

# Ферма как сервис

Кожемякин Никита  
SberDevices

## Обо мне

14 лет в ИТ.

Развивал и поддерживал сетевые инфраструктуры дата-центра, завода и логистического хаба.

В SberDevices отвечаю за техническое сопровождение проекта «Ферма».



## О чем поговорим

- Развитие Фермы: от прототипа хаба и OpenSTF к собственному решению и мини Дата-центру;

## О чем поговорим

- Развитие Фермы: от прототипа хаба и OpenSTF к собственному решению и мини Дата-центру;
- Что есть сейчас: железо, софт, метрики, графики;

## О чем поговорим

- Развитие Фермы: от прототипа хаба и OpenSTF к собственному решению и мини Дата-центру;
- Что есть сейчас: железо, софт, метрики, графики;
- Поговорим про «грабли»: софт, железо, девайсы.

# Что такое Ферма?

Инфраструктурное решение для удаленного тестирования устройств, производимых компанией SberDevices



# Устройства SberDevices



# Устройства SberDevices



# Устройства SberDevices



# Устройства SberDevices



# Что было до Фермы?

- В начале не было ничего;

## Что было до Фермы?

- В начале не было ничего;
- Появились первые устройства;

## Что было до Фермы?

- В начале не было ничего;
- Появились первые устройства;
- Потом возникла потребность: необходимо тестировать эти устройства.

# Как тестировать девайсы без Фермы?

Как тестировать девайсы без Фермы?

— Руками

# 2020. Covid-19

## 2020. Covid-19

— Все разъехались;

## 2020. Covid-19

- Все разъехались;
- Всем нужны девайсы;

## 2020. Covid-19

- Все разъехались;
- Всем нужны девайсы;
- Девайсов не хватает.

2020. Начинаем строить Ферму

## Формируем требования

- Android;

## Формируем требования

- Android;
- Быстро запустить сервис;

## Формируем требования

- Android;
- Быстро запустить сервис;
- Нет ресурса на разработку своего решения.

Нашли решение

Решение: OpenSTF



# Первая отладка



# The end

The end?!

# 2021. Удаленка и первый запуск

## А как подключить SberBox и SberPortal?



# Первые проблемы

Подключаем SberPortal



Первые решения

Собираем свой USB хаб



## Первые решения

Собираем свой USB хаб

- Открывает ADB на релизных девайсах;



## Первые решения

### Собираем свой USB хаб

- Открывает ADB на релизных девайсах;
- Управляет питанием девайсов.



Почему мы не взяли готовые USB хабы?

## Почему мы не взяли готовые USB хабы?

- Устройства при тестировании будут зависеть;

## Почему мы не взяли готовые USB хабы?

- Устройства при тестировании будут зависать;
- При нестабильной прошивке будут падать в recovery;

## Почему мы не взяли готовые USB хабы?

- Устройства при тестировании будут зависать;
- При нестабильной прошивке будут падать в recovery;
- Наш SmartHub дает доступ до отладочных интерфейсов устройства.

## Победа?!

- Устройства стали доступны через ADB;
- Пришло время предоставить к ним удалённый доступ;

## Победа?!

- Устройства стали доступны через ADB;
- Пришло время предоставить к ним удалённый доступ;
- OpenSTF предоставлял доступ к SberPortal через ADB;
- Обеспечивал выполнение команд через Shell-терминал по сети;
- Транслировал экран устройств в веб-интерфейсе для ручных манипуляций пользователям.

# Пробуем подключить SberBox



# Новые проблемы возможности



## Новые возможности

- SberBox не имеет touch экрана, как на телефоне;

## Новые решения

- SberBox не имеет touch экрана, как на телефоне;
- Кастомизируем OpenSTF, пересобираем minitouch;

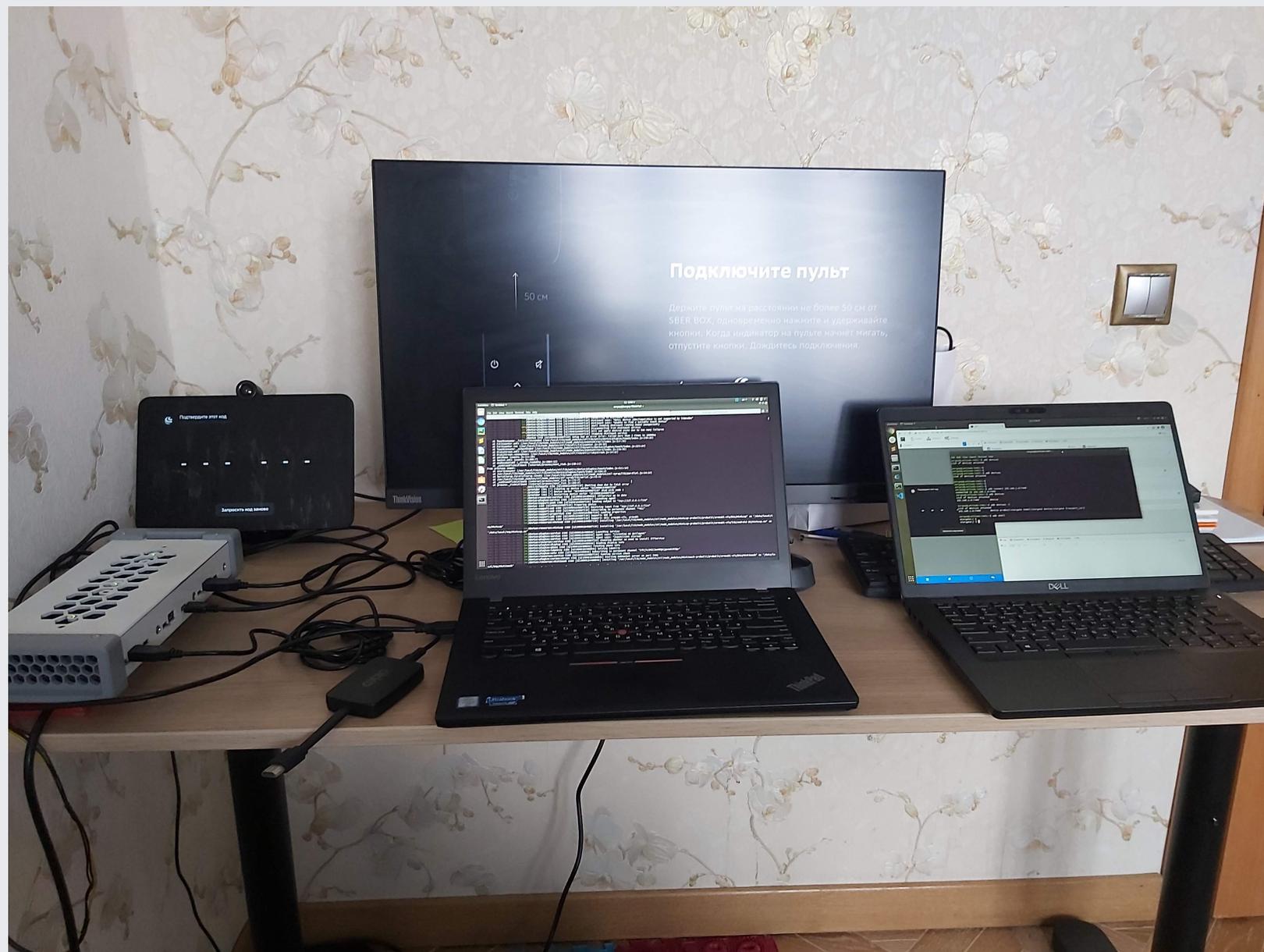
## Новые решения

- SberBox не имеет touch экрана, как на телефоне;
- Кастомизируем OpenSTF, пересобираем minitouch;
- Теперь надо патчить каждую новую версию OpenSTF 👍

Первый запуск

3 девайса

Февраль 2021



# Новый USB хаб

V1



V2



Первое расширение

12 девайсов

Июнь 2021



# 2022. Закат OpenSTF

# OpenSTF и Linux

Новое устройство с Linux на борту



# OpenSTF и Linux

Новое устройство с Linux на борту

— Нет getprop;



# OpenSTF и Linux

Новое устройство с Linux на борту

- Нет getprop;
- Нет дисплея;



# OpenSTF и Linux

Новое устройство с Linux на борту

- Нет getprop;
- Нет дисплея;
- Есть ADB 👍



## Начинаем чинить

- Подкладываем самописный shell-скрипт на девайс, который притворяется getprop;

## Начинаем чинить

- Подкладываем самописный shell-скрипт на девайс, который притворяется getprop;
- Вырезаем из OpenSTF всё, что связано с дисплеем при инициализации SberBoom;

## Начинаем чинить

- Подкладываем самописный shell-скрипт на девайс, который притворяется getprop;
- Вырезаем из OpenSTF всё, что связано с дисплеем при инициализации SberBoom;
- Добавляем иконки SberBoom и других девайсов в OpenSTF.

# Getprop

```
#!/bin/sh

CONFIG_FILE=/SOME_PATH/getprop_config.txt

if [ $# -eq 0 ];
then
    cat $CONFIG_FILE
else
    # grep -Po "(?<=(\[\$1]: \[))(\w+)" $CONFIG_FILE
    grep "$1" $CONFIG_FILE | awk -F"[" '{print $NF}' | awk -F"]" '{print $1}'
fi
```

# Фермерская лавка

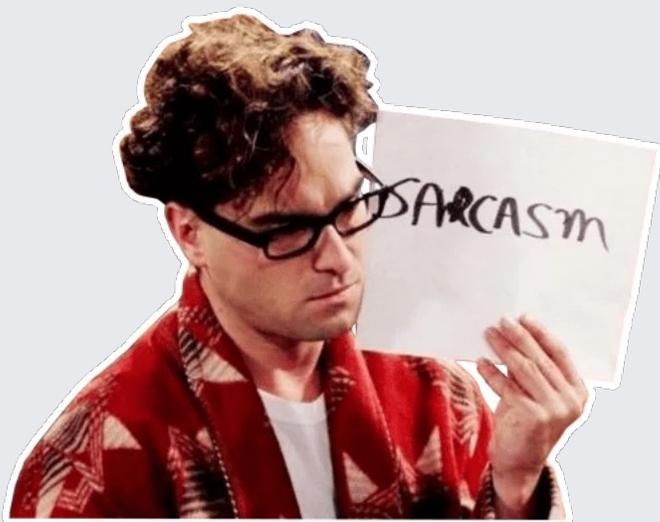
Как же хорошо  
иметь еще более  
кастомную версию  
OpenSTF 👍

The screenshot displays the OpenSTF web interface. At the top, there is a navigation bar with icons for Control, Devices, Groups, and Settings. On the right side of the navigation bar, there are links for Contact Support, Logout, and a user profile icon labeled 'AVTELESHOV'. Below the navigation bar, a dashboard section shows four key metrics: 'TOTAL DEVICES' with a value of 53, 'USABLE DEVICES' with a value of 42, 'BUSY DEVICES' with a value of 6, and a user profile icon with a value of 6. The main content area is titled 'Devices' and contains a grid of device cards. Each card represents a device and includes an image, a name, and a status. The devices shown include SberBox, SmartTV, SberPortal, H-LED32FS5..., SberBoom, and SberBox Top. Some cards have buttons like 'Stop Using' or 'Stop Automation'. The status of the devices varies, with some being 'Disconnected' or 'Preparing'.



# Фермерская лавка

Как же хорошо  
иметь еще более  
кастомную версию  
OpenSTF 👍



The screenshot shows the OpenSTF web interface. At the top, there's a navigation bar with icons for Control, Devices, Groups, and Settings. On the right, there are links for Contact Support, Logout, and a user profile for AVTELESHOV. Below the navigation bar, a dashboard displays four key metrics: TOTAL DEVICES (53), USABLE DEVICES (42), BUSY DEVICES (6), and AVTELESHOV (6). The main content area is titled "Devices" and shows a grid of device cards. Each card represents a device with an icon, a name, and a status. The devices include SberBox, SberPortal, SmartTV, H-LED32FS5..., SberBoom, and SberBox Time. Some cards have buttons like "Stop Using", "Stop Automation", "Preparing", and "Disconnected".



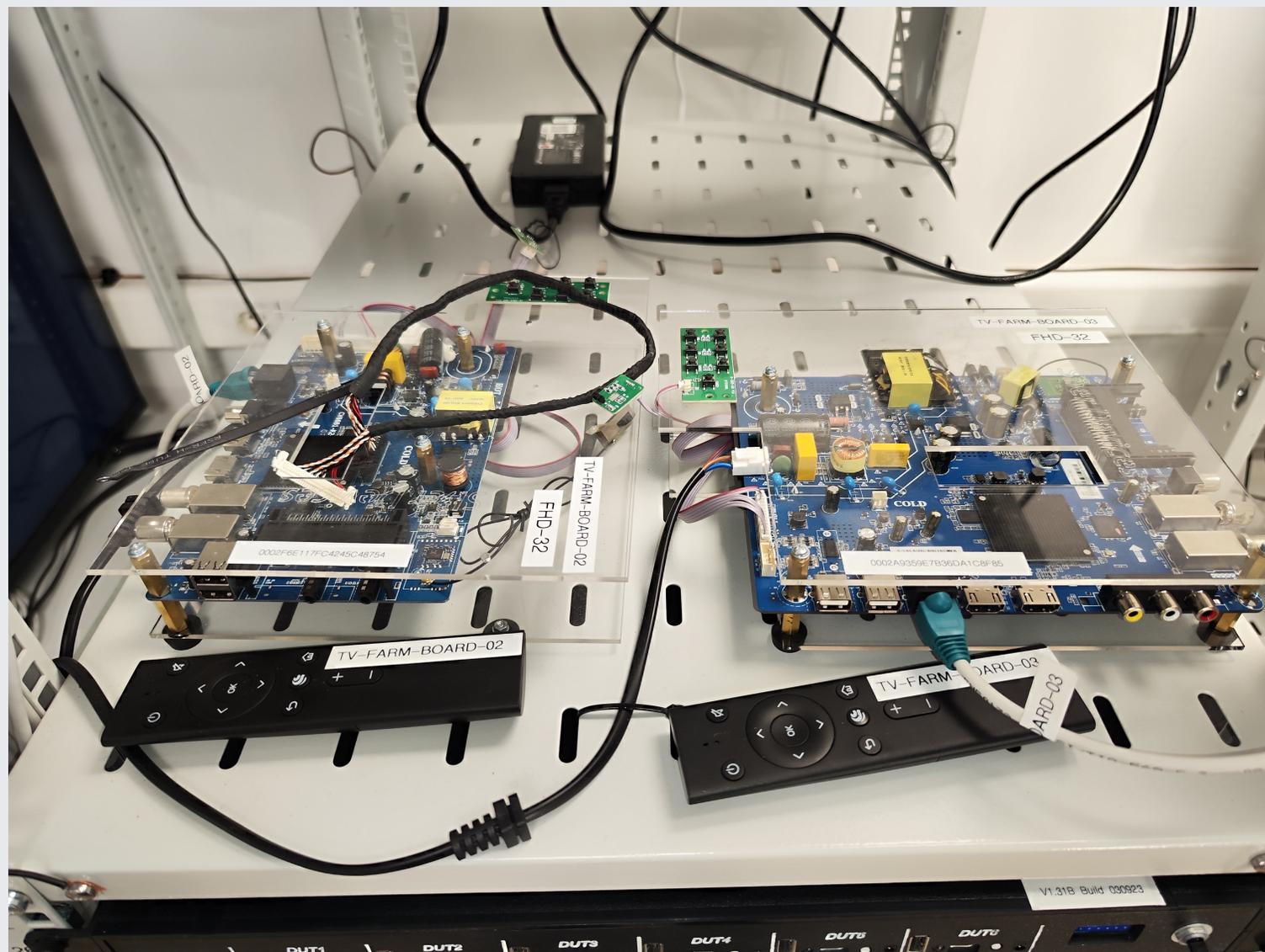
# Телевизоры на Ферме

Добавляем 32 и 43 дюймовые ТВ



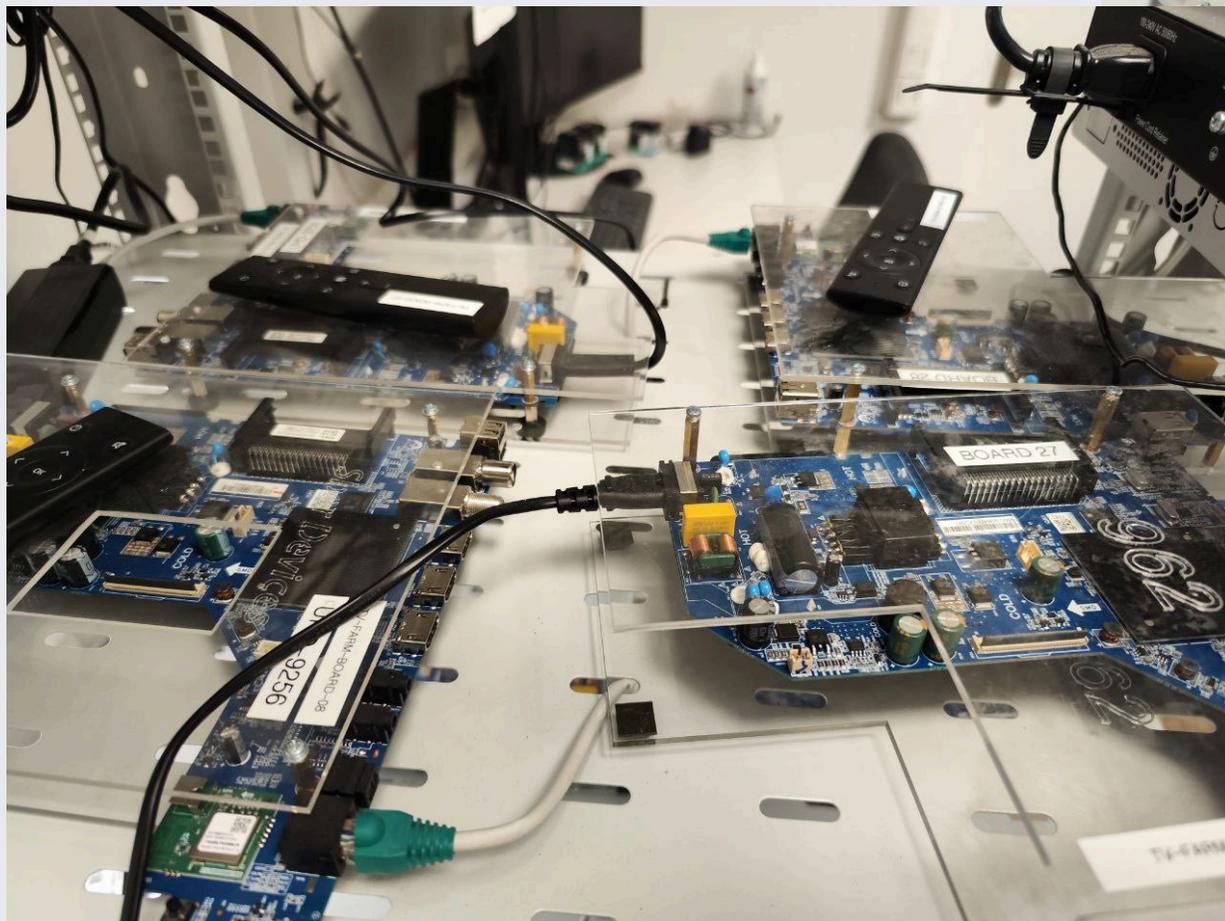
# Телевизоры на Ферме

«закатываем» в акрил



Телевизоры на Ферме

ещё больше

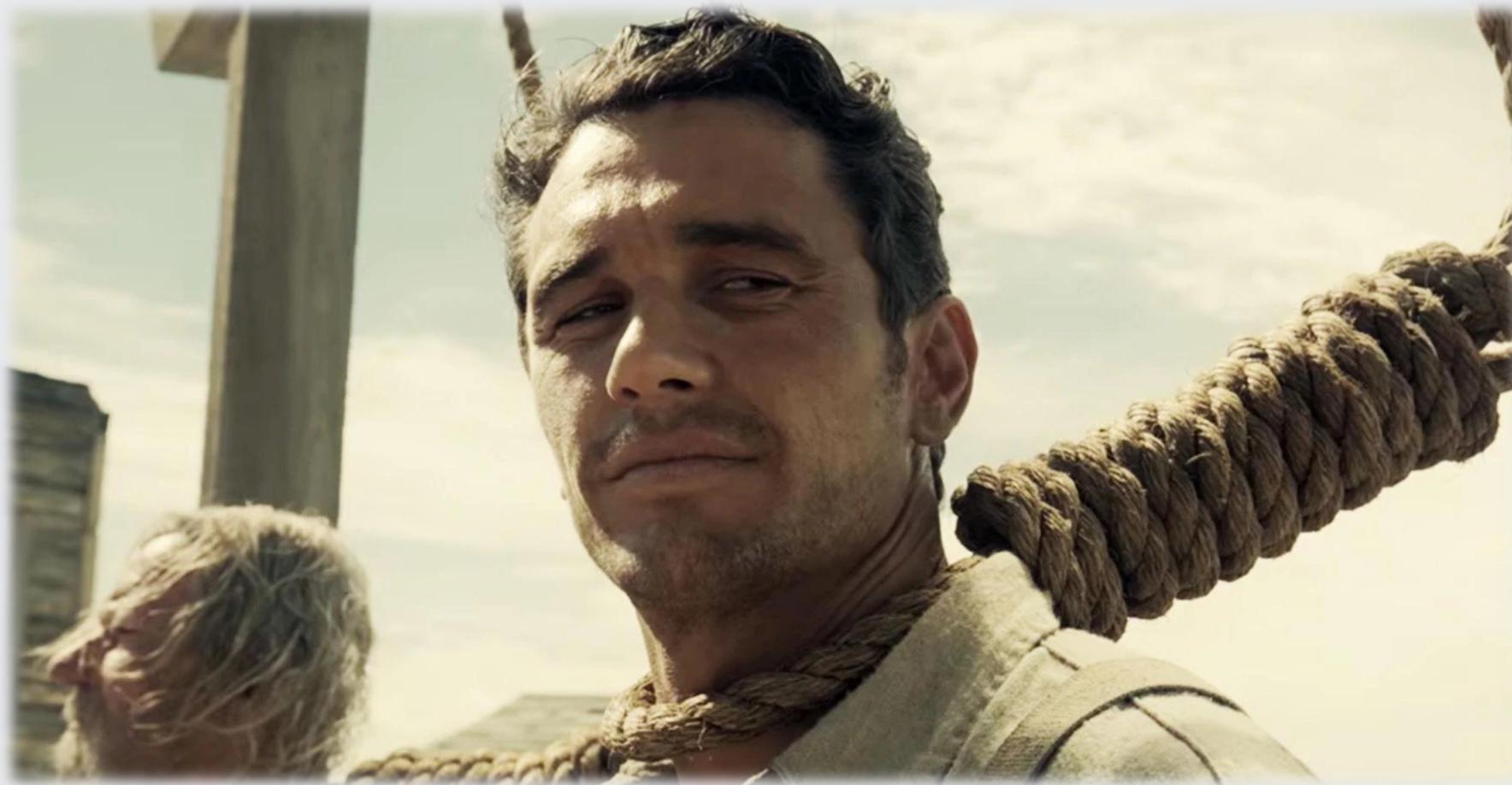


# Телевизоры на Ферме

И ещё больше



## Телевизоры на Ферме «отваливаются»



## Телевизоры на Ферме «отваливаются»

А потом пришел OOM killer...

```
[32m[347381.004113] [0mto free 36476kB on behalf of 'kswapd0' (888) because
[32m[347381.004113] [0mcache 149028kB is below limit 73728kB for oom_score_adj 200
[32m[347381.004113] [0mFree memory is -1872kB above reserved
[32m[347381.004120] [0mTotal_free = 70388kB, free_cma=4304kB, Totalreserve_pages = 34316kB, MAPPED = 120372kB
[32m[347381.004125] [33mlowmemorykiller[0m: total_free is 17597, other_free is -468, free_cma is 1076
[32m[347381.004130] [33mlowmemorykiller[0m: other_file is 37257, active_file is 18116, inactive_file is 19256
[32m[347381.004130] [0m
[32m[347381.004130] [0m
[32m[347381.008788] [33mlowmemorykiller[0m: Killing '.cyberagent.stf' (28686) (tgid 28686), adj 200,
[32m[347381.008788] [0mto free 27108kB on behalf of 'kswapd0' (888) because
[32m[347381.008788] [0mcache 149028kB is below limit 73728kB for oom_score_adj 200
[32m[347381.008788] [0mFree memory is -2244kB above reserved
[32m[347381.008795] [0mTotal_free = 70016kB, free_cma=4304kB, Totalreserve_pages = 34316kB, MAPPED = 120428kB
[32m[347381.008800] [33mlowmemorykiller[0m: total_free is 17504, other_free is -561, free_cma is 1076
[32m[347381.008804] [33mlowmemorykiller[0m: other_file is 37257, active_file is 18116, inactive_file is 19256
[32m[347381.008804] [0m
[32m[347381.008804] [0m
[32m[347381.117631] [0mProfile Saver-28581:mtal_release
[32m[347381.117639] [0mx_os_drv_send_mod_uevent env=COMMAND=4,28581
[32m[347381.153558] [31m==RTC== MHAL_RTC_GetCounter, 244, time:0x6329a3fd, val:0x23, cnt:1
```

# OOM killer

```
m7332_eu:/sys/module/lowmemorykiller/parameters $ ps -A | grep stf
shell      23805  2803 1430420  57896 SyS_epoll_wait    0 S stf.agent
u0_a45    23823  2468 1049916  40320 0                  0 S jp.co.cyberagent.stf
```

# OOM killer

```
m7332_eu:/sys/module/lowmemorykiller/parameters $ ps -A | grep stf
shell          23805   2803 1430420  57896 SyS_epoll_wait    0 S stf.agent
u0_a45         23823   2468 1049916  40320 0                  0 S jp.co.cyberagent.stf
```

```
m7332_eu:/ $ cat proc/23823/oom_adj
3
```

# OOM killer

```
m7332_eu:/sys/module/lowmemorykiller/parameters $ ps -A | grep stf
shell          23805   2803 1430420  57896 SyS_epoll_wait    0 S stf.agent
u0_a45         23823   2468 1049916  40320 0                 0 S jp.co.cyberagent.stf
```

```
m7332_eu:/ $ cat proc/23823/oom_adj
3
```

```
m7332_eu:/ $ echo -17 > proc/23823/oom_adj
/system/bin/sh: can't create proc/23823/oom_adj: Permission denied
```

# OOM killer

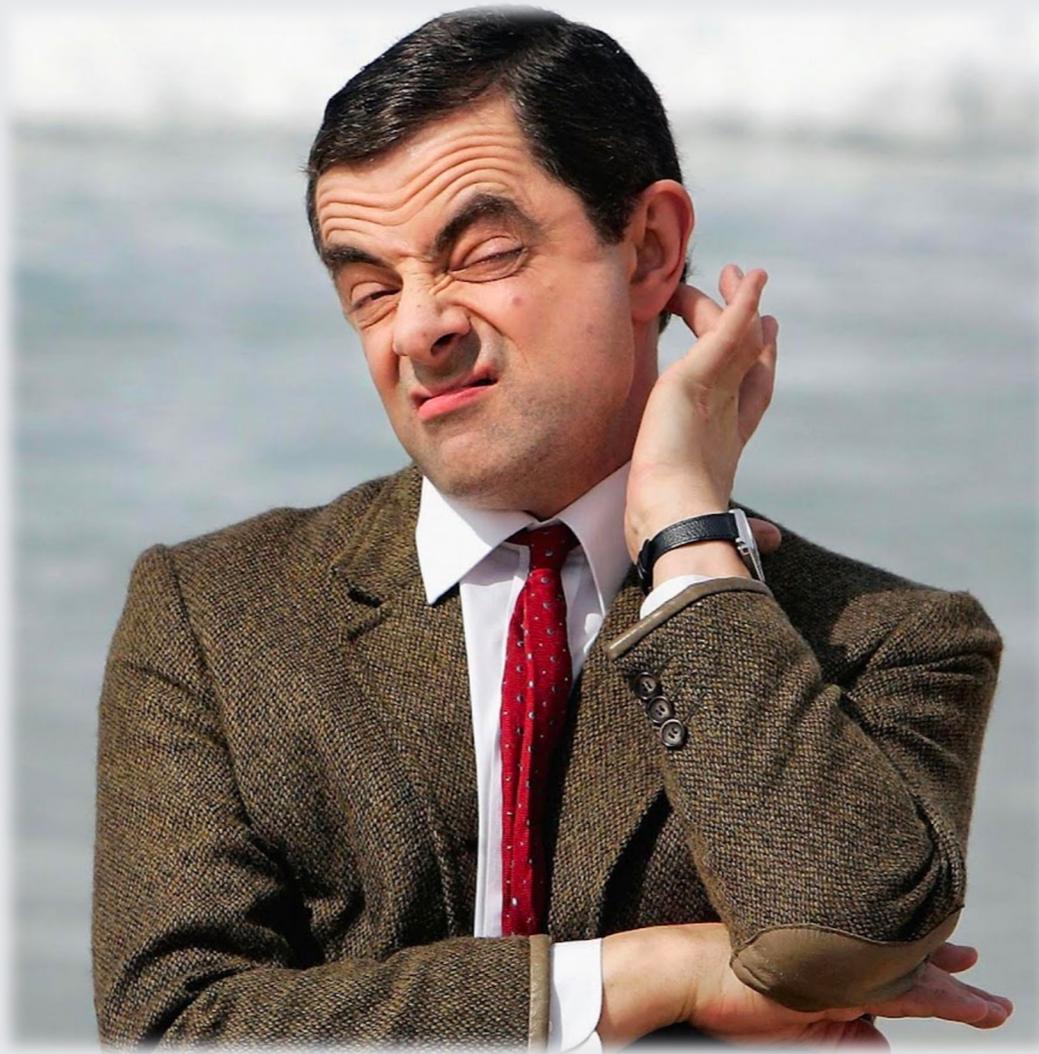
```
m7332_eu:/sys/module/lowmemorykiller/parameters $ ps -A | grep stf
shell      23805  2803 1430420  57896 SyS_epoll_wait    0 S stf.agent
u0_a45     23823  2468 1049916  40320 0                  0 S jp.co.cyberagent.stf
```

```
m7332_eu:/ $ cat proc/23823/oom_adj
3
```

```
m7332_eu:/ $ echo -17 > proc/23823/oom_adj
/system/bin/sh: can't create proc/23823/oom_adj: Permission denied
```

```
akela@akela-Latitude-5401:~$ adb root
adb cannot run as root in production builds
```

Пора...

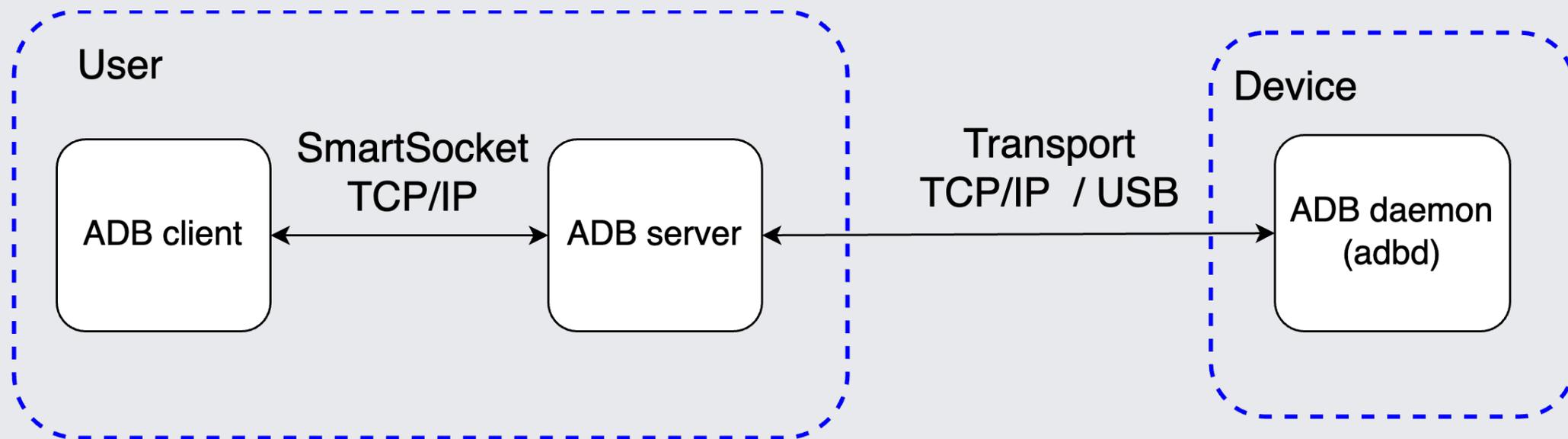


2023. Пишем своё решение

Идея

Ферма не должна ничего загружать  
и запускать на устройстве

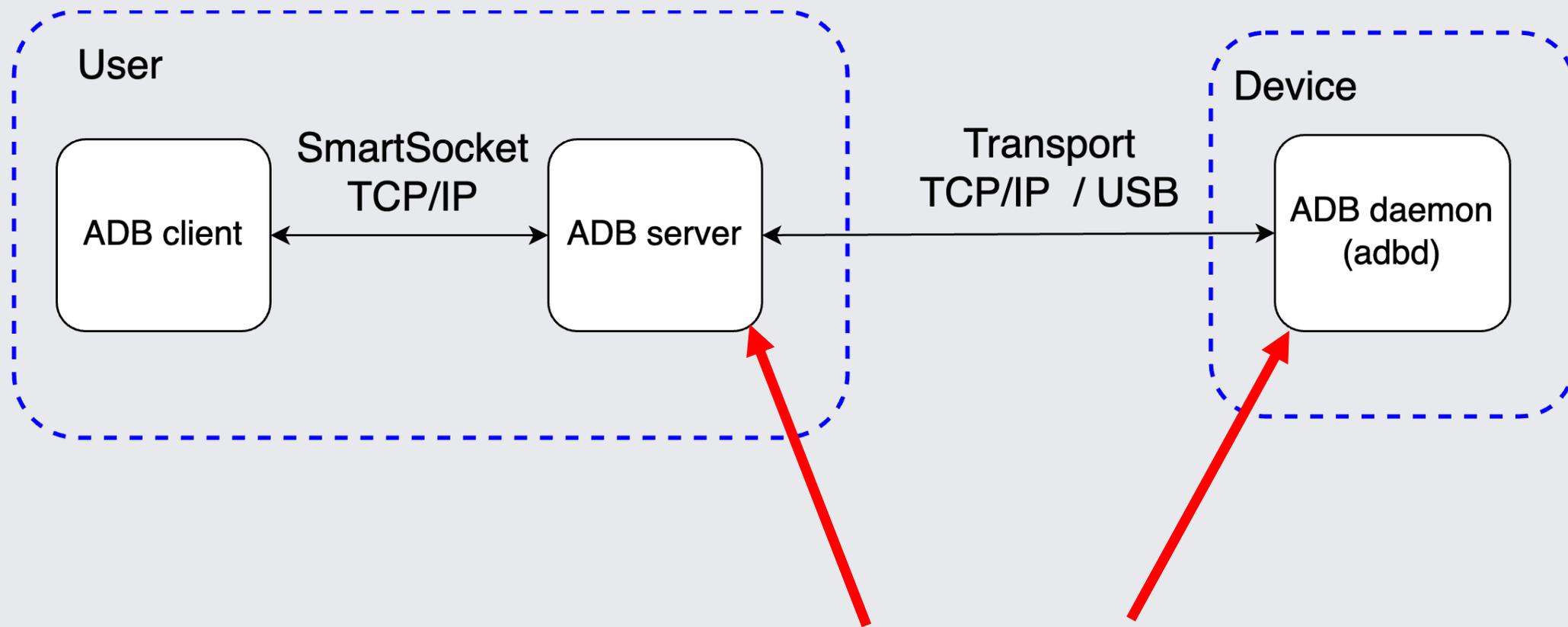
# Как работает ADB



## Как работает ADB

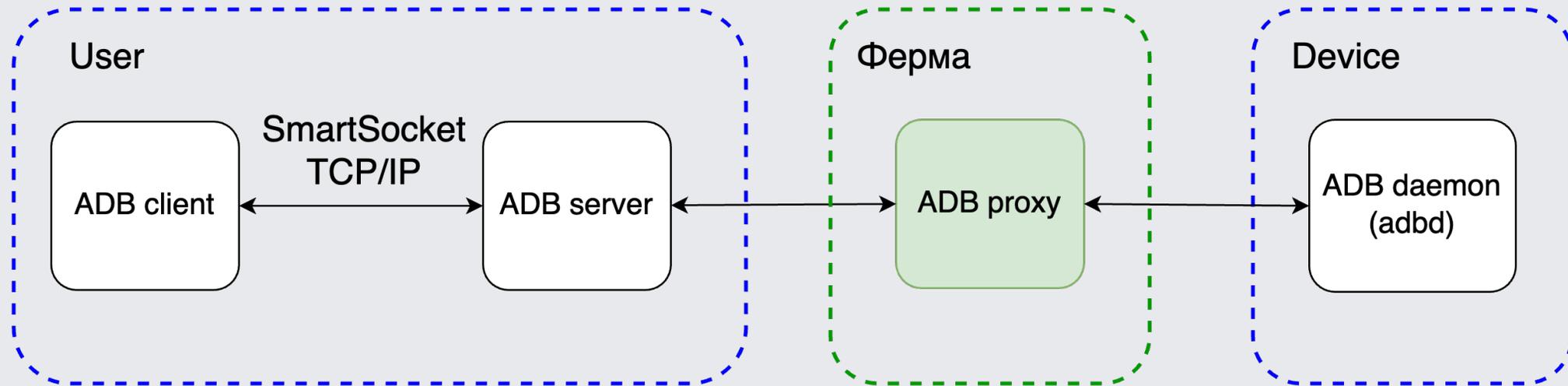


# Как работает ADB

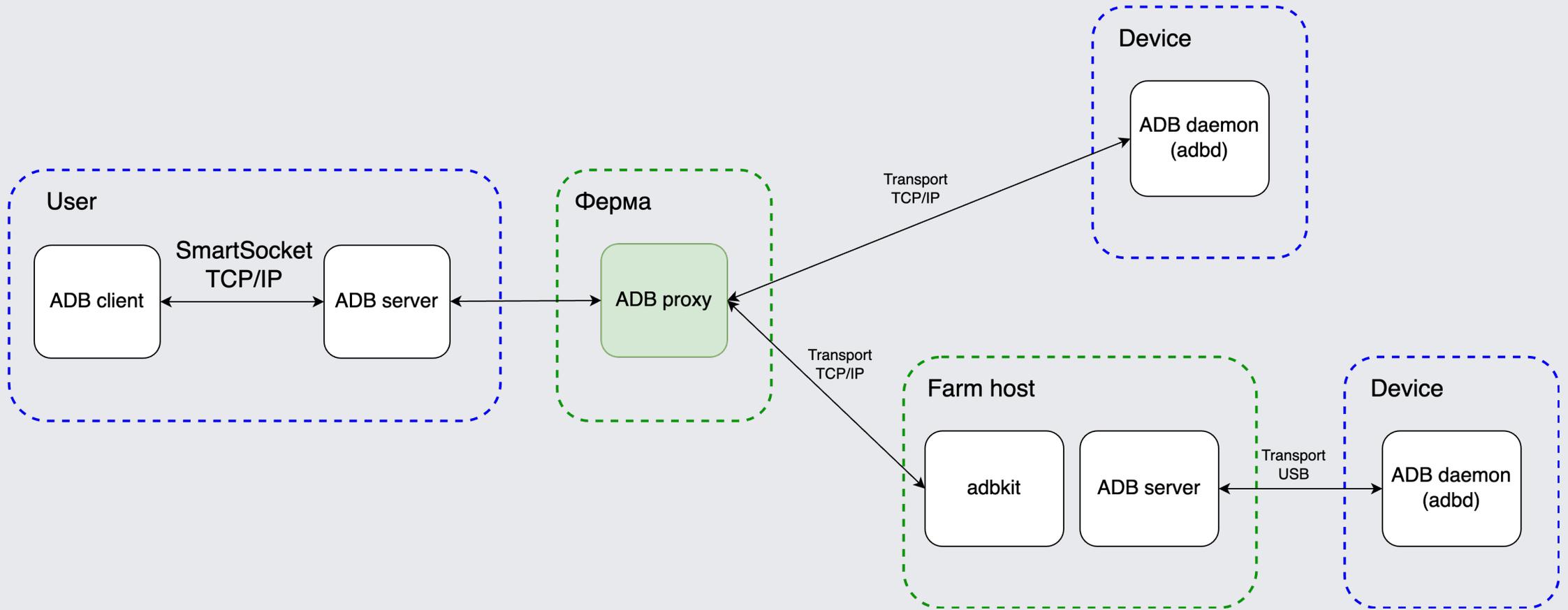


Ферма не должна вмешиваться в обмен между ADB-сервером и ADB-демоном на девайсе, а просто перекладывать трафик

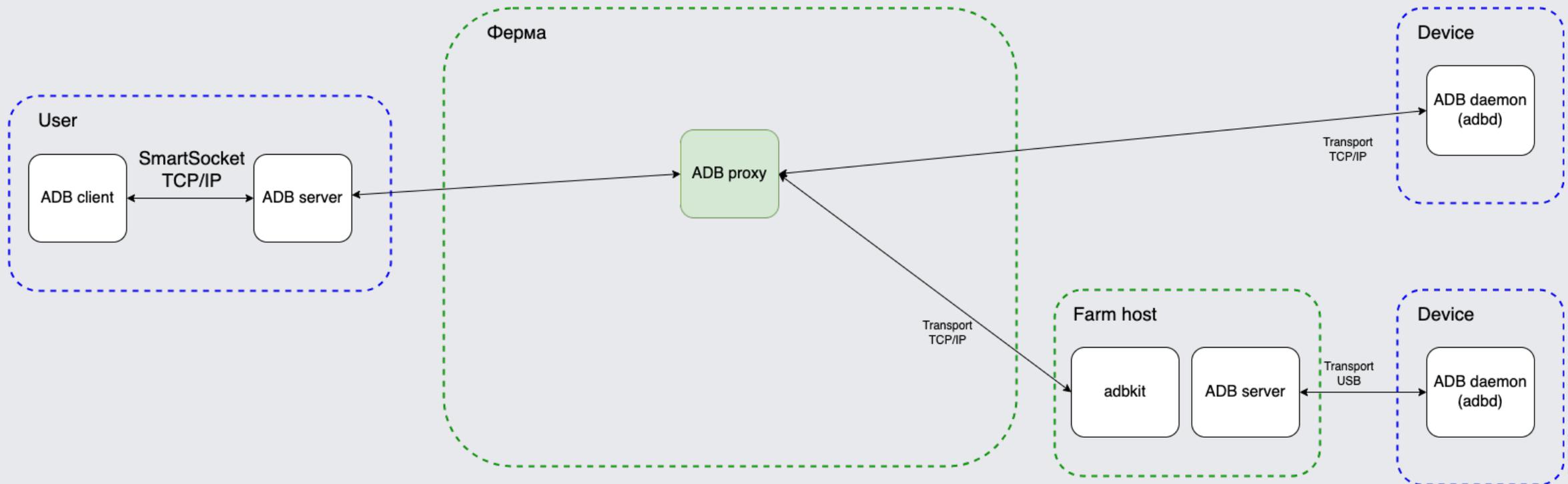
# Концепция



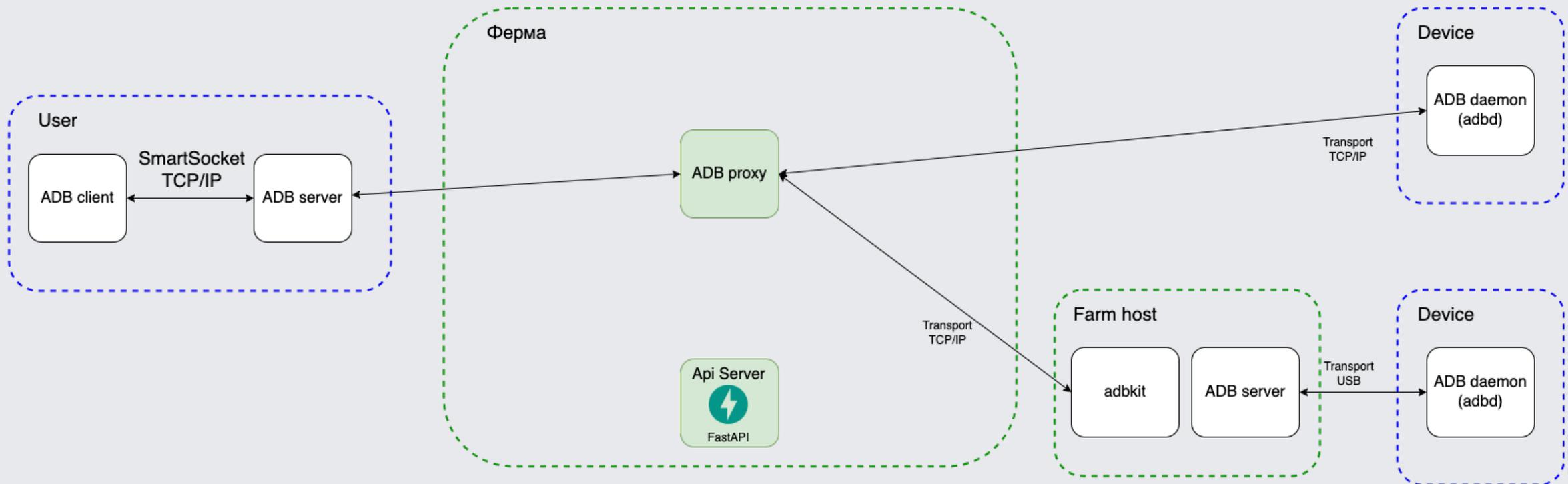
# Первая реализация



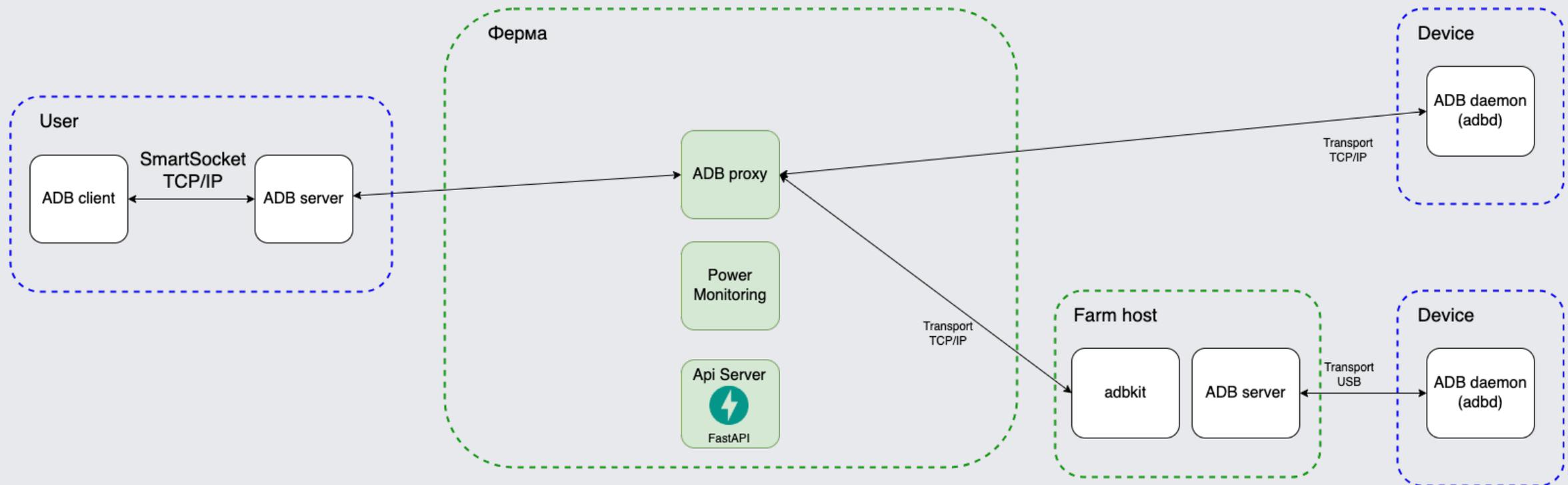
# Первая реализация



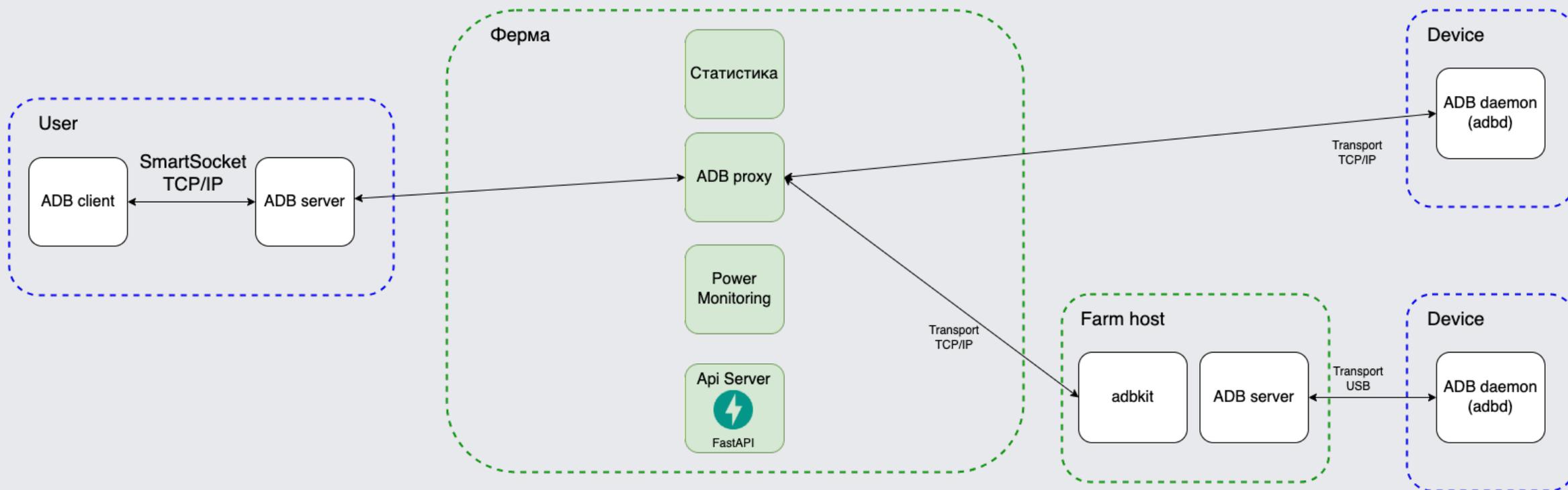
# Первая реализация



# Первая реализация



# Первая реализация



## Переезд из OpenSTF



К нам едет Умный дом (УД)

— Салют,  
умный дом!



## Опять проблемы

- У девайсов УД нет ADB;

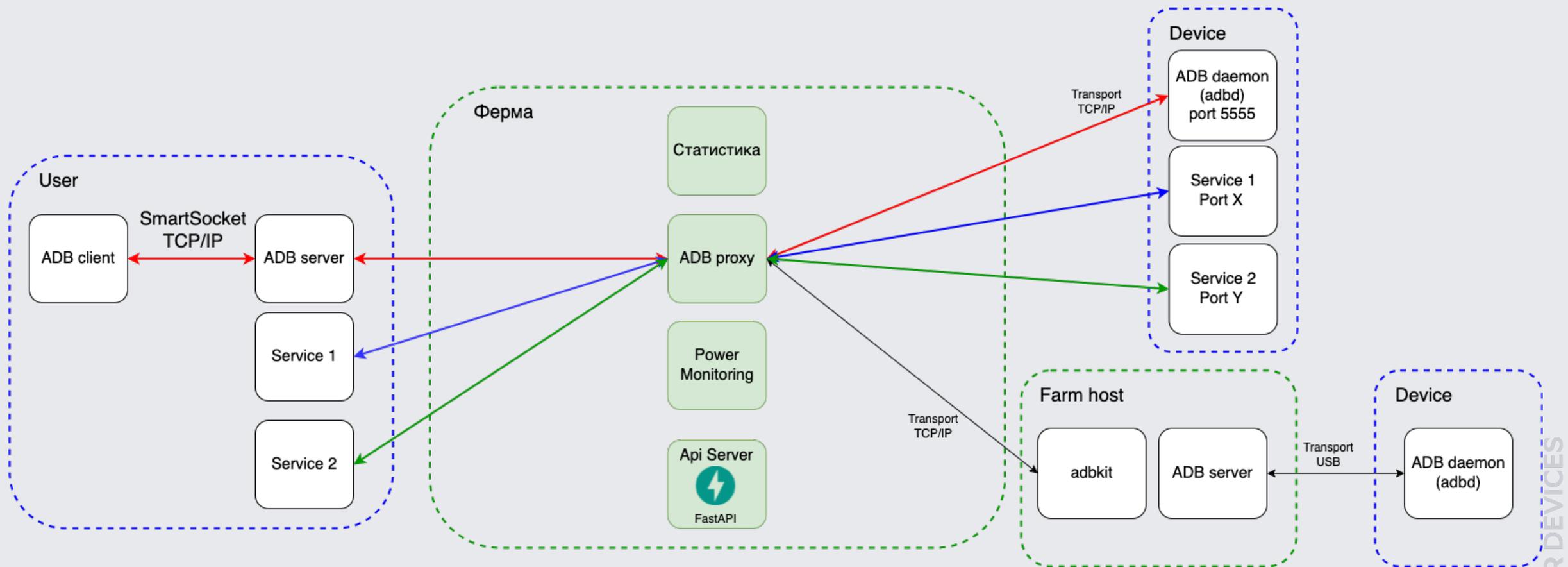
## Опять проблемы

- У девайсов УД нет ADB;
- Для работы с оборудованием нужно много портов.

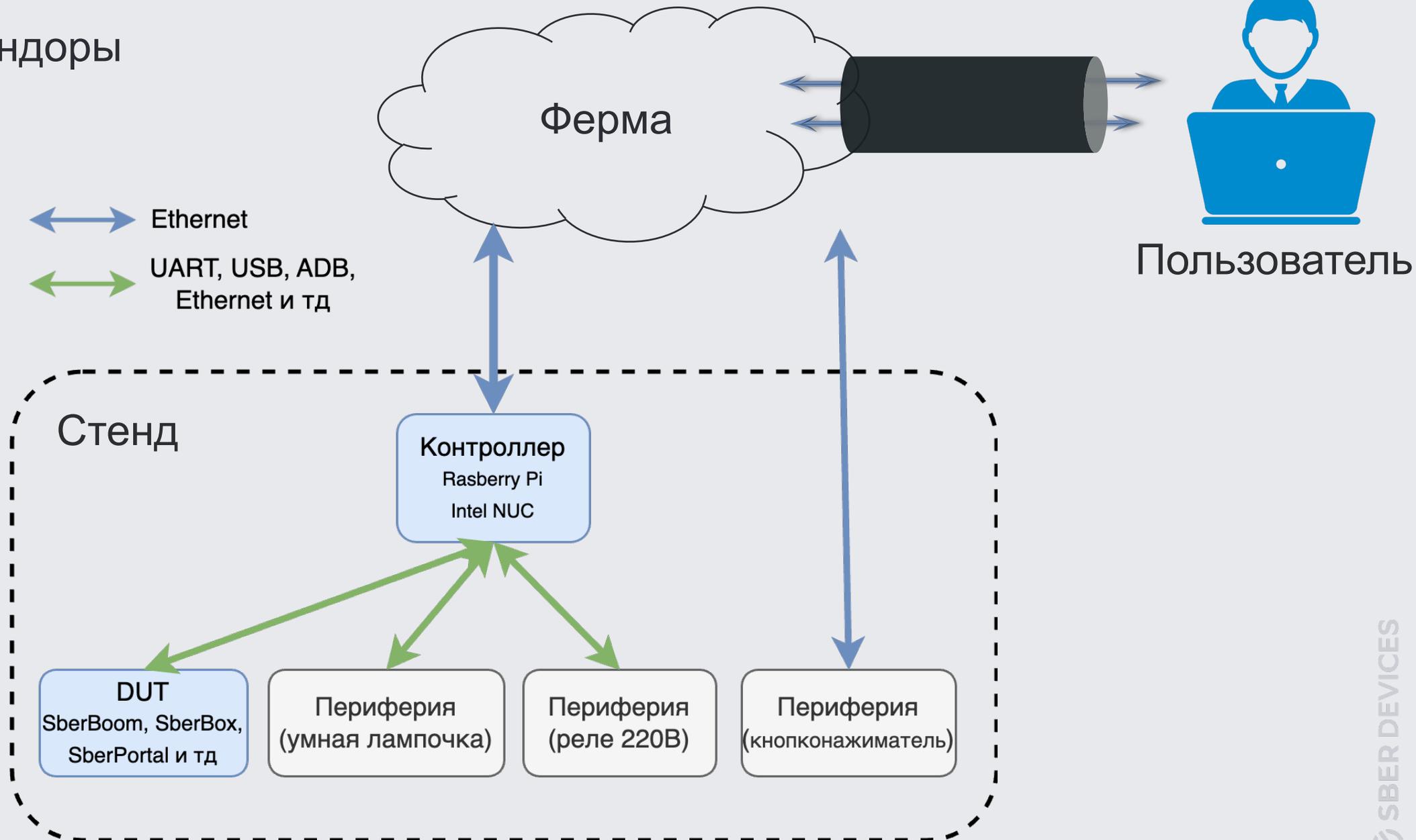
Уточняем идею

Ферма должна просто давать доступ до портов, которые слушают сервисы на девайсе

# Вторая реализация



# Ящик Пандоры



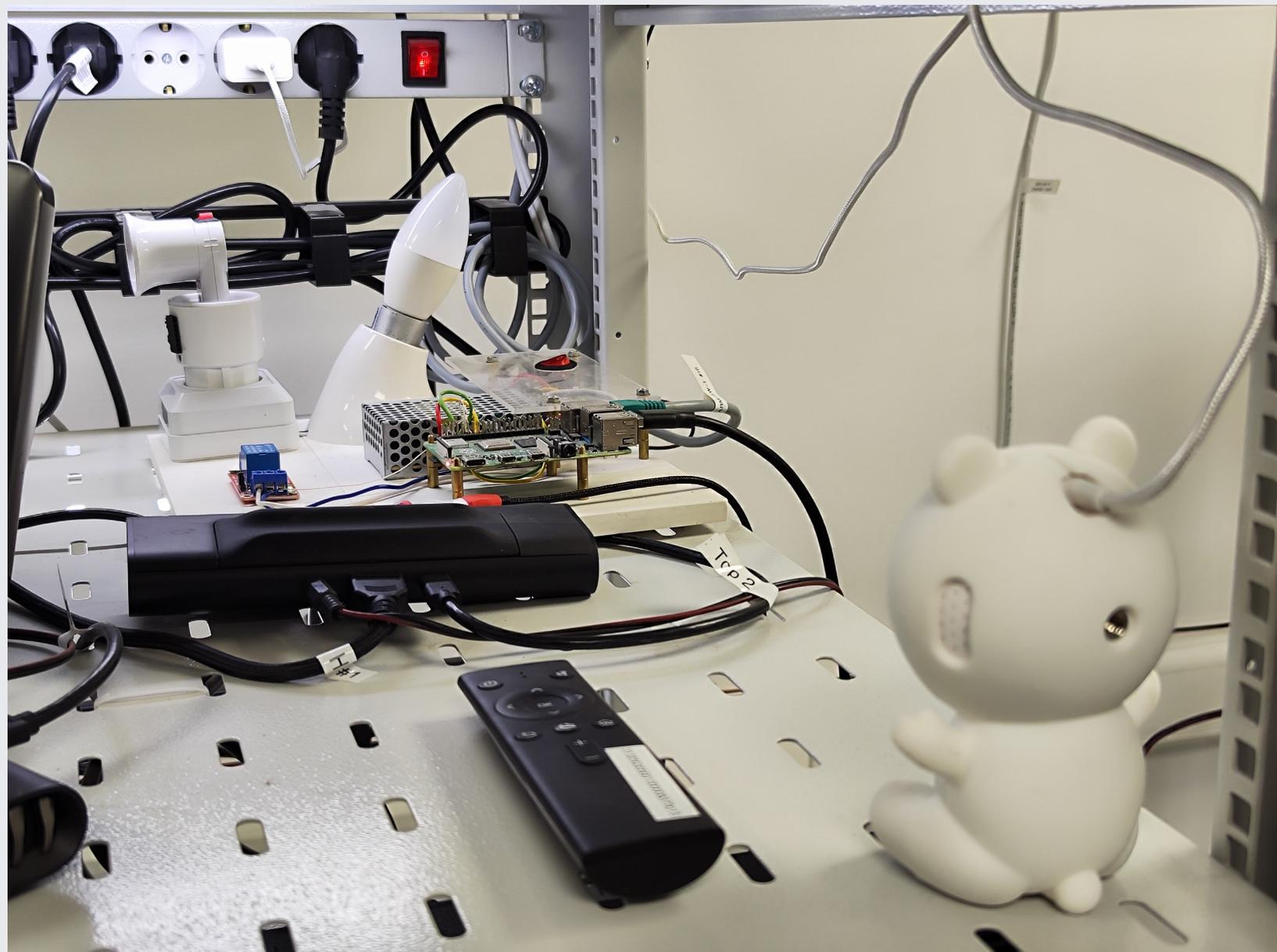
## Стенды умного дома

В результате доработок на Ферму смогли захватить стенды с **WirenBoard**



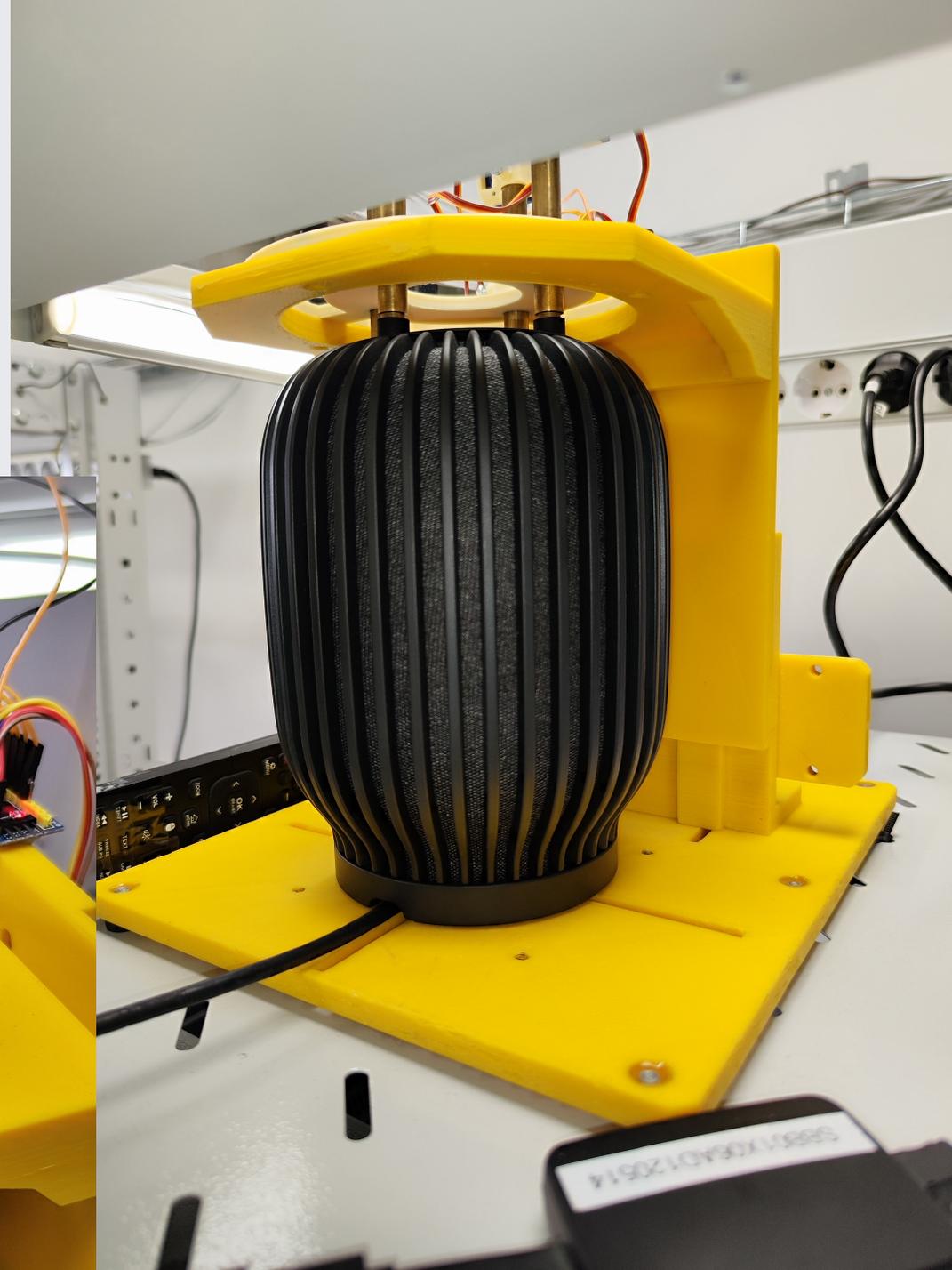
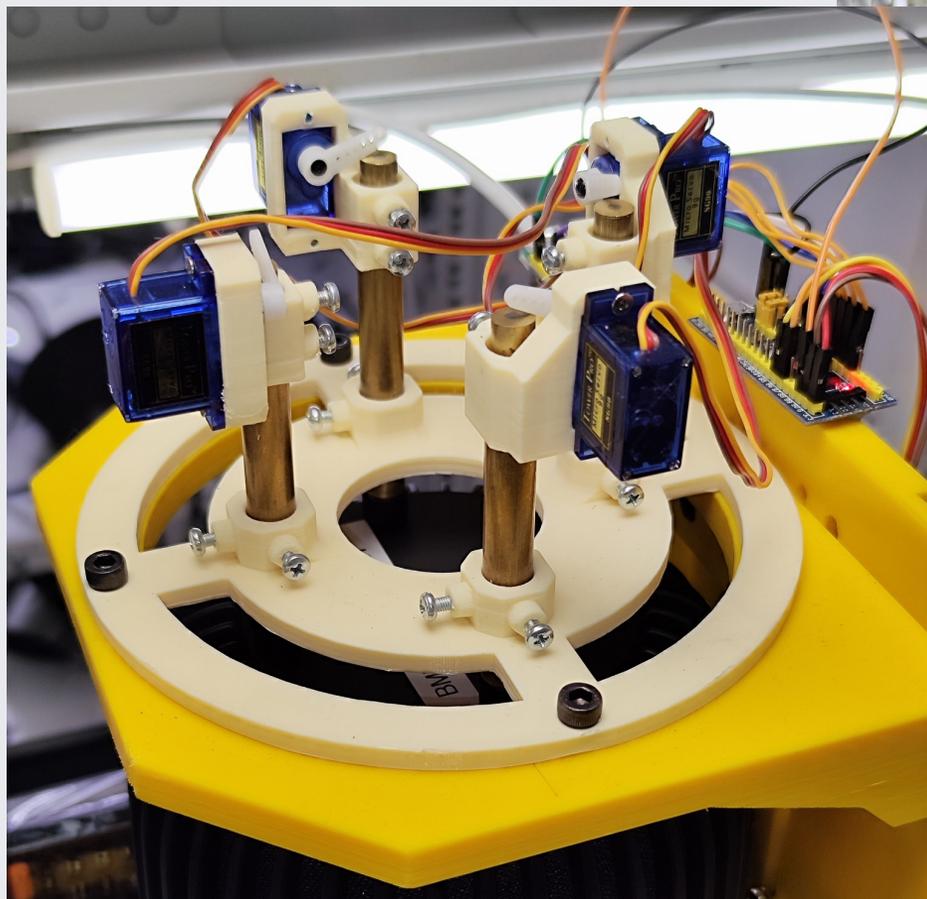
## Стенды умного дома

Или стенд для проверки  
пейринга лампочек  
к SberBox Top

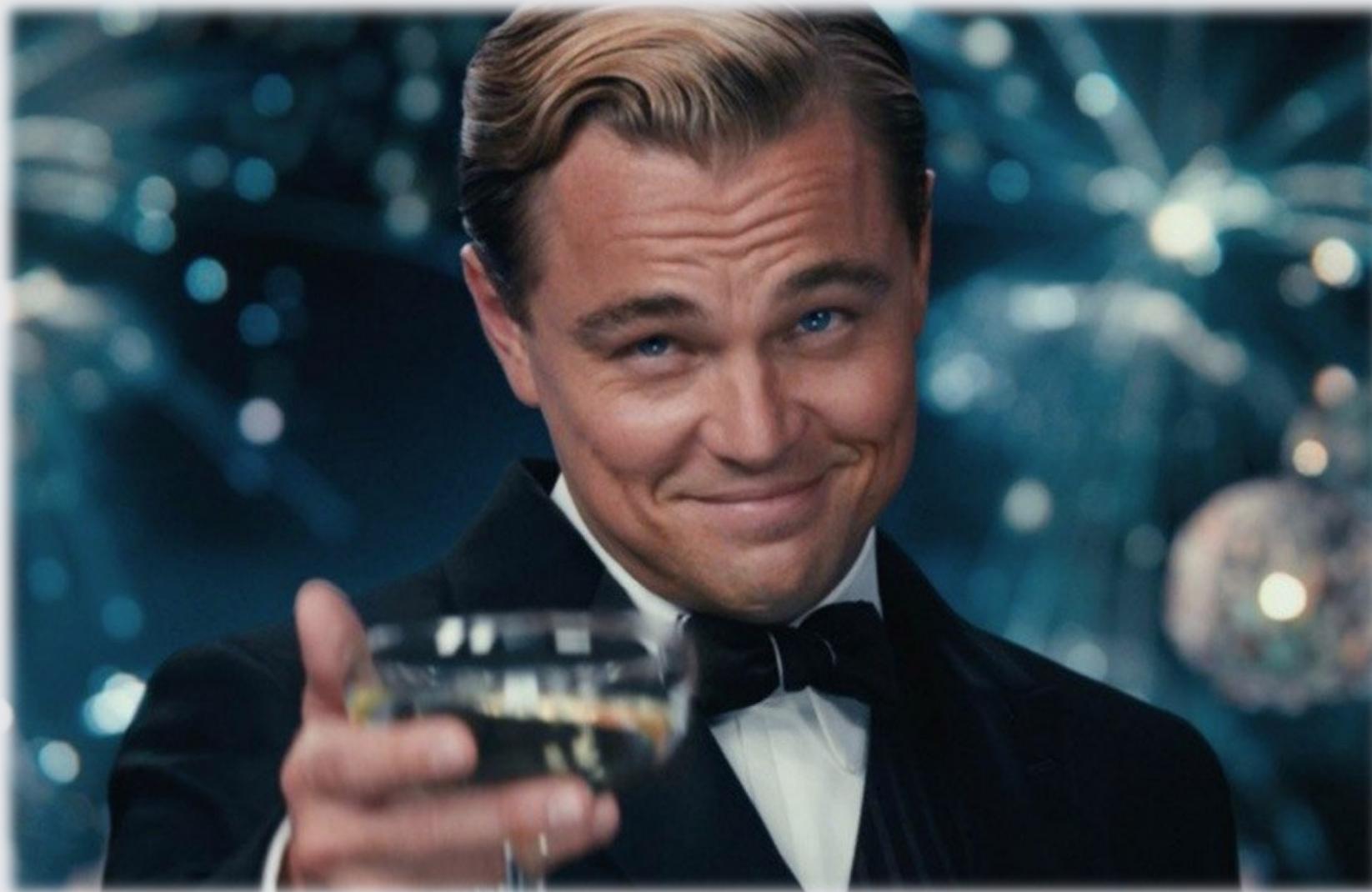


Еще пример станда

Или станд, где пользователи  
могут удаленно нажимать  
на кнопки



# Промышленная эксплуатация



# 2024. Ферма сейчас

Ферма сейчас

90+ девайсов

10 стендов

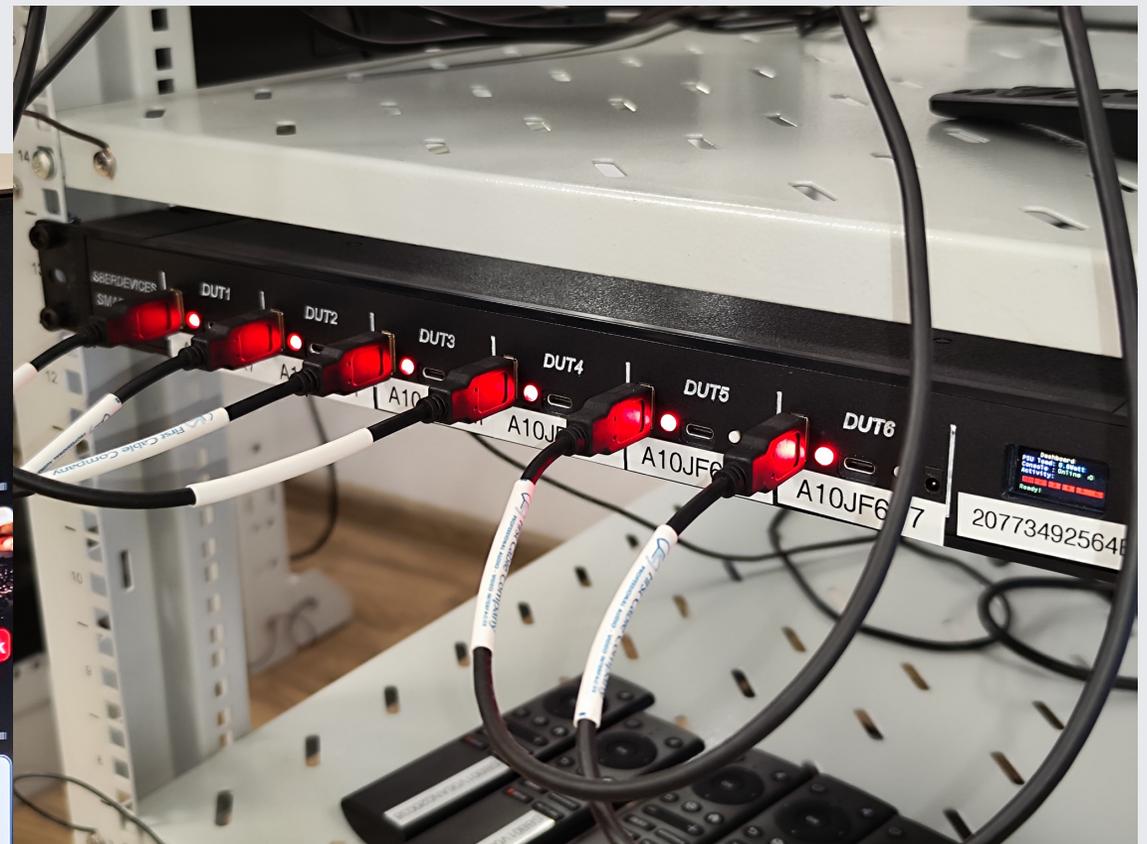
~2000  
использований  
девайсов\*

~500ч работы с  
девайсами\*

\*В неделю



# Ферма сейчас



# Как подключиться и начать работать

## Ферма

20240419.112623+4c927765

OAS 3.1

/openapi.json

- Ссылка на документацию: [Проект Ферма](#);
- По всем вопросам обращайтесь в [Mattermost/~farm](#).

Authorize 

### user основные эндпоинты

[подробное описание](#) 

GET /v1.0/devices Get Devices  

GET /v1.0/devices/{device\_id} Get Device Id  

POST /v1.0/devices/{device\_id}/lock Lock Device  

POST /v1.0/devices/{device\_id}/unlock Unlock Device  

### CI альтернативные версии эндпоинтов, для использования в рамках CI (и прочей автоматизации)

[пример использования](#) 

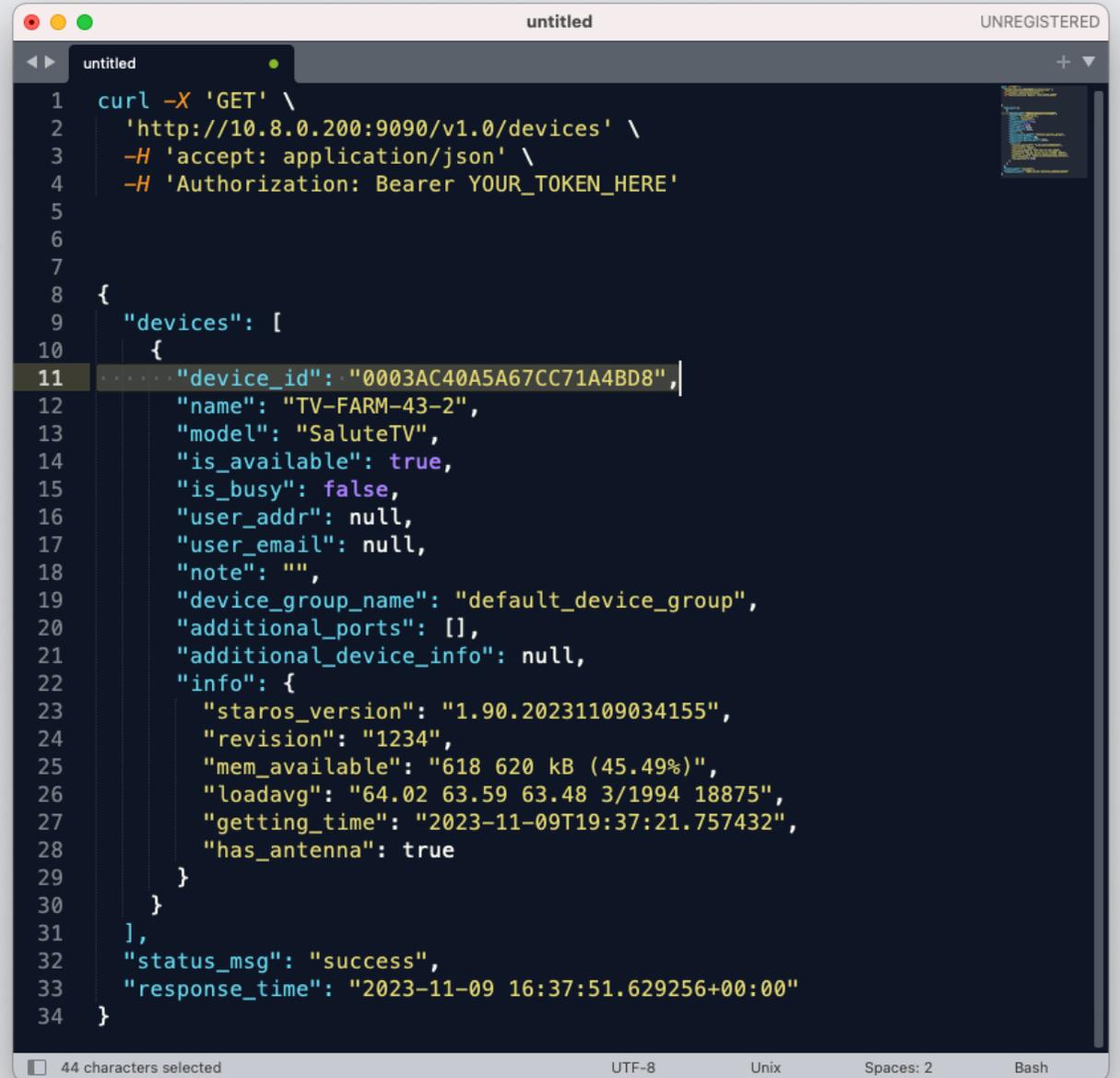
POST /v1.0/ci/devices/{device\_id}/lock Lock Device For Ci  

POST /v1.0/ci/devices/{device\_id}/unlock Unlock Device For Ci  

# Как подключиться и начать работать

## 1

Получаем список всех девайсов, доступны фильтры по разным параметрам, например, модель или наличие антенны



```
untitled UNREGISTERED
1 curl -X 'GET' \
2   'http://10.8.0.200:9090/v1.0/devices' \
3   -H 'accept: application/json' \
4   -H 'Authorization: Bearer YOUR_TOKEN_HERE'
5
6
7
8 {
9   "devices": [
10    {
11     "device_id": "0003AC40A5A67CC71A4BD8",
12     "name": "TV-FARM-43-2",
13     "model": "SaluteTV",
14     "is_available": true,
15     "is_busy": false,
16     "user_addr": null,
17     "user_email": null,
18     "note": "",
19     "device_group_name": "default_device_group",
20     "additional_ports": [],
21     "additional_device_info": null,
22     "info": {
23       "staros_version": "1.90.20231109034155",
24       "revision": "1234",
25       "mem_available": "618 620 kB (45.49%)",
26       "loadavg": "64.02 63.59 63.48 3/1994 18875",
27       "getting_time": "2023-11-09T19:37:21.757432",
28       "has_antenna": true
29     }
30   }
31 ],
32 "status_msg": "success",
33 "response_time": "2023-11-09 16:37:51.629256+00:00"
34 }
```

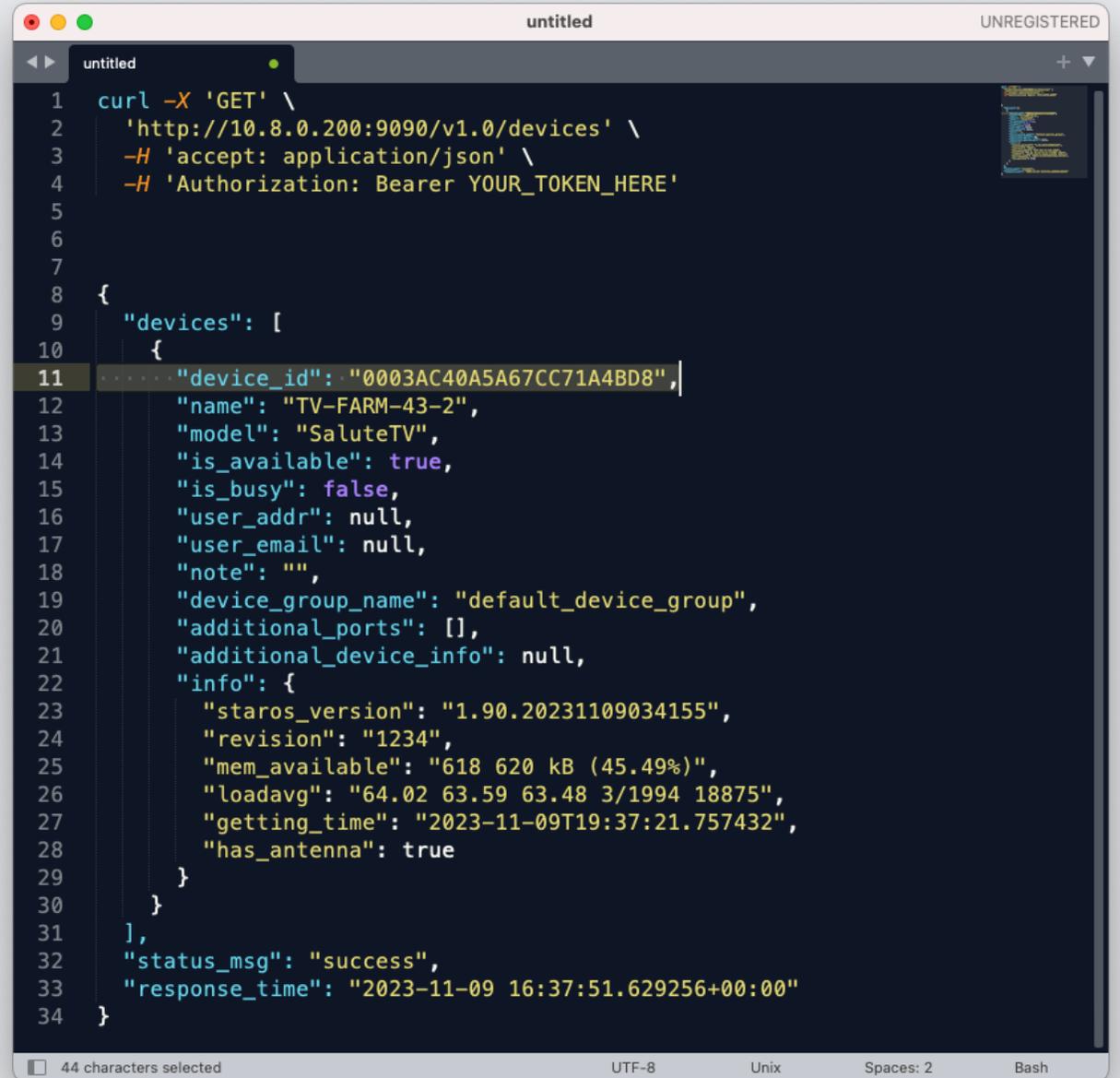
# Как подключиться и начать работать

1

Получаем список всех девайсов, доступны фильтры по разным параметрам, например, модель или наличие антенны

2

Получаем уникальный идентификатор нужного устройства, в случае ADB устройств – device id

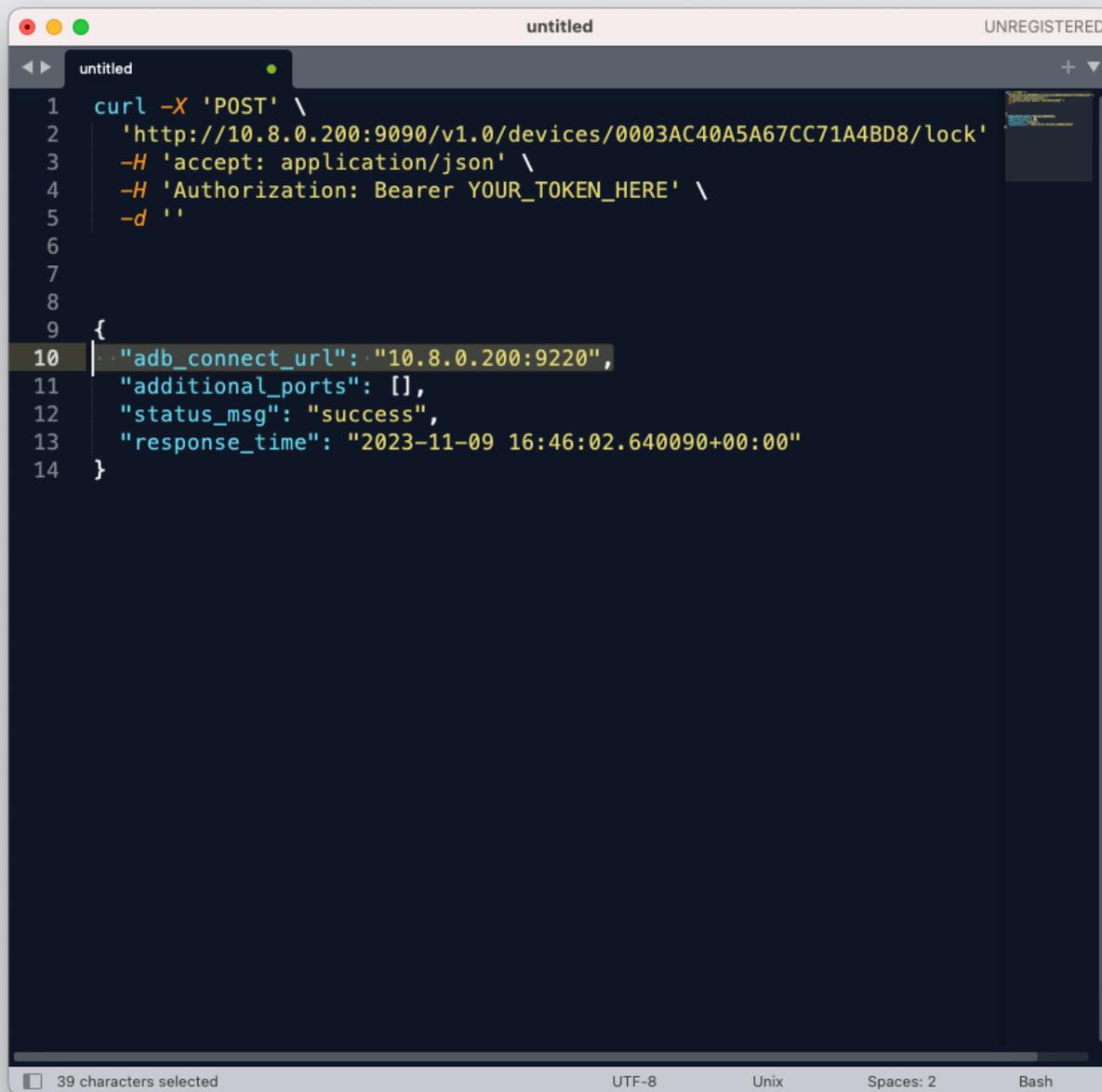


```
untitled UNREGISTERED
1 curl -X 'GET' \
2   'http://10.8.0.200:9090/v1.0/devices' \
3   -H 'accept: application/json' \
4   -H 'Authorization: Bearer YOUR_TOKEN_HERE'
5
6
7
8 {
9   "devices": [
10    {
11     "device_id": "0003AC40A5A67CC71A4BD8",
12     "name": "TV-FARM-43-2",
13     "model": "SaluteTV",
14     "is_available": true,
15     "is_busy": false,
16     "user_addr": null,
17     "user_email": null,
18     "note": "",
19     "device_group_name": "default_device_group",
20     "additional_ports": [],
21     "additional_device_info": null,
22     "info": {
23       "staros_version": "1.90.20231109034155",
24       "revision": "1234",
25       "mem_available": "618 620 kB (45.49%)",
26       "loadavg": "64.02 63.59 63.48 3/1994 18875",
27       "getting_time": "2023-11-09T19:37:21.757432",
28       "has_antenna": true
29     }
30   }
31 ],
32 "status_msg": "success",
33 "response_time": "2023-11-09 16:37:51.629256+00:00"
34 }
```

# Как подключиться и начать работать

## 3

Занимаем нужное устройство



```
1 curl -X 'POST' \  
2   'http://10.8.0.200:9090/v1.0/devices/0003AC40A5A67CC71A4BD8/lock' \  
3   -H 'accept: application/json' \  
4   -H 'Authorization: Bearer YOUR_TOKEN_HERE' \  
5   -d '' \  
6 \  
7 \  
8 \  
9 { \  
10  "adb_connect_url": "10.8.0.200:9220", \  
11  "additional_ports": [], \  
12  "status_msg": "success", \  
13  "response_time": "2023-11-09 16:46:02.640090+00:00" \  
14 }
```

39 characters selected UTF-8 Unix Spaces: 2 Bash

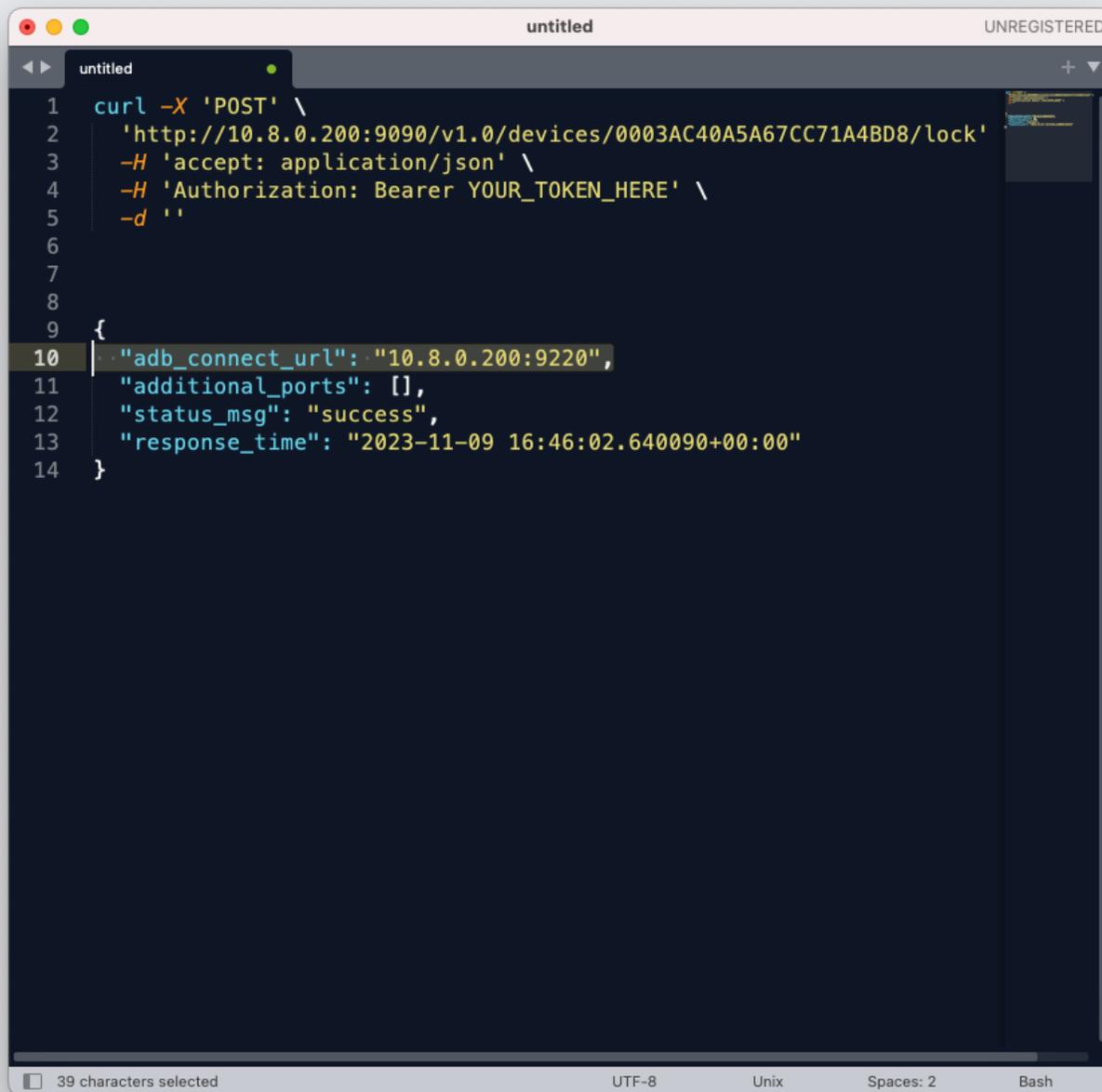
# Как подключиться и начать работать

## 3

Занимаем нужное устройство

## 4

Получаем url для подключения



```
1 curl -X 'POST' \  
2   'http://10.8.0.200:9090/v1.0/devices/0003AC40A5A67CC71A4BD8/lock' \  
3   -H 'accept: application/json' \  
4   -H 'Authorization: Bearer YOUR_TOKEN_HERE' \  
5   -d '' \  
6 \  
7 \  
8 \  
9 { \  
10  "adb_connect_url": "10.8.0.200:9220", \  
11  "additional_ports": [], \  
12  "status_msg": "success", \  
13  "response_time": "2023-11-09 16:46:02.640090+00:00" \  
14 }
```

# Как подключиться и начать работать

## 5

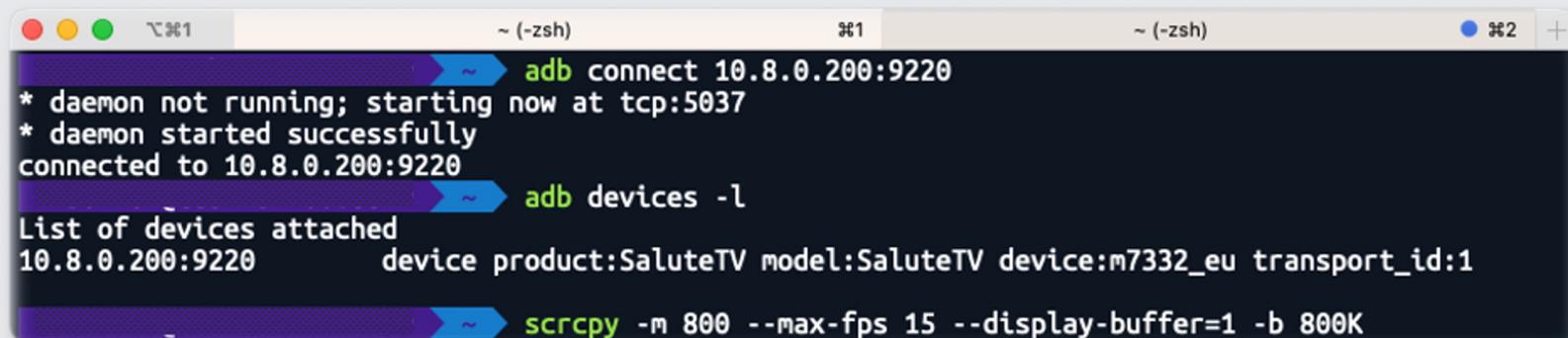
Подключаемся к устройству, например, через ADB

```
adb connect 10.8.0.200:9220
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 10.8.0.200:9220
adb devices -l
List of devices attached
10.8.0.200:9220    device product:SaluteTV model:SaluteTV device:m7332_eu transport_id:1
scrcpy -m 800 --max-fps 15 --display-buffer=1 -b 800K
```

## Как подключиться и начать работать

5

Подключаемся к устройству, например, через ADB

A terminal window with a dark background and light text. The window title bar shows three colored circles (red, yellow, green) on the left, and the text '~ (-zsh)' in the center. The terminal content shows the following commands and output:

```
adb connect 10.8.0.200:9220
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 10.8.0.200:9220
adb devices -l
List of devices attached
10.8.0.200:9220      device product:SaluteTV model:SaluteTV device:m7332_eu transport_id:1
adb -
scrcpy -m 800 --max-fps 15 --display-buffer=1 -b 800K
```

6

Проверяем, что устройство успешно подключилось

# Как подключиться и начать работать

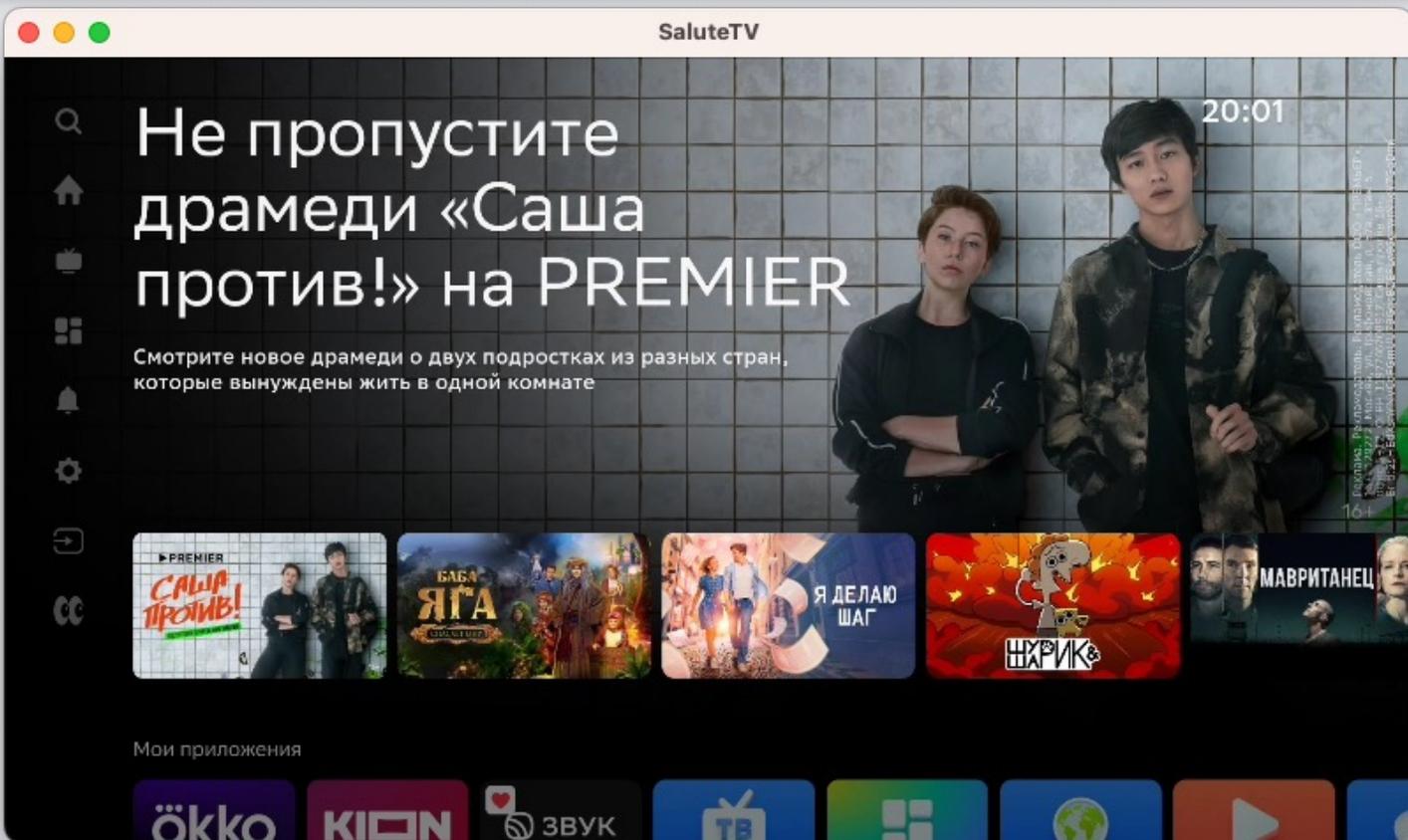
5

Подключаемся к устройству, например, через ADB

```
adb connect 10.8.0.200:9220
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 10.8.0.200:9220
adb devices -l
List of devices attached
10.8.0.200:9220    device product:SaluteTV model:SaluteTV device:m7332_eu transport_id:1
scrcpy -m 800 --max-fps 15 --display-buffer=1 -b 800K
```

6

Проверяем, что устройство успешно подключилось



7

У нас есть доступ к файловой системе устройства, можем, например, стримить видео или загружать свои .apk файлы

## Рецепт «Как собрать собственное решение»

— Команда;

## Рецепт «Как собрать собственное решение»

- Команда;
- Железо;

## Рецепт «Как собрать собственное решение»

- Команда;
- Железо;
- Инструменты разработки;

## Рецепт «Как собрать собственное решение»

- Команда;
- Железо;
- Инструменты разработки;
- Помещение;

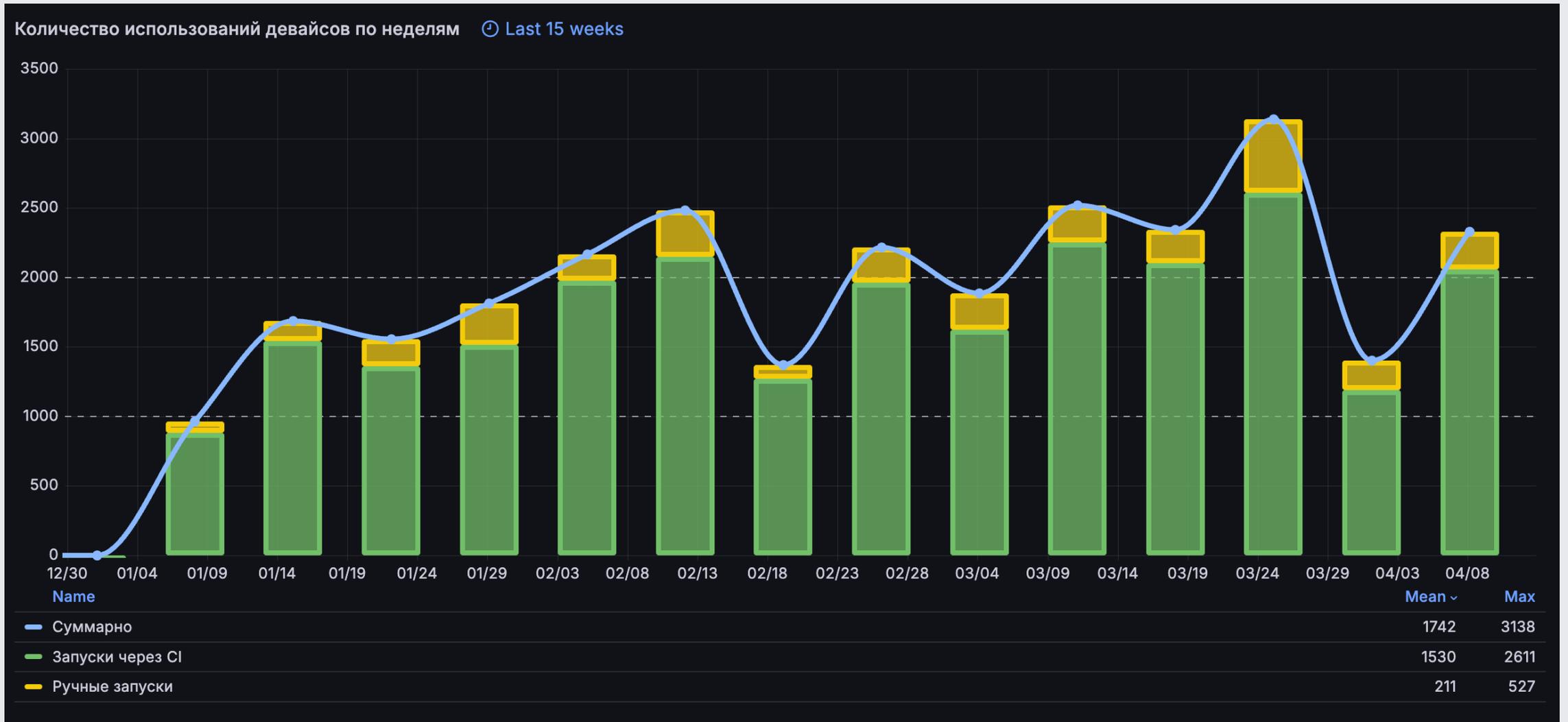
## Рецепт «Как собрать собственное решение»

- Команда;
- Железо;
- Инструменты разработки;
- Помещение;
- Сеть.

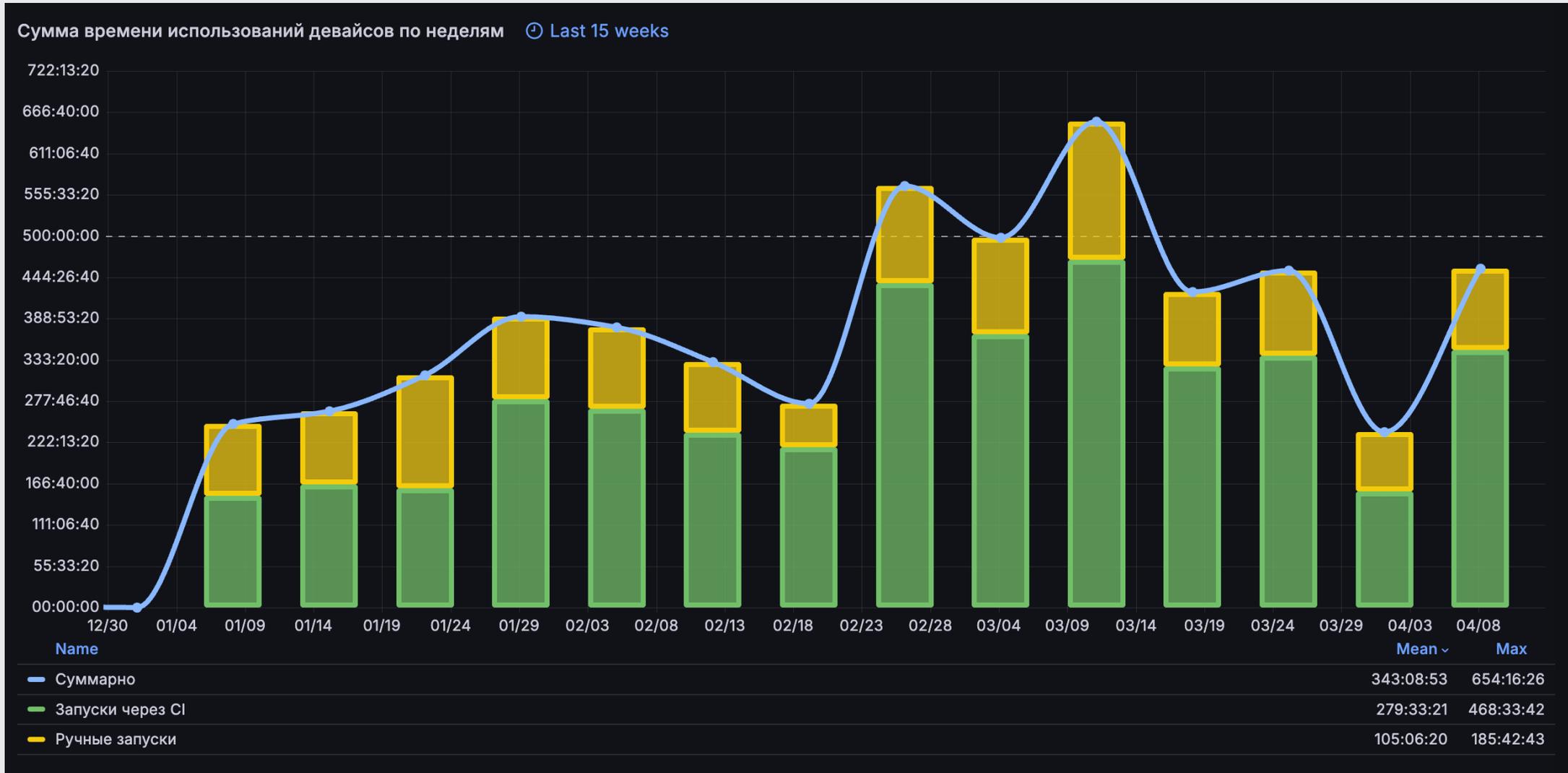
# Красивые графики



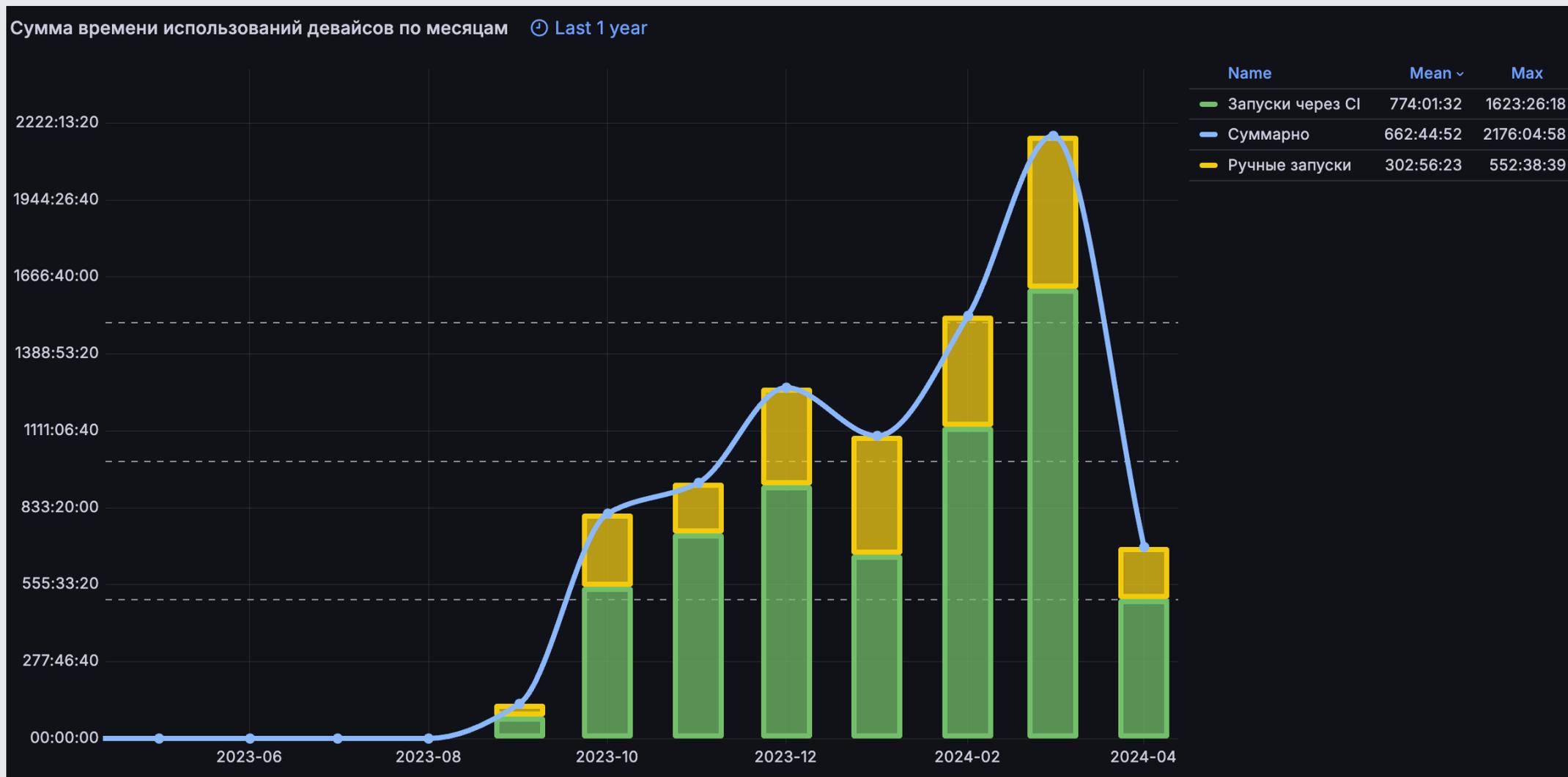
# Красивые графики



# Красивые графики



# Красивые графики



# Красивые графики



## Заключение

- Благодаря Ферме QA командам удалось сократить время тестирования (регрес в 6 раз, смоук в 7 раз);

## Заключение

- Благодаря Ферме QA командам удалось сократить время тестирования (регрес в 6 раз, смоук в 7 раз);
- Ушли от OpenSTF в пользу своего решения, теперь на Ферму можно поставить практически любой девайс или стенд;

## Заключение

- Благодаря Ферме QA командам удалось сократить время тестирования (регрес в 6 раз, смоук в 7 раз);
- Ушли от OpenSTF в пользу своего решения, теперь на Ферму можно поставить практически любой девайс или стенд;
- Появился выделенный инженер, который обрабатывает заявки пользователей и эксплуатирует Ферму (это я 😊);

## Заключение

- Благодаря Ферме QA командам удалось сократить время тестирования (регрес в 6 раз, смоук в 7 раз);
- Ушли от OpenSTF в пользу своего решения, теперь на Ферму можно поставить практически любой девайс или стенд;
- Появился выделенный инженер, который обрабатывает заявки пользователей и эксплуатирует Ферму (это я 😊);
- Порешали кучу инженерных вопросов и открыли для себя еще больше НОВЫХ.

# Контакты



[t.me/KozhemyakinNikita](https://t.me/KozhemyakinNikita)



[Статья про Ферму на Хабре](#)



## P.S. Тот самый анекдот

Берется интервью у женщины, прожившей со своим мужем 50 лет без единой ссоры:

- Мадам, как Вам удалось прожить все это время в мире и гармонии?
- О, все просто! Когда мы обвенчались, мой муж посадил меня на повозку и повез на своё ранчо. Пока мы ехали, одна из лошадей споткнулась и он спокойно сказал: "Раз". Спустя 15 минут, всё в той же поездке эта же лошадь споткнулась еще раз и он сказал: "Два". И уже когда его ранчо было в прямой видимости эта же лошадь споткнулась в третий раз. Он сказал: "Три" ... и пристрелил ее. Я, конечно, начала кричать, закатила истерику, начала жалеть бедное животное...
- И что же?
- И он сказал "Раз".

