

Описание функциональных характеристик программного обеспечения «Orchestra»

Приложение к заявлению о включении сведений о программном обеспечении в единый реестр российских программ для электронных вычислительных машин и баз данных

Составлено во исполнение подпункта «д» пункта 11 Правил формирования и ведения единого реестра российских программ для электронных вычислительных машин и баз данных, утверждённых постановлением Правительства Российской Федерации от 16.11.2015 № 1236.

Поле	Значение
Наименование программного обеспечения	Orchestra
Правообладатель	Общество с ограниченной ответственностью «СИДОН»
ИНН / КПП / ОГРН	2463138169 / 246301001 / 1262400004481
Адрес правообладателя	660062, Красноярский край, г. Красноярск, ул. Лесопарковая, д. 33, кв. 324
Язык реализации	Python (реализация CPython 3.12)
Условия распространения	Открытая лицензия GNU Affero General Public License версии 3, а также возмездная лицензия на условиях правообладателя
Дата составления документа	24.09.2026

1. Назначение программного обеспечения

Orchestra — платформа оркестрации автономных программных агентов, построенных на больших языковых моделях. Программа предназначена для организации в компании постоянно работающей команды таких агентов и для управления ею так же, как руководитель управляет сотрудниками: владелец формулирует цель, а разбиение цели на задачи, назначение исполнителей, контроль результата и приём работы выполняет агент-оркестратор.

Программа решает задачу, которую средства единичного вызова языковой модели не решают: удержание рабочего контекста и состояния исполнителя на протяжении суток и недель, изоляцию параллельно работающих исполнителей друг от друга, машинно-проверяемый приём результата и воспроизводимый учёт израсходованных ресурсов.

Область применения — автоматизация рутинных процессов организации: разработка и сопровождение программного обеспечения, обработка обращений клиентов, подготовка и согласование документов, ведение задач и отчётности, аналитические разборы.

2. Перечень функциональных характеристик

2.1. Управление парком агентов

- создание агента с заданием роли, модели, каталога проекта и границ ответственности;
- сохранение состояния агента в базе данных: агент переживает перезапуск платформы, продолжает прерванный диалог и восстанавливает историю;

- перевод простаивающего агента в спящий режим с освобождением дерева процессов и автоматическое пробуждение при поступлении сообщения;
- принудительная остановка агента с сохранением рабочей копии и сессии, а также полная архивация агента с проверкой незавершённой работы;
- переименование агента, смена его модели, смена системной подсказки и описания роли;
- ручное уплотнение рабочего контекста агента.

2.2. Иерархия и обмен сообщениями

- двухуровневая и трёхуровневая иерархия: владелец → оркестратор → суб-оркестраторы → исполнители;
- прямой обмен сообщениями между агентами без участия человека;
- надёжная доставка сообщений с квитанцией: отправитель получает идентификатор доставки и может проверить её фактический исход;
- доставка задания вновь создаваемому агенту с сохранением квитанции и возможностью повторной отправки, если доставка не состоялась;
- доставка сообщения агенту, находящемуся в работе, со вставкой в текущий ход.

2.3. Изоляция работ и сборка результата

- для каждого исполнителя создаётся отдельная рабочая копия репозитория системы контроля версий Git на собственной ветке;
- запрет на создание двух исполнителей с пересекающимися каталогами ответственности;
- проверка конфликтности работ двух исполнителей до слияния;
- слияние ветки исполнителя в основную ветку одним сжатым коммитом, с фиксацией связи между задачей и вошедшими в неё изменениями;
- ограничение на объём вставленных строк в одном слиянии с возможностью явного и записываемого отступления от него;
- запуск платформой (а не агентом) подмножества автоматических проверок, от результата которых зависит допуск слияния;
- блокировка слияния при отсутствии квитанции о проведённом ревью на точный снимок работы.

2.4. Взаимная проверка результата моделями разных поставщиков

- запуск ревью результата работы моделью другого поставщика, чем тот, что выполнял работу;
- многоаундовый спор исполнителя с ревьюером в рамках одной сохранённой сессии;
- сохранение вердикта ревью в виде файла-артефакта и квитанции в базе данных.

2.5. Ведение задач

- создание, изменение и просмотр задач с полями «название», «описание», «исполнитель», «приоритет», «стоимость», «статус»;
- автоматическая привязка задачи к агенту при выдаче задания и её закрытие при успешном приёме работы;
- связывание задачи с фактическими коммитами, вошедшими в основную ветку;
- приёмочные условия задачи: команда приёмки, перечень обязательных артефактов, закреплённый приёмочный оракул.

2.6. Фоновые задания и расписания

- отложенный запуск по таймеру;
- периодический запуск по расписанию в формате cron;
- наблюдение за файлом, за выводом команды и за потоком удалённой команды с пробуждением агента по совпадению с образцом;
- выполнение длительной команды с пробуждением агента по её завершении с кодом возврата;
- наблюдение за простым собственным деревом агентов.

2.7. Рабочие места оператора

- веб-интерфейс («дашборд») с потоковым обновлением: список агентов, их состояние, диалог, файлы рабочей копии, задачи, фоновые задания, коммиты;
- управление из мессенджера Telegram: текстовые сообщения, голосовые сообщения с автоматической расшифровкой, изображения и документы; отдельные темы группы под агентов;
- отправка оператору файлов, альбомов файлов и построенных платформой диаграмм;
- публикация защищённого от индексирования HTML-артефакта по ссылке с ограниченным сроком жизни.

2.8. Учёт ресурсов и ограничение расхода

- учёт израсходованных ходов, токенов и расчётной стоимости по каждому агенту, модели и периоду;
- панель квот поставщиков с прогнозом исчерпания;
- автоматический отказ в создании новых агентов при приближении к пределу подписки, с жёстким пределом, который не снимается;
- временное снятие мягкого ограничения оператором на срок не более шести часов.

2.9. Память и знания проекта

- лексическая база знаний проекта в каталоге репозитория, оглавление которой платформа вставляет в системную подсказку агента;
- личная память агента, подключаемая к его подсказке;
- поиск по базе знаний, по личным заметкам и по журналам.

2.10. Расширение набора инструментов

- подключение внешних инструментов по протоколу Model Context Protocol как на уровне проекта, так и на уровне отдельного агента;
- запрет отдельных инструментов конкретному агенту, сохраняющийся при перезапуске.

2.11. Однократные параллельные сценарии

- описание сценария на языке Python с параллельным и конвейерным запуском одноразовых агентов, с ограничением числа вызовов, степени параллельности и бюджета;
- журнал прогона с возможностью продолжения после прерывания без повторного выполнения уже завершённых шагов.

3. Требования к системе

Параметр	Значение
Тип программы	серверное приложение с веб-интерфейсом
Операционная система	Linux (проверено на Ubuntu 24.04 LTS и 26.04 LTS)
Среда исполнения	CPython версии 3.12 или новее
Дополнительно	Node.js версии 18 или новее — для внешних агентских клиентов командной строки
Хранилище данных	SQLite, файл в каталоге данных приложения
Минимальные ресурсы	2 вычислительных ядра, 4 ГиБ оперативной памяти, 10 ГиБ дискового пространства
Рекомендуемые ресурсы	8 вычислительных ядер, 16 ГиБ оперативной памяти, 100 ГиБ дискового пространства
Сетевые требования	исходящий доступ к выбранному поставщику языковых моделей

4. Входные и выходные данные

Входные данные. Текст задания от оператора; голосовые сообщения, изображения и документы, поступающие из мессенджера; содержимое файлов рабочего репозитория; ответы внешних инструментов, подключённых по протоколу Model Context Protocol.

Выходные данные. Изменения в рабочем репозитории, оформленные коммитами и ветками; сообщения оператору и другим агентам; записи задач и их статусов; файлы-артефакты отчётов и вердиктов ревью; записи журнала работы и учёта израсходованных ресурсов.

5. Язык интерфейса

Графический пользовательский интерфейс программного обеспечения реализован на русском языке. Проверка 22.09.2026 охватила 41 экран; после исключения имён собственных, идентификаторов и путей, ответов моделей, служебных строк сервера и единиц измерения латинских надписей, требующих перевода, не обнаружено. Перед подачей повторяется ручной аудит экранов командой `uv run --frozen python .orchestra/tasks/V-584/audit_screens.py` в репозитории Orchestra; его отчёт сохраняется в `.orchestra/tasks/V-584/otchet.md`.

6. Ограничения

Программа не предназначена для обработки сведений, составляющих государственную тайну. Программа не относится к средствам защиты информации; защита конфиденциальной информации не является её основной функцией.

Генеральный директор ООО «СИДОН» М. И. Астраханцев

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат 025D50920018B4EC8441E43FD2DF707187
Владелец Астраханцев Максим Игоревич
Действителен с 25.03.2026 по 25.06.2027