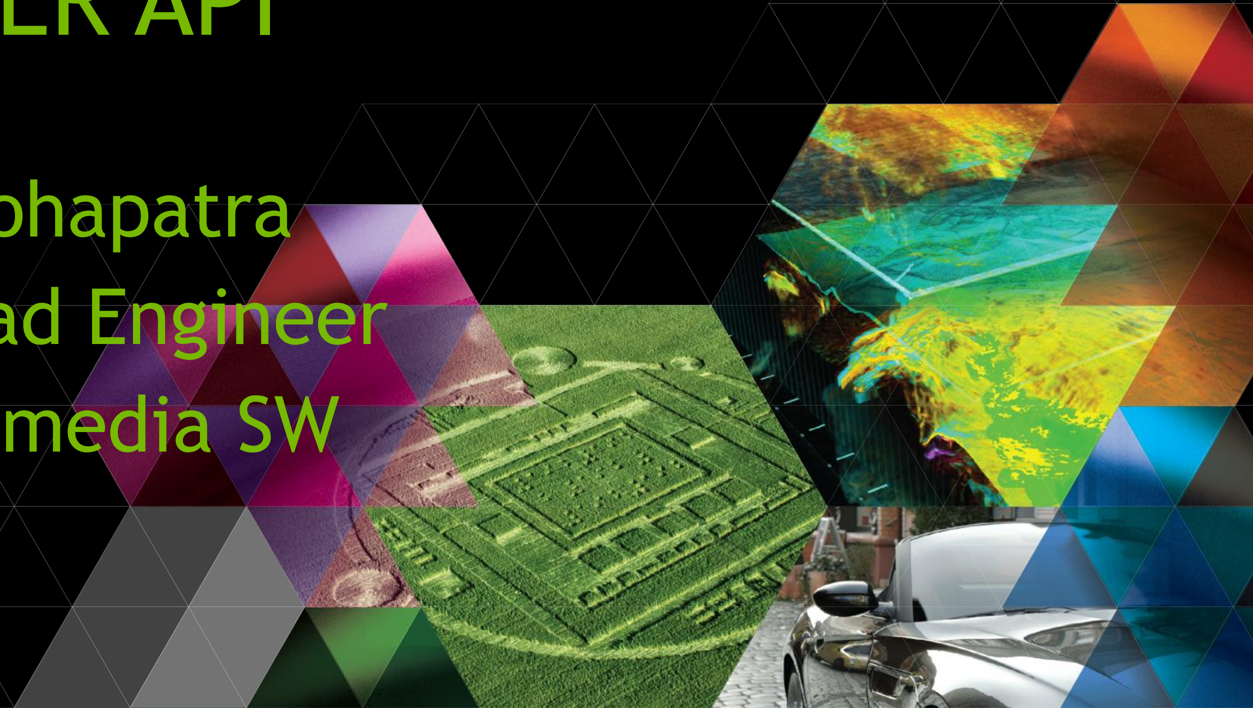
GPU TECHNOLOGY CONFERENCE

DETAIL OVERVIEW OF NVENC ENCODER API

Swagat Mohapatra
Senior Lead Engineer
GPU Multimedia SW



AGENDA

- Introduction to NVENC SDK
- Detailed Overview of NVENC API
- Advanced Topics
  - Rate Control Modes
  - Low Latency Encoding

BENEFITS OF HW BASED ENCODER

- Low power
- Low latency
- High performance
- Ease of Programming

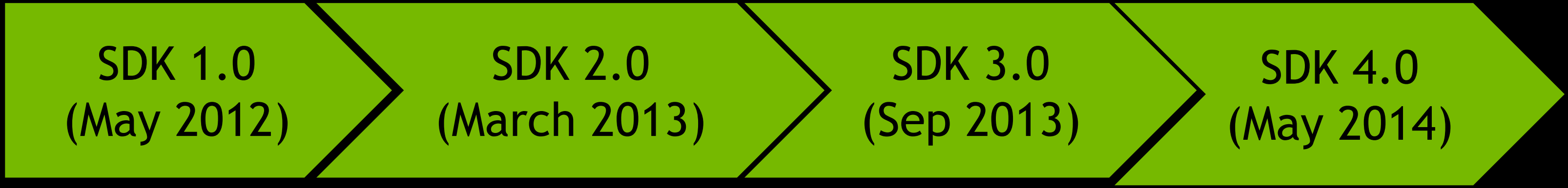
NVENC VIDEO ENCODING SOLUTIONS

- Fixed Function Hardware (NVENC)
  - Entire encode pipeline implemented in hardware
    - ME, intra-prediction, mode decision, VLE
  - High performance, low power
  - Kepler +
  - Proprietary software API(NVENC SDK)
    - Windows (NVENC-DirectX interop, NVENC-CUDA interop)
    - Linux (NVENC-CUDA interop)
  - Can work in hybrid mode with ME on CUDA

NVENC SDK

- Available on NVIDIA developer zone
  - https://developer.nvidia.com/nvidia-video-codec-sdk
- .DLL/.so, interface header, documentation, sample apps
- Unified API for Windows and Linux
- Works on x86/x64

NVENC SDK



SDK 1.0
(May 2012)

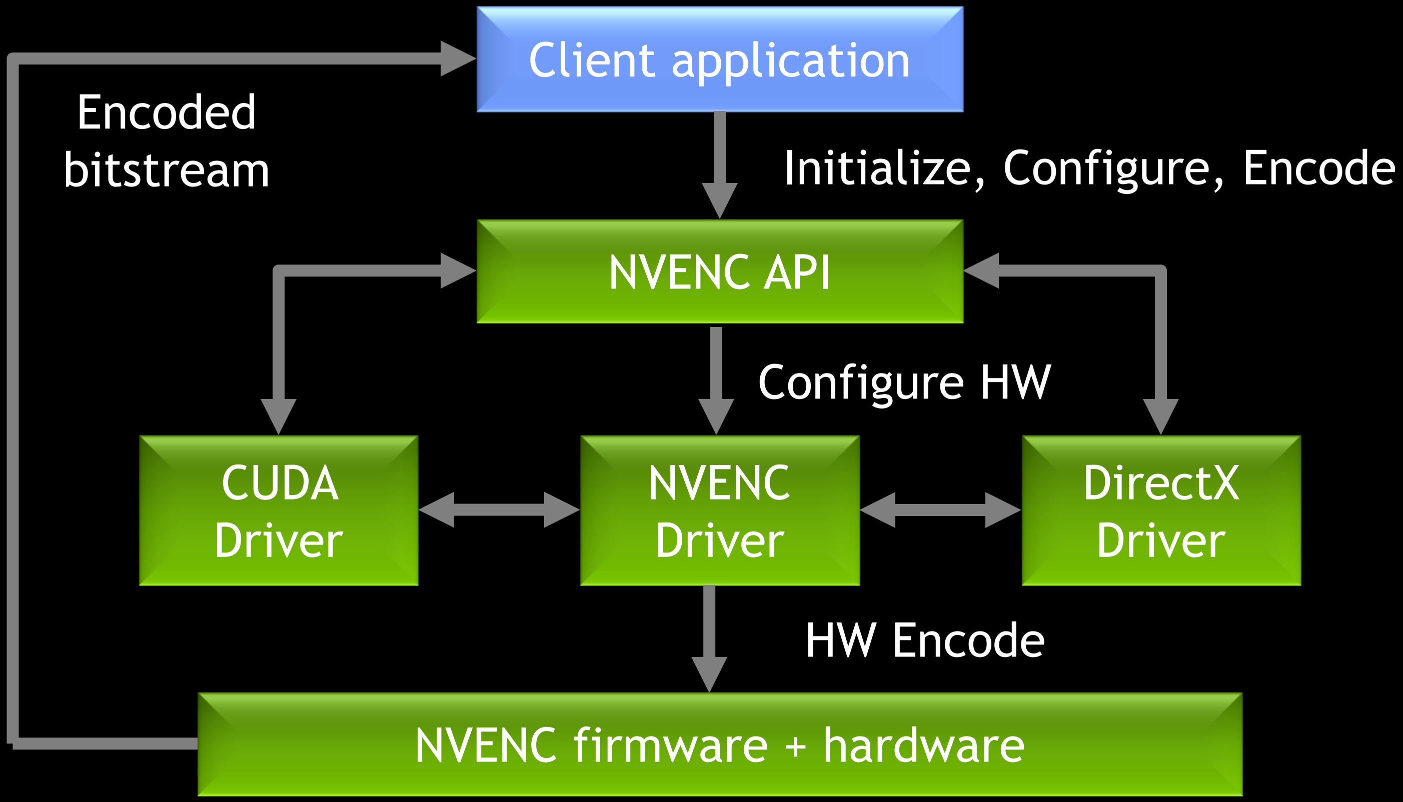
SDK 2.0
(March 2013)

SDK 3.0
(Sep 2013)

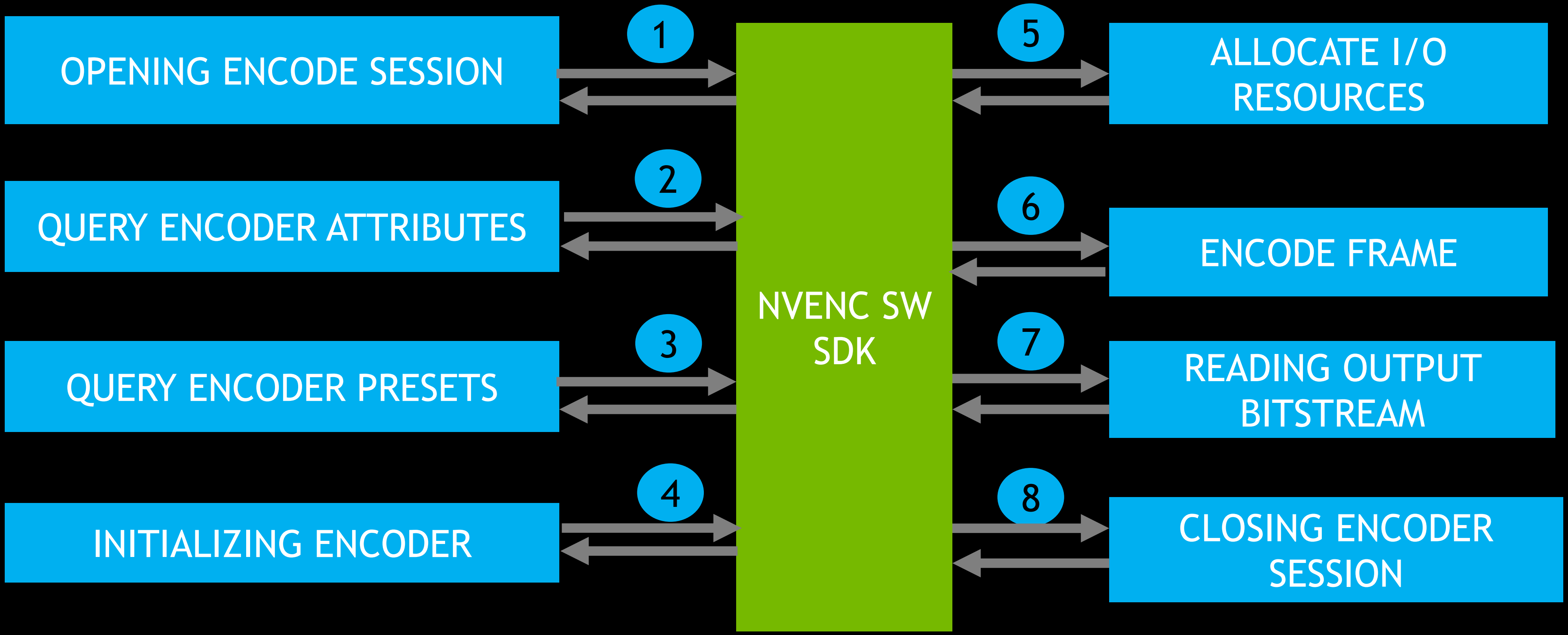
SDK 4.0
(May 2014)

VERSION	
SDK 1.0	Windows Support Only, Transcoding Support
SDK 2.0	Linux Support, Low latency Encoder support
SDK 3.0	Low latency encoding improvements, Reconfigure API
SDK 4.0	Maxwell Support, yuv444 , lossless

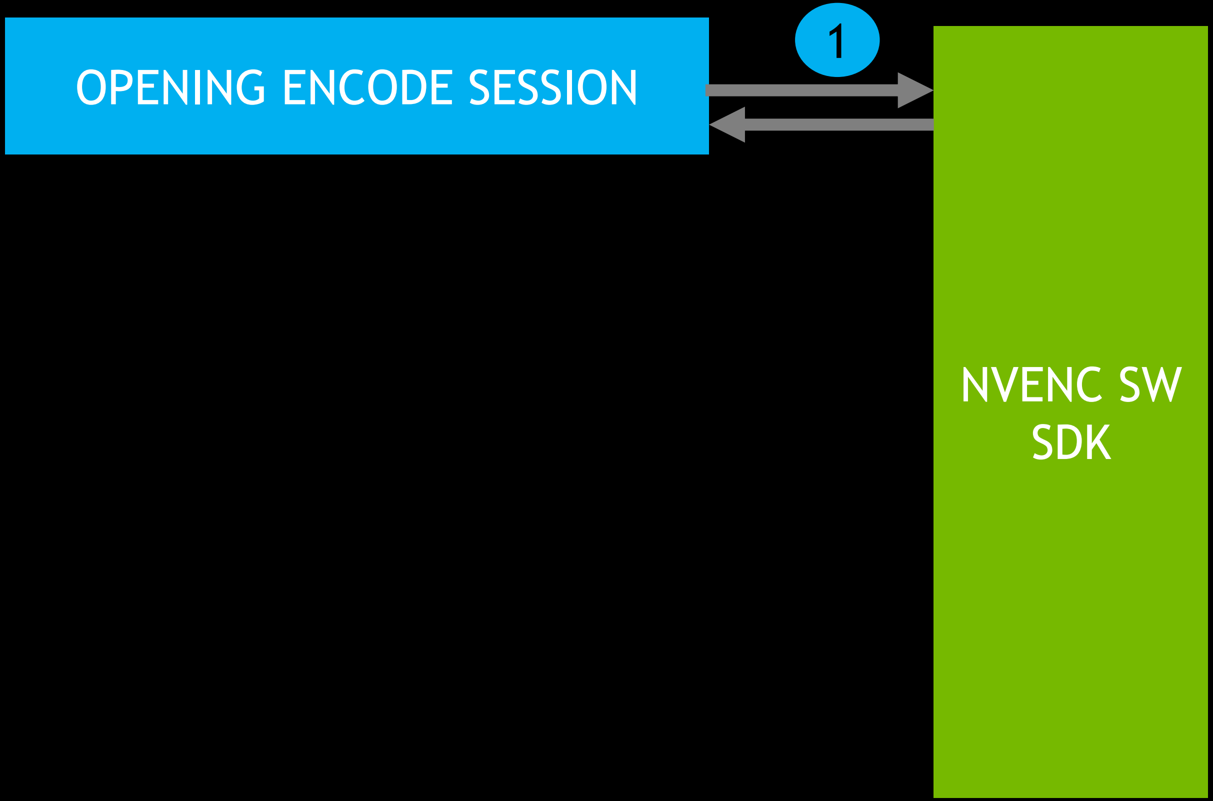
NVENC STACK


Client application
Encoded bitstream
Initialize, Configure, Encode
NVENC API
Configure HW
CUDA Driver
NVENC Driver
DirectX Driver
HW Encode
NVENC firmware + hardware

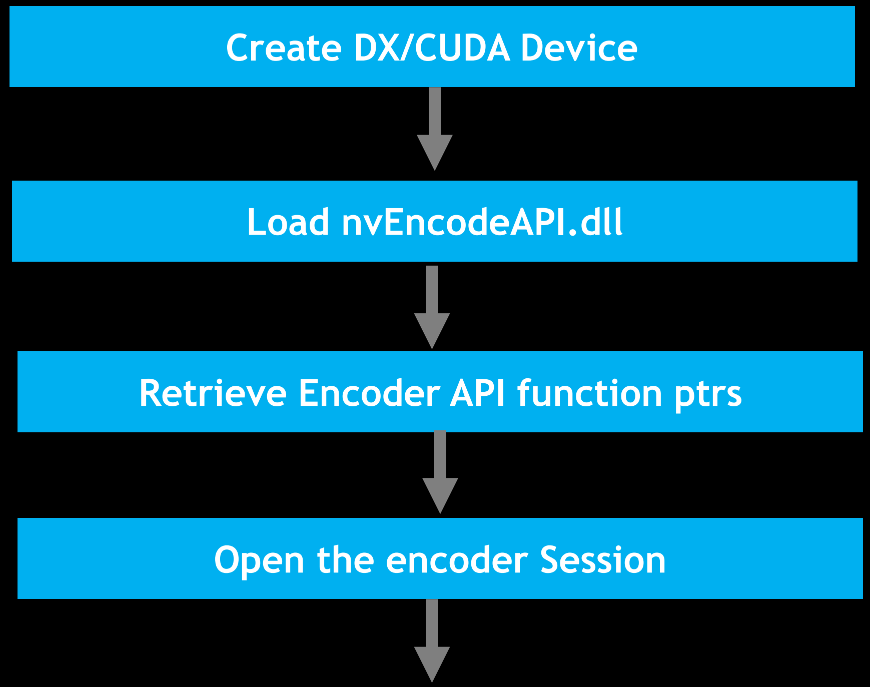
NVENC API FLOW



OPENING ENCODE SESSION



OPENING ENCODE SESSION



Create DX/CUDA Device

Load nvEncodeAPI.dll

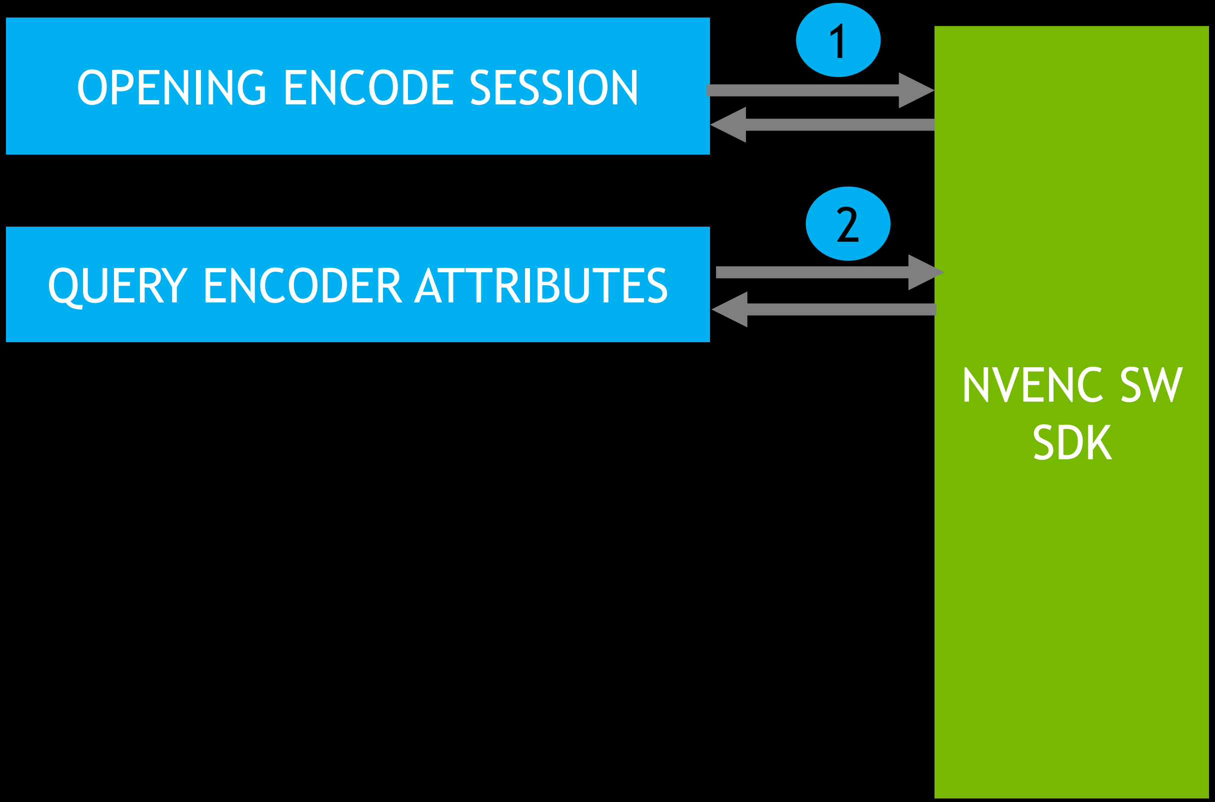
Retrieve Encoder API function ptrs

Open the encoder Session

OPENING ENCODE SESSION

- The NVENC SDK API shared library(dll) name is nvEncodeAPI.dll
- It has a single entry point NvEncodeAPICreateInstance
- NvEncodeAPICreateInstance to retrieve the API function pointers.
- NvEncOpenEncodeSessionEx API to start encode session.
- Application must create a DX or CUDA device , which passed as part of NvEncOpenEncodeSessionEx API.

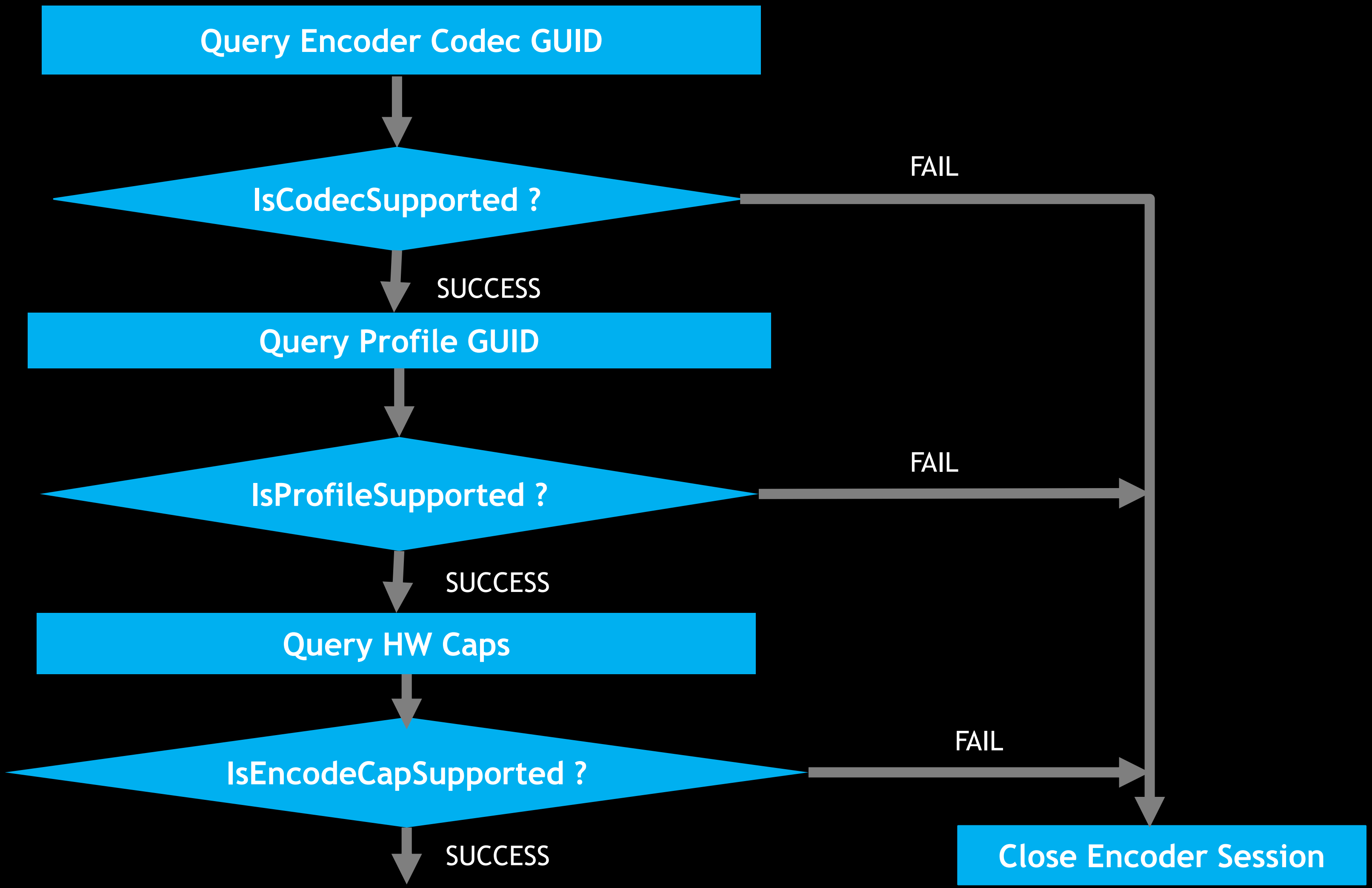
QUERY ENCODER ATTRIBUTES



QUERY ENCODER ATTRIBUTES

HW ENCODER ATTRIBUTES	ATTRIBUTE GUIDS	
ENCODE GUID	NV_ENC_CODEC_H264_GUID	H264/MPEG4 AVC
PROFILE GUID	NV_ENC_H264_PROFILE_BASELINE_GUID NV_ENC_H264_PROFILE_HIGH_GUID NV_ENC_H264_PROFILE_MAIN_GUID	H264 BASELINE PROFILE H264 HIGH PROFILE H264 MAIN PROFILE
ENCODER CAPS	NV_ENC_CAPS_SUPPORTED_RATECONTROL_MODES, NV_ENC_CAPS_SUPPORT_CABAC, NV_ENC_CAPS_SUPPORT_BDIRECTMODE, NV_ENC_CAPS_SUPPORT_STEREO_MVC	

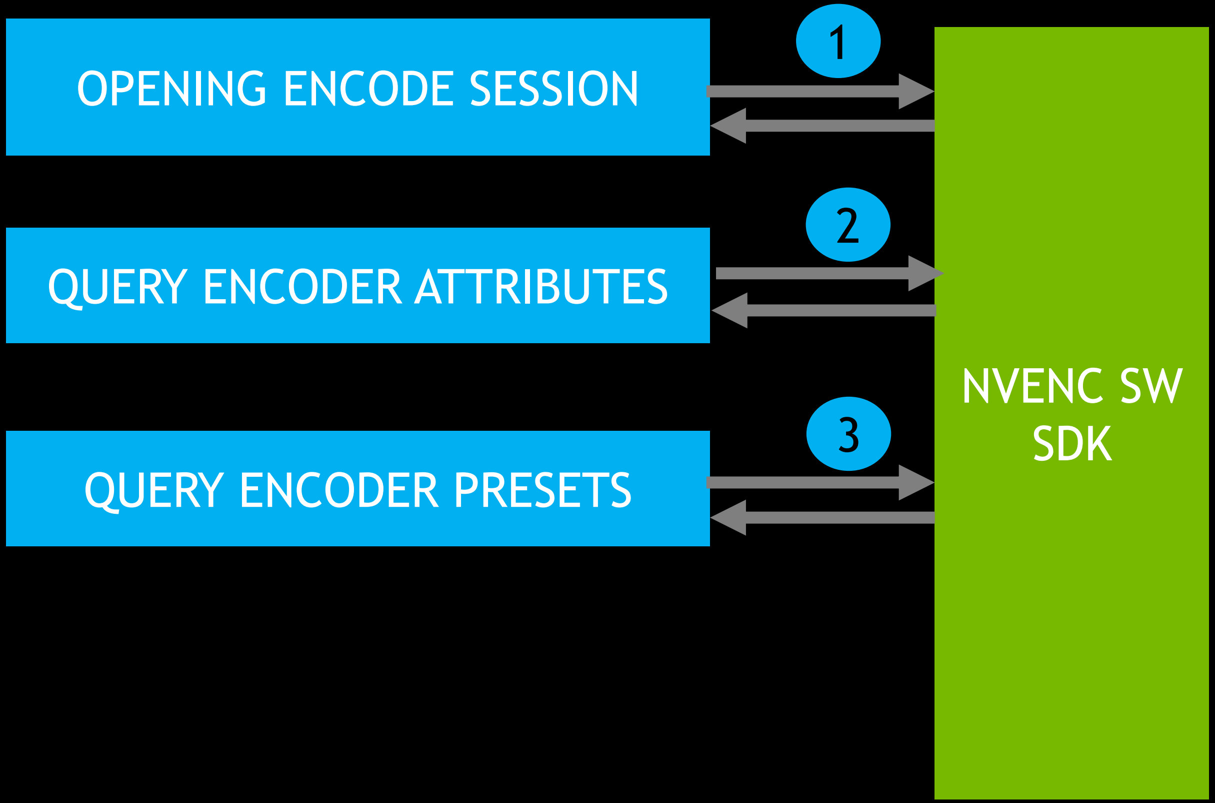
QUERY ENCODER ATTRIBUTES



QUERY ENCODER ATTRIBUTES

- **Query Codec GUID**
  - NvEncGetEncodeGUIDCount
  - NvEncGetEncodeGUIDs
- **Query Profile GUID**
  - NvEncGetEncodeProfileGUIDCount
  - NvEncGetEncodeProfileGUIDs
- **Query Encode Caps**
  - NvEncGetEncodeCaps

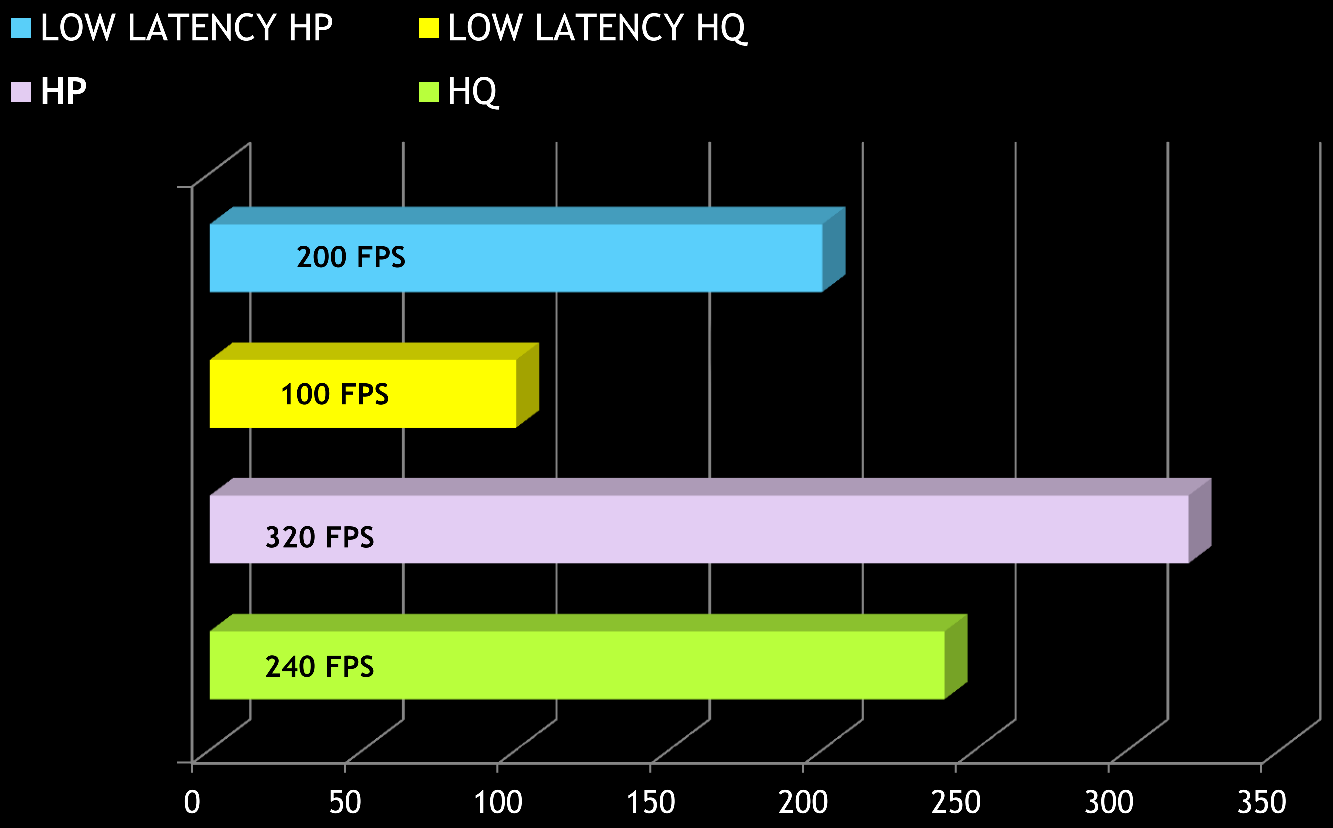
QUERY ENCODER PRESETS



QUERY ENCODER PRESETS

PRESET	Encoder Settings	APPLICATION
HIGH QUALITY	B Frames, CABAC, 8x8 Transform, All Intra Modes, All Inter Modes*, VBR RC, GopLength 30	TRANSCODING HIGH BITRATE
HIGH PERFORMANCE	No B Frames, CAVLC, P16x16, Intra16x16 and Intra4x4 Modes, VBR, GopLength 30	MULTIPLE TRANSCODING
LOW LATENCY HQ	No B Frames, CABAC, All Intra , All Inter Modes, Single frame VBV 2 PASS, Infinite GOP,	CLOUD GAMING, MIRACAST, VIDEO CONFERENCING
LOW LATENCY HP	No B Frames, CABAC, All Intra and Inter Modes, Single frame VBV 2 PASS, Infinite GOP, Smaller Search Range compared to LOW LATENCY HQ	CLOUD GAMING, MIRACAST

ENCODER PRESETS

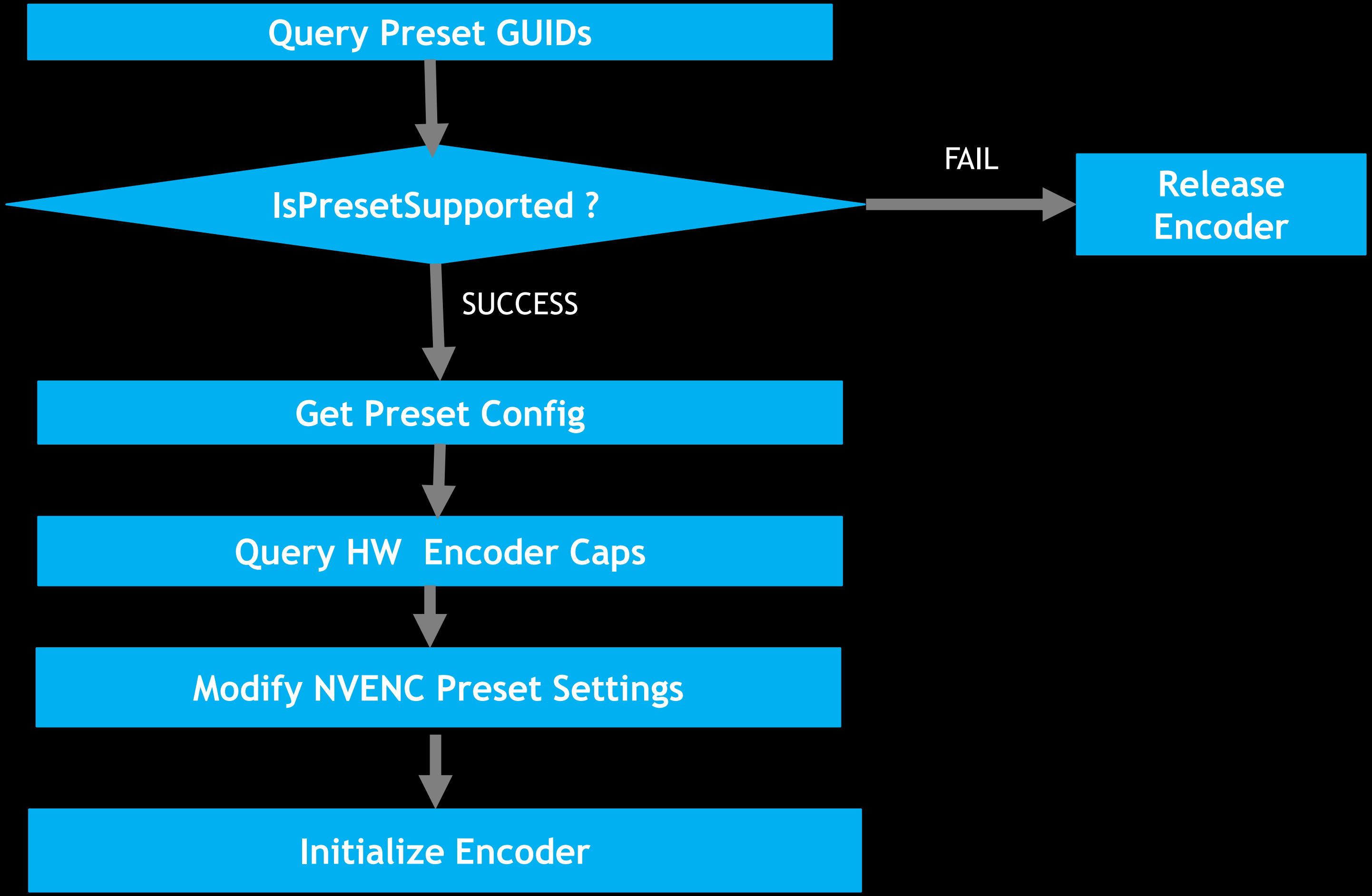


720p Performance on NVIDIA Geforce GTX 650

ENCODER PRESETS

- Query Encoder Presets
  - NvEncGetEncodePresetCount
  - NvEncGetEncodePresets
- Get Encoder Presets settings
  - NvEncGetEncodePresetConfig
  - NvEncGetEncodeCaps API to query HW caps

ENCODER PRESETS



Query Preset GUIDs

IsPresetSupported ?

FAIL

Release Encoder

SUCCESS

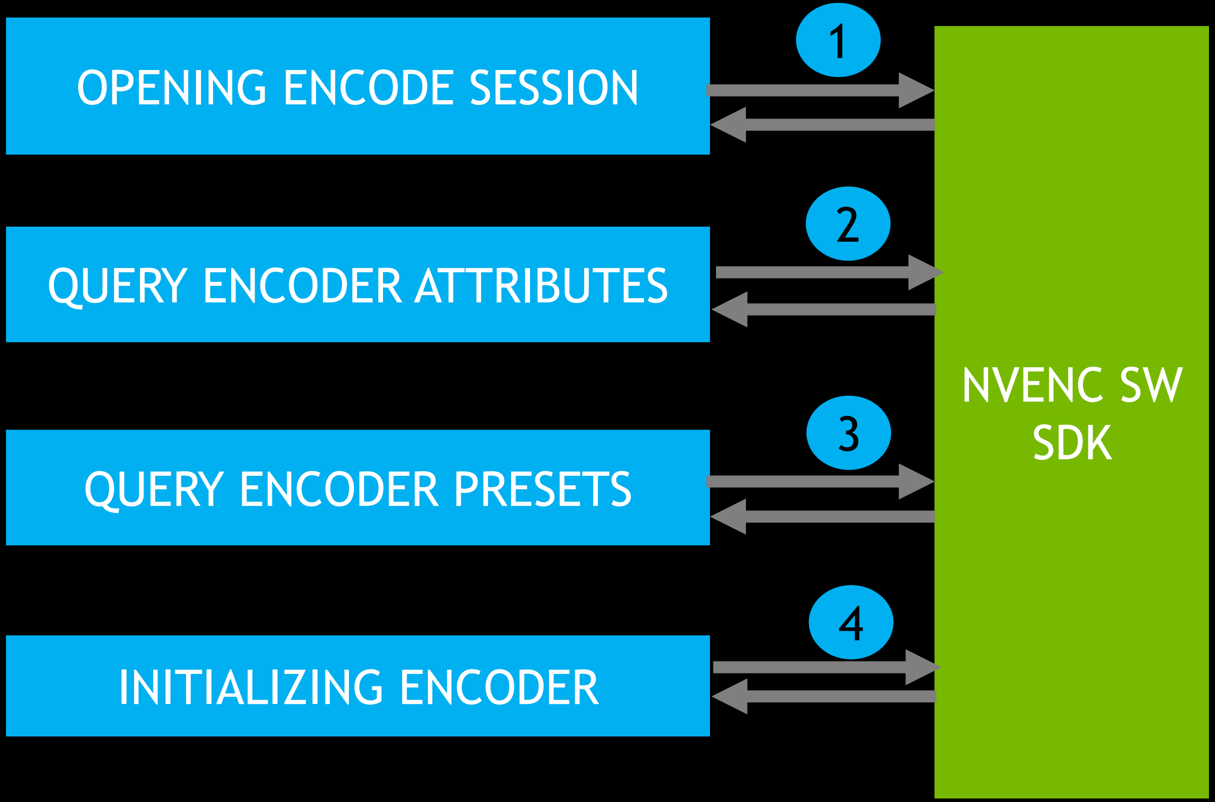
Get Preset Config

Query HW Encoder Caps

Modify NVENC Preset Settings

Initialize Encoder

INITIALIZING ENCODER



INITIALIZING ENCODER

- NvEncInitializeEncoder API.
- Parameters used for Initializing the Encoder
  - **NV_ENC_INITIALIZE_PARAMS**

    Basic Encoder parameters common for all codecs.
  - NV_ENC_CONFIG
    - Optional advance codec parameters for applications which want more control over the encoder and supports various codec specific parameters
    - NV_ENC_CONFIG_H264

INITIALIZING ENCODER

- NV_ENC_INITIALIZE_PARAMS

Description	Parameter Name
Encode Dimensions	encodeWidth , encodeHeight
Codec	encodeGUID
Preset	presetGUID
Display Aspect Ratio	darWidth, darHeight
Frame Rate	frameRateNum, frameRateDen
Async Event Based Signaling	enableEncodeAsync
Picture Type Decision	enablePTD
Low Latency Slice based read back	enableSubFrameWrite
Slice Offsets reporting	reportSliceOffsets

INITIALIZING ENCODER

- NV_ENC_CONFIG

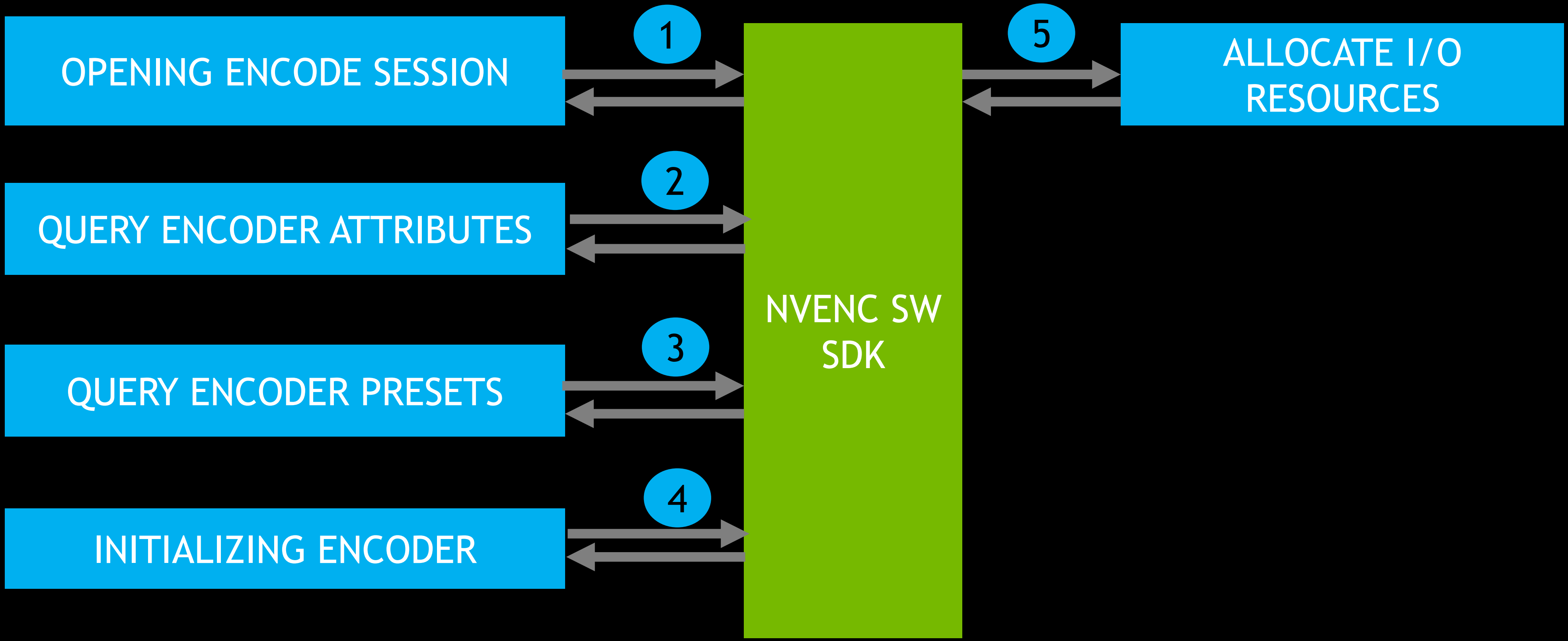
Description	Parameter Name
Profile	profileGUID
GOP structure	gopLength, frameIntervalP
Rate Control Parameters	rcParams
MV Precision(Qpel/Hpel/Fpel)	mvPrecision
Input Frame structure	frameFieldMode
H264 Codec parameters (NV_ENC_CONFIG_H264)	encodeCodecConfig

INITIALIZING ENCODER

- **NV_ENC_CONFIG_H264**

Description	Parameter Name
Key frame interval	idrPeriod
VLE mode	entropyCodingMode
Adaptive Block Transform(8x8)	adaptiveTransformMode
Disable Deblocking Flags	disableDeblockingFilterIDC
Slice Parameters	sliceMode, sliceModeData
H264 VUI Parameters	h264VUIParams
Bdirect Mode	bdirectMode
DPB size	maxNumRefFrames
Intra Refresh	intraRefreshPeriod, intraRefreshCnt

ALLOCATE I/O RESOURCES

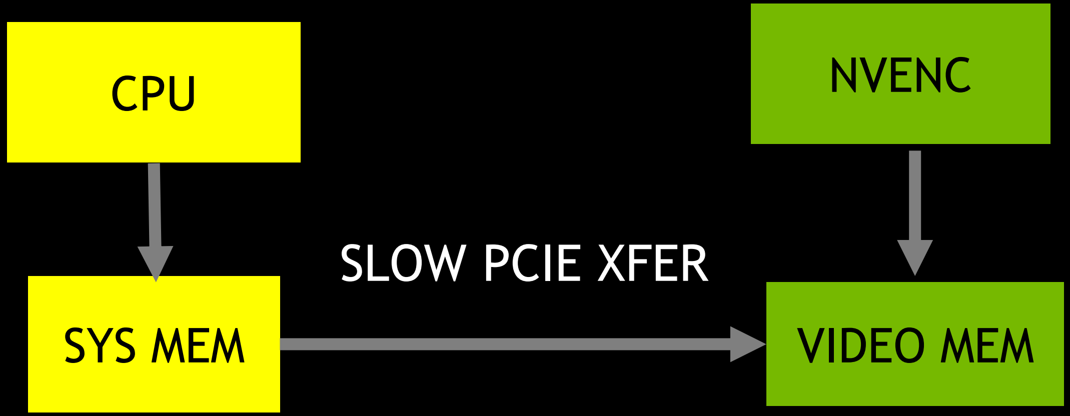


INPUT RESOURCES

- Two types of Input Resources
  - NVENC Input Buffers
  - Externally Allocated DX/Cuda Buffers mapped to NVENC
    - NV_ENC_INPUT_RESOURCE_TYPE_DIRECTX
    - NV_ENC_INPUT_RESOURCE_TYPE_CUDADEVICEPTR

NVENC INPUT BUFFERS

- NVENC Input Buffers
  - Provides a simple interface to load input data from system memory.
  - Includes an expensive copy of input from system to video memory using NvEncLockInputBuffer API.

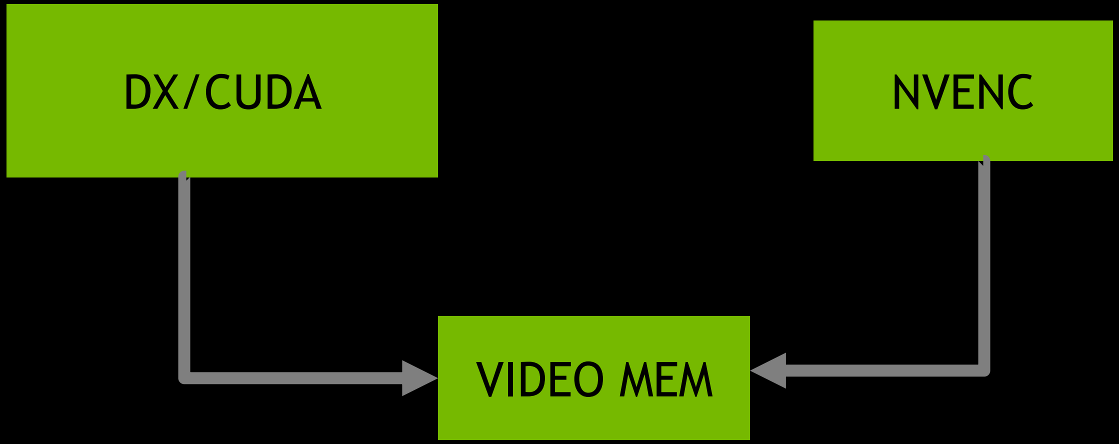


NVENC INPUT BUFFERS

- NVENC Input Buffers are allocated using
  - NvEncCreateInputBuffer
    - Only NV_ENC_BUFFER_FORMAT_NV12_PL is supported
  - NvEncDestroyInputBuffer
- Application loads input data on NVENC Input Buffers using
  - NvEncLockInputBuffer
  - NvEncUnlockInputBuffer

MAPPING DX / CUDA INPUT RESOURCES TO NVENC

- Mapping DX / CUDA Buffers to NVENC
  - Direct mapping of video memory buffer to NVENC address space
  - Removes the expensive copy of system memory data to video memory.
  - Much lower latency than NVENC Input buffer method.



MAPPING DX / CUDA INPUT RESOURCES TO NVENC

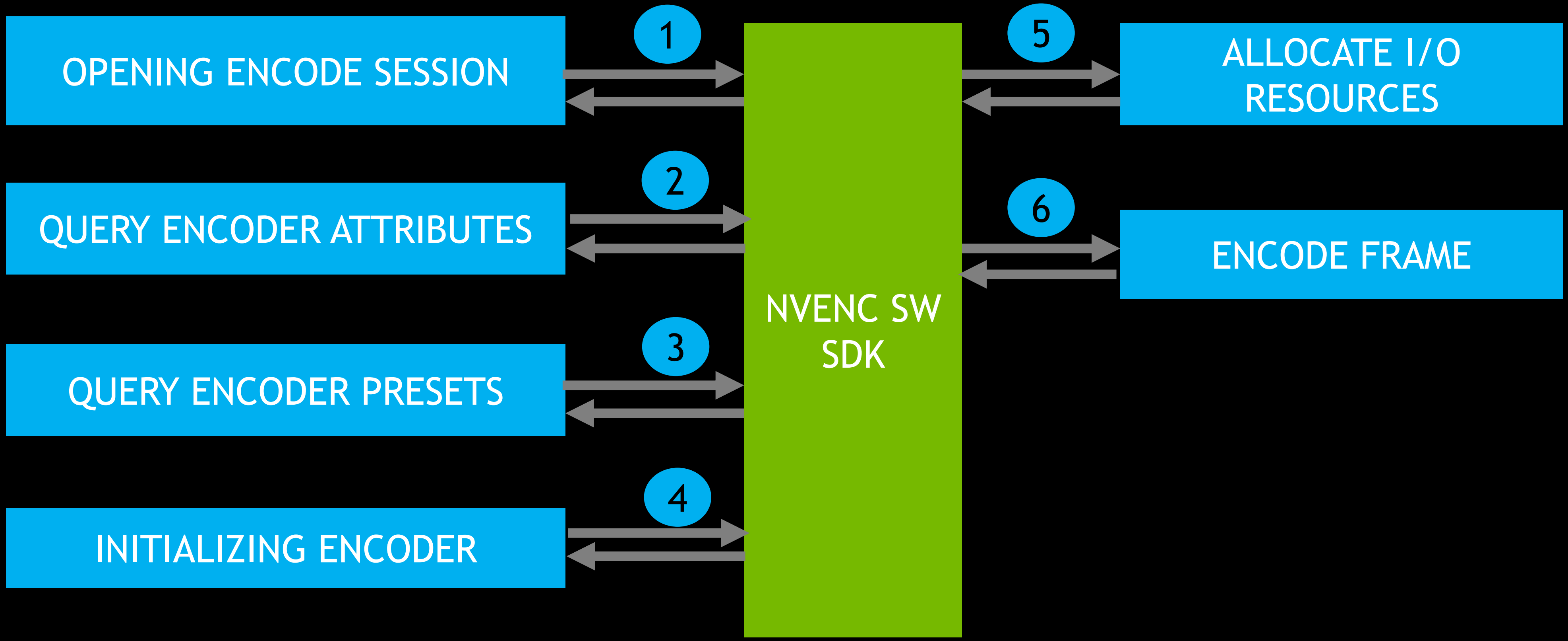
- Mapping DX / CUDA Resources to NVENC
    - Provides DX/CUDA interoperability with NVENC
    - Create an NV12 buffer using DX /CUDA API
    - Register the DX/CUDA Resource with NVENC
        - NvEncRegisterResource
    - Map the DX/CUDA Resource with NVENC before sending it for Encoding
        - NvEncMapInputResource
    - Unmap the DX/CUDA Resource once frame has been encoded
        - NvEncUnMapInputResource
    - Unregister the DX/CUDA Resource before destroying it.
        - NvEncUnRegisterResource

ALLOCATING OUTPUT BUFFERS

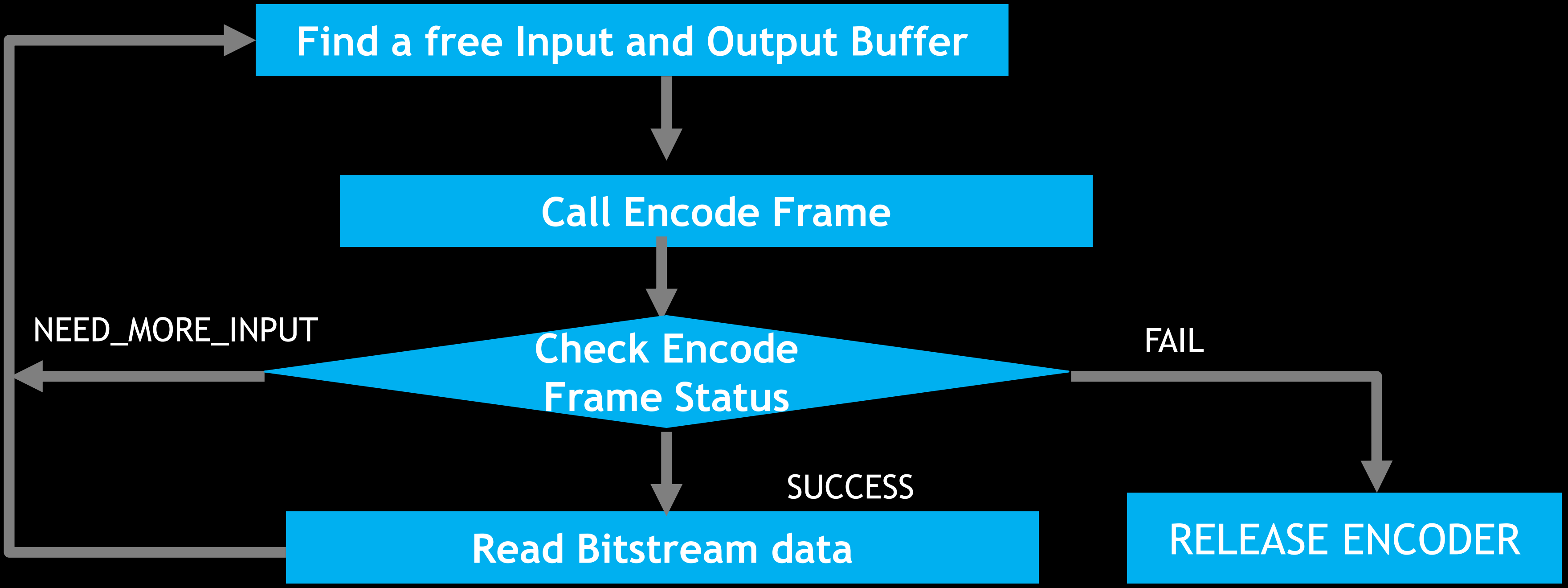
- Allocating Output Bitstream Buffer
  - NvEncCreateBitstreamBuffer
  - NvEncDestroyBitstreamBuffer

- Allocating Output buffer completion Event(*Windows Only)
  - CreateEvent
  - NvEncRegisterAsyncEvent
  - NvEncUnregisterAsyncEvent

ENCODE FRAME



ENCODE FRAME



ENCODE FRAME

- NvEncEncodePicture API used for submitting input buffers for encoding.

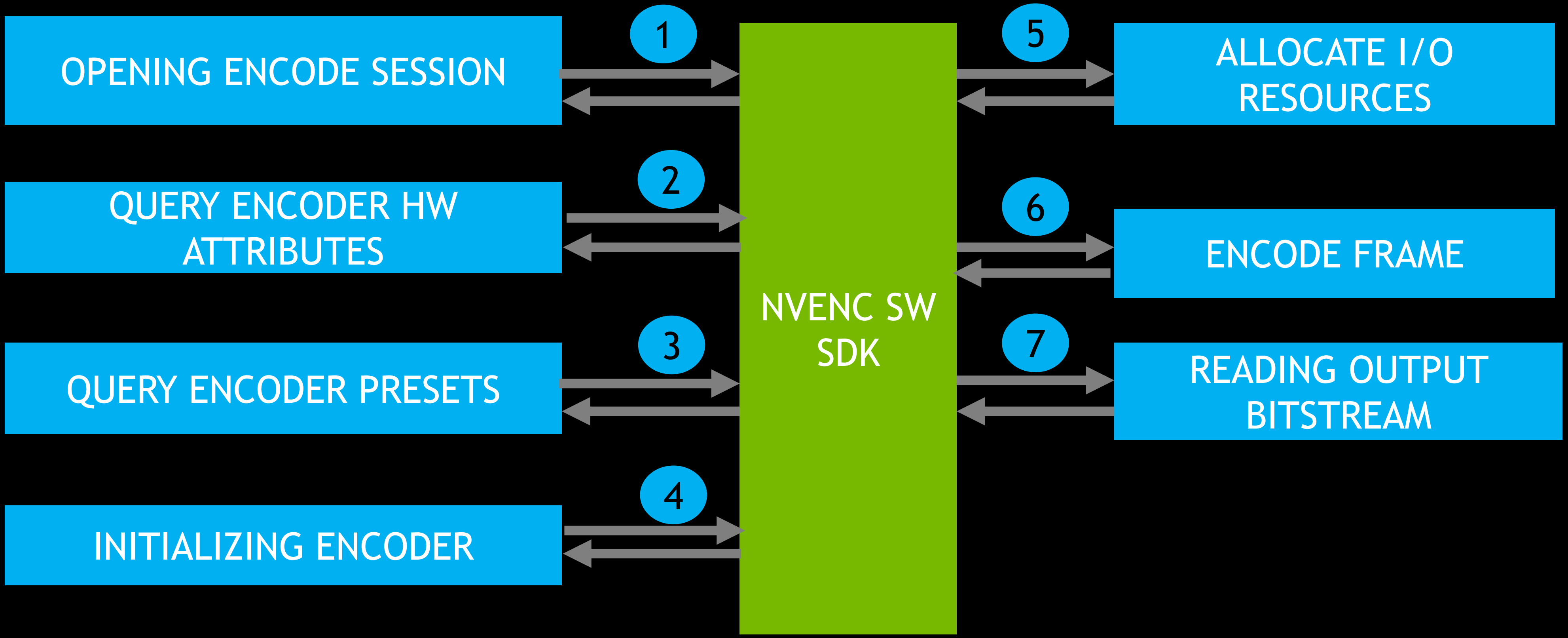
- Input Buffers are submitted
  - Display Order : I B B P B B P
    - Reordering done by NVENC SDK
  - Encoder Order : I P B B P B B
    - Reordering done by Application

ENCODE FRAME

- Application submitting buffers in Encode order must specify
  - NV_ENC_PIC_PARAMS :: pictureType
  - NV_ENC_PIC_PARAMS_H264 :: displayPOCSyntax
  - NV_ENC_PIC_PARAMS_H264 :: refPicFlag
  - NV_ENC_INITIALIZE_PARAMS :: enablePTD to 0

- Application submitting buffers in Display order must specify
  - NV_ENC_CONFIG ::gopLength
  - NV_ENC_CONFIG :: frameIntervalP
  - NV_ENC_CONFIG_H264 :: idrPeriod
  - NV_ENC_INITIALIZE_PARAMS :: enablePTD to 1

READING OUTPUT BITSTREAM



READING OUTPUT BITSTREAM

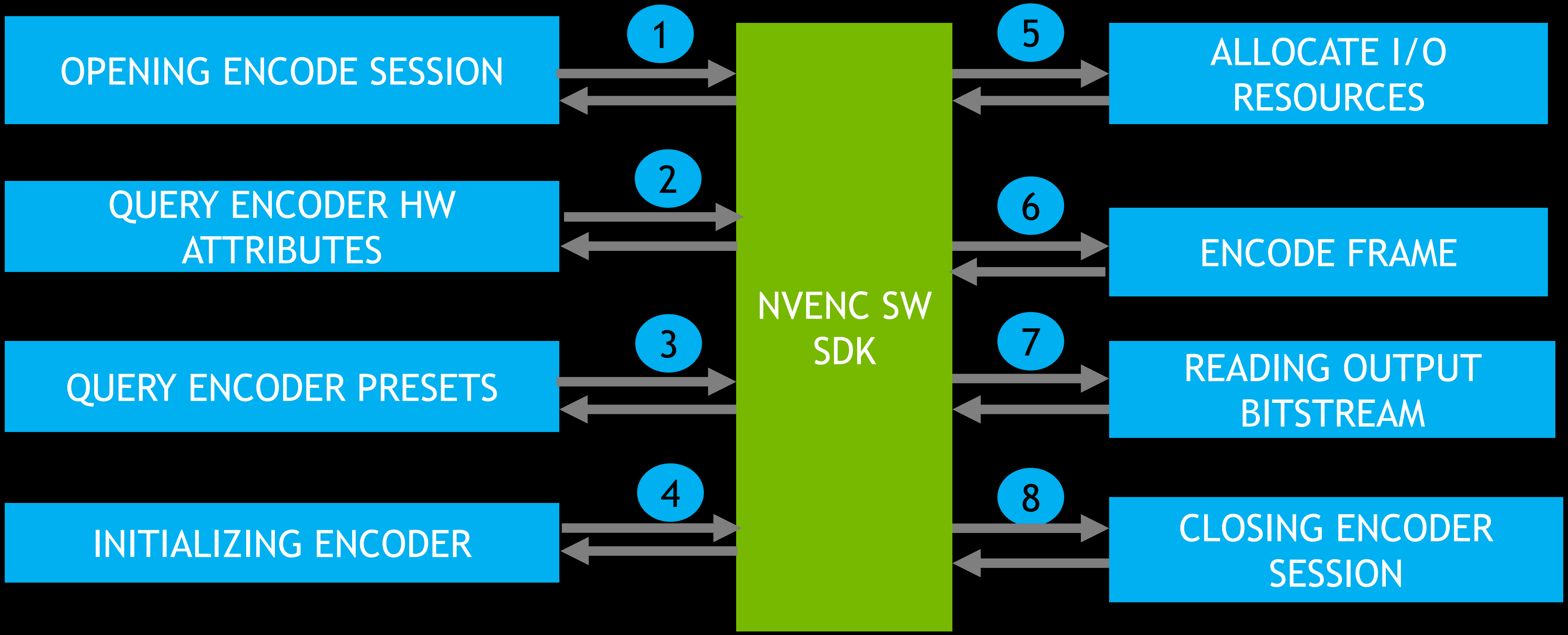
- Reading output buffer after encoding
    - NvEncLockBitstream
    - NvEncUnlockBitstream
- Encode Completion Notification
    - NvEncLockBitstream with doNotWait to 0.
    - Wait on NvENC event (registered with NvEncRegisterAsyncEvent API).
        - Set NV_ENC_INITIALIZE_PARAMS::enableEncodeAsync to 1

READING OUTPUT BITSTREAM

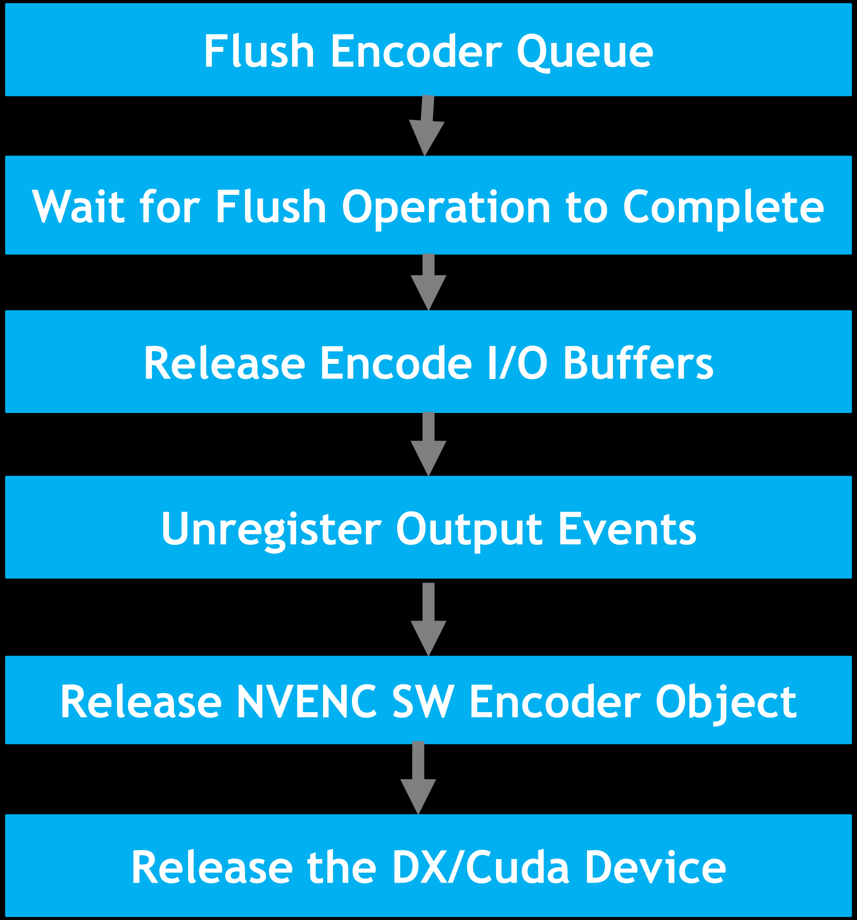
- Slice Level Readback
  - NvEncLockBitstream with doNotWait to 1.
  - Set NV_ENC_INITIALIZE_PARAMS::enableSubFrameWrite to 1
  - Poll and read data till NV_ENC_LOCK_BITSTREAM :: hwEncodeStatus = 2
  - Number slices encoded till that loop is reported

    NV_ENC_LOCK_BITSTREAM ::numSlices
  - Slice offset can also be reported
    - NV_ENC_INITIALIZE_PARAMS::reportSliceOffsets = 1;
    - NV_ENC_LOCK_BITSTREAM ::sliceOffsets[]

CLOSING ENCODER SESSION



CLOSING ENCODER SESSION



Flush Encoder Queue

Wait for Flush Operation to Complete

Release Encode I/O Buffers

Unregister Output Events

Release NVENC SW Encoder Object

Release the DX/Cuda Device

CLOSING ENCODER SESSION

- Flush Encoder Queue : NvEncEncodePicture with NULL input and output buffer
- Release I/O Buffers
  - NvEncDestroyInputBuffer
  - NvEncDestroyBitstreamBuffer
- Unregister Completion Event
  - NvEncUnregisterAsyncEvent API.
- NvEncDestroyEncoder API.

NVENC RATE CONTROL MODES

- RATE CONTROL MODES
  - NV_ENC_PARAMS_RC_CBR
  - NV_ENC_PARAMS_RC_VBR
  - NV_ENC_PARAMS_RC_2_PASS_QUALITY
  - NV_ENC_PARAMS_RC_2_PASS_FRAMESIZE_CAP

NVENC RATE CONTROL MODES

- NV_ENC_PARAMS_RC_CBR
  - Single Pass Constant Bitrate Rate Control Mode
  - Constant Bitrate doesn’t mean constant frame size
  - Mostly used for media streaming with low end to end delay.

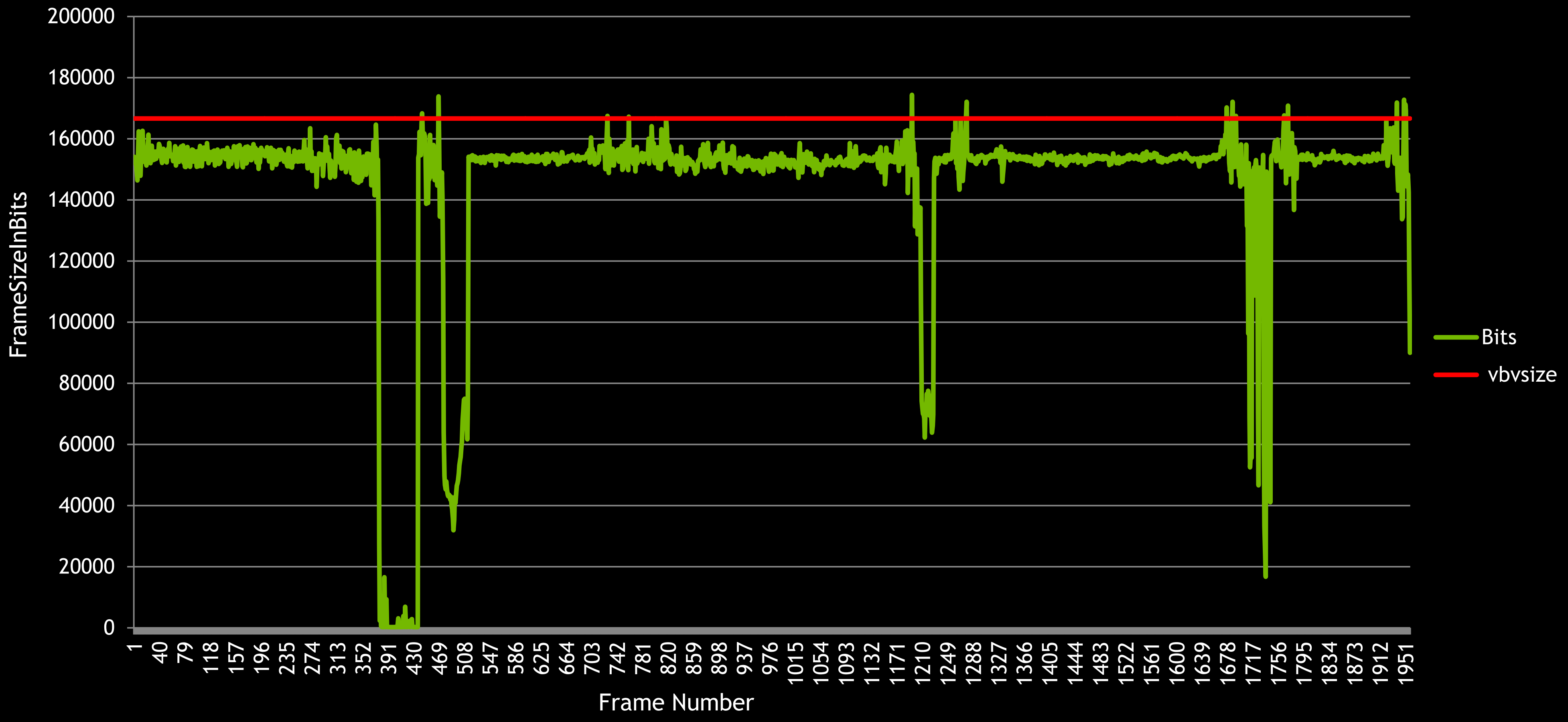
- NV_ENC_PARAMS_RC_VBR
  - Single Pass Variable Bitrate Mode
  - Bitrate varies according to frame complexity.
  - Larger VBV size compared to CBR as a result more flexibility in allocating bits.
  - Mostly used for media storage .

NVENC RATE CONTROL MODES

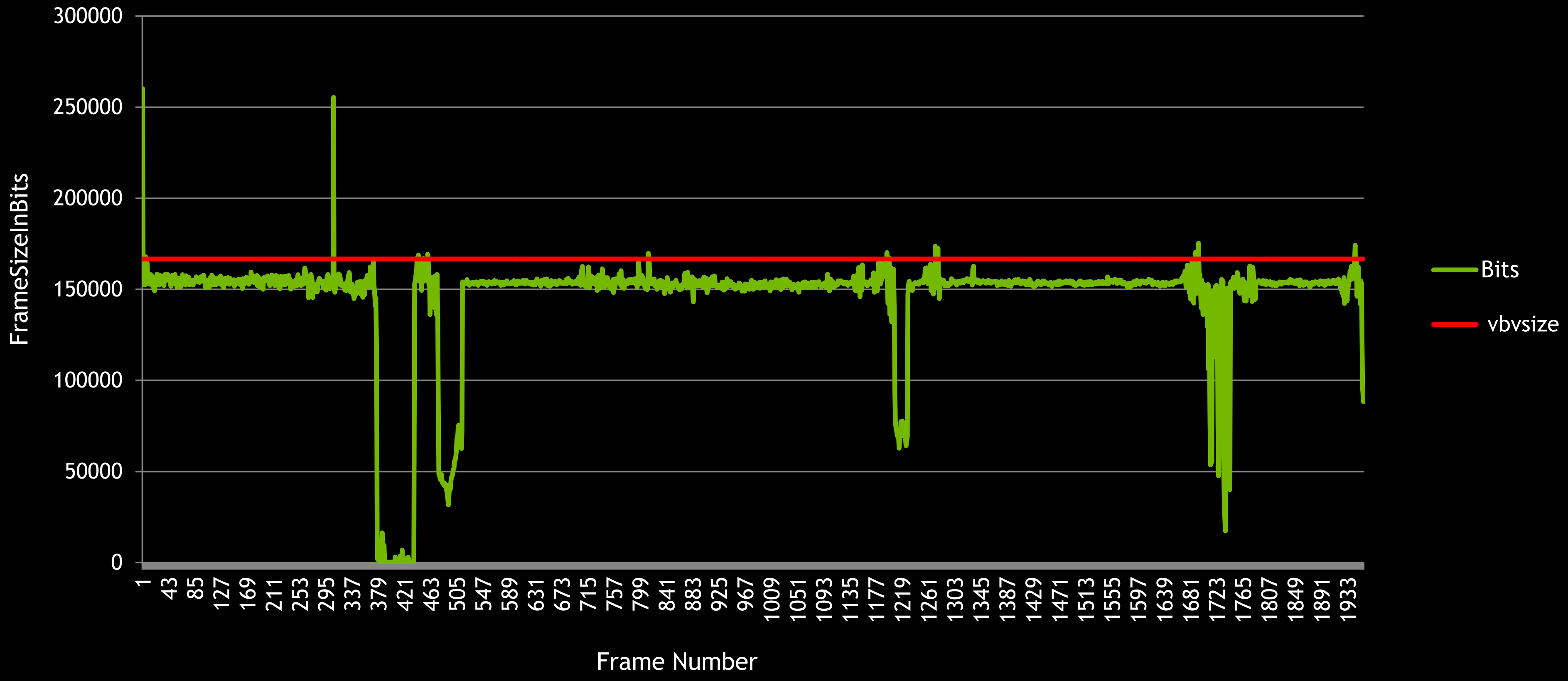
- NV_ENC_PARAMS_RC_2_PASS_FRAMESIZE_CAP
  - Customized two pass CBR for low latency applications
  - First pass analysis without any frame look ahead.
  - Reduces banding effect due to single pass CBR at low bit rate streaming.
  - Mostly used for low delay application like cloud gaming, miracast etc.

- NV_ENC_PARAMS_RC_2_PASS_QUALITY.
  - Customized two pass CBR for single frame VBV cases.
  - Special handling of scene cuts and I frames.

NV_ENC_PARAMS_RC_2_PASS_FRAMESIZE_CAP



NV_ENC_PARAMS_RC_2_PASS_QUALITY



LOW LATENCY ENCODING

- ULTRA LOW LATENCY ENCODER SETTING
- DYNAMIC BITRATE CHANGE
- DYNAMIC RESOLUTION CHANGE
- PERIODIC INTRA REFRESH
- REFERENCE PICTURE INVALIDATION

ULTRA LOW LATENCY ENCODER SETTINGS

- PRESET
    - NV_ENC_PRESET_LOW_LATENCY_HQ_GUID
    - NV_ENC_PRESET_LOW_LATENCY_HP_GUID
        - B FRAMES DISABLED
        - CABAC, 8x8 TRANSFORM, ALL INTRA MODES , ALL INTER MODES
- RATE CONTROL SETTINGS
    - NV_ENC_PARAMS_RC_2_PASS_QUALITY
    - NV_ENC_PARAMS_RC_2_PASS_FRAMESIZE_CAP
    - FIRST PASS ANALYSIS
    - INFINITE GOP
    - SINGLE FRAME VBV
        - VBVSIZE = VBV INITIAL DELAY = BITRATE / FRAME RATE

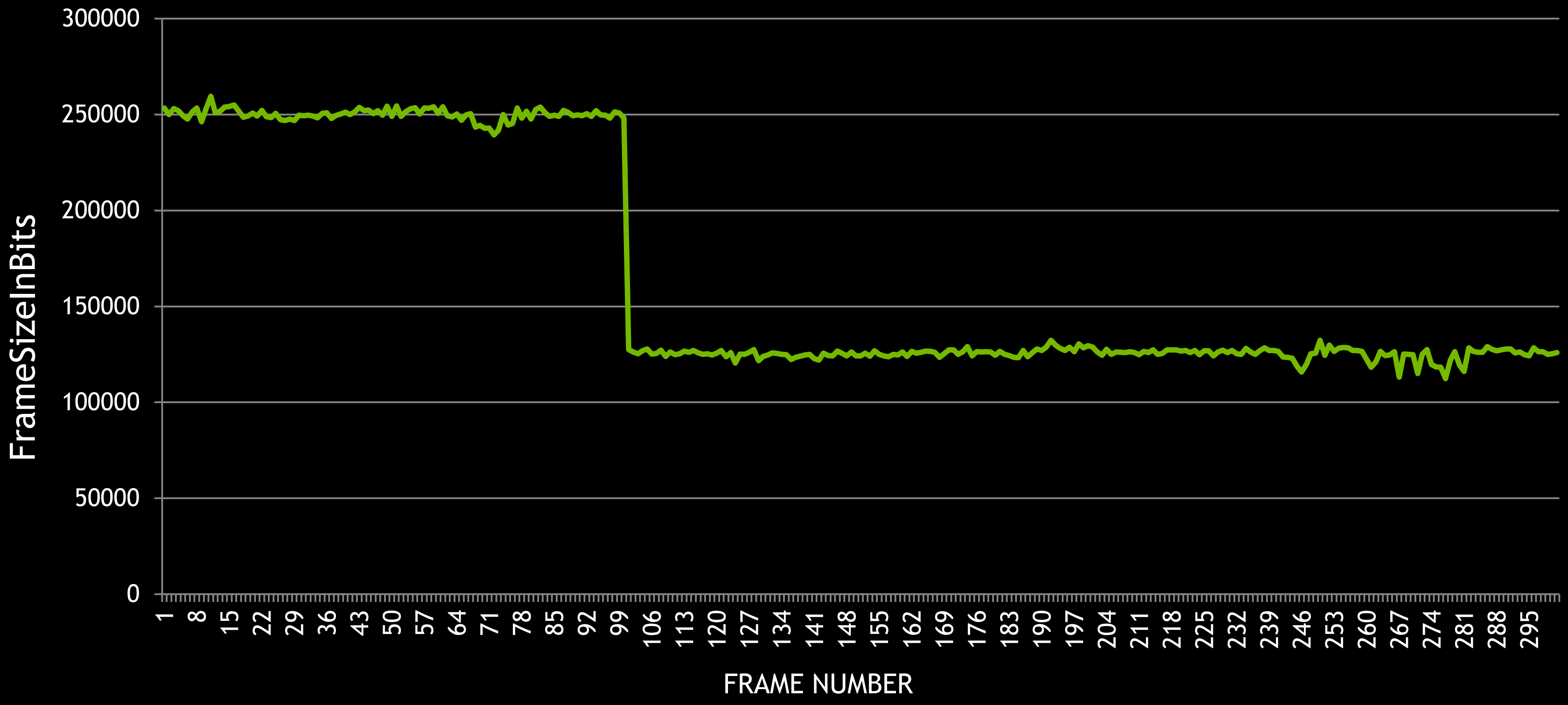
ULTRA LOW LATENCY ENCODER SETTING

- Slice Size In Bytes
- Slice Level Readback of Output Bitstream
- Disable Deblocking across slices
- Constrained Intra Prediction

DYNAMIC BITRATE CHANGE

- NVENC SDK supports dynamic bitrate change within a gop.
- NvEncReconfigureEncoder API
  - NV_ENC_RECONFIGURE_PARAMS :: reInitEncodeParams
  - NV_ENC_CONFIG::rcParams
    - NV_ENC_RC_PARAMS::averageBitRate
    - NV_ENC_RC_PARAMS::maxBitRate
    - NV_ENC_RC_PARAMS::vbvBufferSize
    - NV_ENC_RC_PARAMS::vbvInitialDelay

DYNAMIC BITRATE CHANGE



Bitrate = 8 mbps Frame Number < 100

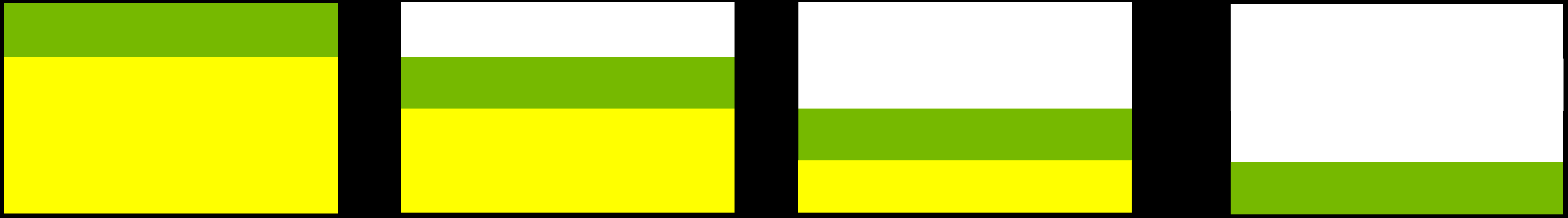
Bitrate = 4 mbps Frame Number > 100

DYNAMIC RESOLUTION CHANGE

- NV_ENC_INITIALIZE_PARAMS::maxEncodeWidth
- NV_ENC_INITIALIZE_PARAMS::maxEncodeWidth
- NvEncReconfigureEncoder API
  - NV_ENC_RECONFIGURE_PARAMS :: reInitEncodeParams
  - NV_ENC_RECONFIGURE_PARAMS :: resetEncoder
  - NV_ENC_RECONFIGURE_PARAMS :: forceIdr

PERIODIC INTRA REFRESH

- NV_ENC_CONFIG_H264::enableIntraRefresh
- NV_ENC_CONFIG_H264:: intraRefreshCnt
- NV_ENC_CONFIG_H264:: intraRefreshPeriod



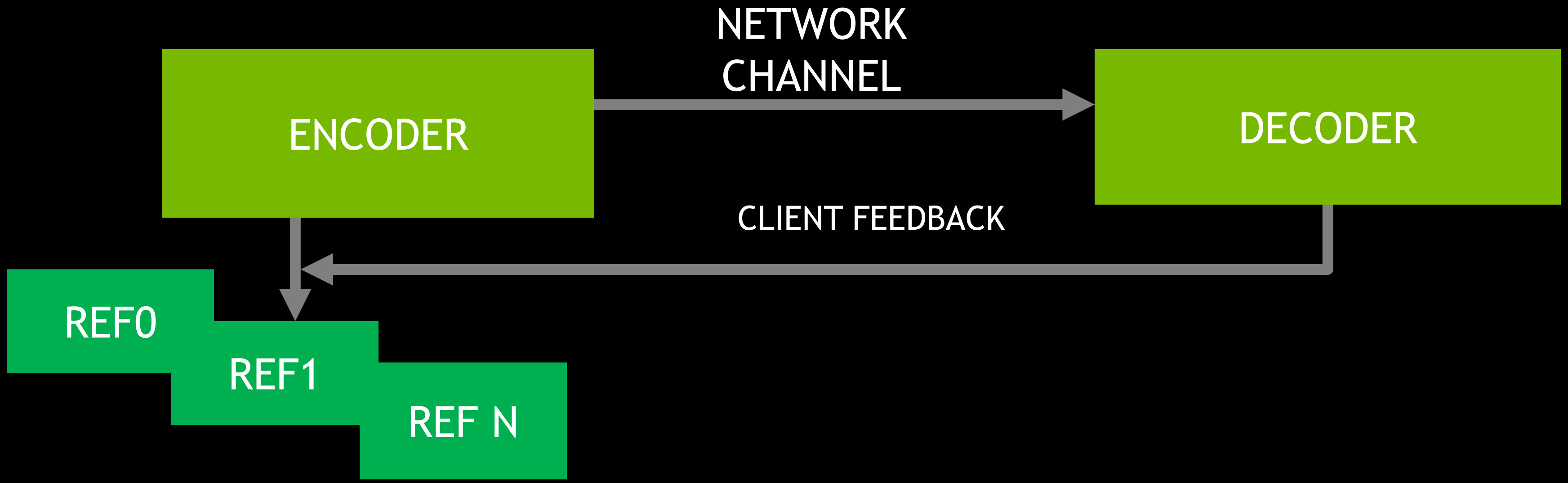
🟩 Intra MBs

🟨 Dirty MBs

⬜ Clean MBs

REFERENCE PICTURE INVALIDATION

- NV_ENC_CONFIG_H264::maxNumRefFrames
- NvEncInvalidateRefFrames API



QUESTIONS?