
HighLoad++ FOUNDATION

CUSTIS® ИТ-РЕШЕНИЯ ДЛЯ РАЗВИТИЯ

# Визуальное проектирование масштабируемых приложений

**Максим Цепков**

Главный архитектор решений CUSTIS
Навигатор в мире Agile, бирюзовых организаций и Спиральной динамики

CUSTIS

# Немного истории: почему нужны новые модели?

# Классические приложения

CUSTIS

- Конкурентная работа, транзакционность и консистентность — в СУБД
- Трехзвенная архитектура добавила серверную бизнес-логику на объектном языке
- Масштабирование — за счет железа и системного софта
- Классическое проектирование: **процедурный** и **объектный подход**
- Есть хорошие средства визуального проектирования: ER-модель, UML и др.

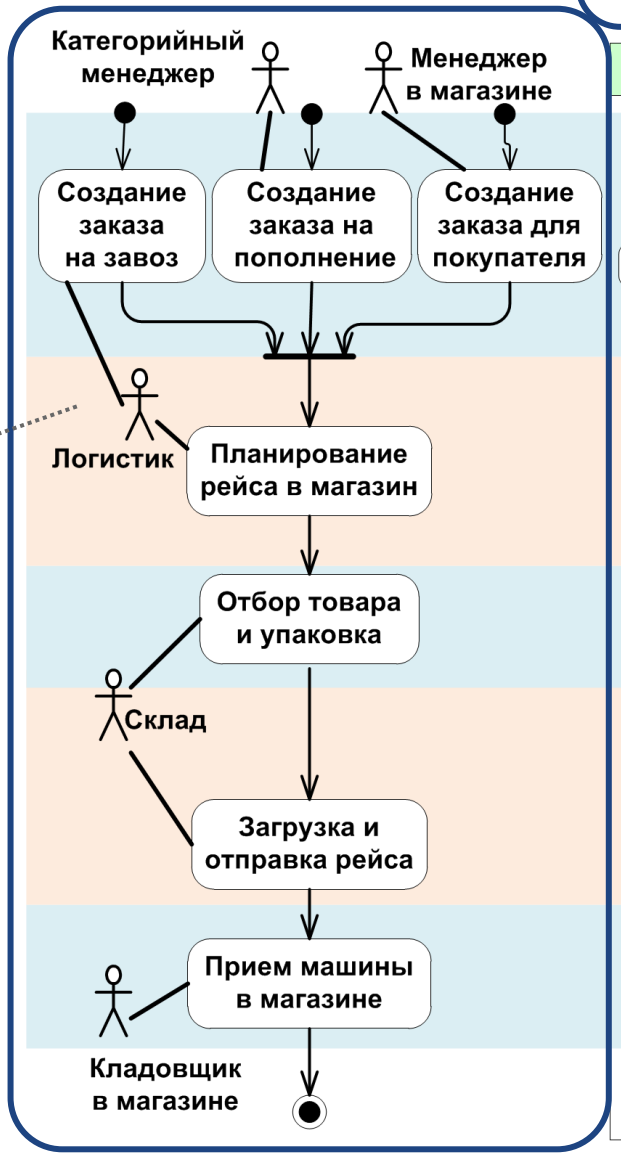
# Классическая постановка

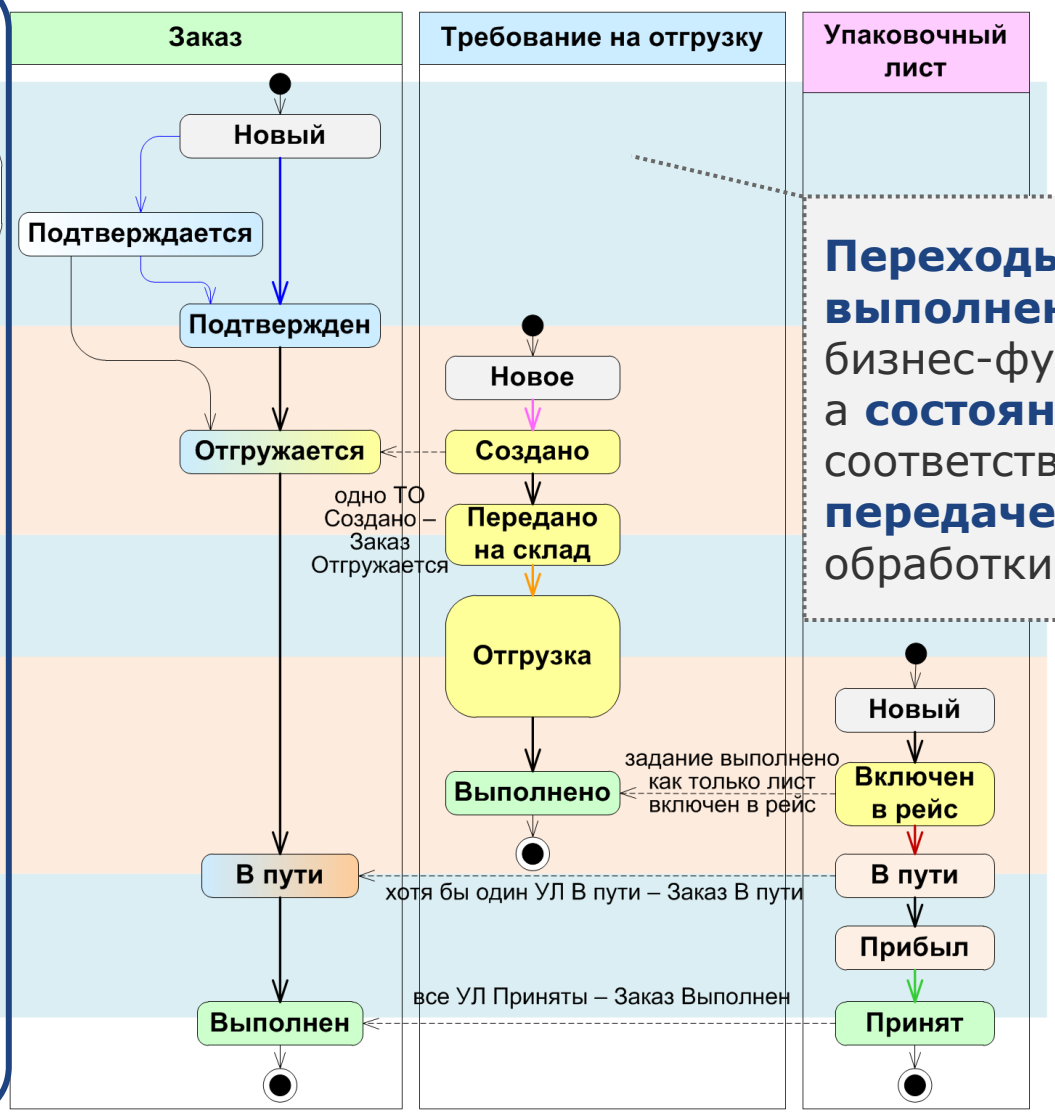
CUSTIS

Снабжение магазинов


Выделяем объекты-документы

Бизнес-процесс: диаграмма активности




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

# Процедурный и объектный подход: пример

CUSTIS

**Задача — интернет-магазин: заказы, оплата, склад, отгрузка, доставка**

## Процедурный подход

- Таблицы товаров, заказов, платежей, остатков на складе, курьеров, доставок
- Алгоритмы: фиксация оплаты, назначение даты доставки, планирование курьеров
- Интерфейсы: какие экраны, какие данные показываем и действия выполняем

## Объектный подход

- Объекты: товар, заказ, платеж, курьер. Доставка — отдельный объект в заказе?
- Алгоритмы: в методах, инкапсуляция внутренней логики объектов
- Интерфейсы: витрины каких объектов представляем, какие методы доступны

Доставка самовывозом и курьером

- Особенности размазаны по всем алгоритмам через условия
- Делаем подтипы, инкапсулируя особенности поведения

# Процедурный и объектный подход: проектирование

CUSTIS

**Процедурный подход**

- Структура БД
- Интерфейсы
- Алгоритмы обработки
- Процедуры API backend и API RPC (если необходимы)

**Объектный подход**

- Типы и статусы объектов
- Методы бизнес-логики
- Интерфейсы в стиле **naked object** — витрины объектов и методы
- REST API

Бизнес-логика

- Распределена по разным процедурам

- Инкапуслирована в объектах

# Развитие объектного подхода

CUSTIS

В реализации может быть анемичная модель транспортных объектов
и соответствующие тем же объектам контролеры для бизнес-логики

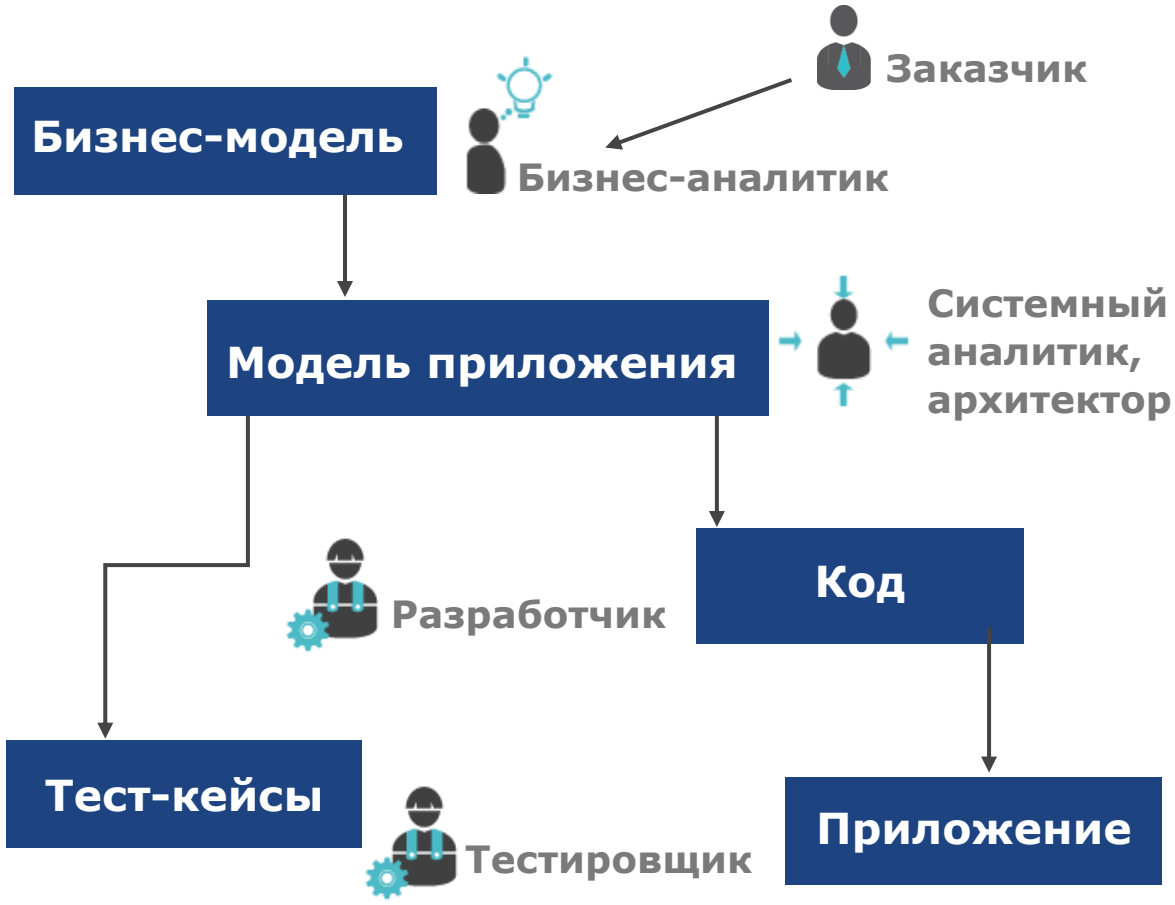
**DDD** распространил объектный подход на модель предметной области:

- Вместо словаря мы делаем **онтологию** понятий и связей
- Концепция **bounded context:** декомпозируем предметную область на фрагменты, применяя методы ООП — инкапсуляция, наследование, выделение общего и др.

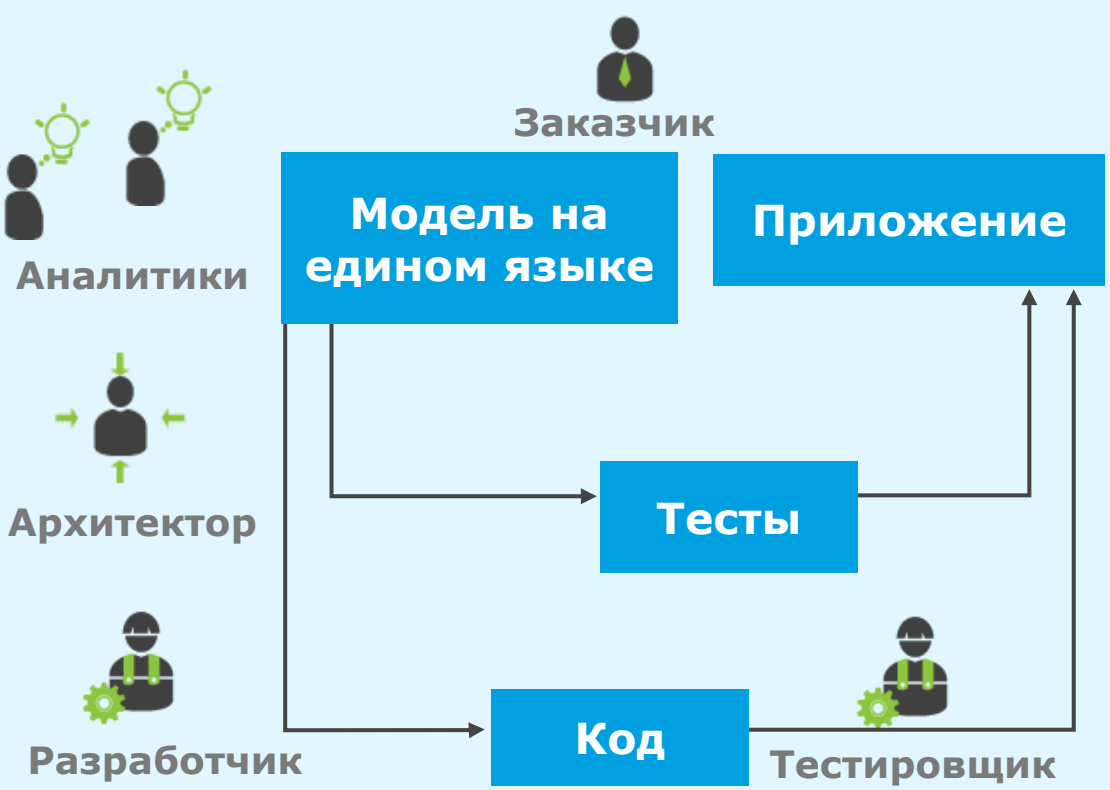
# DDD — единый язык и единая модель приложения

CUSTIS

## Раньше




## DDD




**Единый язык создает общее пространство общения**

# Зачем нужен единый язык?

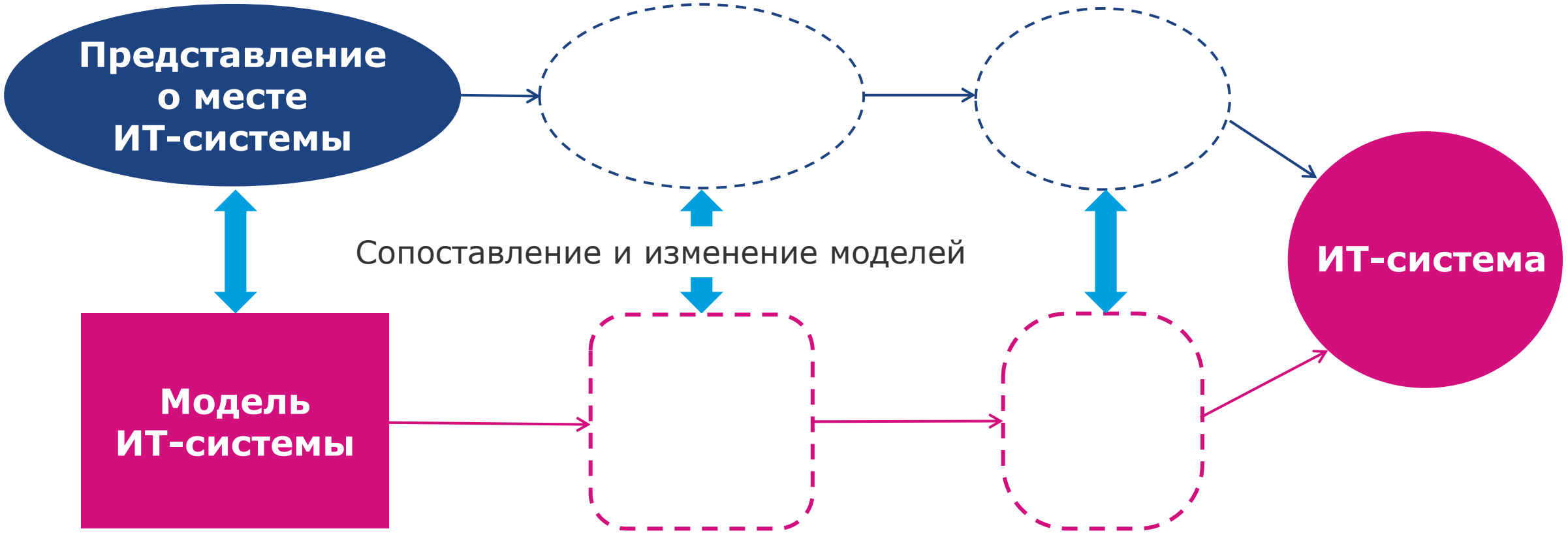
CUSTIS


Модель системы **не соответствует** представлению бизнеса о ее месте в модели предприятия

*Не то чтобы совсем не попал,*
*но только не попал в шарик…*

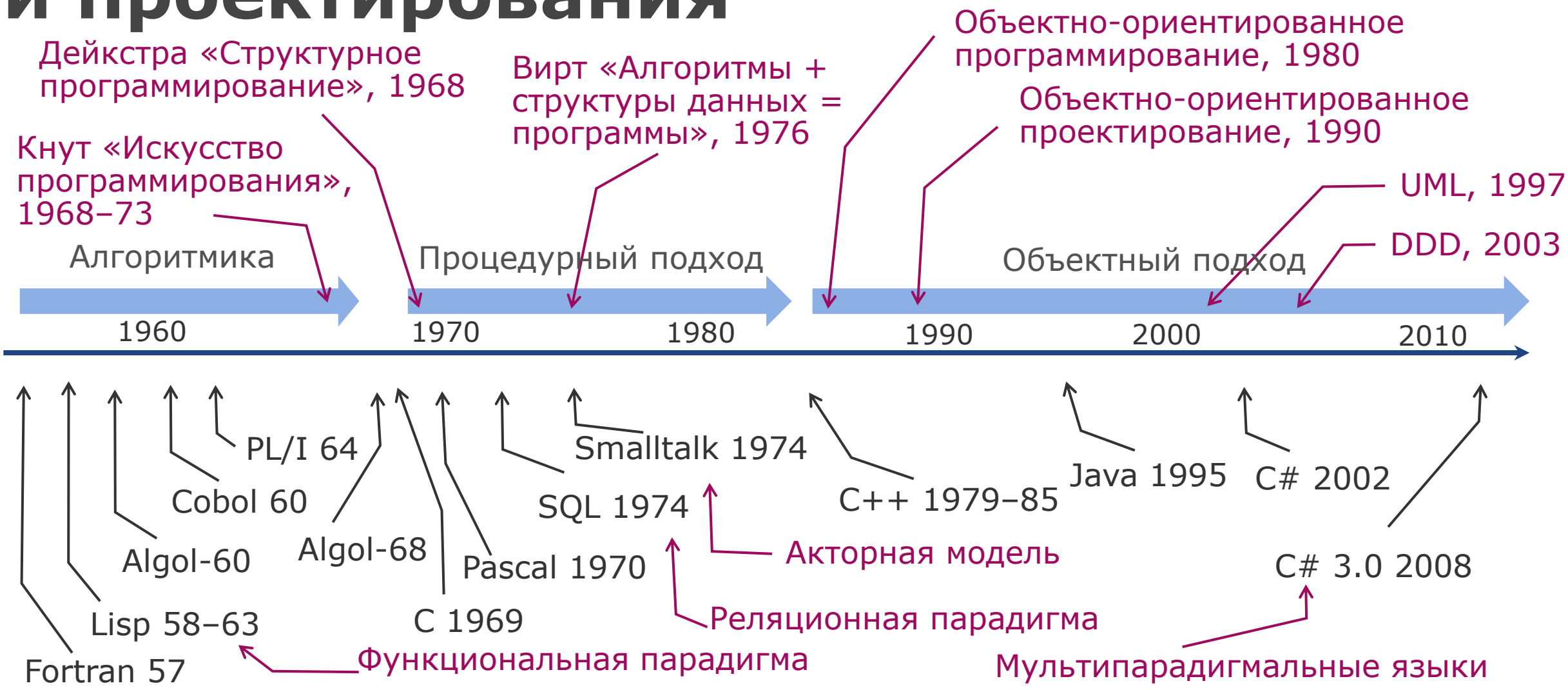
# Итерационное развитие модели

CUSTIS




Единый язык позволяет совместить модель системы
с представлениями бизнеса о ее месте

# История программирования и проектирования




*Мои статьи «История IT. Когда компьютеры были большими...» и «История IT. ООП» и предыдущие доклады*

# Масштабирование public web

## Уход от реляционных СУБД

- NoSQL-базы данных, использование многих БД одним приложением
- Кластерное развертывание с независимым хранением на узлах
- Транзакционность и консистентность обеспечиваются в приложении
- При восстановлении узла кластера данные неконсистентны

## Сервисные и микросервисные архитектуры

- Каждый бизнес-запрос обрабатывает много сервисов
- Много экземпляров одного сервиса для масштабирования
- Экземпляры сервисов падают по ошибкам или блокировкам
- Асинхронные сообщения, очереди выравнивают производительность

# Нужна новая модель проектирования

CUSTIS

- Показывать способы масштабирования для отдельных сервисов
- Отражать способы взаимодействия между множествами сервисов
- Моделировать поведение при падении экземпляров сервисов, проектировать устойчивость системы в целом
- Рассматривать восстановление при сбоях узлов кластера и дата-центров — техника и базовый софт не обеспечивают консистентного восстановления

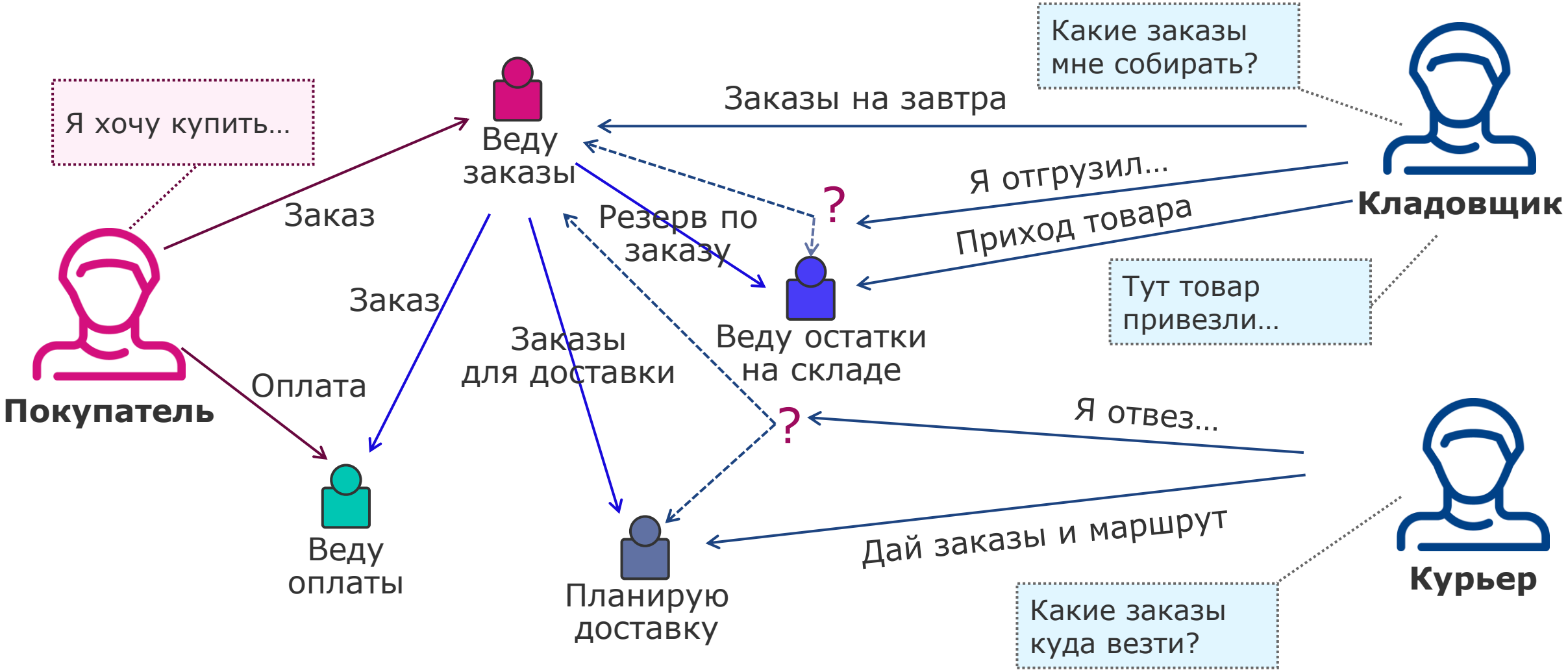
CUSTIS

# Решаем проблему: метафора гномиков — человечков, которые все делают

Гномик представляет актора, а олицетворение делает устройство системы понятным не только разработчикам, но и аналитикам, тестировщикам, бизнес-заказчикам

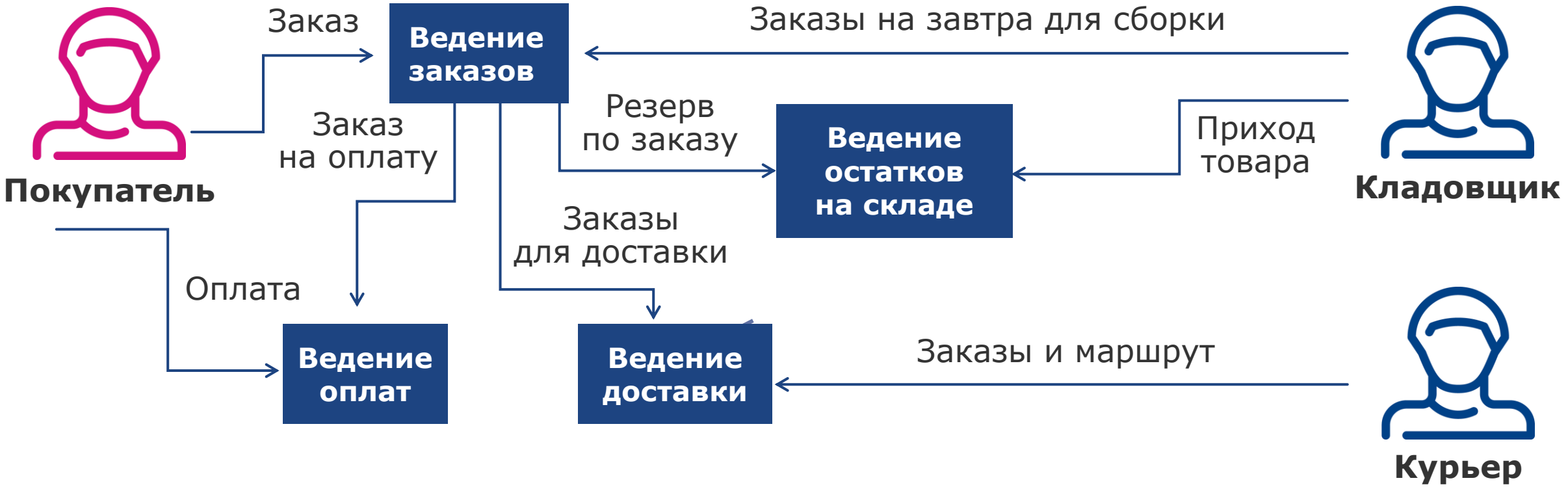
# Гномики для интернет-магазина

CUSTIS

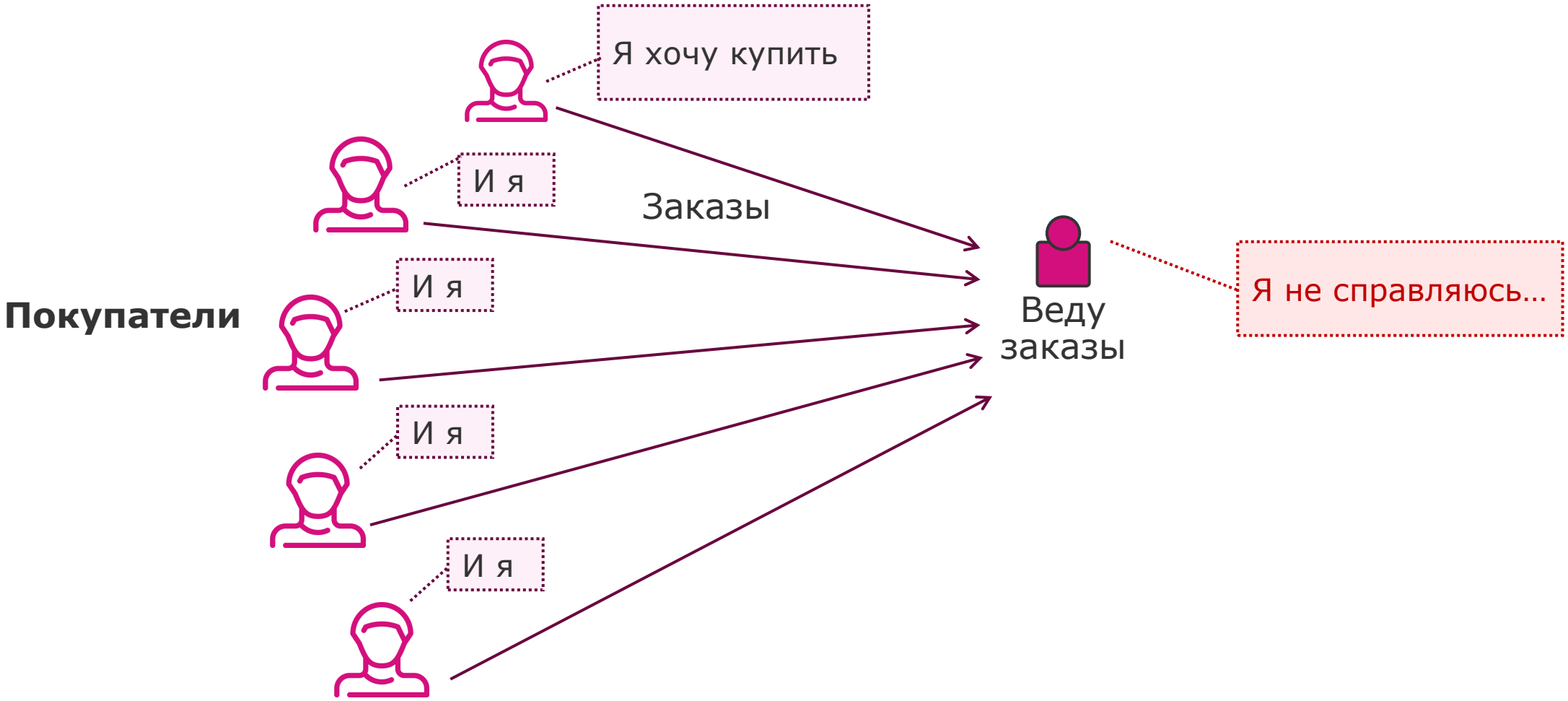


# Схема сервисов для интернет-магазина

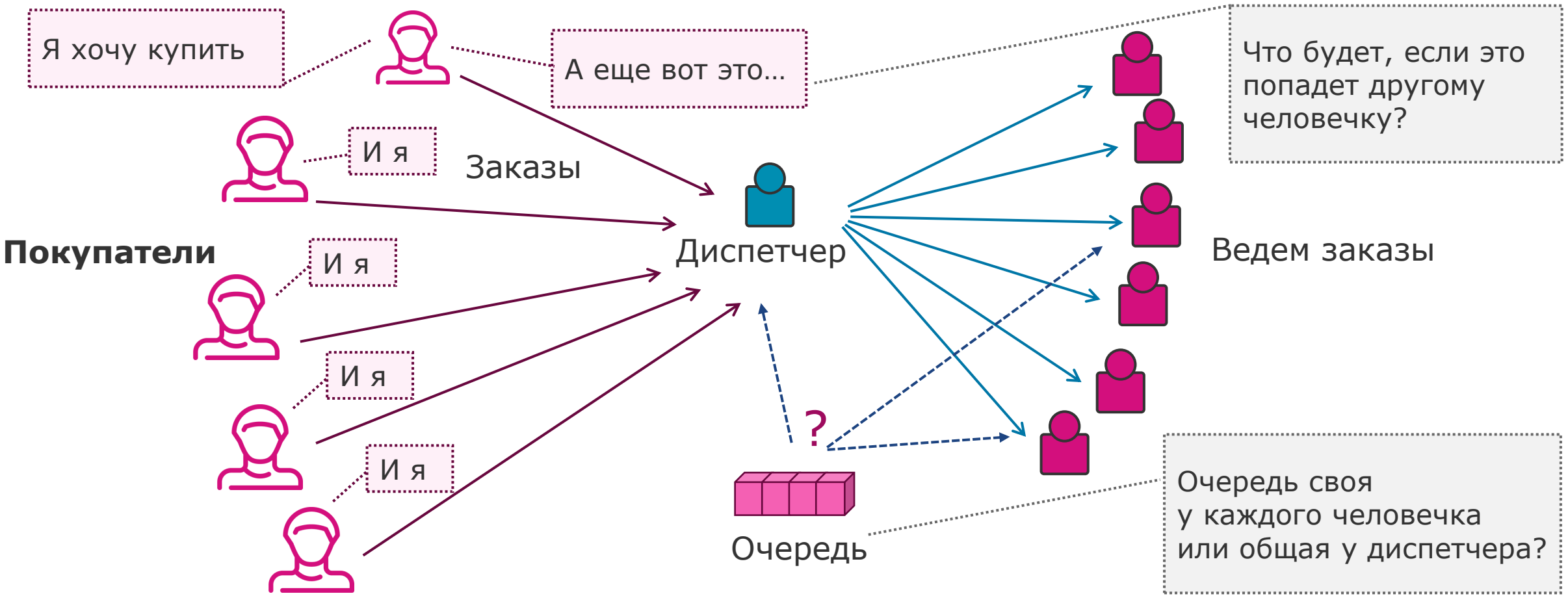
CUSTIS



# Но у нас много покупателей...

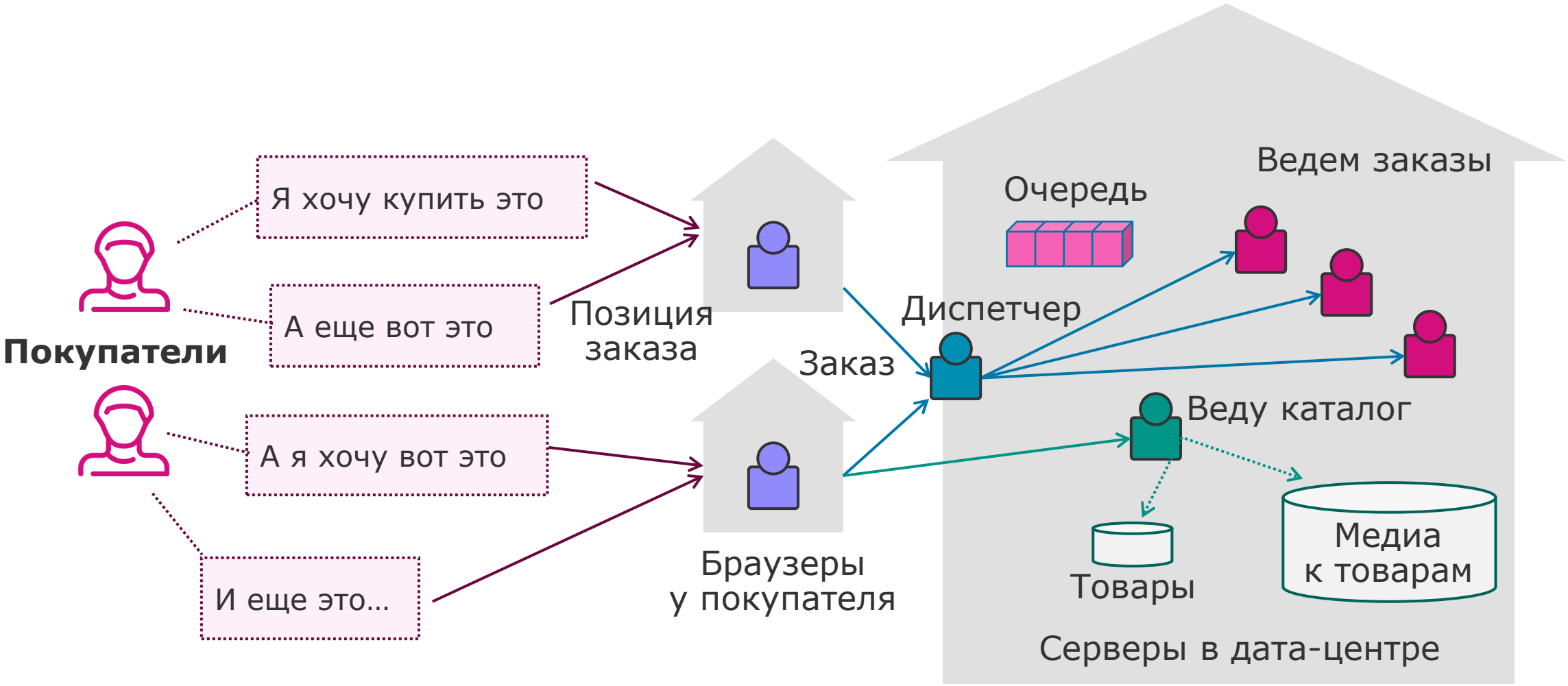


# Делаем кластер сервисов приемки заказов



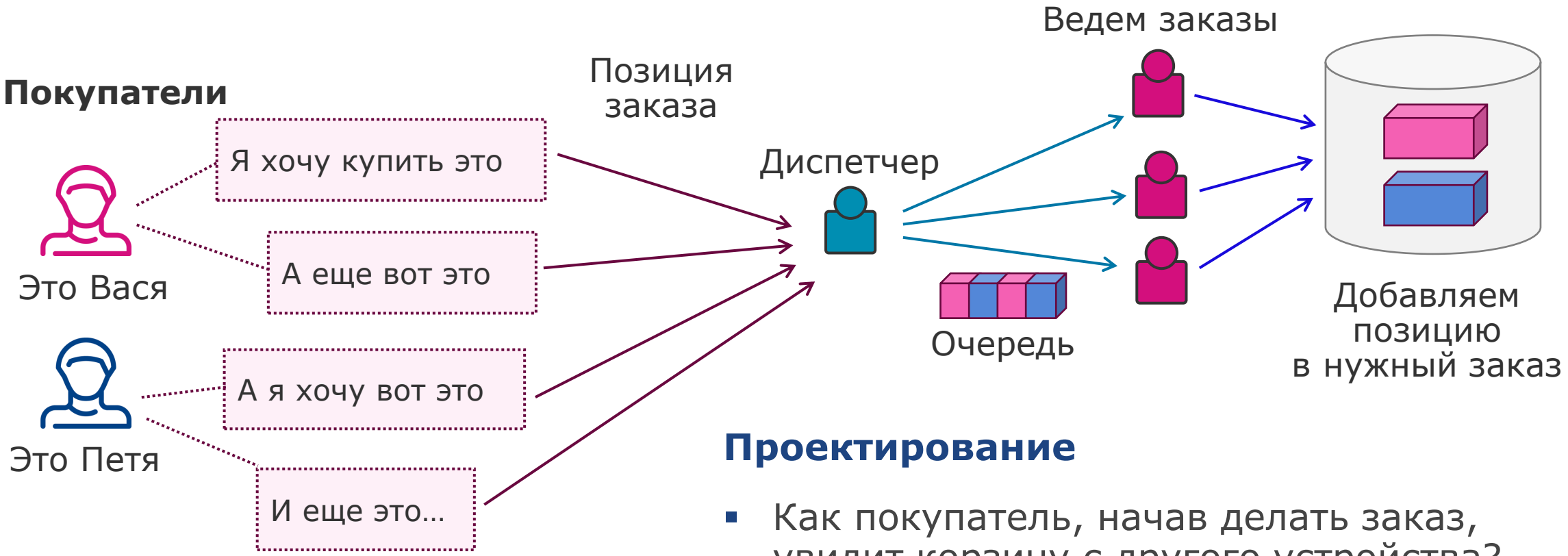
# Собираем заказ в браузере

CUSTIS



# Общая база данных



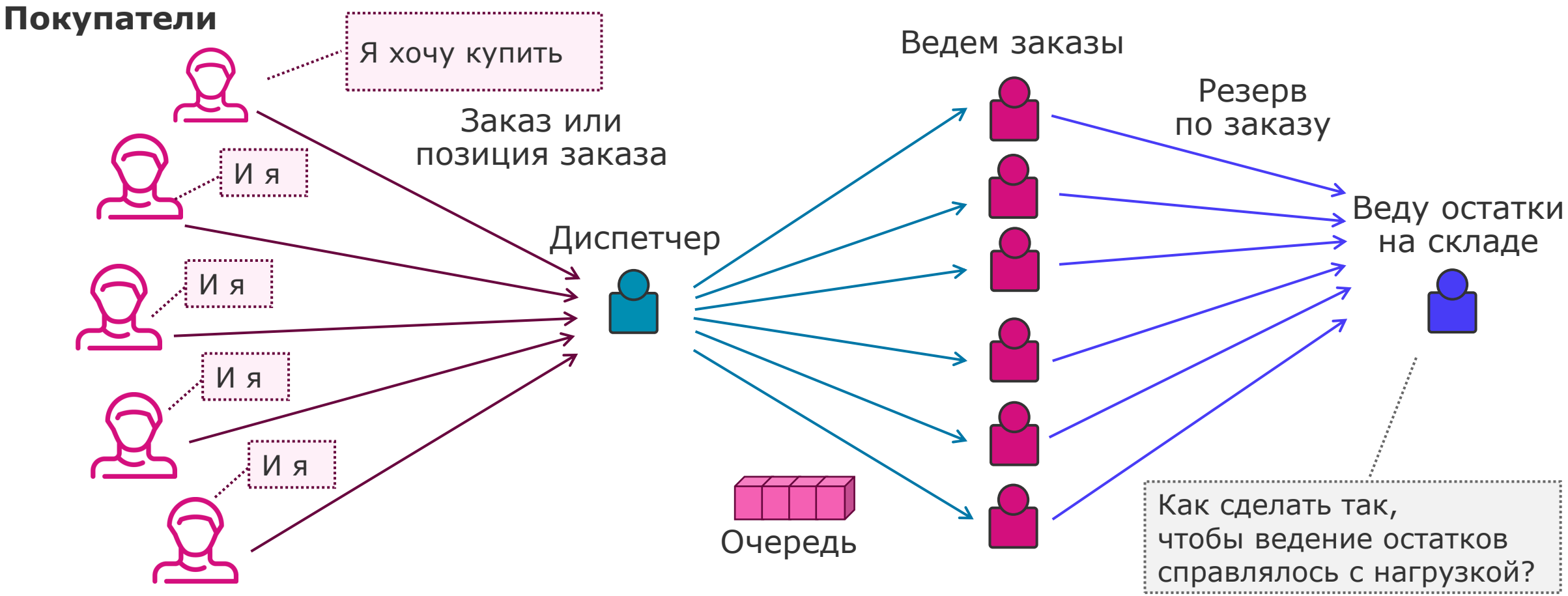



Идентификация — через авторизацию или токены

## Проектирование

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

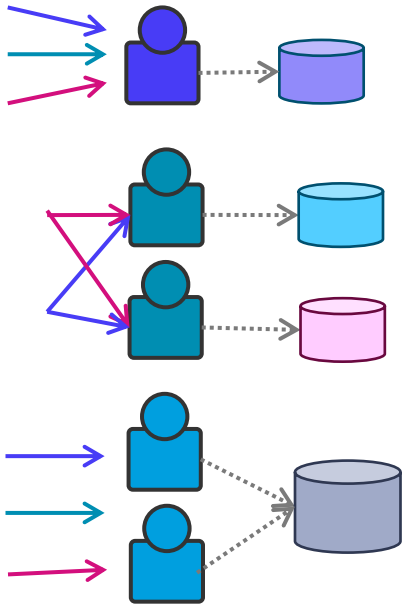
# Проблема: ведение остатка на складе



# Варианты ведения остатка на складе

CUSTIS

- 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 высокопроизводительная БД, если критичны ресурсы процессора, например, встроена проверка полномочий агента на использование остатка

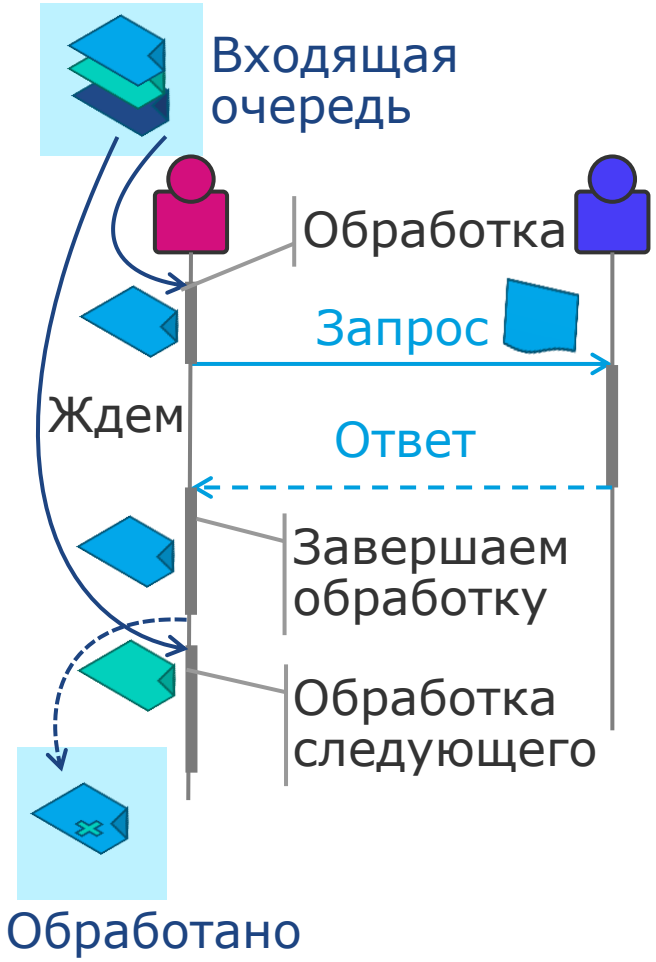


Между гномиками заказов и гномиками остатков будет очередь на резервирование для выравнивания нагрузки (не показана)

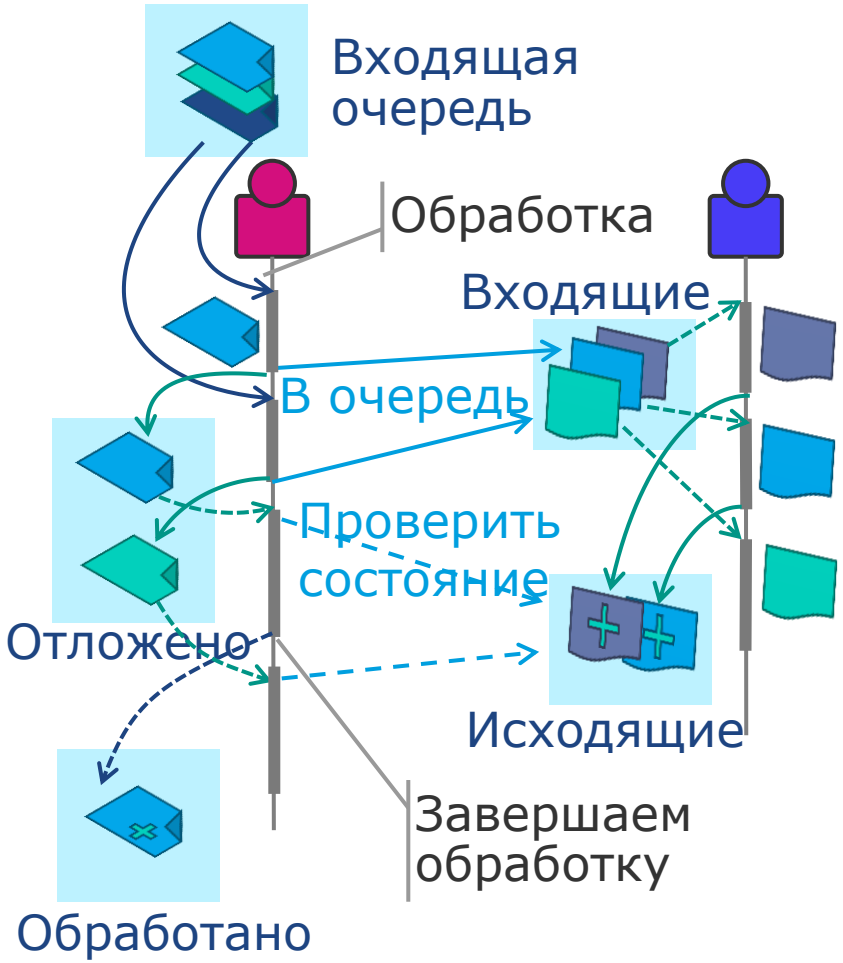
# Варианты межсервисного взаимодействия

CUSTIS

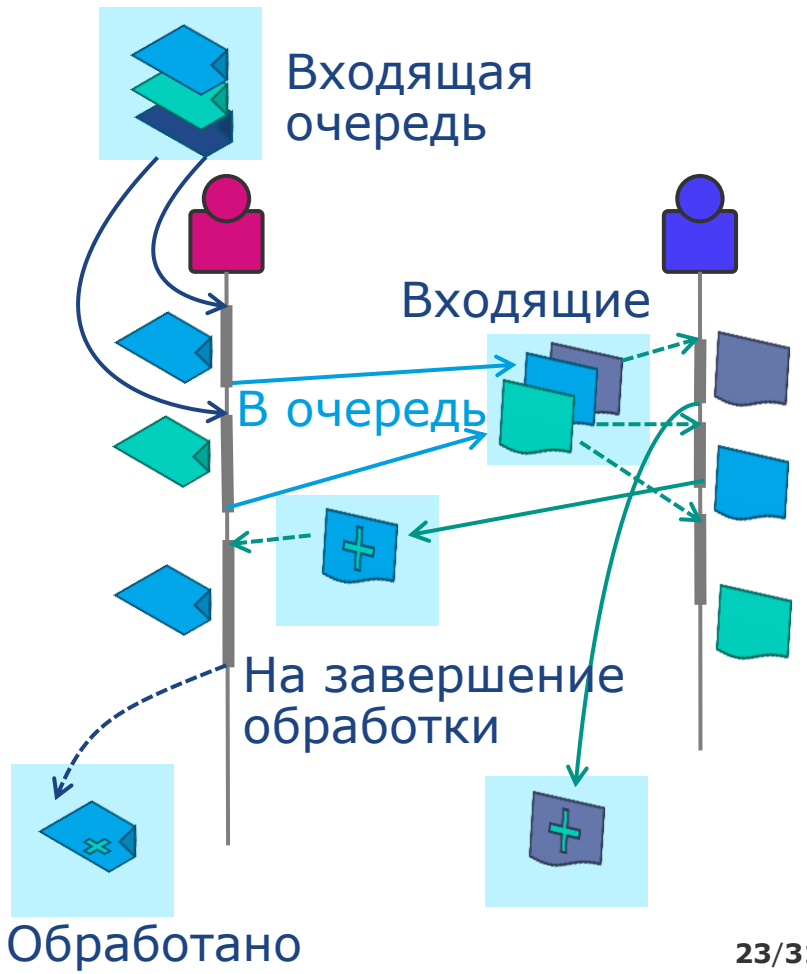
## Синхронное




## Асинхронное

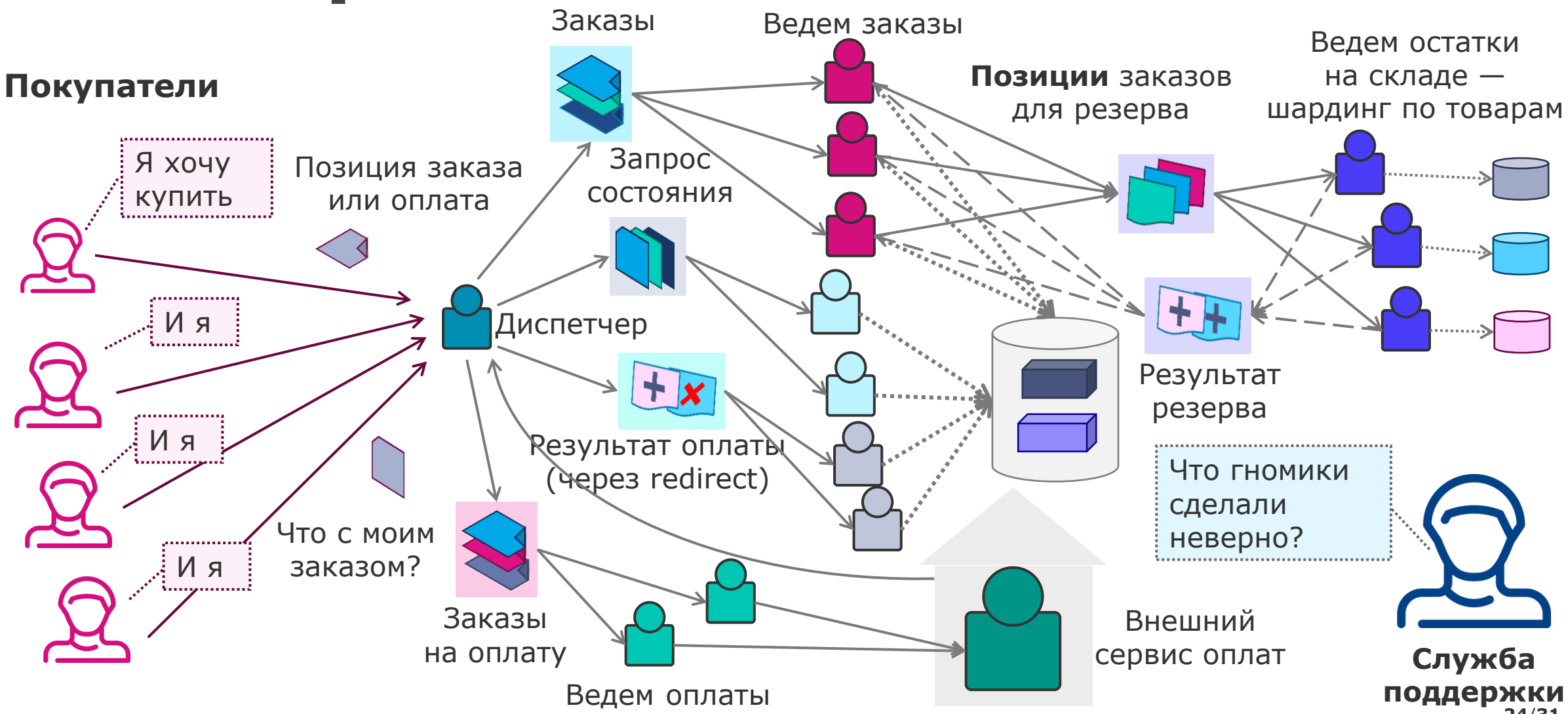



## Реактивное



# Ведение остатка на складе по товарам

CUSTIS

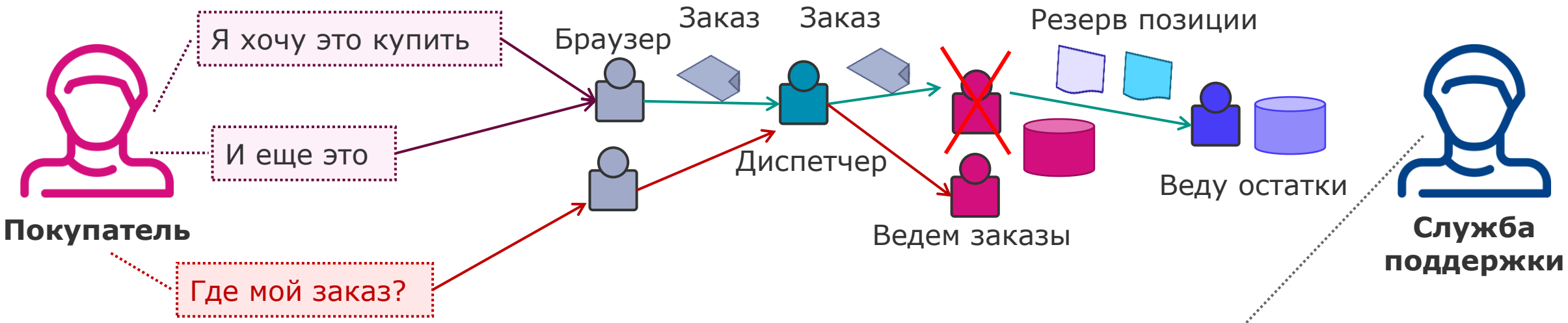


# Кейсы для проектирования

CUSTIS

- Как обрабатывается ситуация, когда покупатель очень быстро добавил позиции, и они попали разным обработчикам одновременно?
- 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?

# Устойчивость: гномики исчезают




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

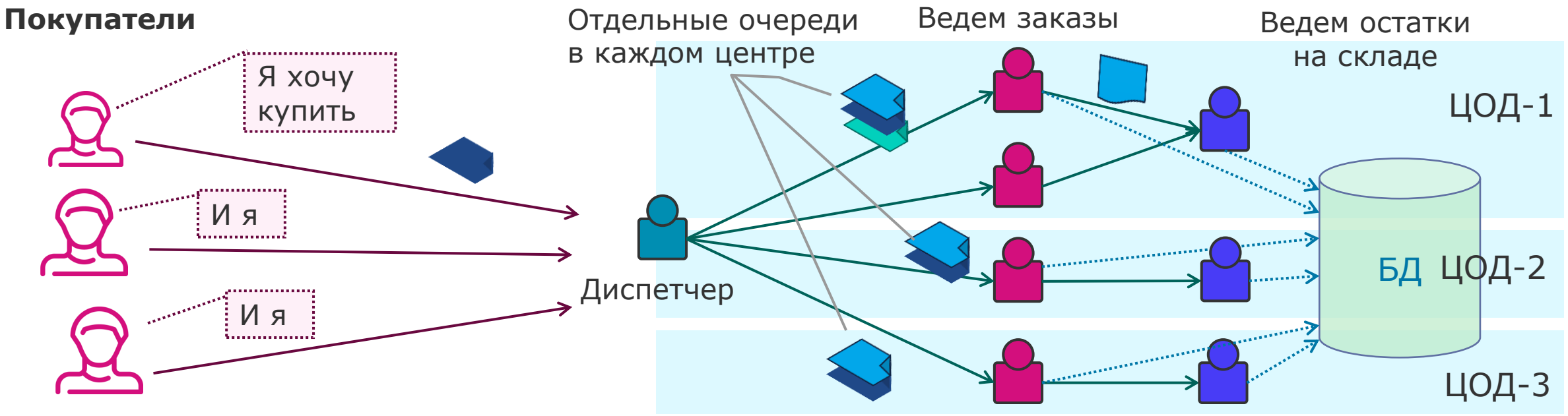
- Инстанс заказов, ведущий резервирование, падает или его глючит…
- Покупатель долго не видит ответа в браузере — и открывает новый

**Вопросы:**

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

Что гномики сделали неверно?

# Надежность: ноды в разных дата-центрах



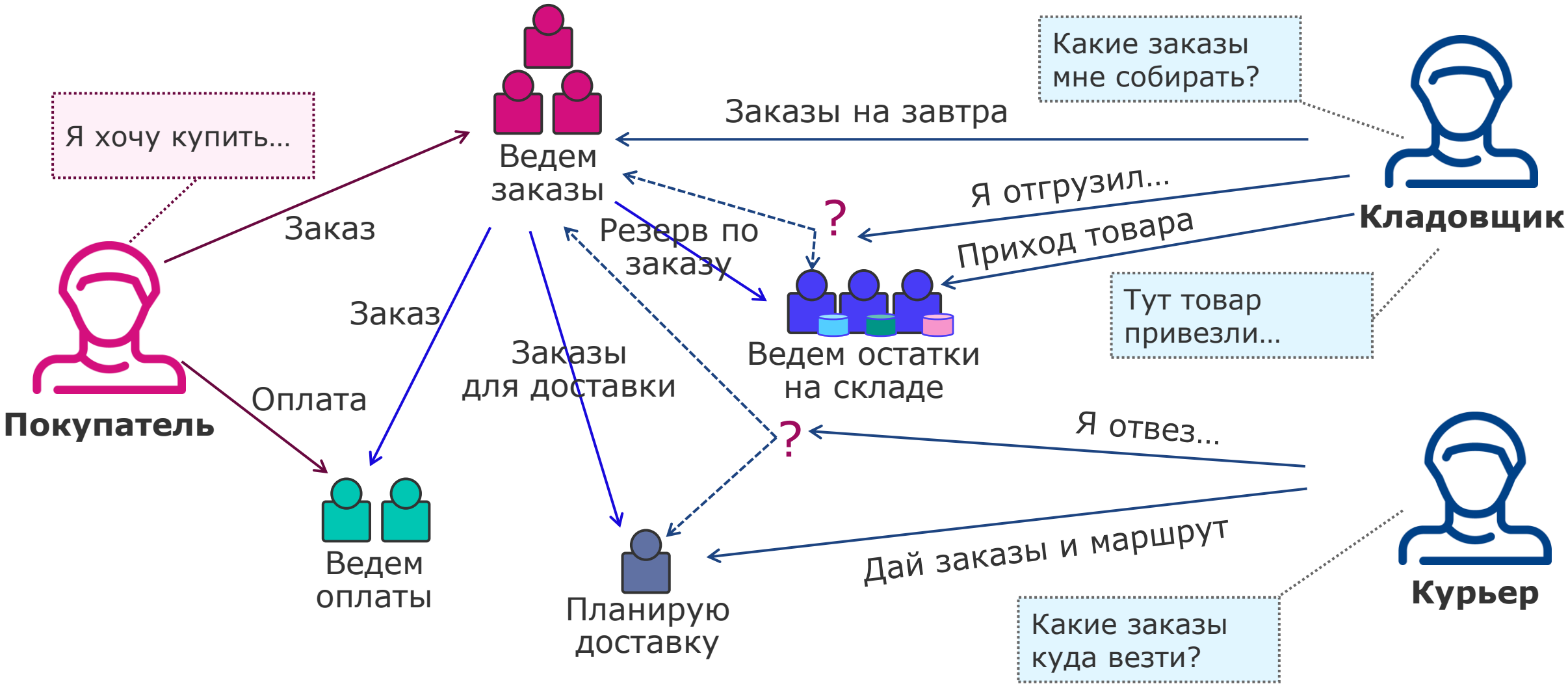

- Метафора: ЦОД — дома для гномиков, а ноды — комнаты
- Обращение в соседнее помещение — дольше или невозможно
- Надо три ноды или ЦОД, чтобы отличить пропажу связи от падения, в метафоре: соседний дом сгорел или телефон не работает

# Чек-лист проектирования

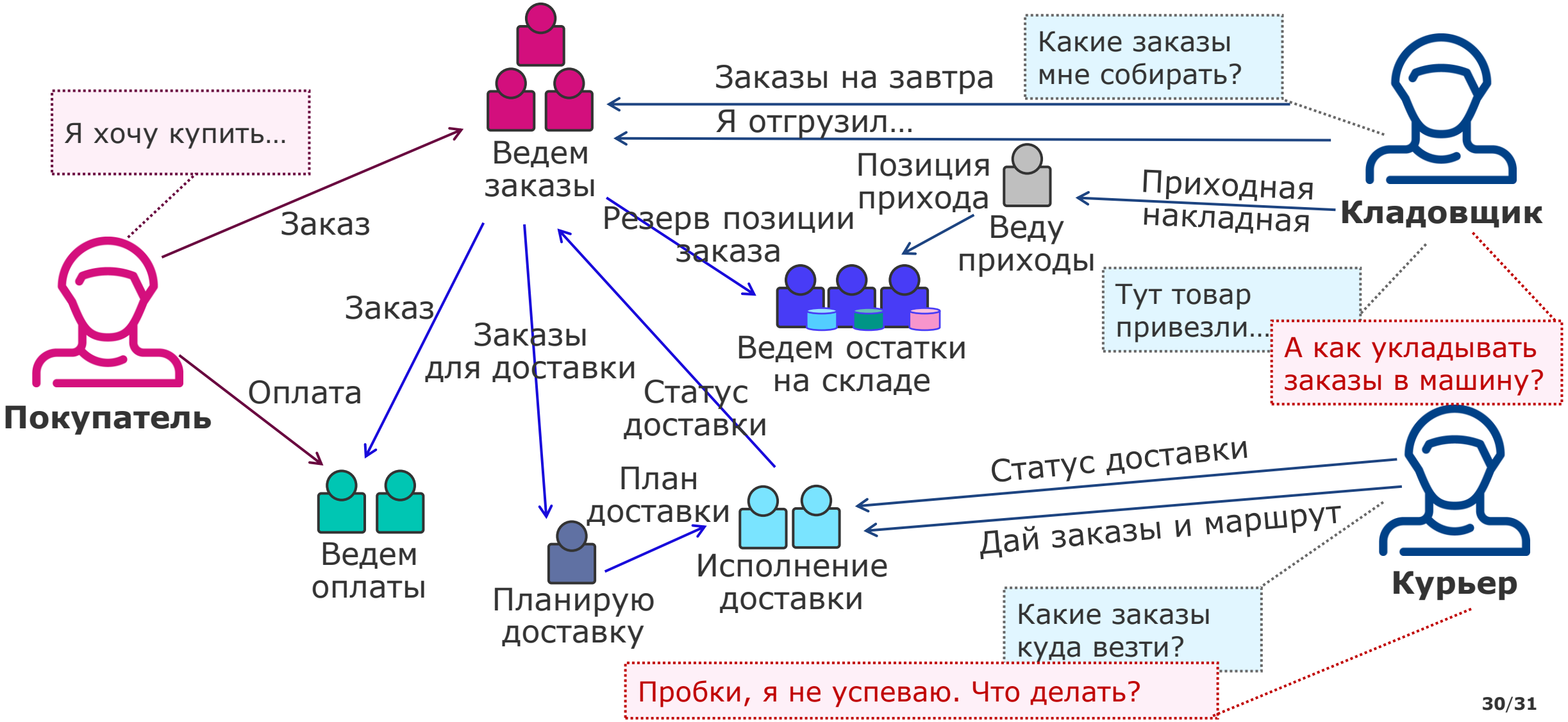
CUSTIS

- Как масштабируется каждый из сервисов под нагрузкой?
- Используется общая БД или отдельные?
- Где в БД возникают блокировки обработки запросов пользователей?
- Как взаимодействуют сервисы, где и какие очереди, что происходит с записями в очереди, когда экземпляры сервисов и ноды падают?
- Как обеспечивается устойчивость при падении экземпляров сервисов?
- Как обеспечивается устойчивость при падении нод и дата-центров?
- Как обеспечивается работа, когда пользователь едет в «Сапсане» или в другом месте с плохой связью, соединение рвется, и он перезагружает страницу?
- Какой мусор остается, если пользователь ушел, и как его чистят?

# Проектируем дальше...



# Проектируем дальше...



# И в заключение

CUSTIS

HL 2022 HighLoad++ FOUNDATION

- Мир изменился, и старые способы описания приложений не работают в современной архитектуре
- Метафора гномиков позволяет эффективно проектировать приложения в акторной модели и понятна не только разработчикам
- Но могут потребоваться и другие метафоры, ищите их

Максим Цепков

**http://mtsepkov.org**

**@MaximTsepkov**

На сайте много материалов по анализу и архитектуре, Agile, ведению проектов, управлению знаниями, мои доклады, статьи и конспекты книг

**Вакансии**

Пишите на hr@custis.ru, подходите с вопросами