



# Быстрая разработка GUI для больших объёмов данных с использованием CQRS парадигмы

Алексей Рагозин

*Passion to Perform*



**2014**

**CEE-SEC(R)**

**Разработка ПО**



# Наш челенж

Deutsche Bank – один из крупнейших инвестиционных банков.

- Тысячи клиентов
- Десятки бизнес линий
- Миллионы транзакций каждый

Условия рынка меняются

- Новые законы
- Новые рынки
- Новые продукты

# Наш челенж



Трейдинговый UI это

**Сотни тысяч бизнес объектов** на экране

*возможно миллионы*

которые меняются в **реальном времени**

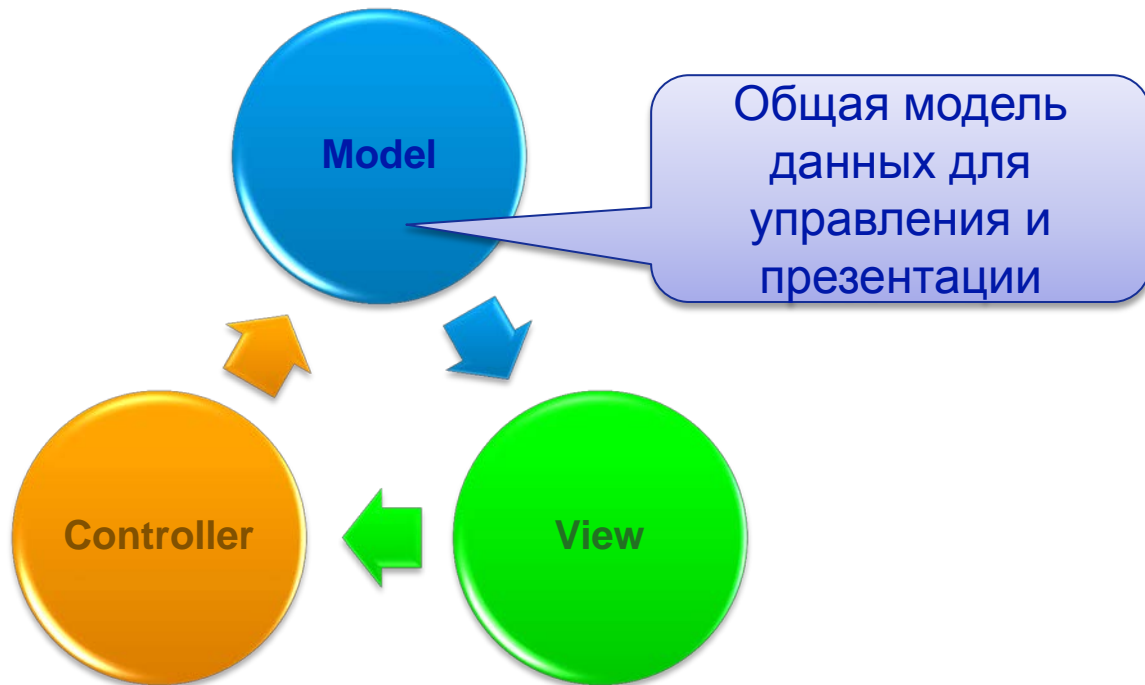
**аннотированные** рыночными котировками

*тикающими каждую секунду*

при этом UI должен быть **быстрым**

*с низким потреблением ресурсов*

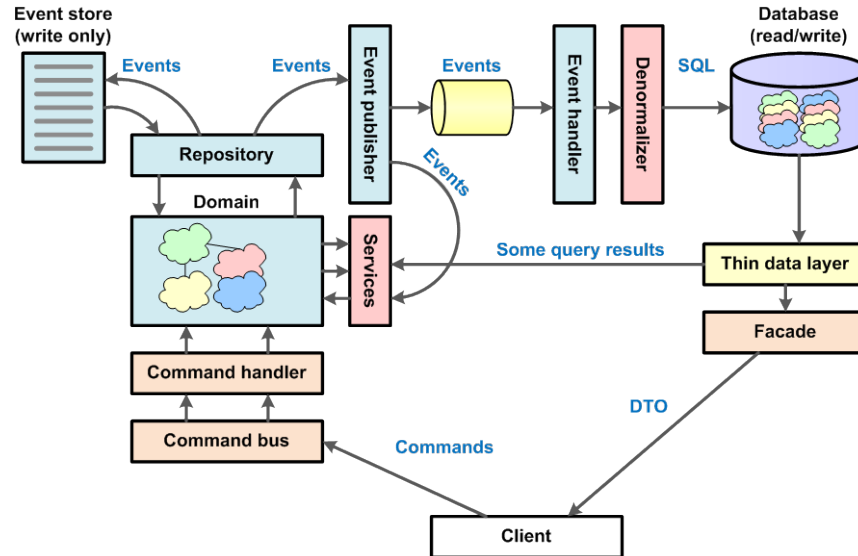
# Парадигма CQRS



# Парадигма CQRS



## Command Query Responsibility Segregation

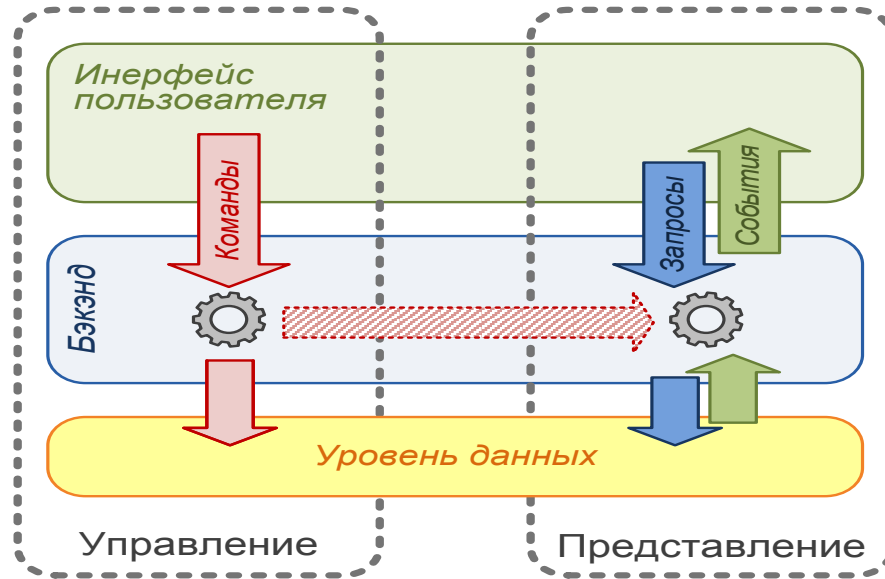


\* Greg Young, Mark Nijhof, <http://cre8ivethought.com/blog/2009/11/12/cqrs--la-greg-young>



# Парадигма CQRS

## Command Query Responsibility Segregation



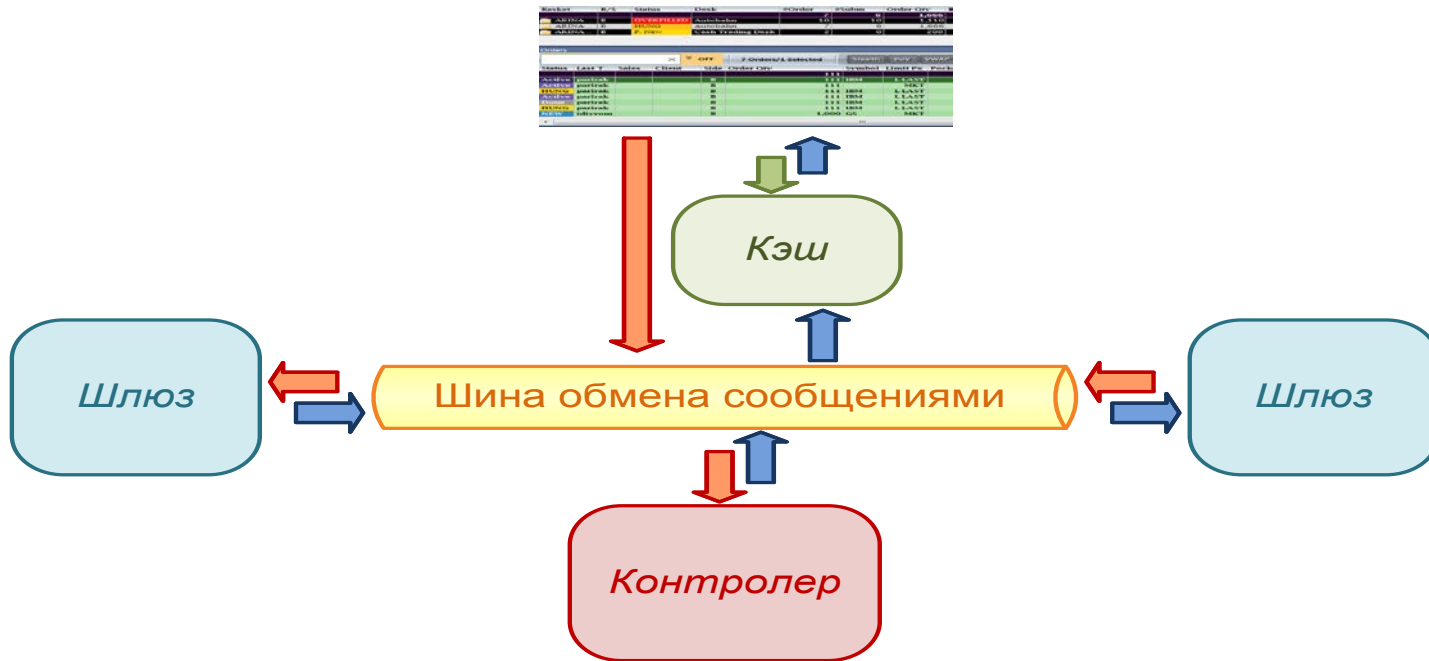


# Парадигма CQRS

## Command Query Responsibility Segregation

- Независимые модели данных
  - Преставления - **События**
  - Управления - **Команды**
- Независимые цепочки компонентов
- Естественная модель для асинхронного выполнения команд

# Традиционная архитектура OMS







# Улучшаем CQRS

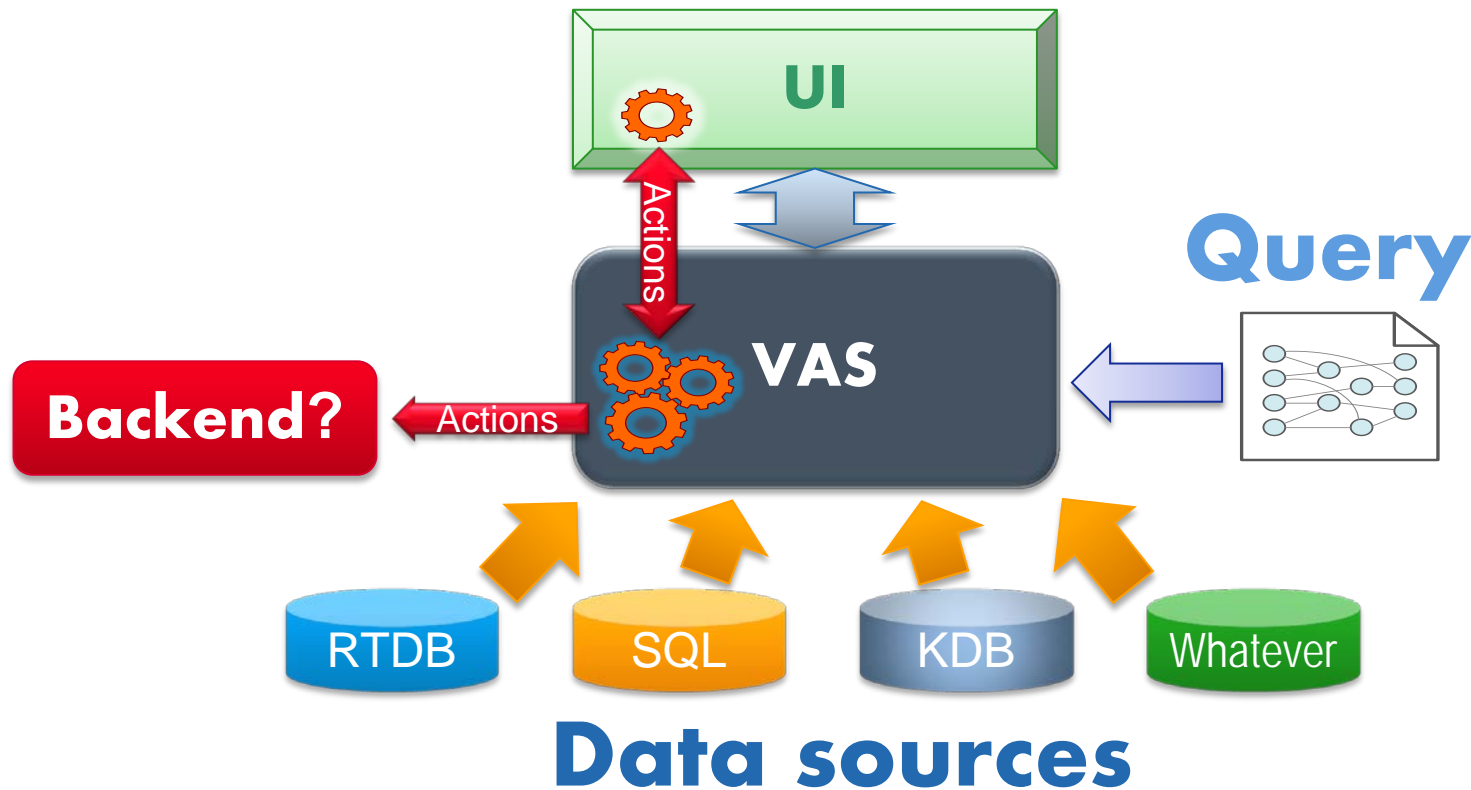
## Данные меняются динамически

- Pull модель не работает
- Используем Push модель
- Continuous Query

## Модель представления

- Модель визуализации – таблицы
- Модель хранения – таблицы (значительная часть)
- Используем реляционную модель для представления

# View Action Server





# View Action Server

- Платформа Java (C# на стороне UI)
  - Запросы в стиле LINQ
- Непрерывная обработка данных
  - Continuous queries
- Реляционная модель и реляционная алгебра
- Унификация источников данных
  - Например, универсальный RDBMS адаптер

# View Action Server



## Виртуализованные таблицы

- Видимы диапазон передаётся на UI
  - Сортировка и фильтрация на стороне VAS
  - Selection model на стороне VAS
  - Master – details линковка таблиц на сервере
- ✓ Потребление памяти GUI не зависит от объёма данных
- ✓ Возможна on-demand подгрузка данных только для видимых строк
- Например: подписки на рыночные котировки только для видимых строк

# “Виртуализация” данных



Агрегаты

Видимый диапазон

Status	Side	Order Qty	Symbol	RIC	CUSIP	Arrival Time	% ADV20d	Lmt Px	Limit Px
		5,687							
	B	100,000	AA	AA.N	013817101	14:16:52	2.51%	0.0000	L LAST
	B	100	GE	GE.N	369604103	20:55:05	0.00%	0.0000	MKT
	B	100	GE	GE.N	369604103	20:55:45	0.00%	0.0000	MKT
	B	500	GE	GE.N	369604103	18:51:02	0.01%	0.0000	MKT
	B	1,500	IBM	IBM.N	459200101	18:59:22	0.15%	0.0000	MKT
	S	1,500	IBM	IBM.N	459200101	18:59:38	0.15%	0.0000	MKT
	B	1,800	IBM	IBM.N	459200101	18:59:50	0.18%	0.0000	MKT
	SS	187	IBM	IBM.N	459200101	19:00:52	0.02%	0.0000	MKT
	B	100	GE	GE.N	369604103	20:59:12	0.00%	0.0000	MKT
	B	125	GE	GE.N	369604103	16:03:25	0.00%	0.0000	MKT

Arrival Time	Sent Time	Status	Side	Order Qty	Symbol	RIC	CUSIP	Filled Qty	% Fill
18:53:13	18:53:13		B	200	GE	GE.N	369604103	200	100.00%
18:52:01	18:52:01		B	200	GE	GE.N	369604103	200	100.00%

Выделен ие

Связанная таблица



# Преимущества подхода

Повторное использование источников данных

- Нет доменных объектов, нет зависимостей на код

Минимальное количество кода на бэк-энде

- Java для реализации команд
- Реляционная алгебра для представления

Быстрый старт разработки

- Простая конфигурация тестового окружения
- Повторное использование компонентов

NFR обеспечивает платформа

- Централизация экспертизы в одной команде
- Эволюция платформы → улучшения для всех бизнес приложений



Спасибо!

Алексей Рагозин

*Passion to Perform*