ОРГАНИЗАЦИЯ МИКРОСЕРВИСНОГО ПРИЛОЖЕНИЯ НА ПРИМЕРЕ ПЛАТЕЖНОЙ-ФИНАНСОВОЙ СИСТЕМЫ

Горбачевский М.В., Горбачевский К.В., студенты гр. 150504

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Куприянова Д.В. – ст. преподаватель

В данной работе рассматривается разработка микросервисного приложения, описываются основные аспекты реализации и анализируется использование микросервисов в платежно-финансовых системах.

Данная работа представляет собой разработку кабинета пользователя платежно-финансовой системы, состоящего из связанных сервисов. Каждый сервис предоставляет собственный API и может работать отдельно от других. Сервисы обмениваются информацией через брокера сообщений [1] – механизм асинхронного взаимодействия, основанный на посылке сообщений.

Если один из сервисов по какой-то причине перестанет функционировать, например, у него пропадет сеть, очередь продолжит хранить сообщения и, когда сервис вновь заработает, он обработает очередь. В такой парадигме сервисы не знают о существовании друг друга, они получают входные данные из брокера, изменяют свое состояние, генерируют выходные данные и отправляют их назад, схема взаимодействия изображена на рисунке 1. Такой подход обеспечивает минимальную связанность системы, что позволяет быстро вносить новый функционал.

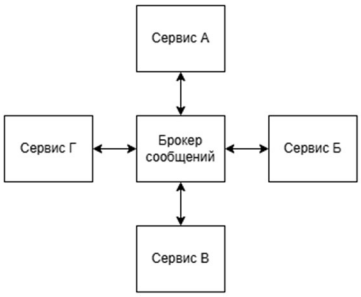

Рисунок 1 – Схема взаимодействия сервисов через посредника в виде брокера сообщений

Разделение сервисов позволяет реализовывать их на разных технологиях, использовать в них различные хранилища и располагать на разных машинах. Также это позволяет делить разработку и поддержку сервисов между разными командами, которые будут минимально зависеть друг от друга. Все это дает преимущества как бизнесу, чьи требования выполняются в срок, так и конечному клиенту, который быстро получает обновления.

В большинстве случаев, любое событие в системе затрагивает несколько сервисов, а значит необходимо как-то обеспечить атомарность операций, то есть система должна обладать транзакционными свойствами, в системе с множеством сервисов, такая транзакционность называется распределенной. Для решения данной задачи предложено использование алгоритмов типа Saga [2]. Суть этих алгоритмов заключается в том, что каждый сервис кроме непосредственного обработчика события, реализует еще и алгоритм его отката, таким образом, при конвейерной обработке события, ошибка на определенном этапе приведет к откату всех предыдущих.

В платежно-финансовых системах, например, в биржах, обычно существует множество разных по характеру возможностей, начиная от создания транзакций и просмотра котировок, заканчивая чатом с другими пользователями, также, например, платежи совершаются при помощи разных платежных систем и все это является причиной использования микросервисов. Использование микросервисной архитектуры позволит сделать систему более надежной, когда выход из строя одного сервиса не повлияет на работу других.

Использование сервисов под отдельные задачи в данной работе позволило отвязаться от языка программирования и СУБД, все сервисы могут быть реализованы на разных языках и хранить данные в разных СУБД.

Список использованных источников:

1. RabbitMQ [Электронный ресурс]. – Режим доступа: https://rabbitmq.com/. – Дата доступа: 07.04.2025.

2. Паттерн Сага [Электронный ресурс]. – Режим доступа: https://habr.com/ru/articles/427705/. – Дата доступа: 07.04.2025.