

Практическая работа №4. Связи между контекстами

Цель работы

Целью данной практической работы является изучение способов коммуникации между сервисами, а также получение навыков принятия решений о дальнейших способах разработки сервисов.

Теоретическое введение

При разработке информационных систем достаточно важно на ранних этапах решить, каким образом будет разрабатываться та или иная часть программного обеспечения. Как правило вариантов всего три:

- самостоятельная разработка сервиса;
- использование существующего сервиса;
- отдать разработку сервиса на аутсорс.

Каждый из вариантов может иметь какие-то вариации, например, можно при разработке использовать разных по опытности сотрудников или, беря готовый сервис, его можно дополнительно доработать, если он не полностью закрывает нужды.

Для микросервисной архитектуры такие решения принимать приходится чуть чаще в связи с тем, что система является не единым приложением, а набором разных подсистем.

Когда архитектура моделируется в применении предметно-ориентированного подхода, для принятия таких решений необходимо определить к какой подобласти относится каждый ограниченный контекст:

- смысловое ядро (core). К этой подобласти относятся наиболее важные сервисы организации, которые дают ей наибольшие конкурентные преимущества и выполняющие основные бизнес-процессы;
- служебная подобласть (supporting). Такие контексты имеют в себе определённую специализацию. Сюда относятся всё ещё важные аспекты бизнеса, но уже не относящиеся к смысловому ядру;

- неспециализированная подобласть (generic). Такая подобласть требуется для бизнеса в целом, а не для конкретной организации. Чаще всего тут используются вещи, которые можно переиспользовать.

Разбить на глаз контексты по типам – задача достаточно сложная и связанная со спорами и разногласиями в команде. Для упрощения принятия такого решения обычно используется оценка сервиса по двум параметрам: конкурентные преимущества и сложность/стоимость разработки.

Конкурентное преимущество – это совокупность различных факторов, которые придают организации уникальные характеристики, выгодно отличающие от конкурентов. Здесь могут учитываться разные факторы: инновационность подходов, снижение стоимости выполнения операций, ценность для пользователя, ценность для бизнеса и так далее. Как правило для такой оценки необходимо оперировать аналитикой.

Для оценки сервисов будет использоваться три категории:

- S – слабое влияние/низкая сложность реализации;
- M – среднее влияние/средняя сложность реализации;
- L – высокое влияние/высокая сложность реализации.

Пример оценки одного из сервисов представлен на Рисунке 4.1.



Рисунок 4.1 – Оценка Контекста управления оплатой

При анализе получившихся оценок используется достаточно простая шкала оценки:

1. Все сервисы, имеющие оценки (L,L) и (L,M), относят к смысловому ядру, так как они обеспечивают конкурентное преимущество, а также являются достаточно сложными в разработке. Такие сервисы будут разрабатываться организацией самостоятельно с привлечением наиболее опытных специалистов.
2. Сервисы, не имеющие чёткой специализации, относят к неспециализированной подобласти. Как правило оценки у таких сервисов (S,M) и (S,S). Такие сервисы будут иметь тот же функционал от организации к организации. Как правило в таких случаях можно купить уже готовое решение.
3. Оставшиеся сервисы можно отнести к служебной подобласти. Для их разработки можно использовать внешних подрядчиков, которые сделают сервис или отдать работу менее опытным разработчикам.

Стиль коммуникации

Стили коммуникации можно разбить на два типа: синхронный и асинхронный.

Синхронный стиль общения (он же блокирующий) используется тогда, когда в процессе выполнения какой-то функции необходимо получить информацию из другого сервиса. Пока информация не будет получена, выполнение функции не продолжится, а в случае неудачного исполнения запроса, функция не будет выполнена до конца. Для синхронного стиля могут быть использованы протоколы REST или RPC.

Асинхронный стиль общения является полной противоположностью синхронному. Его используют тогда, когда надо передать информацию другому сервису. В таком случае Сервис-инициатор общения может, не дожидаясь получения кем-то своего сообщения, продолжить свою работу. В

качестве асинхронного канала связи могут быть использованы различные брокеры сообщений, например, Apache Kafka или RabbitMQ.

Асинхронный стиль общения более предпочтительный, так как уменьшает зависимость сервисов друг от друга, повышая тем самым общую доступность системы. Благодаря этому при полном отказе одного из сервисов остальные могут продолжать корректно функционировать.

Определение отношений между контекстами

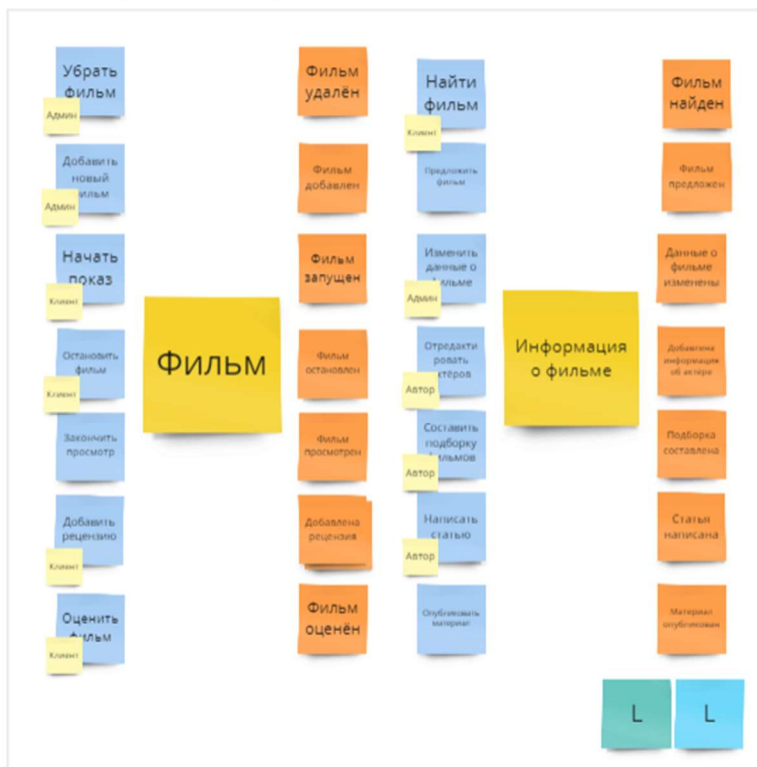
Для определения наличия коммуникаций между сервисами системы необходимо вновь исследовать все команды и события, которые были выделены на начальных этапах моделирования.

В процессе исследования команд необходимо задаться вопросом «Нужно ли для выполнения команды получить информацию из другого сервиса?» Если ответ на вопрос положительный, то в процессе выполнения команды возможно выполнение синхронного запроса к другому сервису.

При исследовании событий вопрос звучит немного по-другому: «Нужно ли после появления события выполнить какую-то другую команду?» В случае положительного ответа необходимо использовать асинхронный стиль общения.

В процессе определения стиля общения также необходимо разобраться с технологией, которая будет использоваться для общения сервисов. Пример смоделированного межпроцессного взаимодействия изображён на Рисунке 4.2.

Контекст управления фильмами



Контекст управления оплатой



Рисунок 4.2 – Межпроцессное взаимодействие в сервисах

Задание на самостоятельную работу

1. По теме Практической работы 3 оценить конкурентные преимущества и сложность каждого из сервисов. Сделать выводы.
2. Определить стили и способы взаимодействия между контекстами.
3. Определить и отобразить базы данных, используемые в полученной архитектуре.
4. Обновить глоссарий с терминами.