

АНО «ГОРЬКИЙ ТЕХ»

**ОПИСАНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ
ХРАНЕНИЯ И КОМПИЛЯЦИИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

«Корпоративный электронный помощник (КЭП)»

Версия 1.0

ГОСТ Р ИСО/МЭК 12207-2010

Нижний Новгород

2025

Оглавление

1. ОБЩИЕ СВЕДЕНИЯ	3
1.1. Назначение документа.....	3
1.2. Область применения	3
1.3. Требования к территориальному размещению	3
2. ТЕХНИЧЕСКИЕ СРЕДСТВА ХРАНЕНИЯ ИСХОДНОГО КОДА	4
2.1. Основное хранилище	4
2.2. Система контроля версий Git	4
2.3. Платформа GitLab	5
2.4. Структура репозитория	5
3. ХРАНЕНИЕ ОБЪЕКТНОГО КОДА	7
3.1. Docker Registry	7
3.2. Структура образов контейнеров	7
3.3. Версионирование образов	8
4. ТЕХНИЧЕСКИЕ СРЕДСТВА КОМПИЛЯЦИИ	9
4.1. Серверы компиляции	9
4.2. Компиляторы и интерпретаторы	9
4.3. Инструменты сборки.....	10
5. СИСТЕМА CI/CD	11
5.1. GitLab CI/CD	11
5.2. Конвейер сборки	11
5.3. Автоматическое тестирование	11
6. РЕЗЕРВНОЕ КОПИРОВАНИЕ И АРХИВИРОВАНИЕ	12
6.1. Стратегия резервного копирования	12
6.2. Хранение резервных копий	12
7. ТРЕБОВАНИЯ К ИНФРАСТРУКТУРЕ	13
7.1. Аппаратные требования	13
7.2. Программные требования.....	13

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Назначение документа

Настоящий документ содержит описание технических средств, используемых для хранения исходного текста и объектного кода программного обеспечения «Корпоративный электронный помощник (КЭП)», а также технических средств компиляции исходного текста в объектный код программного обеспечения.

1.2. Область применения

Документ применяется:

- при организации хранения исходного кода программного обеспечения;
- при настройке систем контроля версий и управления конфигурацией;
- при развертывании инфраструктуры компиляции и сборки;
- при обеспечении территориальной принадлежности исходного кода РФ;
- при организации резервного копирования и архивирования.

1.3. Требования к территориальному размещению

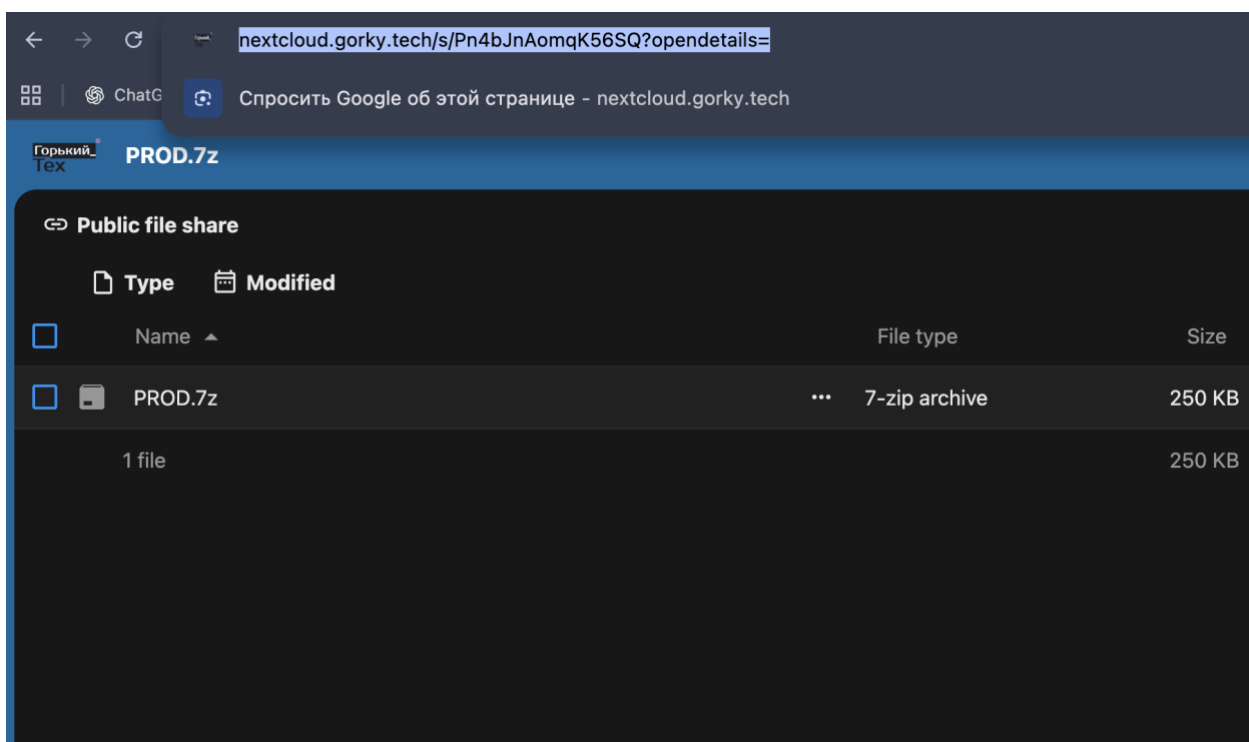
В соответствии с требованиями к включению в реестр российского ПО, все технические средства хранения и компиляции размещаются на собственном сервере на территории Российской Федерации. При обращении к серверам трансграничной передачи данных не осуществляется.

2. ТЕХНИЧЕСКИЕ СРЕДСТВА ХРАНЕНИЯ ИСХОДНОГО КОДА

2.1. Основное хранилище

Исходные коды программного обеспечения хранятся во внутреннем защищенном хранилище Nextcloud по защищенной ссылке:

<https://nextcloud.gorky.tech/s/Pn4bJnAomqK56SQ>



Nextcloud развернут на собственном сервере, который находится по юридическому адресу: Нижний Новгород, Алексеевская, 6/16, 2-й этаж, офис 214 (21).

2.2. Система контроля версий Git

Git — распределенная система контроля версий, используемая для отслеживания изменений в исходном коде. Установлена версия Git 2.40 или

выше. Все операции с репозиториями выполняются через защищенное соединение SSH с использованием ключей RSA-4096.

Применяется стратегия ветвления Git Flow с основными ветками: main (стабильная продуктивная версия), develop (основная ветка разработки), feature/* (ветки новых функций), bugfix/* (исправление ошибок), release/* (подготовка релизов).

2.3. Платформа GitLab

В качестве технических средств хранения кода используются следующие программные продукты: — GitLab, система управления репозиториями кода для Git с собственной вики, системой отслеживания ошибок, CI/CD пайплайном и другими функциями.

Параметр	Значение
Платформа	GitLab Community Edition 16.x
Размещение	Нижний Новгород, Алексеевская, 6/16, 2-й этаж, офис 214 (21)
Операционная система	Ubuntu Server 22.04 LTS
Процессор	8 ядер Intel Xeon (или аналог)
Оперативная память	16 ГБ RAM
Хранилище	SSD NVMe 2 ТБ с RAID 1
Резервное копирование	Ежедневно, хранение 30 дней

2.4. Структура репозитория

Исходный код организован в следующих репозиториях:

1. ker-backend — серверная часть и API;
2. ker-telegram-frontend — Telegram-интерфейс;
3. ker-max-frontend — интеграция с МАХ;

4. `ker-web-frontend` — веб-интерфейс;
5. `ker-vectordb` — компонент векторной базы данных;
6. `ker-docs` — документация проекта.

Доступ к репозиториям осуществляется по протоколу SSH. Настроена двухфакторная аутентификация для всех участников разработки. Применяется ролевая модель доступа с разделением прав на чтение, запись и администрирование.

3. ХРАНЕНИЕ ОБЪЕКТНОГО КОДА

3.1. Docker Registry

Объектный код программного обеспечения хранится на собственном сервере на территории Российской Федерации. В качестве технических средств хранения используется Docker Registry — реестр для хранения Docker-образов.

3.2. Структура образов контейнеров

Программное обеспечение распространяется в виде следующих Docker-образов:

- `ker/backend` — серверное API и бизнес-логика;
- `ker/telegram-frontend` — Telegram-бот;
- `ker/max-frontend` — интеграция с МАХ;
- `ker/web-frontend` — веб-интерфейс;
- `ker/vectordb` — векторная база данных.

Каждый образ содержит все необходимые зависимости и конфигурационные файлы для автономной работы. Образы строятся с использованием многоступенчатой сборки (multi-stage build) для минимизации размера.

3.3. Версионирование образов

Каждый образ имеет уникальный тег версии в формате semantic versioning (MAJOR.MINOR.PATCH), например 1.0.0. Дополнительно используются теги: latest (последняя стабильная версия), develop (текущая разработка), git-COMMIT_SHA (привязка к конкретному коммиту для отслеживаемости).

4. ТЕХНИЧЕСКИЕ СРЕДСТВА КОМПИЛЯЦИИ

4.1. Серверы компиляции

Компиляция исходного кода производится собственном сервере на территории Российской Федерации. В качестве технических средств компиляции кода используются: Python 3.12 — интерпретатор для Python-кода; Docker — система контейнеризации для сборки образов; GitLab CI/CD — система непрерывной интеграции.

Для компиляции исходного кода используются выделенные серверы-раннеры GitLab CI/CD, размещенные в том же дата-центре. Серверы работают под управлением Ubuntu Server 22.04 LTS и оснащены Docker Engine для сборки контейнеров.

Характеристики сервера компиляции:

Параметр	Значение
Процессор	16 ядер Intel Xeon (или аналог)
Оперативная память	32 ГБ RAM
Хранилище	SSD NVMe 1 ТБ
Операционная система	Ubuntu Server 22.04 LTS

4.2. Компиляторы и интерпретаторы

На серверах компиляции установлены следующие средства разработки:

- Python 3.12 — для серверной части и обработки данных;
- Node.js 20 LTS — для сборки фронтенд-приложений;
- npm/yarn — менеджеры пакетов JavaScript;

- `pip` — менеджер пакетов Python;
- Docker 24.x — для создания контейнеров;
- Docker Compose — для оркестрации контейнеров.

4.3. Инструменты сборки

Процесс компиляции включает следующие этапы: установка зависимостей (`pip install` для Python, `npm install` для JavaScript), проверка качества кода (`pylint` для Python, `eslint` для JavaScript), запуск автоматических тестов (`pytest`, `jest`), сборка Docker-образов с использованием многоступенчатой сборки, публикация образов в Docker Registry с соответствующими тегами.

5. СИСТЕМА CI/CD

5.1. GitLab CI/CD

Для автоматизации процессов сборки, тестирования и развертывания используется встроенная система GitLab CI/CD. Конфигурация конвейеров описана в файлах `.gitlab-ci.yml` в корне каждого репозитория. Конвейер запускается автоматически при каждом коммите в репозиторий.

5.2. Конвейер сборки

Типовой конвейер включает следующие стадии:

- `build` — установка зависимостей и компиляция кода;
- `test` — запуск модульных и интеграционных тестов;
- `lint` — проверка качества и стиля кода;
- `docker` — сборка Docker-образов;
- `publish` — публикация образов в Registry;
- `deploy` — развертывание на тестовую или продуктивную среду.

5.3. Автоматическое тестирование

На каждом этапе сборки выполняются автоматические тесты: модульные тесты (`pytest` для Python, `jest` для JavaScript) с покрытием кода не менее 70 процентов, интеграционные тесты компонентов, проверка безопасности кода (SAST), проверка зависимостей на известные уязвимости.

6. РЕЗЕРВНОЕ КОПИРОВАНИЕ И АРХИВИРОВАНИЕ

6.1. Стратегия резервного копирования

Резервное копирование репозитория Git выполняется ежедневно в автоматическом режиме. Копируются все репозитории вместе с историей изменений, метаданными и настройками доступа. Используется инкрементальное копирование для оптимизации использования дискового пространства.

6.2. Хранение резервных копий

Резервные копии хранятся на отдельном сервере в том же дата-центре. Срок хранения составляет 30 дней для ежедневных копий. Ежемесячные архивные копии хранятся в течение 1 года. Все резервные копии шифруются с использованием AES-256.

Периодически выполняется проверка возможности восстановления данных из резервных копий. Процедура восстановления документирована и регулярно тестируется.

7. ТРЕБОВАНИЯ К ИНФРАСТРУКТУРЕ

7.1. Аппаратные требования

Минимальные требования к серверному оборудованию:

- Сервер GitLab: 8 ядер CPU, 16 ГБ RAM, 2 ТБ SSD;
- Сервер CI/CD: 16 ядер CPU, 32 ГБ RAM, 1 ТБ SSD;
- Резервное копирование: 5 ТБ сетевого хранилища;
- Сетевое подключение: 1 Гбит/с и выше;
- Источник бесперебойного питания (ИБП).

7.2. Программные требования

Необходимое программное обеспечение:

- Ubuntu Server 22.04 LTS или совместимая ОС;
- GitLab Community Edition 16.x или выше;
- Docker Engine 24.x или выше;
- Git 2.40 или выше;
- Python 3.12;
- Node.js 20 LTS;
- PostgreSQL 14 или выше (для GitLab).

Все серверы должны быть размещены в защищенной сети с межсетевым экраном. Необходимо обеспечить доступ по SSH (порт 22) и HTTPS (порт 443)

с контролируемых IP-адресов. Требуется настройка VPN для удаленного доступа разработчиков.

Все технические средства хранения и компиляции соответствуют требованиям информационной безопасности и размещаются на территории Российской Федерации в соответствии с законодательством о персональных данных.