Яндекс 360

Оптимизации видеопотока H.264

в RTP/AVP

В кейсе рендеринга UI
для пригласительного меню
переговорных комнат Телемоста

Дмитрий Некрылов

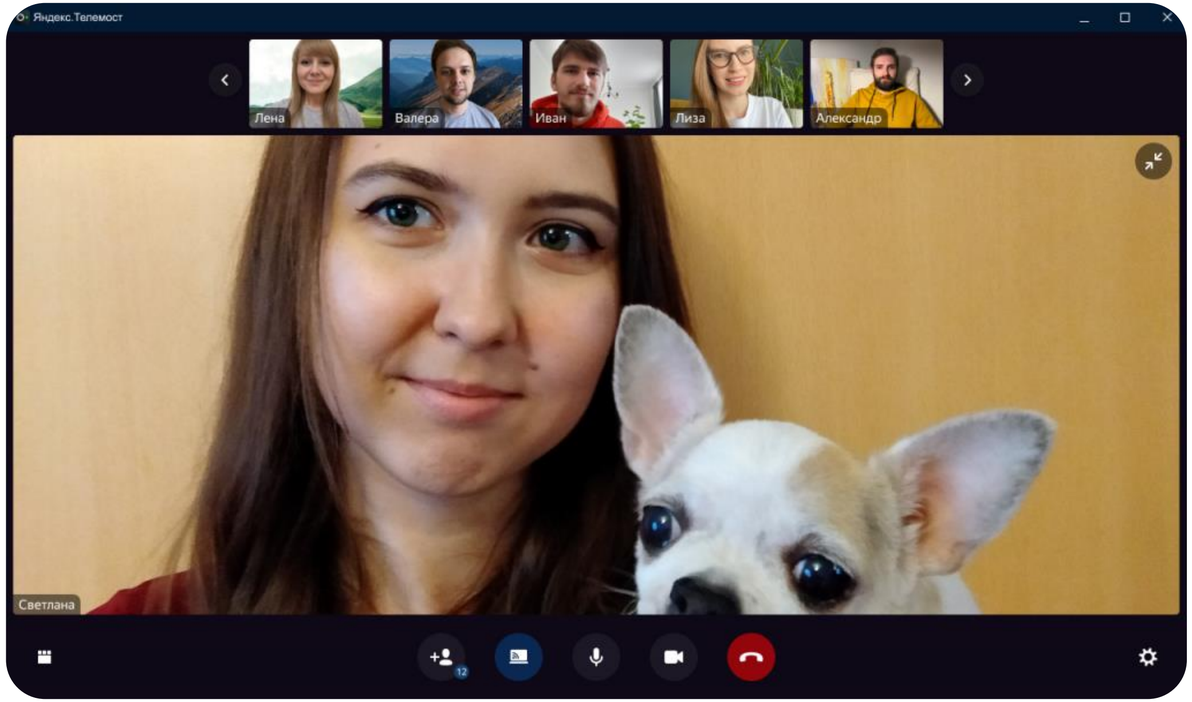
Разработчик, тимлид

Медиатехнологии Кинопоиска

16 лет экспертизы

- Order Capture & Management
- Application Performance
- Video Conferencing
- Video Streaming

Яндекс Телемост



Kadinsky 2.2

Чем полезен доклад

H.264

RTP

LiveJournal.com, black box, mmetalloprom.ru

«Путь» команды телемоста

1. Зачем рендерить UI в видеопоток
2. Делаем «в лоб». Просто. Дорого
3. Оптимизируем по-простому
4. Хачим закодированный H.264
5. Чистый Zero-Delta P-Frame

Немного о вас

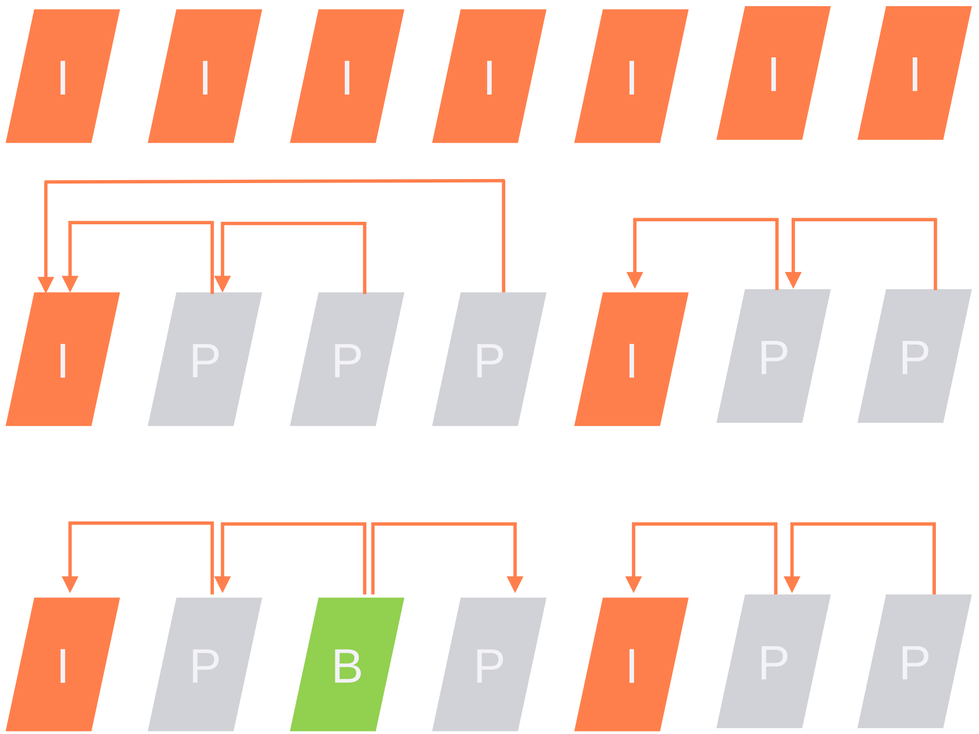
Кто знает, что такое I-, P- и B-Frames?

Кто знает, что такое SDP?

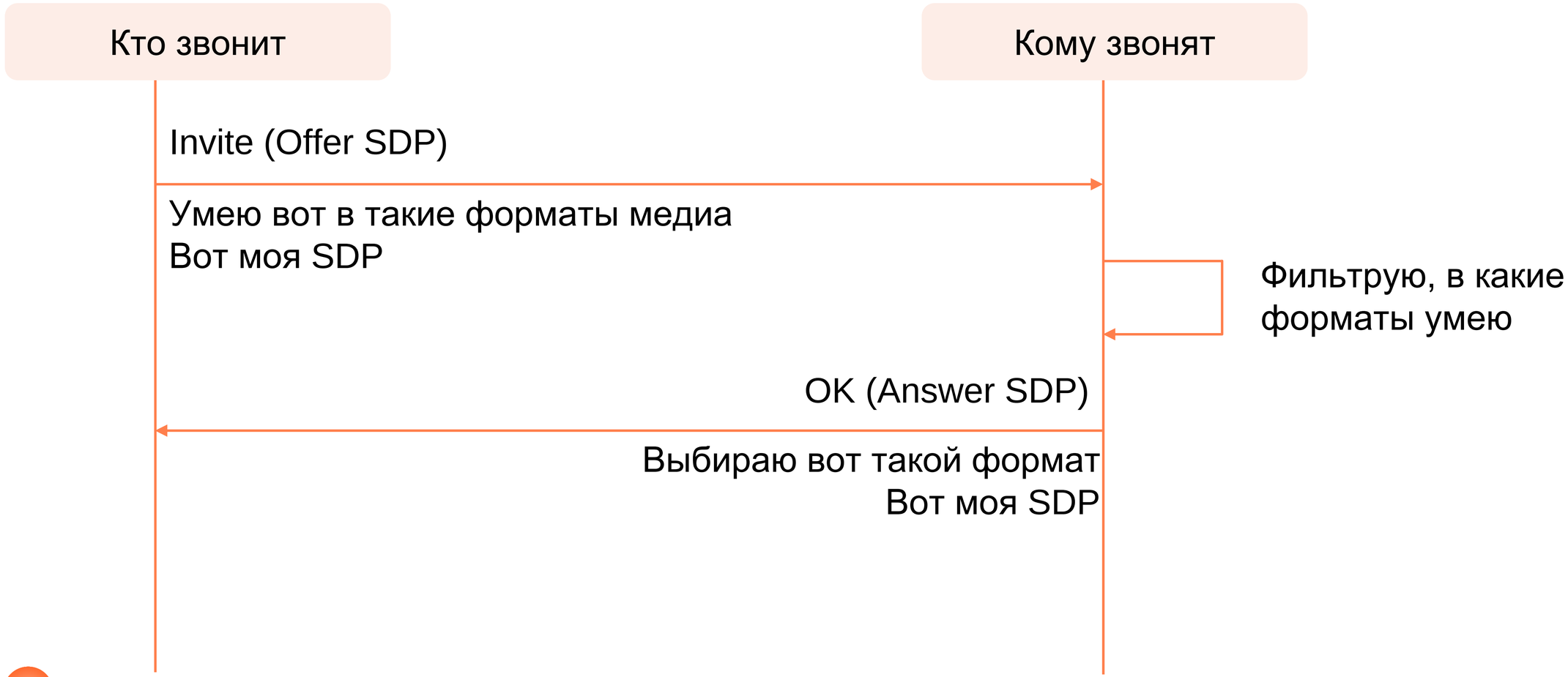
Кто знает, что такое Discrete Cosine Transform?

I-, P- и B-фреймы

- Intra Coded: I
- Inter Coded:
  - P — predictive
  - B — BI-predictive

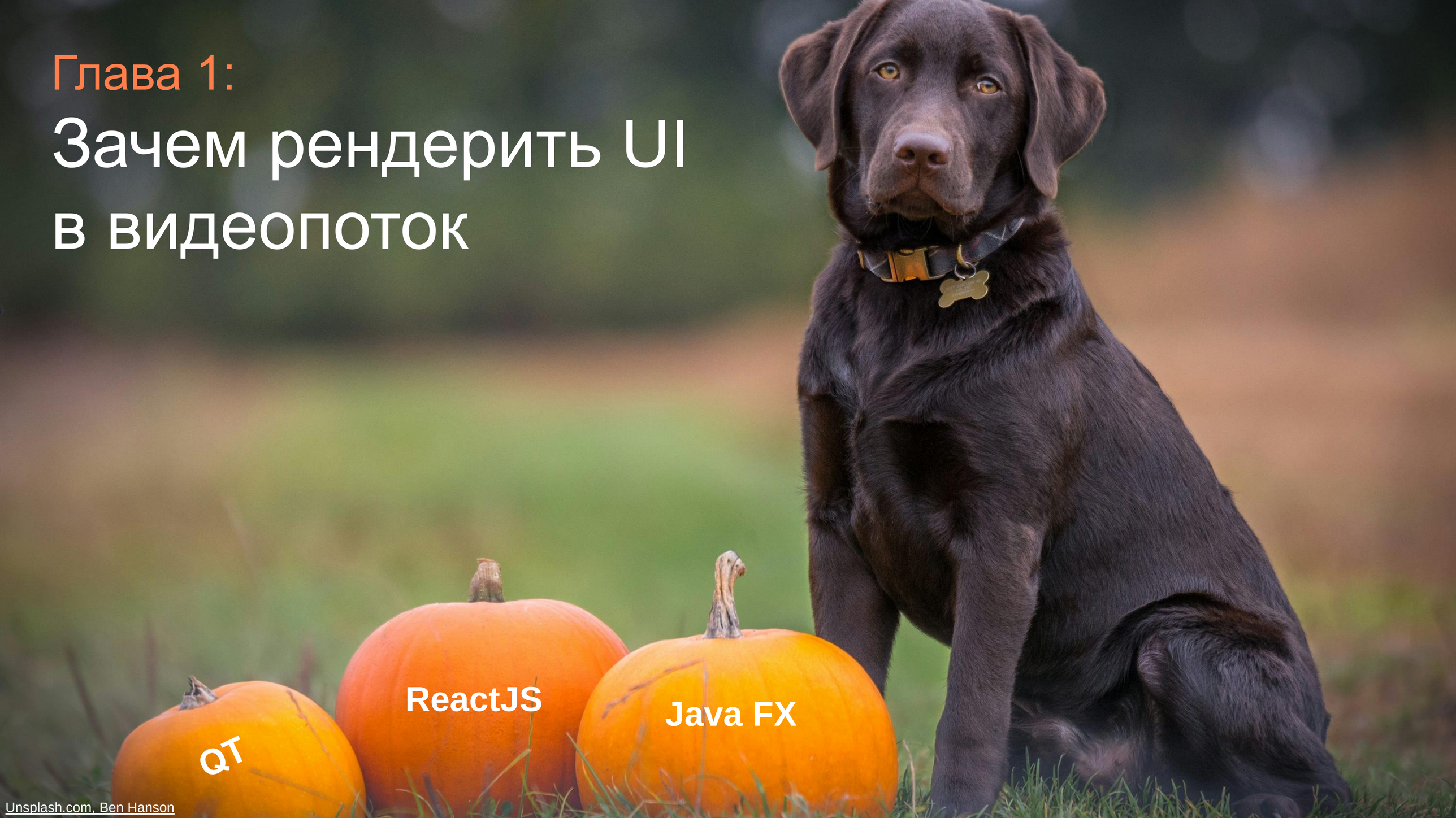


Session Description Protocol



Глава 1:

Зачем рендерить UI в видеопоток



Unsplash.com, Ben Hanson

Подключить переговорки к Телемосту

SIP-коннектор

- Камера
- Экран
- Микрофон
- Динамик

Кодеки для переговорных комнат

- Стек SIP
- Для 1x1 звонков
- Отображает то, что получает
- В видео потоке

Cisco.com, Webex room bar

Дозвон из «дикого» интернета

Сначала поднять трубку

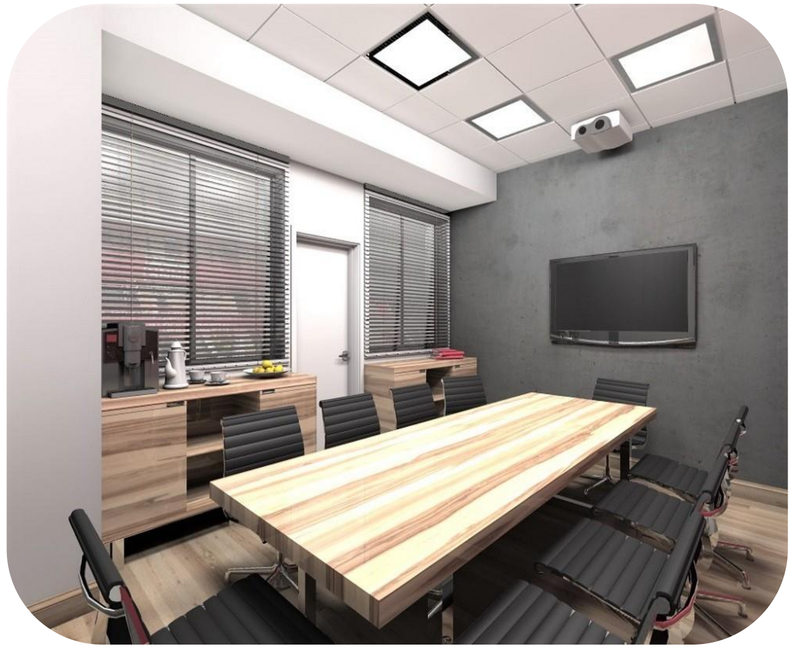
Потом аутентификация

Нужно дешево

По CPU

По Bandwidth

Сервис, который встречает гостей

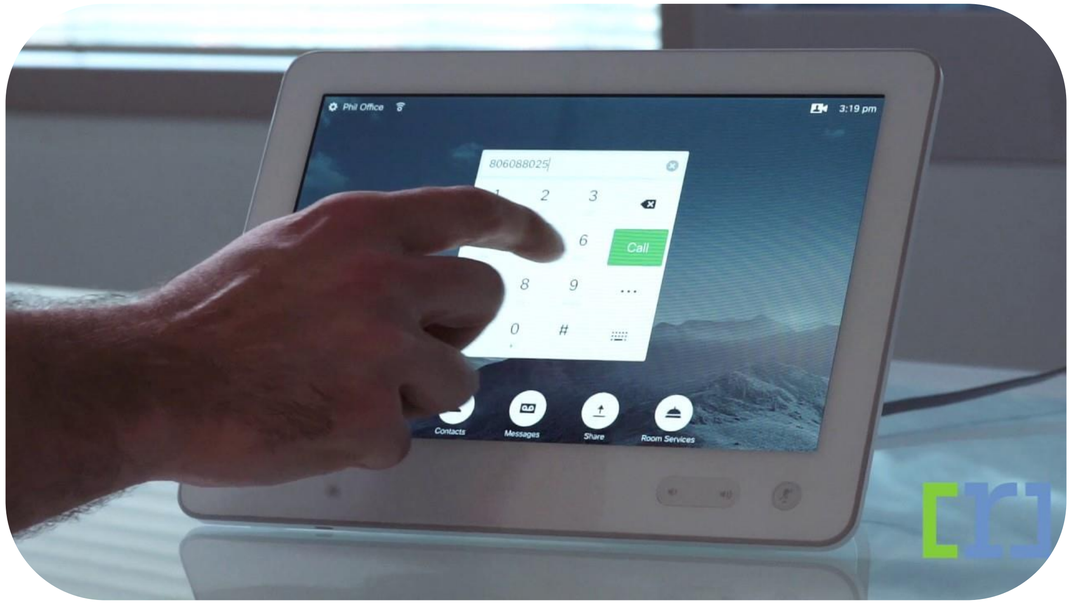


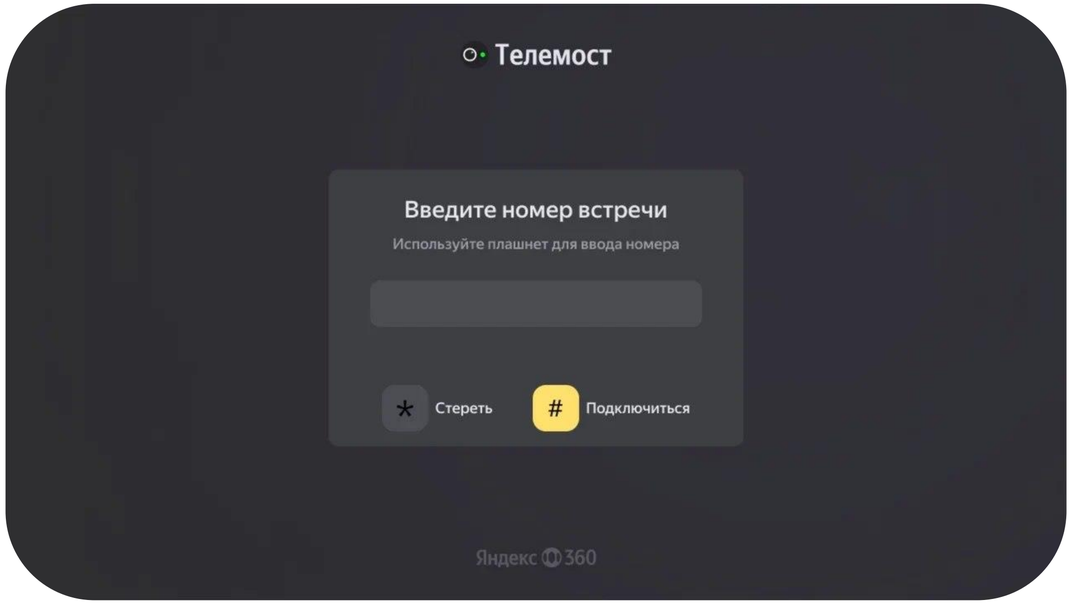
SIP

vavilon-mebel.ru, freepik.com

Welcome страница

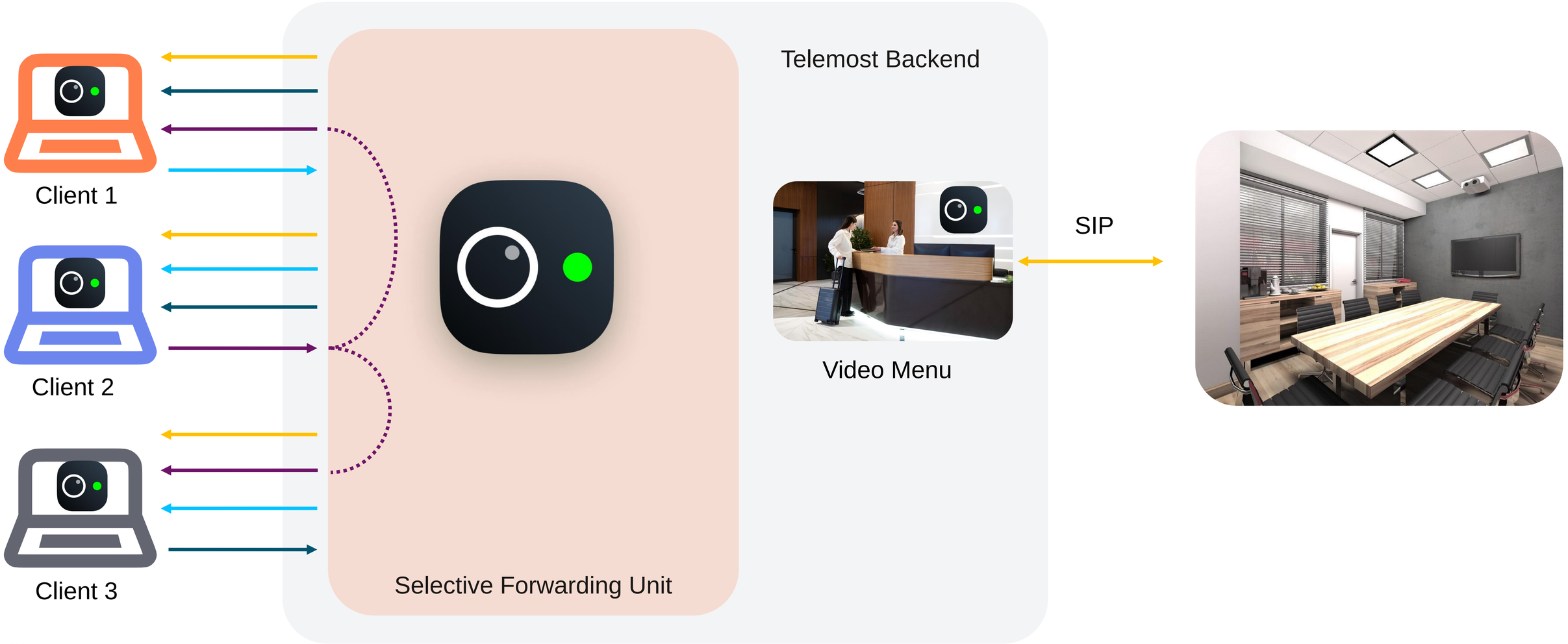
Video Menu





Youtube, RoomReady, How to make a call to a video address

Video Menu для переговорки



vavilon-mebel.ru, freepik.com

Глава 1:
Требования к welcome-странице

1

Обрабатывать
ввод < 100 ms

2

Незарегистрированные
пользователи

3

CPU важнее
Bandwidth

720p

30 FPS

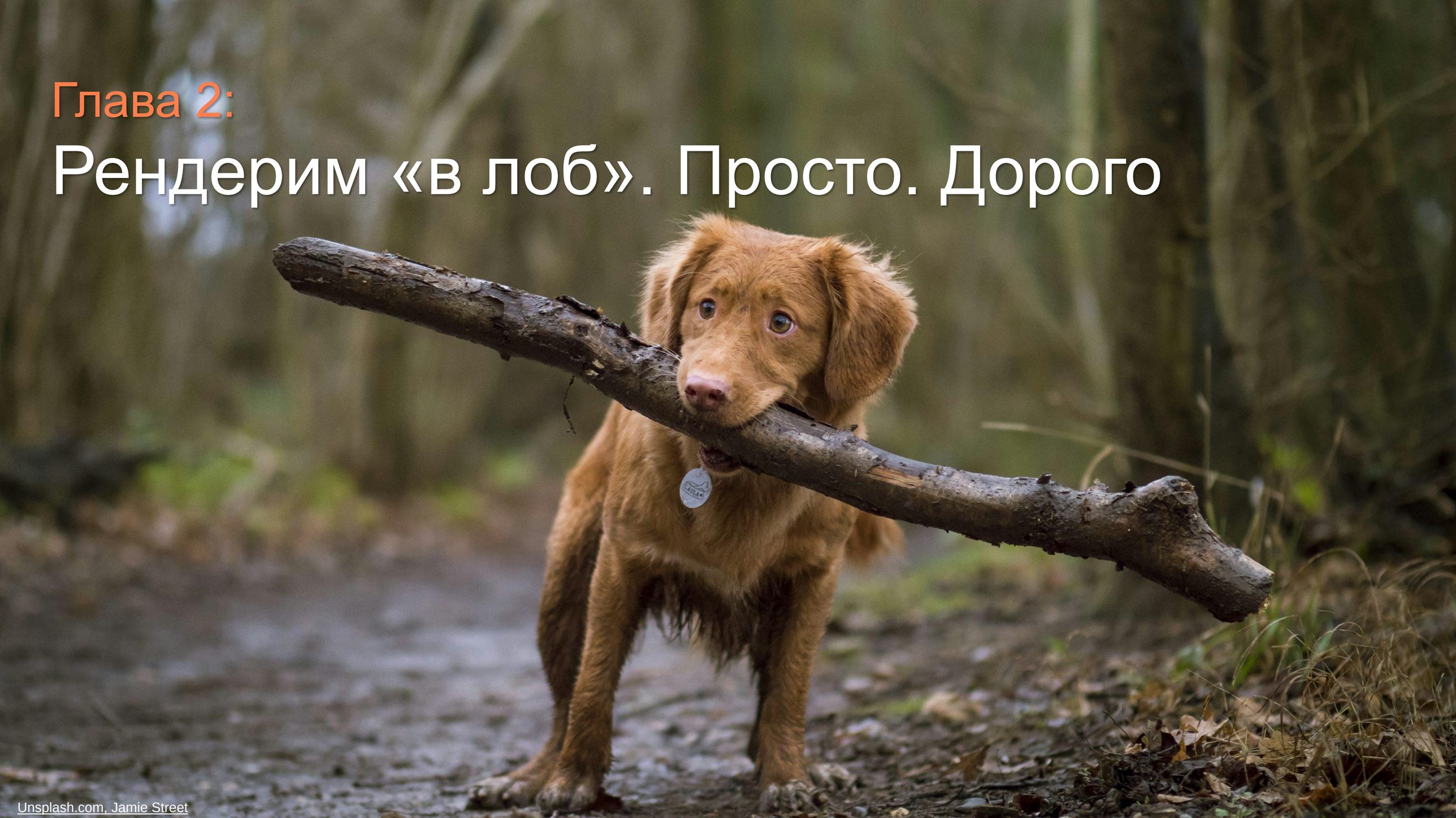
4

Интерфейс —
видеопоток

Youtube, RoomReady, How to make a call to a video address, freepik.com, Shedevrum.ai

Глава 2:

Рендерим «в лоб». Просто. Дорого



Unsplash.com, Jamie Street

В спеку!
Как слать кадры в переговорку?

- Понять, что можно слать
- Упаковать в RTP
- Не потерять на сети

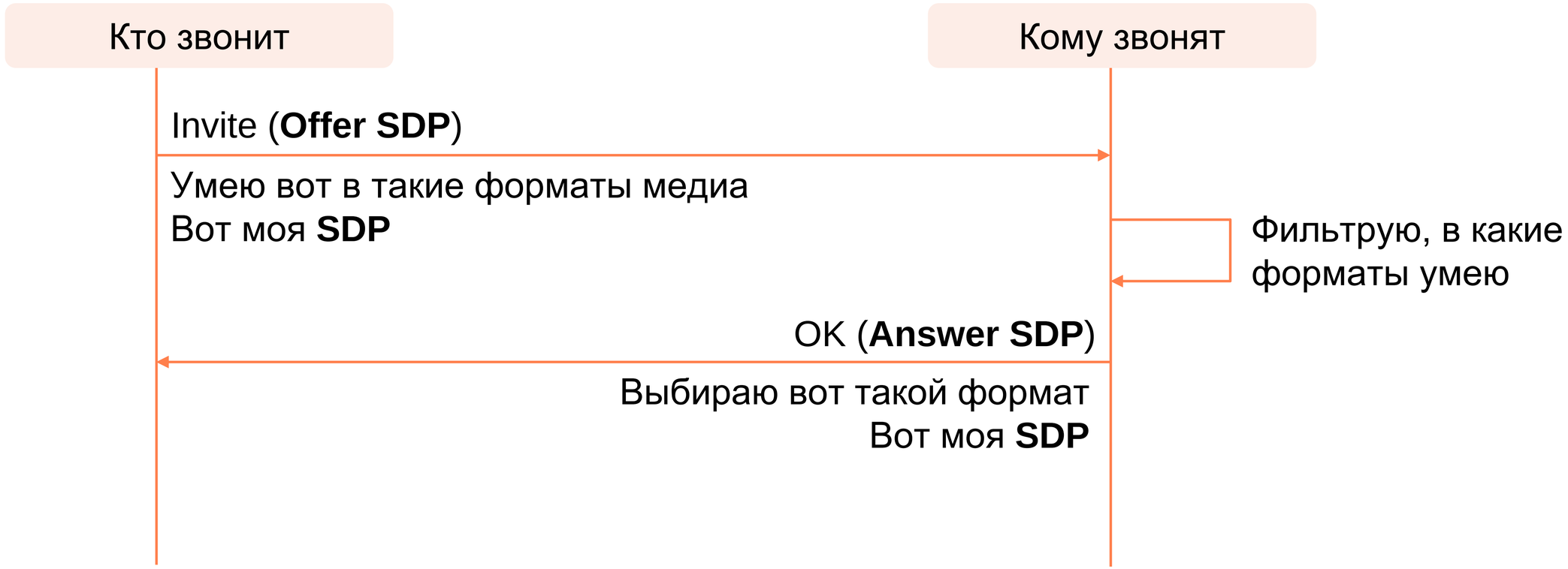
Unsplash.com, Jexo

Понять, что можно слать

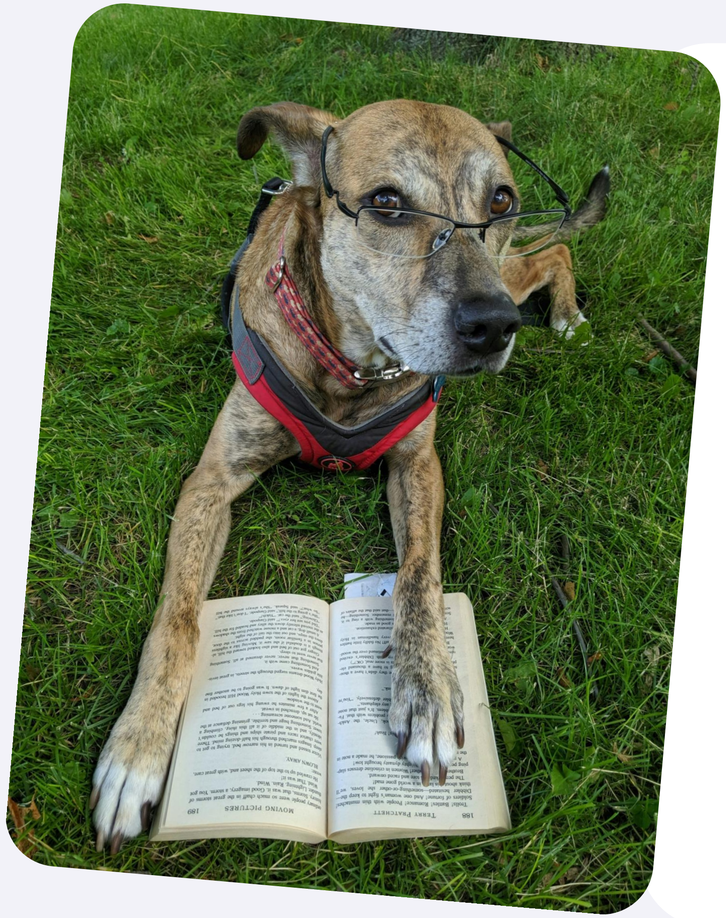
- Какой можно FPS?
- Какие можно размеры кадров?
- Какой можно битрейт?

Unsplash.com, Josh Rakover

Session Description Protocol



Что от нас хочет переговорка?



```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```


Unsplash.com, Josh Rakover

RTP/AVP

RTP

https://datatracker.ietf.org/doc/html/rfc3550

RTP/AVP

https://datatracker.ietf.org/doc/html/rfc3551

H.264 RTP payload format

https://datatracker.ietf.org/doc/html/rfc6184

SDP: Payload Type

- ID формата
- Нам предлагают несколько
  На выбор

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

SDP: частота дискретизации

- Шаг шкалы времени
- 1/90 000 сек.

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

SDP: частота дискретизации

Позволяет выразить presentation timestamp в целых числах

Например для FPS:

- 24 (HDTV): 3750 per frame
- 25 (PAL): 3600 per frame
- 30 (HDTV): 3000 per frame

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

SDP: H.264 profile

- Первый байт: Profile ID
  - 42(base 16) = 66(base 10)
  - Значит - Baseline Profile
- Второй байт: Constrained Flag
- Третий байт: Level ID
  - 16(base16) = 22(base 10)
  - Значит - Level = 2.2

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

H.264 Profile: правила игры энкодера

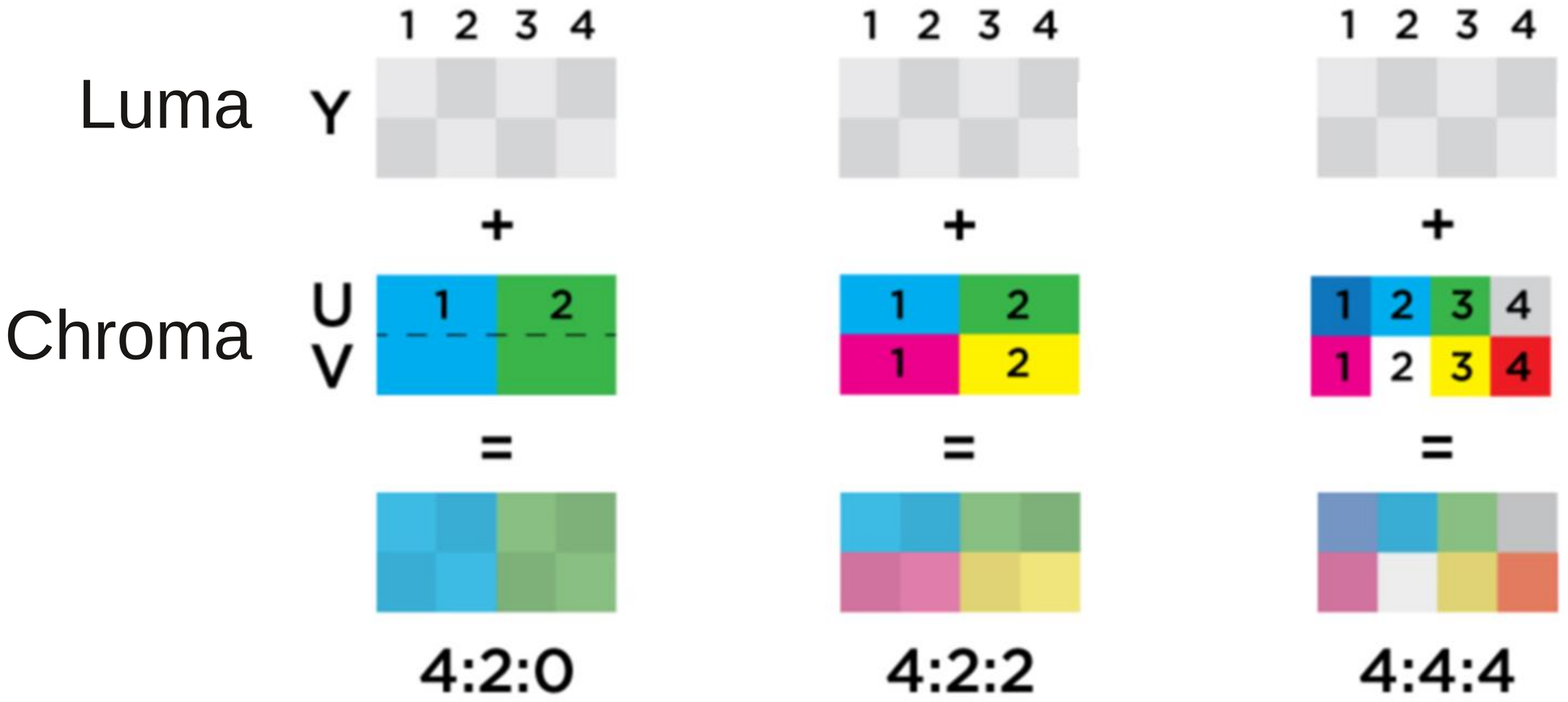
Энкодер обязуется
не использовать сложные
техники кодирования

В профиле Baseline Constrained

- Отключены B-Frames
- Отключено энтропийное кодирование макроблоков
- Только YUV 4:2:0 Chroma Subsampling

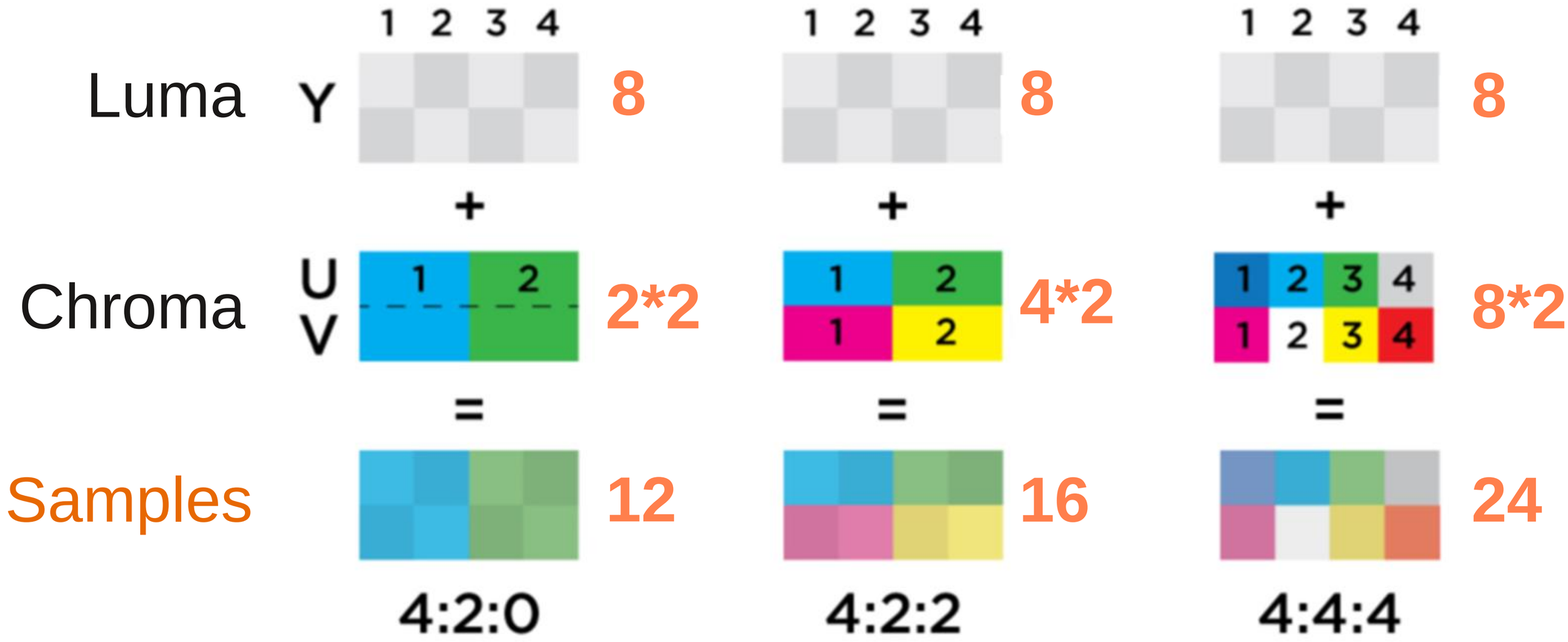
Полный список отключённых фич — в Annex A спецификации

YUV Chroma Subsampling



biamp.com How chroma subsampling works

YUV Chroma Subsampling – эффективность



biamp.com How chroma subsampling works

Profile Level — лимит производительности

Level = 2.2

Будет слать не больше

- 854x480 12.5 FPS

Profile Level — лимит производительности

Level = 2.2

Будет слать не больше

- 854x480 12.5 FPS

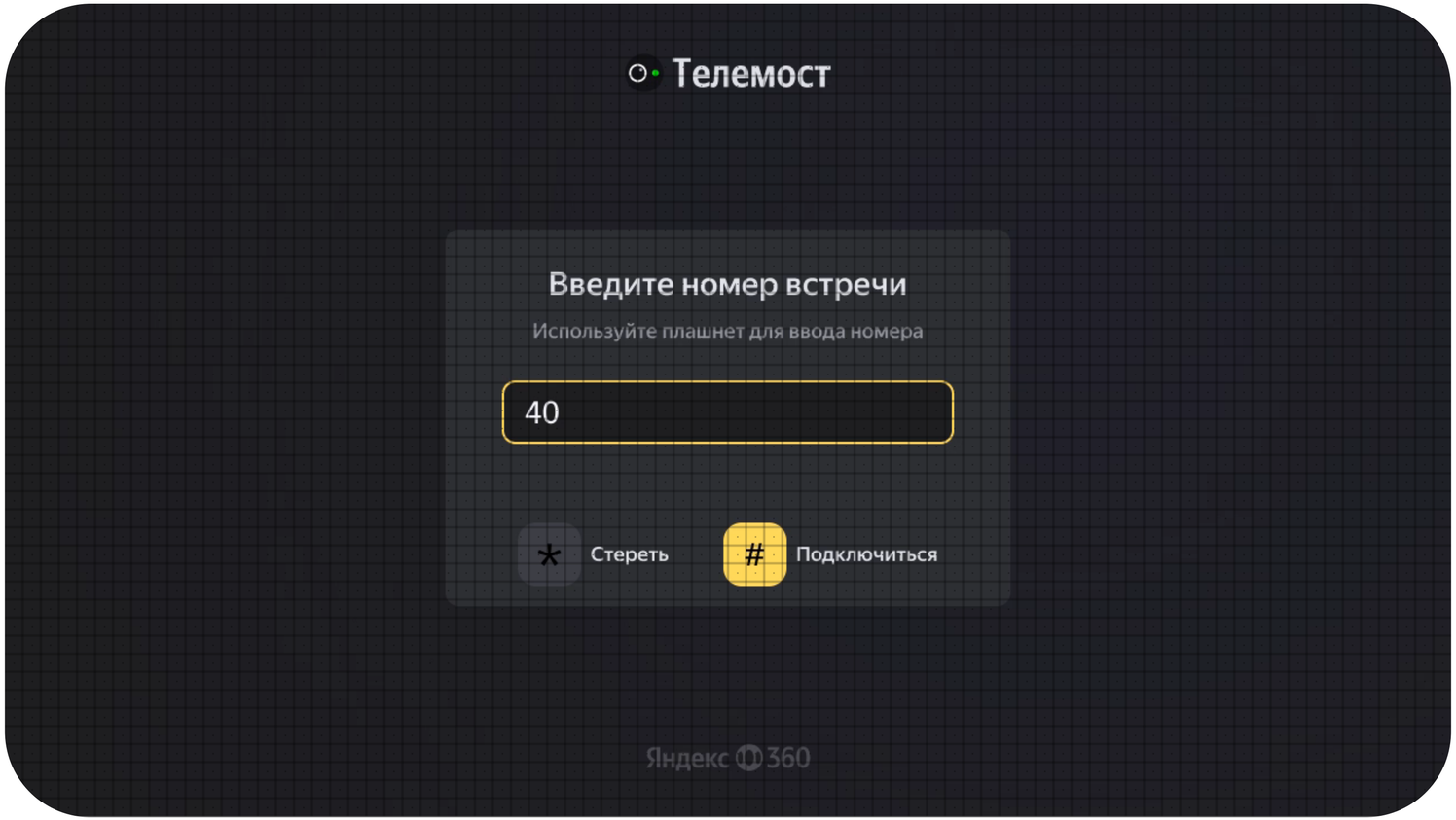
А какой формат сможет принять?

SDP: размер картинки

- max-mbps
  Количество макроблоков / сек
- max-fs
  Размер кадра в макроблоках

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

Знакомьтесь: макроблоки



SDP: размер картинки

- max-mbps

  Количество макроблоков/сек
- max-fs
  - Макроблоков 16 × 16 в кадре
  - 1920 × 1080 / 16^2

    ==

    8100 макроблоков

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

SDP: bitrate

- max-br
  - Максимальный битрейт
  - В юнитах 1200 бит/сек
  - 5000 × 1200/1024^2

    ==

    5.7 Mbps

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

SDP: что нам сказала переговорка?

Предлагает «97» обозначить:

- Мерим время в единицах 1/90 000 сек.
- Готова декодить 490 000 MB/sec
- Готова рисовать 8160 MB (1080p)
- Готова принимать 5.7 Mbps
- Запрещены B-Frames, энтропийное кодирование, etc
- Сама будет слать не больше
  - 854x480 12.5 FPS

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

Что именно поддерживает переговорка?

- Какой можно FPS?

  max_mbps/max_fs = 60 FPS

- Какие можно размеры кадров?
  - 1920 × 1080 / 16^2 = 8100 (16:9)
  - 1664 × 1248 / 16^2 = 8112 (4:3)
  - Разрешено: 8160

- Какой можно битрейт?

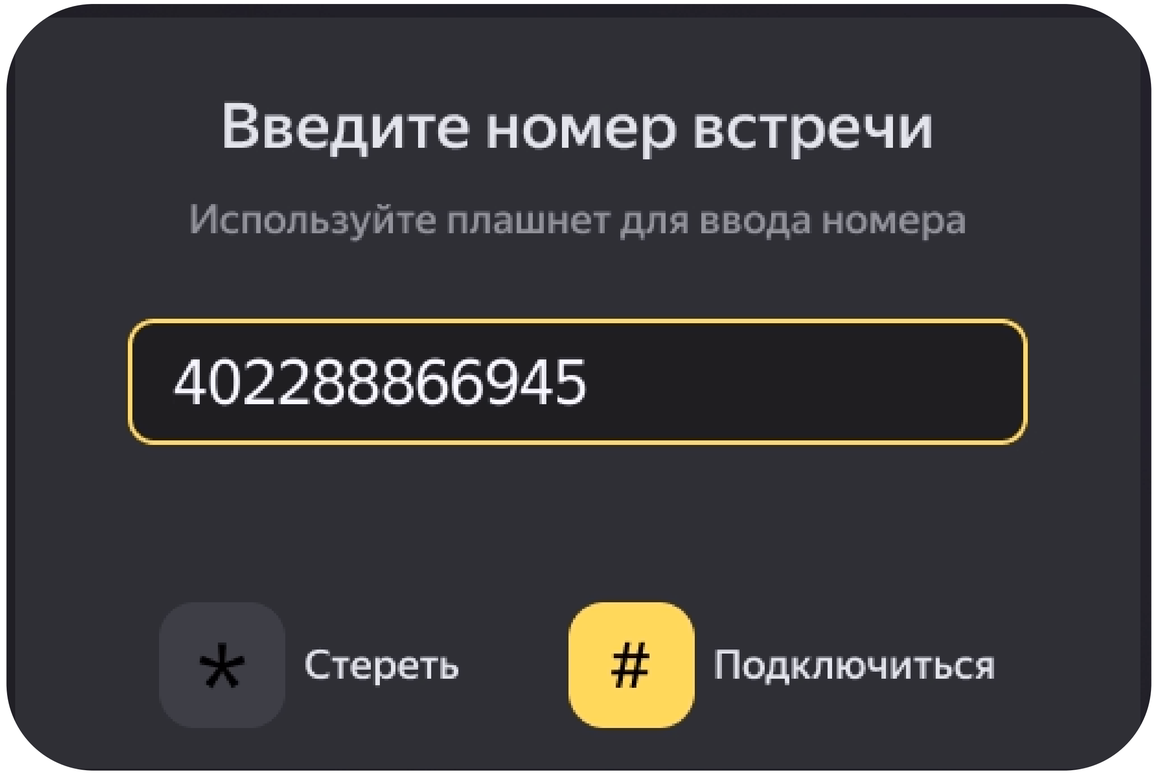
  max_br × 1200 / 1024^2 = 5.7 Mbps

```
a=rtpmap:97 H264/90000
a=fmtp:97
    profile-level-id=428016;
    packetization-mode=0;
    max-mbps=490000;
    max-fs=8160;
    max-cpb=200;
    max-dpb=16320;
    max-br=5000;
    max-smbps=490000;
    max-fps=6000
```

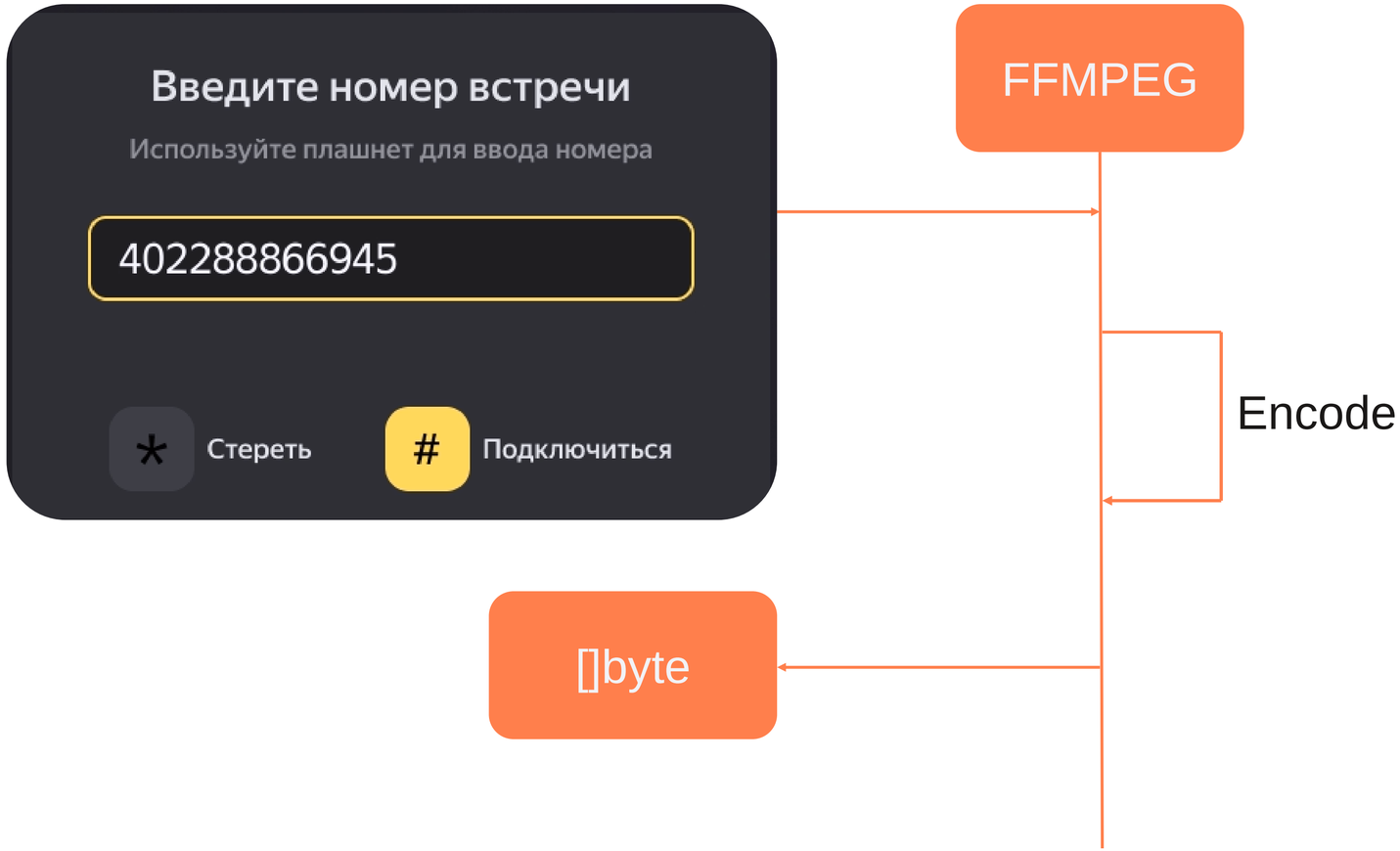
Рисуем UI и кодируем

- Рисуем цифры
  и формочки поверх
  бэкграунда (RGB)

- sws_scale
  RGB → YUV420



Рисуем UI и кодируем 30 FPS



Кодируем 720p, 30 FPS

Encoding:
30 FPS, GOP 1800
0,34 CPU/call

Unsplash.com, Jamie Street

Соцопрос: а если слать только I-Frame?

Станет дороже

Станет дешевле

Ничего не изменится

www.hindustantimes.com

А если слать только keyframe?
Ответ

- Encoding: 30 FPS, GOP 1800:
  0,34 CPU/call

- Encoding: 30 FPS, GOP 1:
  0,40 CPU/call

Кадры стали
по 50 KB каждый (12 Mbps)

Как слать кадры в переговорку?

- ✓ Понять, что можно слать
- Упаковать в RTP
- Не потерять на сети

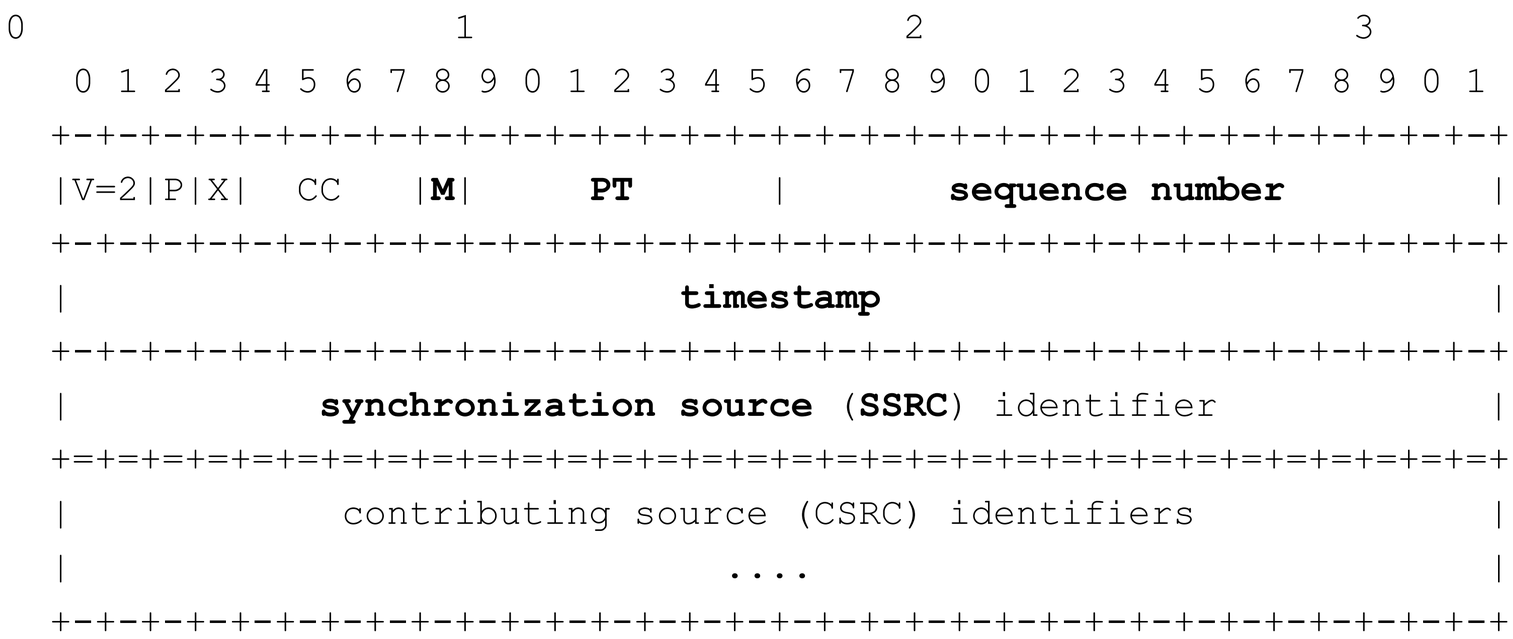
Unsplash.com, Jexo

Знакомьтесь:
Real-Time Transport Protocol (RTP)

- Для передачи медиапотоков
  - Время съёмки кадра
  - Большой кадр в несколько пакетов
- Доставлять необязательно.

  Можно выбросить
- Несколько потоков -> 1 UDP-порт

RTP Packet Header



Упаковка больших кадров в UDP-пакеты

- Timestamp одинаковый
- Sequence number увеличивается
- Marker bit = true — признак последнего пакета кадра

Посмотрим в wireshark

rtp.p_type==102

Info

```
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58549, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58550, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58551, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58552, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58553, Time=2425619682, Mark
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58554, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58555, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58556, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58557, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58558, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58559, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58560, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58561, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58562, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58563, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58564, Time=2425619682, Mark
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58565, Time=2425632641
```

Так, стоп. Почему timestamp одинаковый?

rtp.p_type==102

Info

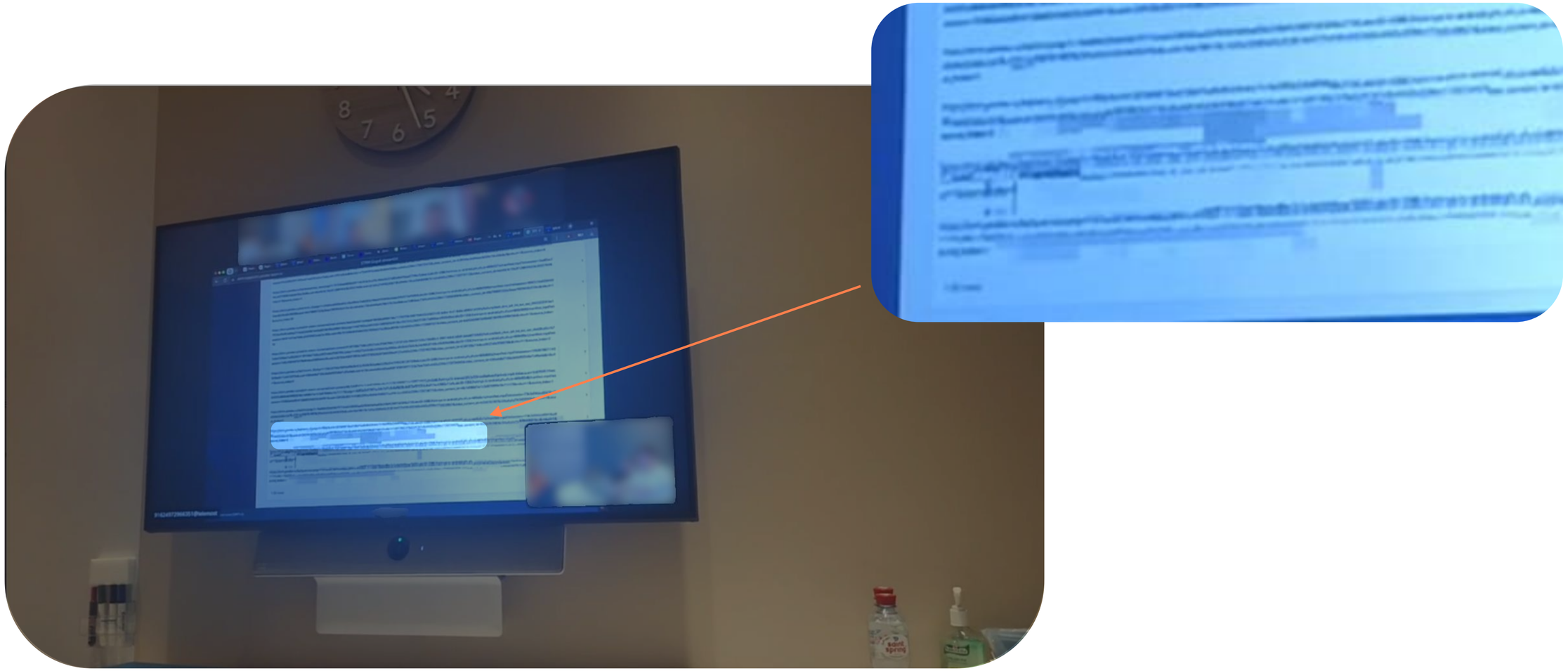
```
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58549, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58550, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58551, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58552, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58553, Time=2425619682, Mark
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58554, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58555, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58556, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58557, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58558, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58559, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58560, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58561, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58562, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58563, Time=2425619682
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58564, Time=2425619682, Mark
PT=DynamicRTP-Type-102, SSRC=0x9048D80, Seq=58565, Time=2425632641
```

Как слать кадры в переговорку?

- ✓ Понять, что можно слать
- ✓ Упаковать в RTP
- Не потерять на сети

Unsplash.com, Jexo

Чем грозит?



Чем грозит?

Потеря пакетов в RTP

Четыре варианта действий:

1. NACK — Negative ACKnowledgement
2. PLI — Picture Loss Indication
3. FEC — Forward Error Correction
4. Conceal — попробовать догадаться, что потерялось

Как ведут себя переговорки?

Некоторые переговорки:

- Не умеют NACK
- Рассчитывают на FEC
- При потере пакетов пытаются в Conceal. На глаз — не всегда успешно

Вывод: FEC важен

Словарик:

- NACK — Negative ACKnowledgement
- PLI — Picture Loss Indication
- FEC — Forward Error Correction

Wikipedia, Golden Jackal

Наша стратегия

PLI:

FFMPEG делает keyframe

NACK:

перешлём из кеша

FEC:

включим

Словарик:

- NACK — Negative ACKnowledgement
- PLI — Picture Loss Indication
- FEC — Forward Error Correction

Wikipedia, Golden Jackal

Как послать []byte по RTP

Нам хватило Pion
WebRTC stack для Golang

Pion Github

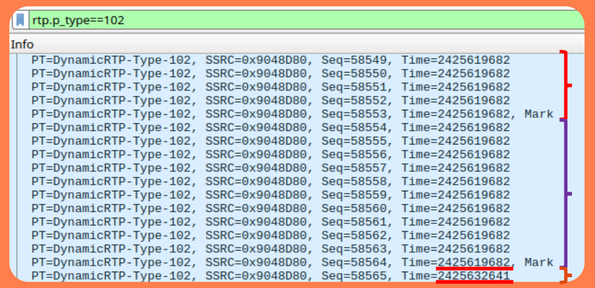
Спека прочитана!

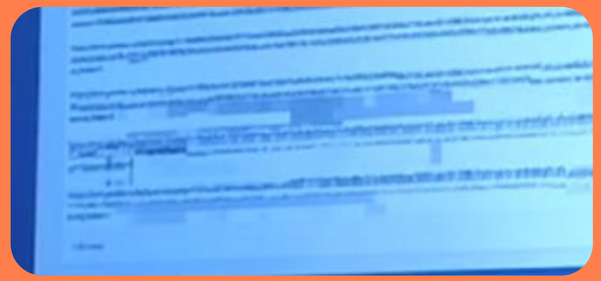
- ✓ Понять, что можно слать
- ✓ Упаковать в RTP
- ✓ Не потерять на сети

Unsplash.com, Jexo, www.flaticon.com, Dog Treat

Результат главы 2

```
a=rtpmap:97 H264/90000
a=fmtp:97
        profile-level-id=428016;
        packetization-mode=0;
        max-mbps=490000;
        max-fs=8160;
        max-cpb=200;
        max-dpb=16320;
        max-br=5000;
        max-smbps=490000;
        max-fps=6000
```





1

Понимаем, чего хочет переговорка

2

Пакетируем в RTP и отправляем

3

Работаем с потерей пакетов

Глава 3:
Оптимизируем по-простому

Unsplash.com, James Lee

Глава 3:

Оптимизируем по-простому

30 FPS для статичной менюшки — много.
Давайте понизим

0,34 CPU на звонок — дорого

Снизим хотя бы до 0,05 CPU/звонок

В спеку!

А можно не слать кадры,
когда ничего не поменялось?

Unsplash.com, Baptist Standaert

Зависит от реализации

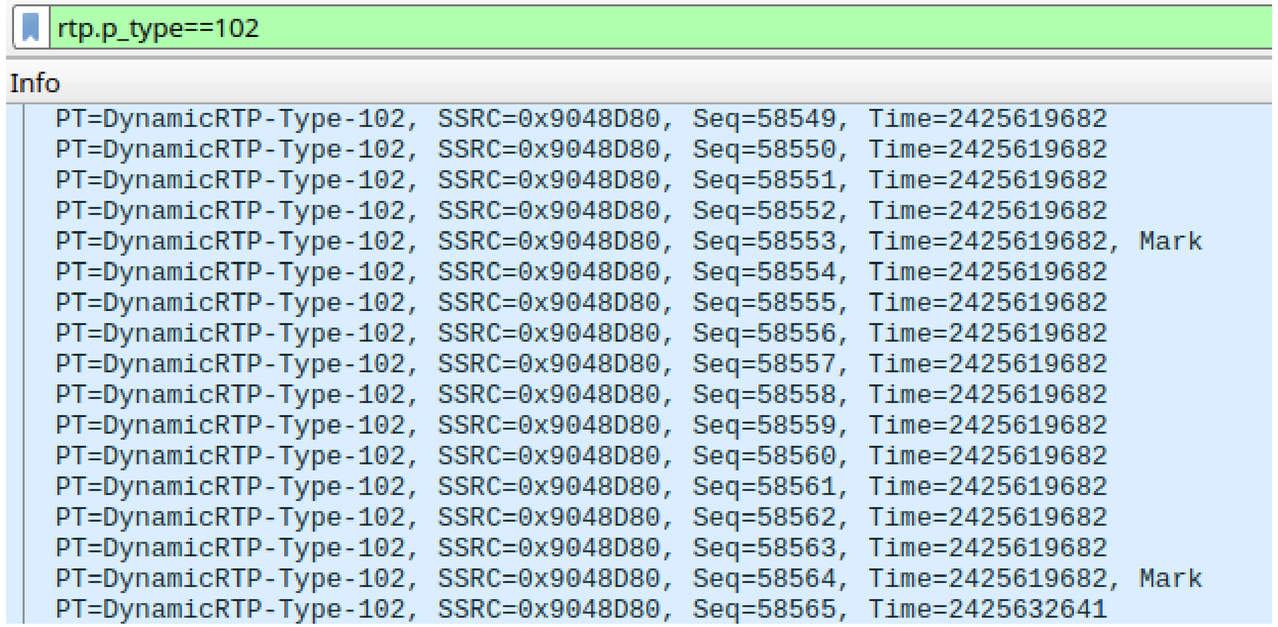
Dora the Explorer

А можно не слать кадры, когда ничего не поменялось?

Chrome:
да!

Но в Chrome же WebRTC!

- Требует DTLS
- Пакеты не закепчурить =(



Достать из Chrome расшифрованный pcap

```
cd /home/topright/tmp/webrtcdump
rm -rf *
chromium --enable-logging --v=3 --force-fieldtrials=WebRTC-
Debugging-RtpDump/Enabled/ --user-data-dir=`realpath .`
'http://localhost:12135/httpstatic/webclient.html'
http://localhost:12135/httpstatic/webclient.html

cat chrome_debug.log | grep RTP_DUMP > in.txt
text2pcap -D -u 1000,2000 -t %H:%M:%S. in.txt out.pcap
```

А можно не слать кадры, когда ничего не поменялось?

Chrome:
да!

Cisco-кодеки:
нет

чёрные экраны через 3 сек

На сколько можно снизить FPS?

Опытным путём выясняем, что ниже 1 FPS опускаться нельзя

Но 1 FPS => в среднем 0,5 сек. для реакции на нажатие

А можно повышать FPS, когда нужно?

Chrome:
да!

Cisco-кодеки:
нет

Все равно рисуют 1 FPS

Значит остановимся на

Постоянном
1 FPS

Unsplash.com, Richard Brutyo

Давайте сравнивать

Что такое типичный звонок?

- Дозвониться
- Ввести номер встречи
  - 0.5 сек задержки между символами
- Ошибиться
- Ввести ещё раз
- Зайти

Затраты

Encoding:
30 FPS, GOP 1800:
0,34 CPU/call

Encoding:
1 FPS, GOP 60:
0,023 CPU/call

Результат главы 3

0,34 → 0,023
CPU/Call

1

Снизили FPS
до 1

2

Умеем управлять
FPS

< 1/3 FPS —
чёрный экран

Непостоянный FPS
не помогает

3

Изучили реакцию
кодеков

¯_(ツ)_/¯

4

Совсем не слать
кадры нельзя

Unsplash.com, James Lee

Глава 4:

Хачим закодированный H.264

Если кадры нужно
отправлять — давайте отправим

P-фреймы с пустой дельтой можно
не энкодить, а списывать

Unsplash.com, 2Photo Pots, idum.uz

Идея

1. Хотим найти P-фрейм с нулевой дельтой
2. Хотим слать одно и то же, а не энкодить
3. В RTP-хедерах есть timestamp
4. ? В потоке H.264 тоже должен быть

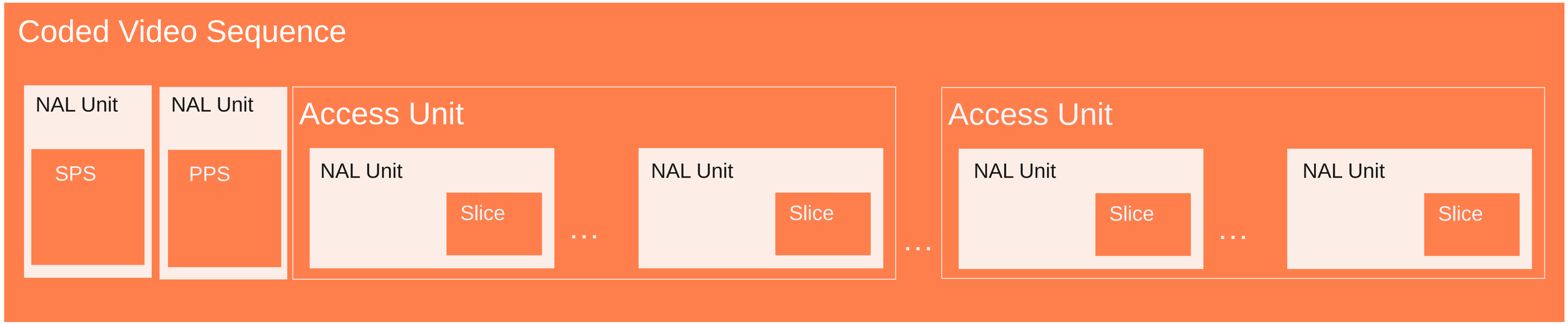
В спеку!

- Разберёмся, где в формате есть номера кадров
- Разберёмся, как их поменять

Unsplash.com, Jamie Street

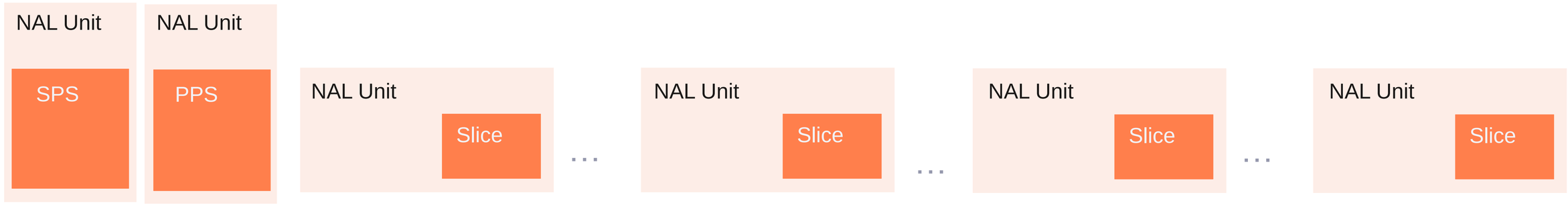
Матрёшка синтаксических элементов H.264

Матрёшка синтаксических элементов H.264



Network Abstraction Layer (NAL) Unit

- «Пакет» в H.264
- Поток байтов в формате H.264 состоит из NAL Unit
- Тело NAL Unit — поток бит. Не байт



Coded Video Sequence

aka Group Of Pictures (GOP)

Начинается с Instant Decoder Reset сегмента

aka I-Frame

Далее ряд зависимых кадров

aka P-Frame, B-Frame

Sequence Parameter Set

- Определяет правила
  для кодирования Coded Picture
  - Profile, Level
  - Width, Height, Crop offset
  - Chroma Format
  - Frame Num len (bits)
    - log2_max_frame_num_minus4
    - Количество закодированных кадров в буфере (минус 4)
    - В нашем случае 0. То есть в буфере 2^4=16 кадров

Picture Parameter Set

Общие параметры сжатия слайсов

Coded Video Sequence может содержать несколько

Кадры будут ссылаться на нужный PPS по ID

Содержит ссылку на SPS

Access Unit

IDR Access Unit

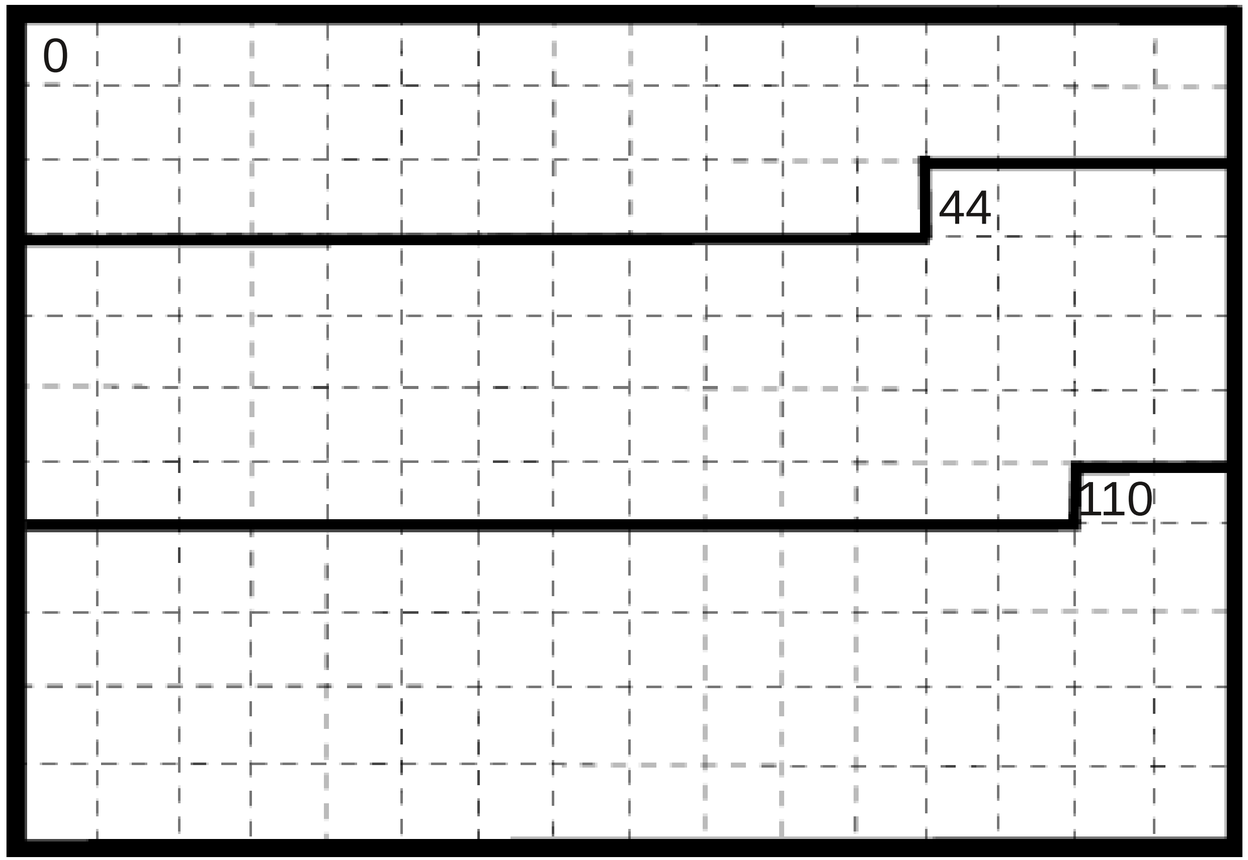
для I-Frame
Intra-coded Frame

Non-IDR Access Unit

для

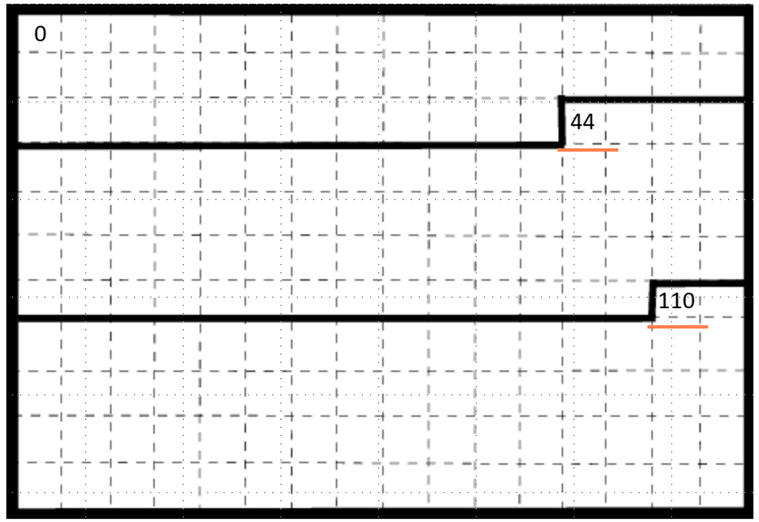
- P-Frame
  Predictive-coded Frame
- B-Frame
  BI-predictive-coded

Slice. Кадры режут на кусочки

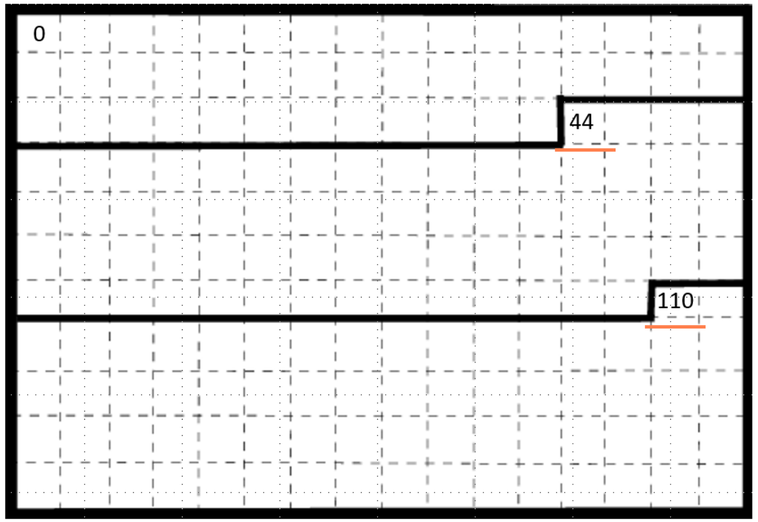


Slice. В разных кадрах одинаковые границы

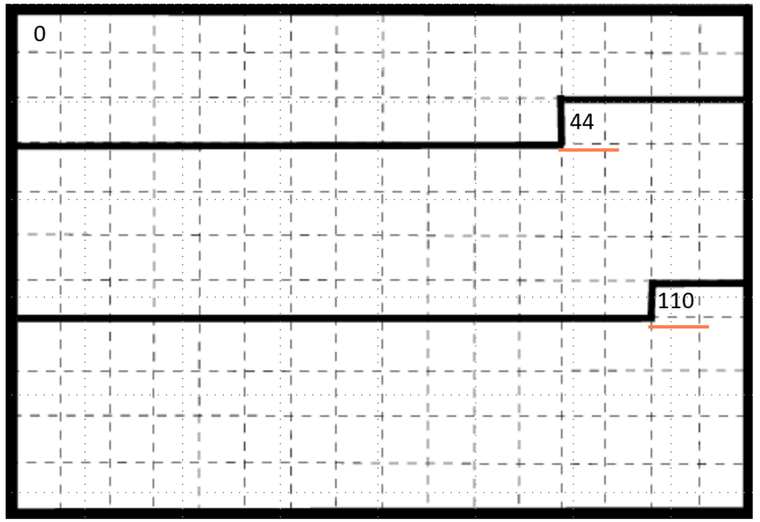
frame_num: 1



frame_num: 2



frame_num: 3



Когда потерялся Slice

Slice: зачем разрезать кадр

- Помещается в UDP-датаграмму
- Параллельное кодирование/декодирование
- Можно послать первый, пока последний ещё не заэнкожен

И совсем не про нас:

- В каждом Slice свои настройки сжатия
- Ограничить Error Propagation между частями кадра

Из чего состоит Slice: Header

- first_mb_in_slice — номер первого макроблока
- slice_type: P, B, I
- pic_parameter_set_id — ссылка на PPS (и на SPS)
- frame_num — номер кадра

...

Нашли! Slice Header!

- first_mb_in_slice — номер первого макроблока
- slice_type: P, B, I
- pic_parameter_set_id — ссылка на PPS (и на SPS)
- frame_num — номер кадра

...

Для чего номер кадра?

1

Порядок кадров
В decoding order

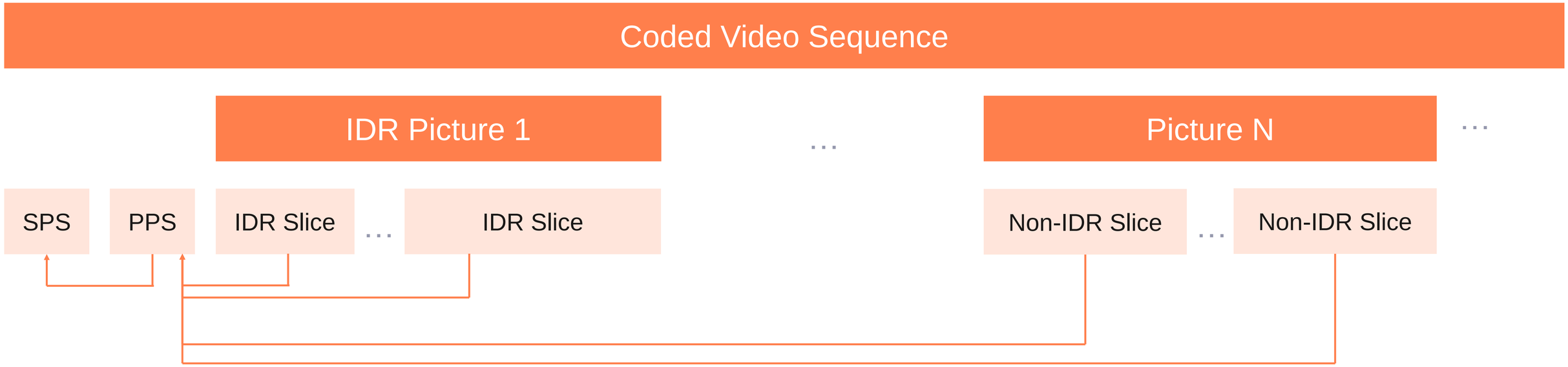
2

Для ссылок на раскодированные кадры

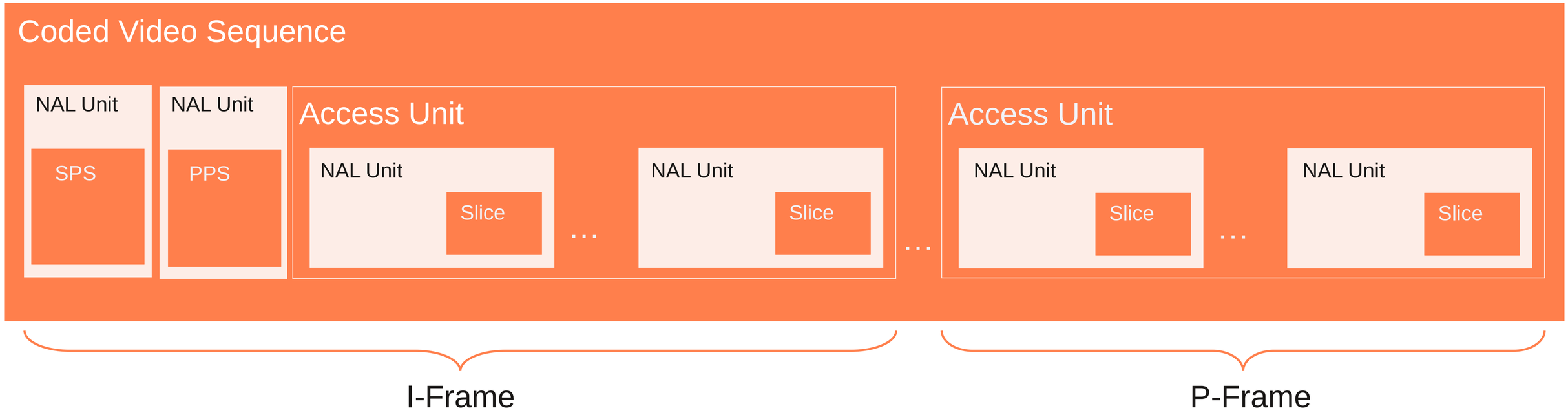
3

В нашем случае принимает значения от 0 до 15

Собираем


Coded Video Sequence
IDR Picture 1
...
Picture N
...
SPS
PPS
IDR Slice
...
IDR Slice
Non-IDR Slice
...
Non-IDR Slice

Что кодирует FFmpeg?



Как добраться до номера кадра?

Разобрать кадр
от FFMPEG
на NAL-юниты

Найти среди них
Slice

Подставить в каждый
Slice номер нужного
кадра

В спеку!

- Разберёмся, где в формате есть номера кадров
- Разберёмся, как их поменять

Unsplash.com, Jamie Street

Знакомьтесь, Slice

01000001 00000000 00010100 10100010

01101010 10000000 11111010 00000000

10100000 11000000

Dora the explorer

Slice — это строго один NAL Unit

u(x) — прочитать x-бит

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)

NAL Unit Header

01000001 00000000 00010100 10100010

01101010 10000000 11111010 00000000

10100000 11000000

forbidden_zero_bit — просто нолик
nal_ref_idc — насколько NAL-юнит жалко потерять
  0 — жалко
  3 — не жалко
nal_unit_type — тип NAL Unit
  1 -> Coded slice of a non-IDR picture

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)

Далее Slice Header

u(x) — прочитать x-бит
ue(v) – прочитать exponential-golomb code

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag == 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		

Exponential Golomb Code

Базово — для записи
Unsigned Integer

Кодирует и значение,
и длину в битах

Как работает Exp-Golomb code?

Пример:

codeNum = 1111b + 0101b = 00010100b = 20

Exp-Golomb Code: эффективность?

Code	Значения	Занял (бит)	Ceil(log2(max_code_num))
1	0	1	0
0 1 x0	1..2	3	2
0 0 1 x1 x0	3..6	5	3
0 0 0 1 x2 x1 x0	7..14	7	4
0 0 0 0 1 x3 x2 x1 x0	15..30	9	5
0 0 0 0 0 1 x4 x3 x2 x1 x0	31..62	11	6

Slice Header

01000001 00000000 00010100 10100010

01101010 10000000 11111010 00000000

10100000 11000000

first_mb_in_slice — номер первого макроблока в слайсе
2640 (из 3600)
slice_type — тип предсказания: I/P/B
5 -> P slice
pic_parameter_set_id — ссылка на PPS
frame_num — нашли!

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag = = 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		

Итого: как добраться до номера кадра?

1. Разбираем Slice Header по спеке
2. Позиция номера кадра будет известна после разбора
3. В битовом стриме меняем номер кадра
4. «Хвост» копируем без изменений

Собираем

- H.264 битстрим — последовательность NAL Units
- NAL Units типа Slice от одной Coded Picture составляют Access Unit
- Slice трёх видов:
  - Intra-coded
  - Predictive-coded
  - Bi-Predictive-coded
- Общие параметры выделены в SPS и PPS

Unsplash.com, Jamie Street

Спека прочитана!

- Разберёмся, где в формате есть номера кадров
- Разберёмся, как их поменять

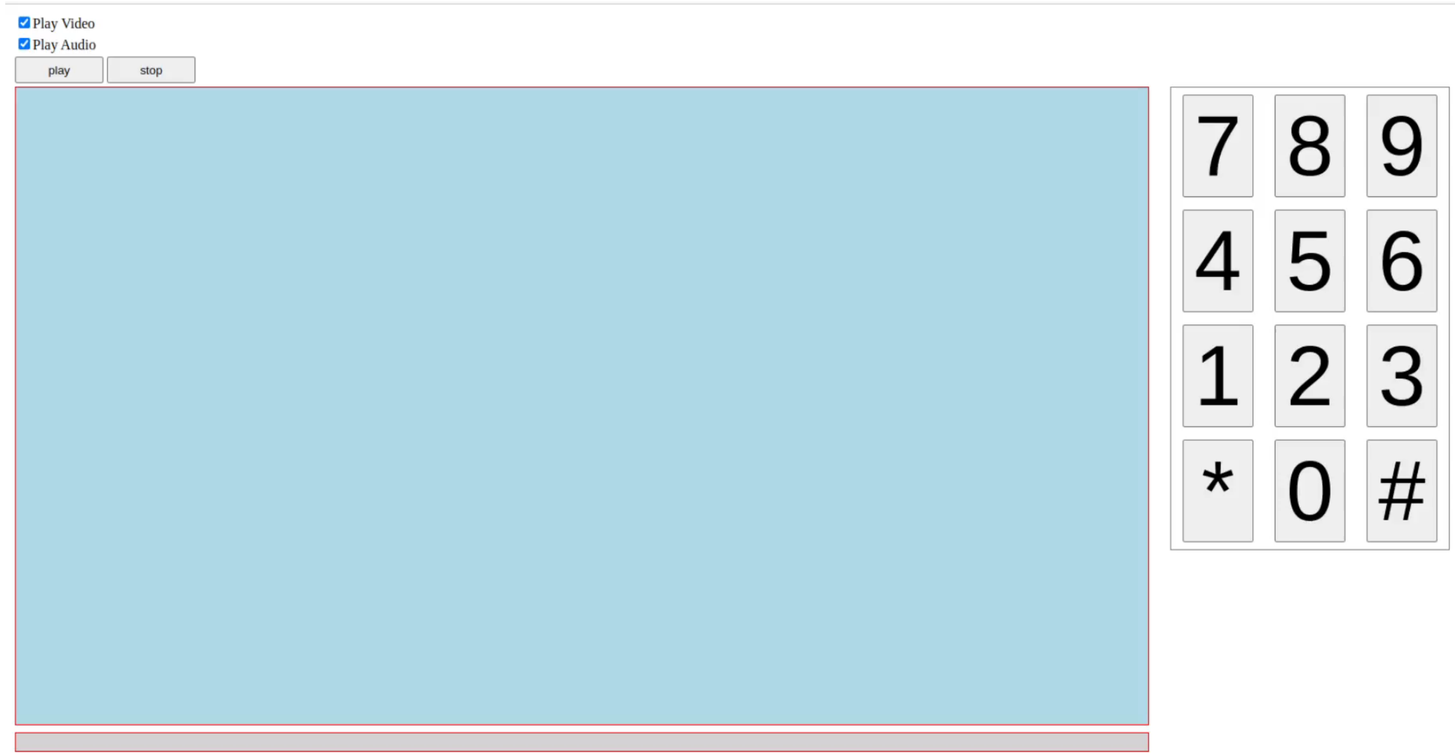
Unsplash.com, Jamie Street, www.flaticon.com, Dog Treat

План

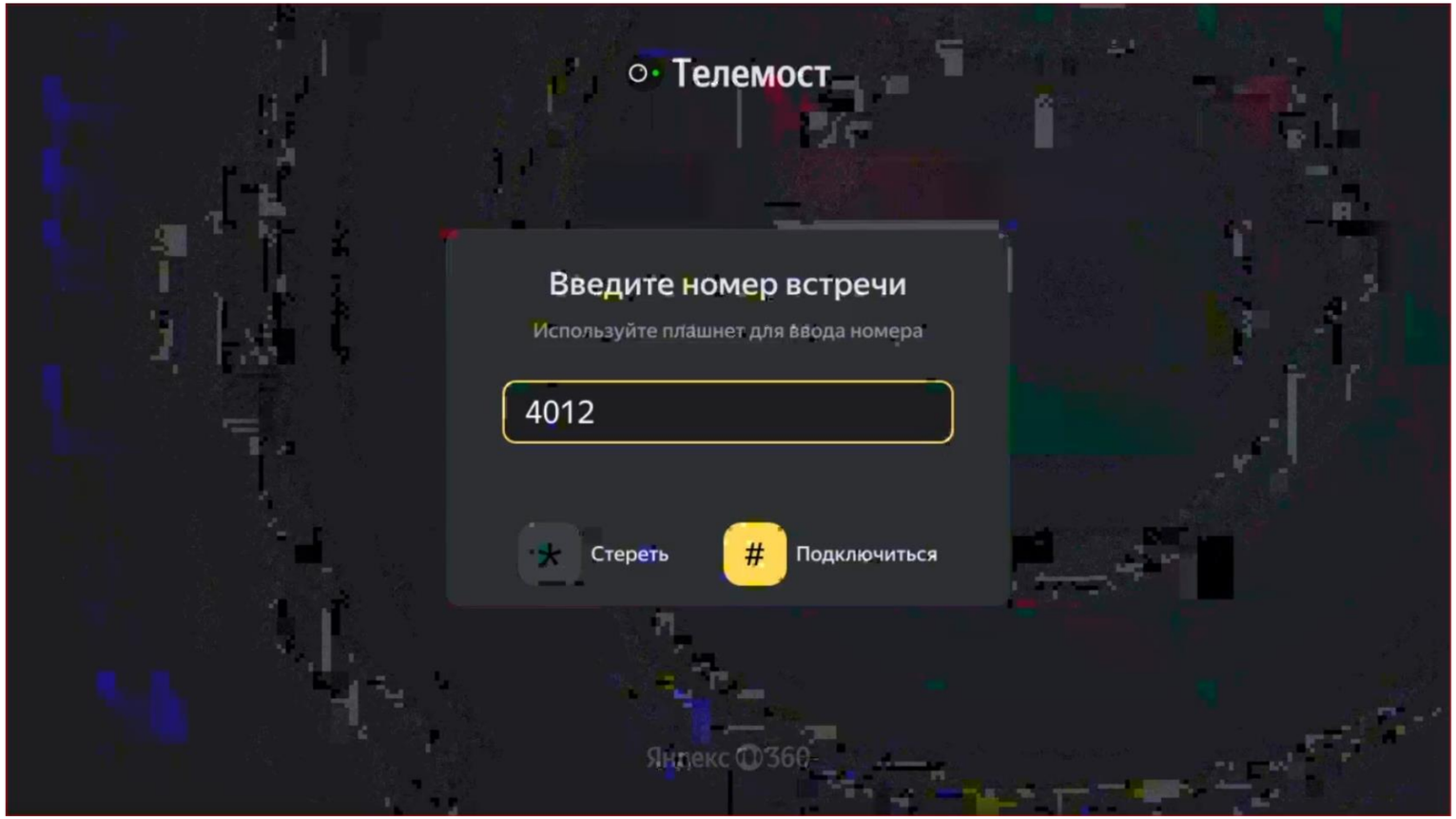
- Мы знаем, когда картинка не меняется
- Первый кадр без изменений — энкодим
- Остальные списываем из первого

Unsplash.com, Jamie Street

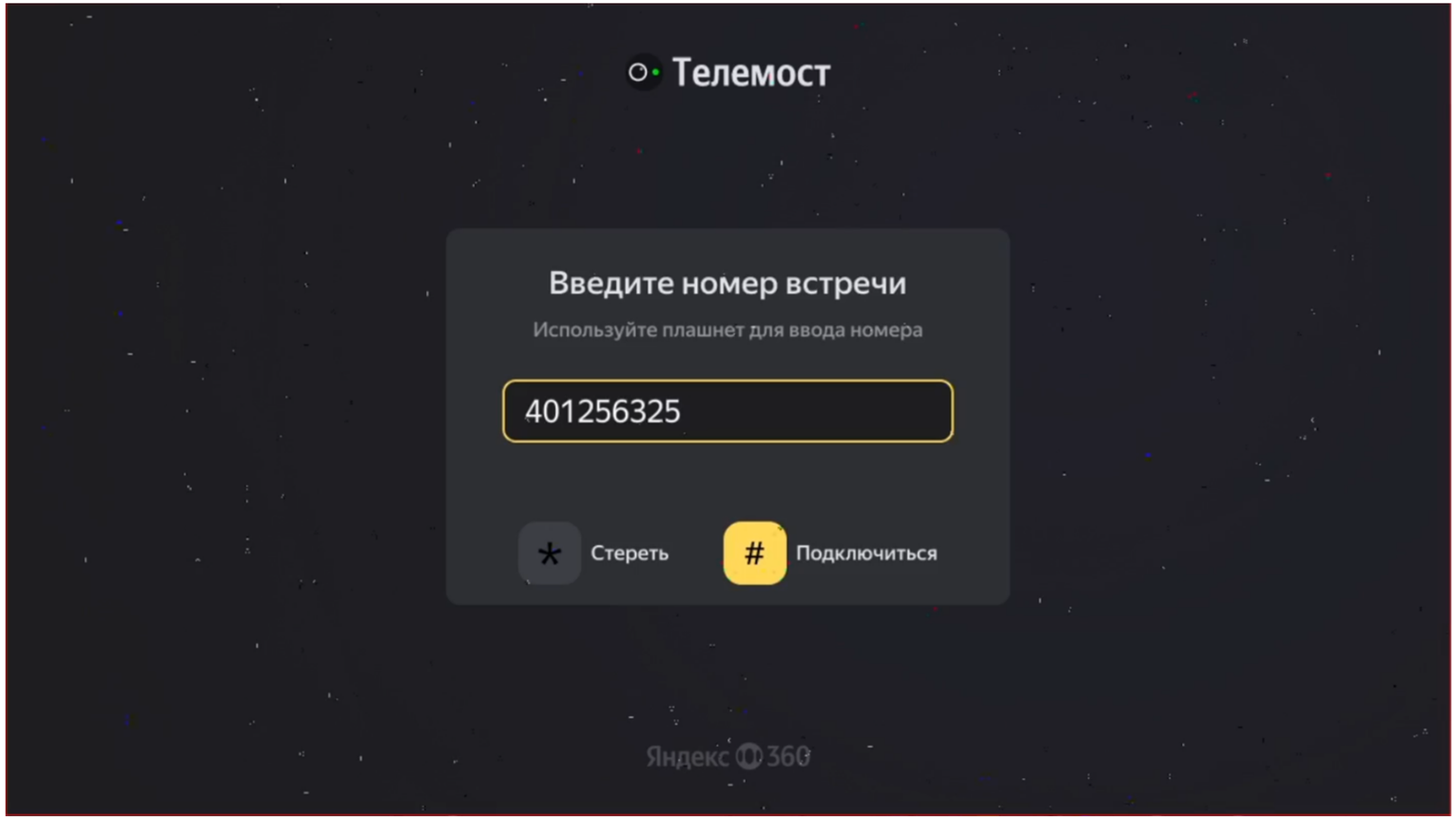
Пробуем «списать» первый пустой P-Frame



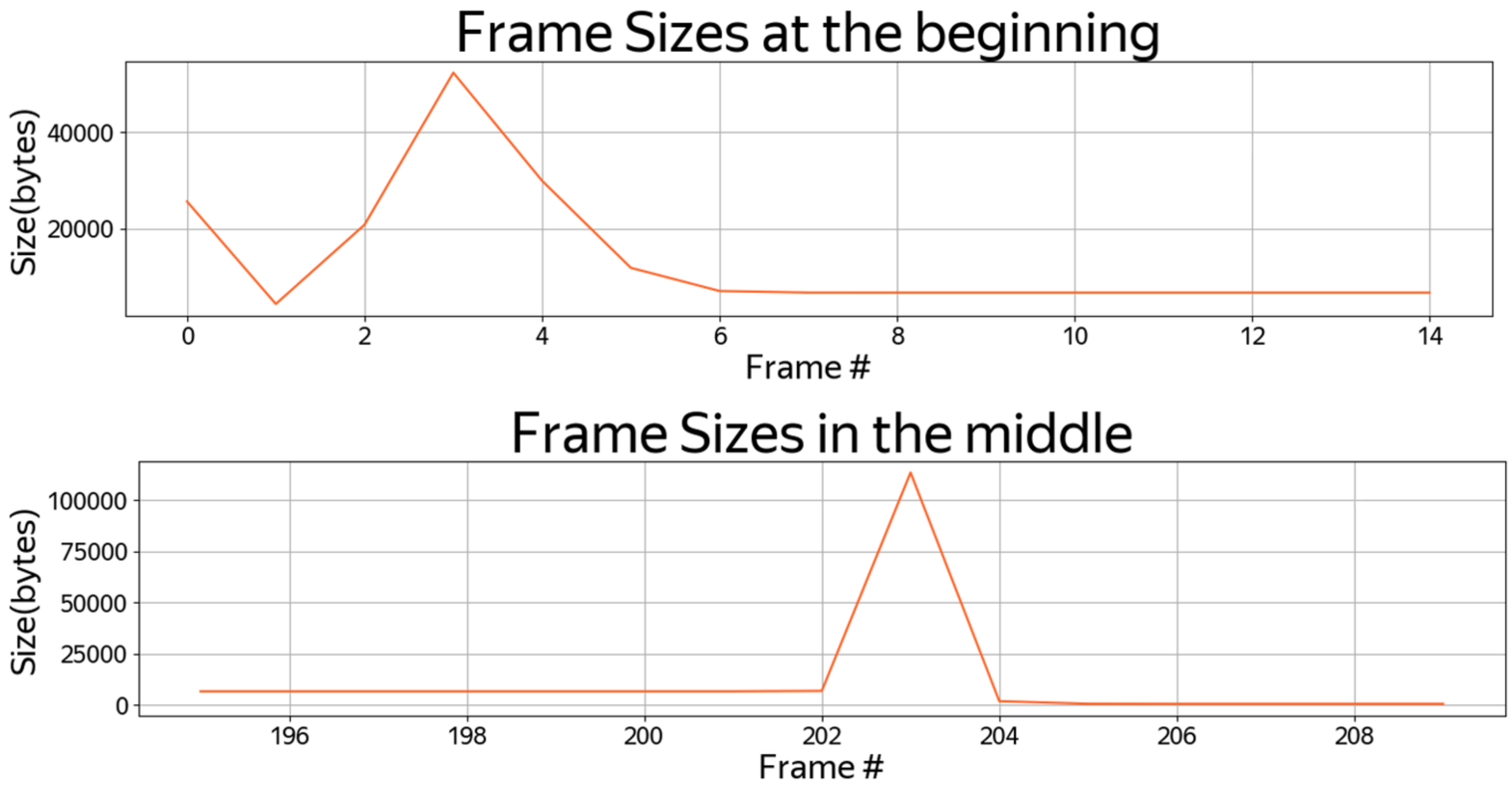
Повторили не-пустой P-Frame много раз



Попробовали другой не-пустой P-Frame



Размеры кадров от энкодера (без списывания)



Затраты на списывание кадра vs энкодинг

- Encoding: 30 FPS, GOP 1800: 0,34 CPU/call
- Encoding: 1 FPS, GOP 60: 0,023 CPU/call
- Replaying: 30 FPS, GOP 1800: 0,056 CPU/call

Результат главы 4

0.34 –> 0.056
CPU/Call

¯_(ツ)_/¯

1

Быстрая реакция
на нажатие клавиш

2

Списать быстрее,
чем энкодить

3

Умеем читать
битстрим H.264

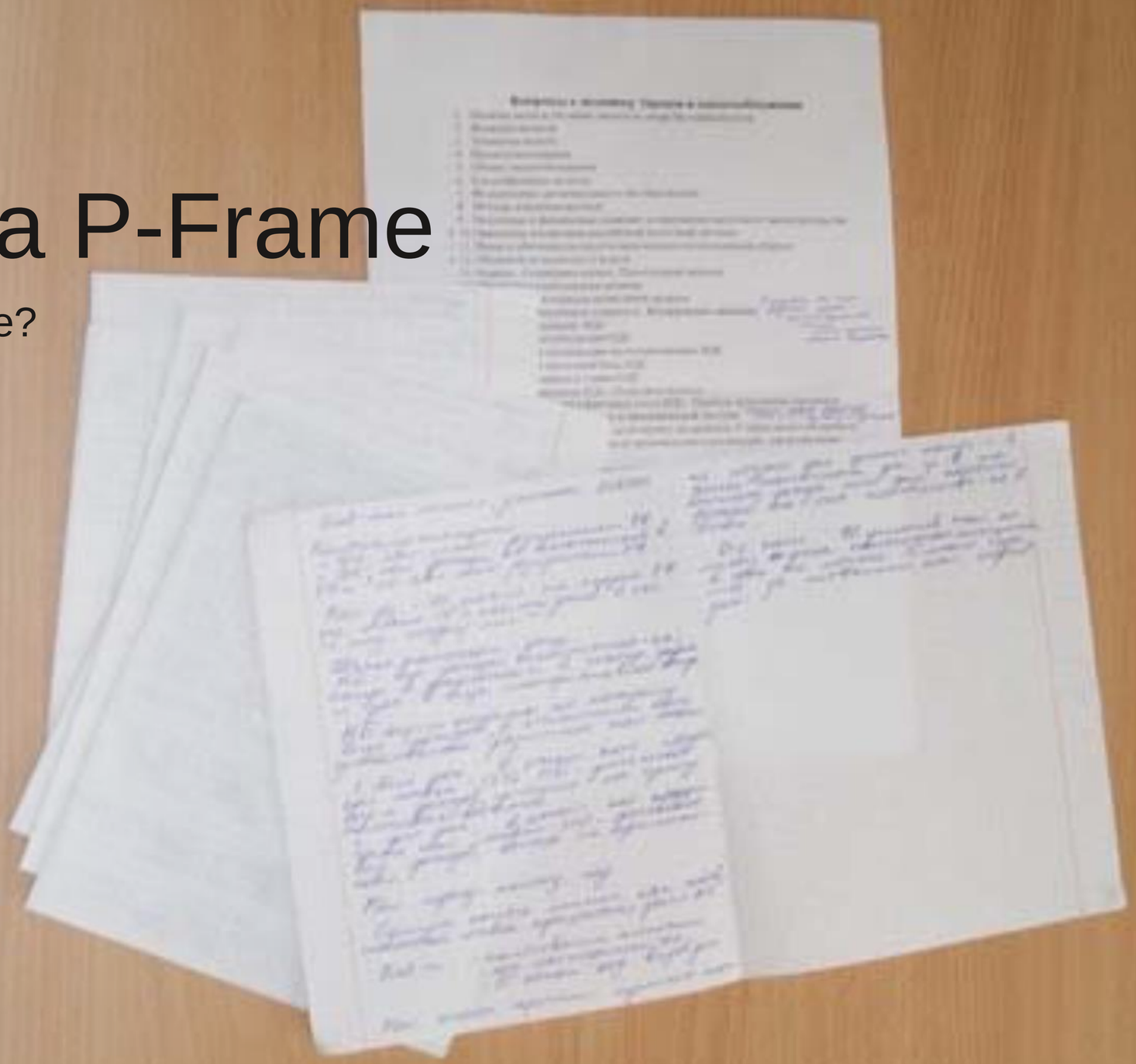
4

Абсолютно пустых
P-Frame нет

Глава 5:

Чистый zero-delta P-Frame

Давайте синтезируем пустой P-Frame?



В спеку!

Как собрать P-Frame
с пустой дельтой?

Приключения электроника

Кодируем non-IDR NAL-юниты

aka P-Frame

Состоят из хедера
и макроблоков 16 × 16

Чтобы записать макроблок, нужно

- I_PCM — выписать матрицу сэмплов в чистом виде
- Предсказать значение
  - Intra* — для I-макроблоков
    Зависит от соседей в том же кадре
  - Inter* — для P-, B-макроблоков
    Зависит от предыдущего и следующего кадров
- Посчитать разницу с требуемой картинкой
- Discrete Cosine Transform разницы
- Понизить точность коэффициентов DCT

Discrete Cosine Transform

Lena Söderberg. Playboy, November 1972

Из чего состоит Slice: Body

- (Repeated):
  - mb_skip_run — сколько макроблоков пропущено
  - (Repeated):
    - macroblock_layer — закодированный макроблок
    - Параметры для предсказания
    - Дельта с предсказанием

Макроблоки можно пропустить

- (Repeated):
  - `mb_skip_run` — сколько макроблоков пропущено
  - (Repeated):
    - macroblock_layer — закодированный макроблок
    - Параметры для предсказания
    - Дельта с предсказанием

Спека прочитана!

- Скипнем все макроблоки в слайсе
- Всего 16 вариантов пустого кадра — с разными frame_num

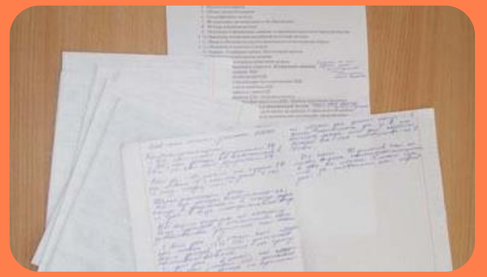
Приключения электроника, www.flaticon.com, Dog Treat

Пробуем.
Почём теперь типичный звонок?

- Encoding: 30 FPS, GOP 1800: 0,34 CPU/call
- Encoding: 1 FPS, GOP 60: 0,023 CPU/call
- Replaying: 30 FPS, GOP 1800: 0,056 CPU/call
- Forging: 30 FPS, GOP 1800: 0,033 CPU/call

Результат главы 5

0.34 -> 0.033 CPU/Call



1

Мы сэкономили CPU

2

Энкодим только кадры с изменениями

3

Без изменений — заготовим заранее

Оценить доклад и забрать слайды

- Видео меню переговорок Яндекс Телемоста
- Формат H.264 в SDP
- Real-time Transport Protocol (RTP)
- Синтаксис H.264
  - NAL Unit
  - SPS, PPS, Group Of Pictures
  - Slice
- Производительность

Дмитрий Некрылов
TG: @topright007

Спасибо за внимание

Дмитрий Некрылов
Медиатехнологии Кинопоиска