

СРАВНИТЕЛЬНАЯ ХАРАКТЕРИСТИКА АРХИТЕКТУРЫ КОРПОРАТИВНЫХ ПРИЛОЖЕНИЙ

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

Белов А. А.

Киринович И. Ф. – доцент, к.ф.м.н.

Целью работы является сравнительная характеристика существующих подходов к построению инфраструктуры корпоративных приложений. Необходимость в интеграции корпоративных приложений обычно возникает, если они представляют собой обособленные модули. При достаточно большом количестве приложений осуществлять интеграцию между ними напрямую ресурсозатратно, поэтому возникли интеграционные платформы, которые закладываются в основу инфраструктуры корпоративных приложений. Первые интеграционные платформы строились на основе центрального брокера сообщений. Их сменили продукты класса ESB, некоторые из которых заменили центральный брокер сообщений на более гибкие структуры. Микросервисы представляют собой обособленные модули, предоставляющие клиентам интерфейсы взаимодействия с ними.

В таблице 1 приведено сравнение решений на базе ESB и микросервисов.

Таблица 1

Характеристика	ESB	Микросервисы
Время, затраченное на разработку	Меньше. Использование готовых инструментов и средств мониторинга, администрирования и проектирования, сокращающих время разработки.	Больше. Выбор инструментов и средств, обеспечение их совместной работы, тестирование взаимодействия между подсистемами, что увеличивает время на разработку.
Гибкость	Меньше. Ограничения платформы развёртывания, монолитность структуры внутренних сервисов, сложность изменения архитектуры или топологии сети.	Большая. Микросервисы представлены в виде отдельных модулей с собственным циклом разработки, не привязаны к конкретной технологии или платформе развёртывания.
Видимость работы внутренних компонентов и сервисов	Высокая. Платформа, как правило, имеет набор панелей и средств состояния системы и ее компонентов, а также средства по управлению ими.	Низкая. С ранних этапов разработки системы необходимо предусмотреть средства управления развёртыванием сервисов, обеспечить системой мониторинга.
Надёжность	Высокая. Архитектура обеспечивает взаимодействие между подсистемами путем обмена сообщениями через центральную шину, что уменьшает количество интерфейсов взаимодействия и, вследствие вероятности отказа.	Высокая гибкость микросервисов может привести к получению сложной и запутанной топологии сети. Надёжность конечного продукта зависит от архитектурных и инженерных подходов к разработке.
Стоимость разработки и поддержки системы	Стоимость разработки зависит от вендора платформы. Необходимость постоянного наличия ресурсов для поддержания системы в активном состоянии.	Как правило, меньше. Open source технологии, применение гибких планов использования ресурсов минимизирует затраты.

Анализ характеристик архитектур позволяет сделать вывод о сложности выбора в пользу одного либо другого из вышеперечисленных подходов. Использование преимуществ микросервисов может привести к получению сложной и ненадёжной системы. Однако, если заложить в основу подход с использованием корпоративной шины, и в дальнейшем выделять отдельные микросервисы и постепенно замещать ими компоненты платформы, то можно оценить разницу и планировать дальнейшее развитие системы.

Список использованных источников:

1. Microservices – a definition of this new architectural term [Электронный ресурс]. – Электронные данные. – Режим доступа <http://www.martinfowler.com/articles/microservices.html>
2. Newman, S Designing Fine-Grained Systems / S. Newman – USA: O'Reilly, 2015.
3. Oracle service bus documentation [Электронный ресурс]. – Электронные данные. – Режим доступа <https://docs.oracle.com/middleware/1213/osb/docs.htm>
4. Microservices, SOA, and APIs [Электронный ресурс]. – Электронные данные. – Режим доступа http://www.ibm.com/developerworks/websphere/library/techarticles/1601_clark-trs/1601_clark.html