



VK Play

Modern codecs and ultra-low latency streaming



Chernikov Kirill

Head of the VK Play Cloud
Client Development Team



About team and project

In cloud gaming
development since
2013

3-6 members
in development team

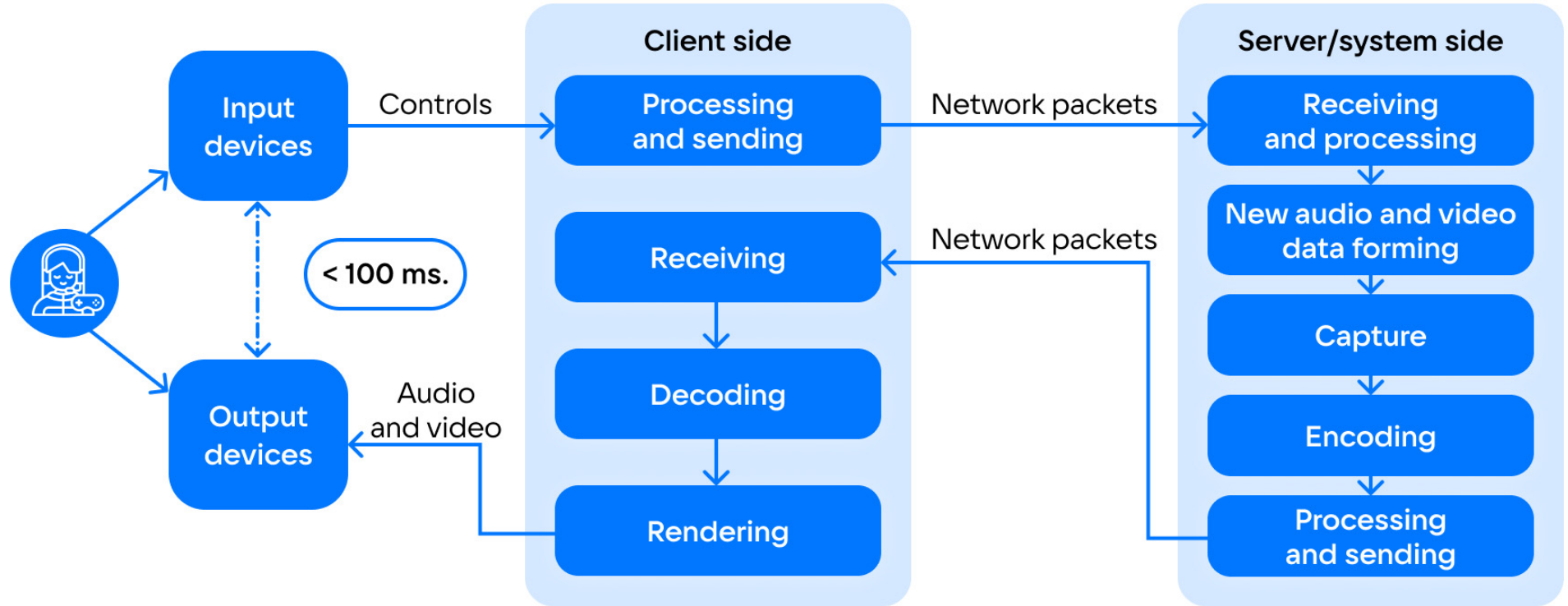


What is ultra-low latency streaming?

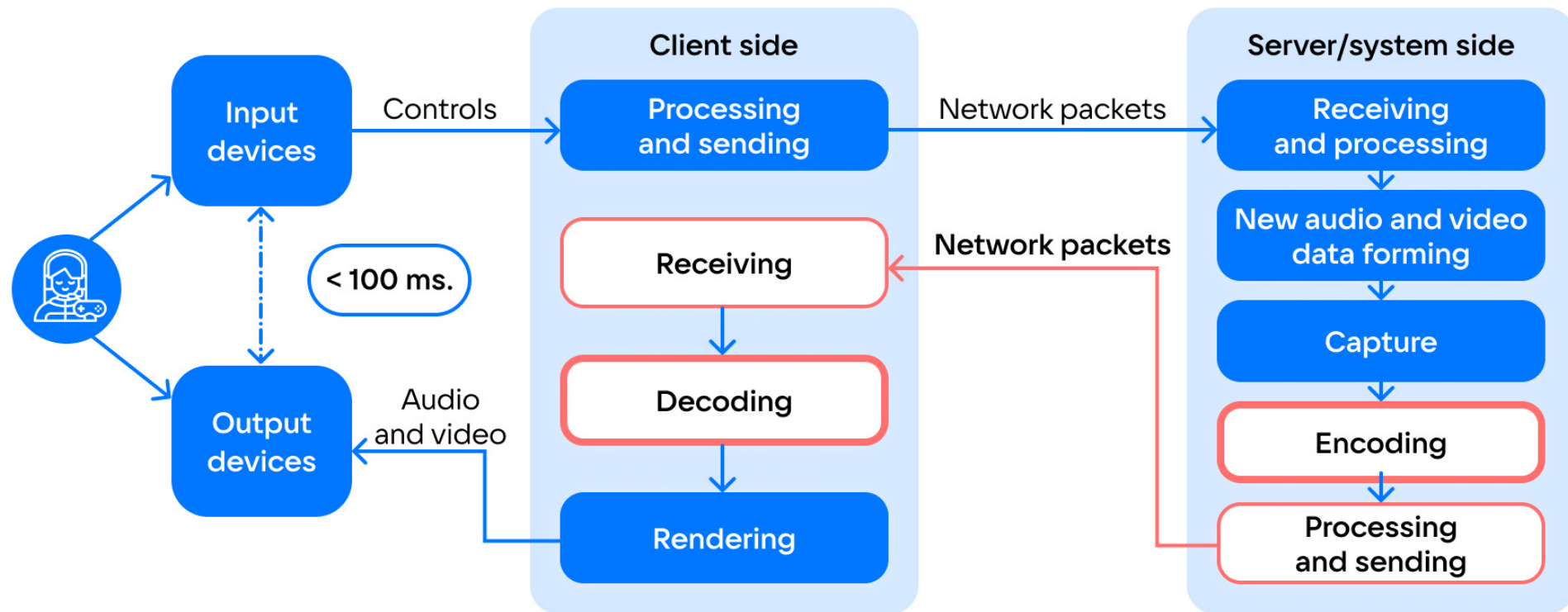
Application type	Latency
Downloading media content	Any
Video on demand (VOD)	1-10 sec.
Video calls	< 500 ms.
Real-time control applications (ultra-low latency applications)	< 100 ms. (50-70 ms.)

Ultra-low latency application scheme

Cloud gaming example



Codec and key stages



Encoder requirements. Encoding

1 The best encoding speed (the fastest uniform speed)

- Search for low latency / ultra-low latency / zerolatency / ultrafast preset
- The required fps gives needed upper bound estimate

```
if( !strcasecmp( preset, "ultrafast" ) )
{
    param->i_frame_reference = 1;
    param->i_scenecut_threshold = 0;
    param->b_deblocking_filter = 0;
    param->b_cabac = 0;
    param->i_bframe = 0;
    param->analyse.intra = 0;
    param->analyse.inter = 0;
    param->analyse.b_transform_8x8 = 0;
    param->analyse.i_me_method = X264_ME_DIA;
    param->analyse.i_subpel_refine = 0;
    param->rc.i_aq_mode = 0;
    param->analyse.b_mixed_references = 0;
    param->analyse.i_trellis = 0;
    param->i_bframe_adaptive = X264_B_ADAPT_NONE;
    param->rc.b_mb_tree = 0;
    param->analyse.i_weighted_pred = X264_WEIGHTP_NONE;
    param->analyse.b_weighted_bipred = 0;
    param->rc.i_lookahead = 0;
}
```

Encoder requirements. Encoding

- 2 The ability to change encoder parameters «on the fly» (bitrate)
- 3 «Independence» of the encoder (no impact on the managed system)



Encoder requirements. Network

1 The best compression ratio

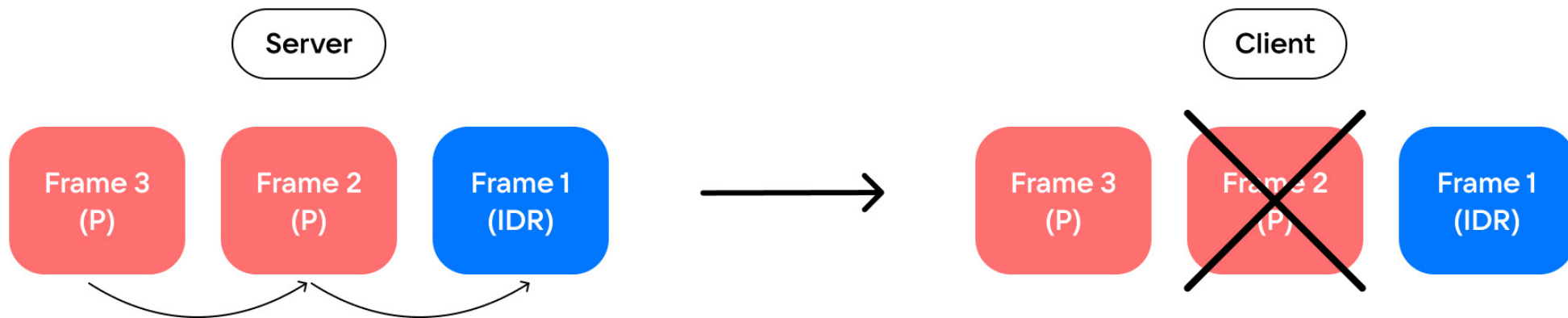
- H264 → H265 → H266
- VP8 → VP9
- AV1

2 Maximally uniform stream (rate control problem)

- Need to solve static image problem
- CBR or VBR
- $\text{VBV / HRD buffer size} = \text{Bitrate} / \text{FPS}$ — no additional latency

3 How fast encoding parameters changes apply

Encoder requirements. Network



4 Methods for restoring the «broken» stream

- IDR
- Intra Refresh
- Invalidate Reference + Reference frames / Big DPB buffer size

See for details:
[VideoTech 2022](#)



Decoder requirements

- 1 The best decoding speed (the fastest uniform speed)
 - The required fps gives needed upper bound estimate
- 2 The ability to work with a «broken» stream =

«decoder adaptation»



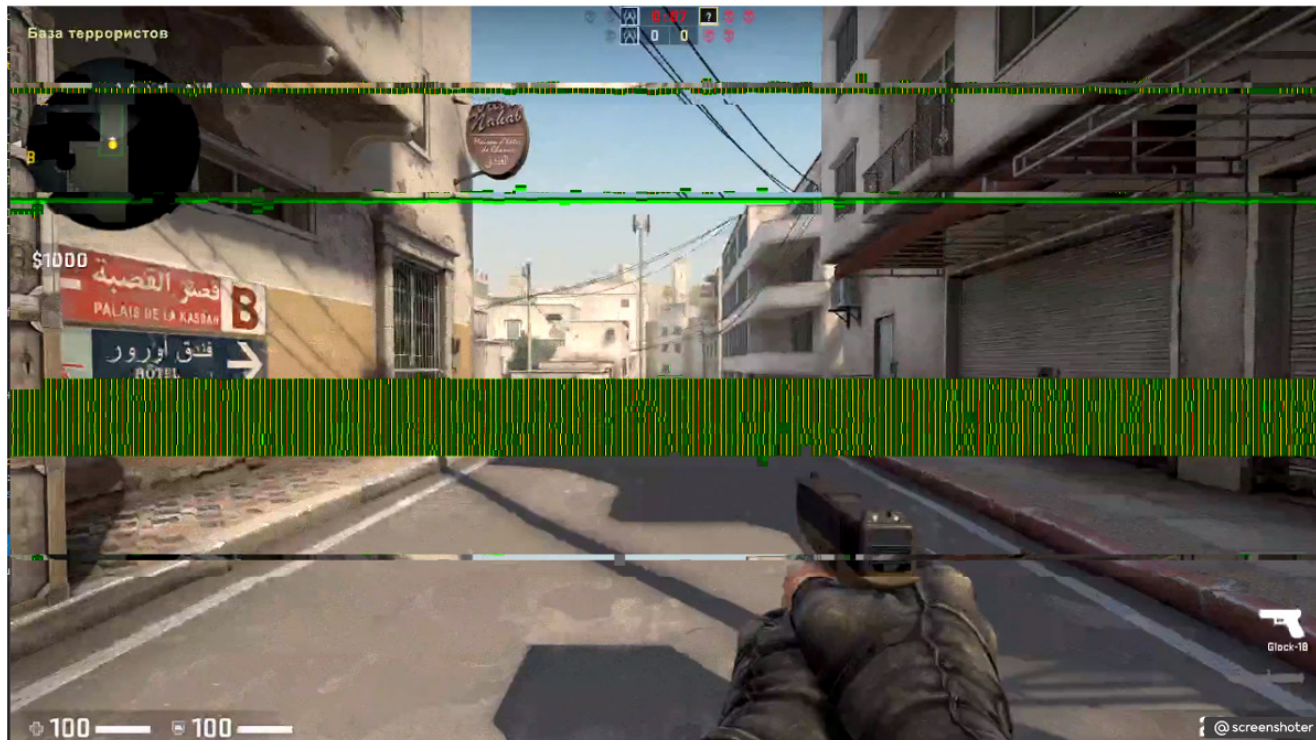
«Broken» stream problems

- 1 Decoder failure (reinitialization, image freeze)
- 2 Decoding errors (image freeze)
- 3 White/grey/green/black stripe or screen (very bad quality)
- 4 "Flashbacks" (very bad quality)
- 5 Blocks artifacts (bad or medium quality)

Grey screen example



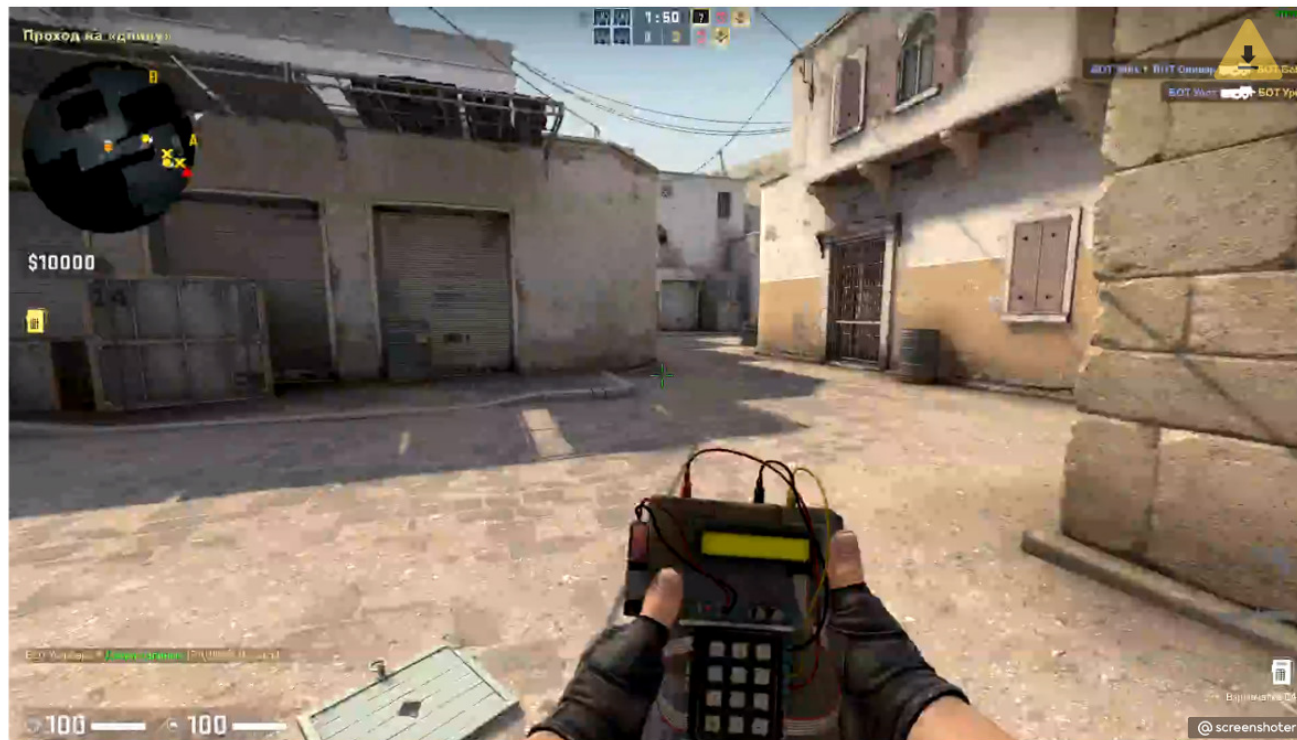
Green stripe example



Big blocks artifacts example



Blocks artifacts example



Video sample



Нажмите ALT+ENTER, чтобы переключиться в полноэкранный режим. Нажмите CTRL+ALT, чтобы освободить курсор мыши. Версия 2.0 849.1341142 Сессия: 39286691

Packet losses modeling

Uniform losses rate = 30%

Losses sequence rate = 0,1% (base = 100 packets)

RTT = 30 ms.

25 Mbit/s., 60 FPS, 1920x1080

IDR, Invalidate Reference, Retransmit request

Client

Server

~10-25% unassembled frames

No «broken» stream to decoder



H264 Intel QSV decoder



H264 Nvidia DXVA2 decoder



H264 AMD UVD decoder

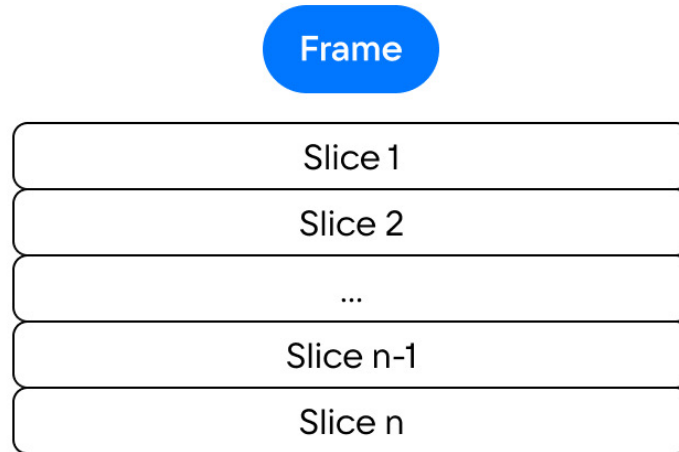


H265 AMD UVD decoder

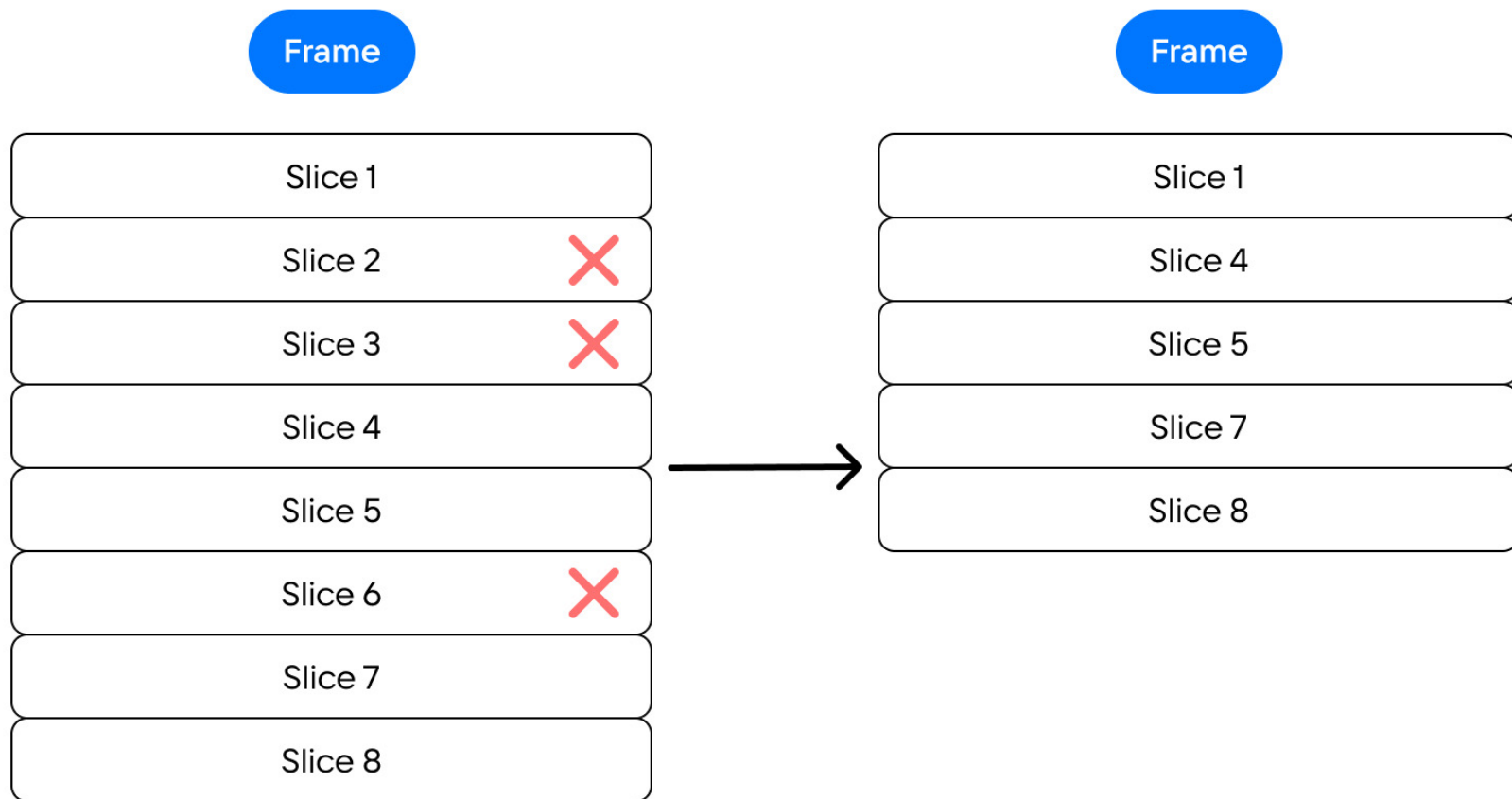


Incomplete and gap frames

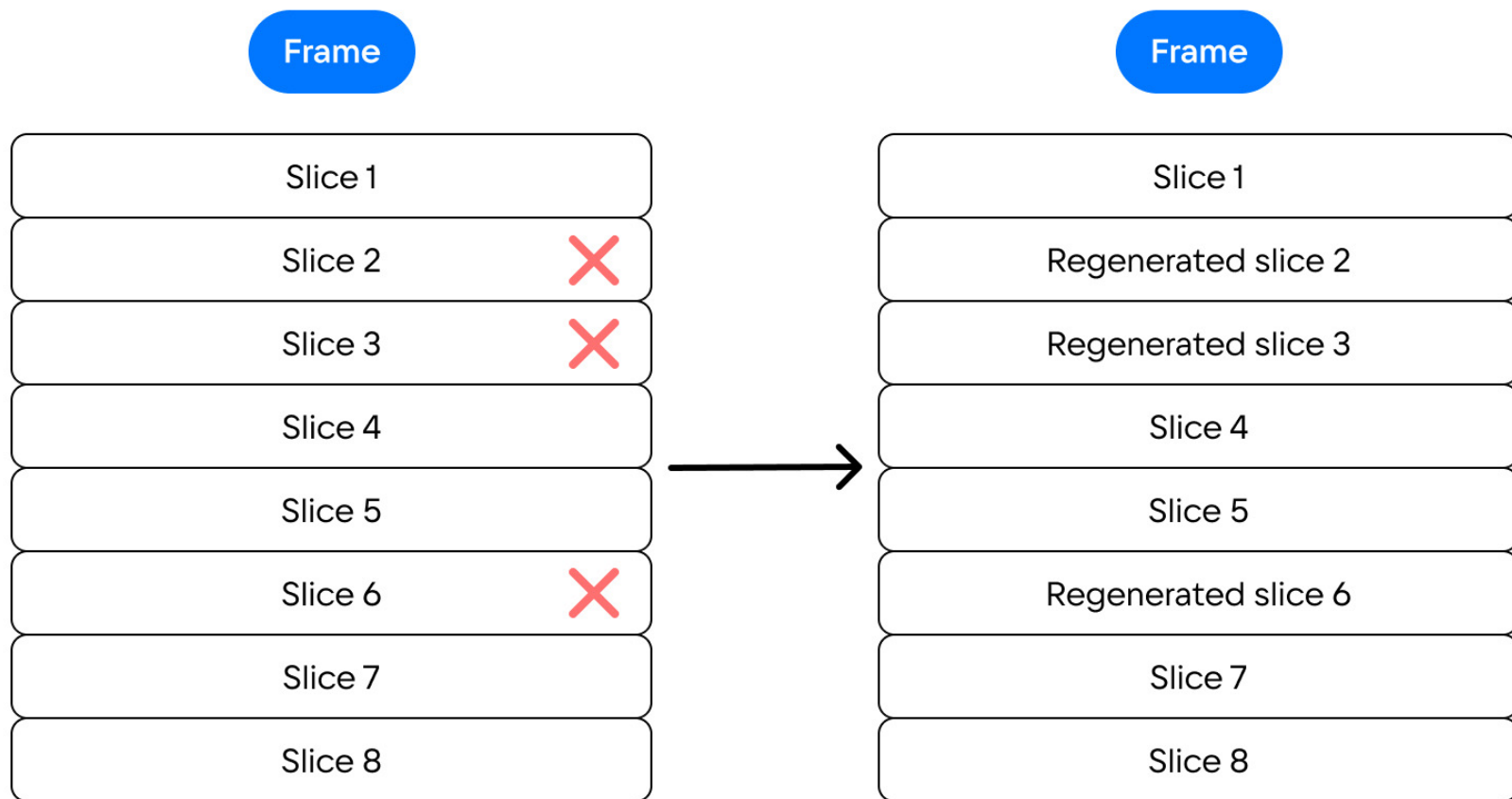
- 1 Is it possible to give incomplete frames to the decoder?
- 2 Is it possible to regenerate missing data?
- 3 Has frame independent elements?



Incomplete frames policy



Recovery frames policy. Slice

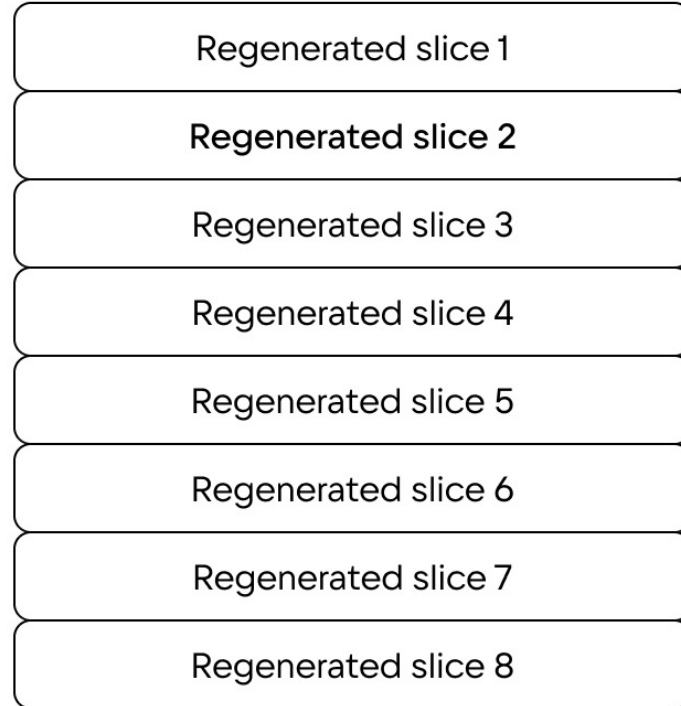


Recovery frames policy. Gap frame

Gap Frame

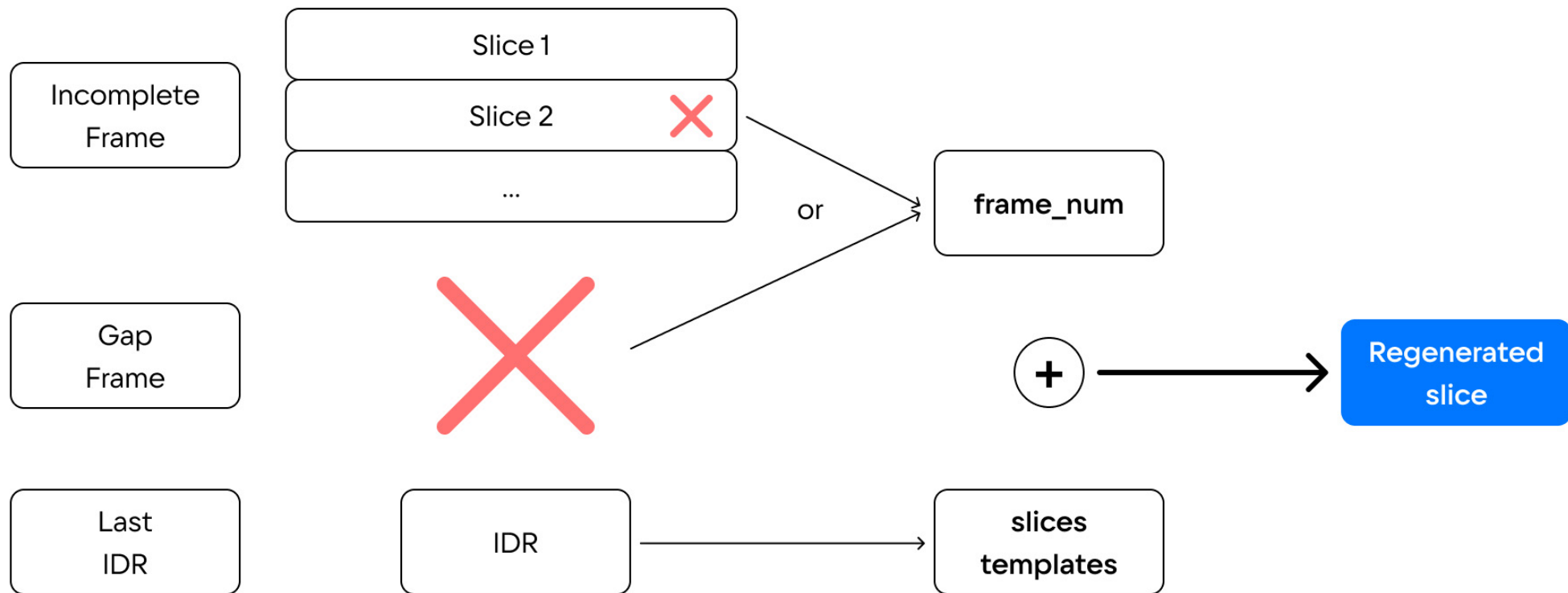


P-Frame always



Regenerated slice (H264 only)

- P-slice always



H264 Nvidia DXVA2 decoder



H264 Nvidia DXVA2 decoder. Recovery frames policy



H264 AMD UVD decoder. Recovery frames policy



Нажмите ALT+ENTER, чтобы переключиться в полноэкранный режим. Нажмите CTRL+ALT, чтобы освободить курсор мыши. Версия 2.0.050.1341142 Сессия 3928811

Positive impact of policies

- 1 H264 and H265 DXVA2 decoder on NVIDIA graphics (Windows)
- 2 Some H264 and H265 Huawei TV hardware decoders (Android)
- 3 Some H264 Amlogic devices hardware decoders (Android)
- 4 Some H264 Mediatek devices hardware decoders (Android)

What we have in the end?

- 1 Multivariate optimization problem
- 2 Conflicting metrics and stream parameters
- 3 «Broken» stream factor
 - Standardization is needed
 - Different codec settings → different «broken» stream problems on the same decoder
 - Use more slices with incomplete and recovery frame policy





VK Play

Thanks
for your attention!



Chernikov Kirill

Head of the VK Play Cloud
Client Development Team
k.chernikov@corp.mail.ru

