

УДК 004.415.23

Сердюков Д.А.

студент

Научный руководитель: Зайцева Т.В., доц, к.т.н

Белгородский государственный национальный исследовательский университет, г. Белгород, Российская Федерация

ОБЗОР КОМПОНЕНТОВ ДЛЯ ПРОЕКТИРОВАНИЯ

АРХИТЕКТУРЫ

Аннотация. В статье рассматривается структура архитектурных компонентов современных информационных систем. Приводится классификация компонентов по уровням (Frontend, Backend, аппаратный уровень), а также описываются их характеристики и роль в проектировании архитектуры. Материал может использоваться в качестве чек-листа при разработке архитектурных схем.

Ключевые слова: архитектура информационных систем, компоненты, системный анализ, Frontend, Backend, микросервисы, API Gateway.

Serdyukov D.A.

student

Academic Advisor: Zaitseva T.V., PhD of Engineering Sciences,

Associate Professor

Belgorod State National Research University,

Russia, Belgorod

ОБЗОР КОМПОНЕНТОВ ДЛЯ ПРОЕКТИРОВАНИЯ

АРХИТЕКТУРЫ

Abstract. This paper presents a structured overview of components used in modern system architecture. A classification by functional level (Frontend,

Backend, Hardware) is provided, along with their core properties and typical use cases. The content can be used as a checklist during architectural design.

Ключевые слова: system architecture, components, systems analysis, Frontend, Backend, microservices, API Gateway.

Современные информационные системы строятся на принципах модульности, изоляции и четко определенных интерфейсов взаимодействия. Это обусловлено необходимостью обеспечения масштабируемости, управляемости и устойчивости архитектурных решений в условиях роста системной сложности и бизнес-требований.

Под компонентом в программной архитектуре обычно подразумевается программный модуль, реализующий некоторую функцию или набор функций и решает определенные подзадачи в рамках общих задач, решаемых системой [1]. Это логически и технологически изолированный модуль. Каждый компонент обладает собственной кодовой базой и конфигурацией, может быть развернут и масштабирован независимо от других, а взаимодействие между компонентами осуществляется через строго формализованные интерфейсы (протоколы, API, SDK и др.).

Проектирование архитектурной схемы предполагает идентификацию всех задействованных компонентов, указание их зон ответственности и способов связи. Отсутствие явно описанных элементов может привести к появлению «слепых зон» в архитектуре, что затрудняет сопровождение и развитие системы.

Фронтенд (англ. front-end) — это клиентская сторона пользовательского интерфейса к программно-аппаратной части сервиса. К технологиям разработки front-end относятся HTML, CSS, JavaScript, JQuery и фреймворки [2].

Компоненты Frontend-уровня включают:

- Веб-сайты, ориентированные на представление информации, получение заявок и работу с минимальным функционалом управления (например, через CMS). Эти решения ограничиваются функцией чтения данных, иногда с возможностью приема платежей или обратной связи.

- Мобильные приложения, разработанные нативно (iOS, Android) или с использованием кросс-платформенных технологий (Flutter, React Native). Они обладают доступом к аппаратным ресурсам устройства, возможностью офлайн кэширования данных и отправки push-уведомлений.

- Веб-приложения, которые могут быть как одностраничными, так и многостраничными. Они работают в браузере, могут быть связаны с отдельным сервером или включать в себя бэкенд (например, в случае монолитной архитектуры).

- Desktopные приложения, устанавливаемые на локальные устройства и имеющие доступ к файловой системе, оборудованию и могут функционировать без подключения к сети. Используются в специфических сферах: банкоматы, POS-терминалы, редакторы и пр.

- Встраиваемые виджеты, добавляющие функциональность во внешние интерфейсы. Такие компоненты реализуются через JavaScript или iframe и не требуют модификации основной системы.

Бэкенд (англ. back-end) — программно-аппаратная часть сервиса, отвечающая за функционирование его внутренней части [2].

Backend-компоненты структурируются следующим образом:

- Монолитное backend-приложение представляет собой единый программный блок, включающий все бизнес-логики и модули. Масштабируется только целиком. В рамках такой архитектуры, как правило, используется одна база данных.

– Сервисы — логически выделенный компонент системы, отдельное backend-приложение средних размеров с ограниченным набором бизнес-логики. Сервисы реализуют выделенные участки логики, разворачиваются независимо, могут использовать как общую базу данных, так и индивидуальную.

– Микросервисы — небольшие по объему автономные компоненты, каждый из которых отвечает за узкий бизнес-контекст. Они имеют собственную базу данных, полностью изолированы и независимо масштабируемы.

– API Gateway служит единой точкой входа в Backend-приложение архитектурой из сервисов или микросервисов. Маршрутизирует запросы между сервисами. Обеспечивает функции аутентификации и авторизации, проверять лимиты по запросам от клиентов, выполнять агрегацию и трансформацию ответов, кэшировать данные и собирать метрики для мониторинга.

– Базы данных могут быть реляционными (например, PostgreSQL, Oracle), NoSQL, time-series и др., и обеспечивают централизованное хранение данных.

– Файловое хранилище предназначено для долговременного хранения документов, изображений, видео и других мультимедийных данных.

– Брокеры сообщений (Kafka, RabbitMQ и др.) реализуют асинхронное взаимодействие между компонентами и используются в распределенных системах.

– Интеграционная шина (Enterprise Service Bus, ESB) представляет собой коммуникационный слой, объединяющий различные системы и реализующий сложную маршрутизацию и трансформацию данных. Информационные системы/приложения рассматриваются здесь как

поставщики и потребительских сервисов [3]. Используется преимущественно в архитектурах класса SOA.

– Внешние системы включают платёжные шлюзы, CRM, системы электронного документооборота и другие платформы, с которыми осуществляется интеграция.

Аппаратный уровень компонентов охватывает устройства ввода-вывода и сенсорные технологии:

- камеры видеонаблюдения, принтеры, терминалы;
- считыватели карт, микрофоны;
- сканеры RFID или штрихкодов;
- IoT-датчики (температура, влажность, движение и др.).

Подобная классификация может быть использована в качестве универсального справочника или контрольного перечня при построении архитектурной схемы. Особенно актуален такой подход в рамках использования нотации С4 или аналогичных визуальных методов моделирования, позволяющих точно обозначить зоны ответственности, интерфейсы и направления потоков данных между компонентами.

Используя этот перечень как чек-лист, системный аналитик или архитектор может убедиться, что в архитектурном описании не осталось «слепых зон». Это помогает в проектировании, разработке и поддержке сложных систем, обеспечивая их надежность, масштабируемость и удобство в эксплуатации.

Использованные источники:

1. Советов, Б. Я. Информационные технологии: теоретические основы : учебное пособие / Б. Я. Советов, В. В. Цехановский. — 2-е изд., стер. — Санкт-Петербург : Лань, 2022. — 444 с. — ISBN 978-5-8114-1912-

8. // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/209876> (дата обращения: 13.05.2025).

2. Маликова Зирек Турдалиевна, Кааров Ысмаил FRONT-END И BACK-END: РАЗЛИЧИЯ И ОСОБЕННОСТИ РАЗРАБОТКИ // Бюллетень науки и практики. 2022. №5. URL: <https://cyberleninka.ru/article/n/front-end-i-back-end-razlichiya-i-osobennosti-razrabotki> (дата обращения: 15.05.2025).

3. Морозова, О. А. Интеграция корпоративных информационных систем : учебное пособие / О. А. Морозова. — Москва : Финансовый университет, 2014. — 140 с. — ISBN 978-5-7942-1135-1. // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/152017> (дата обращения: 18.05.2025).