

# Metabot Agent Stack (MAS)

Мир переходит от интерфейсов кликов к взаимодействию с ИИ через **естественный язык**. Ассистенты и агенты становятся основой нового поколения программ — они умеют **понимать контекст, учиться и действовать**.

Но как программировать таких агентов — без сложной архитектуры и кода?

Ответ — **MAS (Metabot Agent Stack)**. Это не просто набор инструментов.

MAS — это **фреймворк проектирования разумных агентов**, встроенный в инфраструктуру Metabot.

## Что такое MAS

**MAS (Metabot Agent Stack)** — это архитектурный стек, который объединяет:

- **LLM-интеграцию** (OpenAI, Claude, GigaChat, Gemini и др.);
- **Векторные базы знаний** (PgVector, ClickHouse, Elastic);
- **Low-code/Full-code сценарии и скрипты** на JavaScript;
- **Интеграцию с API и CRM/ERP системами**;
- **Память и атрибуты** для хранения контекста;
- **Агентные пайплайны и маршрутизацию намерений**.

MAS разворачивается поверх платформы Metabot и превращает её из конструктора чат-ботов в **полноценную мультиагентную среду**.

## Архитектура MAS

MAS построен по принципу **мультиагентных пайплайнов**, где каждый агент выполняет конкретную роль, а их взаимодействие обеспечивает целостное поведение системы.

Компонент	Назначение
<b>Intent Router</b>	Анализирует запрос пользователя и направляет его в нужный агент.

<b>RAG-агент</b>	Извлекает информацию из внутренних документов, формирует контекст и передаёт в LLM.
<b>SQL-агент</b>	Строит SQL-запросы к базам данных и формирует аналитические ответы.
<b>Knowledge Base Search</b>	Обеспечивает семантический поиск по базе знаний.
<b>LLMClient</b>	Управляет вызовами языковых моделей и их конфигурацией.
<b>Snippets</b>	Хранят параметры, промпты и настройки агента.
<b>Tracing</b>	Отслеживает взаимодействия и reasoning-потоки между агентами.

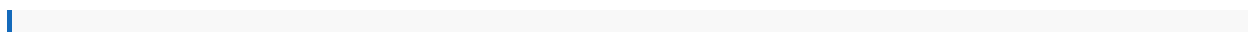
MAS поддерживает работу **в мессенджерах, web-чатах, корпоративных интерфейсах** и может интегрироваться с BI-инструментами и DataLens.

## MAS vs другие фреймворки

Функциональность	MAS (Metabot)	LangChain/LangGraph	LlamaIndex	Botpress / Rasa
Память и контекст	☐	△ вручную	△ ограничено	☐ FSM
Поддержка RAG	☐	☐	☐	△ частично
Асинхронность	☐	△ зависит от infra	△	△
Flow Logic	☐ (визуально)	△ вручную	△	☐
Интеграции (API, CRM, SQL)	☐ встроенные	☐	△	△
JS + Low-Code	☐	☐ (Python)	☐ (Python)	☐
Поддержка мессенджеров	☐	☐	☐	☐
Запуск без кода	☐	☐	☐	△

**MAS** занимает нишу между классическими LLM-фреймворками и бизнес-платформами: он даёт **архитектурную гибкость LangChain/LangGraph**, но упрощает программирование.

## MAS как философия



“Нейросети не мыслят. Интеллект рождается в инфраструктуре вокруг них.”

— Юрий Гарашко, «Сети жизни vs Deep Learning»

MAS воплощает идею **событийно-коммуникативной архитектуры**:

агент не существует изолированно, а живёт в потоке событий, контекстов и действий.

Каждый агент хранит собственную память, имеет зону ответственности и взаимодействует через Intent Router.

Вся система строится вокруг **атрибутов лида** — единой модели состояния, что позволяет связывать диалоги, данные и действия без написания backend-кода.

---

## Что даёт MAS разработчикам и бизнесу

- Быстрая сборка LLM-ассистентов под любые задачи.
- Поддержка RAG-поиска и работы с корпоративными базами.
- Интеграция с CRM, ERP, LMS, аналитикой и BI-дашбордами.
- Автоматизация поддержки, обучения и документооборота.
- Простое масштабирование от одного агента к мультиагентной сети.

MAS позволяет компаниям перейти от «чат-ботов с ИИ» к **когнитивным операционным системам** — где агенты не просто отвечают, а **управляют процессами и принимают решения**.

---

## Рекомендуемые материалы

- [Работа с LLMClient](#)
  - [Knowledge Base и RAG](#)
  - [Snippets и конфигурация агентов](#)
  - [Tracing и отладка reasoning](#)
  - [Интеграция с внешними API](#)
-

Следующий шаг: изучите [уроки по MAS](#) и попробуйте создать собственного агента — участника предстоящего *Баттла ассистентов и рыцарей!*

---

Версия #2

Artem Garashko создал 12 November 2025 13:49:22

Artem Garashko обновил 12 November 2025 14:03:47