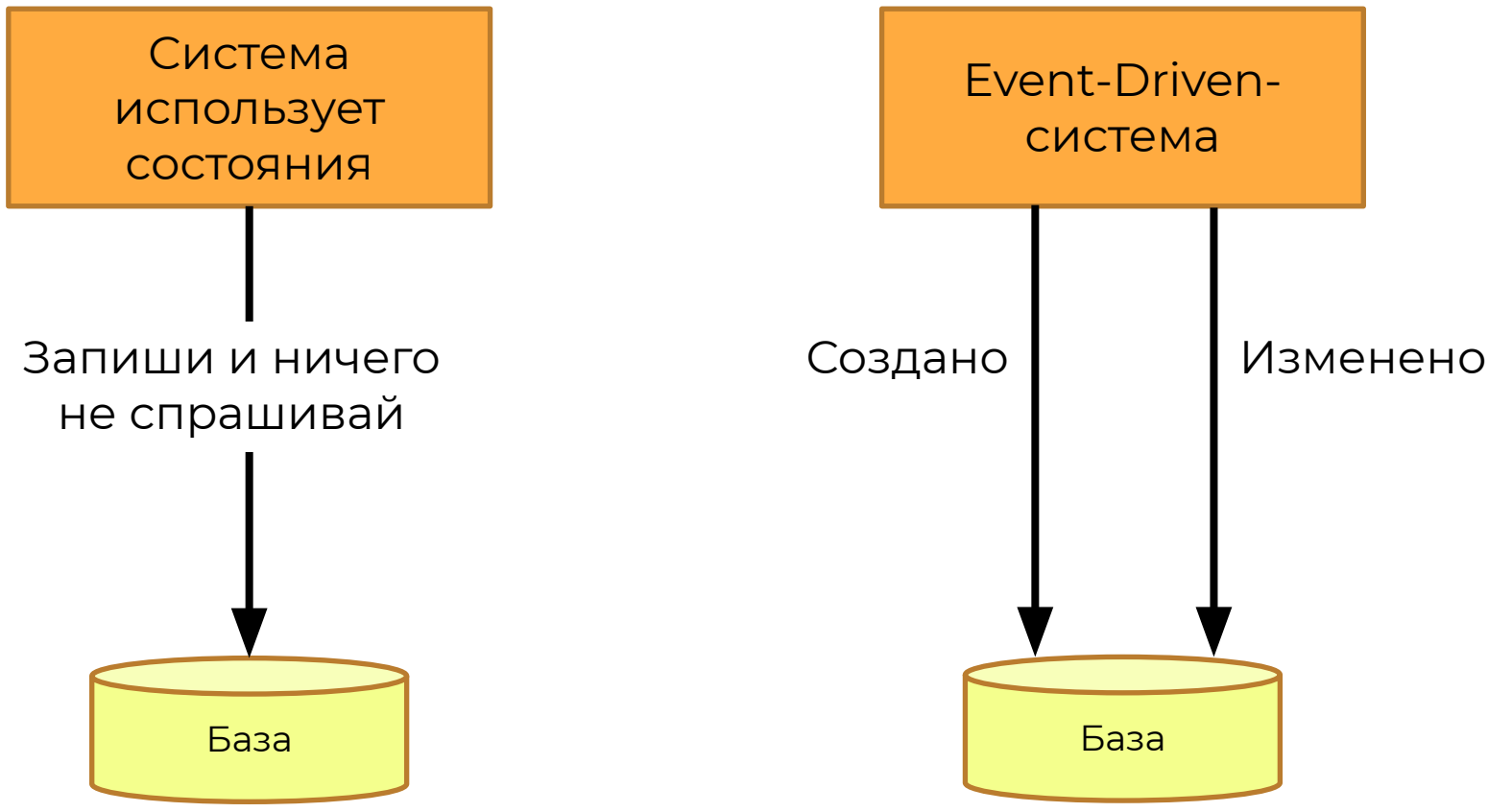
В идеале хорошо иметь три идентификатора:

- уникальный номер сообщения
- указание на инициатора запроса
- уникальный номер процесса обработки запроса

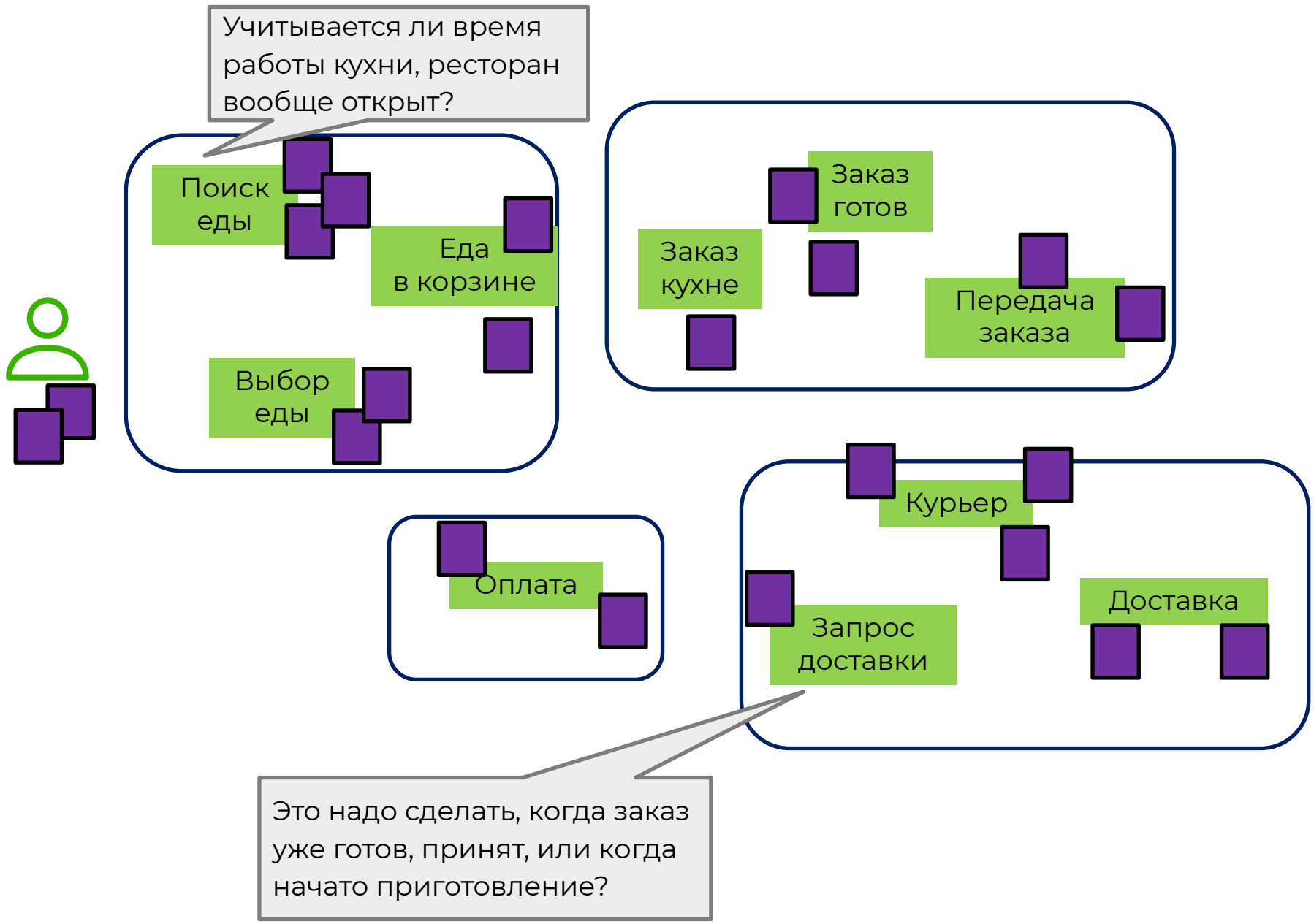
Расследование багов тогда не будет столь мучительным. Часть идентификаторов может быть сгенерирована инфраструктурными элементами, например, OpenTracing.

Коррекции

Исправление ошибок, то есть обновление данных
в Event-Driven-системах, требует дополнительных усилий —
нужны отдельные команда, событие и обработчики.



Фокус на времени



Фокус на времени

Событийная архитектура заставляет нас явно задуматься о порядке сообщений и их причинной связи.

Причинная связь рассматривается только в узких рамках конкретного бизнес-сценария (Quality Attribute Scenarios).

Skillbox

Фреймворки

Требования для CQRS + ES фреймворка

- Functions
- Pattern Matching
- Left fold

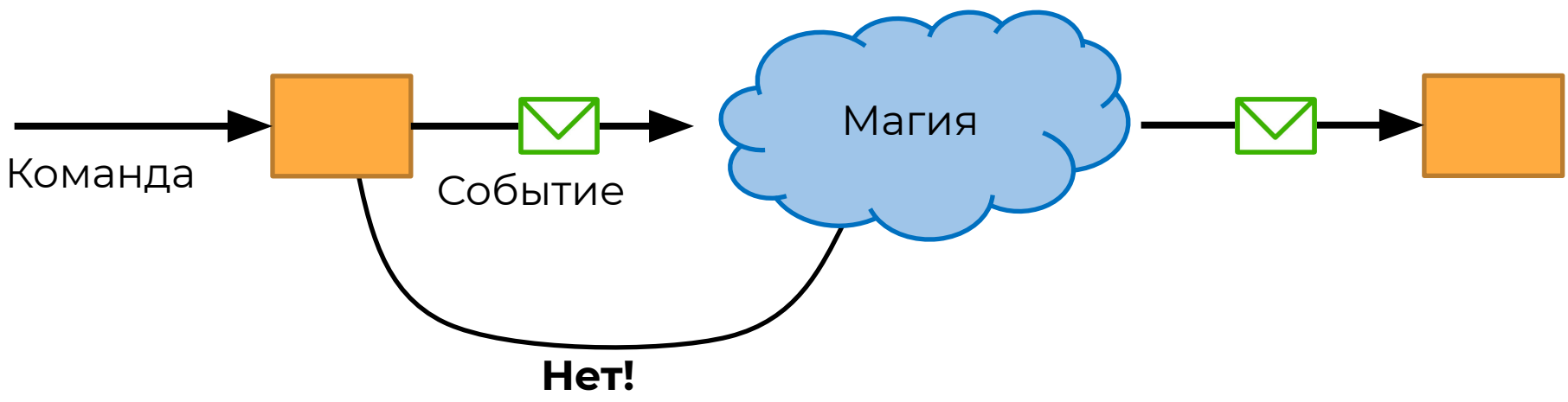
Это уже есть в языках программирования, особенно функциональных.

Не пишите свои и не используйте чужие фреймворки.

Skillbox

Fire & Forget

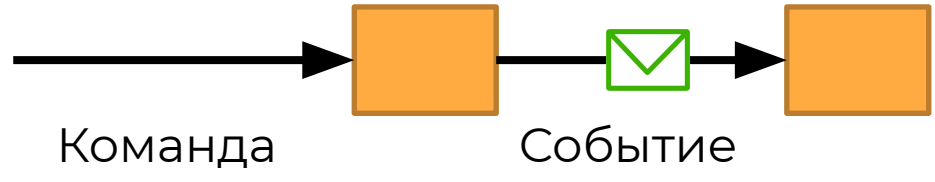
Как работает: создаём команду, пускаем куда-то в систему и не ожидаем ответа.



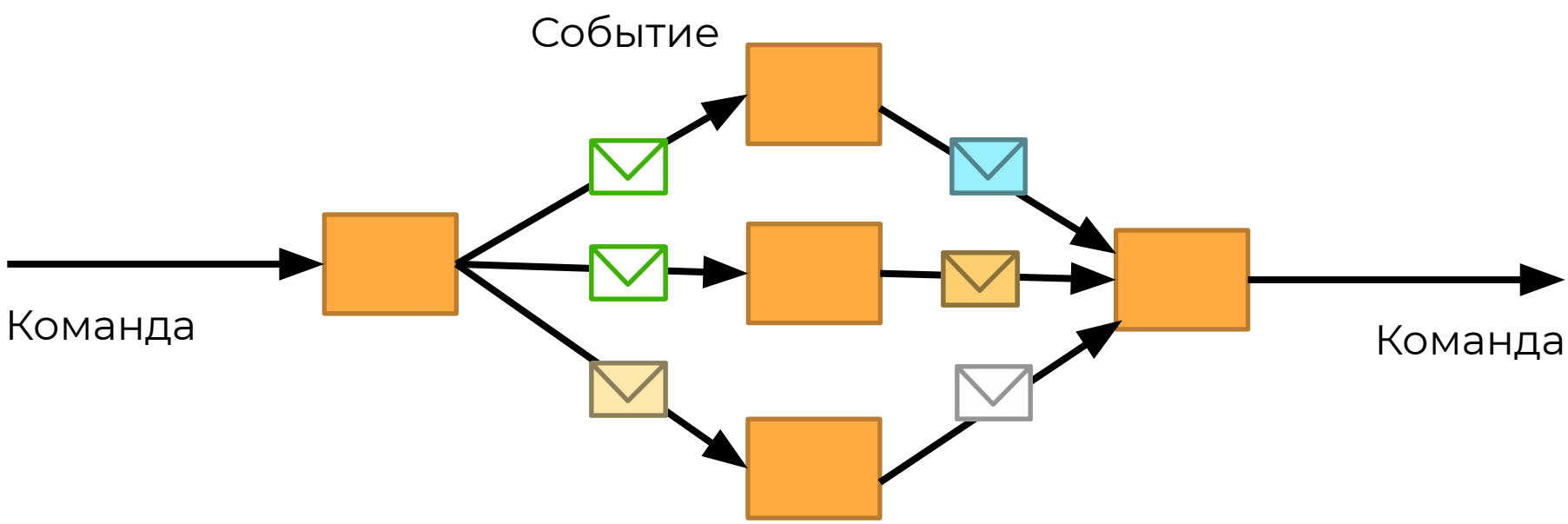
В бизнес-процессах такого не бывает. Сервис может сказать «нет»:

- явно
- с помощью исключения

1 команда = 1 событие



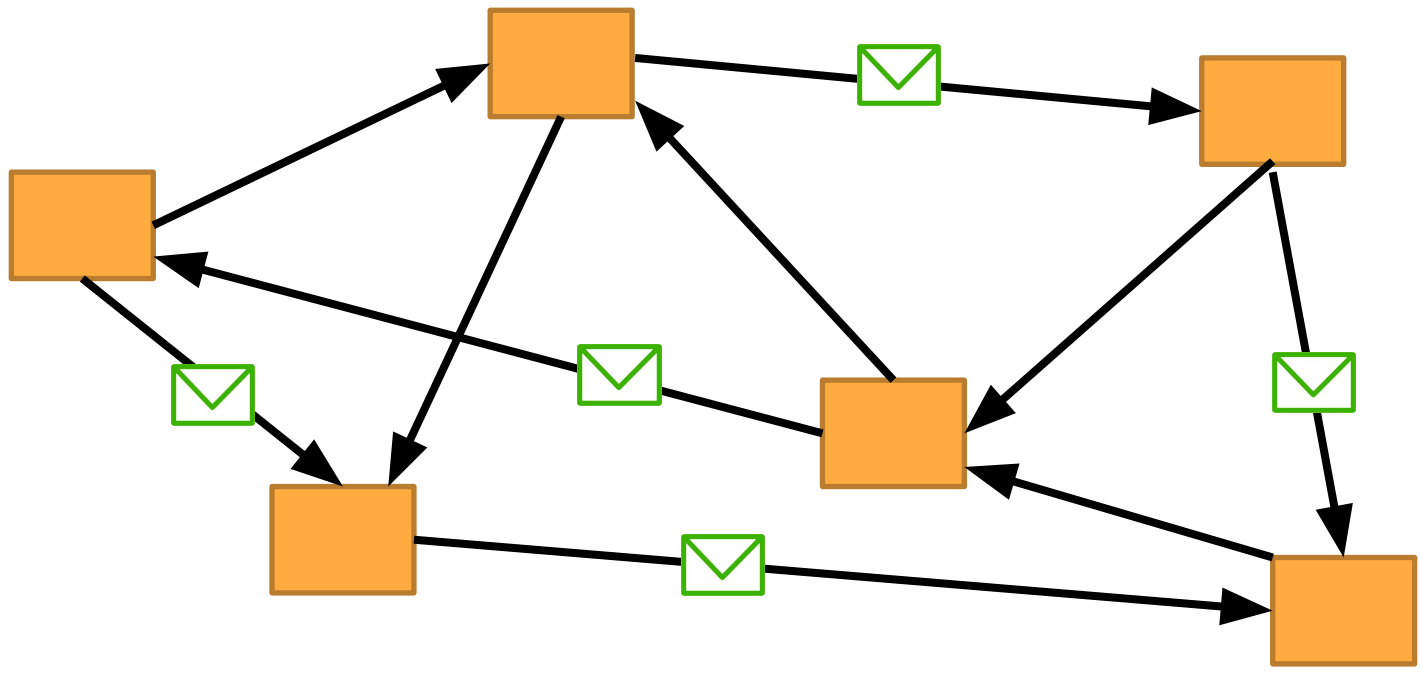
- Одна команда может породить много событий
- События могут быть разных типов



Skillbox

Process Manager

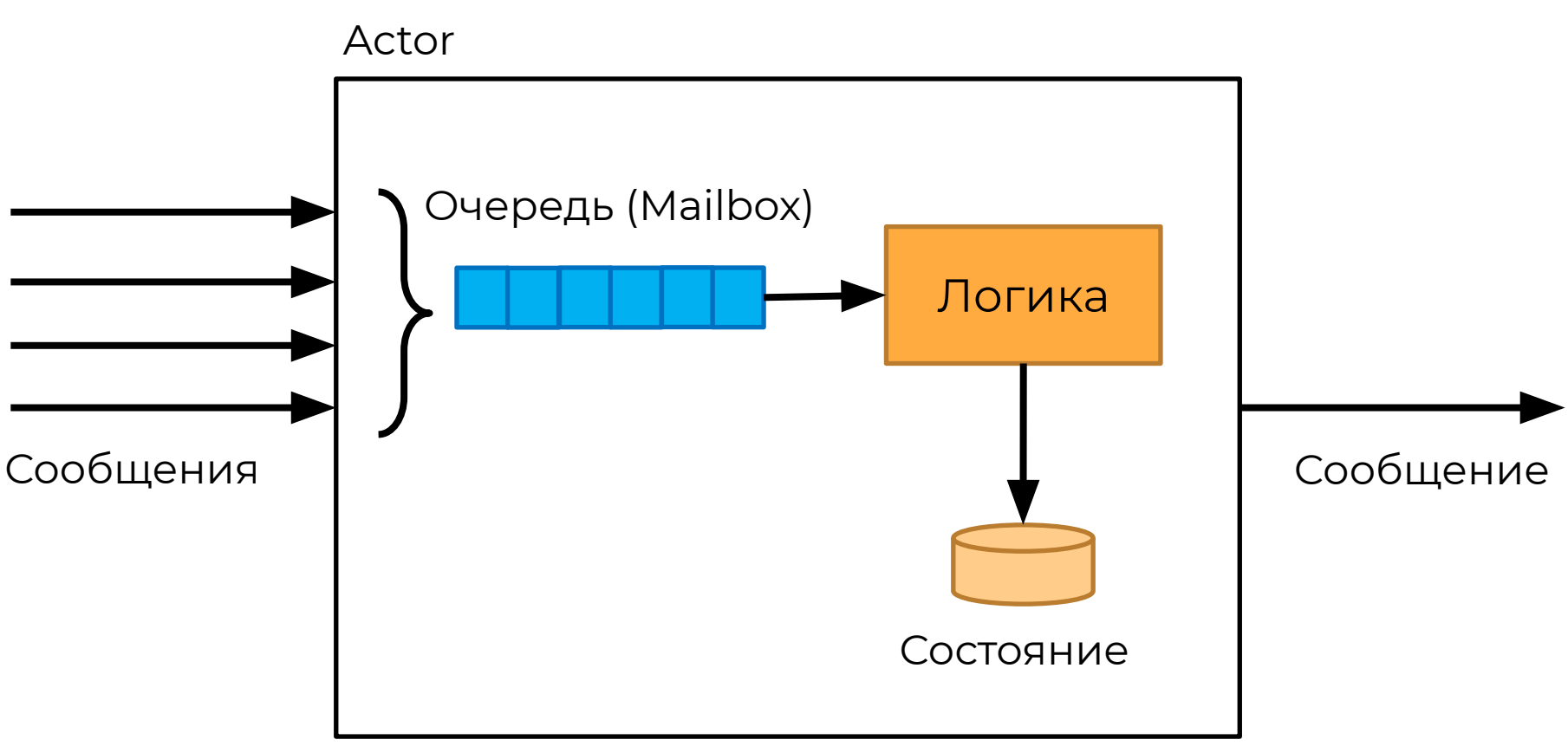
При большом количестве типов событий и больших сервисах воссоздать в голове логику работы сложно.



Skillbox

Actor model

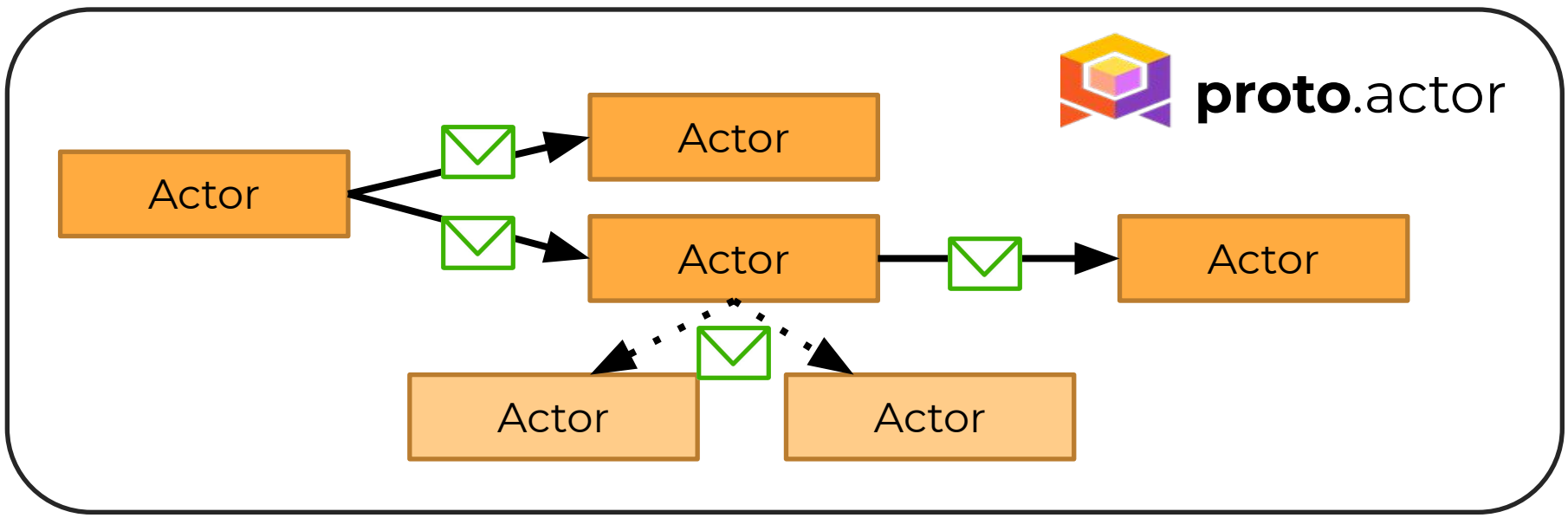
Базовый элемент для многопоточной (concurrent) обработки информации. При этом сам Actor работает строго в однопоточном режиме для изменения своего состояния.



Actor model

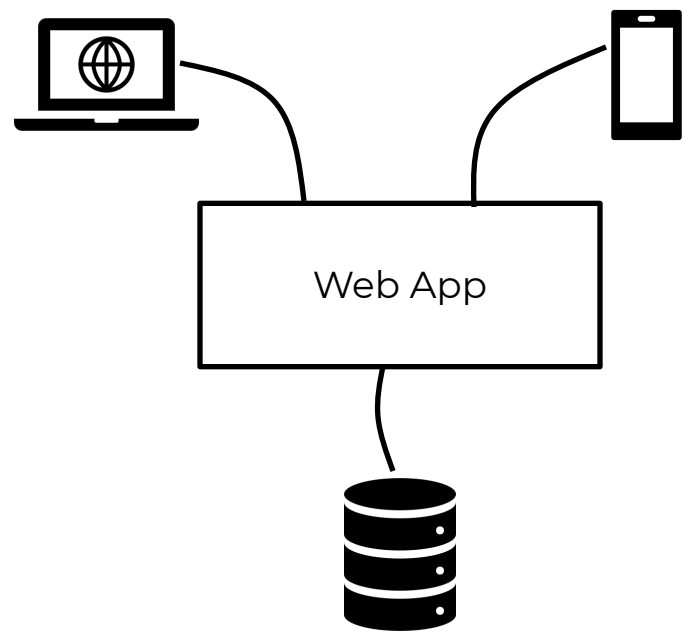
- Работает с одним агрегатом
- Легко масштабируется
- Устойчив к сбоям

- Внутренняя очередь может перегрузиться
- Возможны deadlock’и

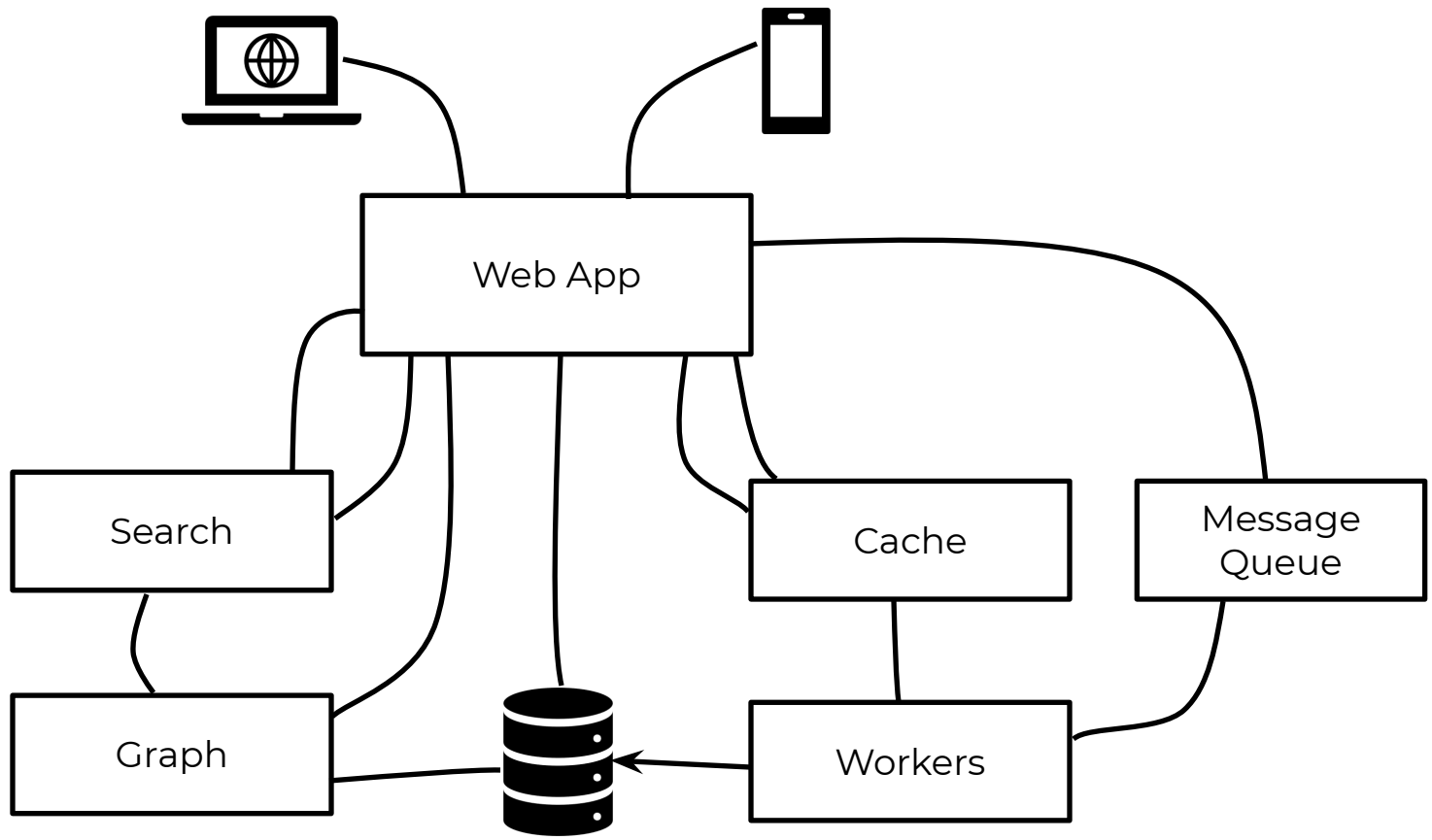


Skillbox

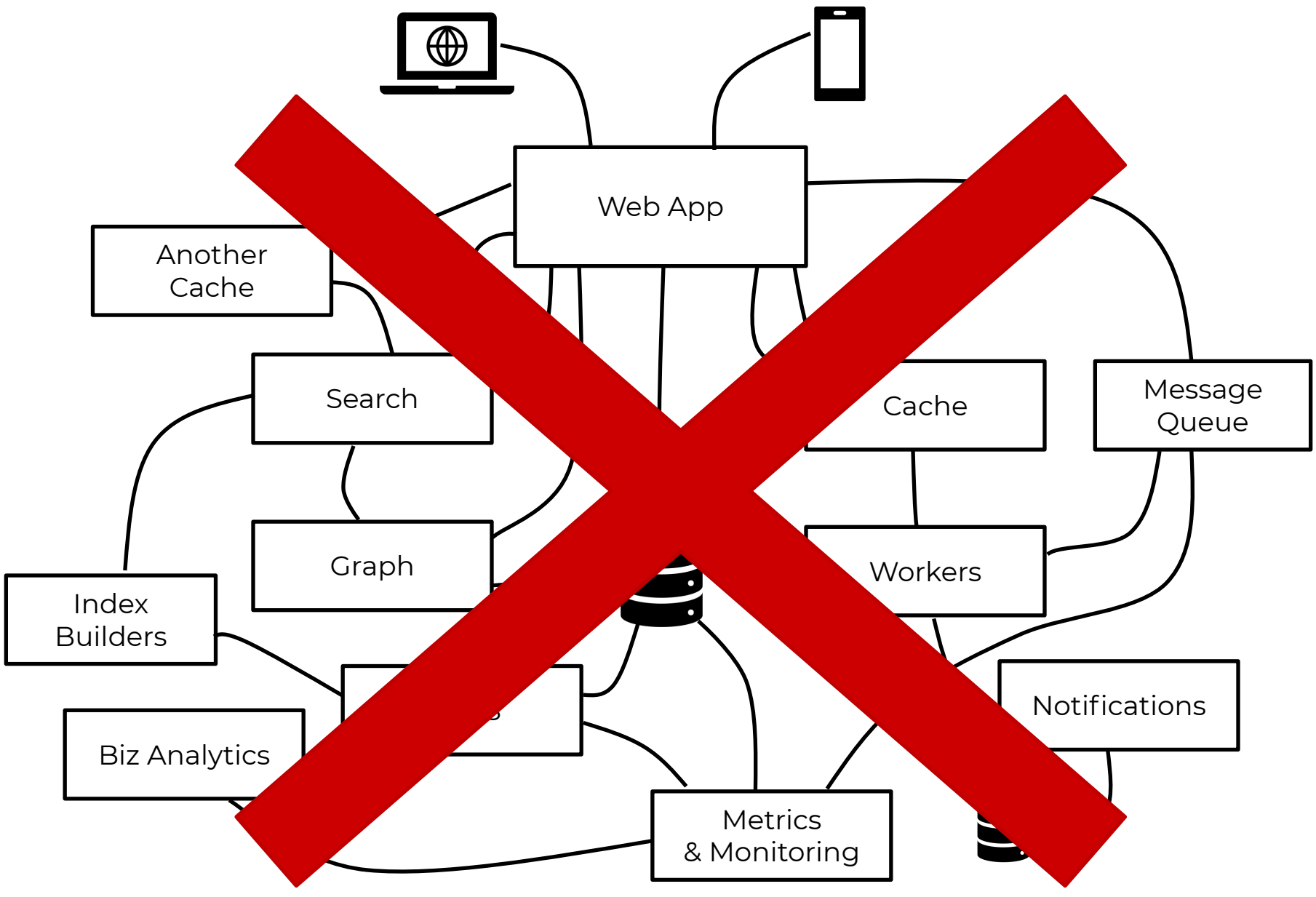
Логи и сервисы



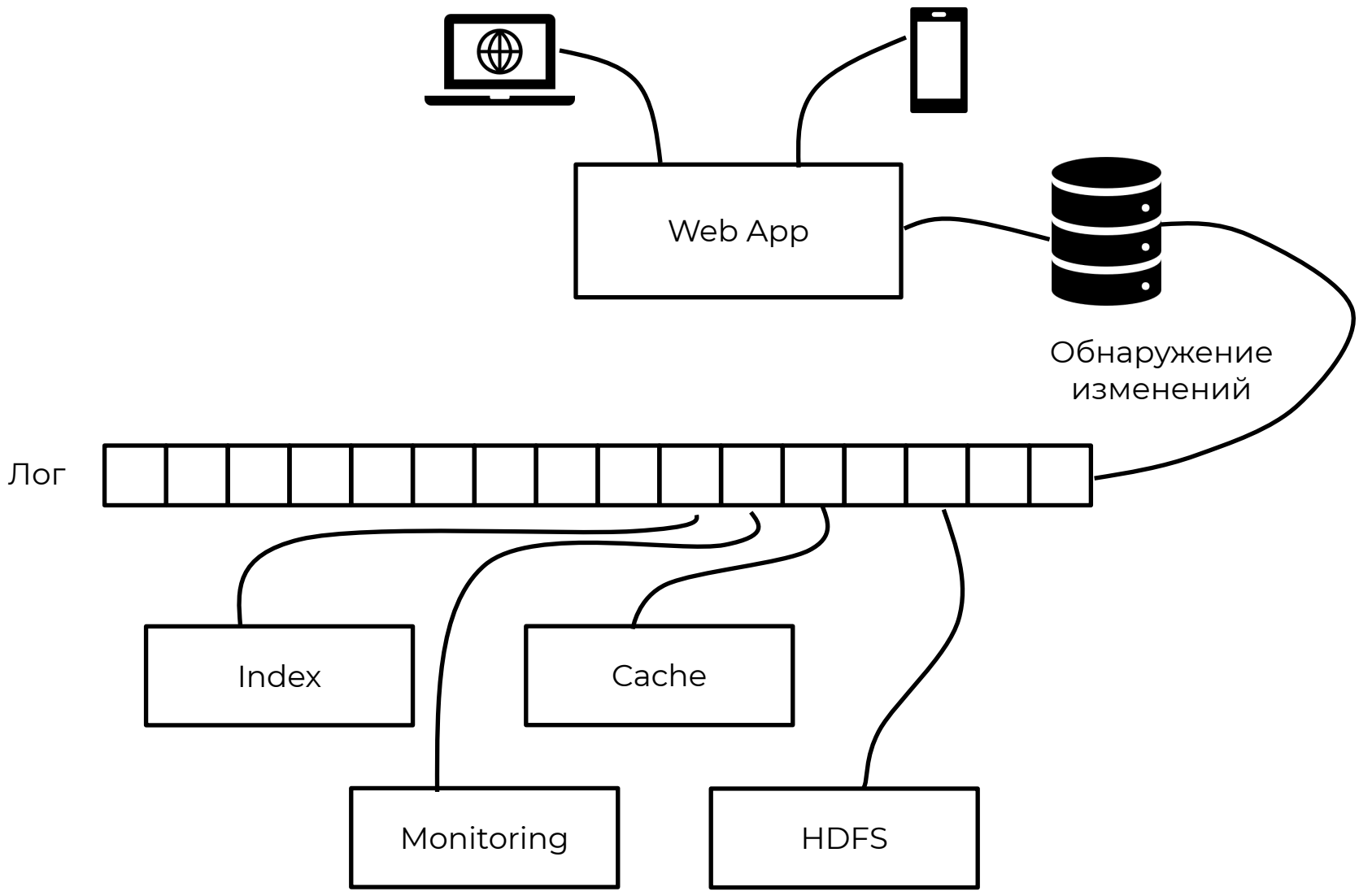
Логи и сервисы



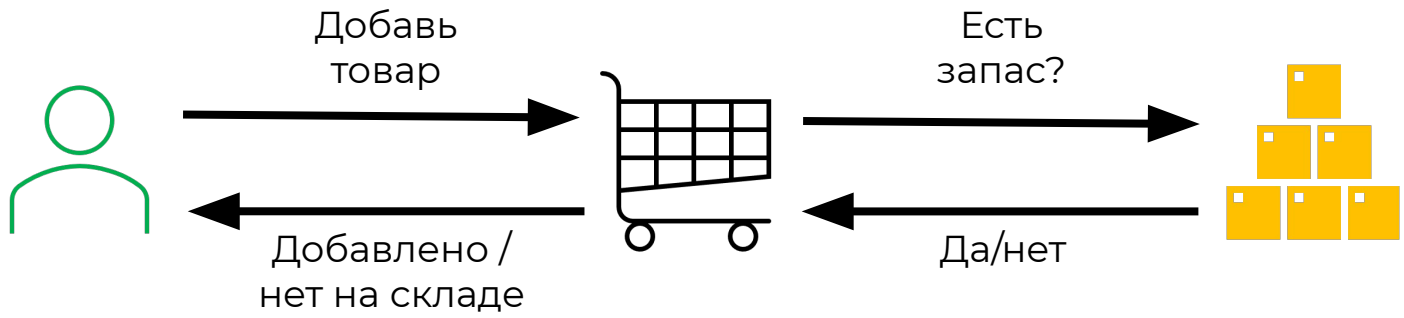
Логи и сервисы



Логи и сервисы



Согласованность данных



Так ли это важно?

Посмотрите на это с точки зрения бизнеса: какой срок нужен на доставку/производство? Каковы затраты?

$$$$$$

Маржа?

$

Согласованность данных

Событийная модель отлично подходит для «временами соединённых систем».

Availability

Consistency

Согласованность данных

Событийная модель отлично подходит для «временами соединённых систем».

История изменений естественным образом обработается на сервере при восстановлении соединения.

Характеристики качества:

- доступность
- слабая согласованность

Availability

Consistency

Выводы

- Записывайте причинно-следственную связь событий в самих событиях
- Моделирование событий позволяет глубже погрузиться в доменную область
- Шаблон «Актор» может упростить построение Event-Driven-системы
- Использование быстрого центрального хранилища логов позволит снизить когнитивную нагрузку системы
- Использование событийного подхода при требовании к строгой согласованности данных не сработает

Итоги модуля

- Команды и события — естественные аналогии из жизни, которые позволяют точнее представить домен в коде
- Использование событийной модели позволяет получить ответ на вопрос: «Почему что-то изменилось в наших данных?»
- Событийная модель позволяет создавать очень гибкие и быстрые системы. Позволяет легко строить аналитические системы
- Создание сопряжено с рядом технических нюансов, и поэтому строить абсолютно всё на событиях может быть невыгодно, поскольку это сложно

Skillbox

Спасибо за внимание!