Event Sourcing

Предпосылки

В БД есть лог транзакций

Если взять лог транзакций и "проиграть" его от начала до конца, то получится текущее состояние БД

Мы берем концепцию лога транзакций и реализуем её в коде в явном виде

Теперь каждое изменение состояние системы не записывается в БД напрямую, а сохраняется в виде Event'а

Откуда взять данные?

Как делать запросы для выборки данных, если мы не храним сами данные?

Мы создаем специальные проекции, основанные на логе Event'ов

Аналог проекций в БД – это View

Разница в том, что View основаны на данных в БД (состоянии), а проекции создаются и обновляются на основе списка Event'ов

Зачем так усложнять?

Примеры бизнес-задач, решаемых Event Sourcing-ом:

- Каким было состояние системы 2 недели назад на момент события X?

- Пользователям надо отменять любые действия в системе

- Имеете ли вы право затереть данные в ячейке новыми? На сколько важны старые данные? Можем ли мы позволить себе потерять старые значения?

- Сами события переходов между состояниями являются важной частью аналитики

Основы

Все изменения, которые попадают в систему, мы записываем в виде дельты - Event

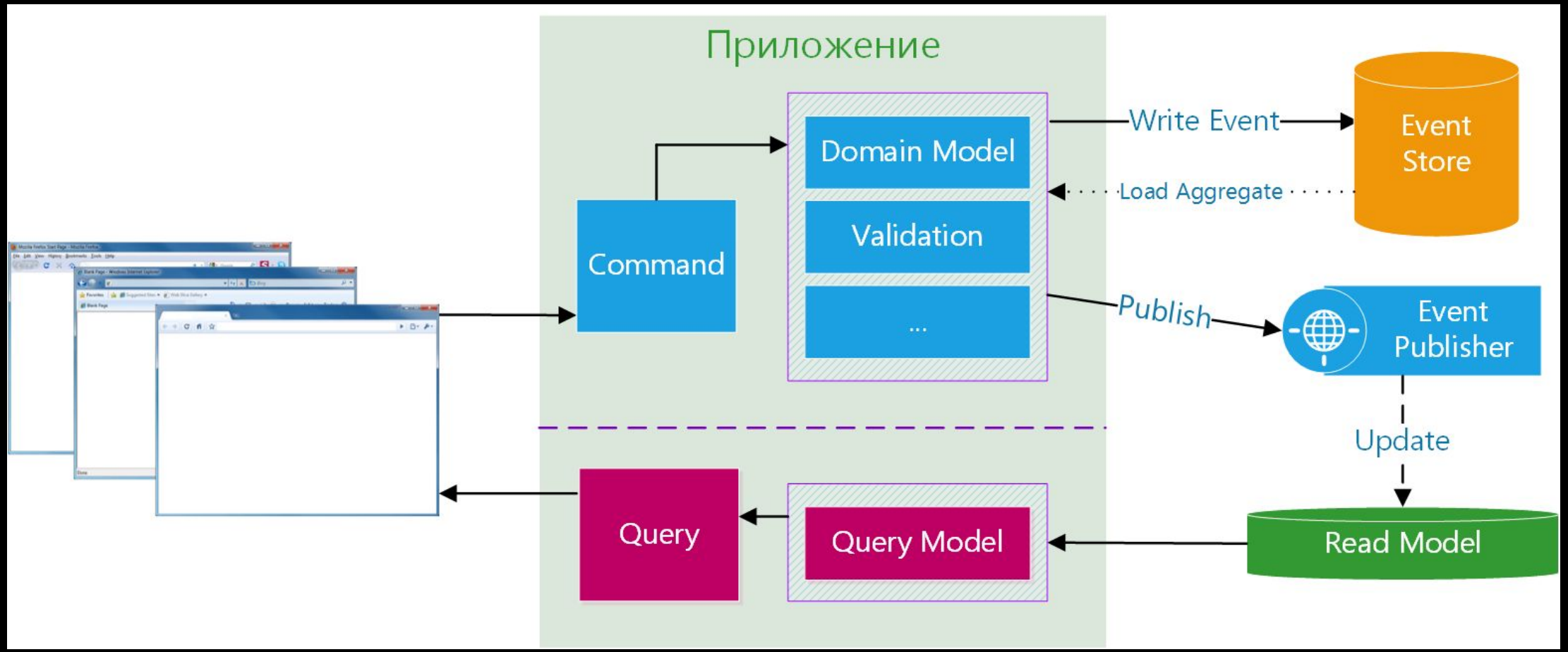
Событие изменения состояния системы должно знать к какому агрегату оно относится, версию и данные об изменении

Текущее состояние домена – это "проигрывание" журнала Event'ов

Выборки делаются на проекциях, сами проекции это "проигранные" Event'ы

Для экономии ресурсов состояние домена не "проигрывается" каждый раз с нуля - мы можем зафиксировать состояние домена на определенную дату

Дизайн проекта



Нужно ли это?

Как проектировать агрегаты?

Как рефакторить агрегаты? Что делать, если корень агрегата был выбран неверно, а события для него уже есть в Event Store?

Как изменять уже произошедшие события?

Как накатывать события, которые зависели от данных стороннего сервиса?