

octoshell v.2

промежуточный отчёт

План

1. Сверка с ТЗ.
2. Анализ задачи.
3. Подробнее о Rails Engines.
4. Описание структуры octoshell v.2
5. Пример создания плагина.
6. Планы на дальнейшую работу.

Сверка с ТЗ

1 этап (март 2014)

Доработка текущей версии: приоритет 1,
приоритет 2.

2 этап (апрель 2014)

Старт проектирования конструктора.

Сверка с ТЗ

3 этап (май 2014)

Альфа версия конструктора, прототипы плагинов.

4 этап (июнь 2014)

Реализация плагинов для абстрактного университета.

Создание передаваемого продукта, передача на тестирование в институты.

Что реализовано

1. Доработана текущая версия: первый приоритет.
2. Спроектирован и реализован конструктор.
3. Реализованы и подготовлены к передаче плагины для абстрактного университета.

Анализ задачи

1. Найти способ единообразно создавать плагины.
2. Требуется соблюдать принятую схему разработки.
3. Способ должен иметь большое распространение и обширную документацию.

В результате был выбран вариант реализации плагинов с помощью Rails Engines.

Rails Engines

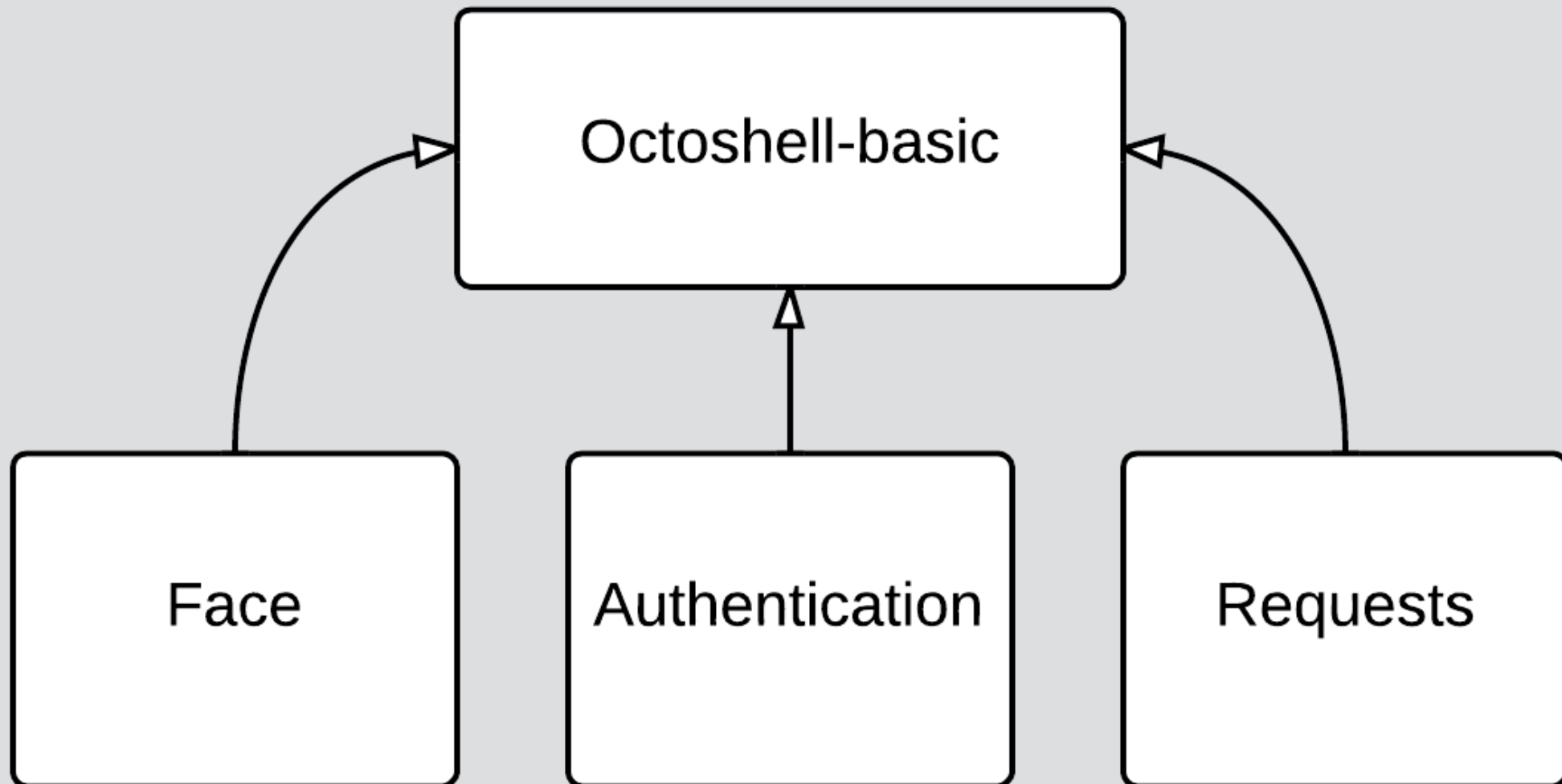
1. Лежит в основе любого Rails-приложения.
2. Существует исчерпывающая документация и большое количество примеров реализации с открытым исходным кодом.
3. Имеется строгая регламентация структуры приложения.
4. Существуют генераторы составных частей Rails Engines.

Этапы работы с Rails Engines

1. За основу берётся Rails-приложение.
2. Engine указывается в списке библиотек базового приложения (Gemfile).
3. Устанавливается engine.
4. Базовое приложение настраивается для работы с engine.

octoshell-basic

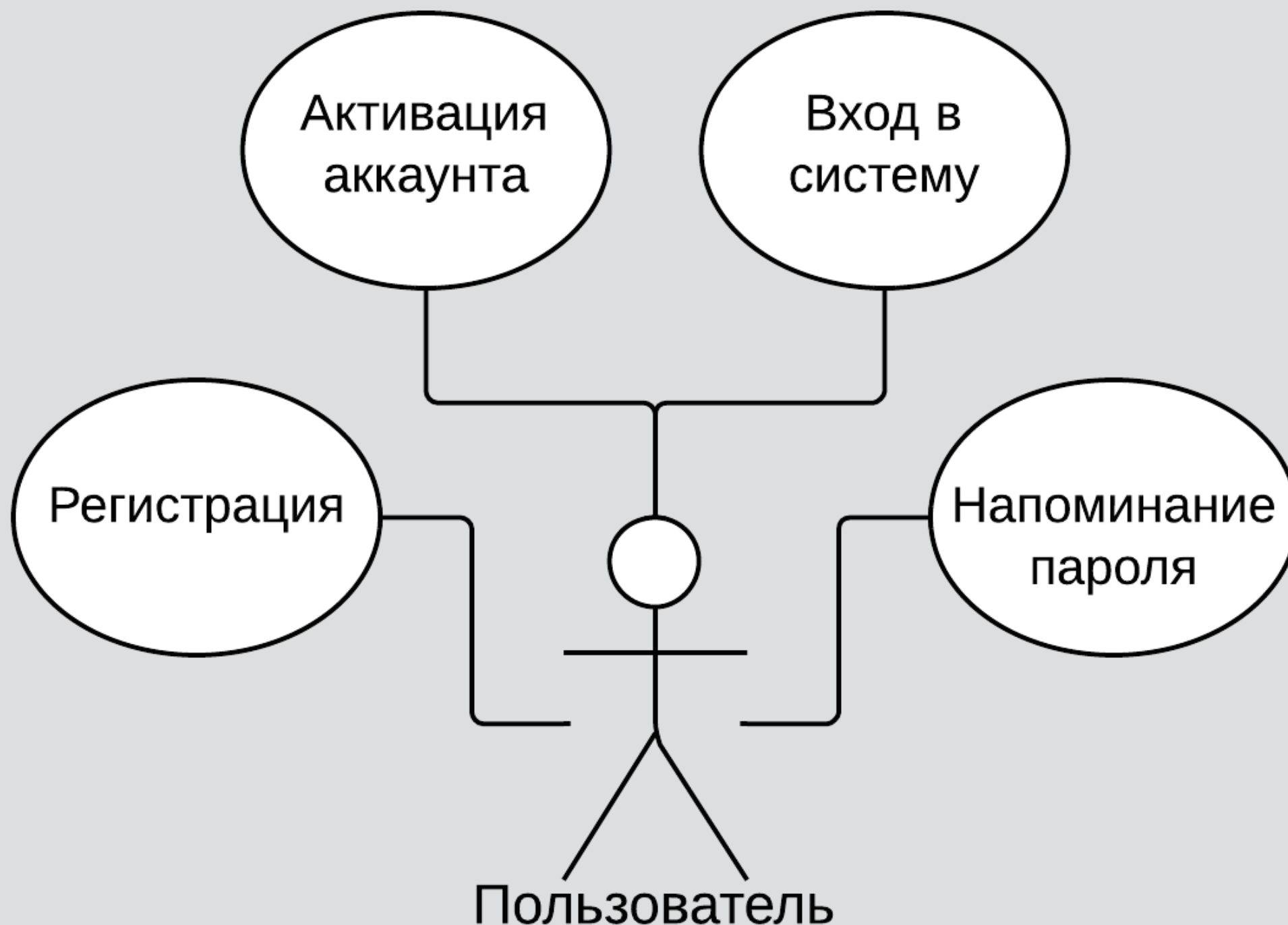
общая схема конструктора для абстрактного
университета



Плагин Face

- задаёт общий вид приложения;
- предоставляет шаблон для отрисовки представлений;
- задаёт основные CSS-стили и скрипты JavaScript.

Плагин Authentication

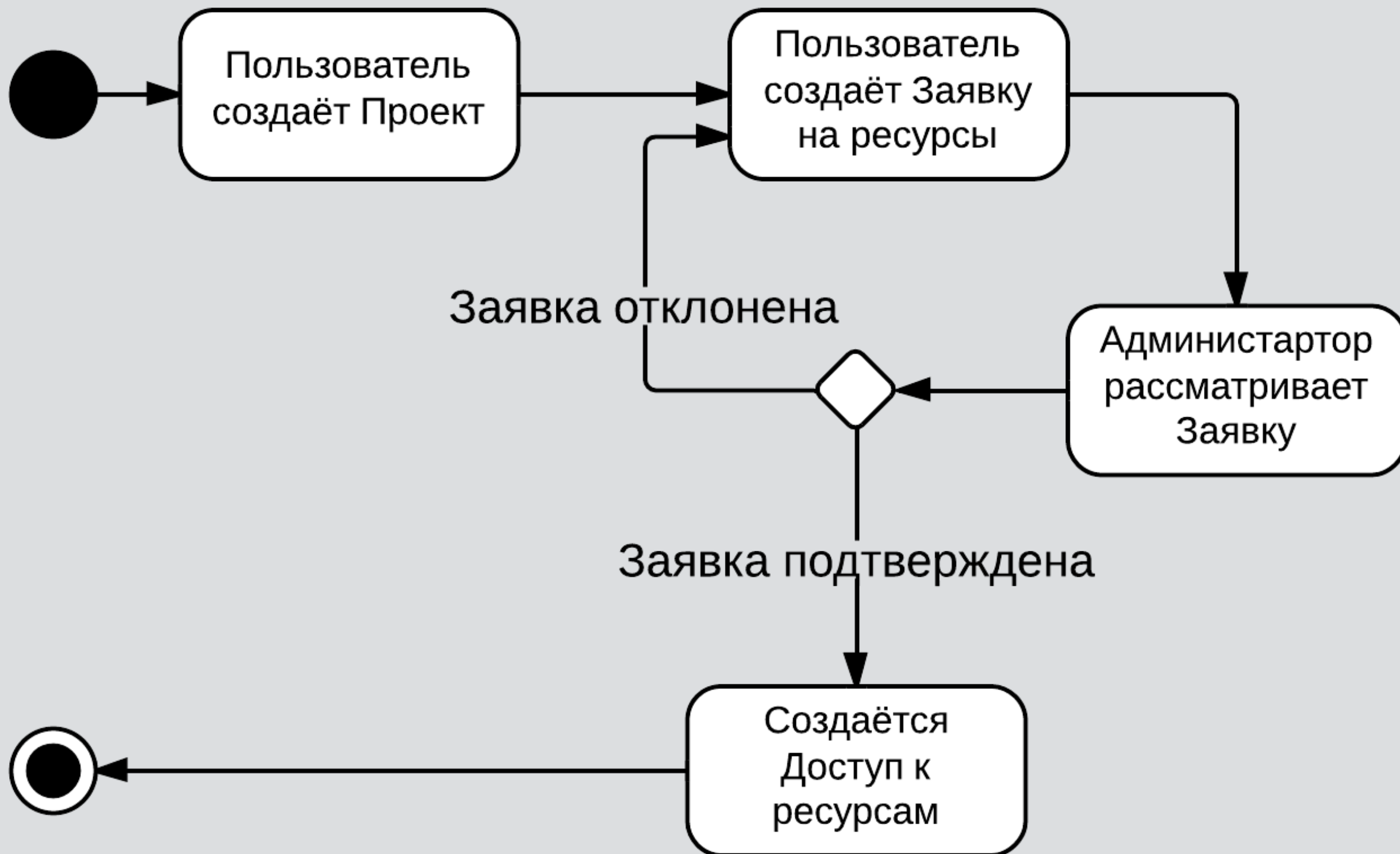


Демонстрация плагина Authentication

Плагин Requests

Общая схема работы:

1. Пользователь создаёт проект.
2. Пользователь создаёт Заявку на ресурс.
3. Администратор рассматривает Заявку.
4. После одобрения заявки открывается Доступ.
5. Проект синхронизируется с кластером через Доступ.



Демонстрация плагина Requests

Плагин создания и показа справки (wiki)

Требования к плагину:

1. Разграничение по правам:

1) администратор создаёт и редактирует справку;

2) пользователи просматривают справку.

2. Форматирование текста в Markdown.

Демонстрация создания плагина Wiki

Дальнейшие планы

1. Реализация плагина Техподдержка.
2. Сборка образа octoshell-basic.
3. Создание документации.
4. Доработка текущей версии: второй приоритет.

Спасибо за
внимание!