



Система автоматического разрешения обращений в техническую поддержку

Шумилов Пётр, 471 группа,
Математико-Механический факультет, СПбГУ
Научный руководитель: Я. А. Кириленко
Рецензент: Н.А. Ражев

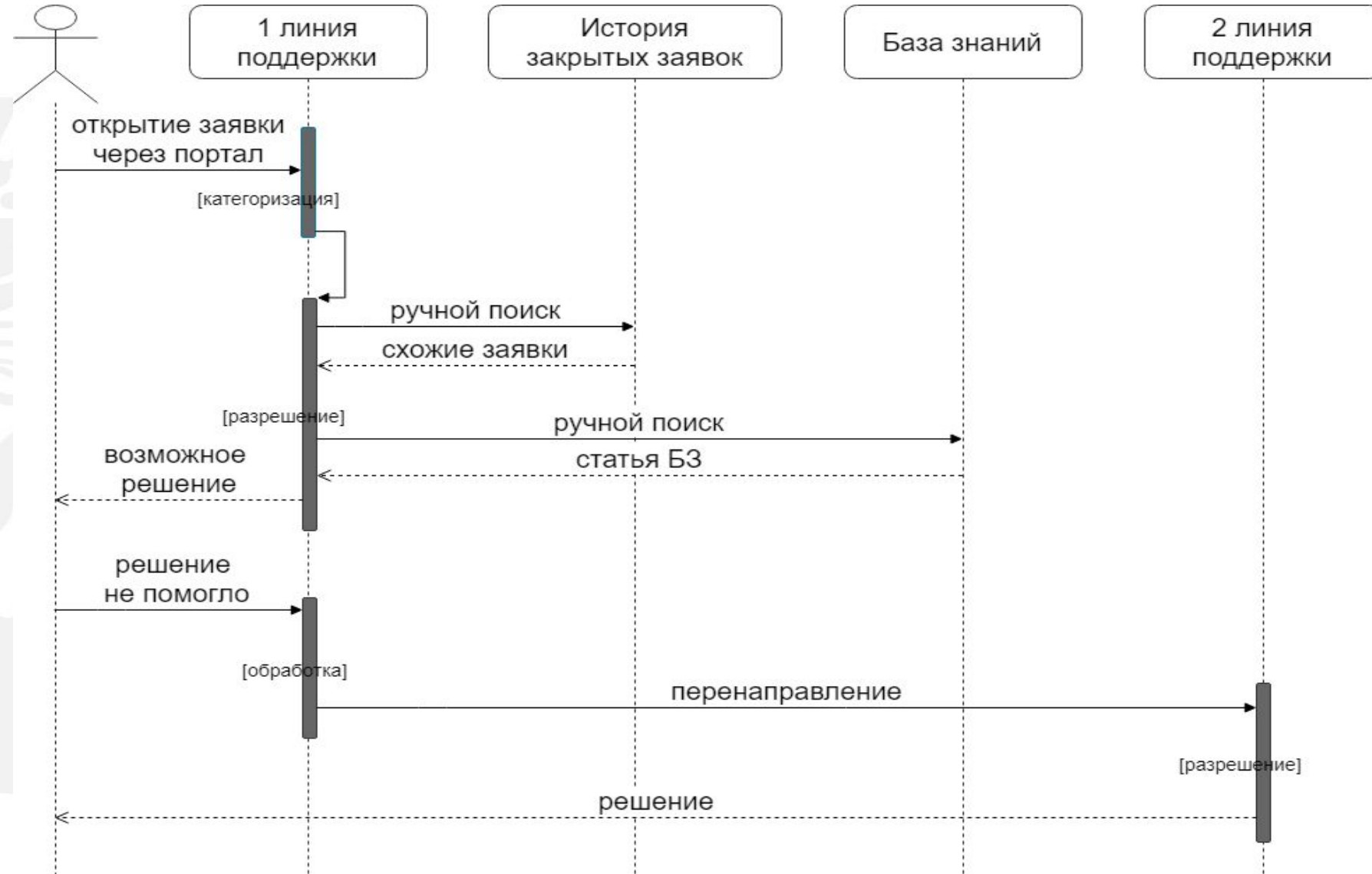


Введение

- Сегодня большинство компаний имеют систему технической поддержки
- Зачастую качество поддержки продукта играет ключевую роль
- Содержание штата специалистов поддержки стоит дорого
- Хорошей практикой является разделение поддержки на уровни
- Первый уровень занимается рутинами – можно автоматизировать



Введение





Цель

Нужно разработать прототип продукта для автоматического разрешения обращения в поддержку посредством рекомендации готового решения на основе:

- текста заявки
- истории ранее закрытых инцидентов
- данных о пользователе



Задачи

- Провести анализ требований
- Разработать архитектуру продукта
- Реализовать следующие компоненты:
 - сервис, который по тексту заявки возвращают топ N семантически схожих ранее разрешённых заявок
 - сервис, который на основе текста заявки определяют категорию обращения
 - сервис, учитывающий историю разрешенных инцидентов и рекомендующий тему, на которую стоит написать статью в базу знаний
- Разработать пользовательский интерфейс
- Апробировать систему у заказчика



Основные требования

- Размещение внутри ландшафта компании
- Интеграция с SAP Solution Manager
- Интеграция с базами знаний, в данном случае с MediaWiki
- Поддержка взаимодействия через чат-бота

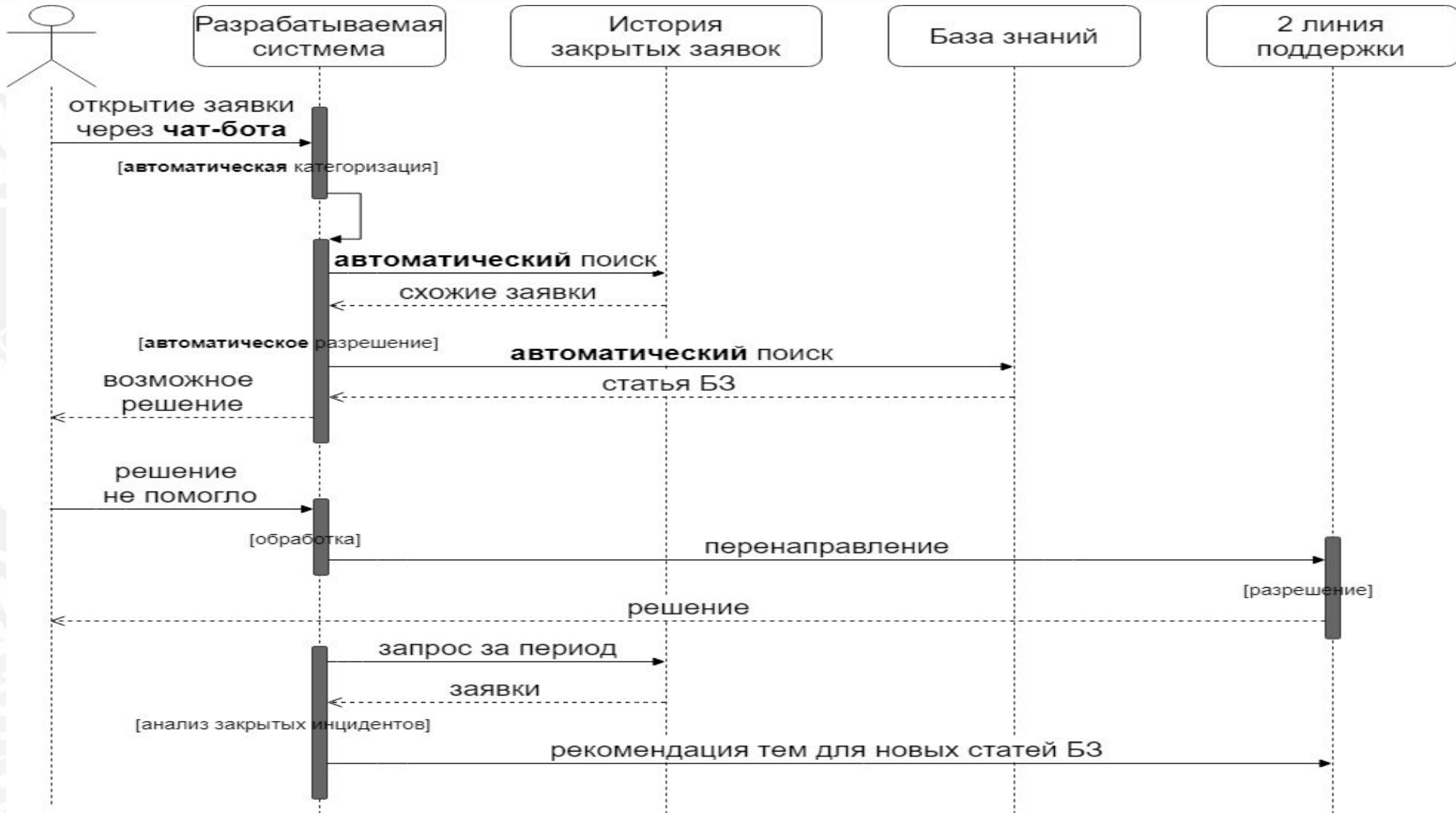


Обзор существующих решений

- ChangeGear Intelligence Platform:
 - Классификация заявок
 - Поиска решений по статьям базы знаний
 - On-premise
 - Невозможна интеграция с Solution Manager
- ServiceNow Intelligent Automation Engine
 - + Интеграция со Slack
 - + Интеграция с Solution Manager
 - Нет поддержки работы в изолированном ландшафте
- Популярные сервисы классификации заявок: только SaaS

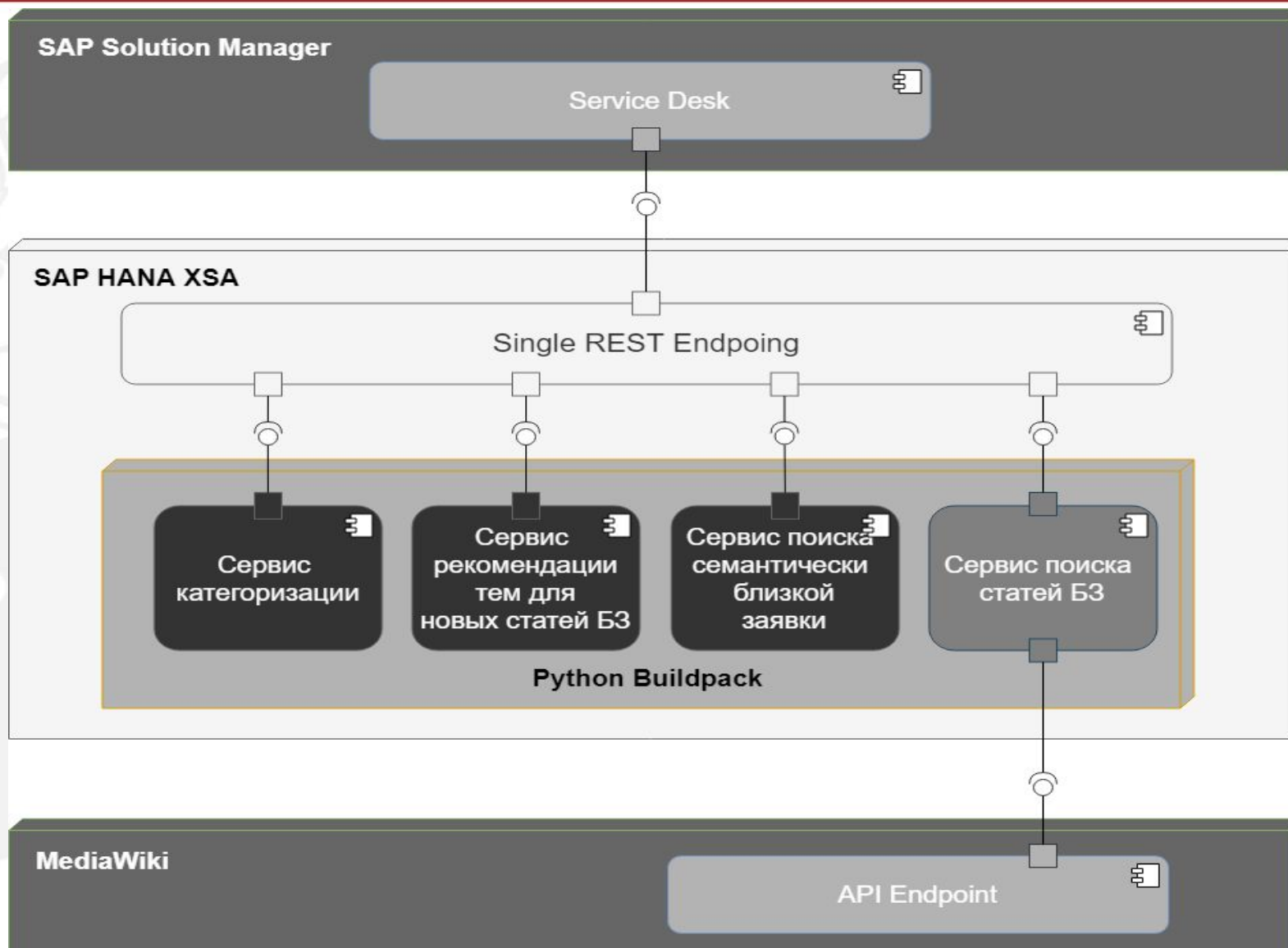


Обновлённый процесс



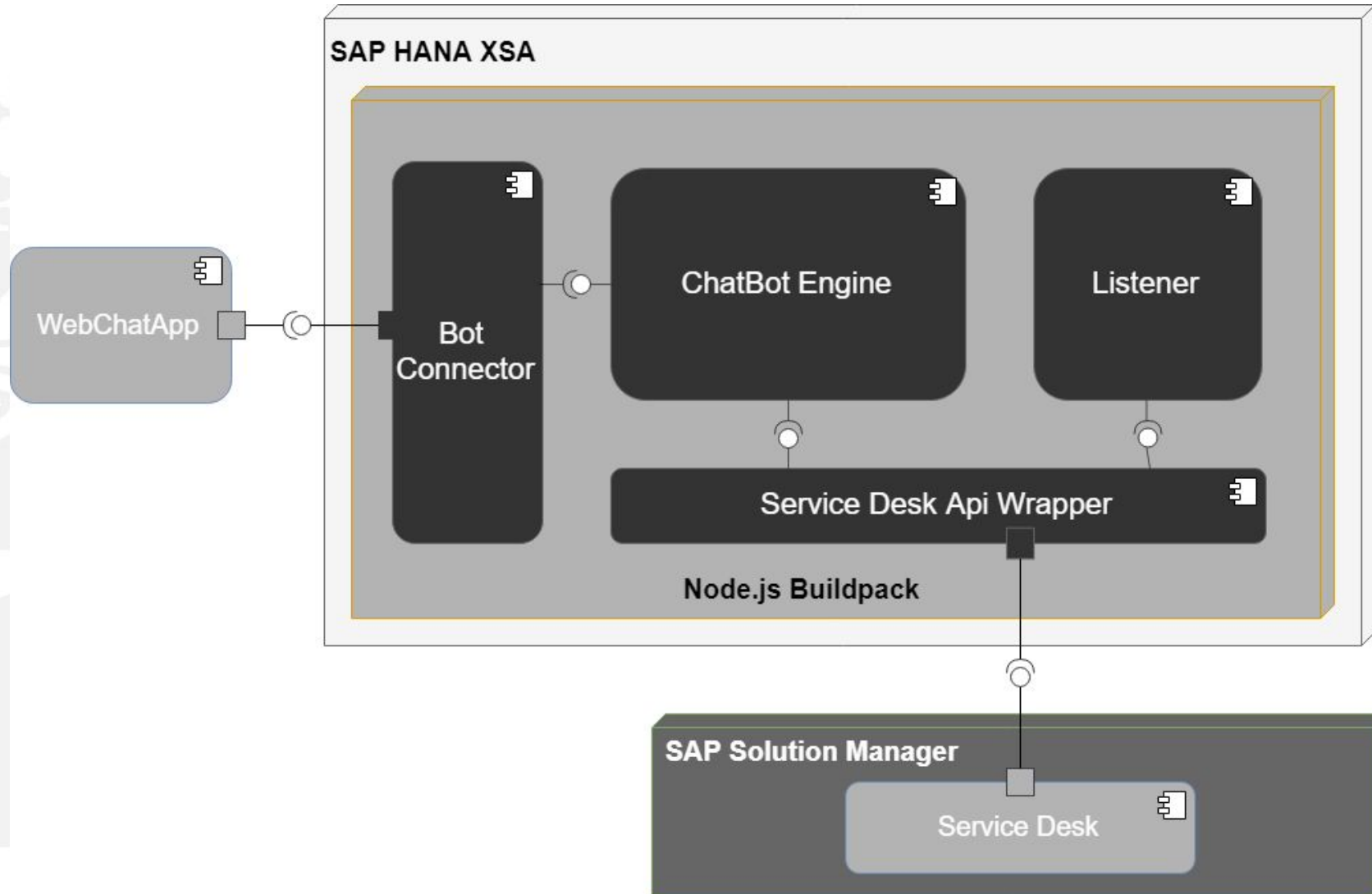


Архитектура





Архитектура чат-бота





- Предобработка истории заявок:
 - токенизация,
 - удаление нерелевантных символов и стоп слов,
 - исправление грамматических ошибок,
 - раскрытие аббревиатур,
 - поиск и удаление имён собственных,
 - нормализация (лемматизация).
- Векторизация документа: Word2Vec + TF/IDF или BoW
- Выделение топиков: K-means + LDA или LDA



Оценка качества

- Есть выборка исторических данных вида: «id заявки – статья БЗ», объём 91000 заявок
- “Comparing different term weighting schemas for Topic Modeling”, 2015
- Метрики качества кластеризации:
 - Adjusted Rand Index (ARI),
 - Silhouette
- Метрика удовлетворённости пользователя:
 - Понятность топика (субъективная)
- Сервис возвращает данные вида: «id заявки – топик»



Сравнение подходов

Подход / Метрика	ARI	Понятн. для пользов.	Коэф. Силуэта	Пример топики
Word2Vec + K-Means + LDA	0.66	+	0.20	«изменить сокращенное наименование кредитора»
BoW + LDA	0.74	-	Отсут.	«создавать дебитор согласно приложению заявка»



- Предобработка истории заявок:
 - токенизация
 - удаление нерелевантных символов и стоп слов
 - исправление грамматических ошибок
 - раскрытие аббревиатур
 - поиск и удаление имён собственных
 - нормализация (лемматизация)
- Векторизация предложения (Word2Vec + TF/IDF)
- Ранжирование по косинусному расстоянию
- Сервис возвращает топ N близких заявок



Оценка качества

- Есть выборка исторических данных вида: «заявка – категория», «заявка – статья БЗ», объём: 91000 заявок
- Заявки «сильно схожи», если у них одна категория и они закрыты с помощью одной статьи БЗ
- «схожи», если они закрыты с помощью одной статьи БЗ
- «слабо схожи», если у них одна категория
- «разные», в остальных случаях
- Для K заявок ищем топ N близких заявок и смотрим сколько в каждой выдаче «сильно схожих», «схожих» и т.д.
- В **61%** случаев сервис возвращает «сильно» схожие заявки



Результаты

- Проведён анализ требований
- Разработана архитектура продукта
- Разработано 3 сервиса:
 - сервис, который по тексту заявки возвращают топ N семантически схожих ранее разрешённых заявок
 - сервис, который на основе текста заявки определяют категорию обращения
 - сервис, учитывающий историю разрешенных инцидентов и рекомендуящий тему, на которую стоит написать статью в базу знаний
- Разработан пользовательский интерфейс
- Система доставлена заказчику и находится в стадии тестирования
- Результаты внедрения: получен отзыв от представителей заказчика

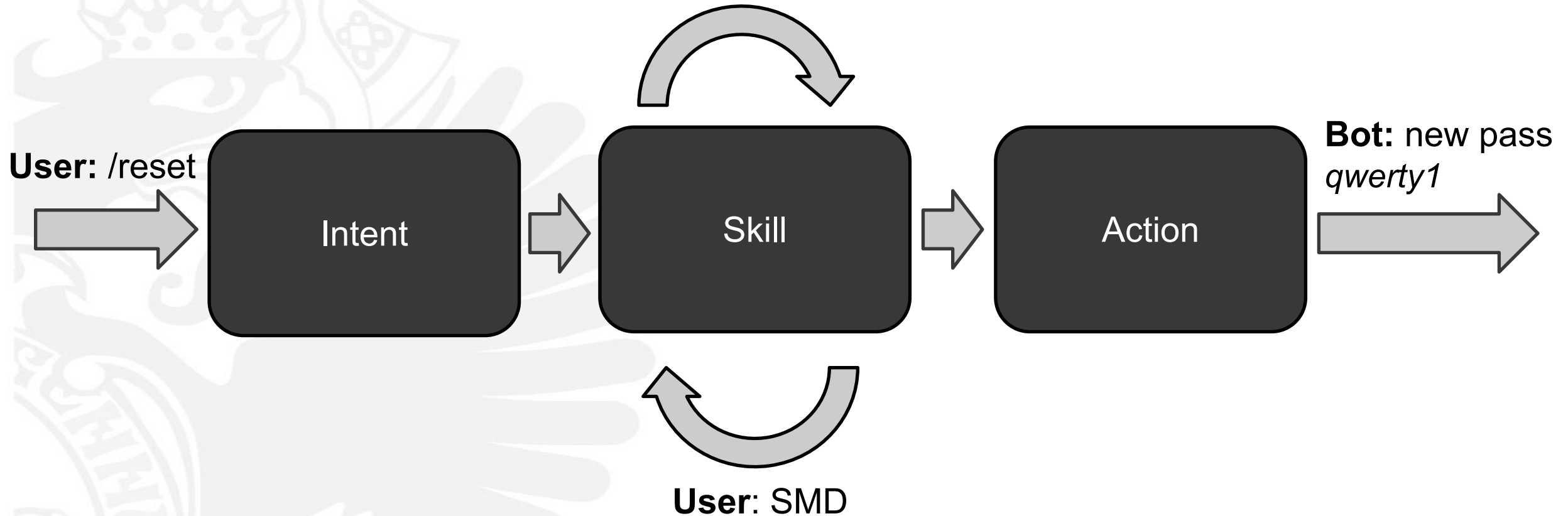


Санкт-Петербургский
государственный университет
spbu.ru



Особенности чат-бота

Bot: what system do you need?





Пример описания поведения

```
• const intents = {  
•   leave_request: {  
•     trigger: {  
•       type: 'string',  
•       locale: {  
•         "ru": '/отпуск'  
•       }  
•     },  
•     skill: 'leave_request'  
•   }  
• }
```

```
• const skills = {  
•   leave_request: {  
•     requirements: {  
•       leave_type: {  
•         type: 'string',  
•         replies: {  
•           ru: {  
•             suggestion: null,  
•             before: 'leave_request_ask_type',  
•             after: null  
•           }  
•         }  
•       },  
•       ...  
•     },  
•     action: [  
•       'leave_request_submit'  
•     ]  
•   }  
• }
```



- Предобработка истории заявок:
 - токенизация,
 - удаление нерелевантных символов и стоп слов,
 - исправление грамматических ошибок,
 - раскрытие аббревиатур,
 - поиск и удаление имён собственных,
 - нормализация (лемматизация).
- Векторизация предложения (Word2Vec + TF/IDF)
- Классификация (Логистическая регрессия)



Оценка качества

- Есть выборка исторических данных вида: «заявка – категория», объём 91000 заявок
- Метод оценки качества: кроссвалидация
- Метрика качества: Ассурасу
 - Вероятность, что правильная категория окажется в первых 3-х выданных сервисом: **87%**
 - Вероятность, что правильная категория окажется 1-ой из выданных сервисом: **74%**
- Количество классов: около 100



Особенность Word2Vec

