

ПРОЕКТИРОВАНИЕ WORKFLOW-СИСТЕМЫ НА ОСНОВЕ МИКРОСЕРВИСНОЙ АРХИТЕКТУРЫ

Гацко А.А

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

Научный руководитель: Тонкович И.Н. – канд.хим.наук, доцент, доцент кафедры ПИКС

Аннотация. Обоснована необходимость использования микросервисной архитектуры для разработки workflow-системы. Предложен вариант построения такой архитектуры. Выделены преимущества микросервисной архитектуры, включая гибкость масштабирования, непрерывность развертывания, ускорение выпуска новых функций на рынок.

Ключевые слова: микросервис, протокол, архитектура, отказоустойчивость, монолит

Введение. Workflow-системы являются критически важными для многих предприятий, поскольку они управляют потоком работ, связанных с выполнением бизнес-процессов, помогают упорядочить повседневные задачи и обеспечивают их эффективное выполнение. Однако создание и управление workflow-системами может оказаться сложной задачей из-за их модульности, комплексности процессов. Чем ближе размеры проекта к уровню enterprise, тем сложнее происходит добавление кастомизаций и переопределение функциональности. Обеспечить решение этой проблемы может микросервисная архитектура, разбивая систему на более мелкие и управляемые компоненты, тем самым снижая сложность и повышая гибкость.

Микросервисная архитектура приобрела огромную популярность в последние годы благодаря своей способности улучшать масштабируемость, ремонтпригодность и общую гибкость программных систем [1]. Она основана на декомпозиции системы на небольшие, независимые компоненты – микросервисы, которые могут разрабатываться, тестироваться и масштабироваться независимо друг от друга. Эти сервисы могут быть реализованы с использованием различных языков программирования и технологий хранения данных.

В данном исследовании рассматривается построение workflow-системы с использованием микросервисов. Необходимость применения данной архитектуры обусловлена следующими факторами:

- workflow-система работает достаточно эффективно до тех пор, пока она не вызывает проблем с масштабированием. Масштабирование отдельных компонент требует масштабирования всей системы, что занимает много времени и средств;
- deploy обновлений workflow-системы при переходе на новую версию происходит также слишком долго;
- изменяющиеся требования к системе и новые варианты использования требуют повторного развертывания всей системы;
- любые изменения в инфраструктуре приводят, как правило, к увеличению стоимости и временных затрат.

Основная часть. Микросервисная архитектура workflow-системы представлена на рисунке 1. Вначале были определены ключевые компоненты процессов workflow-системы. Затем для каждого компонента создавались независимые микросервисы со своими базами данных и API. Следует отметить, что микросервисы могут быть разработаны с использованием различного технологического стека, при условии, что все они придерживаются одного и того же контекста API. Такая гибкость позволяет выбирать наилучшую технологию для каждого компонента, что приводит к созданию более надежного и эффективного приложения.

Таким образом, мы можем разрабатывать, внедрять и масштабировать каждый компонент независимо, что, в свою очередь, повышает общую гибкость workflow-системы, позволяет эффективно справляться с высокими нагрузками.

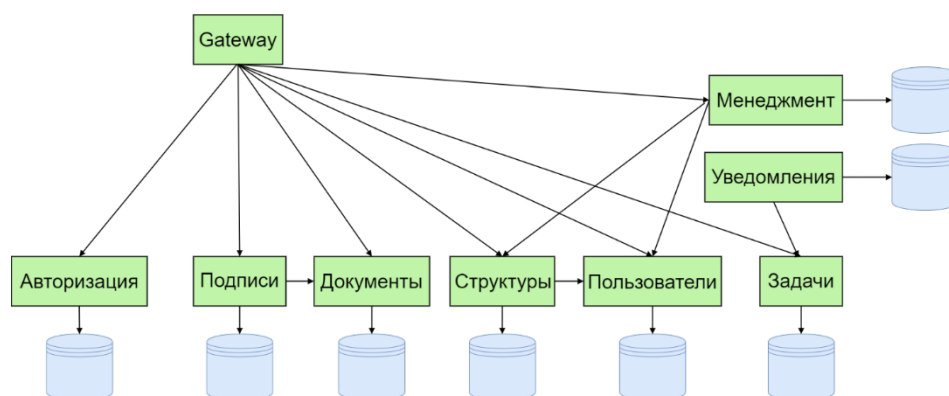


Рисунок 1 – Микросервисная архитектура workflow-системы

Например, если нагрузка на базу данных увеличивается, мы можем масштабировать только этот микросервис, не затрагивая другие компоненты.

Однако внедрение микросервисной архитектуры требует тщательного планирования и управления. Например, нужно убедиться, что каждый микросервис слабо связан и следует общему контракту API, чтобы обеспечить беспрепятственную связь между компонентами. Кроме того, такая архитектура должна иметь надежную систему тестирования и мониторинга для обеспечения надежности и производительности системы в целом.

Заключение. Чем ближе workflow-система к уровню enterprise, тем сложнее происходит масштабирование, добавление кастомизаций и переопределение функциональности. Обеспечивает решение этих проблем микросервисная архитектура. Микросервисы могут разрабатываться, тестироваться, внедряться и масштабироваться независимо друг от друга, что, в свою очередь, повышает общую гибкость workflow-системы.

Микросервисная архитектура дает ряд преимуществ, включая масштабируемость, гибкость в выборе технологий и в разработке, развертывании каждого компонента независимо друг от друга, а также ускорение выпуска новых функций системы на рынок. Однако успешная реализация микросервисной архитектуры требует тщательного планирования и управления, в том числе обеспечения того, чтобы каждый микросервис был слабо связан и придерживался общего контракта API.

Список литературы

1. Ричардсон, К. Микросервисы. Паттерны разработки и рефакторинга / Ричардсон К. – Питер, 2019. – С. 405–406.

UDC 004.415.28

WORKFLOW SYSTEM DESIGN BASED ON MICROSERVICE ARCHITECTURE

Hatsko A.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Tonkovich I.N. – PhD, associate professor, associate professor of the Department of ICSD

Annotation. The necessity of using microservice architecture for the development of a workflow system is substantiated. A variant of constructing such an architecture is proposed. The advantages of microservice architecture are highlighted, including scalability, continuity of deployment, acceleration of the release of new functions to the market.

Keywords: microservice, protocol, architecture, fault tolerance, monolith