

Проектируем микросервисное приложение на схемах. Разбор кейсов



Максим Цепков

Архитектор и аналитик,
навигатор по миру Agile
и бирюзовых организаций

<http://mtsepkov.org>

Анна Абрамова
Аналитик в DataArt
Лидер сообщества
аналитиков Петербурга



19-20 июня 2021 Летний Аналитический Фестиваль

Как будет устроен мастер-класс?

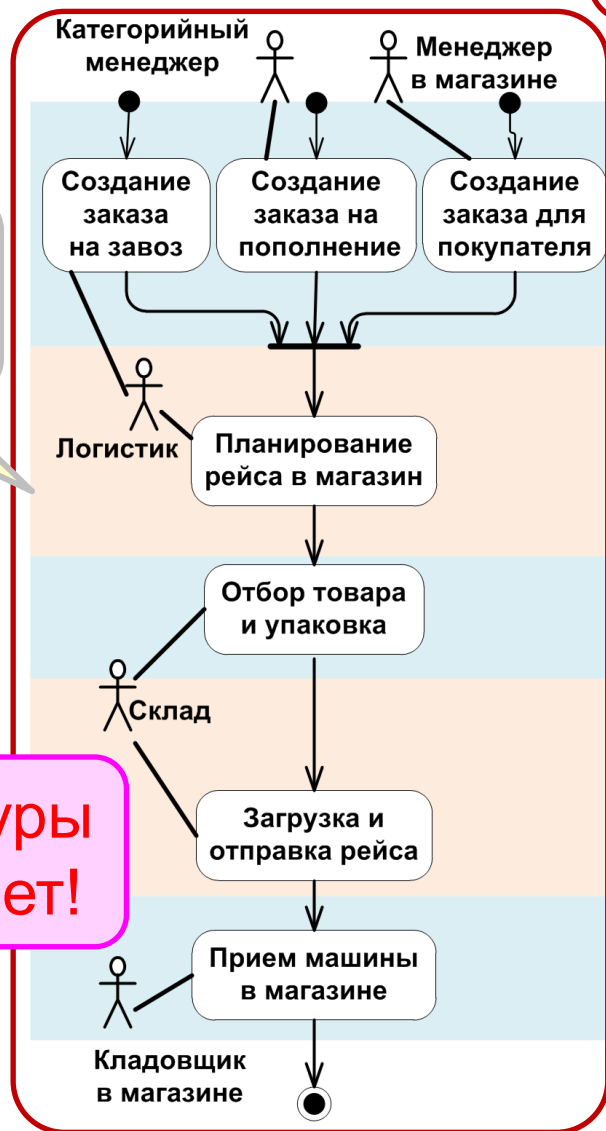
- Современная архитектура приложения включает микросервисы, messaging, акторные модели, распределенное хранение
- Для таких приложений недостаточно спроектировать структуры данных и правила их изменения в ответ на действия на интерфейсы
- На прошлой AnalystDays я рассказывал про модель для этого [Модели предметной области для разных парадигм программирования](#), сейчас я кратко повторю основную идею – для тех, кто не был
- Затем мы соберем кейсы от участников и рассмотрим несколько интересных – посмотрим модель в работе

Что поменялось в архитектуре

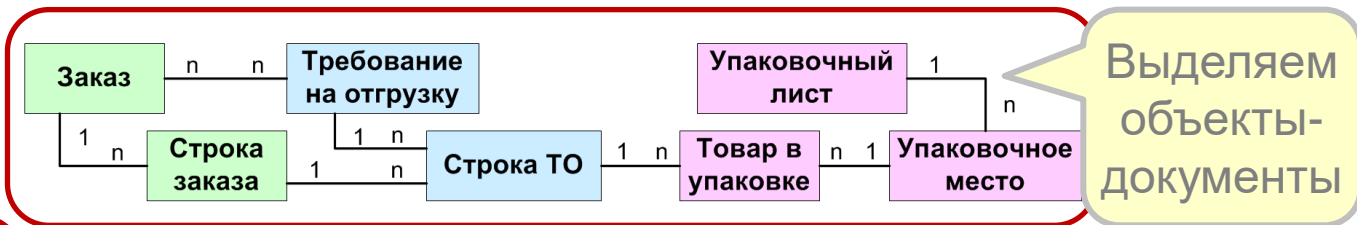
- Сервисные и микросервисные архитектуры, каждый бизнес-запрос обрабатывает много сервисов
- Транзакционность и консистентность обеспечивается в приложении
- Поднимают много экземпляров сервиса, каждый может упасть по ошибкам или блокировкам, а система должна работать устойчиво
- Асинхронные сообщения и очереди для выравнивания производительности разных сервисов
- In-memory хранение в базах данных и очередях, сброс в хранилища
- Восстановление при сбоях узлов кластера и дата-центров – техника и базовый софт не обеспечивают межсистемную консистентность

Обычная постановка Снабжение магазинов

Бизнес-процесс –
диаграмма активности



Новой архитектуры
в этой модели нет!



Выделяем
объекты-
документы



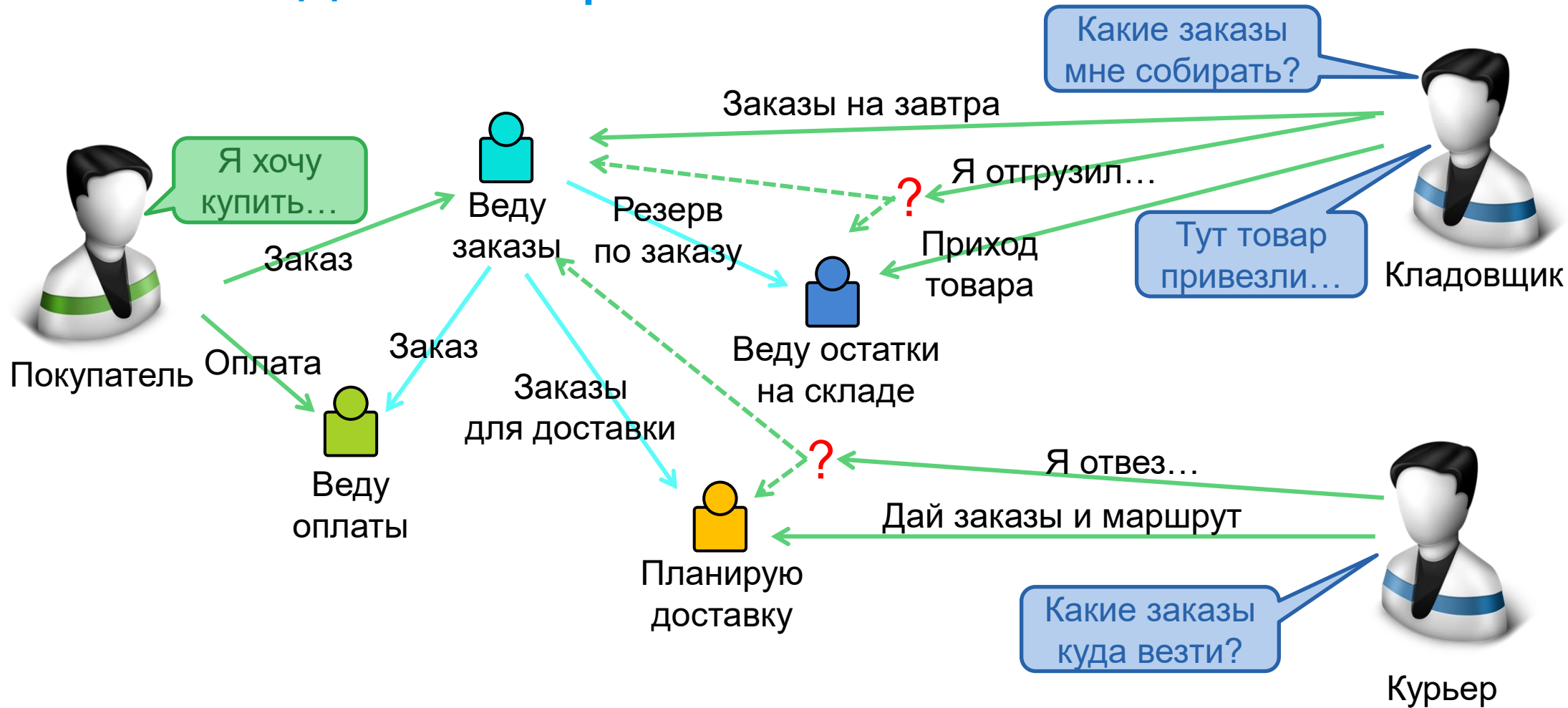
Переходы фиксируют
выполнение бизнес-
функции, а состояния
соответствуют
передаче по этапам
обработки

Какова роль аналитика при проектировании?

- Вариант-1 – отдать все разработчикам, писать user story и use case
 - 😊 Возвращаемся к роли бизнес-аналитика
 - 😞 Не можем проверить реализуемость и целостность постановки
- Вариант-2 – использовать постановки только для передачи контекста
 - 😊 Целостность и реализуемость постановки проверяем, пусть по-старому
 - 😞 Постановка передает задачу разработчику и теряет актуальность
- Вариант-3 – придумать новые метафоры и подходы к постановкам
 - 😊 Проверяем целостность и реализуемость постановки
 - 😊 Постановка соответствует фактической реализации и живет дальше
 - 😞 Для этого надо разобраться и понять, как устроены новые технологии

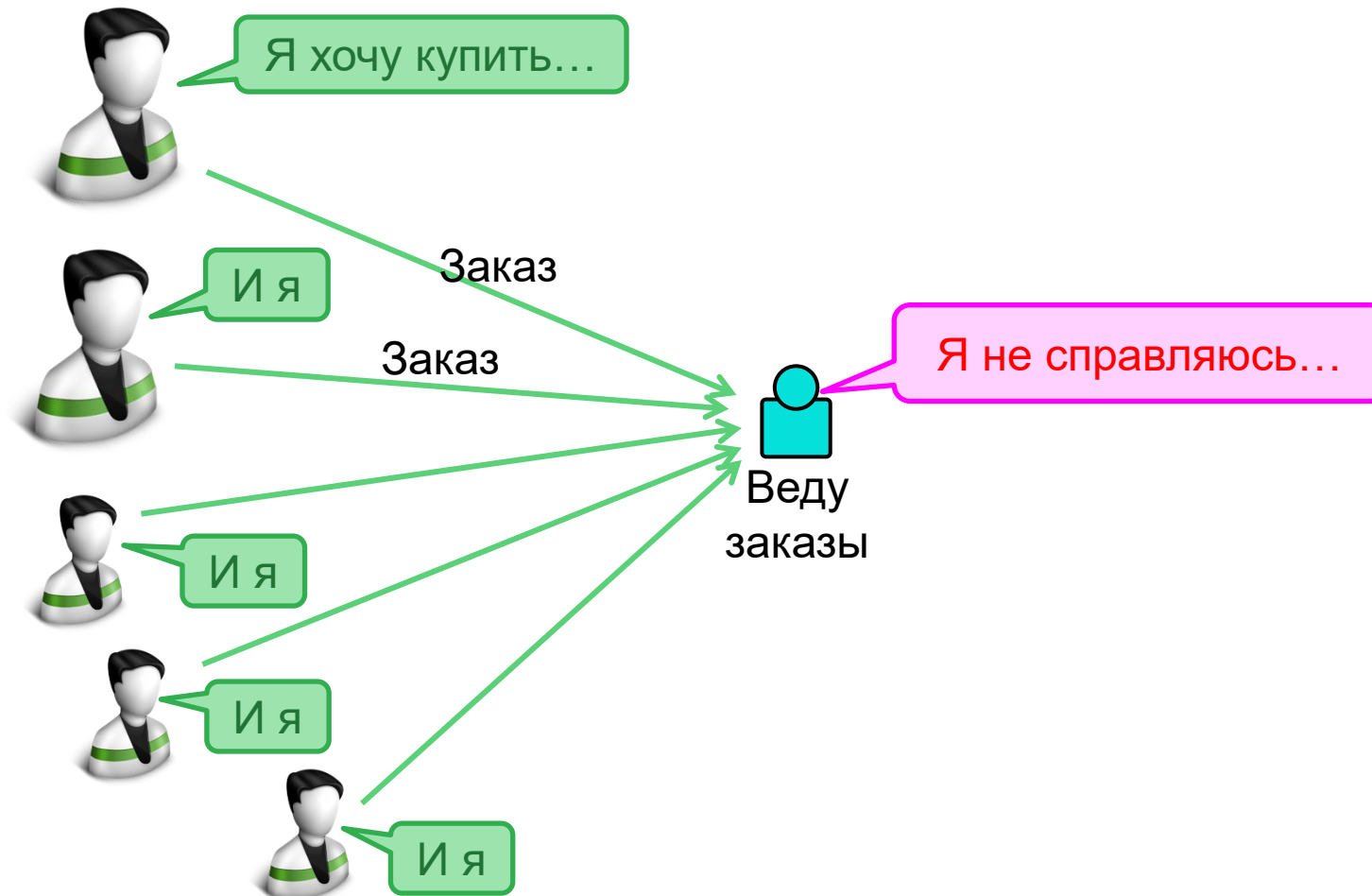
Метафора гномиков –
маленьких человечков,
которые все делают

Гномики для интернет-магазина

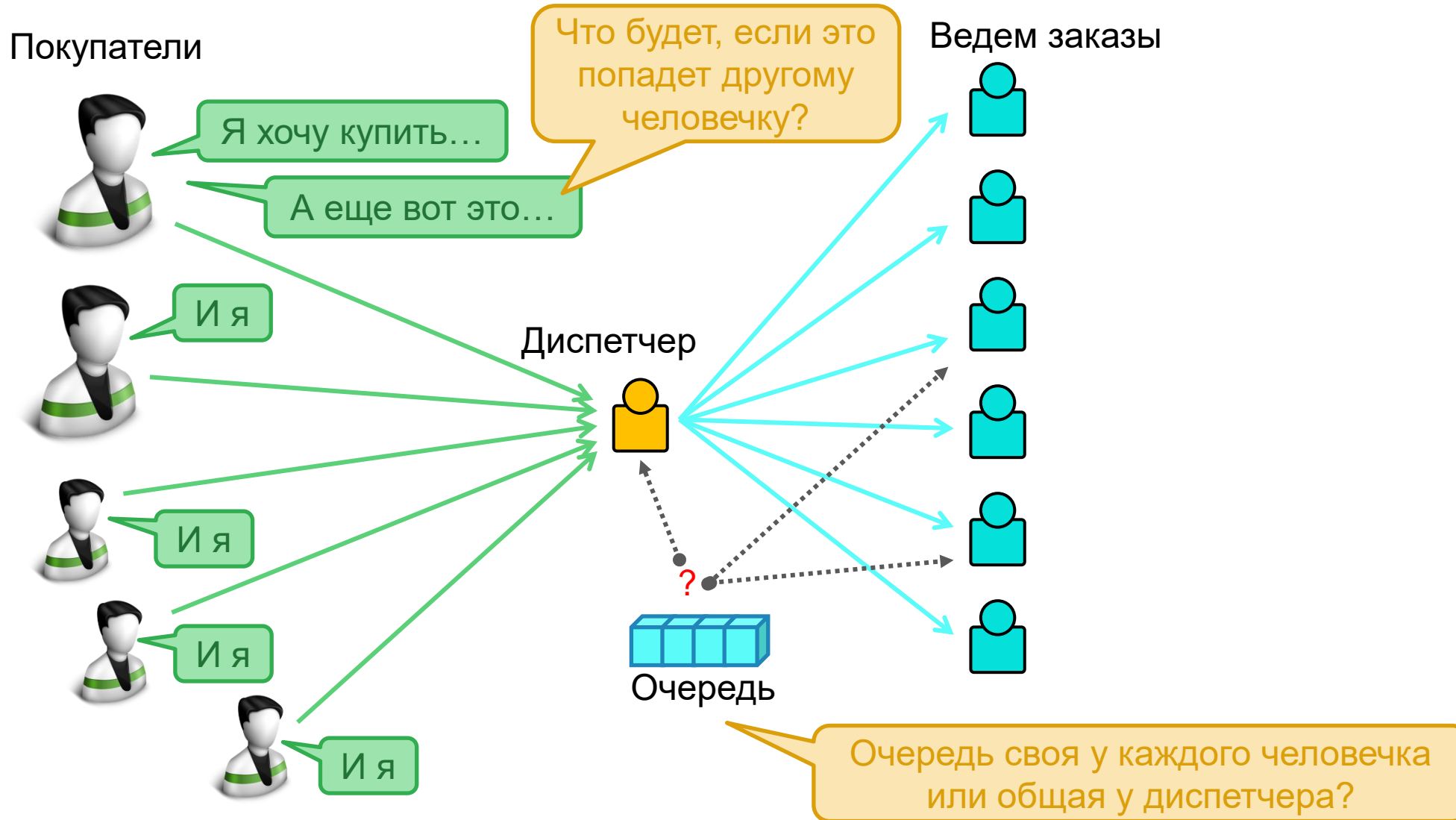


Но у нас много покупателей...

Покупатели

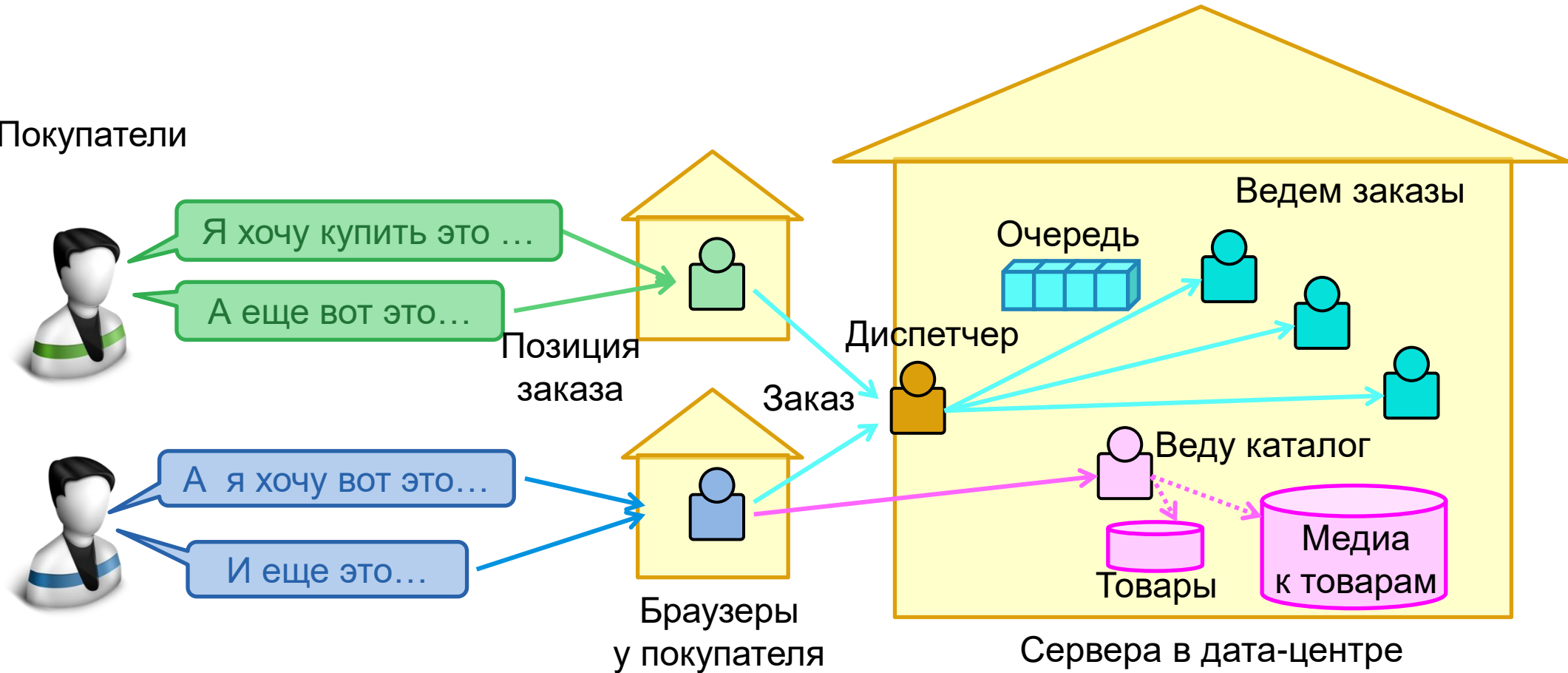


Делаем кластер сервисов приемки заказов



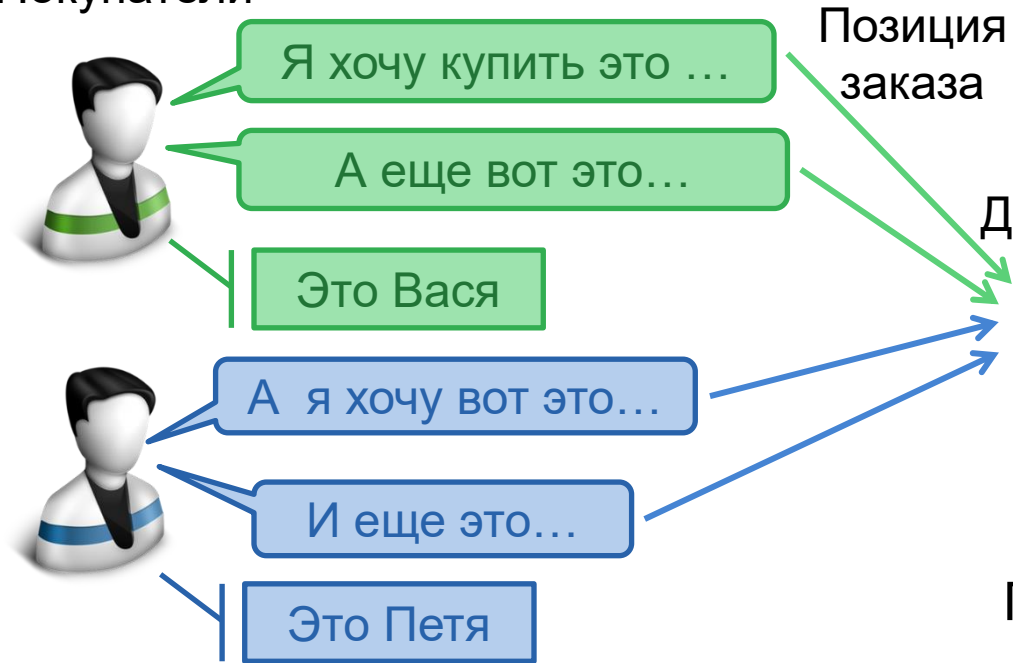
Собираем заказ в браузере

Покупатели



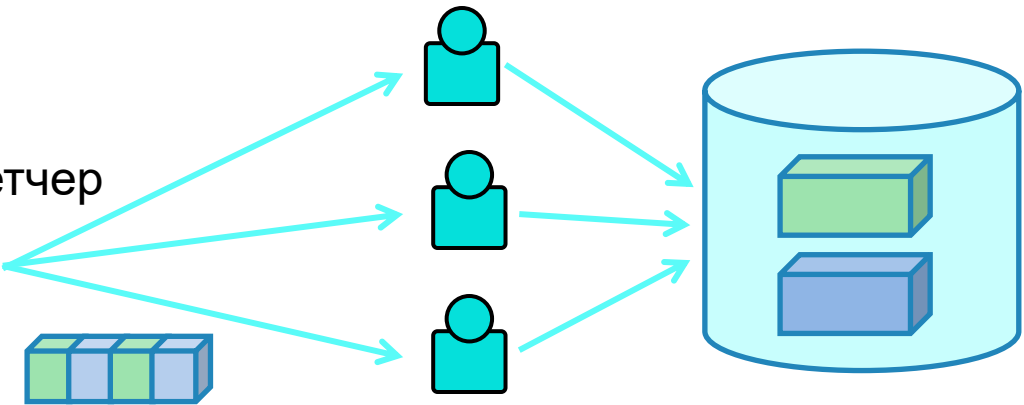
Общая база данных

Покупатели



Идентификация – через авторизацию или токены...

Ведем заказы



Очередь

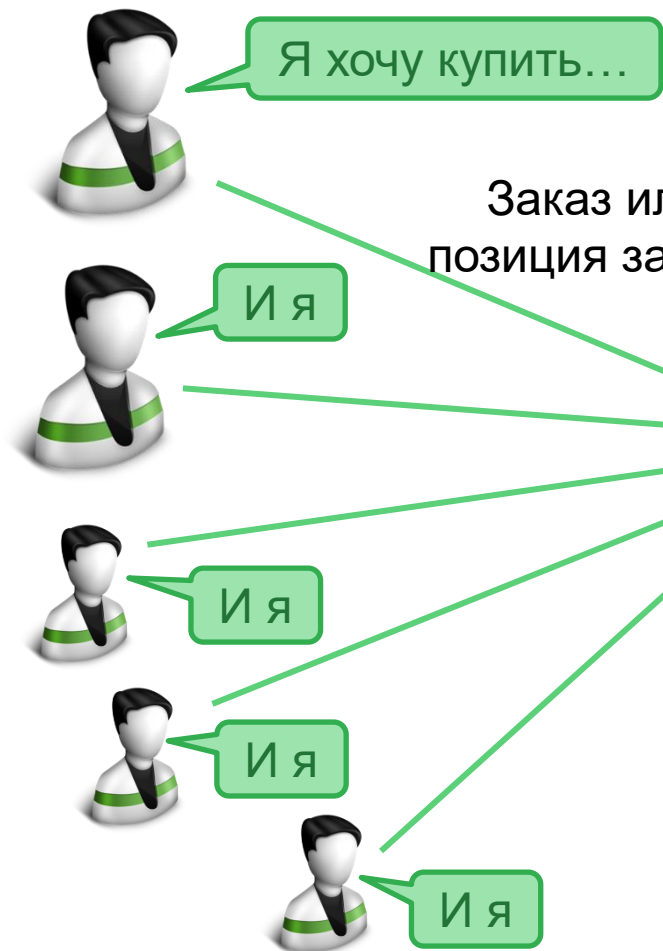
Добавляем позицию в нужный заказ

Проектирование

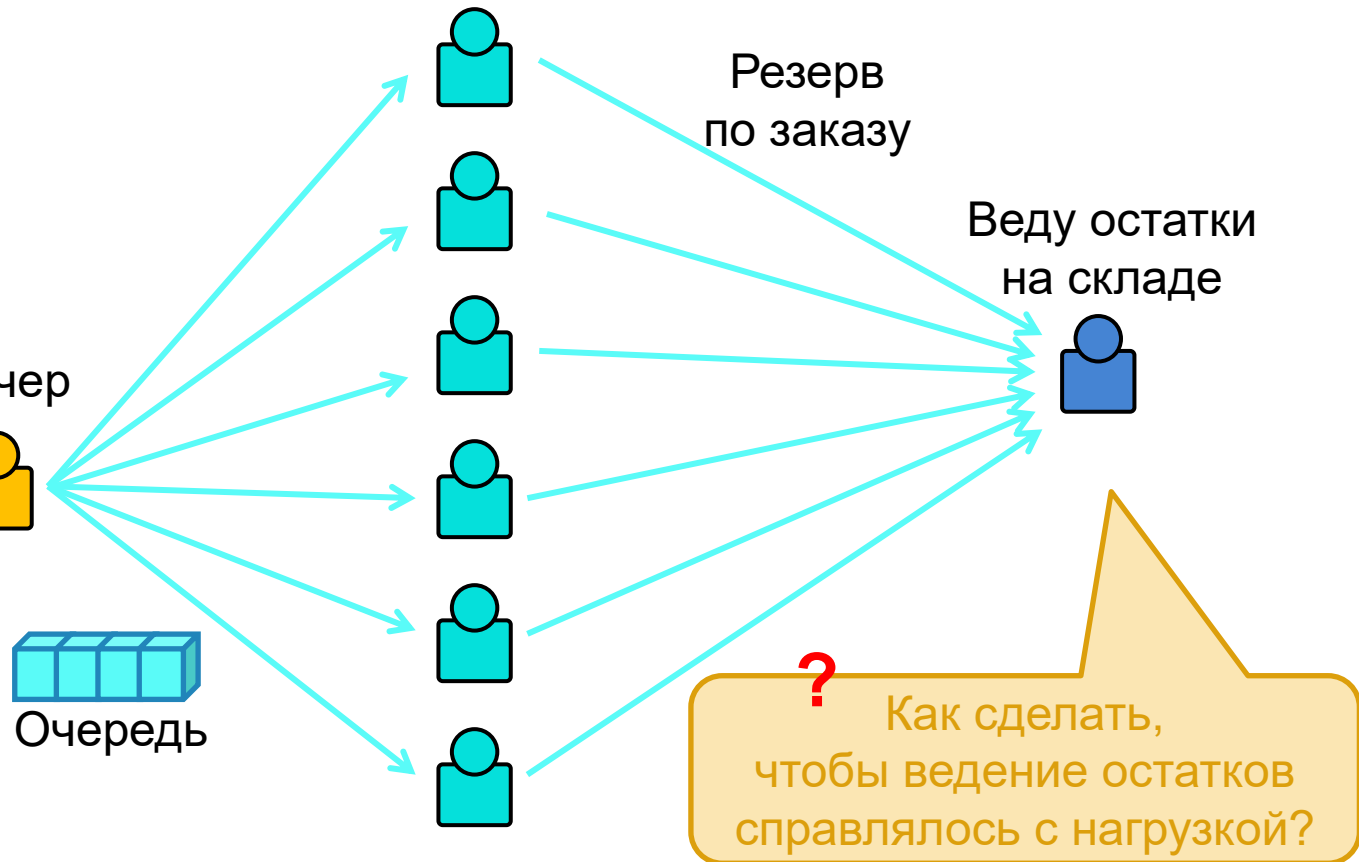
- Как покупатель, начав делать заказ, увидит корзину с другого устройства?
- Как брошенные покупателями корзины не останутся вечным мусором?

Ведение остатка на складе – проблема

Покупатели



Ведем заказы

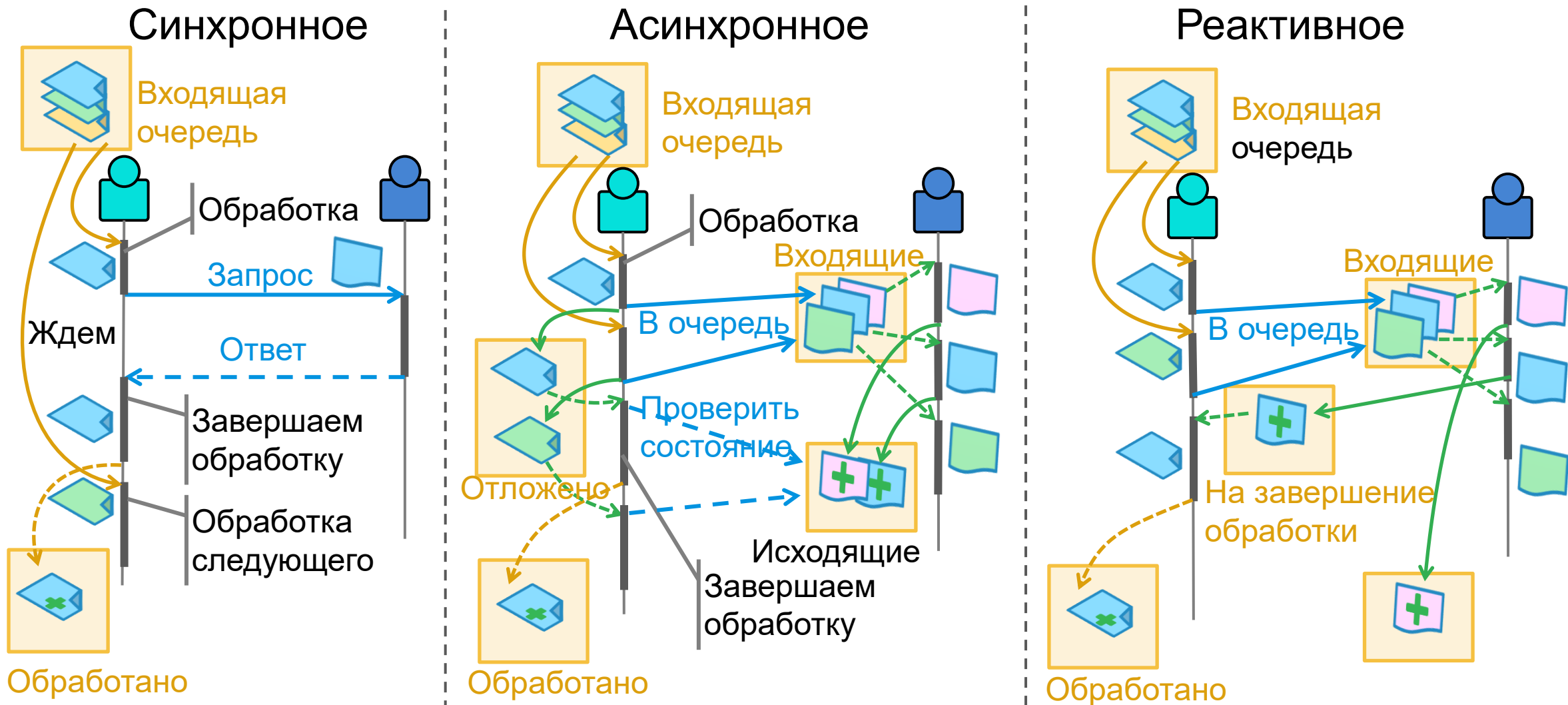


Ведение остатка на складе – варианты

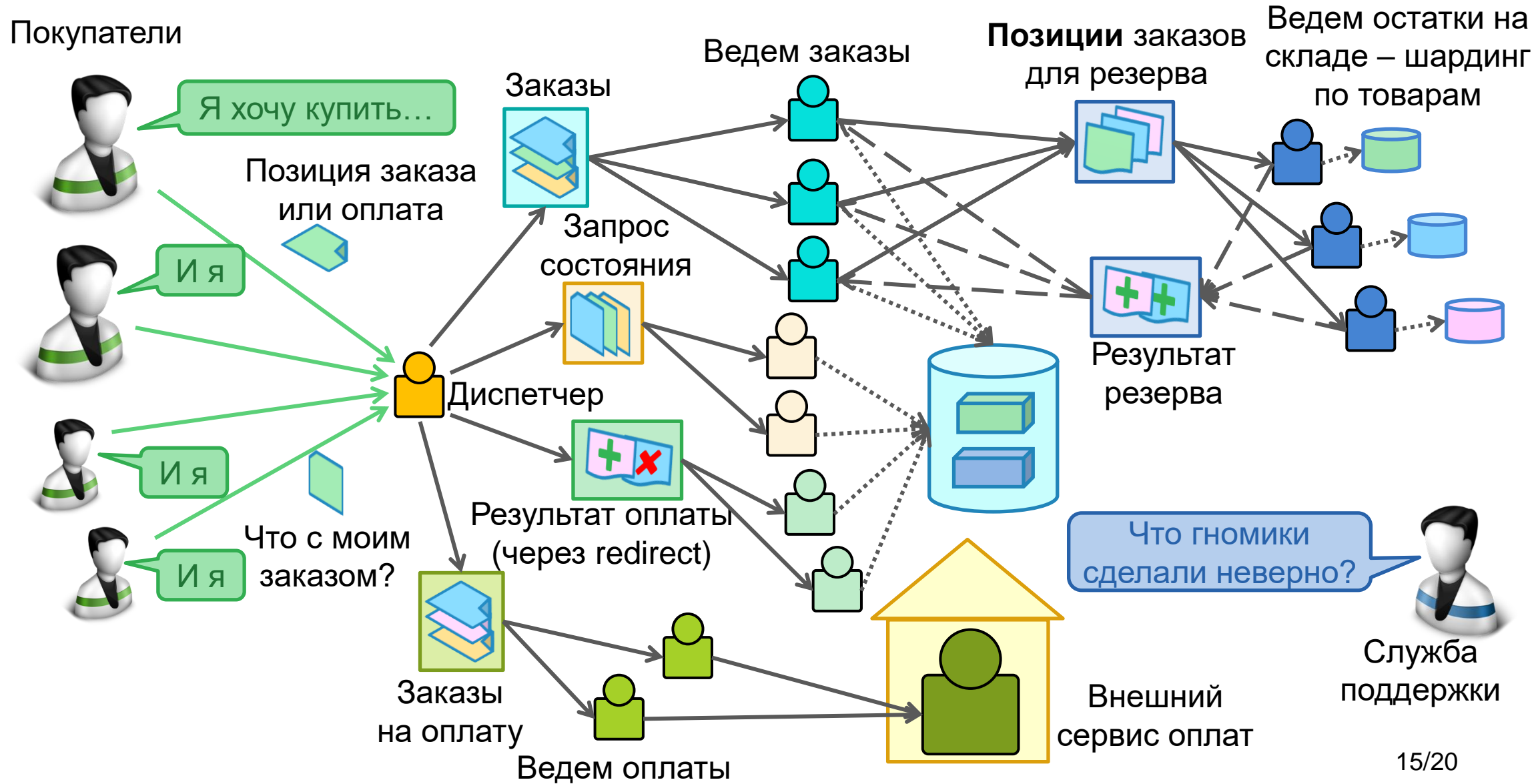
- Очень быстрый гномик: высокопроизводительная БД и железо под узкоспециализированную логику ведения остатков
- Шардирование: несколько гномиков, каждый ведет свои товары, поделить надо равномерно
- Много гномиков логики остатков и быстрая специализированная БД

Между гномиками заказов и гномиками остатков будет очередь на резервирование для выравнивания нагрузки

Варианты межсервисного взаимодействия



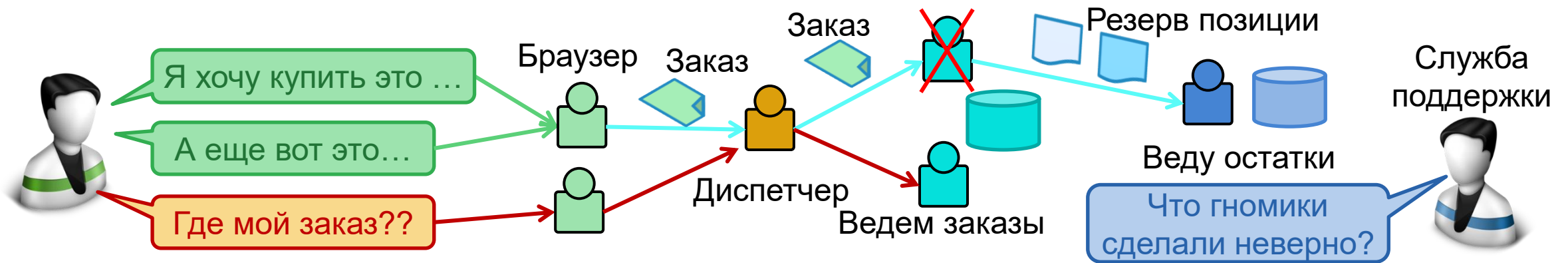
Ведение остатка на складе – по товарам



Кейсы для проектирования

- Как масштабируется каждый из сервисов под нагрузкой?
- Как взаимодействуют сервисы, где и какие есть очереди
- Как обрабатывается ситуация, когда покупатель очень быстро добавил позиции и они попали разным обработчикам одновременно
- Покупатель едет в Сапсане или в месте с плохой связью...
- Покупатель нажал «Оплатить», резервирование идет долго – страница оплаты не появляется – что происходит
- Как решаются ситуации, когда результата оплаты нет?
Можно ли отправить заказ на оплату повторно?
- Как обеспечивается устойчивость при падении экземпляров сервисов?

Устойчивость: гномики умирают, их отстреливают



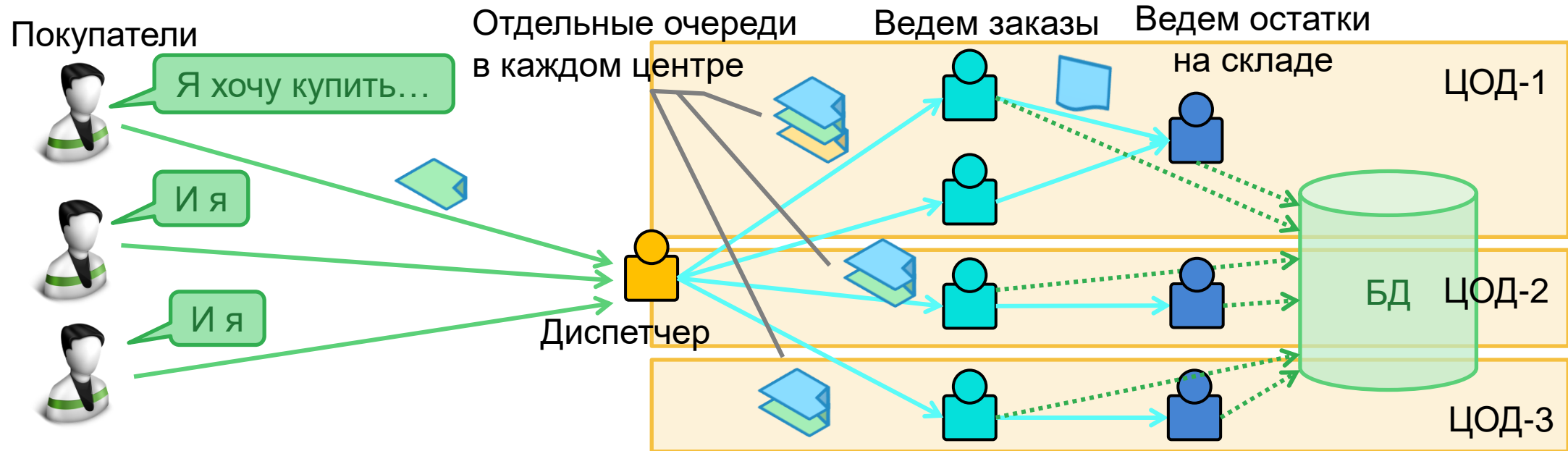
Ситуация: идет обработка и резервирование заказа и в этот момент:

- Инстанс заказов, ведущий резервирование, падает или его убивают...
- Покупатель долго не видит ответа в браузере – и открывает новый...

Вопросы:

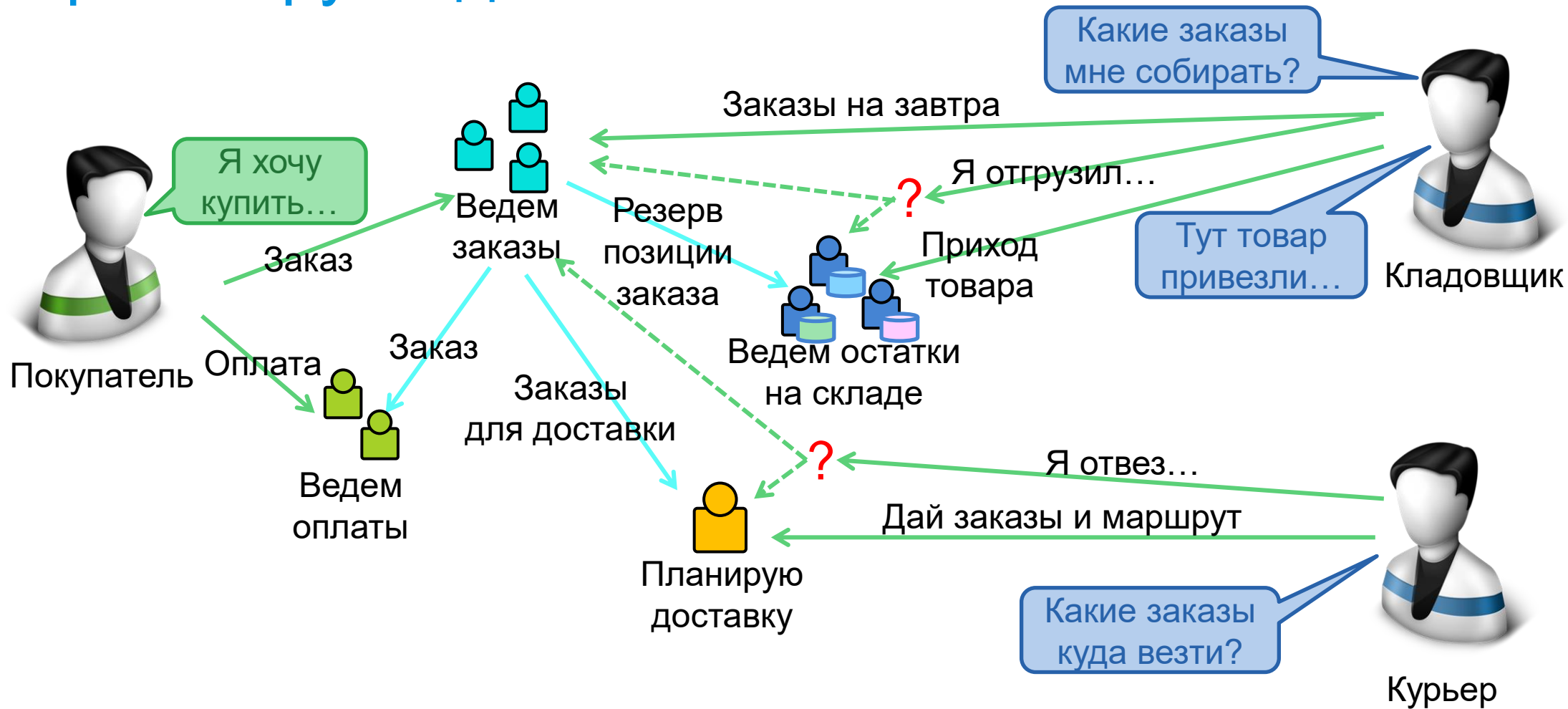
- Как при новом обращении подхватить имеющееся резервирование?
- Как сделать, чтобы резервирования не зависали?

Надежность: ноды в разных датацентрах



- Метафора: ЦОД – дома для гномиков, а ноды – комнаты
- Обращение в соседнее помещение – дольше или невозможно
- Надо 3 ноды или ЦОДа – чтобы отличить пропажу связи от падения, в метафоре: соседний дом сгорел или телефон не работает

Проектируем дальше...



Проектируем дальше...



А теперь –
кто хочет разобрать свой кейс?

Завершаем мастер-класс

- Получилось ли почувствовать, как работает модель?
- Какие вы получили инсайты?
- Что было самым сложным?
- Видите ли вы применение в ваших проектах?
- Будете ли применять?



Вопросы? Обращайтесь!



Максим Цепков

<http://mtsepkov.org>

[Tel: @MaximTsepkov](mailto:@MaximTsepkov)

Анна Абрамова

[Tel: @annasergeevnaorg](mailto:@annasergeevnaorg)

