АО "НТЦ ИТ РОСА"

ОПЕРАЦИОННАЯ СИСТЕМА "РОСА МОБАЙЛ"

Версия 3.1

Руководство администратора

Часть 2. Эксплуатация. Сервисы

РСЮК.10401-03 32 02

Листов 46

2026

АННОТАЦИЯ

Данный документ является руководством администратора операционной системы "РОСА Мобайл" (далее – ОС), установленной на мобильные устройства (далее – МУ).

Документ содержит общее описание серверных сервисов и компонентов ОС.

Документ является неотъемлемой частью "Операционная система РОСА Мобайл. Руководство администратора. Часть 1. Эксплуатация" (индекс РСЮК.10401-03 32 01).

Общие характеристики ОС и МУ, порядок первого запуска ОС, элементы рабочего пространства, способы управления и навигации, а также основные жесты и взаимодействие с физическими и экранными кнопками, сведения о настройке параметров МУ, описание работы с приложениями ОС представлены в документе "Операционная система РОСА Мобайл. Руководство пользователя" (индекс РСЮК.10401-03 34 01).

Перед эксплуатацией ОС рекомендуется внимательно ознакомиться с настоящим руководством.

Для разработки документа использованы ссылки на следующие стандарты:

– ГОСТ Р 2.105-2019 "Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам";

– ГОСТ 2.601 "Единая система программной документации. Виды программных документов";

– ГОСТ 19.101-77 "Единая система программной документации. Виды программ и программных документов";

– ГОСТ 19.105-78 "Единая система программной документации. Общие требования к программным документам";

– ГОСТ 19.503-79 "Единая система программной документации. Руководство системного программиста".

Настоящий документ подготовлен в соответствии с технологической инструкцией "РОСА. Регламент формирования документации к программным продуктам" (индекс РСЮК.11001-02 90 01).

СОДЕРЖАНИЕ

1 Общие сведения **4**

1.1 Назначение 4
1.2 Функции 4
1.3 Область применения 7

2 Структура ОС **9**
3 Управление мобильными устройствами **10**
4 Администрирование сервиса РОСА ID **11**

4.1 Общие сведения 11
4.2 Регистрация клиента 11
4.3 Авторизация пользователя 12
4.4 Получение токена 14
4.5 Обновление токена 17
4.6 Список запросов 19
4.7 ID-токен 21
4.8 Ошибки 21

5 Описание сервиса РОСА Почта **23**

5.1 Добавление аккаунта преднастроенного почтового сервера 23
5.2 Добавление аккаунта вручную 25

6 Описание сервиса РОСА Календарь **28**
7 Администрирование сервиса РОСА Мессенджер **31**

7.1 Общие сведения о сервисе 31
7.2 Обеспечение безопасности в РОСА Мессенджер 32

7.2.1 Шифрование данных 32
7.2.2 Сценарии обмена ключами в индивидуальных и групповых чатах . . 32
7.2.3 Использование протокола HTTPS для обеспечения сетевой безопасности 34

7.3 Управление учетной записью 35
7.4 Подключение к РОСА Мессенджер 37
7.5 Доставка push-уведомлений 45

Термины и сокращения **46**

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Операционная система "РОСА Мобайл" – российская мобильная операционная система для смартфонов, планшетов и других мобильных устройств, основанных на архитектуре ARM.

Сертификат ФСТЭК России №4867 от 13.11.2024 действует до 13.11.2029. ОС соответствует требованиям к средствам защиты информации 4-му уровню доверия и к ОС 4-го класса защиты. Внесено в реестр Российского программного обеспечения (Реестровая запись №16453 от 03.02.2023).

1.2 Функции

ОС "РОСА Мобайл" базируется на ядре Linux, что обеспечивает высокую стабильность, надежность и безопасность, поддерживает многозадачность для выполнения нескольких фоновых процессов одновременно, что позволяет эффективно использовать ресурсы устройства, с переключением между ними, управляет распределением ресурсов (ЦП, память, дисковое пространство) между процессами.

ОС обеспечивает следующие системные и прикладные функции, включая, но не ограничиваясь:

- Базовый системный функционал:
  - Пользовательский интерфейс – графический интерфейс пользователя (GUI) с отображением статических и динамических графических элементов, сенсорный ввод, жесты и анимации.
  - Настраиваемость – гибкая настройка пользовательского интерфейса: смена тем, обоев, расстановка ярлыков приложений и прочих элементов пользовательской среды.
  - Управление жестами и навигация – стандартные для Android жесты (свайпы), панель навигации.
  - Уведомления – центр уведомлений с быстрыми настройками (шторка).
  - Настройки – детализированное меню настроек для управления устройством, сетями, экраном, звуком и т.д.
  - Многопользовательский режим-- поддержка нескольких профилей на одном устройстве (гостевой, рабочий).

  - Работа с учетными записями-- поддержка аккаунтов для почты, календаря, облачных хранилищ.
- Безопасность и управление:
  - Мандатное управление доступом (МУД) – разграничение прав доступа к данным и функциям ОС на основе политик безопасности, заданных администратором (ключевая функция).
  - Изоляция приложений – работа каждого приложения в своей изолированной среде ("песочнице") с исключением прямого доступа к данным других приложений без разрешения.
  - Контроль целостности – проверка неизменности системных файлов и компонентов при загрузке.
  - Поддержка контейнеров – использование контейнеризации для запуска приложений в изолированных средах для повышения безопасности и стабильности ОС.
  - Шифрование данных – полное шифрование пользовательских данных на диске.
  - Безопасная загрузка – защищённая процедура включения устройства для предотвращения запуска неавторизованного кода.
  - Центр управления безопасностью – инструмент для сканирования на угрозы, контроля разрешений приложений, управления VPN.
  - Антивирусная защита-- интеграция с отечественными антивирусными решениями.
  - Политики безопасности для предприятий (MDM) – возможность удалённого управления паролями, установкой приложений, блокировки устройств (в корпоративной версии).
  - Шифрование данных – поддержка шифрования данных на уровне файловой системы для защиты пользовательских данных.
  - Многофакторная аутентификация – поддержка различных методов аутентификации (пароль, PIN-код, NFC-токен).
  - Регулярные обновления безопасности – регулярное обновление системы безопасности для защиты от новых угроз и уязвимостей.
- Связь и коммуникации:
  - Телефония – звонки, SMS, MMS, работа с SIM-картами и eSIM.
  - Диспетчер контактов – синхронизация с аккаунтами.
  - Электронная почта – встроенный клиент, поддерживающий российские почтовые сервисы (Mail.ru, Яндекс.Почта и др.).
  - Сообщения – поддержка современных протоколов обмена сообщениями (RCS).

  - Браузер – собственный или предустановленный российский браузер (например, Спутник, Яндекс.Браузер).
- Сети и подключения:
  - Мобильные сети – поддержка 3G, 4G/LTE, 5G (поддержка 5G зависит от аппаратной платформы устройства) для мобильной связи и передачи данных.
  - Беспроводные интерфейсы – поддержка всех современных стандартов Wi-Fi (включая прямой Wi-Fi Direct), Bluetooth (обмен файлами, наушники и пр.) для беспроводного подключения.
  - Геолокация – поддержка ГЛОНАСС, GPS, определение по вышкам сотовой связи и Wi-Fi.
  - NFC – бесконтактные платежи (с российскими сервисами), эмуляция банковских карт.
- Мультимедиа и работа с файлами:
  - Камера и галерея – приложение для фото- и видеосъёмки, просмотра медиафайлов.
  - Аудиоплеер – воспроизведение музыки и аудиокниг.
  - Диктофон – запись голосовых заметок.
  - Файловый менеджер – просмотр, копирование, управление файлами на внутренней и внешней памяти.
  - Виртуальная файловая система – изолированное хранение файлов разных приложений и пользователей.
- Продуктивность и сервисы:
  - Календарь – синхронизация с российскими серверами.
  - Часы – будильник, таймер, секундомер, мировое время.
  - Калькулятор.
  - Погода – виджеты и приложение с данными от российских поставщиков.
  - Голосовой помощник – интеграция с отечественными решениями (например, от Сбер, Яндекс или собственный).
  - Облачное хранилище – интеграция с российскими облачными сервисами (Яндекс.Диск, VK Cloud, Mail.ru Облако).
  - Карты и навигация – предустановленные российские картографические сервисы (Яндекс.Карты, 2ГИС, Геоскан).
- Магазин приложений и экосистема:
  - RuStore – основной официальный магазин приложений.
  - Магазин приложений – поддержка скачивания и установки приложений из магазина приложений РОСА Маркет.

  - Установка APK – возможность установки приложений вручную из APK-файлов (с контролем безопасности).
- Специализированный функционал:
  - Работа с электронной подписью (ЭП) – встроенная поддержка российских для работы с порталом госуслуг, электронным документооборотом.
  - Государственные сервисы – предустановленное приложение "Госуслуги" и другие ведомственные клиенты.
  - Тонкая настройка энергопотребления – возможность увеличения автономной работы.
  - Режим "киоска" или сенсорного терминала – ограничение функционала устройства для использования в качестве информационного стенда или кассы (для корпоративных решений).
- Разработка и совместимость:
  - Поддержка Android-приложений – совместимость на уровне AOSP, работоспособность большинства приложений из RuStore и многих APK.
  - API для разработчиков – стандартные Android API, а также специальные API для доступа к функциям безопасности и МУД.

1.3 Область применения

ОС "РОСА Мобайл" является универсальной специализированной платформой для решения задач, приоритетами которых являются безопасность, контроль и хранение данных внутри страны.

ОС может быть использована как для личного использования, так и государственными органами, компаниями с повышенными требованиями к информационной безопасности для централизованного управления мобильными устройствами.

Настоящее руководство предназначено для использования системным администратором и специалистом по техническому обслуживанию.

Квалификация системного администратора: высокий уровень знаний и наличие практического опыта выполнения работ по установке, настройке и администрированию программных средств, применяемых в Комплексе, а также наличие профессиональных знаний и практического опыта в области системного администрирования.

Квалификация специалиста по техническому обслуживанию: высокий уровень знаний и наличие практического опыта выполнения работ по

установке, настройке и подключению мобильного, компьютерного и серверного оборудования, применяемого в ОС, а также наличие профессиональных знаний и практического опыта в области технического обслуживания.

Основными обязанностями специалиста по техническому обслуживанию являются:

а) модернизация, настройка и мониторинг работоспособности Комплекса технических средств (серверов, рабочих станций);

б) конфигурирование и настройка программно-технических средств Комплекса;

в) диагностика типовых неисправностей.

Предполагается, что пользователь уже обладает базовыми навыками работы с современными мобильными операционными системами.

РОСА

2 СТРУКТУРА ОС

Архитектура ОС "РОСА Мобайл" состоит из следующих основных функциональных компонентов:

- РОСА Центр управления мобильными устройствами;
- ОС мобильных устройств;
- сервис РОСА ID;
- сервис РОСА Мессенджер;
- Календарь;
- Почта.

3 УПРАВЛЕНИЕ МОБИЛЬНЫМИ УСТРОЙСТВАМИ

Начиная с версии 2.2, РОСА Центр управления поддерживает интеграцию с мобильными устройствами (далее – МУ) на ОС "РОСА Мобайл" и обеспечивает управление МУ.

Функционал управления МУ предоставляется в формате отдельного образа установочного диска (дистрибутива) для редакций "Центр управления для управления АРМ и мобильными устройствами" или "Центр управления для управления мобильными устройствами", установка которых описана в п. 3.5 "РОСА Центр управления. Руководство системного администратора. Часть 1. Установка и настройка" (индекс – РСЮК.10121-11 32 01) документа "РОСА Центр управления. Руководство системного администратора. Часть 1. Установка и настройка".

Описание реализованных классов Puppet функций и фактов управления МУ приведено в п. Управление мобильными устройствами документа "РОСА Центр управления. Руководство системного администратора. Часть 2. Эксплуатация" (индекс РСЮК.10121-10 32 02) .

4 АДМИНИСТРИРОВАНИЕ СЕРВИСА РОСА ID

4.1 Общие сведения

Сервис РОСА ID – это единый и безопасный способ входа (учетная запись) в приложения в экосистеме программных продуктов РОСА.

Клиентом в сервисе РОСА ID называется подключенный внешний сервис, взаимодействующий с зарегистрированным пользователем. Авторизация пользователя во внешнем сервисе осуществляется в соответствии со стандартом OpenID.

4.2 Регистрация клиента

Для регистрации клиента необходимо воспользоваться интерфейсом подключения сервиса к РОСА ID. Клиент при этом будет привязан к авторизованному пользователю (рисунок 1).

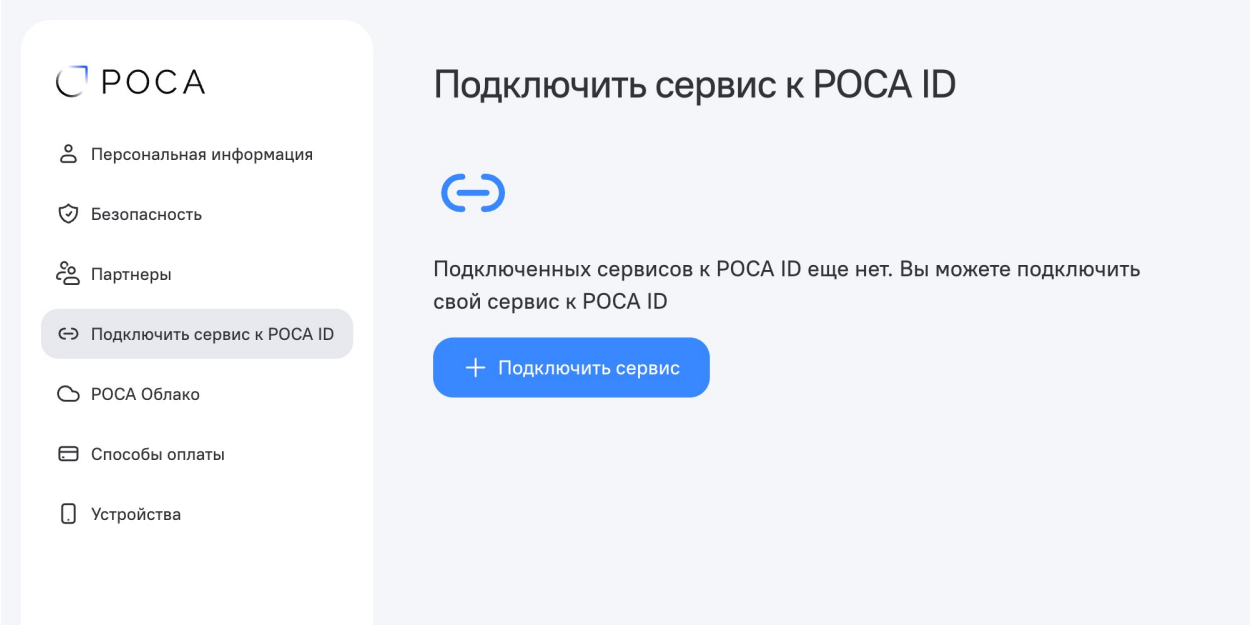

Рисунок 1 – Подключение сервиса к РОСА ID

При создании необходимо указать:

– Наименование – используется для идентификации в интерфейсе приложения;

– Используемые платформы – платформы, на которых планируется использование клиента;

– Redirect URI – URI, на который будет происходить переадресация при попытке авторизации.

После успешной регистрации клиенту будут присвоены Client ID и Client Secret, которые можно посмотреть на странице сведений о сервисе. Значение Client ID является публичным и может передаваться пользователям; значение Client Secret необходимо сохранять в секрете, оно используется для идентификации клиента при запросах к приложению.

4.3 Авторизация пользователя

После успешной регистрации клиента появляется возможность авторизовать пользователя от имени данного клиента. Для этого необходимо направить пользователя по сформированному URL-адресу.

Ссылка для формирования, по которой нужно направить пользователя, желающего авторизоваться с использованием POCA ID, состоит из следующих частей:

– хост – базовый URL приложения POCA ID (например, `https://id.rosalinux.ru`);

– путь – путь до эндпойнта авторизации (всегда `/authorize`);

– query-параметры – параметры для авторизации, состоящие из пар "ключ-значение".

Query-параметры и их возможные значения приведены в таблице 1.

Таблица 1 – Query-параметры

Название параметра	Возможные значения	Описание	Пример
scope	openid name email phone_number birthdate gender address	Список значений, определяющий границы списка возвращаемых полей пользователя, а так же возможных действий, совершаемых над пользователем. Значения списка разделяются пробелом. Описание значений описаны далее	openid email name
response_type	code		code
client_id		Client ID клиента, полученный после регистрации	P000001000
redirect_uri		Redirect URI клиента, указанный при регистрации	https://google.com

Для авторизации включение значения openid в параметре scope является обязательным.

Возможные значения параметра scope приведены в таблице 2.

Таблица 2 – Значения параметра scope

Значение	Поля пользователя в приложении, к которым предоставляется доступ
openid	-
name	name, given_name, family_name, middle_name
email	email, email_verified
phone_number	phone_number, phone_number_verified
birthdate	birthdate
gender	gender
address	address

Пример полной ссылки (URL-encoded):

```
https://id.rosalinux.ru/authorize? scope=openid+email+name&respo
nse_type=code&client_id=P000001000&redire
ct_uri=https%3A%2F%2Fgoogle.com
```

После успешной (или неуспешной) авторизации пользователь будет перенаправлен по установленному Redirect URI. При этом в query-параметрах будет находиться необходимая информация для дальнейшей обработки клиентом.

Список возвращаемых query-параметров:

- При успешной авторизации:
  - code – код авторизации, используемый для дальнейшего получения авторизационного токена, для использования клиентом (процедура описана в следующих разделах);

Пример:

```
https://google.com?code=AAAAAAAAAAAAAAAAAAAAA
```

- При возникновении ошибки:
  - `unsupported_response_type` – значение response_type отличается от единственного поддерживаемого code;
  - `invalid_scope` – запрошено несуществующее значение scope либо отсутствует обязательное для авторизации значение openid;
  - `access_denied` – пользователь отклонил запрос на авторизацию.

Формат возвращаемой ошибки описан в п. 4.8.

Пример:

```
https://google.com? error=invalid_scope&error_description=%22ope
nid%22%20scope%20is%20required
```

4.4 Получение токена

В случае успешного получения кода авторизации клиент может получить авторизационный токен, привязанный к авторизуемому пользователю. В дальнейшем этот токен будет использоваться для всех запросов, касаемых этого пользователя.

Для получения токена клиенту необходимо отправить следующий запрос:

- URL – `api/oauth/client/token`;

- Метод – POST;
- Тип данных – `x-www-form-urlencoded`.

Параметры тела запроса приведены в таблице 3.

Таблица 3 – Параметры теля запроса

Параметр	Возможные значения	Описание
client_id	--	Client ID клиента, полученный после регистрации
client_secret	--	Client Secret клиента, полученный после регистрации
grant_type	authorization_code	Всегда authorization_code
redirect_uri	--	Redirect URI клиента, указанный при регистрации
code	--	Код авторизации, полученный в предыдущем шаге

Пример в формате curl:

```
curl \
--location 'https://id.rosalinux.ru/api/oauth/client/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=P000000002' \
--data-urlencode 'client_secret=9b821de045bf3b55555535\
e84c011cc96217c11b19139cab8786704f9c3ca8b9' \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=AAAAAAAAAAAAAAAAAAAAAA' \
--data-urlencode 'redirect_uri=https://google.com'
```

Ответ может быть возвращен в формате JSON (application/json).

Список полей при успешном запросе приведет в таблице 4.

Таблица 4 – Поля при успешном запросе

Имя поля	Описание
access_token	Авторизационный токен
token_type	Тип полученного токена (всегда Bearer)
refresh_token	Refresh-токен
expires_in	Срок действия авторизационного токена (в секундах)

Имя поля	Описание
scope	Список значений scope. Значения разделяются пробелами. Должен совпадать с запрошенным списком.
id_token	ID-токен, содержащий запрошенные данные пользователя. Представляет из себя зашифрованную JWT-строку.

Формат данных ID-токена описан в п. 4.7.

Пример:


```
{
"access_token":
"e697d038899ccd0204ec41f215e964d63b0fb423d3ff238a1b7632924976804
d",
"token_type": "Bearer", "refresh_token":
"a093eab561a7278548c857b5699ff463f3180c506cec1b4791cf84ff1aaed4a
4",
"expires_in": 86399,
"scope": "email openid",
"id_token":
"eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJyb3NhaWQiLCJzdW
IiOiJh
ZDNlZTI3NS04MzE5LTRhMTMtYmNiMC1iMzI4ZTJjNTZmMDAiLCJhdWQiOiJQMDAw
MDAwMD
cwIiwiZXhwIjoxNzE2ODM0NTgzLCJpYXQiOjE3MTY4MzA5ODMsImVtYWlsIjoidX
NlckBk
dW1teS5tYWlsIiwiZW1haWxfdmVyaWZpZWQiOnRydWV9.hqH_9xYPGK09bKEyYRr
FFmrEc
ZD3Xu-
h6T8M9ffIHJgTwe1nH4XMzgpk0Nax3OUS_XkqNtBSDj1VNWGAgfWrDR8gtCnjFZ3
w50HF3
Qz6w2TljsbrfiA6JeW7_Z7-pz-JNY1-YqpuSs0ir-
cChvJAolmVyYa1pSi4oUyPlvDXq2AhCWXMNm0tSrRlR5iuP3XGjy5aZzD2rymFxB
x4kzD9
Y63hYr6bPSm7GFtZS_gTAguI1VkUt79widOsSlQVE4pqgGekcqHEHn61tECiFN4V
z0S86U
rqgmaHspIPM1tH6Eji49b_A_8rF8Dm9IYB-m0ul-
wuB5g5h4cpPiYf7c4WpJxwzG5wBWKWZv060Ake4sKkW9zBNwdcXrUabYvBL4i1Ng
156j3f
```

```
xN1O1Z3yqF6rfy7Ud7Y0mdXQh65X3s31s3TYC3WNoZkfyIqL-
wHSbQWxxXrdlEoX1JoWEkdTraNZlRORpUFYgruNVgqCBAwXmx7IY6_mWD8Fb4Fk0
JIOwQg
dxBoA5EVy4X71woeEd-TY0NU4DQK9qfrksQCKZr9R4AaXe-7B-
SjMkFPbRHlSDpDyOyUZMWMnT9ta6CG615TyAos3388ZZqbD3GU_xQMJufu7tIYfi
uXlpbw
-226OvBFI6UxIwnlOMKEGkHWDtvZokj0jD9rlLq_Dwq2fxTOUGlU" }
```


Ответ также может возвращать ошибки.

Формат возвращаемых ошибок описан в п. 4.8.

Список ошибок, которые могут возникнуть, приведен в таблице 5.

Таблица 5 – Список ошибок

Название ошибки	Условия возникновения
invalid_request	Неверно сформирован запрос
invalid_grant	Переданный авторизационный код не найден (или не принадлежит этому клиенту)
unsupported_grant_type	Передано неподдерживаемое значение grant_type
server_error	Ошибка на стороне сервера

На основании полученного в ID-токене значения sub клиент может авторизовывать пользователя.

4.5 Обновление токена

После истечения срока полученный токен можно обновить (получить новую пару).

Для этого необходимо отправить следующий запрос:

- URL – api/oauth/client/token;
- Метод – POST;
- Тип данных – x-www-form-urlencoded.

Параметры тела запроса приведены в таблице 6.

Таблица 6 – Параметры тела запроса

Параметр	Возможные значения	Описание
client_id	--	Client ID клиента, полученный после регистрации
client_secret	--	Client Secret клиента, полученный после регистрации
grant_type	refresh_token	Всегда refresh_token
refresh_token	--	Refresh-токен, полученный вместе с авторизационным токеном

Пример в формате curl:

```
curl \
--location 'https://id.rosalinux.ru/api/oauth/client/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=P000000002' \
--data-urlencode 'client_secret=9b821de045bf3b33d83135e84\
c011cc96217c11b19139cab8786704f9c3ca8b9' \
--data-urlencode 'grant_type=refresh_token' \
--data-urlencode 'refresh_token=2de7c7966583ff3a9cf7ae581240\
caf769d1384a0c7cc6a0f7034bacb4d619c3'
```

Ответ возвращается в формате JSON (application/json).

Список полей при успешном запросе приведен в таблице 7.

Таблица 7 – Поля при успешном запросе

Имя поля	Описание
access_token	Авторизационный токен
token_type	Тип полученного токена (всегда Bearer)
refresh_token	Refresh-токен
expires_in	Срок действия авторизационного токена (в секундах)
scope	Список значений scope. Значения разделяются пробелами. Должен совпадать с запрошенным списком.

Пример:

```
{
"access_token":
"e697d038899ccd0204ec41f215e964d63b0fb423d3ff238a1b7632924976804
d",
"token_type": "Bearer",     "refresh_token":
"a093eab561a7278548c857b5699ff463f3180c506cec1b4791cf84ff1aaed4a
4",
"expires_in": 86399,
"scope": "email openid" }
```


В случае возникновения ошибок используется список, приведенный в таблице 8.

Формат возвращаемых ошибок описан в п. 4.8.

Таблица 8 – Список ошибок

Название ошибки	Условия возникновения
invalid_request	Неверно сформирован запрос
invalid_grant	Переданный refresh-токен не найден (или не принадлежит этому клиенту)
unsupported_grant_type	Передано неподдерживаемое значение grant_type
server_error	Ошибка на стороне сервера

4.6 Список запросов

Запросы, которые можно выполнить с применением авторизационного токена:

UserInfo – запрос возвращает информацию о пользователе в соответствии с выданными правами:

- URL – api/oauth/client/userinfo;
- Метод – GET;
- Тип данных – application/json.

Заголовок запроса описан в таблице 9.

Таблица 9 – Заголовок запроса

Имя	Значение	Описание
Authorization	Bearer {{access_token}}	Авторизация с использованием Bearer-токена

Тело запроса отсутствует.

Пример в формате curl:

```
curl \
--location 'https://id.rosalinux.ru/api/oauth/client/userinfo' \
--header 'Authorization: Bearer e12669547d6ee73e4a7f57c80\
ec97e7dfa1a98b1303e8e9ff4aa8228de9236ea'
```

Ответ возвращается в формате JSON (application/json).

Список полей при успешном запросе приведен в таблице 10. Наличие полей определяется выданными разрешениями.

Таблица 10 – Список полей

Имя поля	Тип	Описание
sub	String (UUID)	Уникальный идентификатор пользователя
name	String	Полное имя
given_name	String	Имя
family_name	String	Фамилия
middle_name	String	Отчество
email	String	Email-адрес
email_verified	Boolean	Email-адрес подтвержден
gender	String	Пол
birthdate	String	Дата рождения
phone_number	String	Номер телефона
phone_number_verified	Boolean	Номер телефона подтвержден
address	String	Адрес
avatar	Map	Аватар

Пример:

```
{
"sub": "ad3ee275-8319-4a13-bcb0-b328e2c56f00",
"email": "user@dummy.mail",
"email_verified": true }
```

4.7 ID-токен

ID-токен представляет собой зашифрованную JWT-строку, подписанную соответствующим ключом. При расшифровке **обязательна проверка на соответствие** подписи и публичного ключа данного сервиса.

Содержание JWT приведено в таблицах 11 и 12.

Таблица 11 – Заголовок (header)

Название	Описание	Возможные значения
typ	Тип (всегда JWT)	JWT
alg	Алгоритм шифрования	RS256

Таблица 12 – Данные (data/payload)

Название	Описание	Возможные значения
iss	Эмитент	rosaid
sub	Тема (уникальный идентификатор пользователя)	--
aud	Получатель (ID клиента, запрашивающего данные)	--
exp	Время истечения (в секундах)	--
iat	Время выдачи (в секундах)	--

4.8 Ошибки

OAuth-ошибки состоят из словаря (ключ-значение), содержащего информацию и подробности конкретной ошибки. Ошибка может возвращаться в формате query-значений либо в виде JSON-объекта. В обоих случаях формат остаётся неизменным.

Ошибка содержит следующие поля, приведенные в таблице 13.

Таблица 13 – Поля ошибки

Имя поля	Возможные значения	Описание	Обязательность
error	invalid_request unauthorized_client access_denied unsupported_response_type invalid_scope server_error temporarily_unavailable invalid_client invalid_grant unsupported_grant_type	Одно из фиксированных значений для определения ошибки	Да
error_description	--	Текстовое описание возникшей ошибки	Нет
error_uri	--	URL-адрес, содержащий описание ошибки	

5 ОПИСАНИЕ СЕРВИСА РОСА ПОЧТА

Приложение "Почта" представляет собой почтовый клиент, предназначенный для работы с электронной почтой на МУ под управлением ОС. Приложение обеспечивает получение, отправку, просмотр и управление электронными сообщениями, а также работу с несколькими почтовыми аккаунтами.

Приложение поддерживает подключение почтовых ящиков различных почтовых сервисов и может использоваться для работы с личной и корпоративной электронной почтой.

При первом запуске приложения отображается экран выбора почтового сервиса для настройки почтового аккаунта.

5.1 Добавление аккаунта преднастроенного почтового сервера

Пользователь может выбрать один из преднастроенных почтовых сервисов либо выполнить настройку почтового аккаунта самостоятельно по пункту "Другое" (рисунок 2).

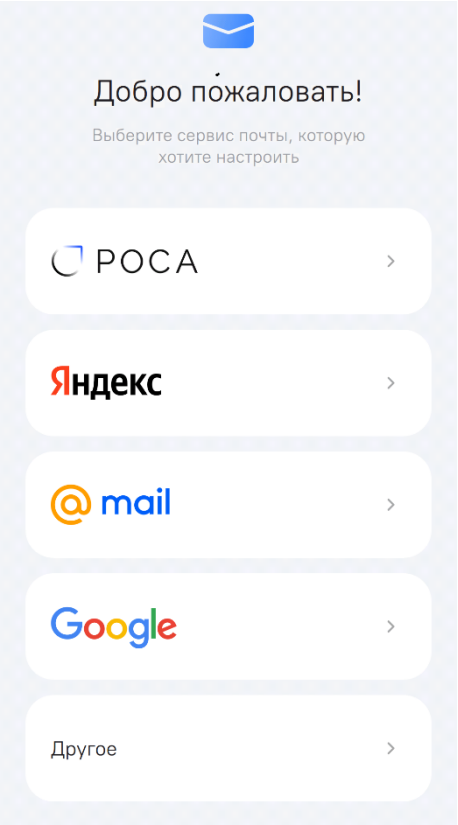

Рисунок 2 – Первый вход в приложение Почта

Если необходимо подключиться к одному из преднастроенных серверов, то для подключения достаточно указать адрес электронной почты и далее пароль от нее (рисунок 3).

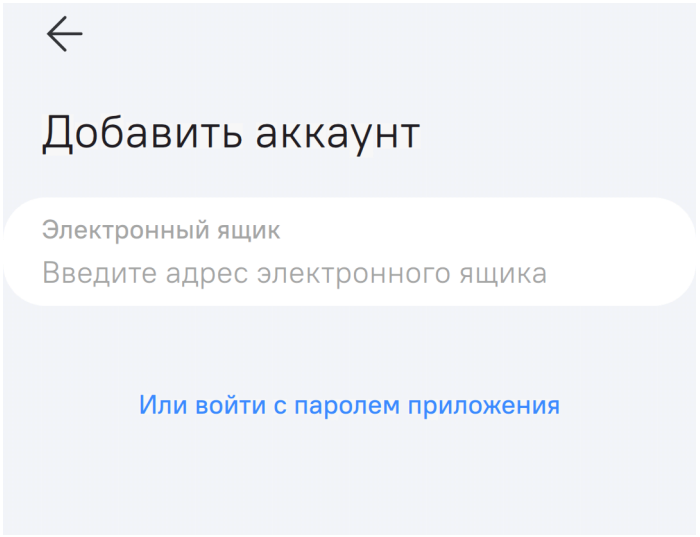

Рисунок 3 – Добавление адреса электронной почты к преднастроенному серверу

5.2 Добавление аккаунта вручную

Ручное добавление почтового аккаунта (пункт "Другое") используется в случае, если требуемый почтовый сервис отсутствует в списке предустановленных провайдеров.

Пользователь может получить данные для ручной настройки следующими способами:

- в личном кабинете почтового сервиса в разделе настроек почты;
- у администратора корпоративной почтовой системы.

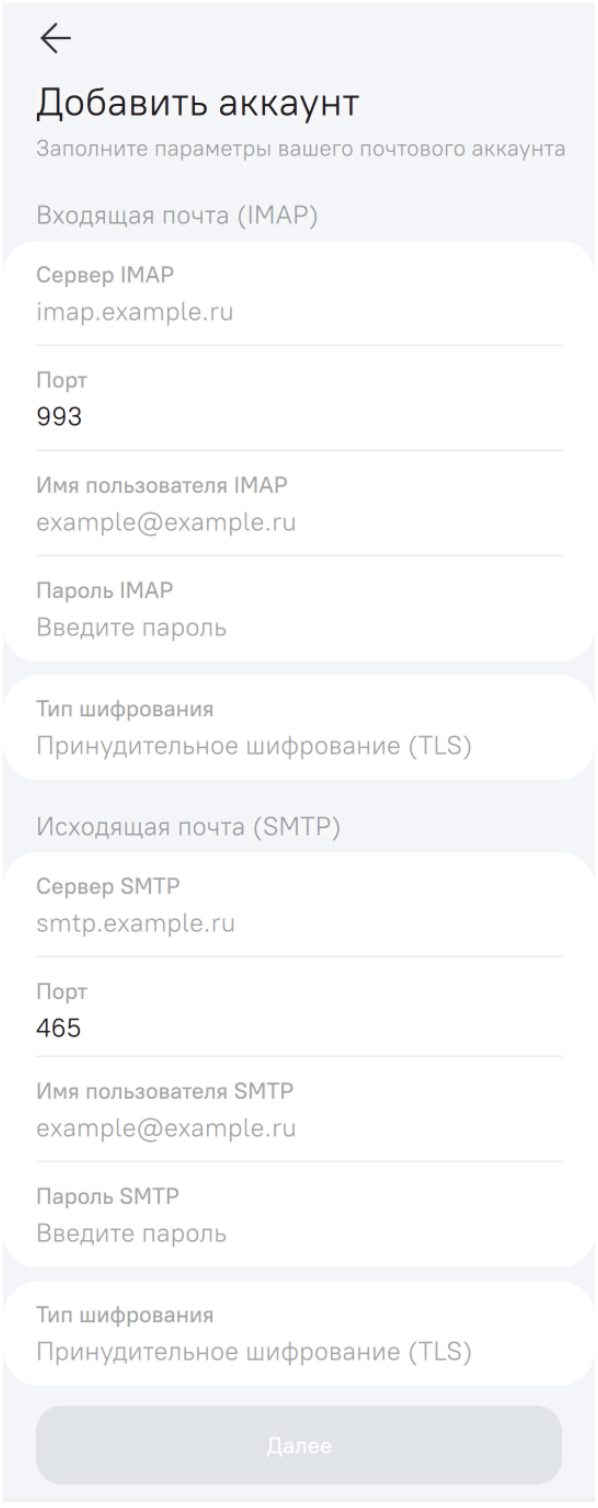

Рисунок 4 – Добавление почтового аккаунта вручную

Для привязки почтового аккаунта требуется вручную указать следующие параметры подключения к серверу (рисунок 4):

- Входящая почта (IMAP):
  - Сервер IMAP – адрес сервера входящей почты;
  - Порт – номер порта IMAP (по умолчанию 993);
  - Имя пользователя IMAP – имя пользователя для подключения, обычно это полный адрес электронной почты;
  - Пароль IMAP – пароль от почтового ящика;
  - Тип шифрования – используется принудительное шифрование TLS.
- Исходящая почта (SMTP):
  - Сервер SMTP – адрес сервера исходящей почты;
  - Порт – номер порта SMTP (по умолчанию 465);
  - Имя пользователя SMTP – имя пользователя для SMTP-аутентификации;
  - Пароль SMTP – пароль от почтового ящика;
  - Тип шифрования – используется принудительное шифрование TLS.

После заполнения всех обязательных полей кнопка `Далее` становится активной. Для завершения настройки аккаунта необходимо нажать кнопку `Далее` и дождаться проверки параметров подключения.

6 ОПИСАНИЕ СЕРВИСА РОСА КАЛЕНДАРЬ

Приложение "Календарь" представляет собой мобильный клиент для просмотра и управления календарными данными пользователей. Приложение поддерживает синхронизацию с внешними календарными сервисами с использованием протокола CalDAV. Поддерживается подключение к календарям следующих сервисов: Google, Apple, Яндекс, Mail.ru и других.

Для синхронизации календаря нужно в приложении добавить учётные записи в зависимости от выбранных сервисов для синхронизации (рисунок 5).

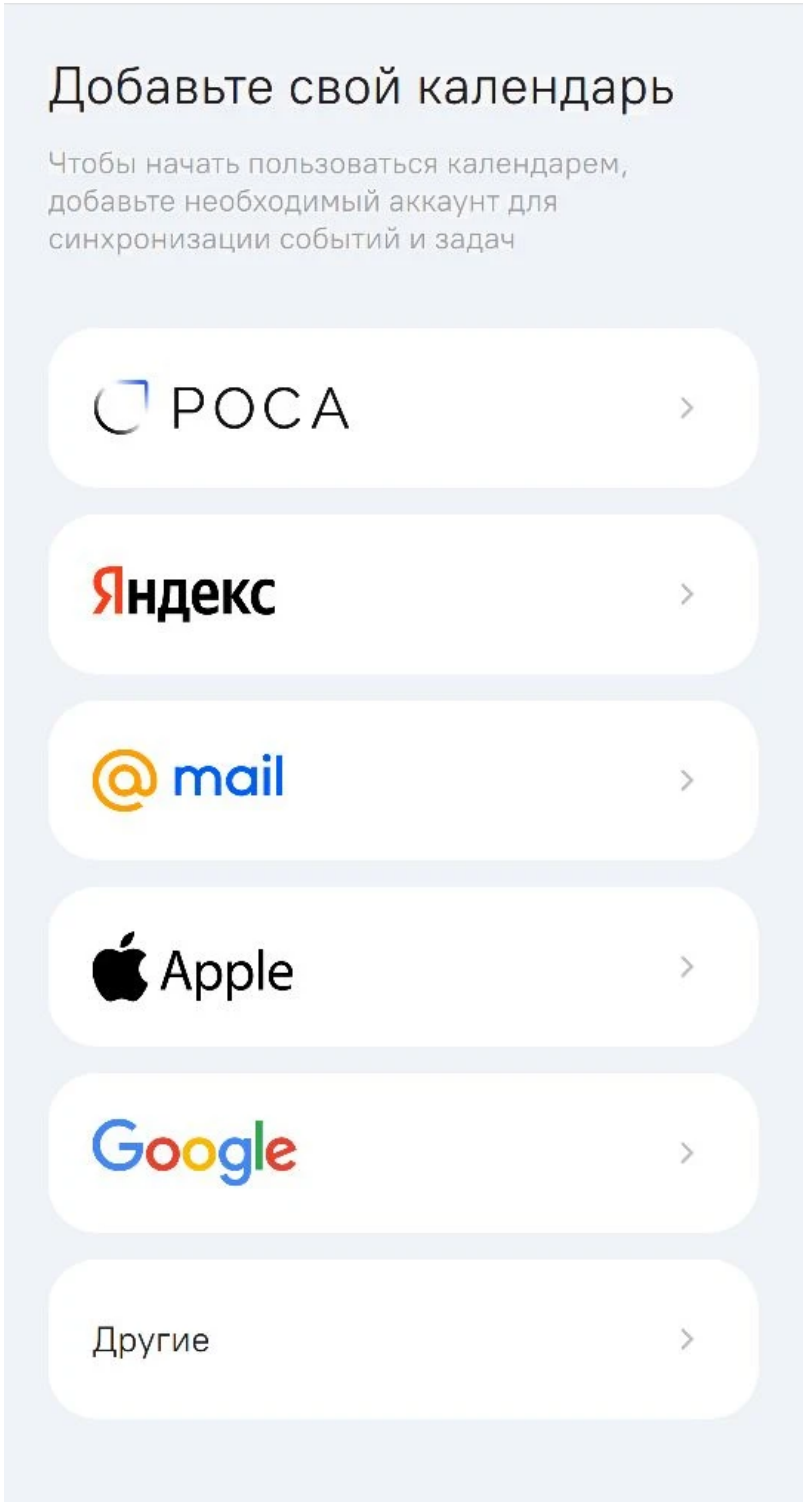

Рисунок 5 – Добавление календаря

После выбора сервиса необходимо пройти авторизацию в нем, для этого необходимо ввести адрес электронной почты, пароль для сервиса календарей и в случае добавления сервисов "Другие" – адрес сервера календаря.

Ручное добавление календаря (пункт "Другие") используется в случае, если требуемый сервис отсутствует в списке предустановленных.

Синхронизация подключённого календаря с сервером выполняется в автоматическом режиме, по заданному периоду времени или вручную с помощью кнопки (Обновить).

7 АДМИНИСТРИРОВАНИЕ СЕРВИСА РОСА МЕССЕНДЖЕР

7.1 Общие сведения о сервисе

Сервис РОСА Мессенджер (далее – Мессенджер) предназначен для обеспечения защищённых и предсказуемых коммуникаций в корпоративной среде. Сервис разворачивается в инфраструктуре организации и управляется её собственными средствами.

Клиент Мессенджера встроен в ОС и работает на российских устройствах, обеспечивая предсказуемость и управляемость.

Развёртывание и администрирование серверной части Мессенджера проводится в инфраструктуре организации с использованием российских технологий, обеспечивая управляемость и безопасность сервиса.

Мессенджер входит в Единую экосистему РОСА и интегрирован с программными продуктами РОСА ID, РОСА Центр управления и Dynamic Directory как часть единой управляемой среды.

Для интеграции в бизнес-процессы и корпоративные приложения доступны API и SDK через Портал разработчика, что позволяет расширять возможности Мессенджера, сохраняя контроль внутри доверенной среды.

Основные функциональные возможности Мессенджера:

- Коммуникации:
  - Чаты для рабочих групп, подразделений и работников;
  - Централизованная доставка push-уведомлений о новых сообщениях при неактивном приложении;
  - Аудиозвонки P2P с высоким качеством звука;
  - Видеозвонки P2P с шифрованием.
- Управление:
  - Авторизация через РОСА ID;
  - Автоматическая загрузка контактов из Dynamic Directory;
  - Централизованные политики и онбординг через РОСА Центр управления.
- Доступ к веб-версии через браузер при сохранении контроля;
- Создание видеоконференций и отправка ссылок на подключение;
- Обеспечение единой коммуникационной платформы через новости организации и рабочие каналы;
- Добавление событий и напоминаний в календарь и заметки.

7.2 Обеспечение безопасности в РОСА Мессенджер

7.2.1 Шифрование данных

В Мессенджере реализовано сквозное шифрование текстовых сообщений в индивидуальных и групповых чатах. Для шифрования сообщений в каждом защищенном чате используется симметричный ключ. Ключу присваивается уникальный идентификатор в рамках соответствующего чата.

Симметричный ключ не передается между участниками в открытом виде. Для каждого участника создается отдельный экземпляр симметричного ключа, зашифрованный с использованием его публичного ключа RSA-4096.

Каждый пользователь имеет асимметричную пару ключей:

– приватный ключ, который хранится локально на МУ и не передается по сети;

– публичный ключ, который используется другими участниками для защищенной передачи симметричного ключа.

7.2.2 Сценарии обмена ключами в индивидуальных и групповых чатах

Порядок создания и распространения симметричных ключей зависит от типа чата.

Для распространения симметричных ключей и запроса отсутствующих ключей используются служебные сообщения Мессенджера:

– DeclareKey – передача симметричного ключа, зашифрованного отдельно для каждого получателя;

– RequestKey – запрос необходимого симметричного ключа участником, у которого данный ключ отсутствует.

7.2.2.1 Индивидуальный чат

При создании индивидуального чата инициатор формирует симметричный ключ с начальным идентификатором keyId = 1. Данный ключ используется для шифрования текстовых сообщений в течение всего времени существования чата.

Для передачи ключа собеседнику инициатор получает публичные ключи всех его устройств и формирует служебное сообщение DeclareKey. В сообщение включаются отдельные экземпляры симметричного ключа, зашифрованные для каждого устройства собеседника. После получения сообщения каждое устройство

расшифровывает предназначенный для него экземпляр ключа с использованием приватного ключа и сохраняет симметричный ключ в локальном хранилище.

При подключении нового устройства сохраненные ранее симметричные ключи чатов на нем отсутствуют. Если новое устройство получает сообщение, зашифрованное неизвестным ключом, оно направляет в чат служебное сообщение RequestKey с идентификатором требуемого ключа.

Любое устройство участника чата, располагающее запрошенным ключом, формирует сообщение DeclareKey, содержащее экземпляр симметричного ключа, зашифрованный публичным ключом нового устройства. После получения и расшифрования ключ сохраняется на новом устройстве, после чего становится доступно расшифрование соответствующих сообщений.

7.2.2.2 Групповой чат

При создании группового чата инициатор формирует симметричный ключ с начальным идентификатором keyId = 1 и получает публичные ключи участников чата. Для каждого участника или его устройства создается отдельный экземпляр симметричного ключа, зашифрованный соответствующим публичным ключом.

Зашифрованные экземпляры ключа объединяются в служебном сообщении DeclareKey и передаются участникам чата. Каждый участник выбирает предназначенный для него экземпляр, расшифровывает его с использованием приватного ключа и сохраняет полученный симметричный ключ в локальном хранилище.

При добавлении нового участника пользователь, выполняющий добавление, должен передать ему действующий симметричный ключ с помощью сообщения DeclareKey. Ключ шифруется публичным ключом нового участника и не передается в открытом виде.

Если у пользователя, добавляющего нового участника, отсутствует действующий симметричный ключ, применяется механизм отложенной передачи. В этом случае сообщение DeclareKey формируется одним из устройств участников чата, на котором соответствующий ключ сохранен.

При подключении нового устройства участника группового чата применяется тот же порядок восстановления ключа, что и для индивидуального чата: устройство направляет сообщение RequestKey, а одно из устройств, располагающих ключом, передает его в сообщении DeclareKey в зашифрованном виде.

7.2.2.3 Административный доступ к зашифрованным сообщениям

Для обеспечения контролируемого доступа уполномоченных лиц к зашифрованным сообщениям предусмотрен механизм административного ключа. Для работы механизма предварительно формируется асимметричная пара ключей RSA-4096:

– приватный административный ключ хранится в защищенной серверной инфраструктуре POCA и не передается клиентским приложениям;

– публичный административный ключ включается в конфигурацию серверной части Мессенджера и используется при формировании ключевых данных чата.

При создании симметричного ключа чата клиент дополнительно формирует его экземпляр, зашифрованный публичным административным ключом. Полученный шифротекст включается в служебное сообщение DeclareKey наряду с экземплярами ключа, предназначенными для участников чата.

При наличии приватного административного ключа серверная часть обеспечивает расшифрование соответствующего симметричного ключа и сообщений, зашифрованных с его использованием.

Клиентские приложения не имеют доступа к приватному административному ключу и не могут расшифровать экземпляр симметричного ключа, предназначенный для административного доступа.

Доступ к приватному административному ключу должен предоставляться только уполномоченным лицам.

7.2.3 Использование протокола HTTPS для обеспечения сетевой безопасности

– Основной обмен данными между клиентом и серверной инфраструктурой Мессенджера выполняется по протоколу HTTPS. Для централизованной доставки push-уведомлений компонент PushService, входящий в состав rosa-im-daemon, устанавливает постоянное соединение с серверным компонентом push-cli с использованием QWebSocket.

– Проверка серверного сертификата, вплоть до зашивания в клиент самоподписанного сертификата SSL, позволяет устранить атаку MITM (Man-In-the-Middle), когда злоумышленник пытается использовать прокси для вскрытия обмена с сервером.

– Возможно использование широчайшего спектра возможностей, программных и аппаратных, для отражения атак типа DDOS на серверы, включая:

- возможности автоматической блокировки IP-адресов, с которых производится атака;
- включение шейперов трафика с ограничением по числу запросов для конкретных пользователей;
- возможности балансировки нагрузки между разными серверами для достижения оптимальной производительности системы.

– Проверка клиентских сертификатов для получения доступа к защищённой инфраструктуре.

– Проксирование запросов HTTPS через специальные прокси, которые бы занимались проверкой валидности пользователей, их возможностью доступа к серверу в определённое время суток, изоляцией доступа к защищённому контуру компании от внешнего мира.

– Использование аппаратных border manager, отличных от используемого по умолчанию nginx, для решения вопросов безопасности при сохранении высокой нагрузочной способности сервера.

7.3 Управление учетной записью

Раздел "Настройки" предназначен для управления параметрами учетной записи пользователя, редактирования профиля и выполнения операций, связанных с управлением аккаунтом в приложении "Мессенджер".

Переход к разделу осуществляется через нижнюю навигационную панель путем выбора вкладки "Настройки" (рисунок 6).

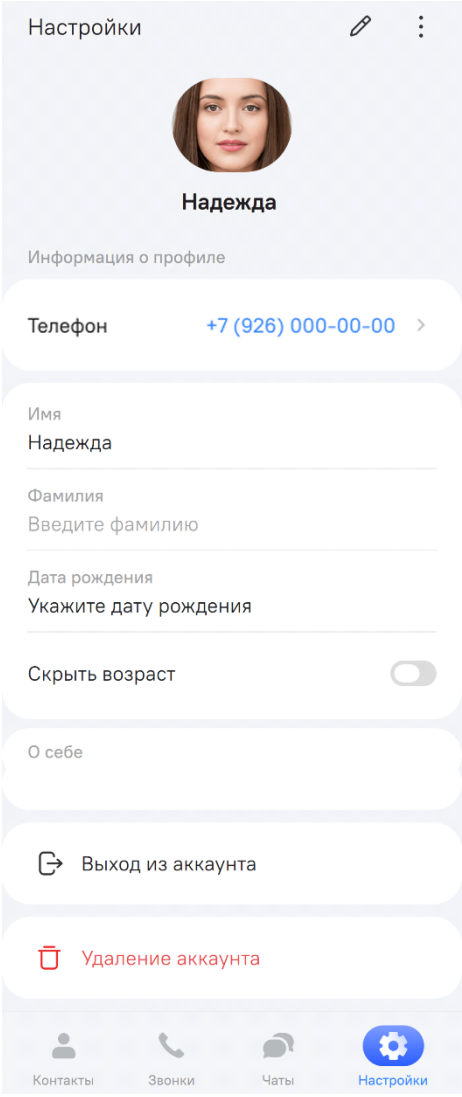

Рисунок 6 – Экран раздела "Настройки"

В разделе "Настройки" отображается номер телефона, привязанный к текущей учетной записи пользователя.

В данном разделе также возможно добавление и редактирование следующей информации о пользователе:

- фотография профиля;
- отображаемое в приложении имя и фамилия;
- дата рождения;
- блок "О себе".

Для изменения данных профиля необходимо нажать на соответствующее поле и ввести или изменить значение. Для изменения фотографии профиля нужно выбрать необходимое изображение в файловом менеджере МУ.

Для выхода из текущей учетной записи необходимо нажать на кнопку `Выход из аккаунта`. После выхода пользователь перенаправляется на экран авторизации.

Для удаления учетной записи необходимо нажать на кнопку `Удаление аккаунта`. Удаление учетной записи приводит к деактивации аккаунта и удалению его данных.

7.4 Подключение к РОСА Мессенджер

При первом запуске приложения (рисунок 7) необходимо пройти процедуру авторизации, после завершения которой пользователь получает доступ к функциональным возможностям приложения.



Рисунок 7 – Первый запуск Мессенджера

Авторизация в "Мессенджере" возможна двумя способами:

а) через РОСА ID – ввод логина и пароля от учетной записи РОСА ID (рисунок 8). Подробнее о РОСА ID см. в п. 4;

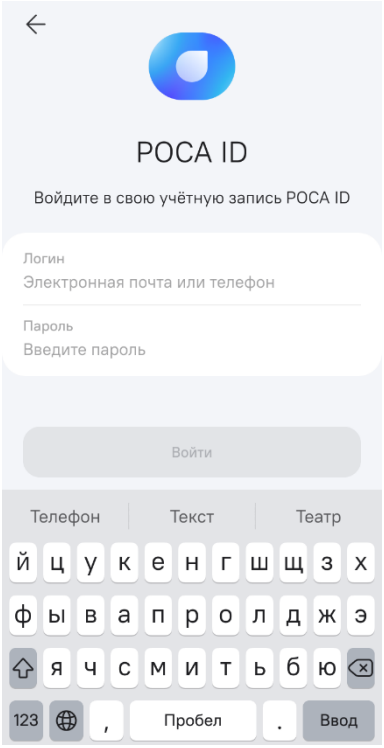

Рисунок 8 – Авторизация через учетную запись РОСА ID

б) подключение к корпоративному серверу – указание адреса корпоративного сервера "РОСА Мессенджер", к которому будет подключаться приложение (рисунок 9).

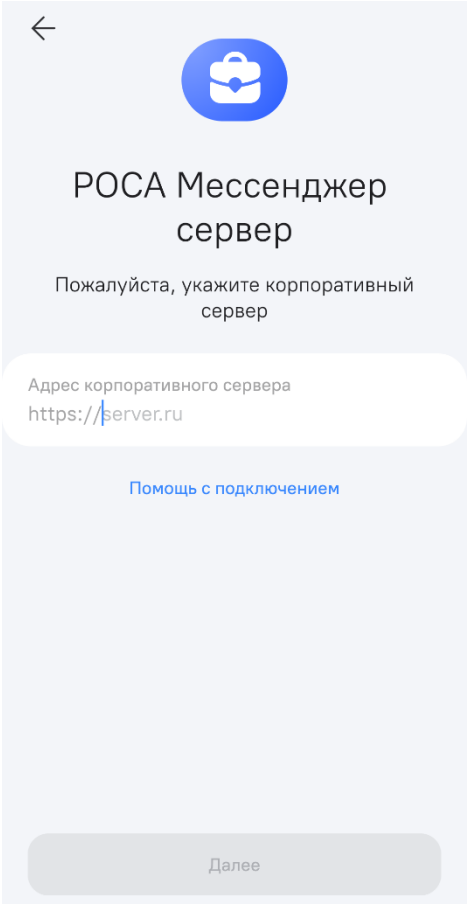

Рисунок 9 – Авторизация через корпоративный сервер

На следующем этапе требуется указать номер мобильного телефона пользователя (рисунок 10). Номер телефона вводится с указанием кода страны и используется для идентификации пользователя в приложении.

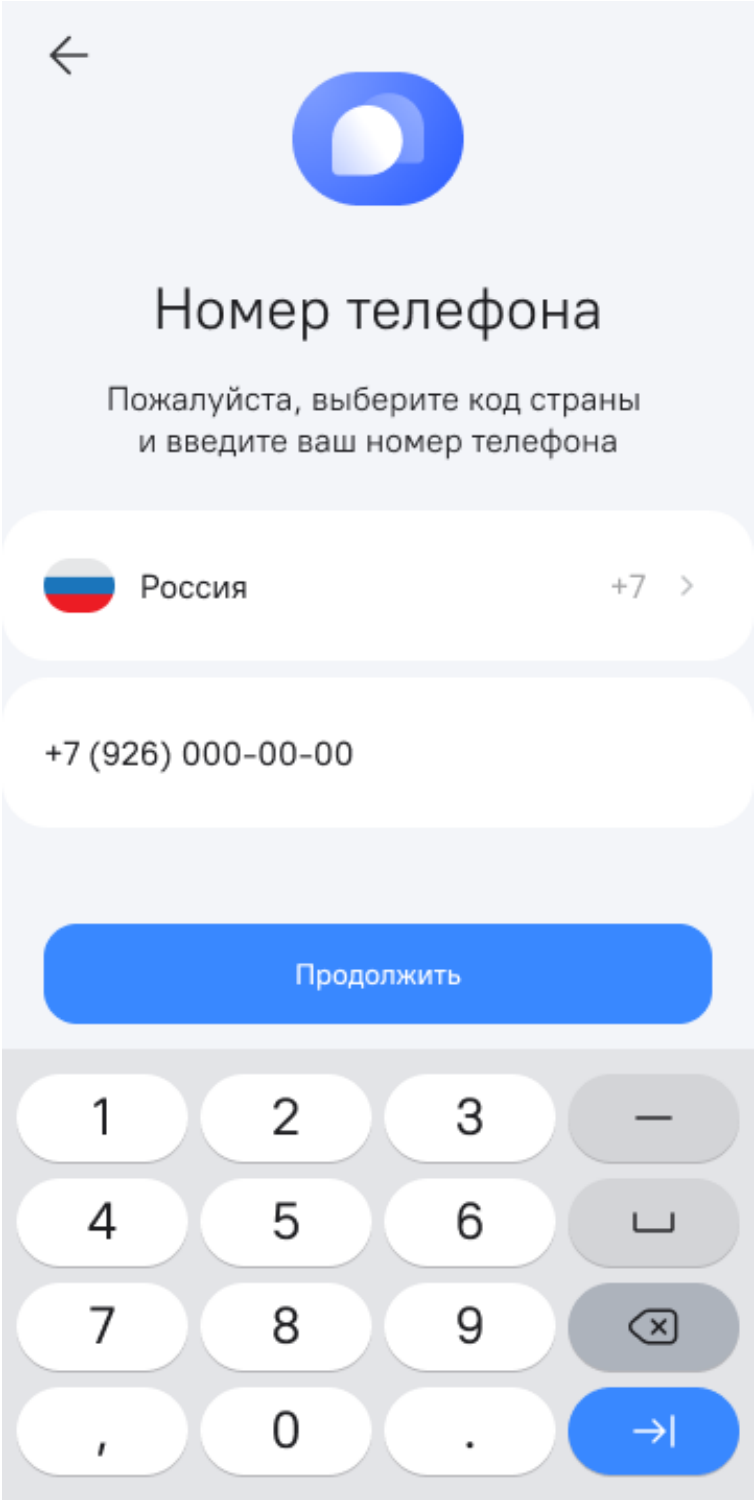

Рисунок 10 – Добавление номера телефона пользователя

После ввода номера телефона на него отправляется сообщение с кодом подтверждения. При авторизации через учетную запись POCA ID код приходит на

подключенную к учетной записи электронную почту. Полученный код необходимо ввести в соответствующее поле на экране приложения (рисунок 11).

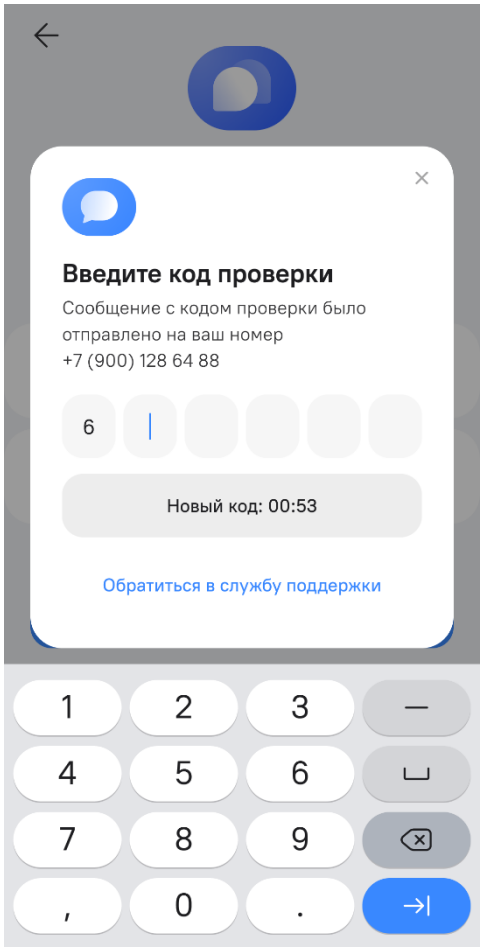

Рисунок 11 – Ввод кода подтверждения

При отсутствии сообщения с кодом подтверждения предусмотрена возможность повторного запроса кода по истечении установленного времени ожидания. В случае возникновения проблем следует обратиться в службу поддержки по нажатию на соответствующую кнопку.

После успешного подтверждения номера телефона отображается экран ввода информации о пользователе, включая добавление или изменение фотографии профиля при необходимости (рисунок 12).

Указанные данные используются для отображения информации о пользователе в приложении "Мессенджер".

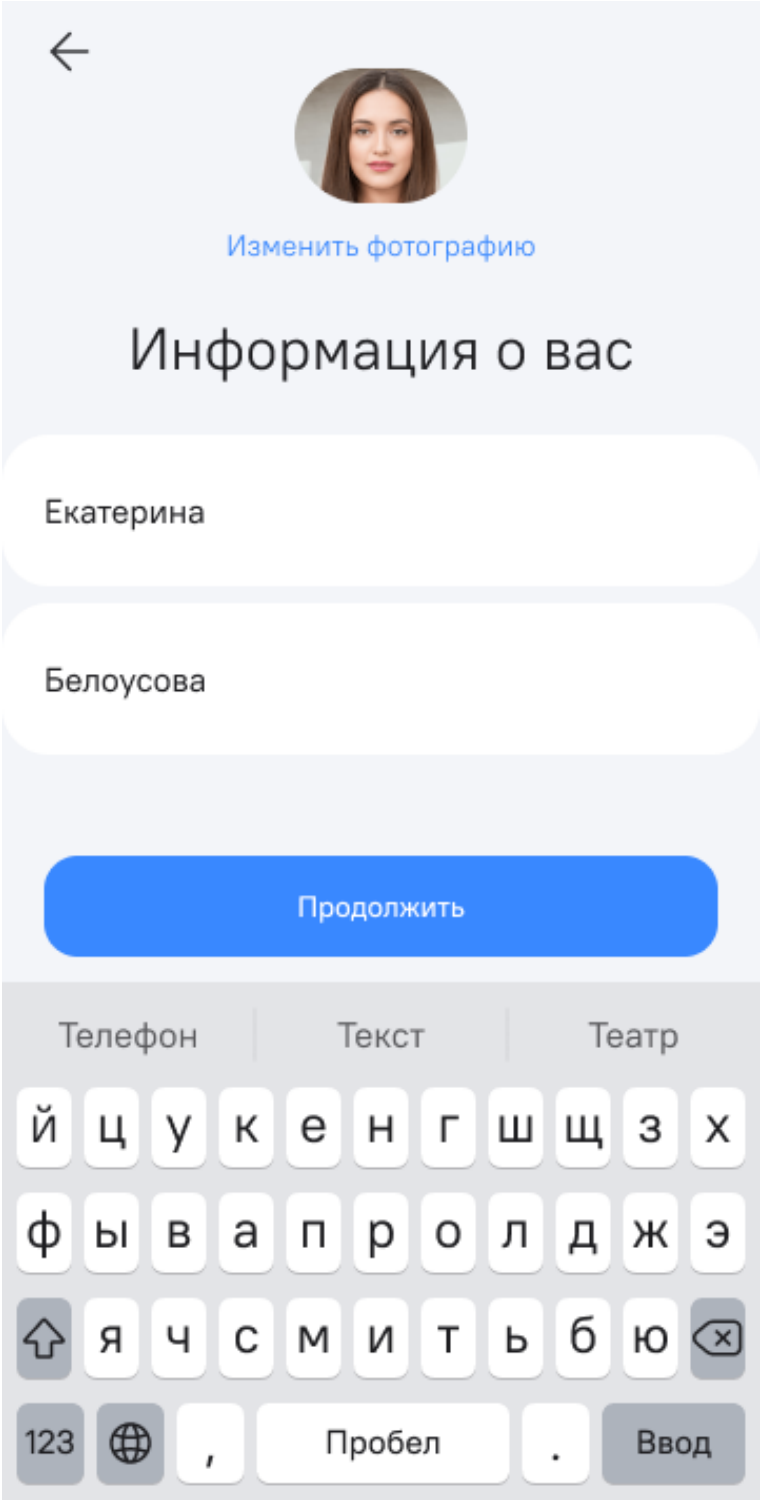

Рисунок 12 – Заполнение данных профиля

После успешного завершения регистрации и заполнения данных профиля открывается основной интерфейс приложения.

При отсутствии активных диалогов отображается экран со списком чатов (рисунок 13), готовый к созданию нового сообщения или началу взаимодействия с контактами.

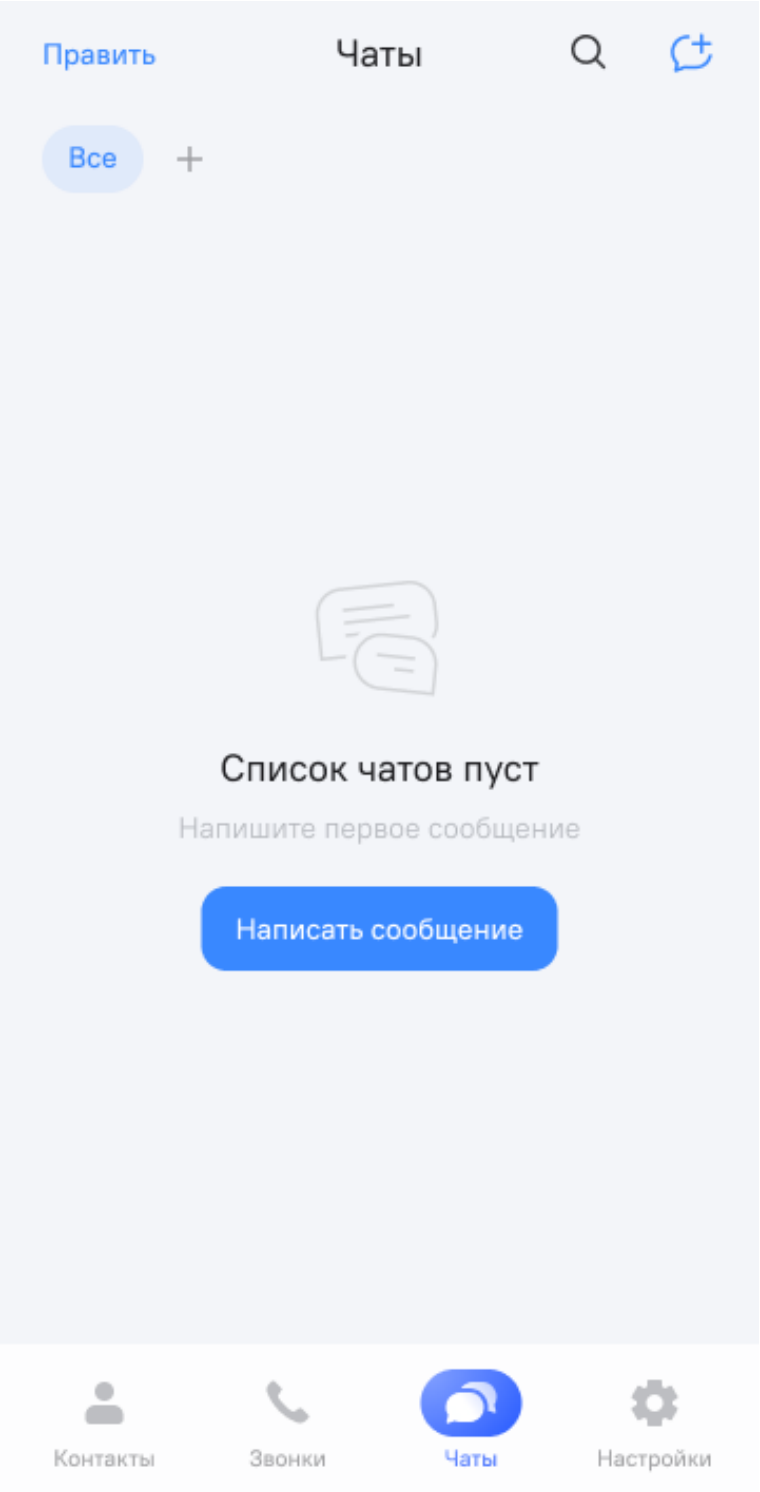

Рисунок 13 – Интерфейс приложения при первом входе

В нижней части экрана располагается панель навигации приложения, обеспечивающая переход между основными разделами:

- Контакты – переход к списку контактов пользователя;
- Звонки – доступ к истории вызовов и функциям аудио- и видеозвонков;
- Чаты – переход к списку чатов (текущий активный раздел);
- Настройки – переход к параметрам приложения и профиля пользователя.

Текущий активный раздел визуально выделяется на панели навигации.

7.5 Доставка push-уведомлений

Для доставки уведомлений о новых сообщениях в ОС используется централизованный механизм push-уведомлений. Использование единого механизма позволяет приложениям получать уведомления без запуска отдельных постоянно работающих процессов для каждого приложения. В версии 3.1 механизм push-уведомлений интегрирован с клиентом РОСА Мессенджер.

В состав серверной части механизма входят следующие компоненты:

- OMMG – шлюзовой компонент, выполняющий первичную фильтрацию входящих сообщений, приведение сообщений к единому внутреннему формату и ведение реестра активных клиентских устройств;
- push-srv – центральный компонент обработки push-сообщений, управляющий сведениями о приложениях, их атрибутах и категориях уведомлений, а также сохраняющий подготовленные данные в хранилище Redis;
- push-cli – серверный компонент, обслуживающий постоянные соединения с клиентскими устройствами, получающий данные из Redis и доставляющий их подключенным клиентам.

На стороне МУ взаимодействие с сервером push-cli выполняет компонент PushService, входящий в состав rosa-im-daemon. Компонент выполняет авторизацию, устанавливает соединение с сервером с использованием QWebSocket и получает предназначенные для устройства уведомления.

Если новое сообщение поступает при закрытом или неактивном приложении Мессенджера, ОС отображает соответствующее уведомление. При открытом активном чате дополнительное push-уведомление о сообщении в данном чате не отображается.

ТЕРМИНЫ И СОКРАЩЕНИЯ

Термин/Сокращение	Определение
ВМ	Виртуальная машина
ИТ	Информационные технологии
МУ	Мобильное устройство
МУД	Мандатное управление доступом
ОС	Операционная система
ПО	Программное обеспечение
СКЗИ	Средство криптографической защиты информации
СПО	Свободное программное обеспечение
ТК	Телефонная книга
ЭП	Электронная подпись
AOSP	Android Open Source Project – открытая версия операционной системы Android без фирменных сервисов Google
API	Application programming interface – программный интерфейс приложения
CLI	Command-line interface – интерфейс командной строки
MAC	Media access control – уникальный идентификатор сетевого оборудования
URL	Uniform resource locator – сетевой адрес ресурса