## Development of the infrastructure of automatic interaction of the university with GIS "Postupaj v vuz onlajn"

Keywords: REST, superservice, Python, JWT, entrant, admission, EDS, Windows services

The article discusses the development and implementation of information system components for interaction with the state information system "Superservice: Postupaj v vuz onlajn" on the basis of the Pacific State University. The prerequisites for the development, legislative, functional and information protection requirements for this development are considered.

## Разработка инфраструктуры автоматического взаимодействия ВУЗ-а с ГИС "Поступай в вуз онлайн"

Ключевые слова: REST, суперсервис, Python, JWT, абитуриент, поступление, ЭЦП, службы Windows

Предметом статьи является разработка вопросно-ответной системы базы знаний Тихоокеанского Государственного Университета. Изучение качества поиска в различных вариациях — при простом поиске по вхождению текста, при индексировании на основе «частотной важности» слов в документе, а также используя оптимизации поискового запроса на основе семантической близости и синтаксической важности членов предложения поискового запроса на основе RNN искусственных нейронных сетей семейств Word2vec и GloVe.

.

## Термины и определения

**ЭВМ** – электронно-вычислительная машина.

**СУБД** – система управления базами данных.

**REST** – Representational State Transfer, репрезентативная передача состояния – это архитектурный стиль, который позволяет создавать реализацию клиента и сервера независимо друг от друга. Сервисы в REST API взаимодействуют по протоколу HTTP.

**API** – это установка функций и правил позволяющая взаимодействовать между программным обеспечением, которое предоставляет API и другими программными компонентами. В Веб разработке, под API обычно подразумевают набор стандартных методов, свойств, событий и URL ссылок для взаимодействия с Веб контентом.

**СКЗИ** — Система(средства) Контроля и Защиты Информации

**JWT** - (сокр. от англ. JSON Web Token) – это JSON-объект, который определен в открытом стандарте RFC 7519. JWT состоит из трех частей: Header, Payload и Signature, представляющих собой строки в формате Base64, разделенные символом точка.

**ЭЦП (ЭП)** — Электронная Цифровая Подпись

**ORM** - Object-Relational Mapping, технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных»

## Введение

Вопросно-ответная система (QA-система; от англ. QA — англ. Question-answering system) — информационная система, способная принимать вопросы и отвечать на них на естественном языке, другими словами, это система с естественно-языковым интерфейсом. Актуальность разработки подобной системы продиктована необходимостью разработки автоматического ассистента службы поддержки ТОГУ, который бы с большей релевантностью предоставлял пользователю ответы на поставленные вопросы. Существующие сторонние разработки менее ориентированы на контекст работы и информационную специфику вуза и требуют больших усилий по интеграции.

Предложенный метод оптимизации пользовательского запроса улучшает семантическую емкость базы вопросов и ответов в тоже время позволяет не прибегать к тяжелым генеративным моделям при поиске заранее известной информации. Существующий ранее механизм поиска по документам в ИС ТОГУ не обеспечивает современный уровень качества работы пользователя с программой на естественном языке. Разработанный и исследованный на практике метод оптимизации поискового запроса помимо улучшения среднего ранка ответов, так же в будущих разработках может быть использован для получения качественного RAG контекста для современных генеративных LLM моделей. Метод использует обученные RNN модели семейства Word2Vec для синтаксического и семантического анализа токенов запроса.

Статья организована следующим образом. Раздел 2 представляет описание теоретической основы разработанных алгоритмов, раздел 3 содержит описание алгоритмов разработанного приложения и задействованных технологий, а раздел 4 сравнительный анализ эффективности разработанных алгоритмов, примененных к задаче поиска. Заключение, итоги, потенциальные направления будущих исследований и разработок представлены в разделе 5.

## Описание алгоритмов разработанного приложения и задействованных технологий

Свои истоки архитектуры моделей для обработки естественного языка ведут из рекуррентных нейронных сетей (RNN). Рекуррентные нейронные сети (RNN) — это нейронные сети с направленными связями между элементами. Выход нейрона может снова подаваться на вход. Такая структура позволяет иметь подобие «памяти» и обрабатывать последовательности данных, например, тексты естественного языка. В 2001 году идея использовать ИНС для анализа текста привела к рождению одной из первых embedding-моделей NLP. Модель принимала на вход векторные представления n предыдущих слов и может «понимать» семантику предложения. Однако, как модели представить слово какого-то языка в этом численном виде чтобы оперировать им? В русскоязычной литературе такие числовые векторы называются эмбеддингами, которые получены из слов или других языковых сущностей. В самой примитивной форме эмбеддинги слов получают простой нумерацией слов в некотором достаточно обширном словаре и установкой значения единицы в длинном векторе размерности, равной числу слов в словаре, такие эмбеддинги называют One-Hot Encoding (OHE) ембенддинги [19]. Как правило в чистом виде такие векторные представления редко используются т.к. вектор не хранит семантической информации о контексте используемого слова, а также необходимости переобновлять всю нумерацию пространства после добавления слова в словарь.

Следующим шагом в теории составлении векторных представлений стал «частотный эмбеддинг», в котором каждому слову в позицию, соответствующую его номеру, ставится в соответствие число - частота слова в корпусе и документе, а точнее, скорректированное значение частоты: формулу TF-IDF. Эта аббревиатура означает "term frequency — inverse document frequency.

|  |  |
| --- | --- |
|  | (1) |

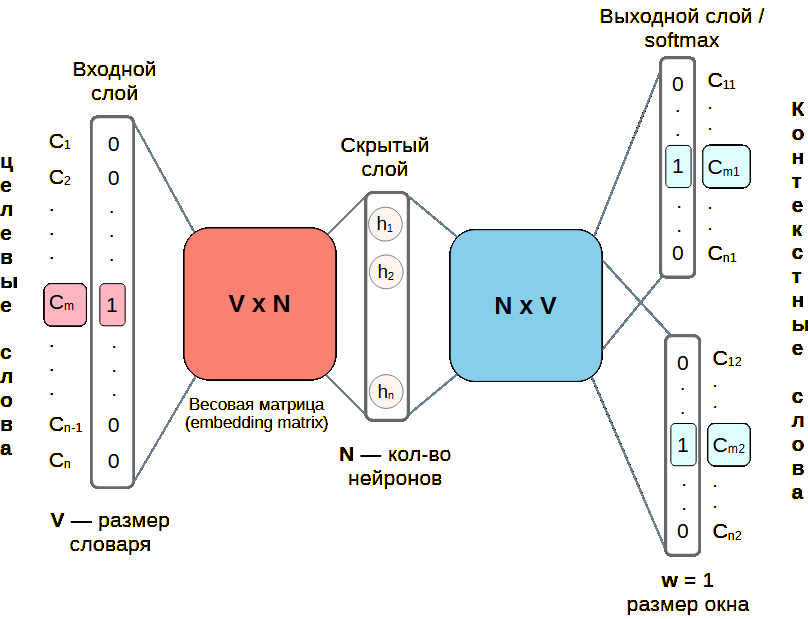
где TF — это частота слова в тексте. IDF — существенно более интересная вещь: это логарифм обратной частоты распространенности слова в корпусе (коллекции текстов). Показатель обратной частоты будет выше, если определённое слово с большой частотой используется в конкретном тексте, но редко - в других документах. Используя эмбеддинги в виде таких векторов, удалось впервые осуществить автоматический семантический анализ текстов, определяя имеющиеся в корпусе текстов темы и классифицировать тексты по основным темам.

Описанные выше подходы были (и остаются) хороши для времен (или областей), где количество текстов мало и словарь ограничен, хотя, как мы видим, там тоже есть свои сложности. В 2013 году Томаш Миколов предложил свой подход к word-embedding, который он назвал word2vec. Его подход основан на другой важной гипотезе, которую в науке принято называть гипотезой локальности — “слова, которые встречаются в одинаковых окружениях, имеют близкие значения”. Близость в данном случае понимается очень широко, как то, что рядом могут стоять только сочетающиеся слова [19]. Мы будем предсказывать вероятность слова по его окружению (контексту). То есть мы будем учить такие вектора слов, чтобы вероятность, присваиваемая моделью слову, была близка к вероятности встретить это слово в этом окружении в реальном тексте [20, 21].

|  |  |
| --- | --- |
|  | (2) |

где — вектор целевого слова,  — это некоторый вектор контекста, вычисленный (например, путем усреднения) из векторов, окружающих нужное слово других слов. А  — это функция, которая двум векторам сопоставляет одно число, например, это может быть косинусное расстояние. Приведенная формула называется softmax, то есть “мягкий максимум”, мягкий — в смысле дифференцируемый. Это нужно для того, чтобы модель могла обучиться с помощью backpropagation, то есть процесса «обратного распространения ошибки».

Вместо алгоритма непрерывного мешка слов, используемого для предсказания следующего слова по нескольким предыдущим по контексту, модель Word2Vec использует Skip-gram (словосочетание с пропуском). Цель этой модели прямо противоположная предыдущей модели – предсказать окружающие слова на основе центрального.

Рисунок 1. архитектура Word2vec ИНС (skip-gram), 1 скрытый слой, окно = 1

Архитектура модели представлена на рисунке 1. V - количество слов в словаре после обучения, каждое слово в словаре описывается как вектор с однократным кодированием (двоичный вектор, в котором только позиция соответствующего слова имеет значение 1), N - количество нейронов (размерность векторного пространства слов). Весовая матрица VxN хранит обученный вектор и модель предсказываются векторы, которые соответствуют словам близким по контексту входному — то есть при обучении находившихся слева и с права в тексте (окно w=1). Для Skip-gram модели объективная функция трансформируется, относительно CBOW, в сумму логарифмической вероятности окружающих n-искомых слов вокруг целевого слова [20]:

|  |  |
| --- | --- |
|  | (3) |

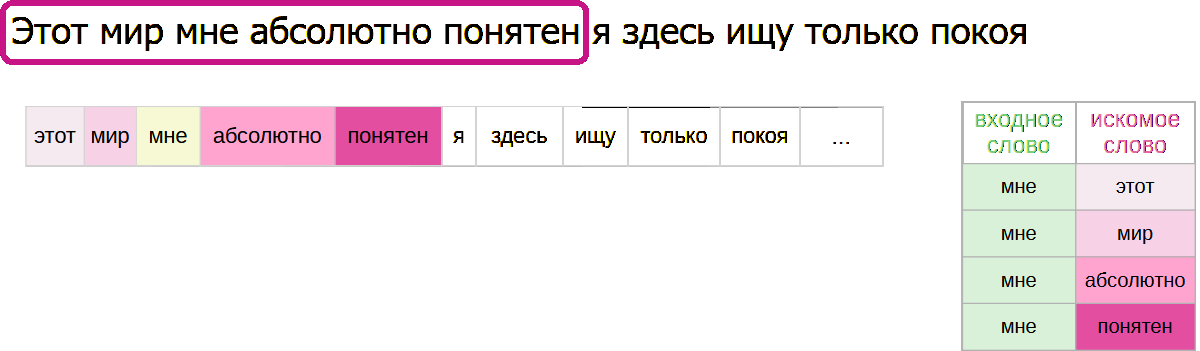


Рисунок 2. Принцип составления датасета для получения Skip-gram-модели Word2Vec для корпуса текста (рамка — контекстное окно)

С помощью упомянутых выше моделей Word2vec и аналогов GloVe возможно относительно дешево вычислительно (относительно тяжелых моделей типа BERT и NNLM моделей и других основанных на латентном семантическом анализе (LSA) и латентном распределении Дирихле (LDA)) находить семантическое сходство, выявлять словосочетания в тексте (парафразы), подходящие по контексту слова и другие операции имея предобученые на подходящем корпусе заданной тематике текста ембеддинги. В программе требуется работать с довольно независимым коротким пользовательским запросом, отчего контекст токенов не может быть большим. Кроме того, т.к. программа базы знаний призвана лишь оптимизировать пользовательский запрос для заданной поисковой машины, которая на данный работает по принципам лексемизации базы и вычисления статистической важности той или иной лексемы в наборе документов — поэтому достаточно моделей семейства Word2Vec и дистиллированных WordCNN моделей. Для программы разработано два алгоритма, использующих Word2Vec.

1. Алгоритм синтаксического анализа запроса, выявление основной части запроса. На рисунке 3 представлена блок-схема алгоритма

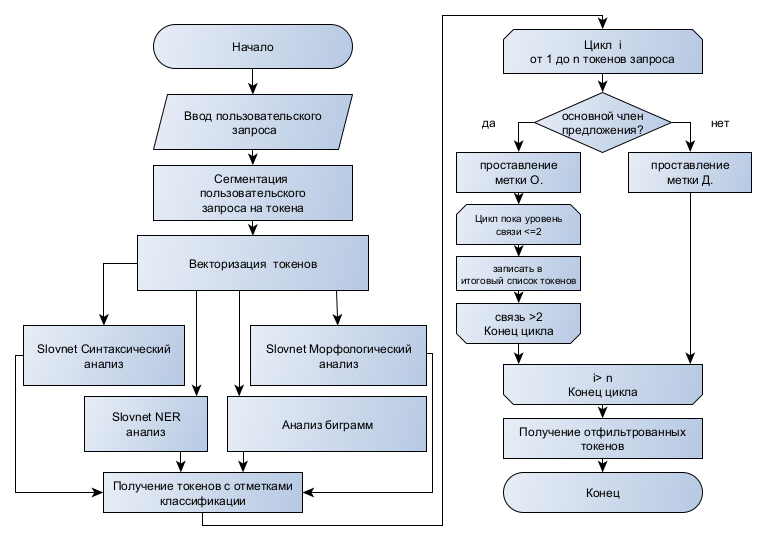


Рисунок 3. Алгоритм синтаксического анализа запроса, выявление основной части

* Начинается алгоритм с инициализации запроса в качестве документа, сегментированного на токены, в качестве разделителей используются знаки препинания и пробелы. Во время сегментации по устоявшимся стандартам NLP анализа токены менее 3 символов пропускаются — т. к. считается что их семантическая значимость низка и имеет повышеную омонимичность с другими морфемами слов что вносит излишную энтропию в анализ.
* После сегментации на токены — документ (запрос) векторизируется с помощью модели эмбеддингов Navec, задача модели представить токены (слова) как векторы векторного пространства, обученного на корпусе «navec\_news\_v1\_1B\_250K\_300d\_100q.tar».
* После векторизации параллельно происходит применение к документу (с уже векторизоваными токенами) две предобученых модели проекта Slovnet: Морфологический анализатор, Синтаксический анализатор. Эти модели тоже обучены на размеченных корпусах текста новостных статей.
* Сегментированный на токены документ поискового запроса тестируется на предмет наличия «биграмм» или словосочетаний из двух токенов и именованых сущностей (NER) двумя моделями. Для выявления биграмм в рамках данной работы была обучена статистическая модель на корпусе текстов проекта «Leipzig Corpora Collection» [25] и «Corus» [23]. Для выявления именованых сущностей используется предобученная модель из проекта Slovnet [26]. В токены документа которые относятся к биграмам или к NER добавляется атрибут принадлежности токена.
* После применения моделей для анализа текста — токены помечаются дополнительными атрибутами: определяется часть речи токена, и каким членом предложения является токен, также отмечается относится ли токен к именам собственным или персональным именам (NER) — дело в том, что синтаксическая значимость именовыных сущностей выше чем нарицательных существительных, NER, позднее в агоритме оптимизации NER точно также, как и словосочетания исключаются из оптимизаций.
* Полученые от модели синтаксического анализа маркеры связи членов предложения позволяют выделить основные члены предложения первого уровня связи как подлежащее и сказуемое и т.д.. Если запрос представляет собой сложное предложение, то запрос сокращается до 2 уровня связи — то есть основные и дополнительные члены предложения — подлежащее, сказуемое, одно определение или дополнение

1. Алгоритм оптимизации по семантической близости и TF-IDF. На рисунке 4 представлена блок-схема алгоритма

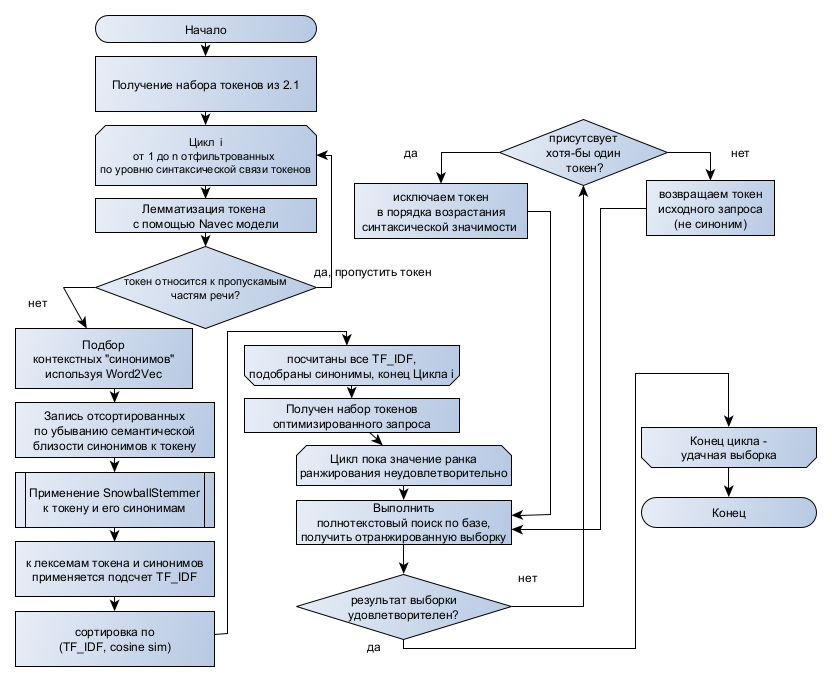


Рисунок 4. Алгоритм оптимизации по семантической близости и TF-IDF

* Программа использует набор токенов полученный после алгоритма 2.1. Токены итерируются, исходя из результатов синтаксического анализа группируются в группы основных и дополнительных членов предложения
* Токены содержащие отметки о NER или что это часть биграммы — сохраняются независимо от части речи и предложения
* Отфильтрованные по уровню анализируемой связи членов предложения (3) все исходные токены фильтруются, оставля лишь основные + NER + биграммы. В зависимости от типов предложения (полные, неполные, односоставные, многосоставные, однородные и т.д.) определение условного уровня связи может различаться
* После фильтрации каждый токен лемматизируется с помощью семантической модели
* После лемматизации из набора токенов дополнительно убираются члены предложения: знаки препинания, предлоги, союзы, местоимения не относящиеся к NER и биграммам.
* По лемматизированным токенам подбираются синонимы предобученной моделью Word2Vec, определяется индекс близости (косинусное расстояние). Синонимы сортируются по убыванию сходства и записываются в токен.
* К токену и его синонимам применяем стеммер (обрезаем изменяемые части слова, оставляем лексему, такую же как хранится в базе данных полнотекстового поиска)
* После этого цикла начинается еще один уже по оставшимся токенам, для которых найдены и синонимы и применены стеммеры. Для каждого токена и синонима выполняется функция подсчета показателя TF\_IDF в базе полнотекстового поиска.
* Для каждого оставшегося токена и синонимов ранжируется ряд по двум ключам сортировки — с первым ключом TF\_IDF лексемы, и вторым ее синтаксическое сходство с исходным токеном. Если у синонима TF\_IDF выше чем исходного токена — то он времено подменяет исходный токен в оставшемся запросе. Для всех оставшихся токенов применяется данный алгоритм. После него остается оптимизирвоанный список лексем которые имеют наибольшую среднюю меру TF\_IDF нашей базы знаний.
* Полученый набор токенов/синонимов отправляется как полнотекстовый запрос базе данных. По результатам запроса мы получаем выборку и значение Ранка для каждого результата.
* Если результат неудовлетворителен (выборка не преодолела некий порог ранка) то пробуем исключить токен/синоним начиная с неосновных членов предложения, повторяем поиск до тех пор пока результат поиска не будет удовлетворителен. Если после последнего токена результат по-прежнему не удовлетворителен — алгоритм аналогично в обратном порядке заменяет «оптимизированный» токен исходным.

## Особенности реализации, ключевые особенности

Исходя из выдвинутых спецификацией требований — программный комплекс АИС ТОГУ для взаимодействия с ССПВО должен содержать следующие части:

1. Модуль для приведения вида данных к требуемому виду из БД АИС ТОГУ
2. Модуль для генерации Электронной Цифровой Подписи по алгоритму ГОСТ 34.10-2012
3. Модуль «синхронного» взаимодействия с «Сервисом Приема» по протоколу REST
4. Модуль «асинхронного взаимодействия» с «Сервисом Приема»
5. Удовлетворение требований ФСТЭК по аттестации рабочего места для доступа к закрытой сети №13833
6. Пользовательский интерфейс для взаимодействия оператора с системой

Определим наиболее подходящие в контексте Тихоокеанского Государственного Университета технологии для реализации частей программного комплекса.

Пункт 5 уже реализован ВУЗ в рамках рабочего места с помощью ПАК ViPNet Coordinator. [3]

Для исполнения пункта 2 воспользуемся проприетарным программным обеспечением КриптоПро csp соответствующим крипто провайдером ГОСТ 34.10-2012, устанавливающимся вместе с программой. Выбор КриптоПро обусловлен тем что данный пакет имеет все необходимые сертификаты по реализации криптографических функций, обширную документацию и широкий набор библиотек, утилит и плагинов для интеграции.

Для исполнения пунктов 1, 3, 4 удобно использовать язык программирования Python, исходя из простоты интеграции в АИС ТОГУ. Язык имеет широкие возможности для Объектно-Ориентированного подхода к реализации логики взаимодействия, множество написанных открытых прикладных библиотек, отличную кроссплатформенность, возможность асинхронного программирования и работы с потоками.

Пункт 1 и 3 в более узком смысле будут требовать также задействования дополнительных библиотек:

- СУБД — PostgreSQL, свободная кроссплатформенная объектно-реляционная система управления базами данных

- для взаимодействия с СУБД — SQLAlchemy. Это многофункциональная ORM-библиотека, позволяющая быстро конфигурировать запросы к БД, обезопасить пользовательский ввод, используемый в запросах, а также создавать сложные запросы взаимосвязанные (databinding) с сущностями используемого языка программирования.

- связка Message Queueing broker сервера (RabbitMQ) и библиотеки Celery – для «отправки» потока сообщений от «веб-интерфейса пользователя» и «получение» их Celery для последующего параллельного основному серверу АИС ТОГУ исполнения. [6]

- библиотека Memcached - программное обеспечение, реализующее сервис кэширования данных в оперативной памяти на основе хеш-таблицы, используется в программном комплексе для сопутствующего кэширования и общего хранилища памяти между разными исполнителями задач для «синхронизации».

Исходя из требований и технологий, которые необходимо использовать можно составить схему архитектуры программного комплекса:

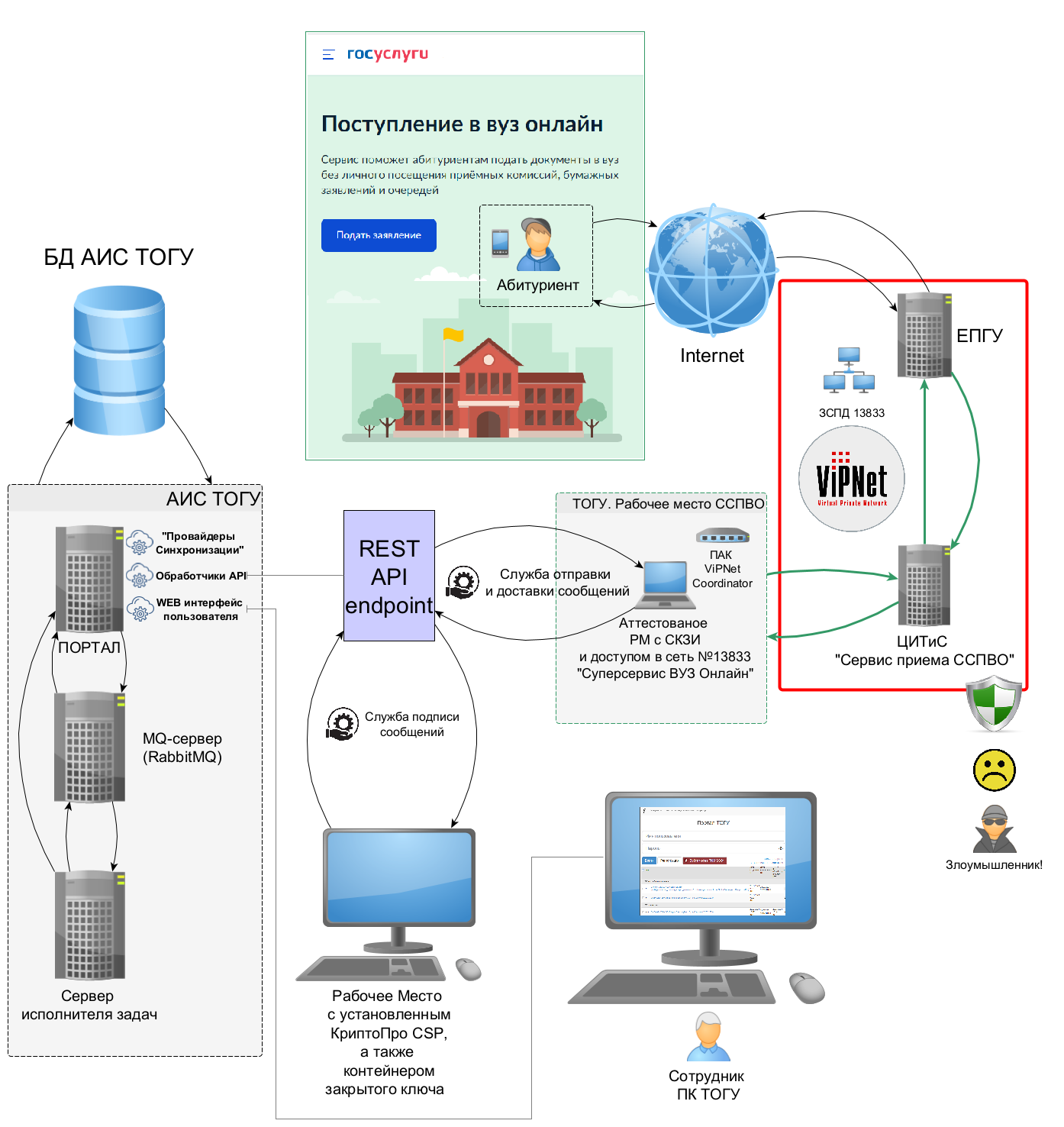


Рисунок 3. Схема взаимодействия с суперсервисом «Поступи в вуз онлайн»

**Описание программного комплекса**

Программный комплекс состоит из трех групп, разделенных по узлам схемы взаимодействия.

1. «ПОРТАЛ ТОГУ».

В рамках этой группы сформирована концепция "Провайдер синхронизации" – это сущность АИС ТОГУ, реализованная на ЯП Python, заключающая

а) логику трансформации исходящих данных, интерфейса пользователя и отдельного обработчика ответа, асинхронно пришедшего от ССПВО. Результат исполнения провайдера есть сущность "сообщение ССПВО" которое доставляется по маршруту дальше. Провайдер может исполнятся с определенным «контекстом вызова» как автоматически, по достижению каких-то условий в АИС ТОГУ, так и в ручном режиме. Провайдер может инициировать запуск дочерних провайдеров.

б) логику трансформации входящих данных (сообщений ССПВО) в АИС ТОГУ, логирование изменений АИС ТОГУ внутри неподтвержденной транзакции БД - для получения того как сообщение ССПВО потенциально изменит БД без проведенных изменений.

Web интерфейс работы с сообщениями ССПВО представлен на рисунке ниже.

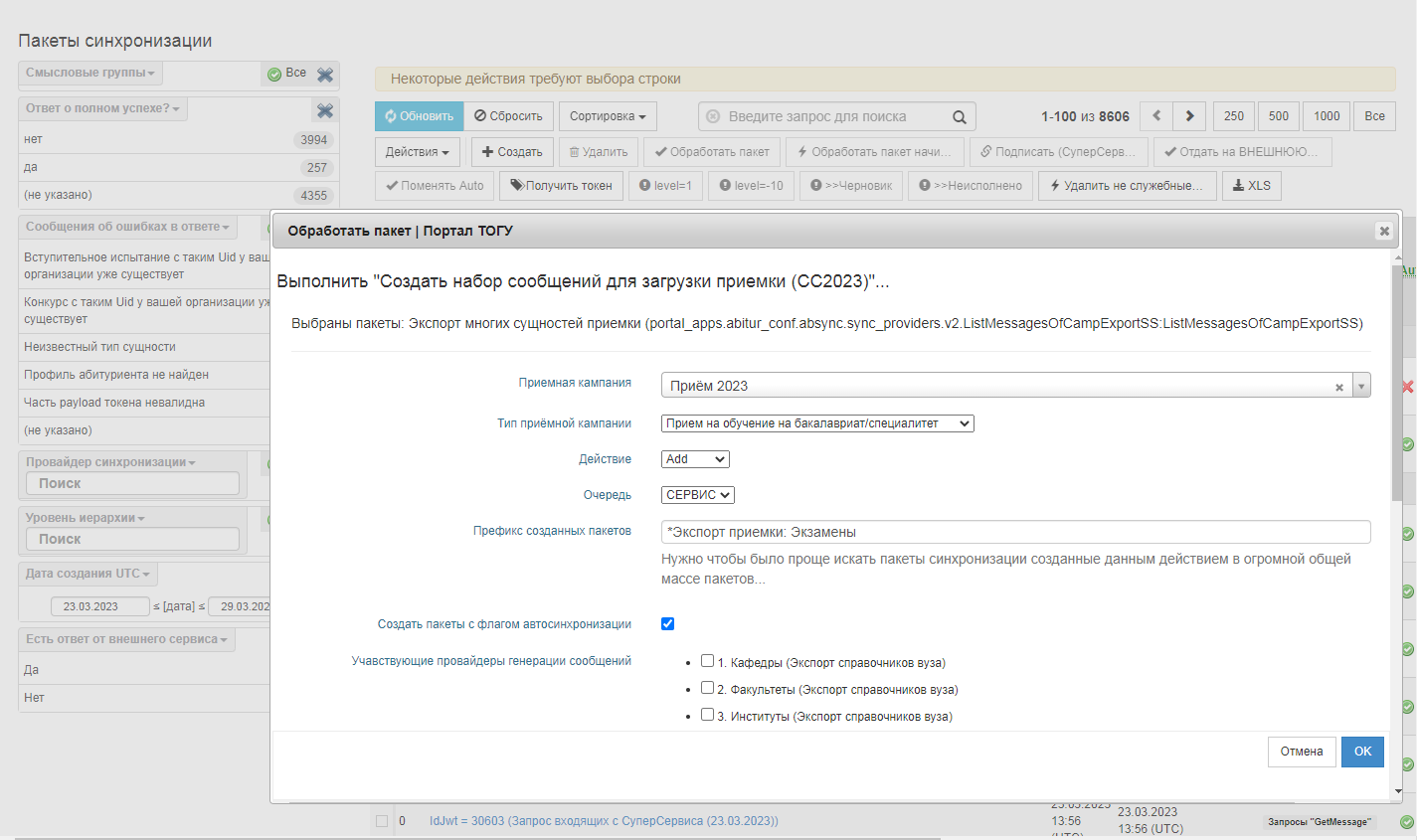


Рисунок 4. Интерфейс управления системой для пользователя

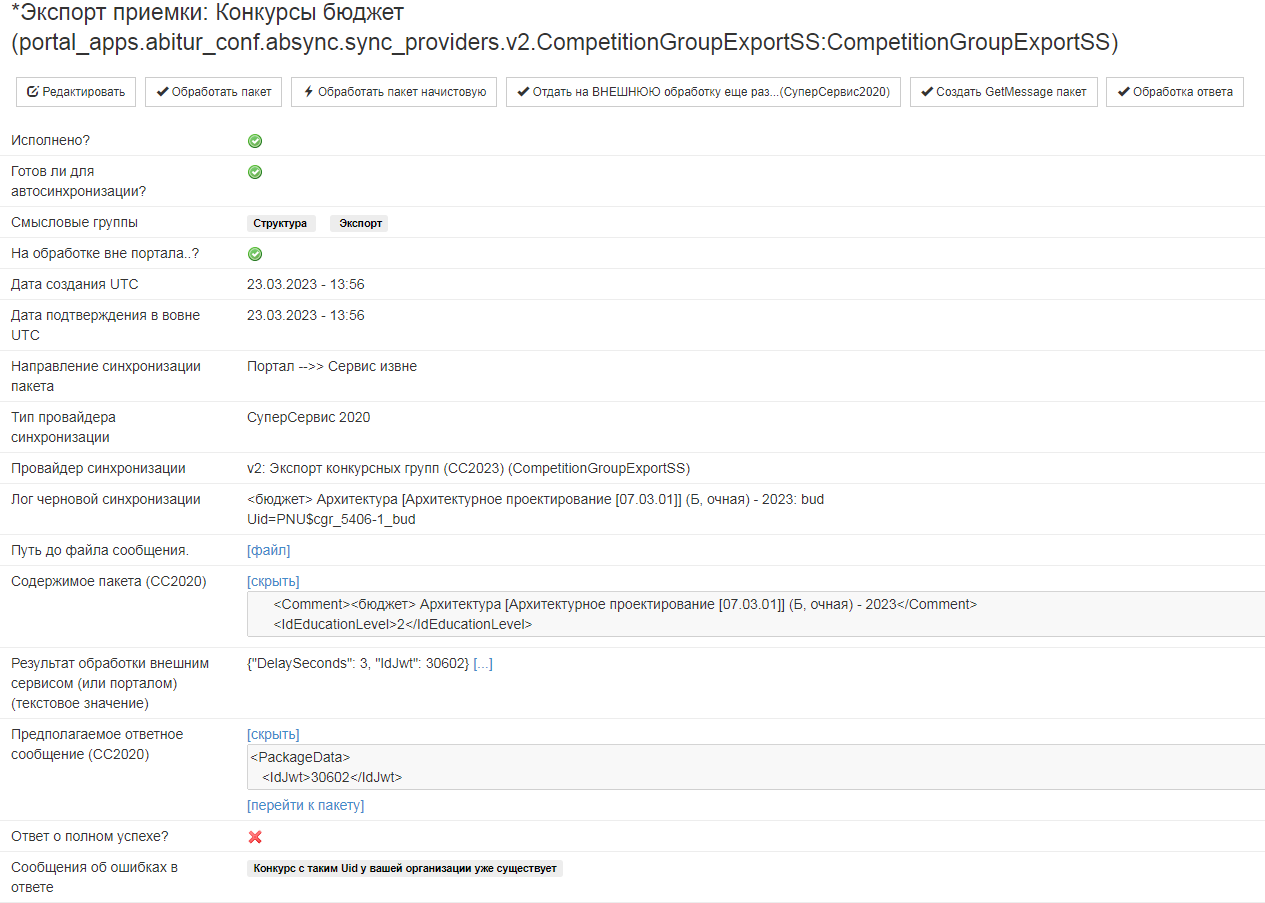


Рисунок 5. Интерфейс просмотра и редактирования сообщений ССПВО

После формирования сообщений ССПВО (в автоматическом режиме, после определенных условий) или в ручном, ограниченные операции над ними станут доступны по REST API ТОГУ.

2. «Узел ЭЦП».

Далее, по документации устанавливающей взаимодействие сказано, что некоторые типы сообщений ССПВО будут требовать указания в сообщения отделённой ЭЦП. Подпись таких сообщений осуществляется в автоматическом режиме службой, установленной на рабочем месте с токеном закрытого ключа с помощью криптопровайдера «Crypto-Pro GOST R 34.10-\* Cryptographic Service Provider». Закрытый ключ не передается ни по каким каналам, формирование подписи происходит на локальной машине на которой была запущена служба. Служба реализована на языке программирования Python 3.7 с помощью библиотеки асинхронных HTTP запросов aiohttp. [1]

Окно работы службы показано на рисунке ниже.

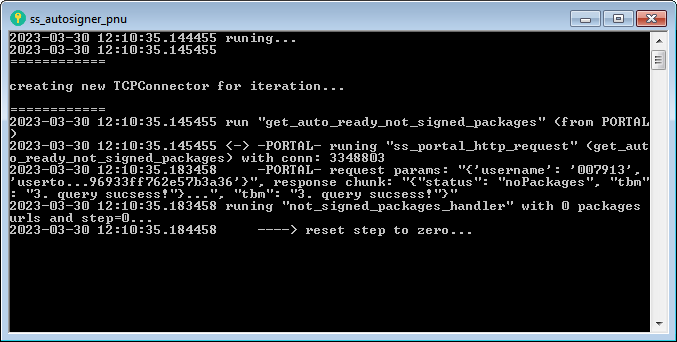


Рисунок 6. Служба подписи сообщений ССПВО перед отправкой

3. «Узел отправки и получения сообщений ССПВО».

Далее, после успешной подписи (если она требовалась сообщению), другая служба, уже на рабочем месте внутри защищенной сети № 13833, отправляет «ожидающие отправки» сообщения из АИС ТОГУ в Сервис приема ССПВО, и по REST API возвращает АИС ТОГУ результат отправки. Также служба принимает входящие сообщения от сервиса приема ССПВО (ЕПГУ) и по REST API записывает их в АИС ТОГУ. Служба реализована на языке программирования Python 3.7 с помощью библиотеки асинхронных HTTP запросов aiohttp. [1]

Окно работы службы показано на рисунке ниже:

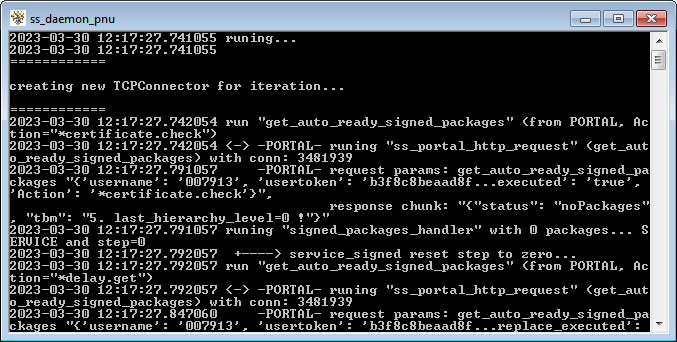


Рисунок 7. Служба отправки/приема сообщений ССПВО

## Заключение

В ходе работ по проектированию программного комплекса для взаимодействия с ГИС "Поступай в вуз онлайн" была создана целая инфраструктура, включающая несколько узлов, программ и каналов взаимодействия между ними. Реализация включила в себя создание подходящих таблиц в СУБД, программирование логики трансформации сущностей БД ТОГУ, событий обработки сообщений, проектирование интерфейса пользователя, написание служб автоматической отправки и получения сообщений. Выбранные технологические решения, хотя и имеют незначительные проблемы с пиковой производительностью, однако благодаря выбранной асинхронной микросервисной архитектуре очень хорошо горизонтально масштабируется – при надобности можно поднимать дополнительные узлы, также узлы не зависят от «совместимости решений» - например в части межсетевых экранов и криптопровайдеров различных компаний, а также отдельные части платформо-независимы в части ОС.

## Список использованных источников

1. AIOHTTP documentation. – Текст: электронный // AIOHTTP: официальный сайт. – URL:

https://docs.aiohttp.org/en/stable/ (дата обращения: 30.03.2023).

2. Постановление Правительства РФ № 89 от 26.03.2023. – Текст: электронный // Официальный интернет-портал правовой информации: официальный сайт. – URL:

http://publication.pravo.gov.ru/Document/View/0001202301270024?index=0&rangeSize=1 (дата обращения: 30.03.2023).

3. Компоненты виртуальной сети ViPNet. – Текст: электронный // Infotecs: официальный сайт. – URL:

https://infotecs.ru/about/press-centr/publikatsii/printsipy-marshrutizatsii-i-preobrazovaniya-ip-trafika-v-vpn-seti-sozdannoy-s-ispolzovaniem-tekhnolo.html (дата обращения: 30.03.2023).

4. Семафор (программирование). – Текст: электронный // Wikipedia: официальный сайт. – URL:

https://ru.wikipedia.org/wiki/Семафор\_(программирование) (дата обращения: 30.03.2023).

5. ГОСТ 34.10-2018. – Текст: электронный // Wikipedia: официальный сайт. – URL:

https://ru.wikipedia.org/wiki/%D0%93%D0%9E%D0%A1%D0%A2\_34.10-2018 (дата обращения: 30.03.2023).

6. RabbitMQ Documentation. – Текст: электронный // RabbitMQ: официальный сайт. – URL:

https://www.rabbitmq.com/documentation.html (дата обращения: 30.03.2023).

7. Документация АПИ Сервиса приема ССПВО . – Текст: электронный // Сервис приема ССПВО (тестовый контур): официальный сайт. – URL:

http://85.142.162.21:8032/ (дата обращения: 30.03.2023).