

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И АНАЛИЗА ДАННЫХ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

# КУРСОВОЕ ПРОЕКТИРОВАНИЕ

## Интеграция автоматизированной системы расчетов и автоматической телефонной станции

Б-ИН-21-167437. 11414-с. 02. КП

Студент

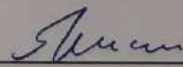
гр. БИН-21-01



Толстов А.А.

Руководитель,

Профессор каф. ИТС, д.т.н



Гриняк В. М.

Владивосток 2025

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ  
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СИСТЕМ

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ  
на курсовое проектирование

Студенту гр. БИН-21-1 Толстову Андрею Анатольевичу

Тема курсовой работы: «Интеграция автоматизированной системы расчетов и автоматической телефонной станции»

Дата утверждения «26» декабря 2024 года

Срок сдачи работы «17» января 2025

**Техническое задание:**

**Цель:**

Автоматизация модуля интеграции учетных записей пользователя и расчетов трафика и остальной информации, поступающей из биллинга, унификация форматов вывода информации со всех станций.

**Технические требования:**

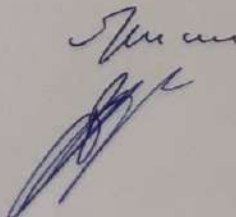
10; Язык программирования – Java, Python; Система будет предназначена для ОС Windows

Структура курсовой работы:

1. Содержание;
2. Введение;
3. Обоснование выбора темы;
4. Цели и задачи;
5. Изучение решений и требований к ним;
6. Разработка требований к системе;
7. Проектирование системы;
8. Заключение;
9. Список использованных источников.

Руководитель,  
Профессор каф. ИТС, д.т.н

Задание получил



В.М. Гриняк

А. А. Толстов

## Аннотация

Курсовая работа посвящена теме «Интеграция автоматизированной системы расчетов и автоматической телефонной станции».

Автор: студент гр. БИН-21-1, Толстов Андрей.

Руководитель: Гриняк В.М.

Цель работы – межсистемная интеграция АСР и АТС в условиях мультиоператорского присоединения.

В данной работе описывается процесс исследования комплекса задач, связанных с разработкой решения по межсистемной интеграции АСР и АТС в условиях мультиоператорского присоединения, с использованием протокола AAA Radius, для компании «ПотрТелеком». Были изучены предметная область и принципы работы системы, текущее состояние модуля интеграции учетных записей, модификация данного модуля и последующее изучение системы Lanbilling и системы Eltex. Проведено проектирование процессов, что включает в себя понимание принципа взаимодействия биллинга с АТС, серверное представление программного интерфейса. В ходе работы рассмотрены особенности компании к процессу разработки, анализ путей интеграции, а также регламенты и правила команды. Работа представляет интерес с точки зрения практического применения в сфере телекоммуникации.

## Содержание

Введение.....	5
1 Обоснование выбора темы .....	6
2 Цели и задачи интеграции .....	7
3 Изучение существующих решений.....	9
3.1 Требования и стандарты систем автоматизации.....	9
3.2 Описание и анализ существующих систем .....	11
3.3 Определение уже существующей системы.....	16
4 Разработка требований к интеграции .....	18
4.1 Определение функциональных требований.....	18
4.2 Определение необходимых данных для работы системы .....	22
5 Проектирование интеграции.....	23
5.1 Протекающие процессы.....	23
5.2 Выбор языка программирования.....	24
5.3 Описание компонентов системы .....	26
Заключение.....	28
Список использованных источников .....	29
Приложение А .....	31
Приложение Б.....	32
Приложение В .....	33

## Введение

В настоящее время, когда клиенты все более требовательны к скорости и качеству обслуживания, интеграция АСР и АТС открывает новые возможности для повышения эффективности работы организации. Данная интеграция дает ряд значительных преимуществ, которые можно обобщить следующим образом:

- Точная и своевременная тариффикация (биллинг);
- Автоматизация учета УЗ;
- Управление услугами и тарифами;
- Улучшение клиентского опыта.

Актуальность исследования данной темы определяется тем, что интеграция АСР и АТС с применением протокола AAA Radius позволяет не только оптимизировать внутренние процессы компании и повысить качество обслуживания клиентов, но и значительно усилить безопасность информационных систем, защищая чувствительные данные от несанкционированного доступа или неправильной настройки, что порой приводит к большим экономическим проблемам. При этом, выбор оптимальных методов и технологий интеграции, а также проектирование эффективного решения с учетом требований безопасности, является непростой задачей, требующей глубокого анализа и обоснования.

В рамках курсовой работы будет проведён анализ существующих подходов и технологий интеграции, разработана модель межсистемного интеграционного решения для автоматизированной системы расчетов и автоматической телефонной станции.

## 1 Обоснование выбора темы

С развитием технологий и увеличением объёма потребляемых и передаваемых конечным пользователям данных системы обработки расчетов тарификаций и информации в целом, компании сталкиваются с необходимостью поддержания высокой планки качества обслуживания, скорости реагирования и уменьшения ошибок и затрат в работе до необходимого минимума. Однако, выполнение работы чаще всего включает в себя человеческий фактор, что сильно затягивает и порой усложняет процесс создания, записи, разработки или тестирования различных методов исполнения задач программой.

Человеческий фактор во время заполнения шаблонных данных может повлиять на появление пустых данных или других ошибок, а отсутствие системы учета данных, к примеру, может привести к глобальным сбоям уже интегрированных процессов, которые долгое время копили в себе ошибки. Точность и достоверность данных зависят от состояния человека и исправности оборудования, и конечно же от правильности его использования. Также эффективность работы программы, в которой требуется присутствие условного оператора, может подвергаться сомнению, если за время работы программы были обнаружены ошибки в данных, которые заполнялись человеком. Все эти факторы заставляют задуматься на счет замены человека на автоматизированную программу сбора информации, которой нужно задать на вход данные, и она не допустит ошибок.

Таким образом, интеграция системы автоматизированных расчетов и автоматической телефонной станции является наиболее эффективным методом, позволяющим улучшить контроль над качеством услуг, уменьшить затраты экономических средств, а также уменьшить человеческие ресурсы на исполнение тех или иных задач в области автоматизации.

## 2 Цели и задачи интеграции

Первым делом стоит разобраться, что такое АСР, АТС, а также, что означает их интеграция в условиях мультиоператорского присоединения.

Автоматическая телефонная станция(АТС) - это, по сути, коммутационный центр в телефонной сети. Ее основная задача заключается в установлении, маршрутизации и управлении телефонными соединениями. Представьте, что АТС – это интеллектуальный «диспетчер» телефонных звонков. Когда вы набираете номер, АТС определяет, куда нужно направить этот вызов. Она отвечает за то, чтобы ваш звонок достиг нужного адресата, будь то другой абонент в той же сети, абонент другой телефонной сети, или даже международный номер. Кроме маршрутизации, АТС выполняет и другие важные функции. Она обеспечивает возможность подключения множества телефонных аппаратов к сети и управления этими подключениями. АТС следит за состоянием линий связи, определяя, свободна линия или занята. Она может обрабатывать дополнительные сервисы, такие как переадресация вызова, удержание вызова, конференц-связь и многое другое. АТС также может вести записи о телефонных соединениях, которые впоследствии используются для биллинга. В общем, АТС – это сердце телефонной сети, обеспечивающее её работоспособность и предоставление пользователям возможности связываться друг с другом.

В свою очередь, Автоматизированная система расчетов(АСР) представляет собой программно-аппаратный комплекс, который занимается учетом и тарификацией услуг связи. Если АТС можно сравнить с «диспетчером» звонков, то АСР – это бухгалтерская книга оператора связи. Ее основная задача заключается в сборе информации о предоставленных услугах связи, их тарификации и формировании счетов для абонентов. АСР получает данные о звонках, SMS, интернет-трафике и других услугах связи от различных источников, включая АТС. На основании этих данных АСР применяет заданные тарифные планы, рассчитывая стоимость использованных услуг. АСР учитывает всевозможные факторы, такие как время суток, продолжительность звонков, использованный объем трафика, подключенные дополнительные услуги и т.д. Она отслеживает баланс абонентов, обрабатывает платежи, формирует счета и отправляет их абонентам. Кроме того, АСР часто используется для ведения статистики, анализа потребления услуг и разработки тарифных планов. В итоге, АСР обеспечивает финансовую прозрачность деятельности оператора связи, позволяя ему правильно тарифицировать услуги и получать оплату за их предоставление.

Интеграция автоматизированной системы расчетов и автоматической станции означает создание тесной и автоматизированной связи между двумя системами, обеспечивающей их совместную работу и обмен данными. Это не простое их подключение к одной сети или базе

данных, а именно глубокое взаимодействие, при котором АТС и АСР работают как единое целое, дополняя друг друга.

В контексте данной работы, основные цели интеграция АТС и АСР будут выглядеть следующим образом:

1. Повышение точности и оперативности биллинга;
2. Автоматизация процессов и снижение операционных затрат;
3. Повышение гибкости и масштабируемости системы;
4. Обеспечение надежности и стабильности работы.

Задачами данной интеграции будут являться следующие пункты:

1. Создание механизма, который будет передавать данные из АТС в АСР;
2. Возможность обеспечить тарификацию услуг связи в соответствии с тарифами;
3. Автоматизировать формирование отчетов и их отправку;
4. Возможность изменения или настройки тарифов;
5. Возможность собирать и предоставлять статистические данные;
6. Интеграция должна обеспечивать защиту и безопасность данных.



## 3 Изучение существующих решений

### 3.1 Требования и стандарты систем автоматизации

Перед тем, как приступить непосредственно к анализу уже существующих решений, следует провести исследование требований и стандартов, требуемых для подобных систем. Проектируемая система должна соответствовать стандартам для введения её в эксплуатацию. Отсутствие предварительного анализа может повлечь за собой отказ от внедрения в систему, проблемы с интеграцией, отказ от партнёрства и прочие проблемы.

В настоящее время существует несколько федеральных законов, которые содержат в себе определённые требования для телефонии, и, следовательно, ко всем системам, касающимся работы этих систем.

Федеральный закон №152-ФЗ "О персональных данных" устанавливает строгие правила обработки любой информации, относящейся к конкретному физическому лицу. Этот закон является краеугольным камнем в защите частной жизни и требует от операторов данных (в том числе телекоммуникационных компаний) принятия целого ряда мер. В свою очередь, Федеральный закон №149-ФЗ "Об информации, информационных технологиях и о защите информации" закрепляет общие принципы обращения с информацией и определяет требования к безопасности информационных систем. В совокупности эти два закона требуют от телекоммуникационных систем обеспечения защиты от несанкционированного доступа, предотвращения утечек информации, шифрования всех чувствительных данных перед передачей и ограничения сроков хранения данных.

Федеральный закон №126-ФЗ "О связи" регулирует деятельность операторов связи и определяет требования к качеству предоставляемых ими услуг. В контексте системы расчета это означает, что системы, обеспечивающие автоматизированную обработку данных, должны быть устойчивы к различным видам сбоев, включая отказы оборудования или перегрузки сети. В частности, данный закон требует обеспечения работоспособности в условиях отказа оборудования и устойчивость к перегрузкам.

Федеральный закон №223-ФЗ "О закупках товаров, работ, услуг отдельными видами юридических лиц" устанавливает правила закупочной деятельности для организаций, которые не подпадают под действие Федерального закона №44-ФЗ. Данный закон влияет на разработку и внедрение любых автоматизированных систем расчета или иных автоматизированных систем, требуя принимать в расчет сертифицированное оборудование, соблюдать необходимые стандарты безопасности а также документировать все процедуры.

Федеральный закон №242-ФЗ "О внесении изменений в отдельные законодательные акты Российской Федерации" устанавливает требование о том, что обработка персональных данных должна происходить исключительно на территории Российской Федерации. Это означает, что телекоммуникационные системы должны предусматривать отказ обработки при использовании серверов за пределами РФ. Если в цепи обработки данных используются серверы, расположенные за пределами Российской Федерации, то обработка персональных данных должна быть немедленно прекращена.

В свою очередь, для стабильной и регламентированной работы, система должна следовать стандартам, установленным в ГОСТ Р 59162-2020 и ГОСТ Р 52872-2019, а также их влиянию на функциональность и надежность телекоммуникационных систем.

ГОСТ Р 59162-2020 представляет собой государственный комплекс требований, направленный на обеспечение безопасности данных и их надлежащего учета в телекоммуникационных системах. Он устанавливает строгие правила, которые призваны защитить информацию от несанкционированного доступа, утечек и искажения. Для соответствия этим требованиям, система должна быть оснащена целым рядом механизмов и средств, обеспечивающих надежную защиту и прозрачность процессов обработки данных. В частности, для соответствия стандарту ГОСТ Р 59162-2020 система должна обеспечивать надежную авторизацию абонентов с использованием криптографических алгоритмов, логирование всех операций для отчетности и мониторинга и модульную архитектуру для интеграции с различными сервисами.

ГОСТ Р 52872-2019 представляет собой стандарт, устанавливающий общие требования к сетям связи, охватывая аспекты, связанные с качеством предоставляемых услуг, производительностью и надежностью. Он определяет параметры, которые являются критически важными для обеспечения удовлетворенности пользователей и эффективного функционирования сетей связи. В соответствии с ГОСТ Р 52872-2019, телекоммуникационная система должна обеспечивать высокое качество связи, включая минимальную задержку, высокую пропускную способность и устойчивость к перегрузкам, детализированные отчеты которые соответствуют стандартам учета и биллинга, а также соответствие всем актуальным нормативным актам которые регулируют деятельность в сфере телекоммуникаций.

### 3.2 Описание и анализ существующих систем

Перед началом проектирования следует проанализировать существующие решения, которые уже есть на рынке. Проводя исследование уже существующих программ и систем, направленных на автоматизацию расчетов, можно сделать вывод о необходимости покупки новой АТС и АСР со своими требованиями. Также изучив преимущества и недостатки конкурентов, можно предусмотреть, что необходимо переработать, что можно включить в систему, а что отложить на будущее или вовсе исключить из стадии проектирования. Таким образом в данном разделе будут рассматриваться существующие системы АТС и АСР. Начиная с малого, стоит дать более конкретное определение АСР.

АСР – Автоматизированная система расчета для операторов связи; Это программное обеспечение, подготовленное для работы телекоммуникационных и интернет-провайдеров. Основными предложениями на рынке являются типовые решения от разработчиков, однако свои услуги также предлагают компании, которые разрабатывают индивидуальные продукты (например Битрикс24). Такие решения характеризуются несколькими недостатками. Один из них – весьма высокая стоимость, поскольку для персональной разработки нужно привлекать немалые ресурсы. Также индивидуальные автоматизированные системы расчета характеризуются узкой направленностью, что иногда становится минусом а не плюсом. АСР за услуги электросвязи являются самостоятельными изделиями, применяемыми операторами связи, и, на основании ст. 16 Федерального закона «О связи» подлежат сертификации «Электросвязь». Поэтому, выбирая АСР, требуется посмотреть наличие сертификата, дабы избежать административного наложения штрафа.

Первой и одной из самых популярных АСР является «Carbon Billing 5». Это мощное программное решение, способное обработать данные от 1,5 млн абонентов. Она обладает расширенной функциональностью. Разработчики постоянно совершенствуют продукт. Уже сейчас данная система имеет свыше 300 функций, и поддерживает десятки моделей роутеров. Биллинг от «Carbon» используется предприятиями, которые предоставляют доступ в интернет при помощи спутника, внедряют канальный и беспроводной доступ и многое другое. Также, Carbon Billing 5 имеет много готовых решений для различных ситуаций, быструю миграцию с принятых компанией решений и полноценную документацию вместе с поддержкой от партнеров. Однако это коммерческая платформа, что означает высокую стоимость лицензии и обслуживания, особенно для малых и средних провайдеров, а также возможные дополнительные затраты на модули и интеграции. Интерфейс системы сложен, требует обучения персонала и имеет множество настроек, что может запутать новых сотрудников. Обслуживание системы также может потребовать квалифицированного персонала. Поскольку

система проприетарная, существует зависимость от разработчика в плане обновлений, поддержки и внесения изменений, что можно привести к задержкам исправления ошибок или реализаций новых функций. Возможности кастомизации ограничены рамками платформы, а интеграция со сторонними системами может быть практически нереальной или очень ресурсоемкой, особенно при больших охватах данных. Также существует риск её устаревания в более короткие сроки, нежели в заявленные разработчиком.

Следующая АСР на очереди – «pgBill». Это OpenSource биллинговая система. Она предназначена для автоматизации расчетов с абонентами и пользователями на общественных сетях связи, обеспечивает информационную поддержку взаиморасчетов между операторами связи, обработку информации об оказанных услугах связи для 2 млн. абонентов. Система предназначена для автоматизации расчетов по местной, внутризонавой и междугородней телефонной связи в выделенной сети, передаче данных, телематические услуги и предоставление каналов связи. Также её функционал вполне достаточен и продолжает улучшаться с каждым обновлением. Однако, по причине открытости данной системы, её минусы отличаются от коммерческих решений. Для её развертывания и настройки требуются технические навыки, а также углубленное знание Linux, SQL и других технологий. Необходим штат квалифицированных специалистов для поддержки и развития системы, что может быть затратно. В стандартной поставке может не хватать некоторых продвинутых функций, которые есть в коммерческих системах, а для получения нужного функционала может потребоваться доработка или использование дополнительных модулей. Отсутствует гарантированная коммерческая поддержка, приходится полагаться на сообщество или нанимать специалистов. Масштабирование системы также может потребовать дополнительных усилий и ресурсов, и нужно самостоятельно следить за безопасностью системы и вовремя устанавливать обновления. Интерфейс не всегда интуитивно понятный, и может потребоваться время на его изучение. Развитие системы зависит от активности сообщества разработчиков.

Крайней АСР будет являться «Поток». Она представляет собой полнофункциональную сертифицированную систему для автоматизации предприятий, предоставляющих различные виды услуг связи на сетях до 50 млн. абонентов. «Поток» является конвергентной биллинговой системой и поддерживает тарификацию различных услуг связи. Система поддерживает пакетное предоставление услуг, а также различные типы телекоммуникационного оборудования и широкую номенклатуру источников об оказанных услугах связи. Данная система может быть дорогой, особенно для крупных провайдеров, а стоимость зависит от выбранных модулей и тарифа. Интерфейс может показаться перегруженным и не интуитивно понятным для новых пользователей, а настройка различных параметров и тарификаций

сложной. Гибкость системы не всегда достаточна для индивидуальных требований, и интеграция со сторонними системами может быть ограничена. Качество и оперативность технической поддержки могут варьироваться, а при больших нагрузках могут возникать проблемы с производительностью и стабильностью. Дополнительные затраты на обучение персонала, внедрение и сопровождение также возможны.

И, наконец, АСР LanBilling - это коммерческая система автоматизации биллинга. Она отличается высокой стоимостью лицензий и технической поддержки, зависящей от количества абонентов, модулей и выбранных услуг. Настройка и администрирование системы сложны и требуют квалифицированного персонала. Интерфейс может показаться перегруженным и не всегда интуитивно понятным. Как коммерческий продукт, LanBilling зависит от политики и планов разработчика, а возможности кастомизации могут быть ограничены. Интеграция со сторонними системами может быть сложной. Система ресурсоемка и требует мощного оборудования для обеспечения стабильной работы. При большой нагрузке или некорректной настройке могут возникать проблемы с производительностью и стабильностью. Внедрение системы может быть длительным и трудоемким процессом, а миграция с другой биллинговой системы на LanBilling может быть сложным процессом.

Все вышеуказанные системы объединяет надежность, популярность и их производительность, а также возможность несложной интеграции, но также каждая из АСР имеют и свои существенные минусы для компании.

Теперь о АТС. Помимо ECSS-10, которую компания ООО «ПортТелеком» и хотела бы приобрести, рассмотрим также Asterisk и FreeSWITCH, которые являются популярными Open Source решениями. Asterisk, например, обладает огромной гибкостью и широкими возможностями, но его настройка требует значительных технических знаний, а для крупных систем могут потребоваться серьезные доработки и оптимизации. FreeSWITCH также отличается гибкостью и масштабируемостью, но его настройка может быть сложнее, чем у Asterisk, и он тоже требует глубоких технических знаний и опыта. Оба решения могут быть бесплатными в плане лицензии, но требуют затрат на специалистов и ресурсы.

В отличие от них, АТС ECSS-10 (от компании "Элтекс") — это коммерческая платформа. Она имеет высокую стоимость лицензий и поддержки, особенно для крупных организаций. Дополнительные расходы могут быть связаны с модулями, интеграцией и расширением функционала. Настройка и администрирование сложны, требуют высокой квалификации IT-специалистов. Интерфейс системы сложен для неподготовленных пользователей, а настройка различных функций и маршрутизаций может быть нетривиальной. Существует зависимость от разработчика в плане обновлений, поддержки и исправления

ошибок. Гибкость системы ограничена, и интеграция со сторонними системами может быть сложной. Система ресурсоемка, требует мощного оборудования, а масштабирование для обслуживания большого количества абонентов может быть проблематичным. Персоналу потребуется специализированное обучение, а обновления системы могут быть сложными и рискованными. Однако, несмотря на эти минусы, ECSS-10 обеспечивает более предсказуемую и надежную работу в корпоративной среде по сравнению с Open Source решениями. Её ориентированность на корпоративный сегмент, профессиональная поддержка и документация делают её предпочтительной для многих организаций, которые ценят стабильность и предсказуемость работы.

В заключение, выбор системы зависит от конкретных потребностей, бюджета и технических возможностей организации. Carbon Billing 5 и Поток имеют преимущества в функциональности, но они дорогие, сложные и зависимые от разработчиков. PgBill бесплатная и гибкая, но требует технических навыков и самостоятельной поддержки. Asterisk и FreeSWITCH могут быть хорошими вариантами для гибких и настраиваемых решений, но требуют квалифицированных специалистов. LanBilling и ECSS-10, хотя и имеют свои минусы, являются оптимальными для интеграции в нашем случае, обеспечивая надежность и необходимый функционал для предоставления качественных телекоммуникационных услуг. Перед принятием окончательного решения по интеграции необходимо провести тщательное тестирование и проанализировать все нюансы.

Таким образом, для нашей интеграции мы выбираем связку LanBilling и ECSS-10. LanBilling, несмотря на свою сложность и стоимость, является наиболее подходящим решением для биллинга услуг связи, учитывая её специализацию и развитый функционал. АТС ECSS-10, хотя и имеет высокую стоимость и сложность настройки, обеспечивает стабильность, предсказуемость и качественную техническую поддержку, что делает её предпочтительной для использования в корпоративной среде по сравнению с Open Source вариантами. Выбор в пользу этих систем обусловлен необходимостью в надежной и проверенной платформе для предоставления телекоммуникационных услуг, даже при учете их минусов.



### 3.3 Определение уже существующей системы

Для начала следует разобрать аббревиатуру AAA. AAA – Это аббревиатура, которая расшифровывается как Authentication, Authorization, and Accounting. Это концепция, которая является основой для управления доступом и отслеживания использования ресурсов в компьютерных сетях. Authentication (Аутентификация): Проверка личности пользователя. Подтверждение того, что пользователь именно тот, за кого себя выдает. Authorization (Авторизация): Определение прав и привилегий пользователя. Какие ресурсы и функции он может использовать после успешной аутентификации. Accounting (Учет): Отслеживание использования ресурсов пользователем. Сбор информации о сессиях, трафике, времени подключения и других параметрах.

RADIUS (Remote Authentication Dial-In User Service) – это сетевой протокол, который реализует AAA для централизованного управления доступом к сетевым ресурсам. Он используется для аутентификации, авторизации и учета пользователей, подключающихся к сети, например, через Wi-Fi, VPN или коммутируемое соединение. Принцип работы Radius не так сложен, когда пользователь пытается получить доступ к сети, сетевое устройство (например, точка доступа Wi-Fi, VPN-сервер или маршрутизатор) отправляет запрос аутентификации Radius-серверу. Далее идет проверка подлинности, Radius-сервер проверяет учетные данные пользователя (например, логин и пароль) по базе данных или с помощью других методов аутентификации. Следующим шагом идёт ответ на запрос. Если аутентификация прошла успешно, сервер отправляет ответ с информацией о правах доступа пользователя (авторизация), а если аутентификация не удалась, сервер отклоняет запрос. Во время сеанса Radius-сервер собирает информацию об использовании сети пользователем (например, время подключения или объем трафика). Эти данные используются для учета биллинга.

Основными компонентами Radius являются Radius-клиент – сетевое устройство, которое взаимодействует с Radius-сервером, Radius-сервер – централизованный сервер для обработки запросов аутентификации, авторизации и аккаунтинга, и базы данных с информацией об учетных записях пользователей, включая логины и пароли а также права доступа.

Преимуществами на фоне других протоколов у Radius можно выделить следующие аспекты:

- централизованное управление;
- усиленная безопасность;
- масштабируемость;



- учет и биллинг;
- стандартизация.

Недостатками в свою очередь будут являться:

- сложность настройки;
- отказоустойчивость;
- проблемы с производительностью

На данный момент существуют и другие протоколы, которые в свою очередь имеют новые или более широкие возможности, но они используются не в тех сетях, в которых производится деятельность компании. Потому такие альтернативы протоколов, как Diameter и TACACS+ не имеют на данный момент никакой материальной пользы в контексте замены или улучшения протокола Radius, соответственно, сама интеграция всесторонне обязана содержать в себе требования только для данного протокола.

Таким образом можно сделать вывод, что ныне активная система, построенная вокруг протокола Radius по принципу AAA является основополагающей технологией системы, и требует интеграции автоматизированной системы расчетов и автоматической телефонной станции под конкретные цели и задачи, и с заранее прописанными входными условиями и критериями, которые могут напрямую повлиять на работу системы.



Функциональные требования для интеграции ACP LanBilling и ATC ECSS-10 должны обеспечивать эффективную и надежную работу объединенной системы. Ключевым является управление абонентами и их услугами, что включает автоматическую синхронизацию данных абонентов между LanBilling и ECSS-10, а также возможность через LanBilling подключать, отключать, приостанавливать и изменять услуги, предоставляемые через ECSS-10, такие как активация/деактивация номеров, изменение тарифов и управление дополнительными сервисами. Необходимо, чтобы в LanBilling отображался полный профиль абонента, включая активные услуги, тарифы, историю платежей и вызовов. В части тарификации и биллинга требуется автоматическая регистрация всех вызовов ECSS-10 с подробной информацией, применение тарифных планов из LanBilling к этим вызовам, автоматическое формирование счетов, поддержка различных моделей тарификации, как предоплаты, так и постоплаты, возможность применения скидок и акций, а также формирование отчетов по использованию услуг.

Управление номерами и ресурсами должно включать управление выделенными номерами, ресурсами ATC (например, SIP-транки, голосовые каналы) через LanBilling, а также динамическое выделение ресурсов в зависимости от нагрузки. Мониторинг и управление системой должны предоставлять мониторинг в реальном времени статуса ATC ECSS-10, отслеживание транзакций, оповещение администраторов о проблемах и возможность централизованного управления через интерфейс LanBilling. Интеграция и взаимодействие должны обеспечивать бесшовную работу между LanBilling и ECSS-10, использование API, поддержку необходимых протоколов, отказоустойчивость и ведение подробных логов. Безопасность системы должна включать защищенную передачу данных, контроль доступа, а также надежную аутентификацию и авторизацию.

Дополнительные функциональные требования могут включать интеграцию с CRM-системами, платежными системами, автоматизацию процессов, например, отключение за неуплату, и интеграцию с SMS-шлюзом. Ключевыми аспектами функциональных требований являются автоматизация для снижения ручного труда, надежность и стабильность работы, масштабируемость при увеличении абонентов, гибкость настройки под специфику организации, и удобство использования системы. Эти функциональные требования следует адаптировать под конкретные нужды вашей организации, учитывая текущие и будущие потребности бизнеса, а также технические возможности обеих систем.

### Интеграция с Radius-сервером:

- АСР должна иметь возможность получать от RADIUS-сервера данные о начале и окончании сессий пользователей, а также информацию об использованных ими ресурсах (например, время подключения, объем переданного трафика). Для этого АСР должна поддерживать стандартные RADIUS атрибуты, а также иметь гибкую настройку для работы с нестандартными атрибутами;
- АСР должна корректно обрабатывать поступающие от RADIUS-сервера данные, включая разбор RADIUS пакетов, декодирование атрибутов и сохранение их в структурированном виде в своей базе данных;
- АСР должна обрабатывать различные типы RADIUS запросов, включая Accounting-Request, Accounting-Response и другие, связанные с учетом сессий и ресурсов;
- АСР должна поддерживать RADIUS CoA, позволяя изменять параметры сессии пользователя (например, скорость или доступ к определенным ресурсам) в реальном времени.

### Учет тарификации и ценообразование:

- АСР должна предоставлять интерфейс для создания, редактирования и управления различными тарифными планами, учитывающими различные параметры (например, время, объем трафика, тип услуги);
- АСР должна поддерживать гибкие механизмы тарификации, позволяющие устанавливать различные цены в зависимости от времени суток, дня недели, объема использованного ресурса и других параметров;
- АСР должна автоматически рассчитывать стоимость услуг на основе данных учета и выбранного тарифного плана, предоставляя детализированные счета для пользователей;
- АСР должна поддерживать возможность предоставления скидок и льгот отдельным категориям пользователей или на определенные виды услуг.

#### Биллинг и формирование отчетов:

- АСР должна автоматически формировать счета для пользователей на основе данных учета и тарификации;
- АСР должна предоставлять возможность гибкой настройки формата счетов, позволяя включать необходимую информацию и логотипы компании;
- АСР должна поддерживать различные способы оплаты, включая электронные платежи, банковские переводы, и другие;
- АСР должна предоставлять широкий набор отчетов, позволяющих анализировать использование ресурсов, доходы и расходы, а также другую информацию, необходимую для принятия управленческих решений.

#### Управление абонентами и их данными:

- АСР должна предоставлять интерфейс для управления абонентскими данными, включая добавление новых пользователей, изменение их данных и удаление;
- АСР должна обеспечивать разграничение прав доступа для различных категорий пользователей и администраторов системы;
- АСР должна иметь возможность интеграции с другими системами, такими как CRM или ERP, для обмена данными о пользователях и услугах.

#### Безопасность:

- АСР должна обеспечивать защиту от несанкционированного доступа к данным и функциям системы;
- Передача данных между АСР и RADIUS-сервером должна осуществляться по защищенным каналам с использованием шифрования;
- АСР должна вести журнал всех действий пользователей и администраторов системы, что позволяет отслеживать изменения и выявлять потенциальные проблемы.

#### Производительность и масштабируемость:

- АСР должна обрабатывать большое количество данных и транзакций с высокой скоростью, обеспечивая стабильную работу системы;
- АСР должна быть масштабируемой, позволяя увеличивать количество пользователей и услуг без снижения производительности системы;
- АСР должна обеспечивать отказоустойчивость, минимизируя время простоя системы в случае сбоев.

## 4.2 Определение необходимых данных для работы системы

Следующим шагом нужно определить данные, которые будут необходимы для корректной работы АСР, которая в свою очередь подключена к AAA Radius и АТС. Также обязательным условием будет то, что АСР должна правильно считывать всю информацию, и такой информации нужно несколько видов.

Во-первых, ей нужно знать, кто и как пользуется сетью. Эта информация приходит от RADIUS-сервера. Он говорит системе, кто подключился к сети, какой IP-адрес он получил, сколько времени он был онлайн и сколько данных скачал или отправил. Также системе нужно знать, какие цены установлены на разные услуги. Эти цены хранятся в тарифных планах, где указано, сколько стоит минута звонка, трафик из одного города в другой, или какая-то другая услуга. В тарифном плане может быть учтено время суток, когда стоимость услуг меняется, или какие-то специальные условия, как скидки для постоянных клиентов. И соответственно, системе нужны данные о самих пользователях, чтобы знать, кому какой тарифный план назначить и кому выставлять счета. В этой базе хранятся логины пользователей, их имена, электронная почта, номера телефонов и другая контактная информация. Кроме того, там же хранится информация об их текущем балансе, если используется предоплатная система.

Во-вторых, система должна хранить информацию о платежах, чтобы понимать, кто и сколько заплатил. Система должна иметь детальную информацию о каждом сеансе пользователя. Когда сессия началась, когда закончилась, сколько длилась, сколько трафика было использовано и сколько все это стоит в итоге. Это нужно для формирования детализированных счетов и для анализа. Помимо этого, нужны данные о выставленных счетах – номер счета, кому выставлен, за какой период, какую сумму нужно оплатить и какой статус оплаты. И, наконец, системе нужны данные для формирования отчетов – чтобы видеть общую картину, анализировать использование ресурсов и доходы. Вся эта информация будет храниться в базе данных, организованной так, чтобы ее было легко и быстро находить и обрабатывать.

## 5 Проектирование интеграции

### 5.1 Протекающие процессы

При проектировании системы автоматизированного расчета важно учитывать то, как происходит процесс обмена информацией между пользователями, АТС и биллингом в промышленной среде. Взаимодействие биллинга с АТС программе будет понять просто, но цепи, проходящие от условных пользователей к условным адаптерам, которые отправляют запрос из точки «А» в точку «Б» и далее, нужно заложить в программу дабы избежать сбора данных с ошибками во время передачи информации между пользователями (рис.2) Также примеры диаграмм процессов обрыва соединений и схемы связей учетных записей пользователей находятся в Приложении А, Приложении Б, Приложении В.

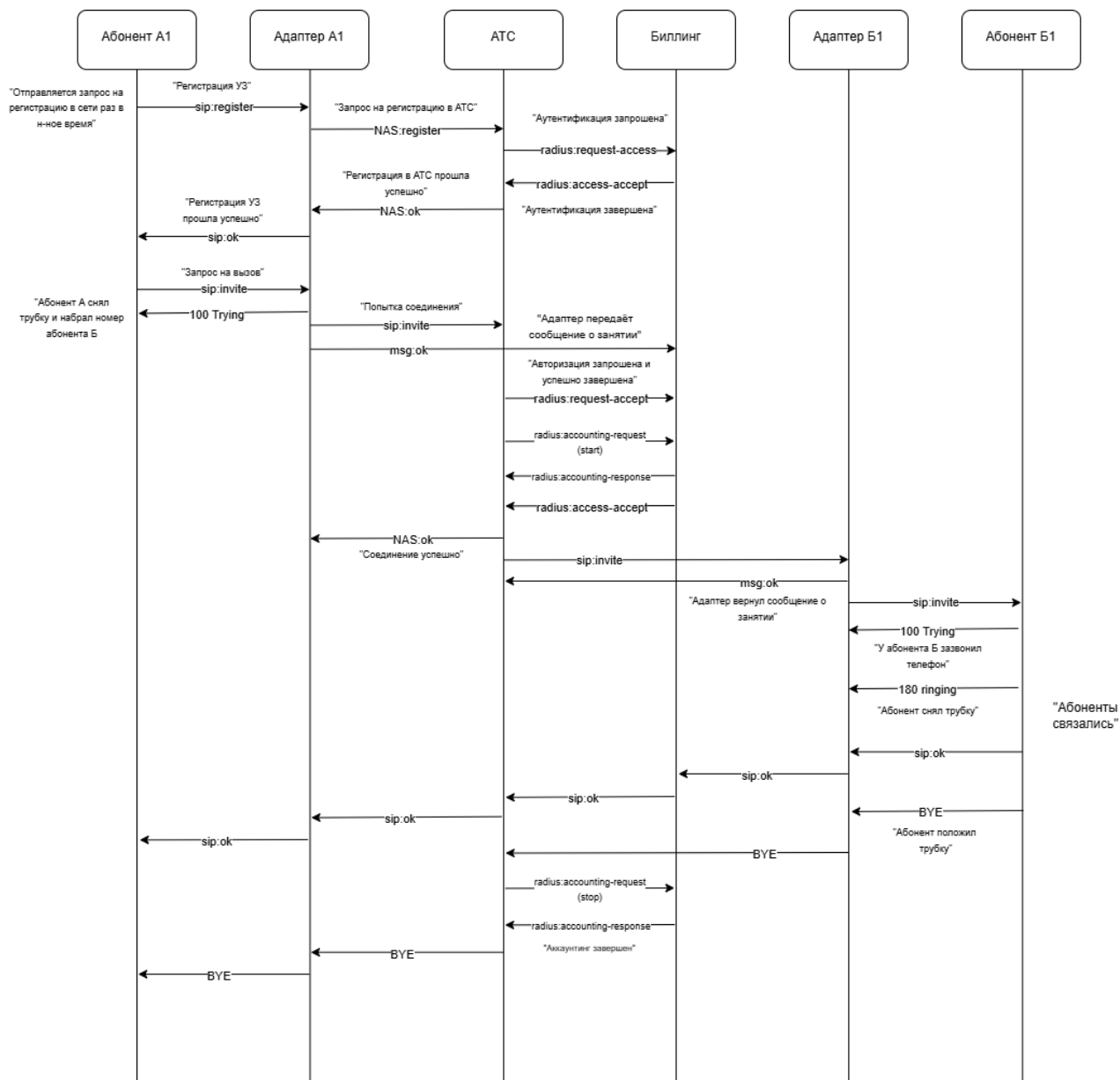


Рисунок 2 – Диаграмма процесса прохождения вызова в одну сторону

После завершения любого из процессов, адаптер, получая ответ о завершении передачи сообщения, отправляет запрос биллингу на окончание аккаунтинга, после чего на АСР должен приходиться отчет в виде CRM-документа со всей информацией, прошедшей через узел связи между пользователями. Такие файлы, как правило, включают в себя абсолютно всю информацию без исключения.

## 5.2 Выбор языка программирования

Для проектирования интеграции важно выбрать язык программирования, позволяющий в полной мере реализовать все функции, которые должна включать в себя АСР, и обеспечить непрерывную связь между АТС и АСР, при этом соответствуя всем стандартам и требованиям, предъявляемым системе. Работа с протоколом Radius требует надежности и производительности, а также гибкости и крайней безопасности всех процессов. Возможность интеграции с уже существующей системой и наличие достаточного инструментария и документаций позволяет сильно облегчить разработку по причине уже готовых наработок и материалов.

Анализ языков программирования для интеграции АТС и АСР показывает, что, хотя существует несколько подходящих вариантов, JavaScript с использованием Node.js выделяется как наиболее оптимальный. Рассмотрим несколько языков: JavaScript - это интерпретируемый, высокоуровневый язык, изначально предназначенный для веб-разработки, но с эволюцией Node.js стал универсальным для фронтенда и бэкенда, а также мобильной разработки. Преимущества JavaScript включают в себя широкую распространенность, гибкость и универсальность, возможность использования в браузере и на сервере благодаря Node.js, неблокирующий ввод-вывод, большое количество библиотек и фреймворков, относительную простоту обучения и хорошую поддержку REST API. К минусам можно отнести интерпретируемость, которая может влиять на производительность в некоторых случаях, сложность работы с асинхронным кодом для начинающих и динамическую типизацию, которая может приводить к ошибкам во время выполнения, хотя использование TypeScript нивелирует эти недостатки. Python - это интерпретируемый язык с чистым и понятным синтаксисом, большим сообществом и множеством библиотек для различных задач, включая веб-разработку. Преимущества Python заключаются в читаемости, простоте, быстрой разработке и хорошей производительности, однако он имеет глобальную блокировку интерпретатора (GIL) и меньшую распространенность во фронтенд-разработке. Java - это компилируемый, объектно-ориентированный язык с высокой производительностью, надежностью и широким использованием в корпоративном секторе. Java имеет большое количество библиотек и фреймворков, но может быть сложнее в освоении, имеет



многословный синтаксис и меньшую гибкость, чем JavaScript или Python. Go (Golang) - это компилируемый, статический язык с высокой производительностью, простотой, поддержкой конкурентности и хорошей поддержкой облачных технологий, но имеет меньшее сообщество и количество библиотек, и ограниченно используется в веб-фронтеде.

Для интеграции АТС и АСР, особенно с учетом создания API и обмена данными, JavaScript (Node.js) является оптимальным выбором. Его универсальность позволяет создавать и бэкенд и фронтенд компоненты, что позволяет покрыть все аспекты интеграционного решения одним языком. Node.js предоставляет мощные инструменты для работы с сетью, файловой системой и базами данных, идеально подходящие для создания интеграционных сервисов и позволяя обрабатывать множество одновременных соединений, что критично для интеграции с АТС и АСР. Кроме того, JavaScript прекрасно подходит для создания RESTful API, и имеет большое сообщество и множество библиотек и фреймворков, которые упрощают разработку, управление данными и асинхронное взаимодействие. JavaScript обеспечивает быструю разработку и позволяет эффективно обрабатывать параллельные запросы, а также относительно прост в освоении, что делает его идеальным вариантом для интеграционного решения. Таким образом, несмотря на то что Python, Java и Go имеют свои преимущества, для интеграции АТС и АСР, JavaScript (Node.js) является наиболее подходящим выбором из-за его универсальности, гибкости, развитой экосистемы, производительности и возможности быстрого прототипирования, обеспечивая создание эффективного, надежного и масштабируемого решения.

В результате анализа выбор был сделан в пользу JavaScript, который предоставит оптимальное сочетание производительности, гибкости и удобства, а также надежности и необходимой точности настройки. Помимо этого, как вспомогательный язык, был выбран Python, который, несмотря на несколько меньшую производительность, компенсирует это гибкостью, простотой, скоростью разработки, развитой экосистемой и простотой масштабирования, что делает его отличным инструментом для разработки дополнительных модулей, работающих с сетевыми протоколами, особенно при ограниченных сроках. Вспомогательный язык был выбран по нескольким причинам. В перспективе доработки будущего кода на JavaScript, возможно потребуется интегрировать грамотный интерфейс к коду, что Python способен сделать в приемлемом и простом для удобства виде.

### 5.3 Описание компонентов системы

JavaScript, работающий в среде Node.js, играет ключевую роль во взаимодействии компонентов интеграции АТС и АСР. В качестве API шлюза, Node.js с использованием фреймворков, таких как Express.js или Koa.js, создает RESTful API для обмена данными между АТС и АСР. JavaScript обрабатывает входящие запросы, маршрутизирует их к соответствующим сервисам, преобразует форматы данных, обеспечивает авторизацию и аутентификацию, а также ведет логирование и мониторинг.

Для модуля синхронизации данных, Node.js обрабатывает и преобразовывает данные об абонентах, услугах и тарифах, получаемые через API от АТС и АСР, и обеспечивает их синхронизацию, используя такие инструменты, как Kafka или RabbitMQ для обмена сообщениями и базы данных для хранения состояний. В модуле обработки вызовов (CDR), JavaScript получает данные о вызовах из АТС, часто в формате RADIUS или через API, преобразует эти данные в формат, понятный АСР, и отправляет их для тарификации, применяя при этом асинхронные операции для обработки большого объема данных.

JavaScript также используется в модуле мониторинга и управления для использования API для сбора и обработки метрик, отправки уведомлений об ошибках и ведения логов, используя такие инструменты, как Prometheus, Grafana и ELK stack, а также предоставляет интерфейсы для управления конфигурацией. В целом, JavaScript обеспечивает взаимодействие между всеми компонентами интеграции, используя HTTP протокол для коммуникации между сервисами, асинхронность для параллельной обработки запросов, и гибкость для создания необходимых адаптеров данных, что делает его ключевым языком для построения эффективной, масштабируемой и надежной интеграции АТС и АСР

Архитектура интеграции выглядит следующим образом (рис.3)

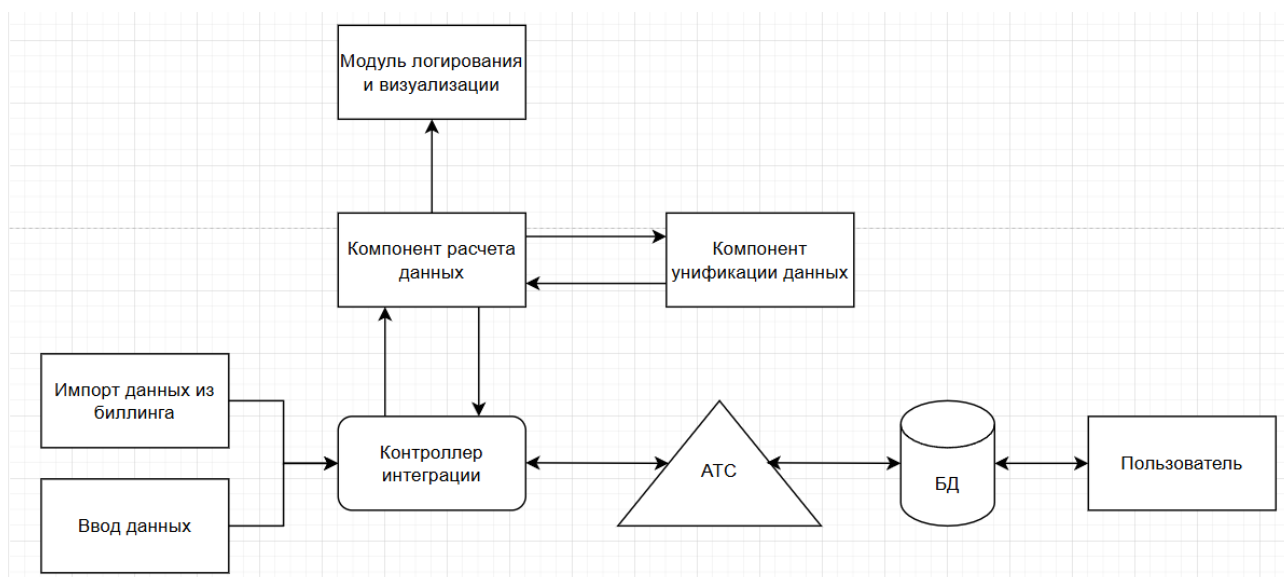


Рисунок 3 – Архитектура разрабатываемой интеграции

На данном рисунке можно увидеть путь информации, проходящей от начала её использования и заканчивая выводом сразу к пользователю в руки. Такая архитектура сократит излишние движения поступающей из CRM-документов, а также позволит изменять или упорядочивать иным способом информацию в базе данных путём её прямого изменения, если того требует ситуация, например, во время звонка произошел сбой на АТС, и с помощью модуля логирования мы сможем отследить ошибку в информации в реальном времени, а прямой доступ к базе данных позволит сразу изменить всю ошибочно собранную информацию на корректную, и избежать ошибок в работе других модулей. Помимо этого, было решено вынести компонент унификации отдельно от компонента расчета данных в пользу производительности, так как два компонента, которые выполняют совершенно разную работу, будут занимать слишком много как сетевых ресурсов, так и ресурсов оборудования, что может повлечь за собой перегрузку системы, если данных поступит слишком много в один момент, или же к утере данных, если произойдет какой-либо сбой в работе биллинга или АТС.

## Заключение

Создание интеграций в телефонии является важным и актуальным направлением в современной сфере телекоммуникационных связей. В условиях постоянного развития потребительских предпочтений, сетевых инфраструктур и качества продукции услуг, подобные разработки приобретают особое значение. Системы АСР в современном мире применимы практически ко всем сферам деятельности, что создаёт конкуренцию между разработчиками, потому качество подобных систем должно только расти, и количество интеграций к ним должно также поднимать планку качества на высший уровень.

В рамках данной курсовой работы было выполнено проектирование интеграции АТС и АСР на основе уже существующей системы, которая в свою очередь использует AAA Radius как основной протокол. Основная цель проектирования заключается в создании интеграции, способной работать в условиях большой нагрузки и множестве подключенных операторов. Интеграция была спроектирована с учетом возможности расширения функционала и возможностью иных интеграций, а также с учетом перехода всей системы на другой язык или формат данных. Спроектированная интеграция обладает надежностью и гибкостью настройки, а также поддержкой различных сценариев.

При соответствии конечной интеграции требованиям, изложенным в данной курсовой работе, компонент может стать мощным и востребованным инструментом в сфере телекоммуникаций, уменьшив вероятность ошибок, экономических затрат и затрат времени.

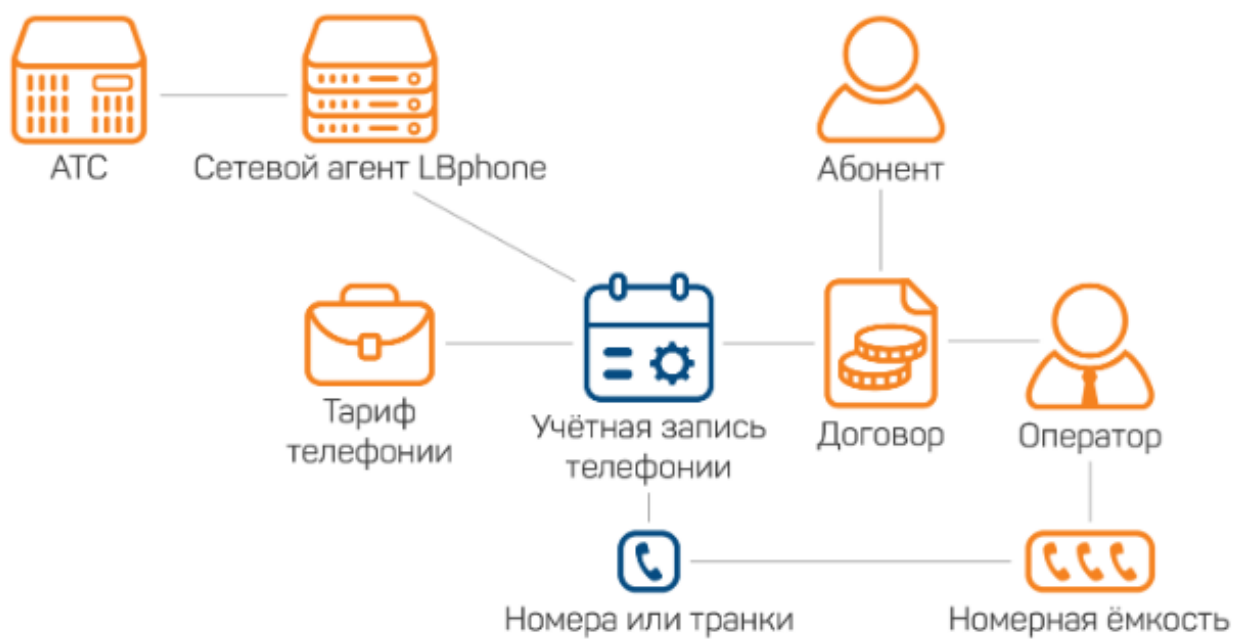
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. API Lanbilling: [сайт]. - URL: <https://www.lanbilling.ru/technical-information/documentation/api/> (дата обращения 06.12.2024).
2. Документация Lanbilling 2.0.49: [сайт]. - URL: <https://docs.lanbilling.ru/48/> (дата обращения 06.12.2024).
3. Документация Eltex: [сайт]. - URL: <https://docs.eltex-co.ru/e15> (дата обращения 10.12.2024)
4. Документация Lanbilling 2.0.46 : [сайт]. - URL: [https://docs.lanbilling.ru/49/accounts/telephony/create\\_account\\_telephony/](https://docs.lanbilling.ru/49/accounts/telephony/create_account_telephony/) (дата обращения 20.12.2024)
5. Как работает виртуальная АТС: [сайт]. - URL: <https://www.mtt.ru/support/blog/kak-rabotaet-virtualnaya-ats/> (дата обращения 03.01.2025)
6. AAA Server Groups: [сайт]. – URL: [https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec\\_usr\\_rad/configuration/xr-16/test/sec-usr-rad-xr-16-book/sec-rad-aaa-server-groups.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_rad/configuration/xr-16/test/sec-usr-rad-xr-16-book/sec-rad-aaa-server-groups.html) (дата обращения 03.01.2025)
7. RFS-2867 Radius Tunnel Accounting: []. – URL: [https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec\\_usr\\_rad/configuration/xr-16/test/sec-usr-rad-xr-16-book/sec-rad-acctg-2867.html](https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/sec_usr_rad/configuration/xr-16/test/sec-usr-rad-xr-16-book/sec-rad-acctg-2867.html) (дата обращения 03.01.2025)
8. Структура АСР: [сайт]. – URL: [https://xn--d1aml.xn--90aeoaoap.xn--p1ai/index.html?ma\\_monitoring\\_system\\_and\\_accident\\_recovery.html](https://xn--d1aml.xn--90aeoaoap.xn--p1ai/index.html?ma_monitoring_system_and_accident_recovery.html) (дата обращения 05.01.2025)
9. Стандарты голосовой связи: [сайт]. - URL: <https://habr.com/ru/companies/mtt/articles/321138/> (дата обращения 05.01.2025)
10. Voip-протоколы: [сайт]. – URL: <https://telcom.pro/voip-protokoly-v-ip-telefoniy/> (дата обращения 05.01.2025)
11. AAA, 802.1X, EAP & RADIUS: [сайт]. – URL: <https://help.ui.com/hc/en-us/articles/115007253447-Intro-to-Networking-AAA-802-1X-EAP-RADIUS> (дата обращения 06.01.2025)
12. RFC-2865: [сайт]. – URL: <https://datatracker.ietf.org/doc/html/rfc2865> (дата обращения 06.01.2025)
13. Как работать с sip: [сайт]. – URL: <https://ru.stackoverflow.com/questions/721295/%D0%9A%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D1%82%D1%8C-%D1%81-sip> (дата обращения 08.01.2025)

14. Нативные библиотеки Java: [сайт]. – URL: <https://github.com/docops-hq/learnapidoc-ru/blob/master/Native-library/Overview-of-library.md> (дата обращения 08.01.2025)
15. Федеральный закон № 152-ФЗ «О персональных данных» от 27.07.2006 (ред. от 06.02.2023): [Электронный ресурс]. - URL: <https://government.ru/> - режим доступа: <https://government.ru/docs/all/98196> (дата обращения 10.01.2025)
16. Федеральный закон № 126-ФЗ «О связи» от 07.07.2003 (ред. от 30.12.2021): [Электронный ресурс]. - URL: <https://www.consultant.ru/> - режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_43224/](https://www.consultant.ru/document/cons_doc_LAW_43224/) (дата обращения 10.01.2025)
17. Федеральный закон № 242-ФЗ "О государственной геномной регистрации в Российской Федерации" от 03.12.2008 (ред. от 08.08.2024): [Электронный ресурс]. - URL: <https://www.consultant.ru/> - режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_482486/](https://www.consultant.ru/document/cons_doc_LAW_482486/) (дата обращения 10.01.2025)
18. Федеральный закон № 223-ФЗ «О закупках товаров, работ, услуг отдельными видами» от 18.07.2011 (ред. от 08.08.2024): [Электронный ресурс]. - URL: <https://www.consultant.ru/> - режим доступа: [https://www.consultant.ru/document/cons\\_doc\\_LAW\\_11454/](https://www.consultant.ru/document/cons_doc_LAW_11454/) (дата обращения 10.01.2025)
19. Национальный стандарт РФ ГОСТ Р 59162-2020: [Электронный ресурс]. - URL: <https://docs.cntd.ru/document/573219431> (дата обращения 10.01.2025)
20. Национальный стандарт РФ ГОСТ Р 52872-2019: [Электронный ресурс]. - URL: <https://meganorm.ru/Data2/1/4293776/> (дата обращения 10.01.2025)

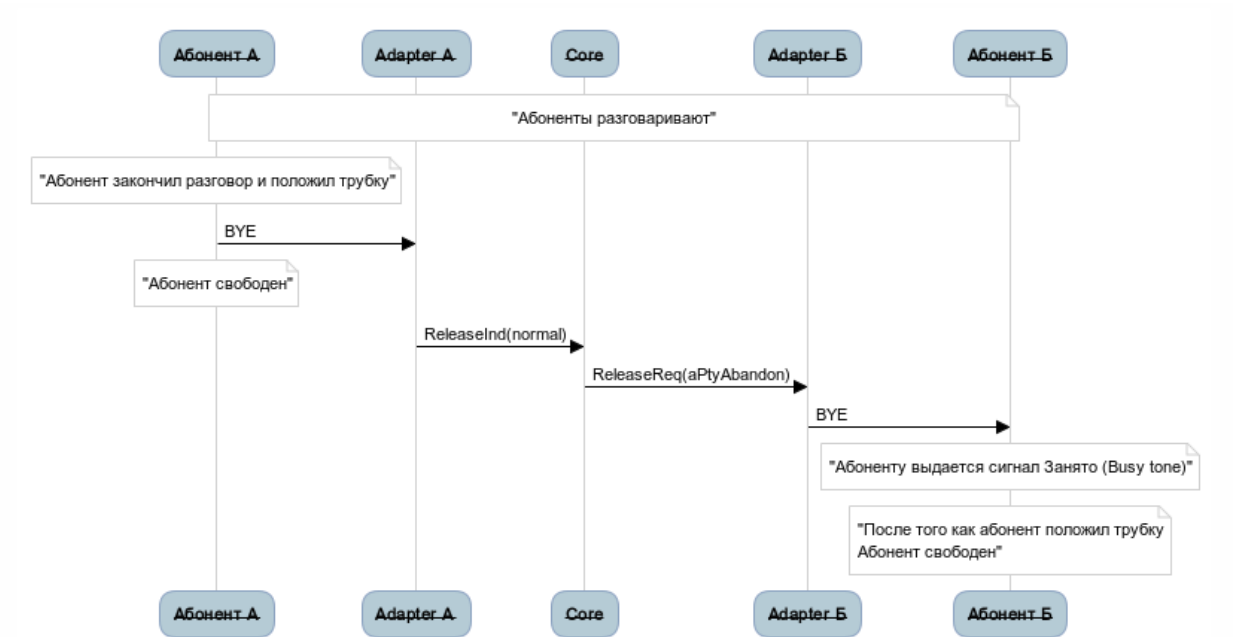
## Приложение А

### Схема связей справочника учетных записей



## Приложение Б

### Разъединение вызова по инициативе абонента «А»





## Приложение В

### Разъединение вызова по инициативе абонента «Б»

