

The *xvc* video codec

draft-samuelsson-netvc-xvc-00

Jonatan Samuelsson, Per Hermansson (Divideon)

IETF 101, London, March 2018



Outline

- ◆ What is xvc?
- ◆ Design philosophy
- ◆ Technology in xvc
- ◆ Restriction flags
- ◆ Version handling
- ◆ xvc in WebRTC
- ◆ Results
- ◆ xvc as candidate for NETVC



What is xvc?

- ◆ A next-generation video codec, first released in September 2017
- ◆ Higher compression performance than all other codecs
- ◆ Developed by Divideon
- ◆ Source code publicly available
- ◆ Commercial license that covers both software and patents
- ◆ Definition of a royalty-free baseline profile under investigation
- ◆ Well defined framework for handling evolution of the codec
- ◆ Efficient decoder implementation: [demo page](#)

The xvc codec at 120 kbps



The h.264 codec at 120 kbps

Design philosophy

- ◆ The xvc codec has been designed to use the best available technology
- ◆ Technology is avoided and/or replaced only if it has been determined that the technology cannot be used under the xvc license

Low performance

High performance

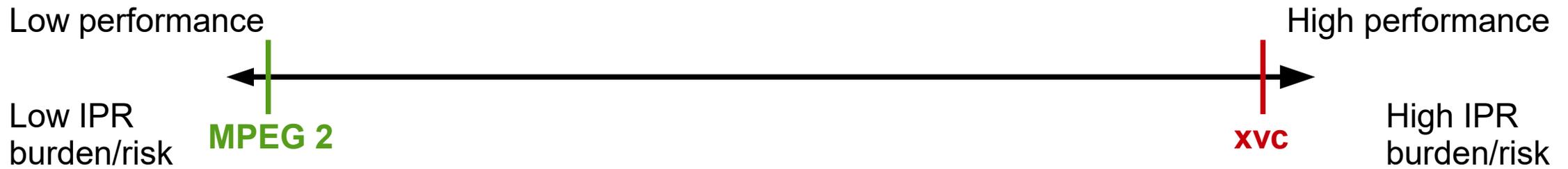
Low IPR
burden/risk

High IPR
burden/risk



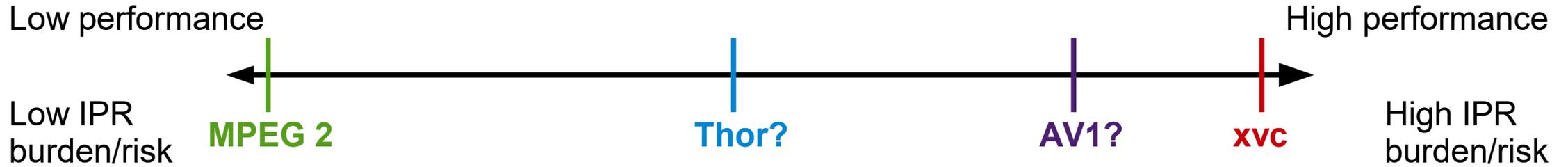
Design philosophy

- ◆ The xvc codec has been designed to use the best available technology
- ◆ Technology is avoided and/or replaced only if it has been determined that the technology cannot be used under the xvc license



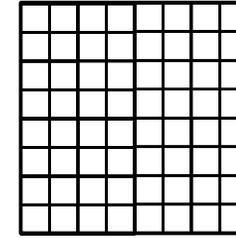
Design philosophy

- ◆ The xvc codec has been designed to use the best available technology
- ◆ Technology is avoided and/or replaced only if it has been determined that the technology cannot be used under the xvc license

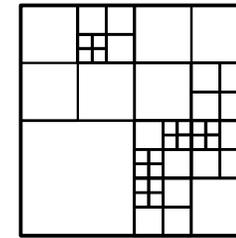


Technology in xvc

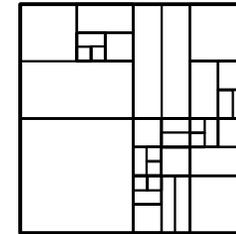
- ◆ Version 1.0 of xvc (Sept. 2017) included 62 coding tools
- ◆ Version 2.0 of xvc to be released during 2018
- ◆ Currently 14 new tools targeting version 2.0, involving:
 - More intra directions and intra predictors
 - Cross component prediction
 - Adaptive full-pel motion vectors
 - Affine motion prediction
 - High precision motion vectors
 - Local illumination compensation
 - RDO based transform selection
 - Advanced coefficient coding



AVC/H.264



HEVC/H.265



XVC

Restriction flags

- ◆ Each coding tool can be turned off by control information in the bitstream
- ◆ The restriction flags are examined during run time (for each segment of the coded video)
- ◆ The decoder uses a fallback solution when a tool is disabled
- ◆ There are currently 76 restriction flags in xvc
- ◆ Average bitrate cost for disabling a tool is below 1%

```

77 void IntraPrediction::Predict(IntraMode intra_mode, const CodingUnit &cu,
78                             YuvComponent comp, const RefState &ref_state,
79                             const YuvPicture &rec_pic,
80                             SampleBuffer *output_buffer) {
81     const int width = cu.GetWidth(comp);
82     const int height = cu.GetHeight(comp);
83     Sample *out_ptr = output_buffer->GetDataPtr();
84     const ptrdiff_t out_stride = output_buffer->GetStride();
85     const Sample *ref_samples = &ref_state.ref_samples[0];
86     if (Restrictions::Get().disable_intra_planar &&
87         intra_mode == IntraMode::kPlanar) {
88         intra_mode = IntraMode::kDc;
89     }
90     if (util::IsLuma(comp)) { ... }
95     const bool post_filter = util::IsLuma(comp) && width <= 16 && height <= 16;
96     switch (intra_mode) {
97     case IntraMode::kPlanar:
98         PlanarPred(width, height, ref_samples, kRefSampleStride_,
99                 out_ptr, out_stride);
100     break;

```

Version handling

- ♦ All xvc bitstreams indicate major xvc version and minor xvc version
- ♦ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ♦ Minor version corresponds to backward compatible changes i.e. removal of tools
- ♦ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)

Two aspects to take into account:

Bitstream upgrade

- typically centralized and only once per bitstream
- can be much more lightweight than re-encoding
- can give better performance due to improvements of the codec

Client upgrade

- new xvc decoder pushed out for new major version
- two versions of xvc decoders supported during upgrade period

Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



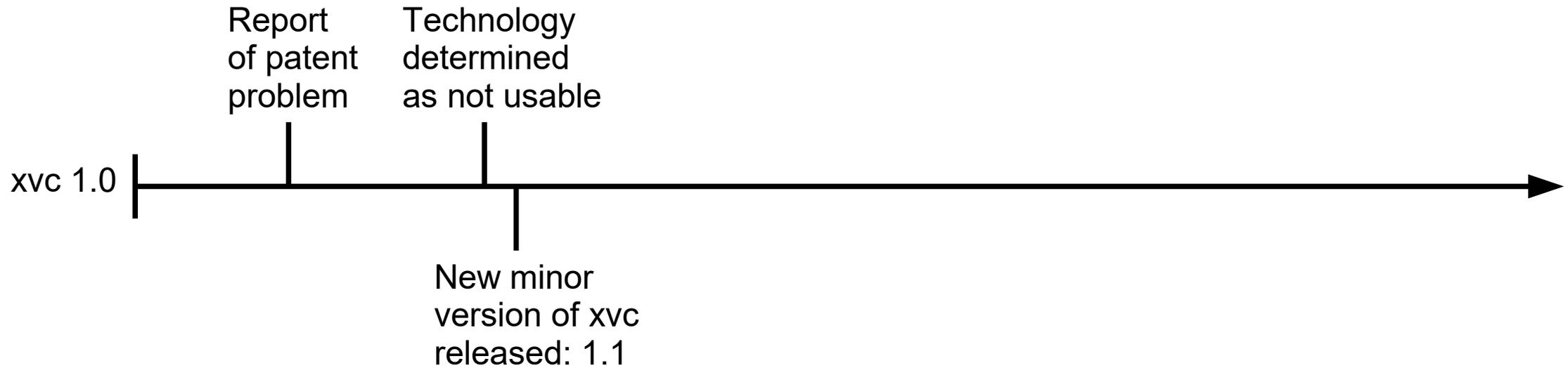
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



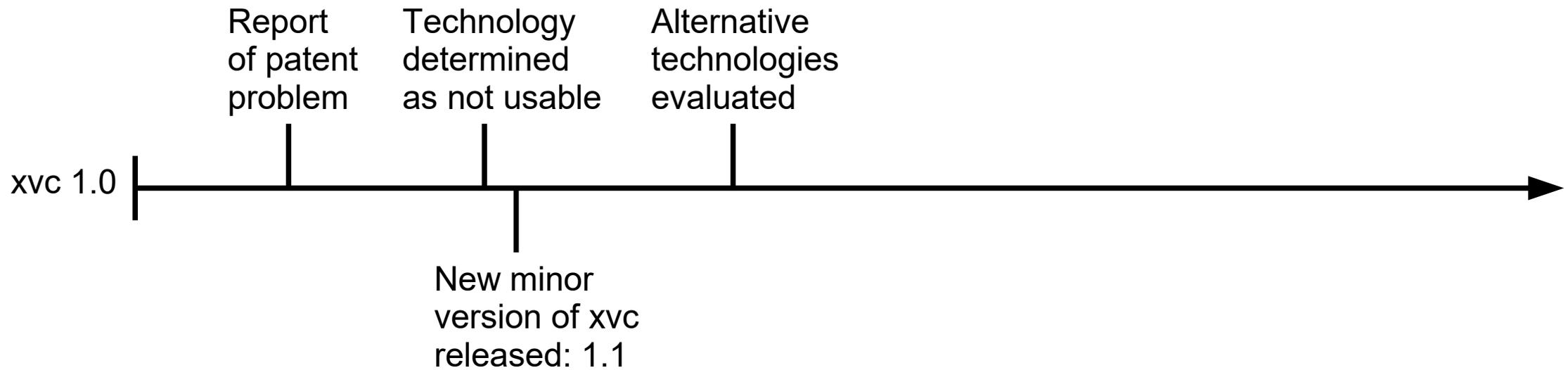
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



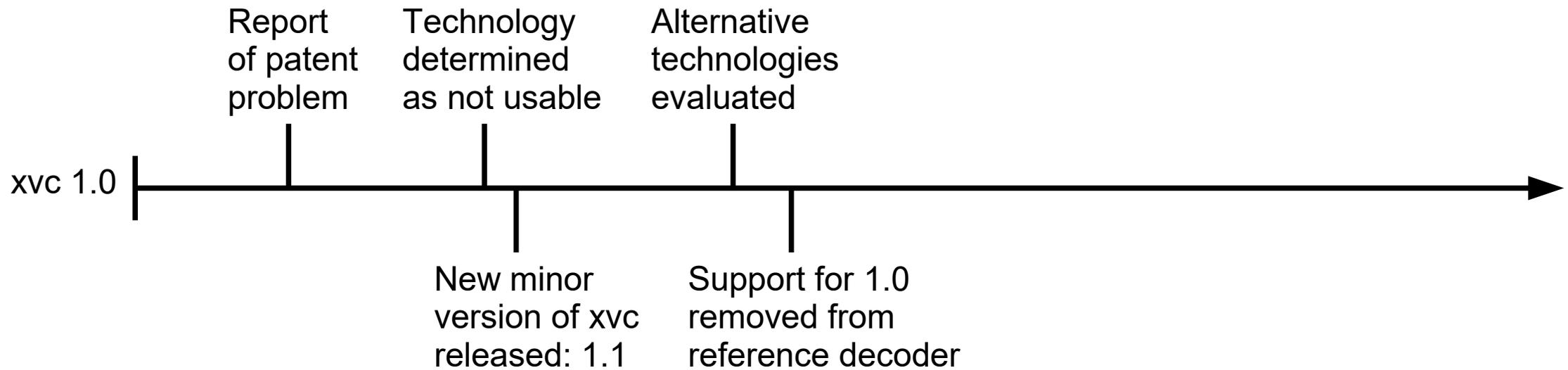
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



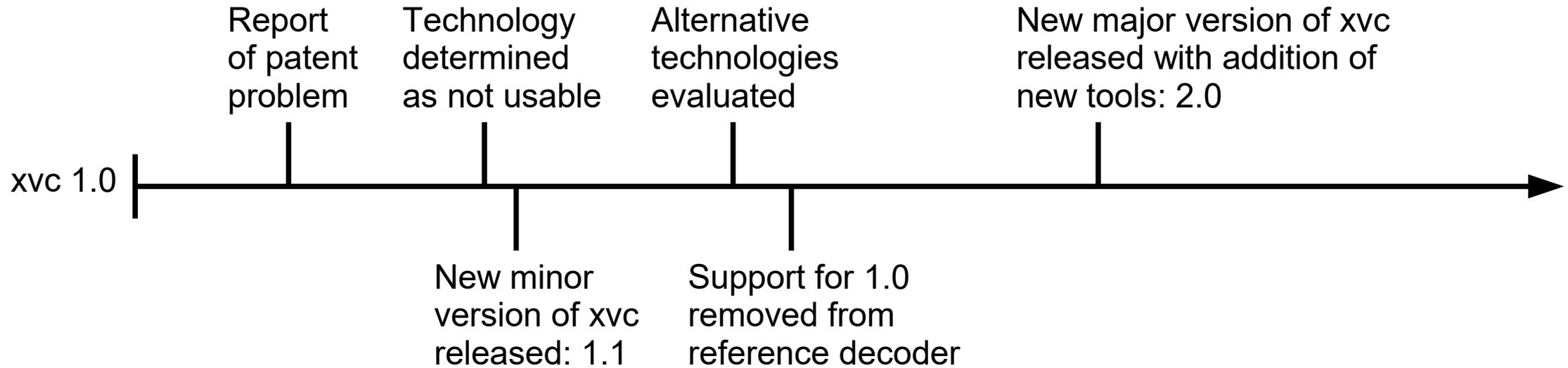
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



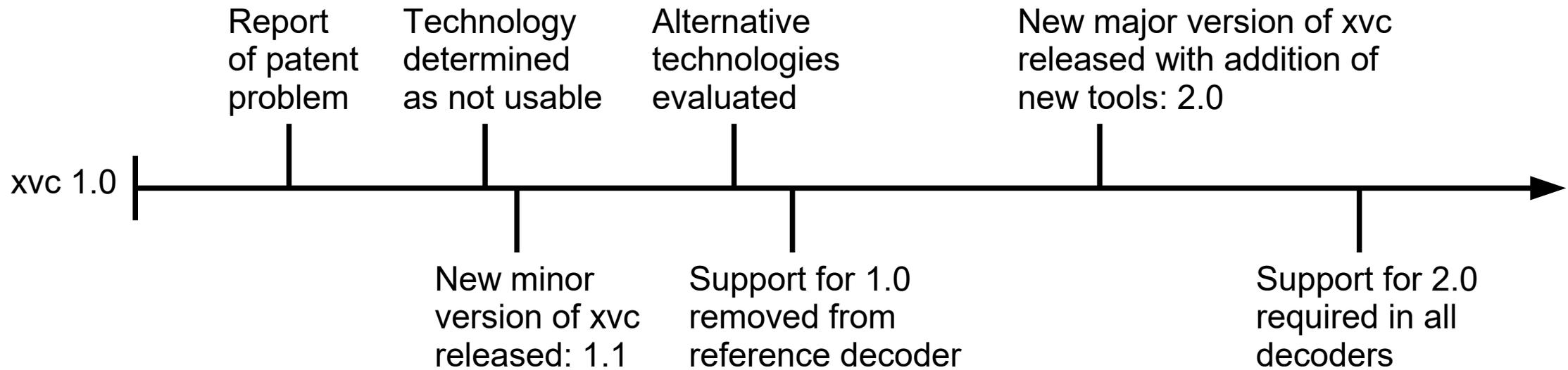
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



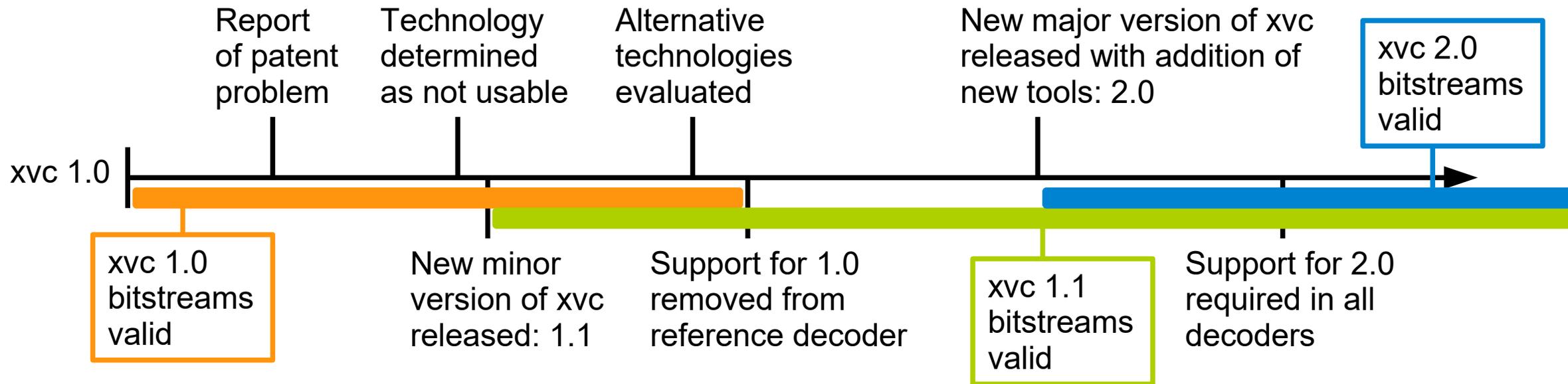
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



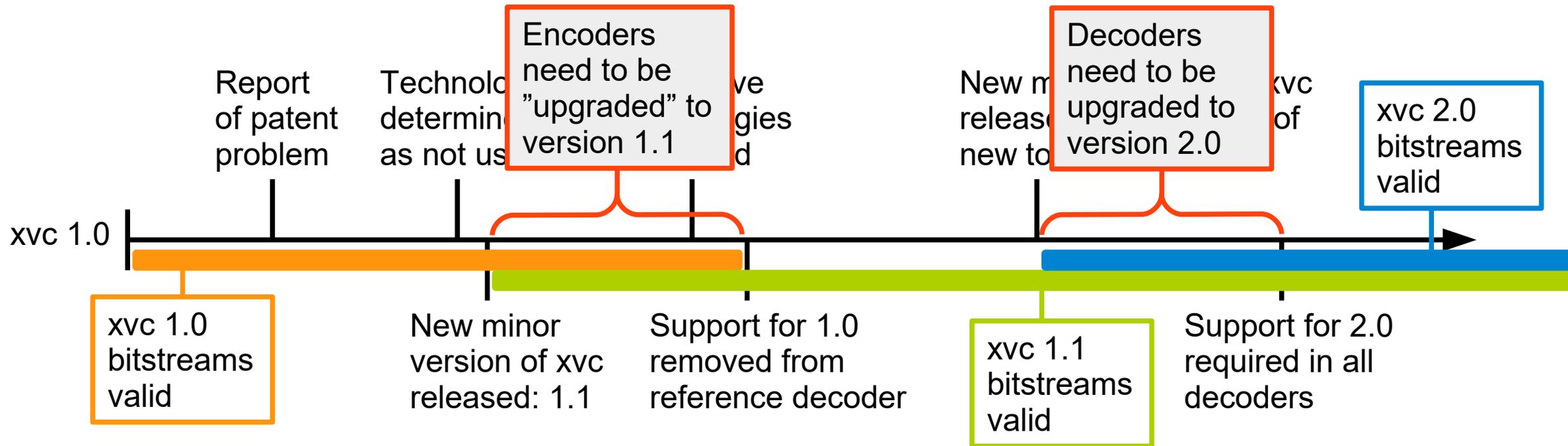
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



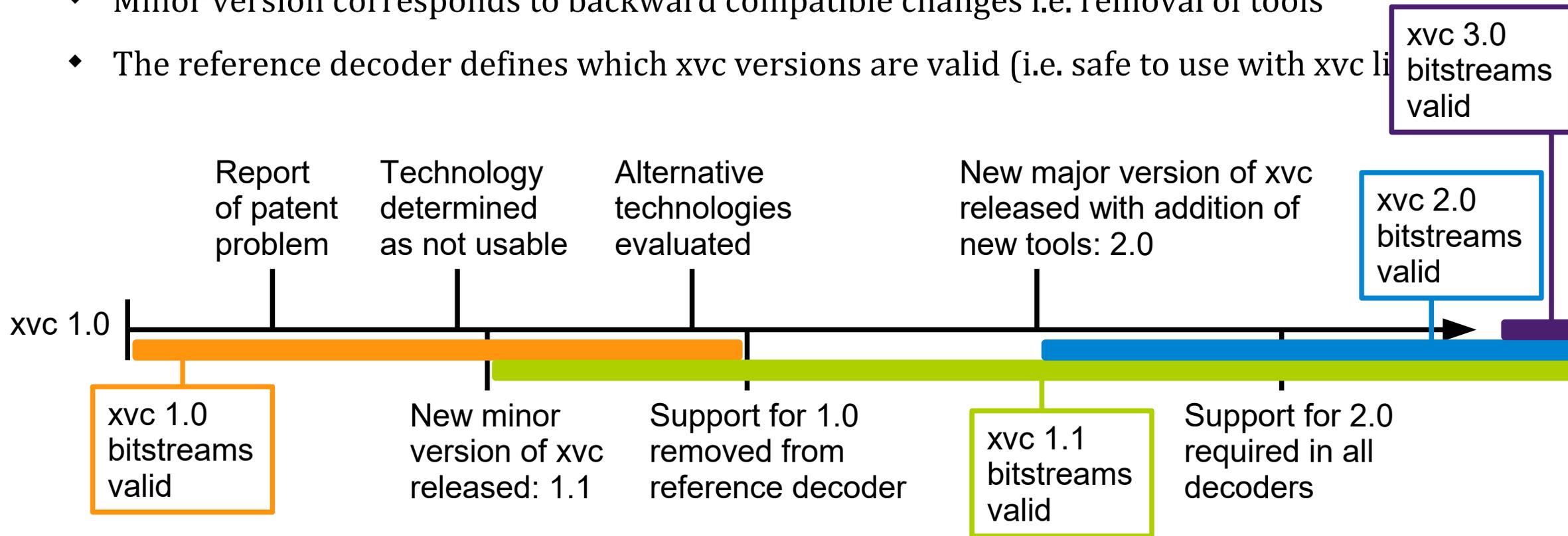
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc license)



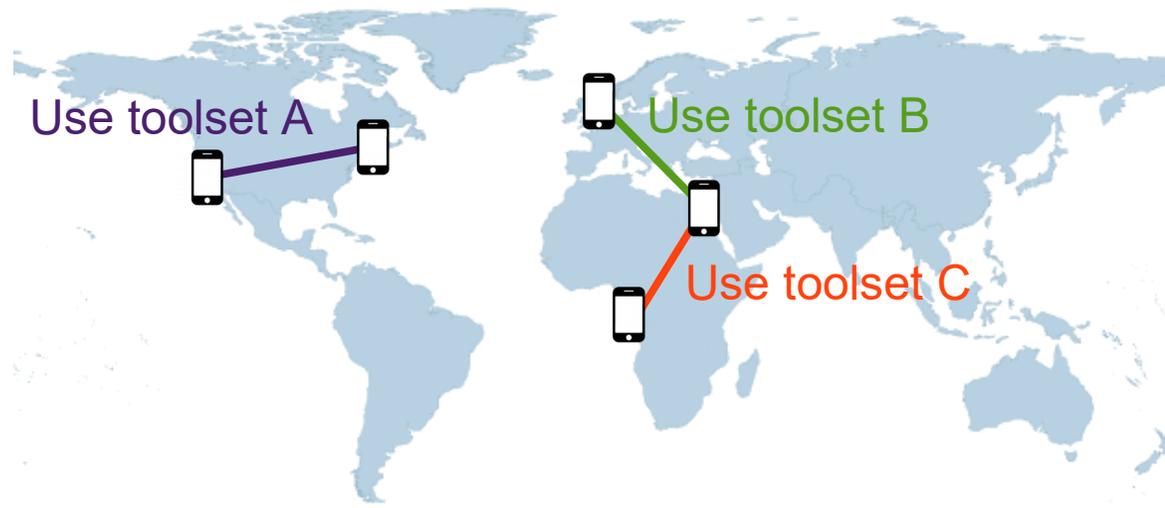
Version handling

- ◆ All xvc bitstreams indicate major xvc version and minor xvc version
- ◆ Major version corresponds to non-backward compatible changes i.e. addition of new tools
- ◆ Minor version corresponds to backward compatible changes i.e. removal of tools
- ◆ The reference decoder defines which xvc versions are valid (i.e. safe to use with xvc li



xvc in WebRTC

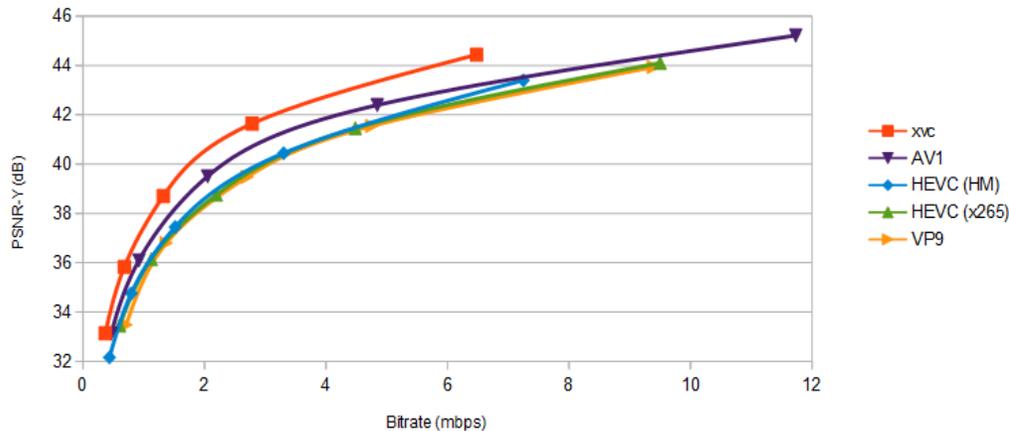
- ◆ The xvc codec can be used in low delay configuration, without picture reordering
- ◆ In realtime communication applications it is possible to use the restriction flags to negotiate on different sets of tools for each session
- ◆ Negotiation may be related to resources, complexity, parallelism etc. but it would also be possible to take IPR status into account in order to make the session royalty-free
- ◆ It would even be possible to adjust the set of tools based on the location of the participants



Results

- ◆ The xvc codec has been tested using the test conditions from draft-ietf-netvc-testing-06
- ◆ The AreWeCompressedYet? framework has been used
- ◆ All results available at awcy.divideon.com

Netflix_RollerCoaster_1280x720_60fps_8bit_420_60f.y4m



Single pass Random-Access xvc relative to HM:

	PSNR	PSNR Cb	PSNR Cr	PSNR HVS	SSIM	MS SSIM
1080p	-16.8	-29.9	-28.9	-15.0	-17.9	-16.8
1080psc	-13.7	-44.5	-40.4	-15.4	-16.7	-17.0
720p	-20.8	-30.0	-32.6	-20.1	-23.8	-22.7
360p	-26.1	-24.7	-28.8	-26.4	-30.2	-29.6
Average	-19.5	-30.7	-31.3	-19.1	-22.0	-21.2

Single pass Random-Access xvc relative to AV1:

	PSNR	PSNR Cb	PSNR Cr	PSNR HVS	SSIM	MS SSIM
720p	-13.3	-0.8	-4.4	-16.4	-20.2	-20.5
360p	-19.7	-9.6	-3.4	-22.5	-23.0	-26.1
Average	-16.5	-5.2	-3.9	-19.4	-21.6	-23.3

Multi-pass Random-Access xvc relative to AV1:

	PSNR	PSNR Cb	PSNR Cr	PSNR HVS	SSIM	MS SSIM
1080p	-6.0	-4.5	-3.2	-5.6	-10.9	-9.7
1080psc	8.3	18.5	15.8	5.9	6.2	3.9
720p	-0.5	0.4	4.8	-1.4	-6.0	-5.5
360p	-15.9	-6.2	11.2	-19.9	-19.0	-21.2
Average	-5.1	-0.7	4.6	-6.4	-9.4	-9.6

xvc as candidate for NETVC

- ◆ The xvc codec is brought as a candidate proposal for NETVC
- ◆ We believe that xvc is well positioned to meet the objectives:
 1. Is competitive (in the sense of having comparable or better performance) with current video codecs in widespread use.
 2. Is optimized for use in interactive web applications.
 3. Is viewed as having IPR licensing terms that allow it to be widely implemented and deployed.



Thank you!

