**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационные системы и технологии |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Разработка, сопровождение и обеспечение безопасности информационных систем |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Реферат**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **по дисциплине** |  | **Инструментальные средства информационных систем** | | |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | **Шипилов Евгений Александрович** |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | **ВБИо-206рсоб** |  |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | **Сибирев Иван Валерьевич** |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Тема:** Методы отображения моделей компонентов между уровнями архитектуры предприятия

**Введение**

Архитектура предприятия — это комплексная система, объединяющая технологические платформы, сервисы, модули и интерфейсы для достижения бизнес-целей. В условиях цифровой трансформации корректное отображение связей между компонентами различных уровней становится критически важным. Оно позволяет оптимизировать разработку, улучшить управление ресурсами и обеспечить прозрачность процессов. В данном реферате рассматриваются методы визуализации таких связей на примере компании, разрабатывающей SaaS-решение для автоматической обработки мультимедийного контента

**1. Уровни архитектуры предприятия**

Архитектура предприятия делится на четыре ключевых уровня, каждый из которых выполняет уникальные функции:

1. **Технологическая платформа** — базовый слой, включающий облачную инфраструктуру (AWS, Google Cloud), ML-кластеры (Kubernetes) и инструменты предобработки данных (FFmpeg).
2. **Сервисы** — функциональные блоки: AI-движки (Whisper, PyTorch), системы безопасности (OAuth 2.0, AES-256), интеграционные решения (Zoom API).
3. **Модули** — специализированные компоненты: транскрипция аудио, контроль качества, аналитика пользовательской активности.
4. **Интерфейсы** — точки взаимодействия с внешним миром: REST API, веб-приложения (React.js), мобильные приложения (Flutter).

**2. Методы визуализации связей**

**2.1. Слоистая модель (Layered Architecture)**

Слоистая модель демонстрирует вертикальную иерархию уровней. Например:

* **Технологическая платформа** → предоставляет ресурсы для **сервисов** (GPU-серверы для обучения AI).
* **Сервисы** → обеспечивают работу **модулей** (NLP-алгоритмы для транскрипции).
* **Модули** → взаимодействуют с **интерфейсами** (экспорт текста через веб-приложение).  
  **Преимущества:**
* Четкое разделение ответственности.
* Упрощение масштабирования (например, замена облачного провайдера не затрагивает сервисы).

**2.2. Диаграммы потоков данных (DFD)**

DFD фокусируется на горизонтальных взаимодействиях. Пример для SaaS-платформы:

1. Пользователь загружает аудио через **веб-интерфейс**.
2. Файл передается в **сервис предобработки** (конвертация в WAV).
3. Данные обрабатываются **AI-движком**.
4. Результат проверяется **модулем контроля качества**.
5. Текст экспортируется через **REST API**.  
   **Преимущества:**

* Наглядность последовательности операций.
* Выявление «узких мест» (например, задержек при конвертации).

**2.3. Матрица ответственности (RACI)**

RACI определяет роли компонентов в процессах. Пример:

|  |  |  |  |
| --- | --- | --- | --- |
| **Процесс** | **Технологическая платформа** | **Сервисы** | **Модули** |
| Обработка аудио | Responsible (AWS S3) | Accountable (AI) | Consulted (QC) |
| Интеграция с Zoom | Informed (DevOps) | Responsible (API) | — |

|  |
| --- |
|  |

**Преимущества:**

* Устранение дублирования функций.
* Повышение координации между командами.

**2.4. UML-диаграммы**

UML (Unified Modeling Language) используется для детализации технических взаимодействий:

* **Диаграмма компонентов:** REST API ↔ Сервис интеграций ↔ Внешние платформы.
* **Диаграмма последовательностей:** Запрос от клиента → Обработка в AI-движке → Возврат результата.  
  **Преимущества:**
* Стандартизация документации.
* Упрощение коммуникации между разработчиками и архитекторами.

**2.5. BPMN (Business Process Model and Notation)**

BPMN визуализирует бизнес-процессы. Пример:

1. **Старт:** Клиент загружает файл.
2. **Действия:** Конвертация → Транскрипция → Проверка.
3. **Исключения:** Ошибка обработки → Уведомление поддержки.
4. **Финиш:** Отправка текста клиенту.  
   **Преимущества:**

* Возможность симуляции процессов (например, расчет времени выполнения).
* Оптимизация workflow.

**3. Практическое применение методов**

Для SaaS-платформы, специализирующейся на обработке медиаконтента, методы визуализации применяются следующим образом:

**3.1. Слоистая модель**

* **Инфраструктура:** AWS EC2 + Kubernetes для оркестрации AI-моделей.
* **Сервисы:** FastAPI для интеграции с CRM-системами.
* **Модули:** Elasticsearch для аналитики пользовательского поведения.
* **Интерфейсы:** React.js для веб-панели управления.

**3.2. DFD для процесса транскрипции**

* **Уровень 0:** Общая схема: Загрузка → Обработка → Экспорт.
* **Уровень 1:** Детализация AI-движка: Распознавание речи → Суммаризация → Форматирование.

**3.3. BPMN для контроля качества**

* **Логика:** Если точность AI < 95%, транскрипция передается лингвистам.
* **Автоматизация:** Уведомления в Slack при необходимости ручной проверки.

**4. Вызовы и решения**

**4.1. Проблемы**

* **Сложность интеграции уровней:** Например, несовместимость форматов данных между модулями.
* **Риски безопасности:** Утечки данных при передаче между сервисами.

**4.2. Решения**

* **Стандартизация API:** Использование GraphQL для унификации запросов.
* **E2E-шифрование:** Внедрение AES-256 на всех этапах обработки.

**Заключение**

Отображение связей между уровнями архитектуры предприятия — неотъемлемая часть управления IT-системами. Применение слоистых моделей, DFD, UML и BPMN позволяет:

* Повысить прозрачность процессов.
* Ускорить устранение инцидентов.
* Обеспечить устойчивое развитие продукта.  
  На примере SaaS-платформы показано, как комбинация методов способствует созданию гибкой и масштабируемой архитектуры, отвечающей требованиям цифровой эпохи.

**Список литературы**

1. TOGAF 9.2 — The Open Group Architecture Framework.
2. Fowler M. Patterns of Enterprise Application Architecture. — 2002.
3. ГОСТ Р 57193-2016 «Архитектура предприятия».
4. Розански Н., Вудс Э. Архитектура программных систем. — 2015.