

**ООО «МИ-МИ»**

Правообладатель программного обеспечения «FinMe»

**Описание технических средств  
хранения и компиляции программного обеспечения  
«FinMe»**

Средства хранения исходного кода, сборки, хранения артефактов, среда разработки и резервное копирование

Москва, 2026

## 1. Общие сведения

Настоящий документ содержит описание технических средств хранения исходного кода (текста программы, из которого формируется работающее программное обеспечение) и средств компиляции (преобразования исходного кода в готовые к исполнению файлы) программного обеспечения «FinMe», правообладателем которого является ООО «МИ-МИ».

Программное обеспечение «FinMe» представляет собой универсальную систему управления партнерскими сетями, работающими по модели оплаты за целевое действие (CPA — Cost Per Action, модель расчетов, при которой оплата производится за совершенное пользователем целевое действие). Система обеспечивает учет переходов пользователей, обработку постбэков (postback — служебное уведомление о совершенном целевом действии, передаваемое от партнерской платформы), формирование отчетности и управление маршрутизацией трафика (потока пользовательских переходов).

Программное обеспечение построено по клиент-серверной архитектуре (модель взаимодействия, при которой серверная часть выполняет обработку данных, а клиентская часть отвечает за отображение интерфейса пользователя). Ниже приведены основные применяемые технологии.

| Компонент                 | Технология                 | Назначение                                                                                    |
|---------------------------|----------------------------|-----------------------------------------------------------------------------------------------|
| Серверная часть           | Java 21, Spring Boot 3.3.4 | Обработка бизнес-логики (правил работы системы), предоставление программных интерфейсов (API) |
| Клиентская часть          | JavaScript, Vue.js         | Пользовательский интерфейс, отображение данных в браузере                                     |
| Реляционная база данных   | PostgreSQL 15              | Основное хранилище связанных данных системы                                                   |
| Аналитическая база данных | ClickHouse                 | Хранилище данных для отчетности и статистических расчетов                                     |
| Объектное хранилище       | S3                         | Хранение файлов и вложений (объектов) системы                                                 |
| Управление задачами       | Яндекс Трекер              | Планирование и учет задач разработки                                                          |

Для хранения исходного кода, его сборки и хранения полученных артефактов (готовых результатов сборки) применяется набор технических средств, размещенных на инфраструктуре российской облачной платформы Yandex Cloud. Состав и порядок применения этих средств подробно раскрыты в последующих разделах документа.

## 2. Средства хранения исходного кода

Хранение исходного кода программного обеспечения «FinMe» организовано с использованием системы контроля версий Git (система учета изменений исходного кода, обеспечивающая хранение полной истории правок и одновременную работу нескольких разработчиков). Git относится к распределенным системам контроля версий: каждая рабочая копия репозитория (хранилища исходного кода) содержит полную историю изменений.

Исходный код размещается в частных (закрытых для внешнего доступа) репозиториях. В организации используется самостоятельно управляемый сервер GitLab (программная платформа для размещения Git-репозитория и управления совместной разработкой), развернутый на инфраструктуре облачной платформы Yandex Cloud, расположенной на территории Российской Федерации. Такой подход обеспечивает контроль организации над хранением исходного кода и его размещение в российском правовом поле.

## 2.1. Структура веток и обозначение версий

Работа с исходным кодом ведется в рамках модели ветвления (схемы организации параллельных линий разработки). Применяется следующая структура веток (branch — обособленная линия разработки внутри репозитория).

- **Основная ветка** — содержит стабильное (проверенное и готовое к использованию) состояние исходного кода.
- **Отдельные ветки для доработок** — создаются под конкретные изменения и новую функциональность; после проверки их содержимое переносится в основную ветку.
- **Обозначение версий тегами** — выпускаемые версии программного обеспечения помечаются тегами (именованными метками фиксированного состояния кода) вида vX.Y.Z, где X, Y и Z — числовые обозначения версии.

Такая организация позволяет вести доработки изолированно, не затрагивая стабильное состояние продукта, и однозначно идентифицировать каждую выпущенную версию по тегу.

## 2.2. Разграничение доступа

Доступ к репозиториям предоставляется по индивидуальным учетным записям с разграничением прав доступа (назначением каждому пользователю только тех полномочий, которые необходимы для выполнения его обязанностей). Это обеспечивает контроль над тем, кто и какие операции может выполнять с исходным кодом, а также сохранность и конфиденциальность исходного кода.

## 3. Средства сборки и компиляции серверной части

Для сборки серверной части программного обеспечения, реализованной на языке Java 21 с применением платформы Spring Boot 3.3.4, используется система сборки Apache Maven (инструмент автоматизации сборки проектов на языке Java, управляющий зависимостями и порядком компиляции).

Для запуска сборки применяется обертка Maven Wrapper (сценарий ./mvnw, автоматически использующий требуемую версию Maven и не требующий его отдельной установки на машине разработчика). Основная команда сборки серверной части приведена ниже.

```
./mvnw clean package
```

Команда выполняет очистку предыдущих результатов сборки (clean) и полную сборку с упаковкой в исполняемый файл (package). Для корректной сборки серверной части необходим

установленный комплект разработчика Java (JDK — Java Development Kit, набор средств для компиляции и выполнения программ на языке Java) версии 21.

| Параметр           | Значение                                                                                     |
|--------------------|----------------------------------------------------------------------------------------------|
| Система сборки     | Apache Maven (через обертку Maven Wrapper — <code>./mvnw</code> )                            |
| Команда сборки     | <code>./mvnw clean package</code>                                                            |
| Требования к среде | Комплект разработчика Java (JDK) версии 21                                                   |
| Формат результата  | Исполняемый файл в формате JAR (Java Archive — архив с исполняемой программой на языке Java) |

Результатом сборки является исполняемый файл в формате JAR — единый архив, содержащий скомпилированный серверный код и необходимые для его работы компоненты, готовый к запуску.

## 4. Средства сборки клиентской части

Клиентская часть программного обеспечения реализована на языке JavaScript с применением каркаса Vue.js (программного каркаса для построения пользовательских интерфейсов в браузере). Для сборки клиентской части используется набор инструментов, описанный ниже.

- **Менеджер пакетов Bun** (инструмент установки и управления зависимостями клиентской части) — установка зависимостей выполняется командой `bun install`. Допускается также применение менеджера пакетов `npm` (стандартного менеджера пакетов среды JavaScript).
- **Сборщик Vite** (инструмент сборки клиентской части, выполняющий преобразование исходного кода в готовый набор файлов).
- **Команда сборки** — выполняет проверку типов (контроль соответствия типов данных) и последующую сборку проекта.

Установка зависимостей и сборка клиентской части выполняются следующими командами.

```
bun install
bun run build
```

Результатом сборки клиентской части является набор статических файлов (готовых к размещению файлов интерфейса), помещаемых в каталог `dist`. Эти файлы содержат собранный клиентский код и предназначены для последующей публикации и отдачи пользователям.

## 5. Хранение собранных артефактов

Собранные серверная и клиентская части упаковываются в образы контейнеров Docker (стандартизированные пакеты, содержащие программу и необходимое для ее работы окружение). Образы контейнеров обеспечивают единообразное развертывание программного обеспечения независимо от особенностей конкретного сервера.

Готовые образы публикуются в реестре контейнеров Yandex Container Registry (хранилище образов контейнеров на российской облачной платформе Yandex Cloud, доступное по адресу `cr.yandex`). Версия публикуемого образа фиксируется в файле сборки `Makefile` (файле,

описывающем правила и параметры сборки), что обеспечивает однозначное соответствие между версией продукта и опубликованным образом.

### 5.1. Локальное хранение результатов сборки

Дополнительно результаты сборки сохраняются локально в каталогах сборки проекта на машине разработчика.

| Артефакт                          | Местоположение                        | Способ хранения                                           |
|-----------------------------------|---------------------------------------|-----------------------------------------------------------|
| Образ контейнера серверной части  | Yandex Container Registry (cr.yandex) | Публикация образа Docker, версия зафиксирована в Makefile |
| Образ контейнера клиентской части | Yandex Container Registry (cr.yandex) | Публикация образа Docker, версия зафиксирована в Makefile |
| Результат сборки серверной части  | Каталог target                        | Локальное хранение файла в формате JAR                    |
| Результат сборки клиентской части | Каталог dist                          | Локальное хранение набора статических файлов              |

Таким образом, собранные артефакты хранятся как в централизованном реестре контейнеров на российской облачной платформе, так и локально в каталогах сборки проекта (target — для серверной части, dist — для клиентской части).

## 6. Среда разработки

Разработка программного обеспечения «FinMe» ведется с применением интегрированных сред разработки (IDE — Integrated Development Environment, программных комплексов, объединяющих средства написания, проверки и сборки кода). Состав применяемых сред приведен в таблице.

| Компонент разработки | Интегрированная среда разработки |
|----------------------|----------------------------------|
| Серверная часть      | IntelliJ IDEA                    |
| Клиентская часть     | WebStorm, Visual Studio Code     |

Разработка ведется на рабочих машинах под управлением различных операционных систем: macOS, Windows и Linux. Для корректной сборки серверной части на рабочей машине разработчика применяется комплект разработчика Java (JDK) версии 21.

- **IntelliJ IDEA** — среда разработки серверной части (Java, Spring Boot).
- **WebStorm** и **Visual Studio Code** — среды разработки клиентской части (JavaScript, Vue.js).
- **Операционные системы** — macOS, Windows, Linux.
- **Комплект разработчика Java (JDK) версии 21** — обязательное требование для сборки серверной части.

## 7. Резервное копирование исходного кода

Резервное копирование исходного кода (создание и хранение дополнительных копий на случай утраты основной) обеспечивается несколькими взаимодополняющими способами.

Основным способом является хранение исходного кода в удаленном (централизованном) репозитории GitLab. Инфраструктура, на которой развернут сервер GitLab, обеспечивает резервное копирование средствами российской облачной платформы Yandex Cloud, что защищает исходный код от утраты при возможных сбоях оборудования.

Дополнительным способом резервирования служит распределенный характер системы контроля версий Git. Каждая рабочая копия исходного кода на машине разработчика содержит полную историю изменений репозитория. Это означает, что при наличии нескольких рабочих копий сохраняется несколько независимых полных копий исходного кода.

- **Удаленный репозиторий GitLab** — основное централизованное хранилище исходного кода.
- **Резервирование средствами облачной платформы** — инфраструктура GitLab обеспечивает резервное копирование средствами Yandex Cloud.
- **Распределенные рабочие копии** — рабочие копии исходного кода на машинах разработчиков содержат полную историю изменений (свойство распределенной системы Git).

Совокупность перечисленных мер обеспечивает надежное хранение и сохранность исходного кода программного обеспечения «FinMe».