

Архитектура единого входа

Надёжно внутри · проект архитектуры, редакция 15.10.2026

Как устроен вход в системы, которые мы делаем: кто где хранит данные, через что входят пользователи и почему один сбой не останавливает всё. Схемы — в нотации C4: от общей картины к деталям.

Принципы

Кубики, а не монолит

Вход собирается из готовых проверенных блоков: пароль, второй фактор, телефон, МАХ, внешние системы. Проекту — только нужные.

Данные — у владельца

Профили пользователей и роли хранит сам проект. У системы заказчика — его пользователи и его база, без адресов Kelesida.

Никаких скрытых зависимостей

Всё, куда проект обращается наружу, записано в его сценарии со ссылкой на требование. Остальное не пройдёт проверку перед выкладкой.

Только расширяем

Новый способ входа — новый кубик. Ядро и подключённые проекты не меняются, поэтому ничего не ломается.

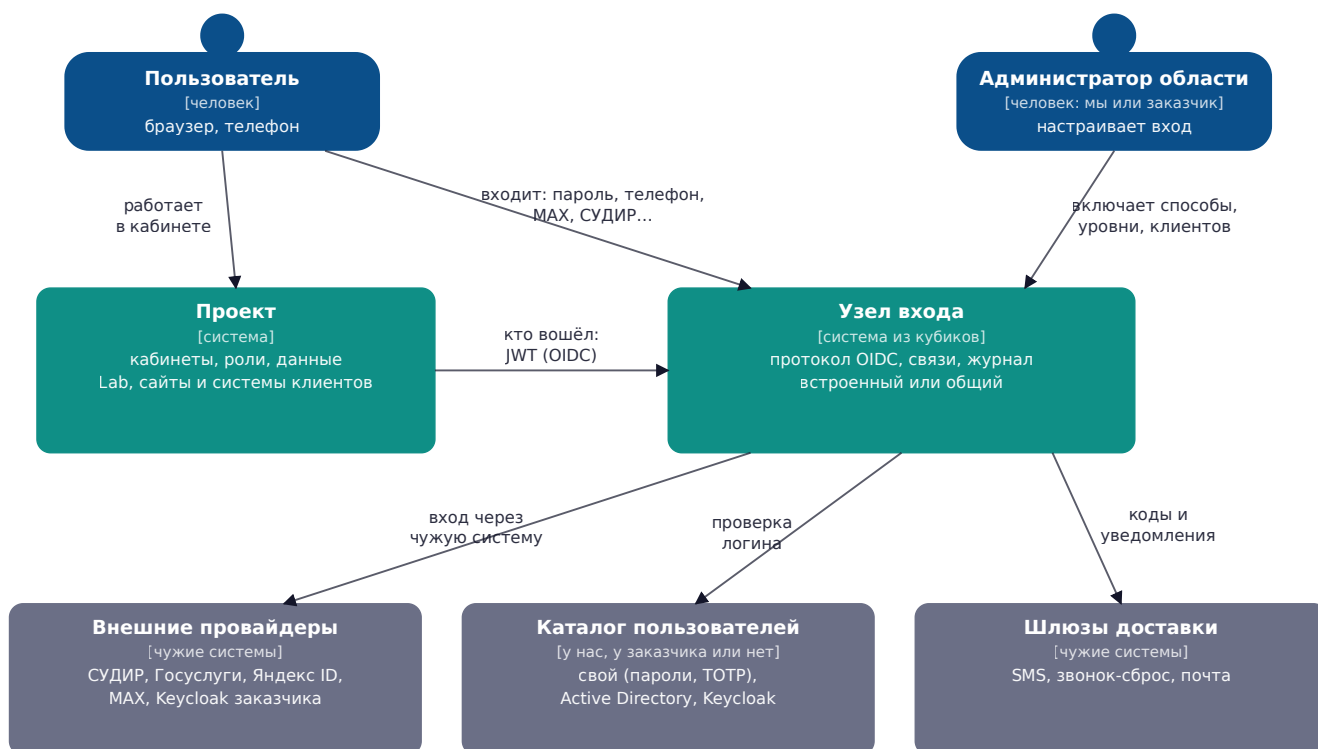
Один человек — один вход

Пароль сегодня, МАХ завтра — система понимает, что это один пользователь. Связывает только тот, кто доказал владение обоими входами.

Сбой не останавливает работу

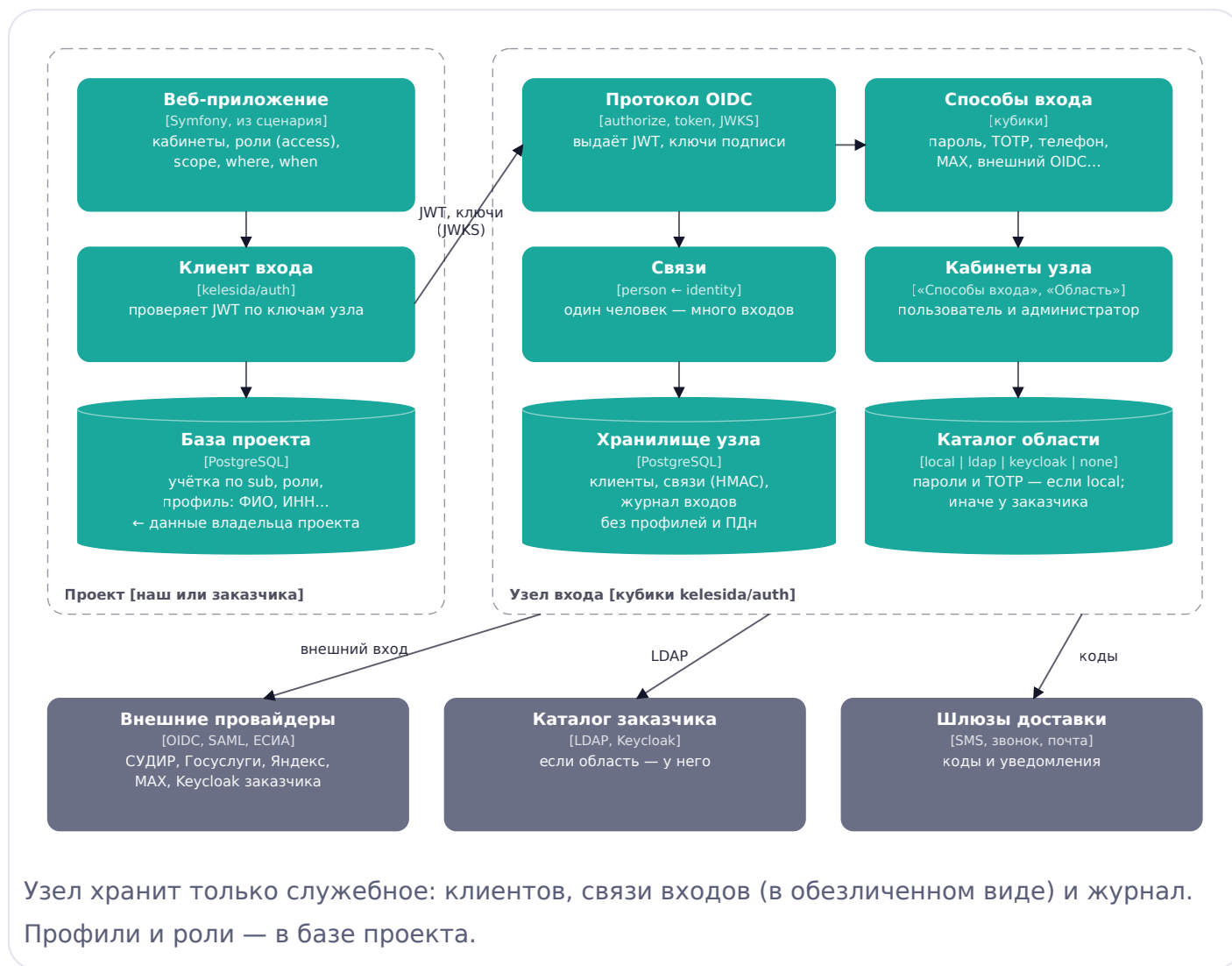
Кто уже вошёл, продолжает работать. Для администратора есть аварийный вход без узла.

Кто участвует

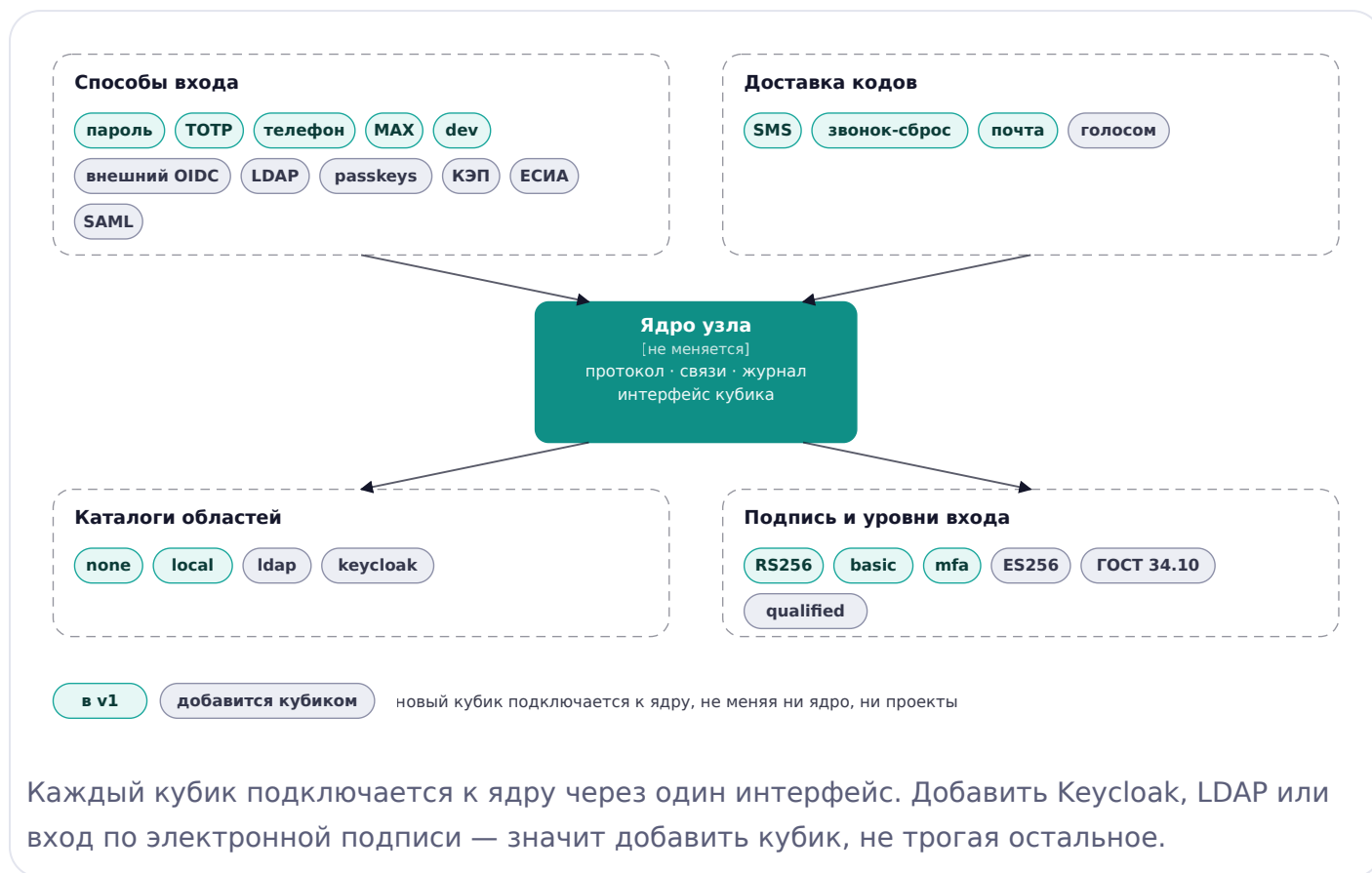


Пользователь работает в проекте, а входит через узел. Узел проверяет его сам (пароль, телефон) или через чужую систему (СУДИР, Госуслуги, MAX) и сообщает проекту, кто вошёл.

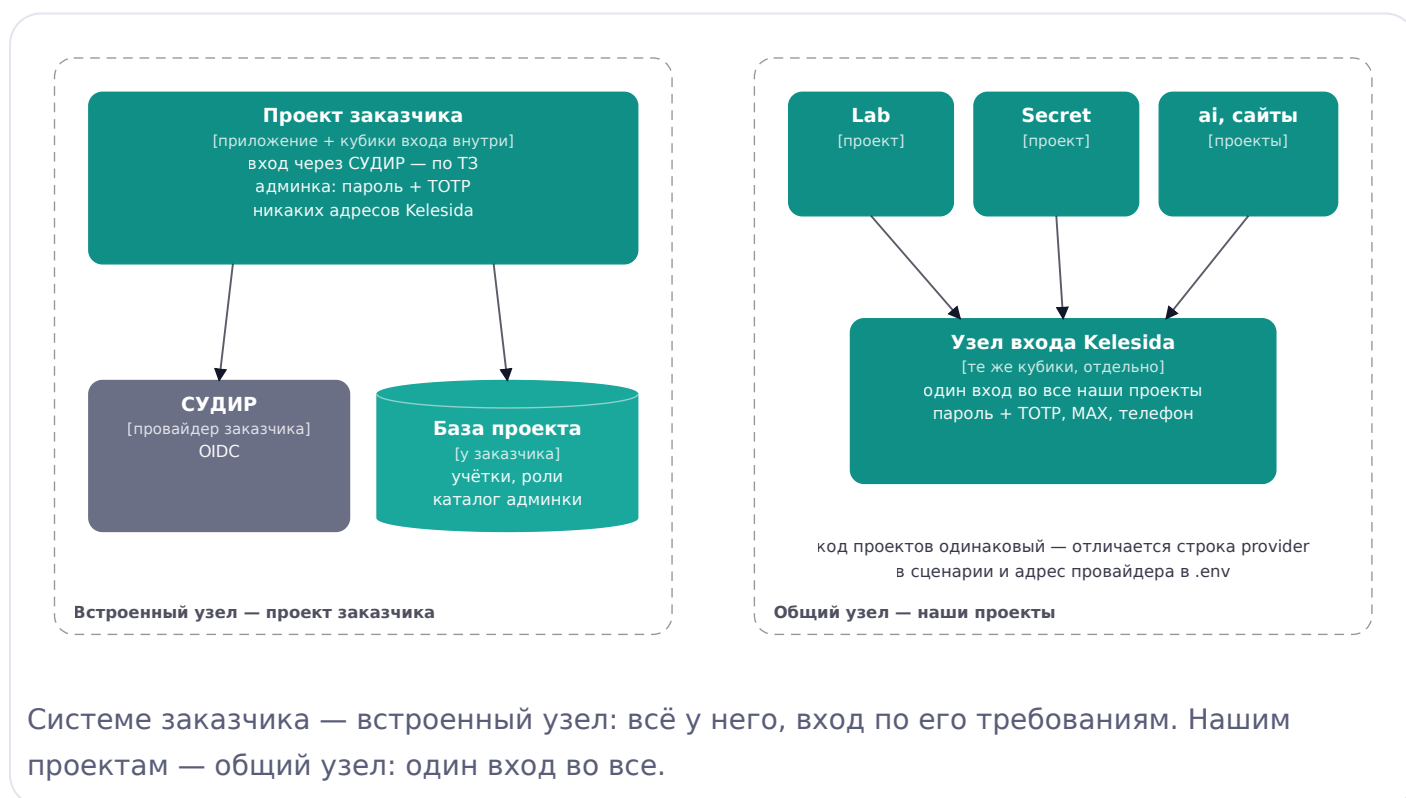
Из чего состоят проект и узел



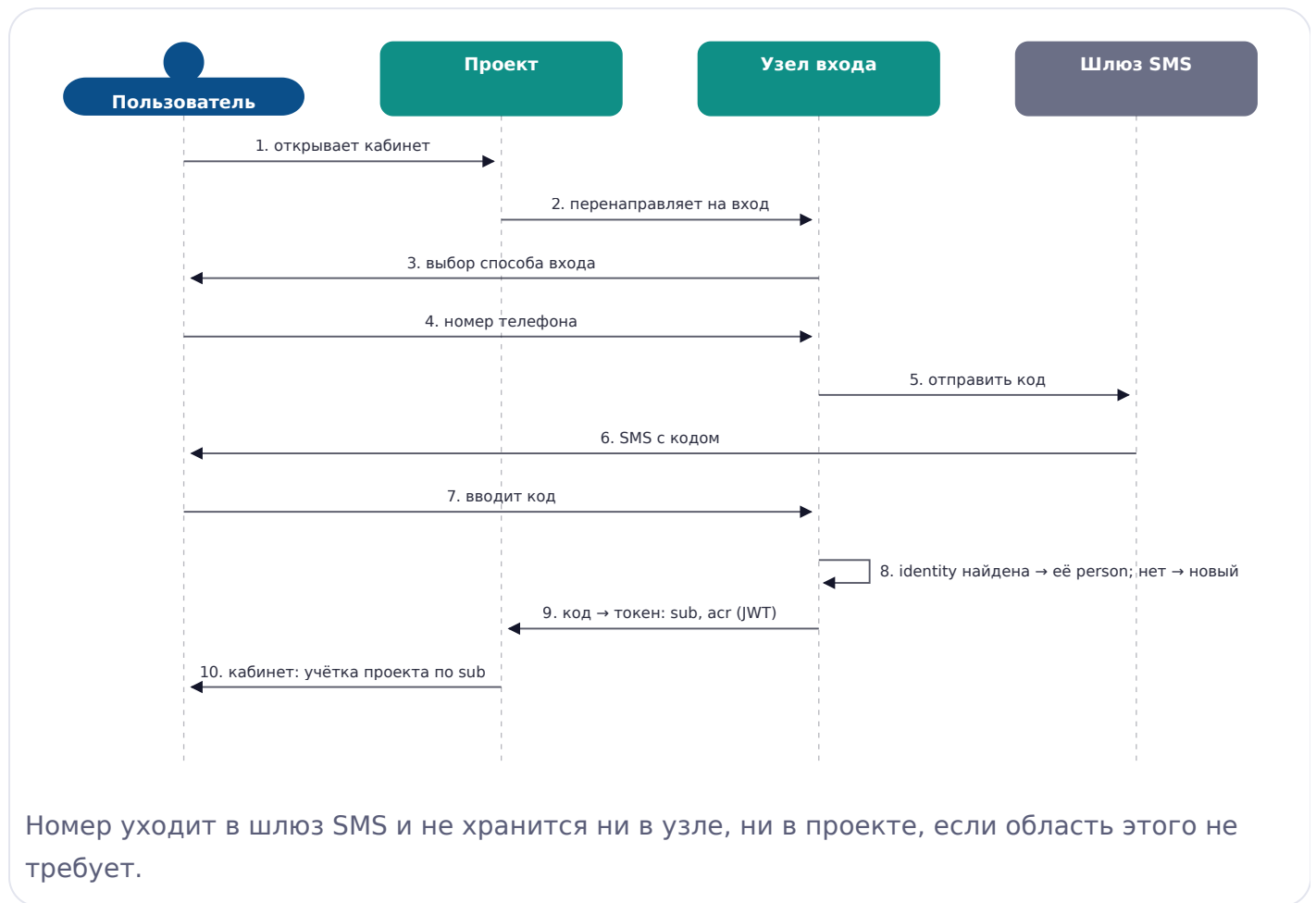
Кубики узла



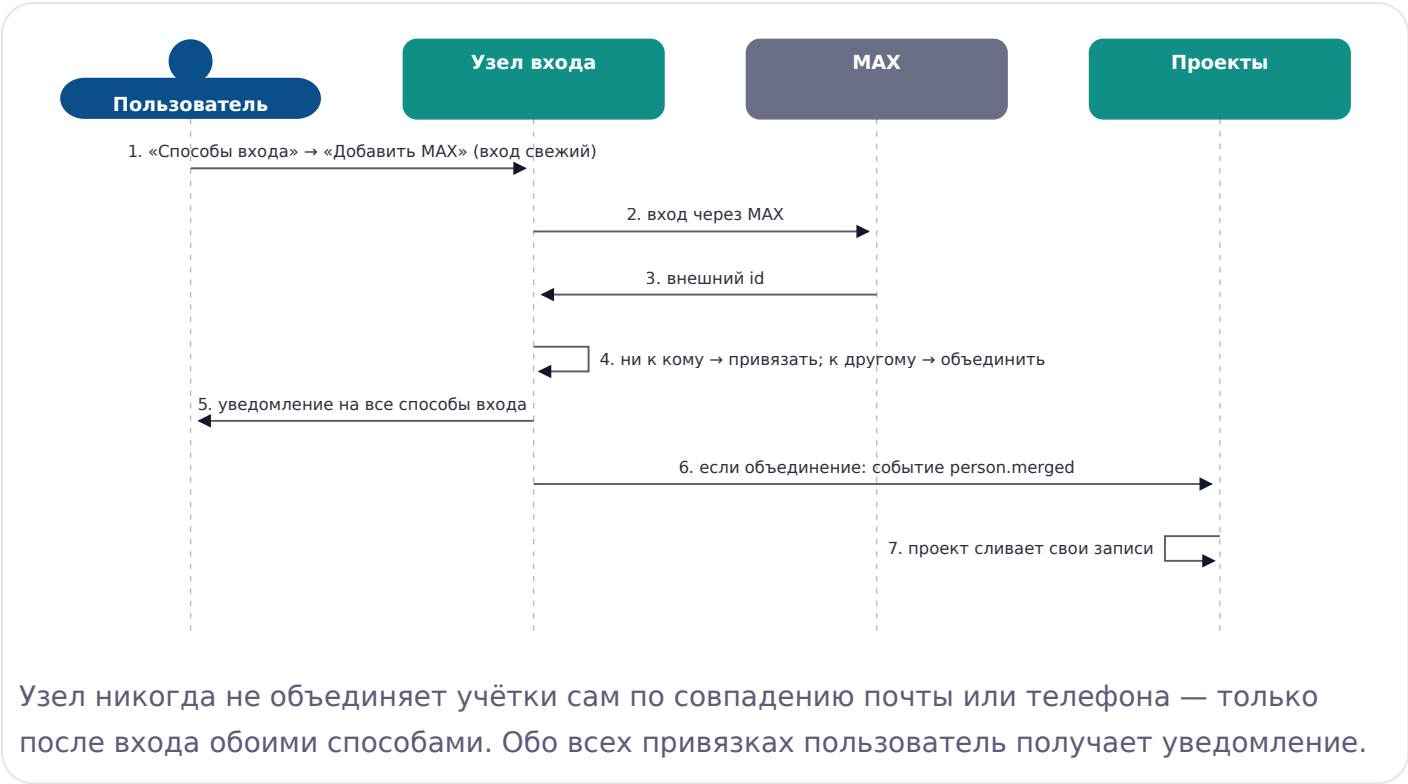
Два варианта сборки



Вход по телефону



Несколько входов — один пользователь



Кто за что отвечает

Часть	Где	Что хранит	Чьи данные
Учётка, роли, профиль	проект	sub, роли, ФИО, ИНН...	владельца проекта
Клиент входа	проект	ничего	—
Протокол OIDC	узел	клиенты, ключи подписи, короткие коды	—
Связи входов	узел	хеши внешних id, маски («MAX ...1234»)	обезличено
Журнал	узел	кто, когда, каким способом	обезличено
Каталог «свой»	узел области	логины, хеши паролей, TOTP	владельца области
Каталог заказчика	у заказчика	всё своё	заказчика
Внешние провайдеры, шлюзы	третьи стороны	своё	провайдера

Токены

Токен	Формат	Зачем
id_token	JWT, подпись узла	кто вошёл и как: sub, уровень входа
access_token	JWT, 5–15 минут	доступ к API; проверяется подписью, без обращения к узлу
refresh_token	случайная строка	продлить вход; хранится в узле и может быть отозван
сессия в кабинете	cookie, на сервере	работа в браузере — не JWT, чтобы её нельзя было украсть и нельзя было не отозвать

Подпись — RS256, при необходимости ES256 или ГОСТ 34.10-2012. Алгоритм «none» и общие секреты клиент не принимает.

Если узел недоступен

- Кто уже вошёл — продолжает работать: сессия живёт в проекте, токены проверяются по сохранённым ключам.
- Новые входы ждут восстановления узла. Узел небольшой и поднимается из репозитория за минуты; ключи, связи и каталог — в ежедневной резервной копии.
- Администратор проекта может войти аварийно — одноразовой ссылкой с сервера, без узла.
- У системы заказчика свой встроенный узел, и от нашего она не зависит вовсе.