

Skillbox

Работа с событиями. События

Андрей Гордиенков

Solution Architect

ABAX

В прошлом модуле

- Разобрались с Open API
- Узнали, что API — это тоже продукт

В этом модуле

- Такие разные события: что подразумевается под «событием» в архитектуре
- Детальный взгляд на Event Sourcing и CQRS
- Плюсы и минусы event-driven архитектуры

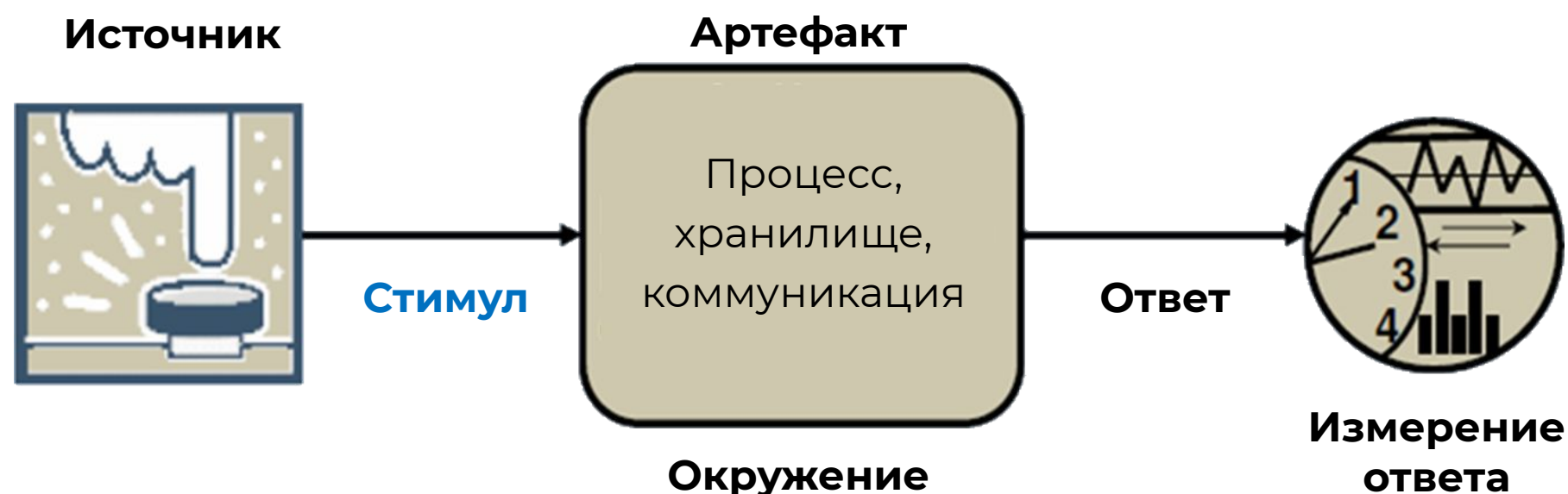
В этом уроке

- Концепт события
- Типы событий
- Применение событий

Что такое событие?

- То, что имеет место, происходит, наступает в произвольной точке пространства/времени
- То, что происходит в некоторый момент времени и рассматривается как изменение состояния мира и так далее

Что такое событие?



Информация о некотором стимуле, который приводит систему в новое состояние.

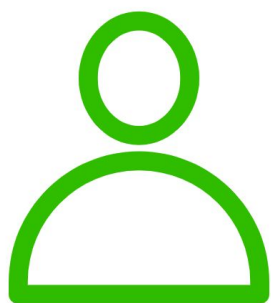
Обычная доменная модель

```
shoppingCart.Add(selectedItem);  
notification.Ping(user, msg);  
warehouse.Reserve(selectedItem);
```

Понятно, что делает код?

А что привело к его выполнению?

Источники изменений



Пользователь



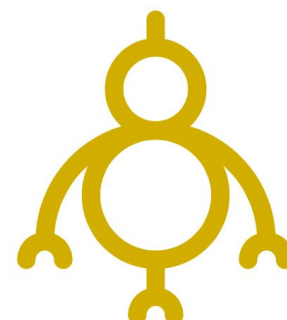
Первый
заказ



Запланированные
действия



Автоматическое
пополнение



Другие системы

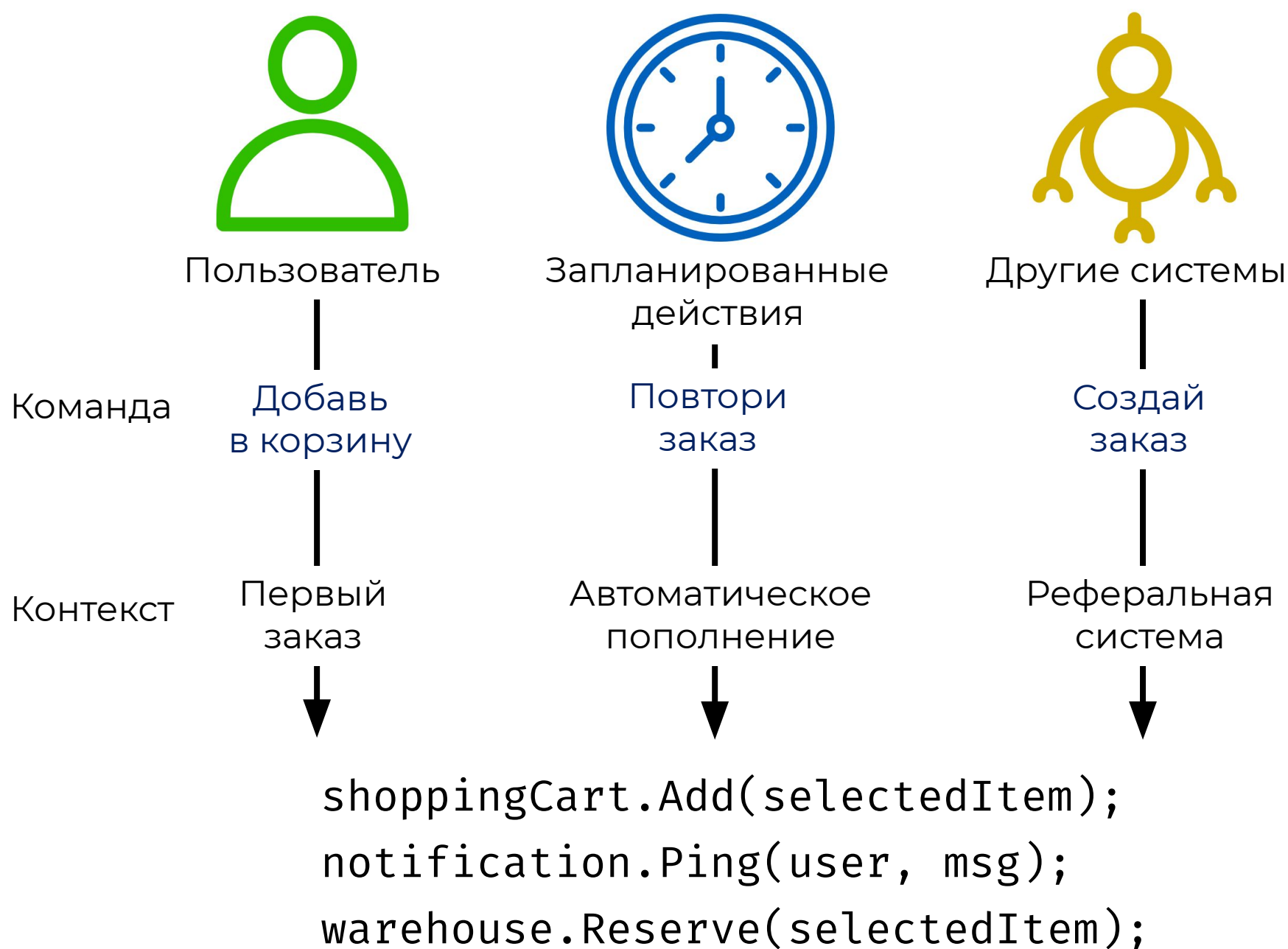


Реферальная
система



```
shoppingCart.Add(selectedItem);  
notification.Ping(user, msg);  
warehouse.Reserve(selectedItem);
```

Событие как команда



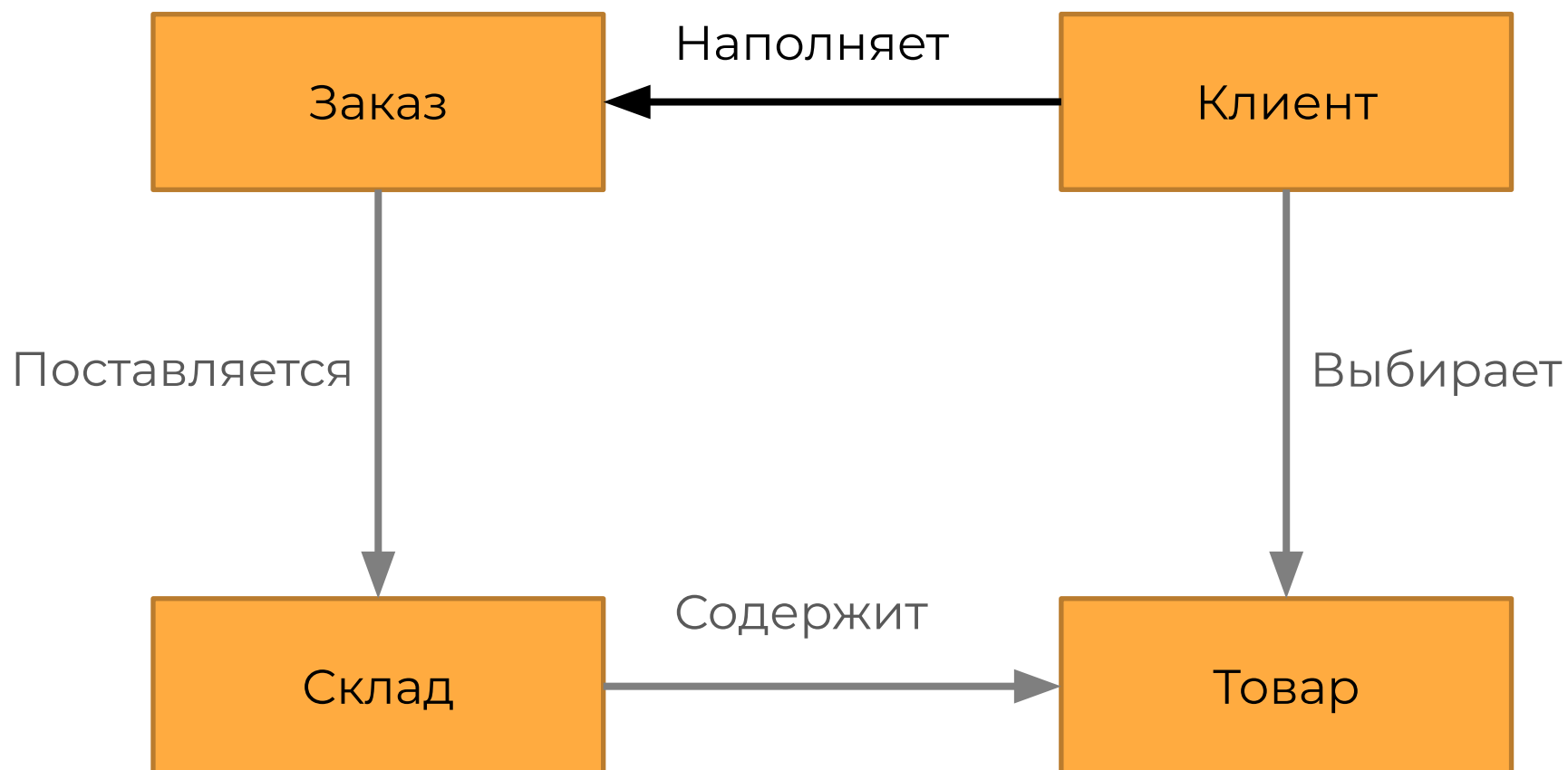
Событие как команда



Skillbox

Вся активность вокруг бизнес-процессов построена как реакция на некоторые события, которые происходят в произвольные моменты времени.

Будут ли события представлены явно или неявно в вашей системе?

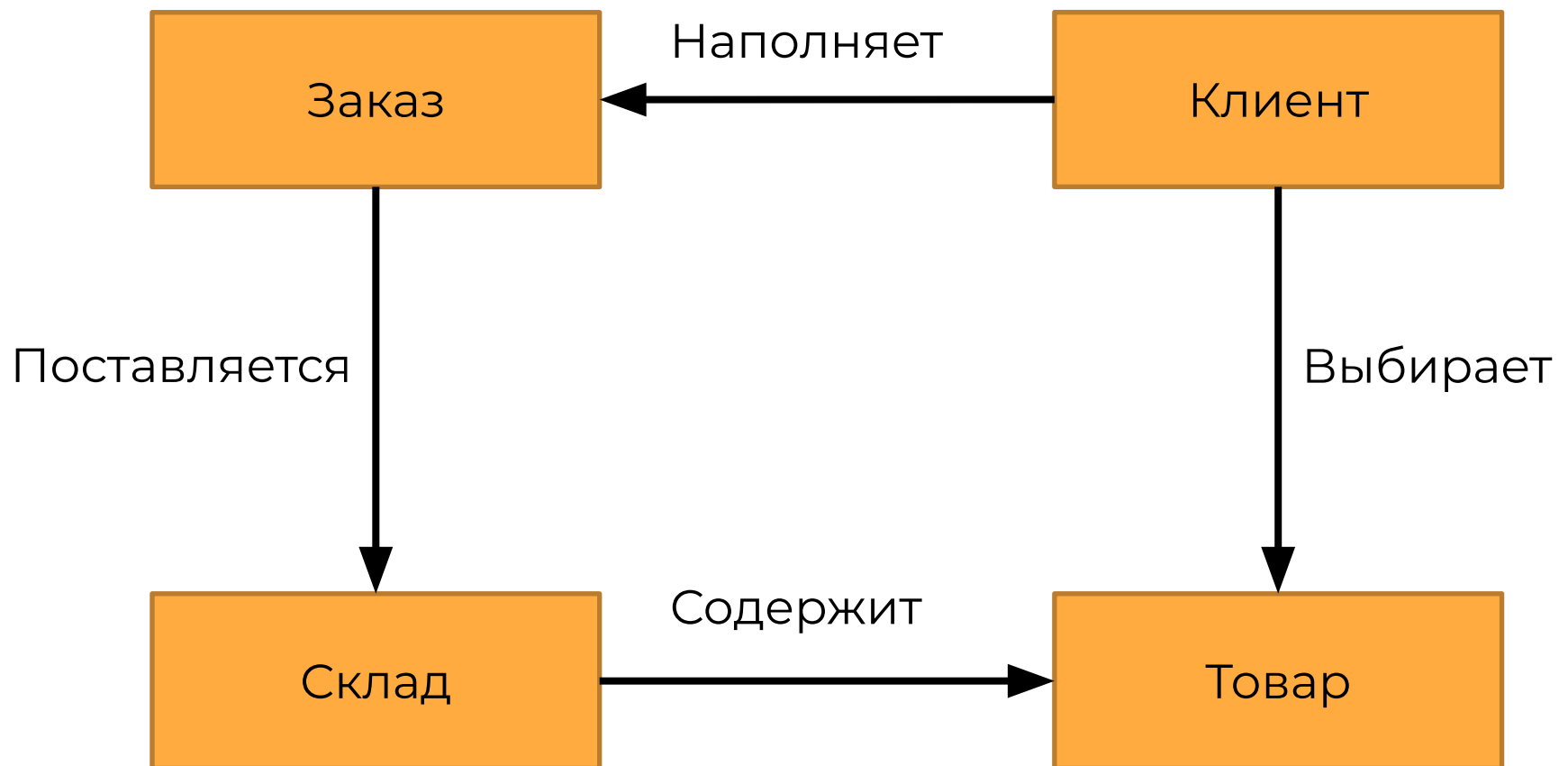


1

```
client.Select(goods);  
order = client.CreateOrder();
```

2

```
event = new ItemAddedToShoppingCart(client, item);  
broker.Send(event);
```

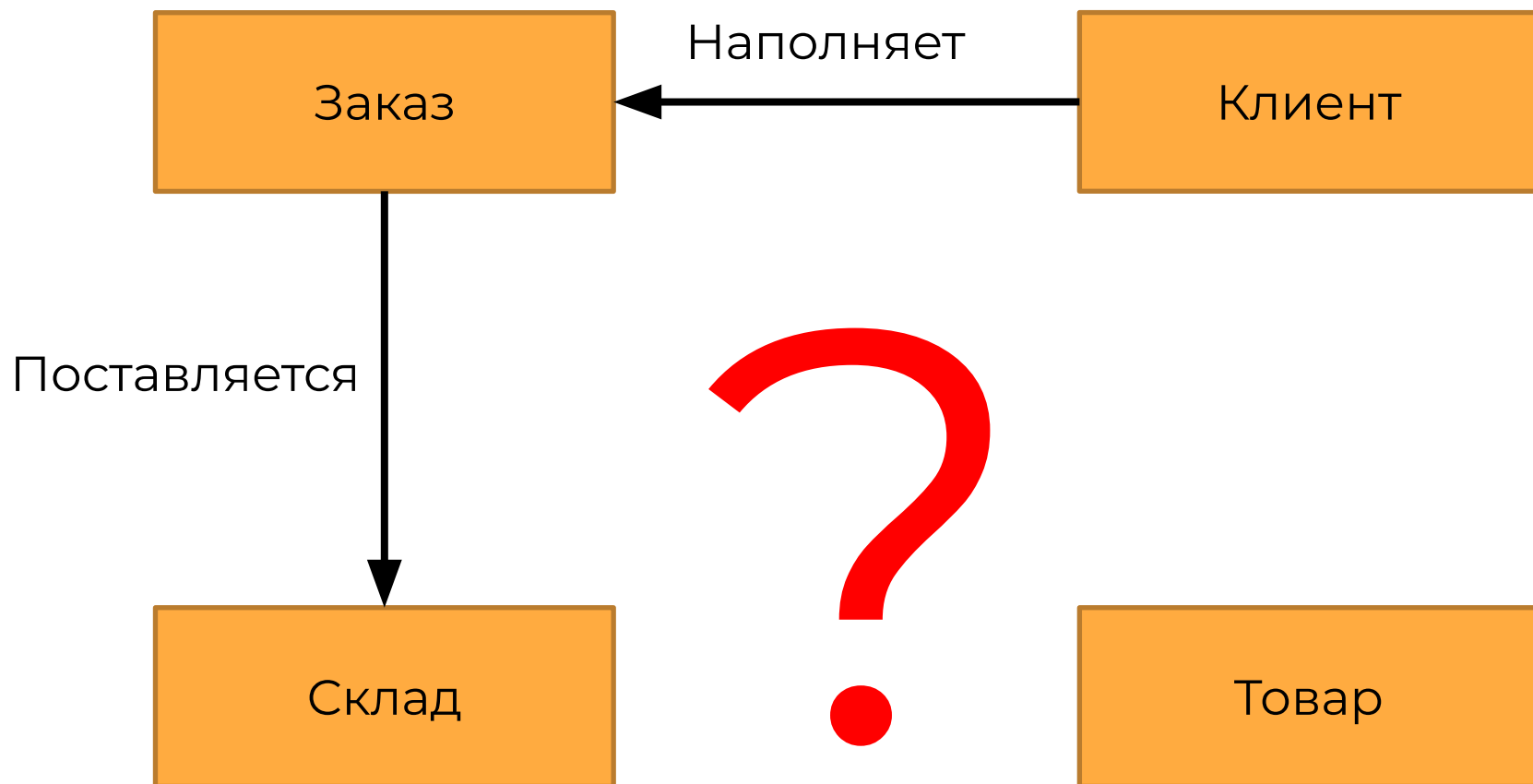


```
client.Select(goods);
```

```
order = client.CreateOrder();
```

```
order.Execute(store);
```

Skillbox

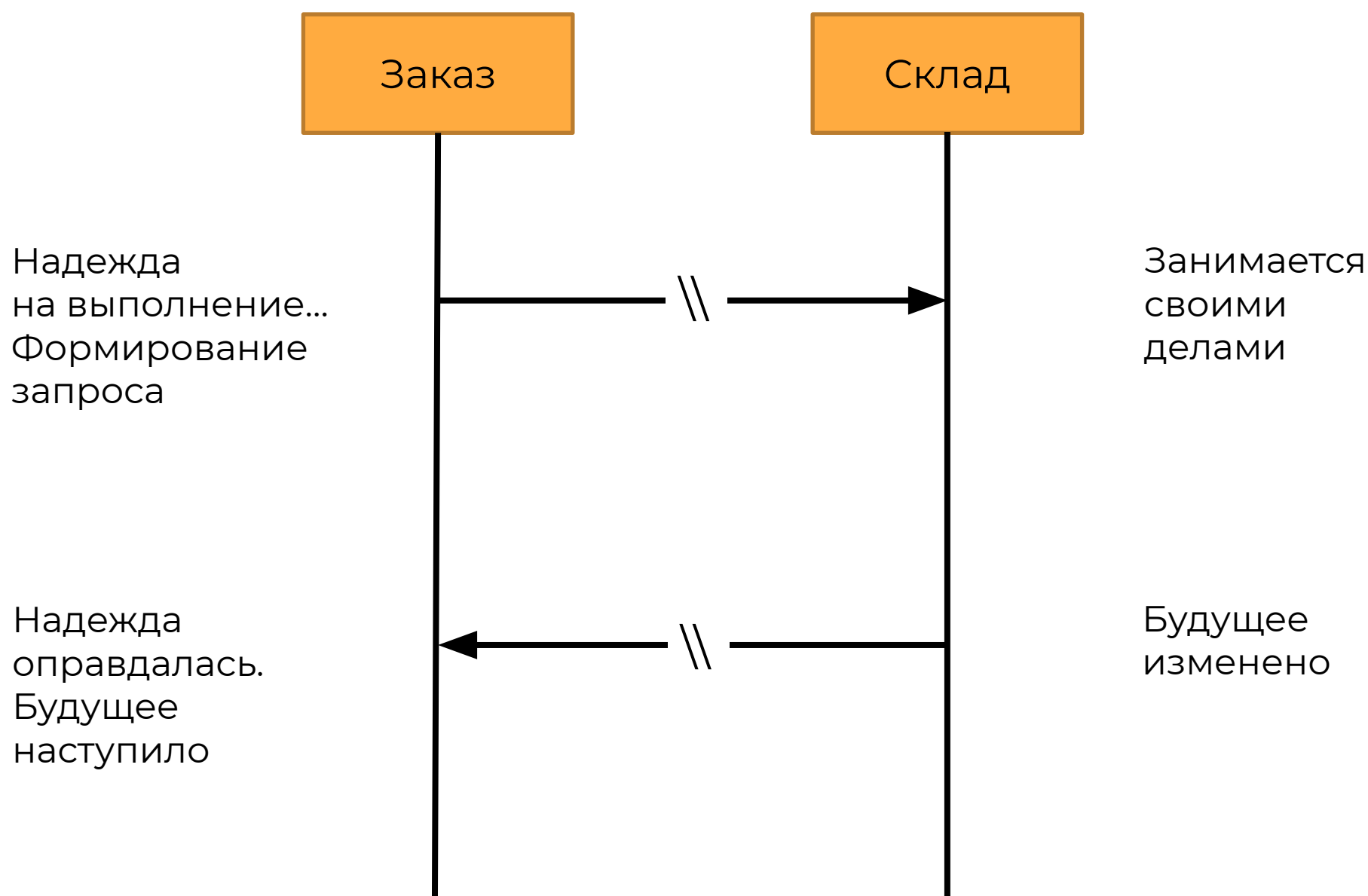


```
client.Select(goods);
```

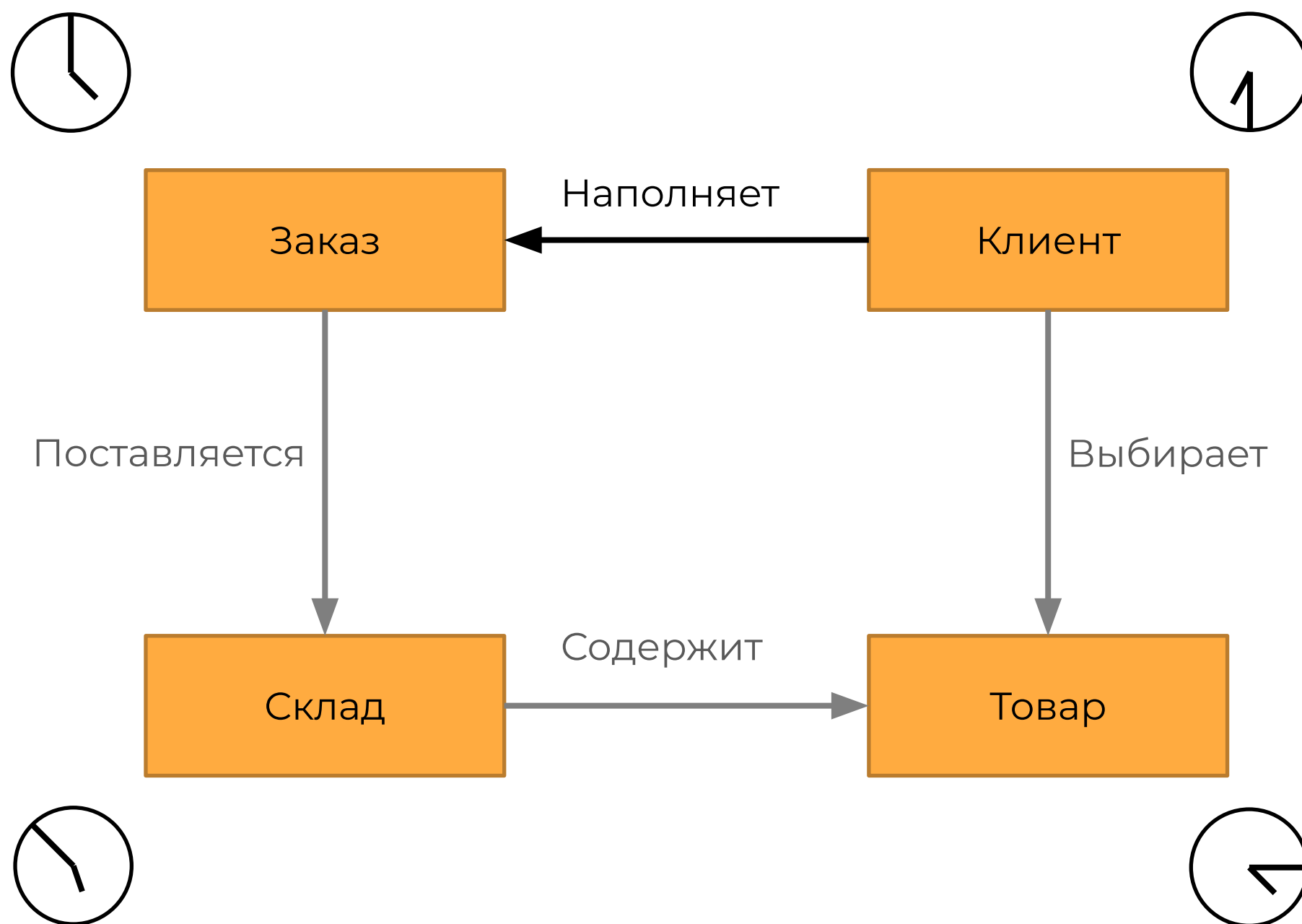
```
order.setClient(client);
```

```
order = new Order(client);
```

События и команды



События и команды

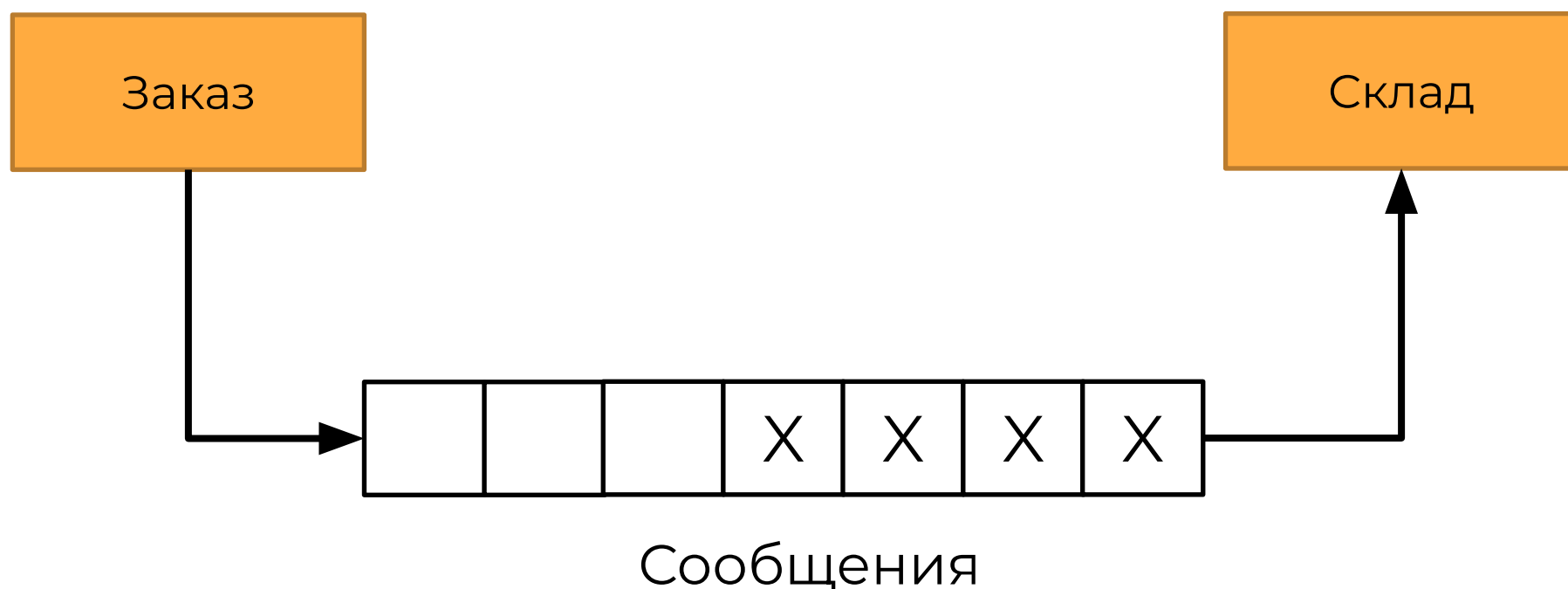


События и команды

Команда — запрос в «будущее», надежда, что оно наступит. Название всегда в повелительном наклонении.

Событие — сообщение о том, что «будущее» наступило, что-то где-то произошло. Событие не отменить, на него можно только реагировать или не реагировать. Название всегда в прошедшем времени.

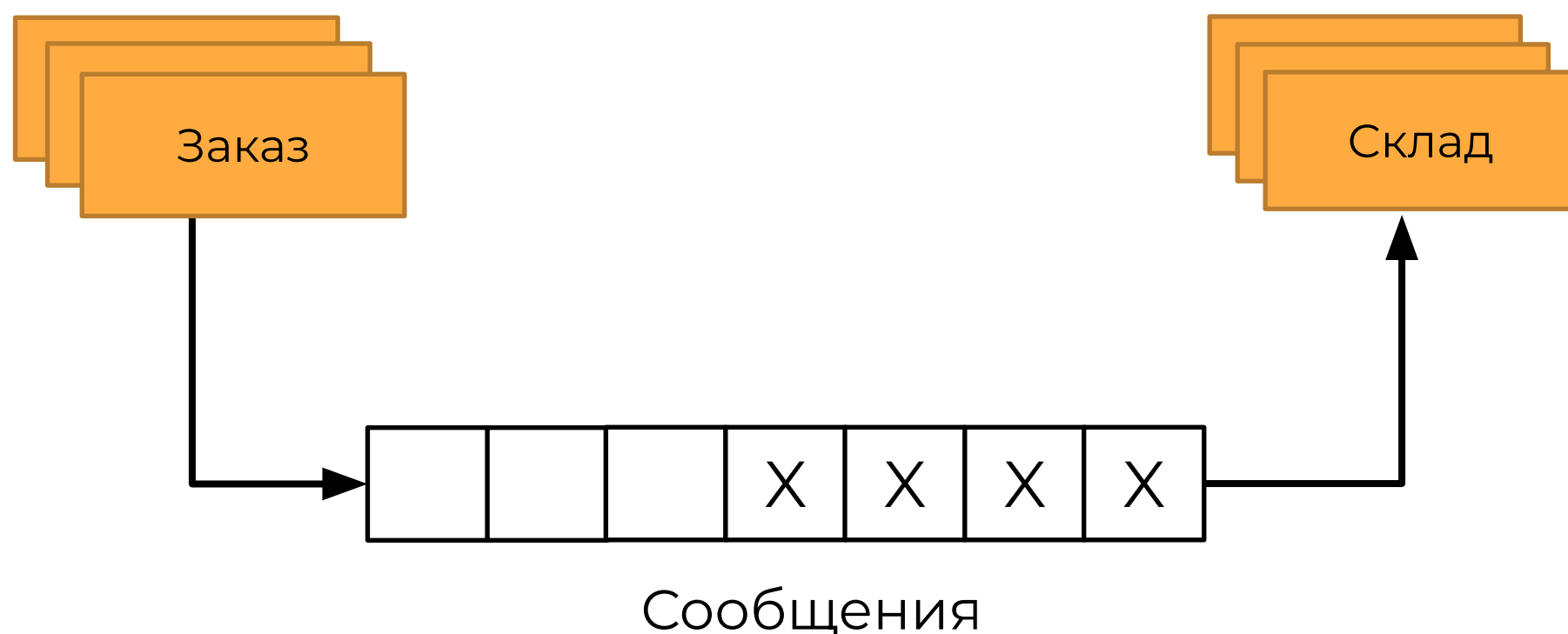
Очереди



Сообщения о том, что что-то произошло, могут быть командами или событиями.

Асинхронное взаимодействие между сервисами.

Очереди



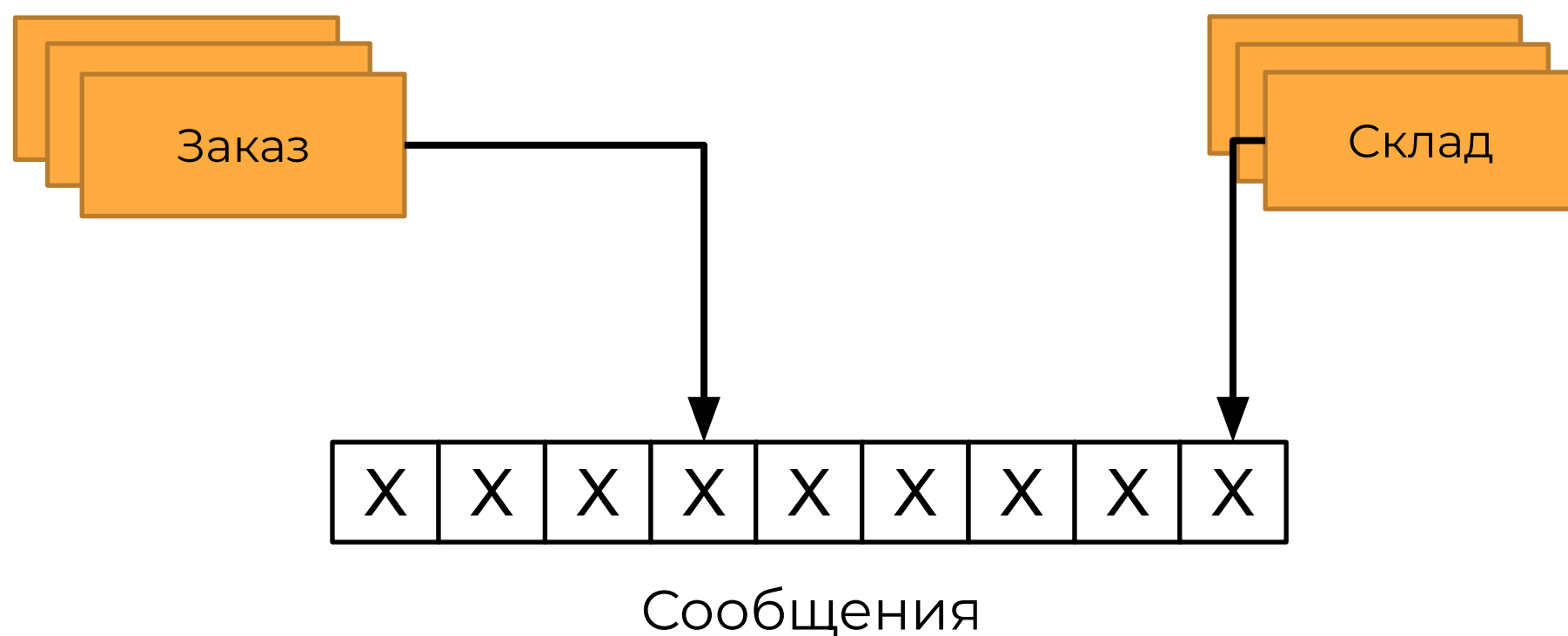
Сообщения о том, что что-то произошло, могут быть командами или событиями.

Асинхронное взаимодействие между сервисами.

Очереди



Логи



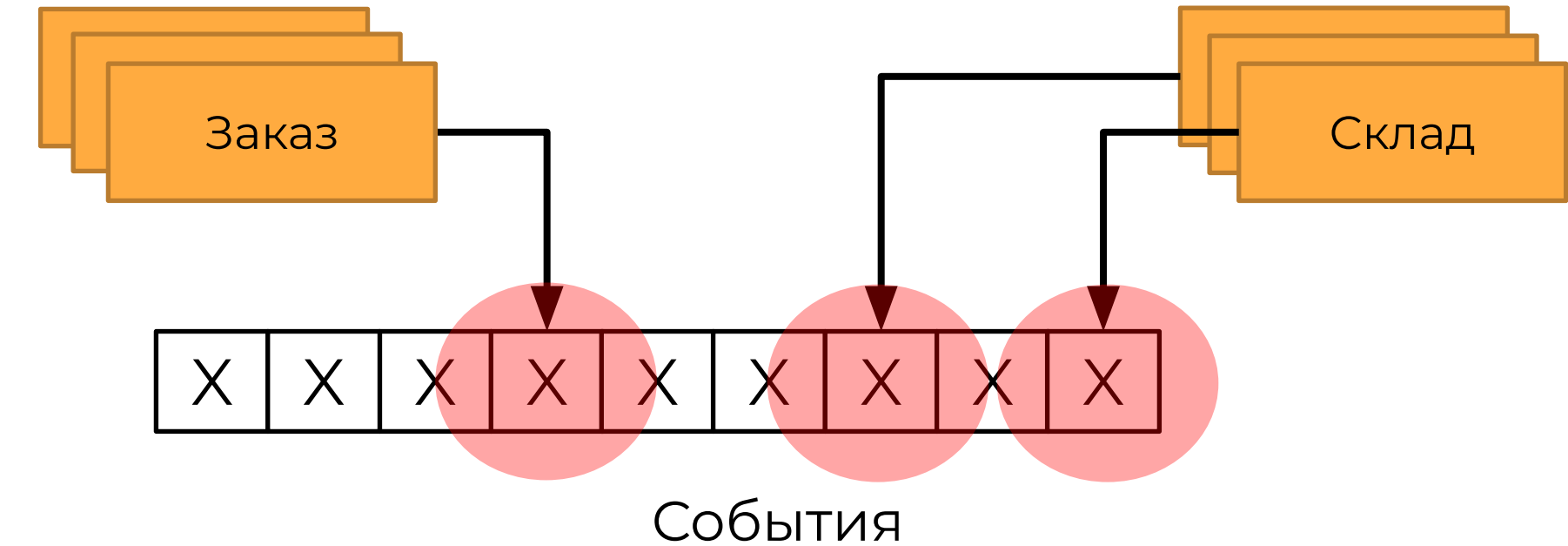
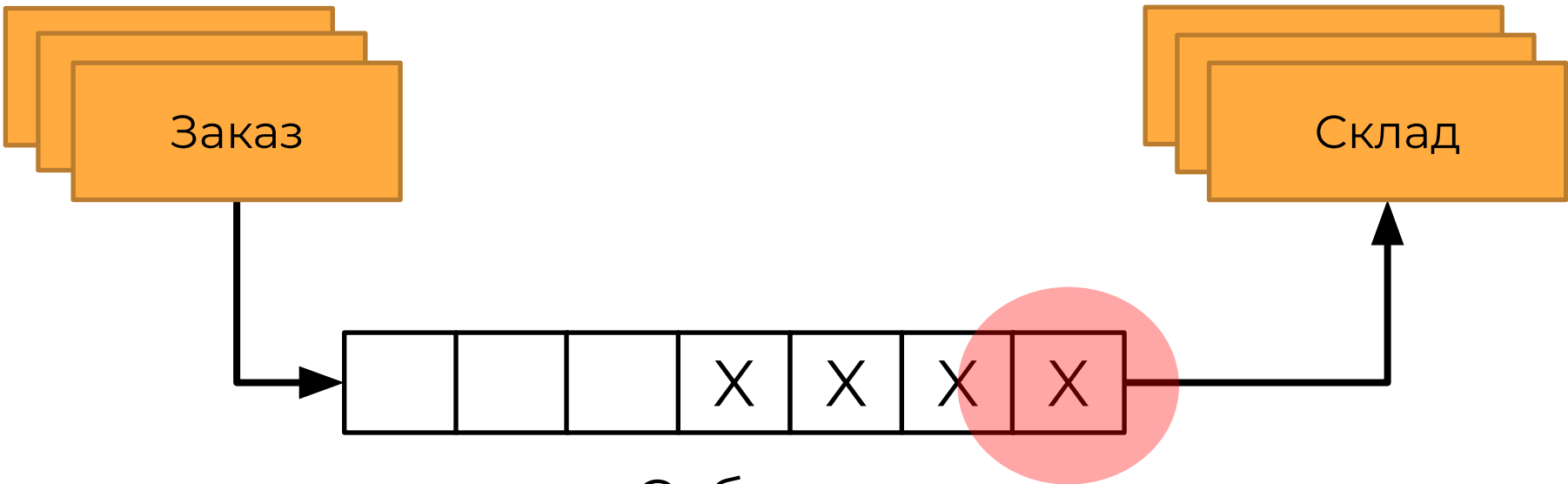
Сообщения — события о том, что что-то произошло.

Асинхронное взаимодействие между сервисами.

Логи



Очереди и логи



Очереди и логи

Очередь:

- доступ к первому элементу в очереди
- сообщения после обработки не сохраняются

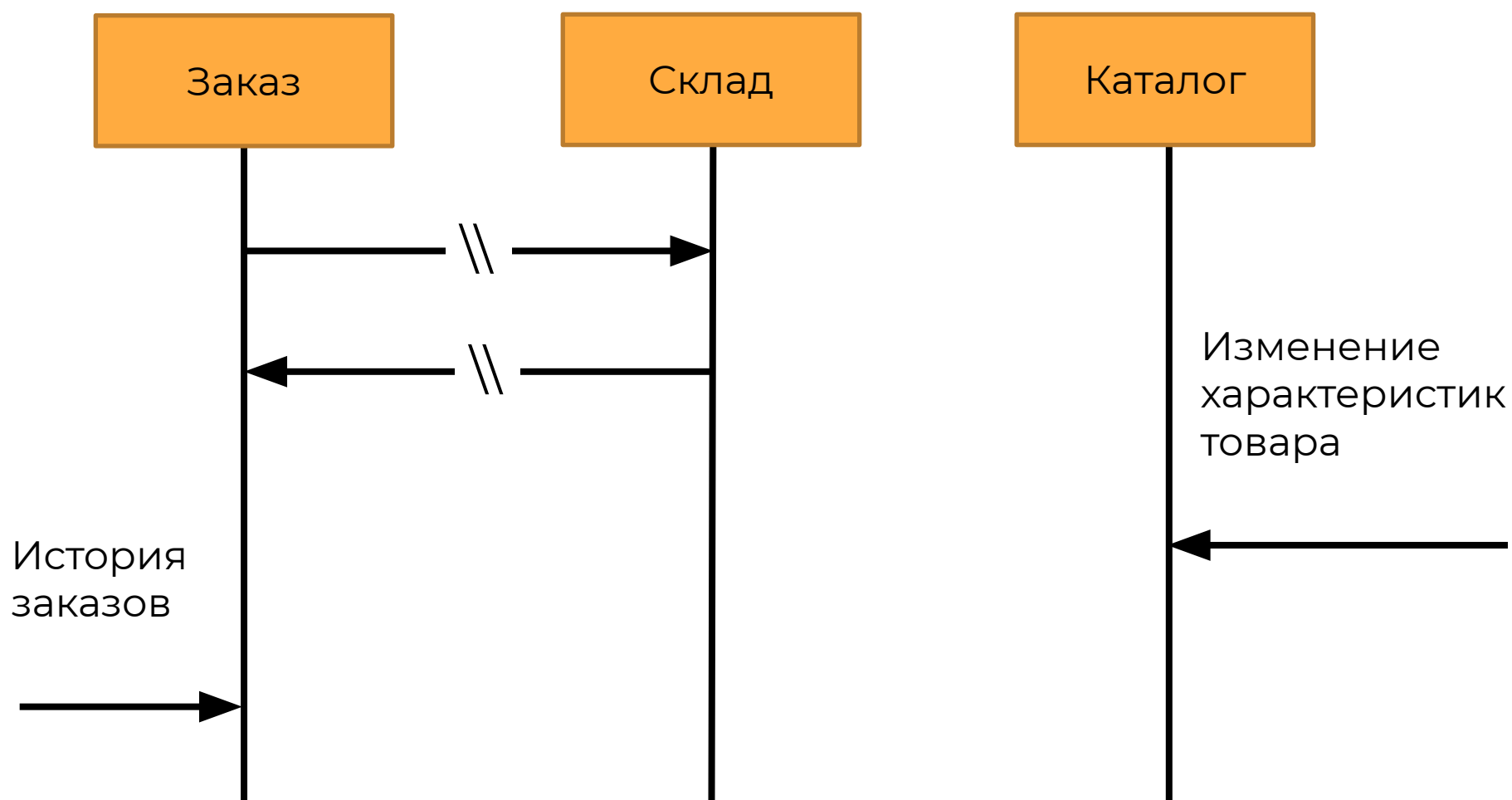
Лог:

- произвольный доступ к элементам
- сообщения сохраняются после обработки

Практическое правило:

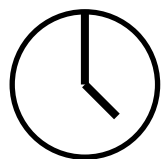
- очередь — для команд
- логи — для событий

Передача состояния

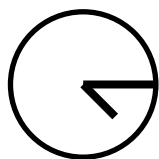


Как будет показан товар в истории заказов?

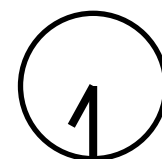
Передача состояния



Заказ



Склад



Каталог

Команда и событие должны содержать всю необходимую информацию для обработки.

Данные вне сервисов — неизменяемые.

Передача состояния

```
{  
  "товар" : 8f5395b9-d50b-4337-a66c-ab27439df0b0,  
  "количество": 10  
}
```

Передача состояния

```
{  
  "товар" : 8f5395b9-d50b-4337-a66c-ab27439df0b0,  
  "количество": 10  
}
```

```
{  
  "название_товара" : "полка",  
  "артикул_товара" : "П-034",  
  "количество": 10  
}
```

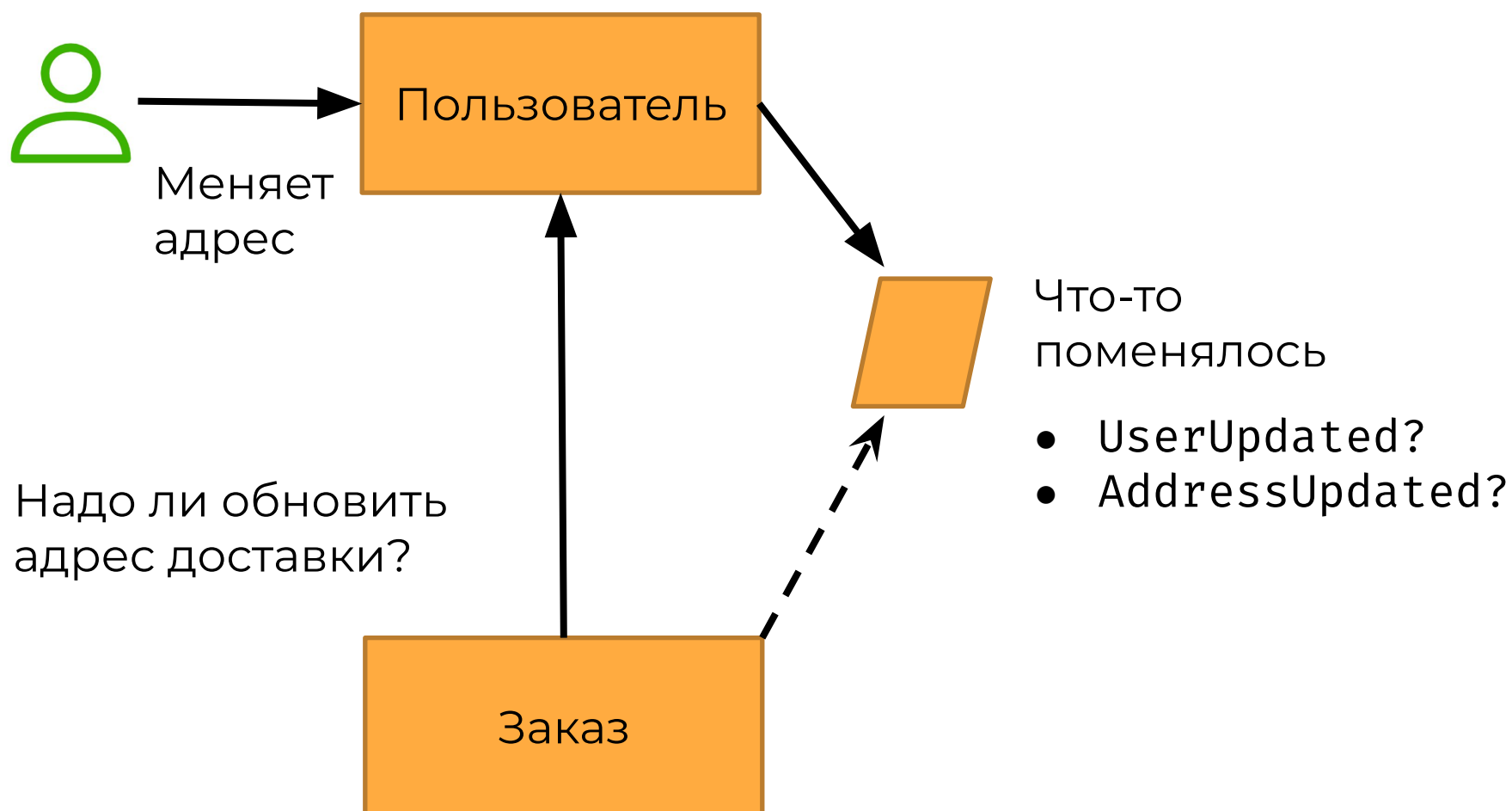
Передача состояния

Самое сложное — решить, какой уровень информации достаточен для повторной обработки.

Нюансы:

- размер сообщения
- изменение в субъектах
- изменение формата самого сообщения
- сетевая нагрузка на сервисы

Передача состояния



Выводы

- Реальные бизнес-процессы реагируют на события
- Команды — желание изменить будущее, события — информация о том, что произошло. Они должны быть легко различимы для разработчиков
- Сообщения — неизменяемы, но инициаторы и обработчики постоянно меняются
- Надо осознанно выбирать инструменты для обработки команд и событий
- События — это история, она не должна зависеть от текущего состояния субъектов события

Что дальше?

- Что такое Event Sourcing в деталях?
- Нюансы реализации

Skillbox

**Спасибо
за внимание!**