

Вопросы и ответы

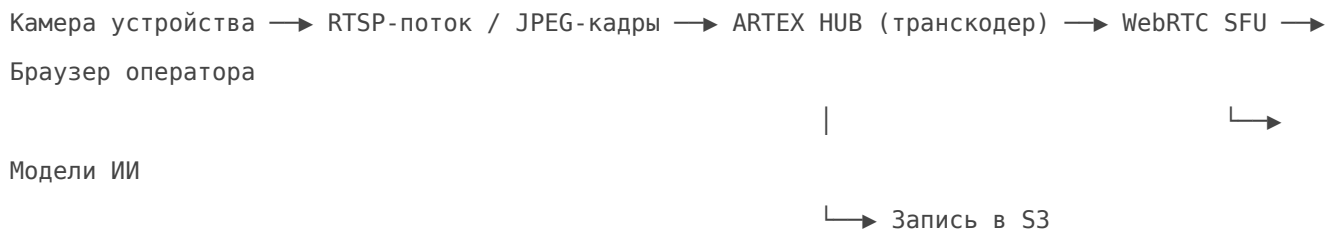
На странице описано, по каким протоколам участники обмениваются данными с ХАБом платформы и как движется каждый поток.

Таблица потоков

№	Связь	Поток	Протокол
1	Робот ↔ ХАБ	Телеметрия, признаки	Socket (в текущей версии SDK — Socket.IO, namespace <code>/robot</code>)
2	Робот ↔ ХАБ	Управление	Socket (Socket.IO, событие <code>command</code>)
3	Робот → ХАБ	Видео	RTSP / WebRTC или «кадры» через Socket (в текущей версии — JPEG-кадры через Socket.IO)
4	Внутри ХАБа	Все потоки	Общая память — неблокирующий кэш (словарь)
5	ХАБ ↔ оператор	Контроль и обучение робота	WebSocket (данные и команды), WebRTC (видео)
6	ХАБ ↔ модели ИИ (вычислительная среда)	Телеметрия, управление, видео	Socket (телеметрия и управление), WebRTC / Socket (видео)
7	Модули сервера между собой	Служебные вызовы	REST API + gRPC
8	IoT-датчики → ХАБ	Показания датчиков	MQTT (publish/subscribe, QoS 1) — планируется
9	Администратор → сервер	Обслуживание	SSH

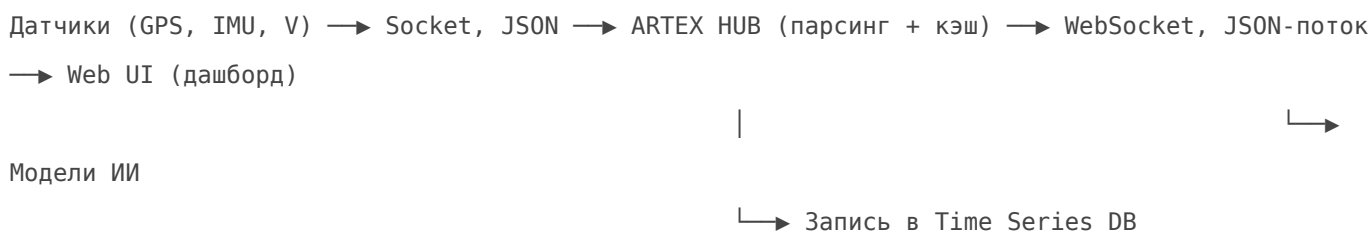
Протокол робота подробно описан в книге «ПО для робота» на странице [Протокол обмена с сервером](#).

1. Вопросы



- Робот передаёт видео либо потоком RTSP, либо отдельными кадрами (JPEG, качество 80) через Socket.
- ХАБ хранит в кэше **актуальный кадр** (или набор из X последних кадров) и направляет его в вычислительную среду и оператору.
- Для браузера видео преобразуется в WebRTC; SFU ретранслирует поток нескольким получателям без повторного кодирования.
- Параллельно видео записывается в объектное хранилище S3.

????? 2. ???????????



- Телеметрия передаётся в формате JSON, например: `{"speed": 1.2, "gps": [55.65, 37.66], "battery": 11.8, "imu": {...}, "timestamp": 1790155251.12}`.
- Состав полей не фиксирован: робот может добавить новый датчик без изменения сервера и интерфейса.
- ХАБ хранит **актуальный вектор** телеметрии с учётом времени создания и раздаёт его потребителям.

????? 3. ???????????



- Команда содержит автора (`author`) и полезную нагрузку вида `{тип_команды: команда}`.
- ХАБ проверяет права автора на работа и применяет правило приоритета: команда оператора имеет приоритет над командой модели (режим «подстраховки»).

- Команды `Стоп` и `DISARM` обрабатываются вне очереди.

???????? ????????????? ??????????

Параметр	Целевое значение
Камера → оператор	< 200 мс
Команда → привод	< 50 мс
Частота телеметрии	до 30 сообщений/с (JSON)
Частота отправки данных SDK по умолчанию	20 сообщений/с (<code>max_send_freq = 0.05</code> с)

???????????? ????? ?????????????????????????????????????

1. Робот открывает соединение с сервером и передаёт паспорт устройства (`name`, `type`, `telem_can`, `video_can`).
2. Сервер присваивает подключению идентификатор (SID) и регистрирует робота в реестре устройств.
3. Сервер периодически отправляет **heartbeat**: запрашивает динамическое состояние (`d_meta`) и сигнал готовности (`Ready`).
4. Получив `Ready`, робот запускает циклы отправки телеметрии и видео.
5. Сервер пересылает роботу команды от операторов и моделей.
6. При обрыве связи SDK автоматически переподключается (до 5 попыток с задержкой 1-5 с); сервер помечает робота как отключённого и уведомляет интерфейс (порт `robot.disconnected`).

???????? ?????????? ?????????? ??????

ХАБ предоставляет операторам и моделям два базовых метода работы с общей памятью:

Метод	Назначение
<code>push(key, value)</code>	Записать новое значение (кадр, вектор телеметрии, команду) с отметкой времени. Не блокирует читателей.
<code>get(key)</code>	Получить последнее актуальное значение. Не ждёт появления нового значения.

Рекомендуемая схема именования ключей — `<робот>.<поток>[.<канал>]`, например `robot_1.video.rgb`, `robot_1.telemetry`, `robot_1.command`.

