

АРХИТЕКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Программное обеспечение
Джим («Jim»)

Аннотация

В документе приведено описание архитектуры программного обеспечения Джим («Jim», далее – Система).

Содержание

Аннотация	2
Содержание	3
Термины и определения.....	4
1 Архитектура приложения	5
1.1 Компонент «Ядро»	6
1.2 Компонент «Коннекторы»	6
1.3 Компонент «API»	6
1.4 Компонент «МНП».....	7
1.5 Компонент «БД»	7

Термины и определения

В документе используются термины и определения, приведенные в таблице 1.

Таблица 1 – Термины и определения

Термин	Определение
Система	ПО Джим («Jim»)
БД	База данных
МНП	Минимальный набор прав
API	Application Programming Interface

1 Архитектура приложения

Архитектура Системы является распределенной и состоит из нескольких компонентов:

1. Ядро,
2. МНП,
3. Коннекторы,
4. API,
5. БД.

На Рисунок 1 представлена схема архитектуры Системы.

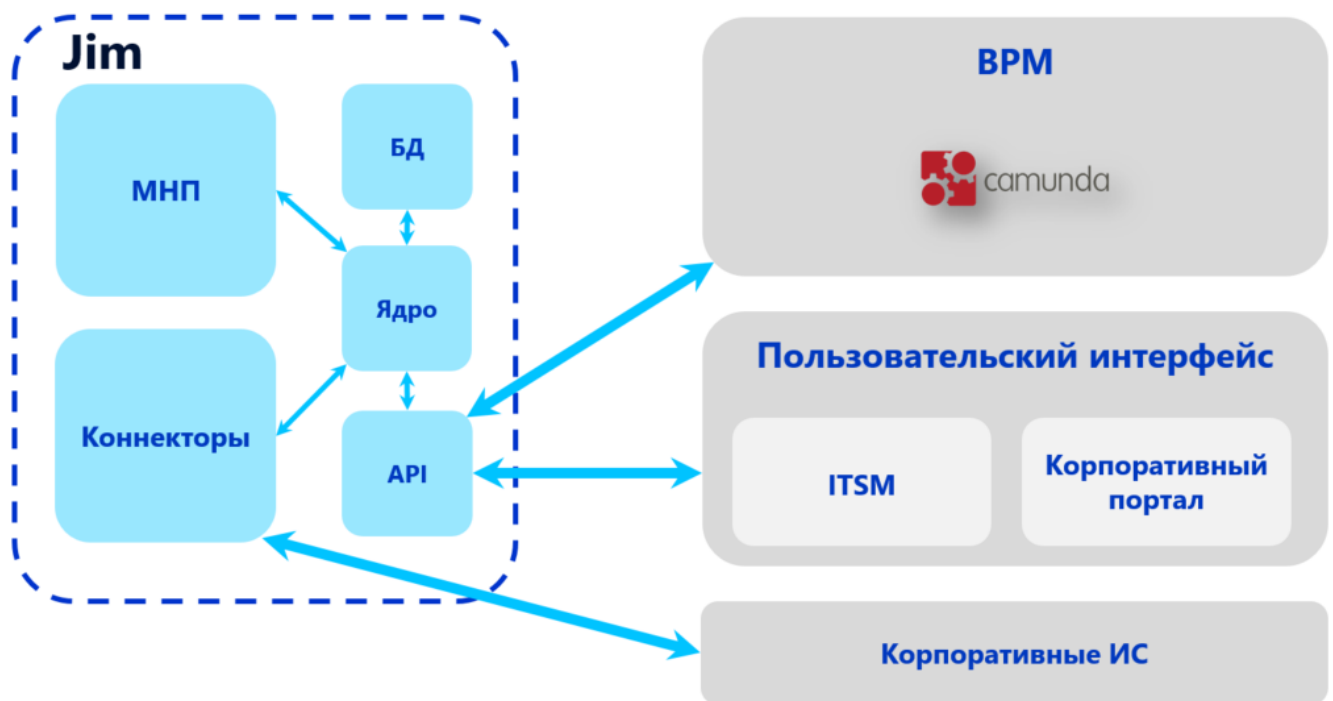


Рисунок 1 Архитектура Системы

1.1 Компонент «Ядро»

Это центральный компонент системы, реализующий функционал работы с объектной моделью и оркестровки других компонентов. Представляет собой java-приложение, реализующее:

- объектный слой системы - представление объектов БД в виде Java-классов,
- функционал работы с БД (ORM),
- функционал исполнения задач по расписанию (на базе Quartz),
- функционал авторизации,
- функционал Event Handler-ов.

Ядро поддерживает работу в составе нескольких экземпляров, которым могут быть присвоены функциональные роли. Таким образом обеспечивается кластеризация уровня приложения.

1.2 Компонент «Коннекторы»

Компонент «Коннекторы» обеспечивает синхронизацию данных между Системой и подключенными информационными системами: получает задачи по управлению правами доступа через вызовы со стороны Ядра и обеспечивает их выполнение, взаимодействуя с подключенными ИС при помощи выбранных механизмов взаимодействия.

Коннекторы реализуются на основе стандарта ConnID и исполняются под управлением Сервера коннекторов (Connector Server).

1.3 Компонент «API»

Это программный интерфейс (REST), обеспечивающий подключение к Системе других внешних систем для получения ими данных из Системы или выполнения операций над объектами Системы.

В качестве таких внешних систем могут выступать, например:

- ITSM-система или корпоративный портал, которые в разрабатываемом решении на базе ПО будут выполнять роль пользовательского интерфейса для создания и согласования заявок.
- платформа Camunda для моделирования и автоматизации бизнес-процессов управления правами доступа.

Авторизация при вызове API реализуется внутренней ролевой моделью Системы.

1.4 Компонент «МНП»

Это механизм, позволяющий настраивать правила, в соответствии с которыми пользователям автоматически назначаются роли в Системе.

Правила формулируются на языке SQL и возвращают набор пар пользователь-роль. Частью компонента также является механизм пересчёта правил, состоящий из очереди пересчёта и задачи пересчёта, которая исполняется по расписанию и обрабатывает очередь. Интерфейсом данного компонента является «матрица доступа» – таблица пар пользователь-роль, определяющая каким пользователям какие роли должны быть назначены в данный момент времени. При изменении состава данной таблицы срабатывают Event Handler-ы, которые приводят полномочия пользователей к текущему состоянию матрицы путём назначения или отзыва ролей.

1.5 Компонент «БД»

БД представляет собой сервер хранения реляционных данных PostgreSQL. Она обеспечивает хранение данных и метаданных, необходимых для функционирования системы. Для БД может быть настроена кластерная конфигурация стандартными средствами.

Модель данных является расширяемой, в рамках внедрения продукта в БД могут быть добавлены таблицы или столбцы.