KVS Days
Intro to Amazon Kinesis Video Streams

What customers are building

Connected home

Video doorbells

Baby monitors

Outdoor cameras

Industrial

Safety management

Predictive maintenance models

Quality assurance

Public sector

Traffic management with 24/7 video

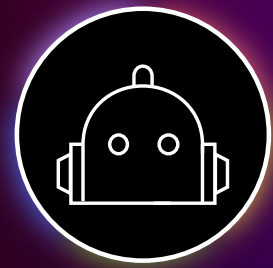
Public safety

Automotive

Sentinel mode for vehicle monitoring

Telematics

Video dash cameras



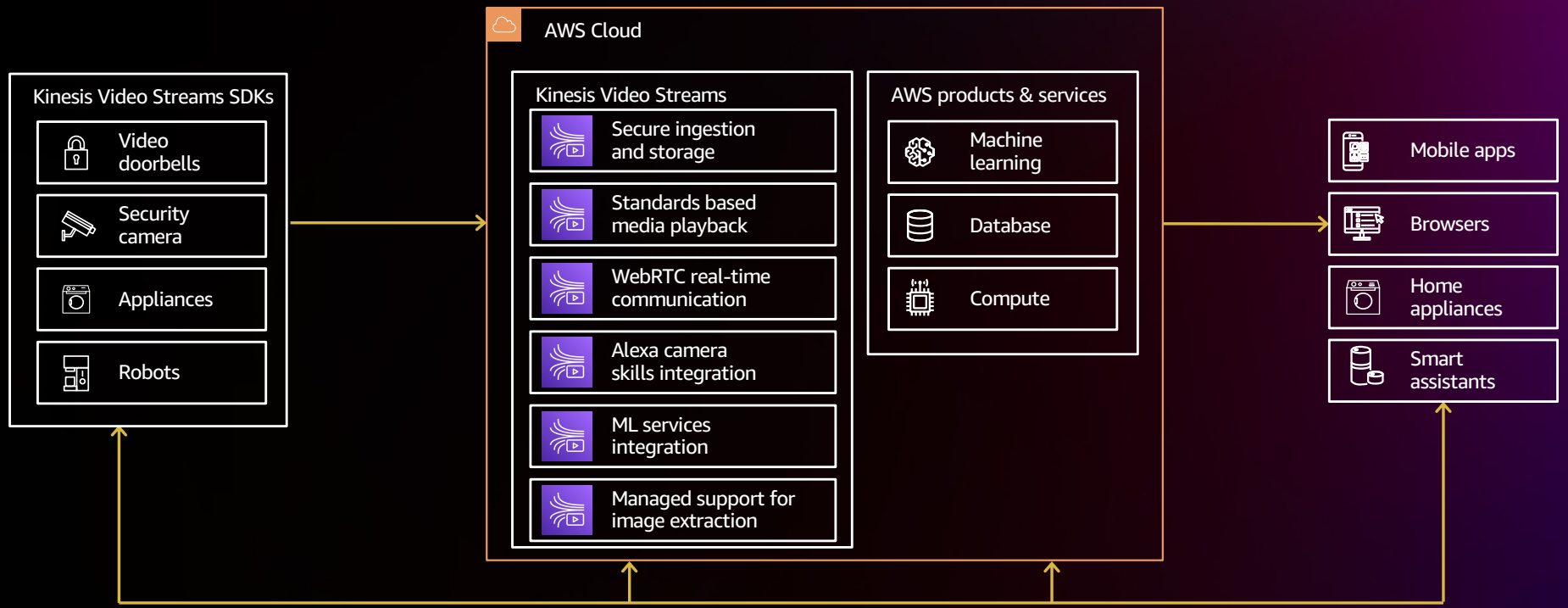
Robotics

Remote control

Other Use Cases

- Amazon Connect for Call recordings
- Audio-only integrations for low-latency transcription via AWS ML services
- Non-RGB including infrared
- Really anywhere you could imagine using a connected camera

Kinesis Video Streams



Connect camera devices with open-source Kinesis Video Streams SDKs to securely stream video or upload clips

Ingest media in Kinesis Video Streams for secure storage, on-demand playback, and integration with ML services; connect over WebRTC for real-time communication and smart assistant integration

Connect to AWS AI/ML services for computer vision and other AWS services to build a unique product experience

Develop customer-facing apps and experiences

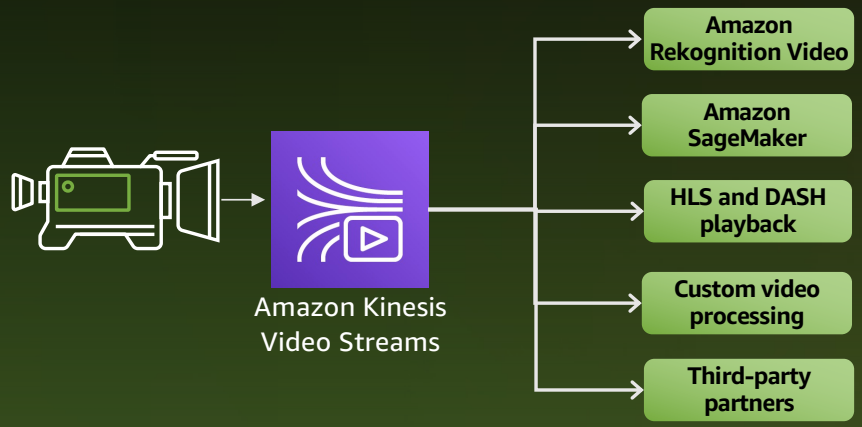
Two streaming methods

Kinesis Video Streams

Secure data ingestion from millions of camera devices

Ingest media and store, consume, and play back time-indexed media data

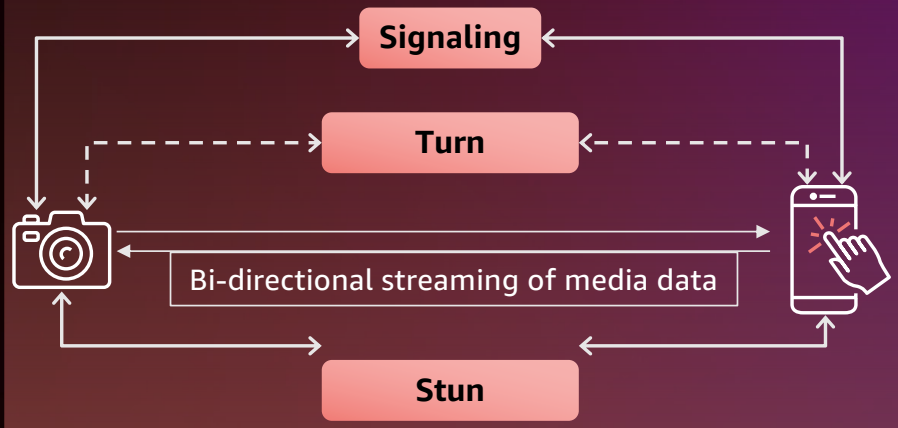
Integration with AI/ML services



Kinesis Video Streams

Low-latency and two-way media streaming with WebRTC

Managed signaling, STUN, and TURN servers



"Stream" and "WebRTC" are used for convenience in this presentation

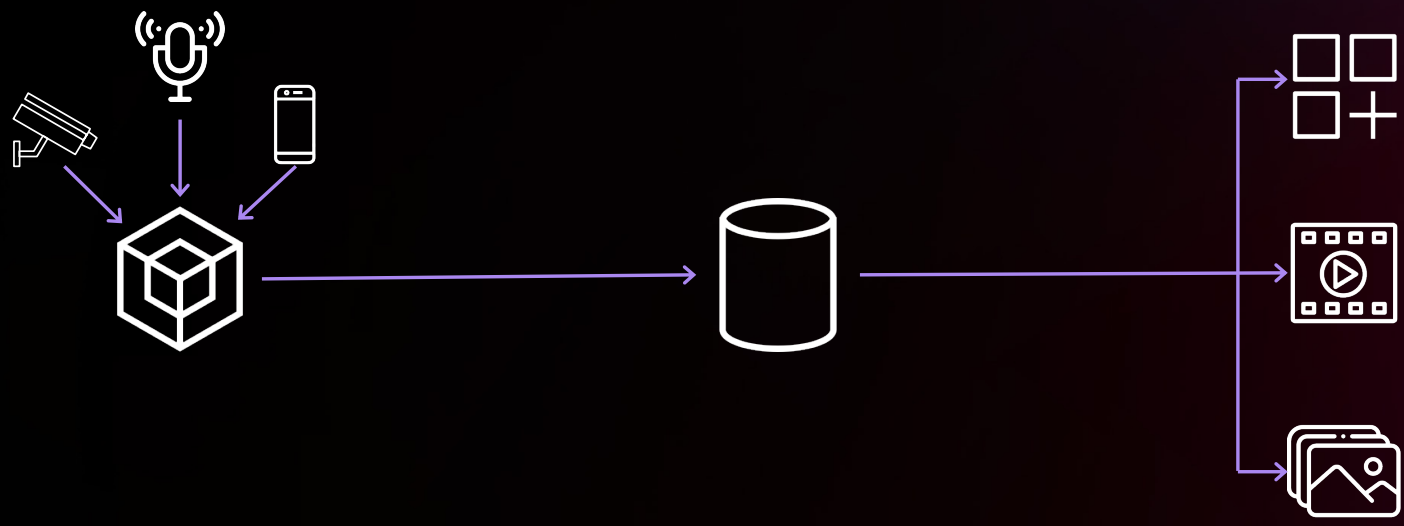
Media Streaming with KVS

KVS - Key APIs

Control Plane	Data Plane	PlayBack
CreateStream	PutMedia	GetHLSStreamingSessionURLMedia
DeleteStream	GetMedia	GetHLSMasterPlaylist
DescribeStream	GetMediaForFragmentList	GetHLSMediaPlaylist
ListStreams	ListFragments	GetMP4InitFragment
GetDataEndpoint		GetMP4MediaFragment
		GetClip

Media Streaming for Camera Devices

Streams



Process
API-based frame by frame access to media for processing

Playback
Real-time and on-demand retrieval of media for playback

Extract
Extract frames from ingested video for enhanced playback (e.g., thumbnails) or AI/ML applications

Ingest
Producer SDKs manage receiving data from device media source and securely transmit to a Kinesis video stream

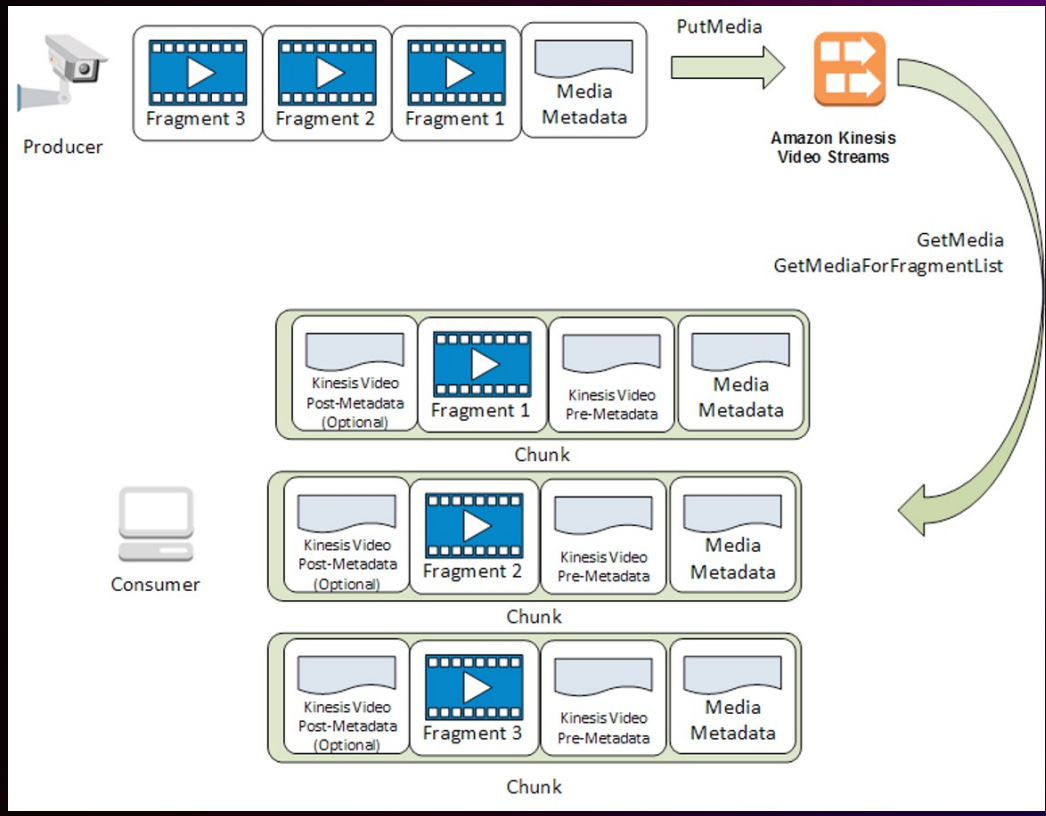
Store
Kinesis Video Streams durably stores
and indexes streams of data

Consume
Retrieve media in real-time or on-demand for playback and processing

aws

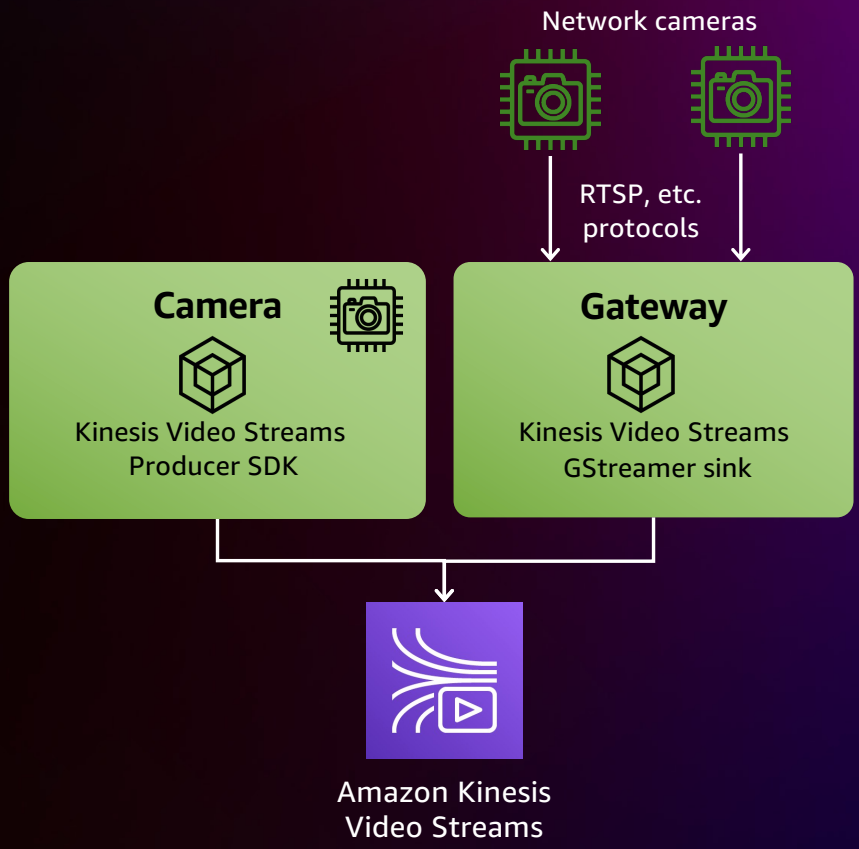
Amazon Kinesis Video Streams - Concepts

- Producer
  - Initiates PutMedia request
  - Sends media meta-data in payload
  - Sends sequence of media fragments
- Kinesis Video Streams
  - Stores media as KVS chunks
  - Media and KVS meta-data
  - Long-Term Persistence
- Consumer
  - Continuous or Batch



Ingest - Kinesis Video Streams

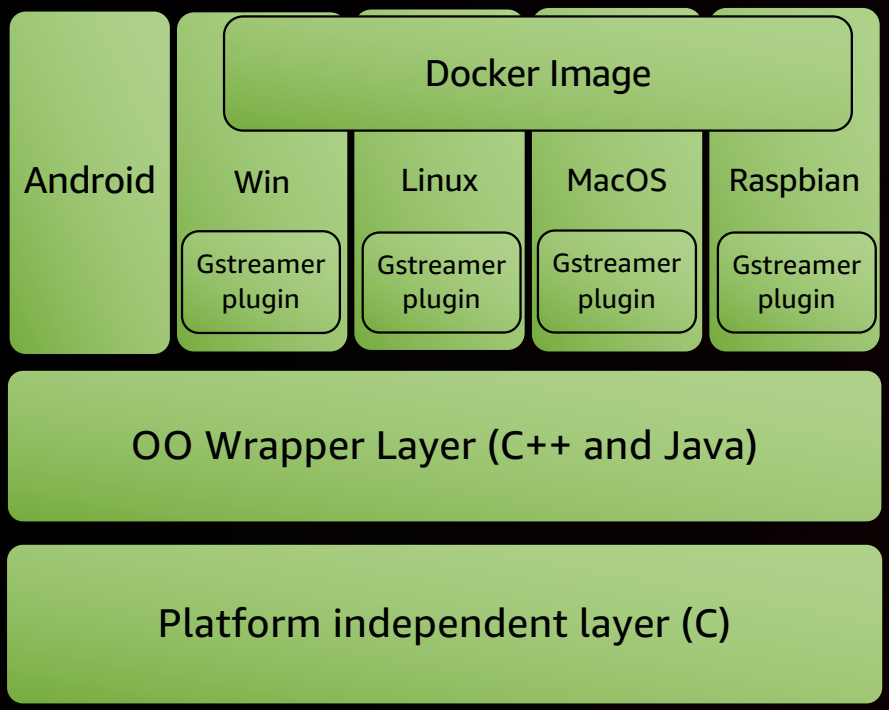
- Stream directly from supported cameras
- Use a proxy/gateway that connects with device on the same network
- Both approaches use the Kinesis Video Streams producer SDKs
- Local media buffering for intermittent network, authentication via AWS IoT Core, credential rotation
- Embed metadata at the individual fragment level
  - Nonpersistent – ad-hoc eg motion detected
  - Persistent – affixed to all frames eg lat/long



Ingest - Producer SDKs

Streams

Flexible SDKs for integration with on-device hardware media pipelines



Minimum storage requirement of 170 MB and a recommended storage requirement of 512 MB

For **app devs** to install on target OS to build applications. Does not satisfy all underlying hardware media pipeline, source listing but makes it easy to do so

For **camera-level integrators** who want a object-oriented, high-performance integration model that satisfies the service calls but is flexible to customize.

For **manufacturers** who want full flexibility to integrate at source and firmware level for different hardware platform types.

aws

How does ingestion work?

Streams

Open-source Java, C, C++ SDKs

Application Callbacks
IoT Credential Provider
Integration
Buffering

Application

```
while( true )
{
    Frame = ReadFrame();
    PutKinesisVideoFrame(Frame);
}
```

PutKinesisVideoFrame()

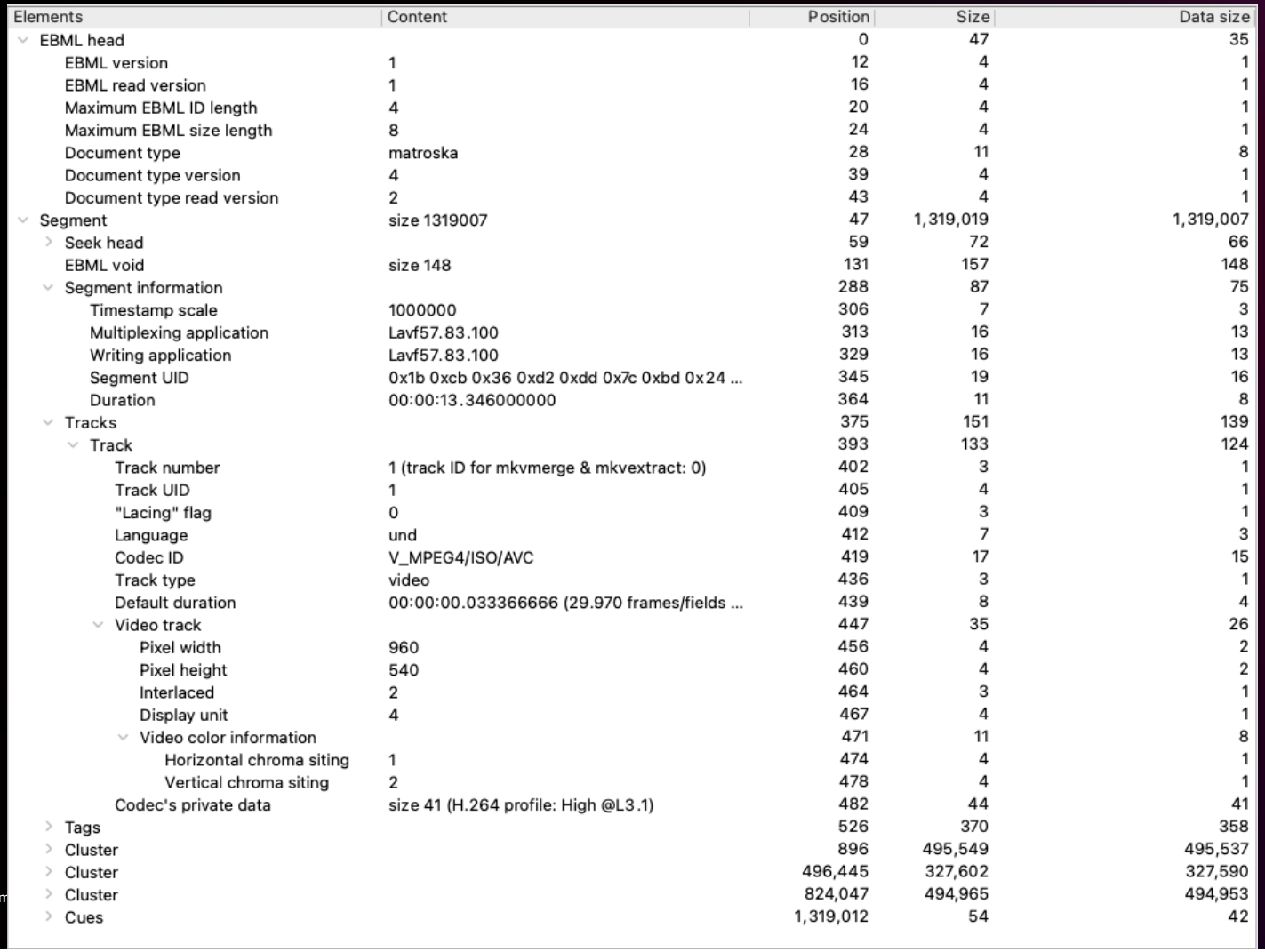
MKV Container

PutMedia

Matroska (MKV) Overview

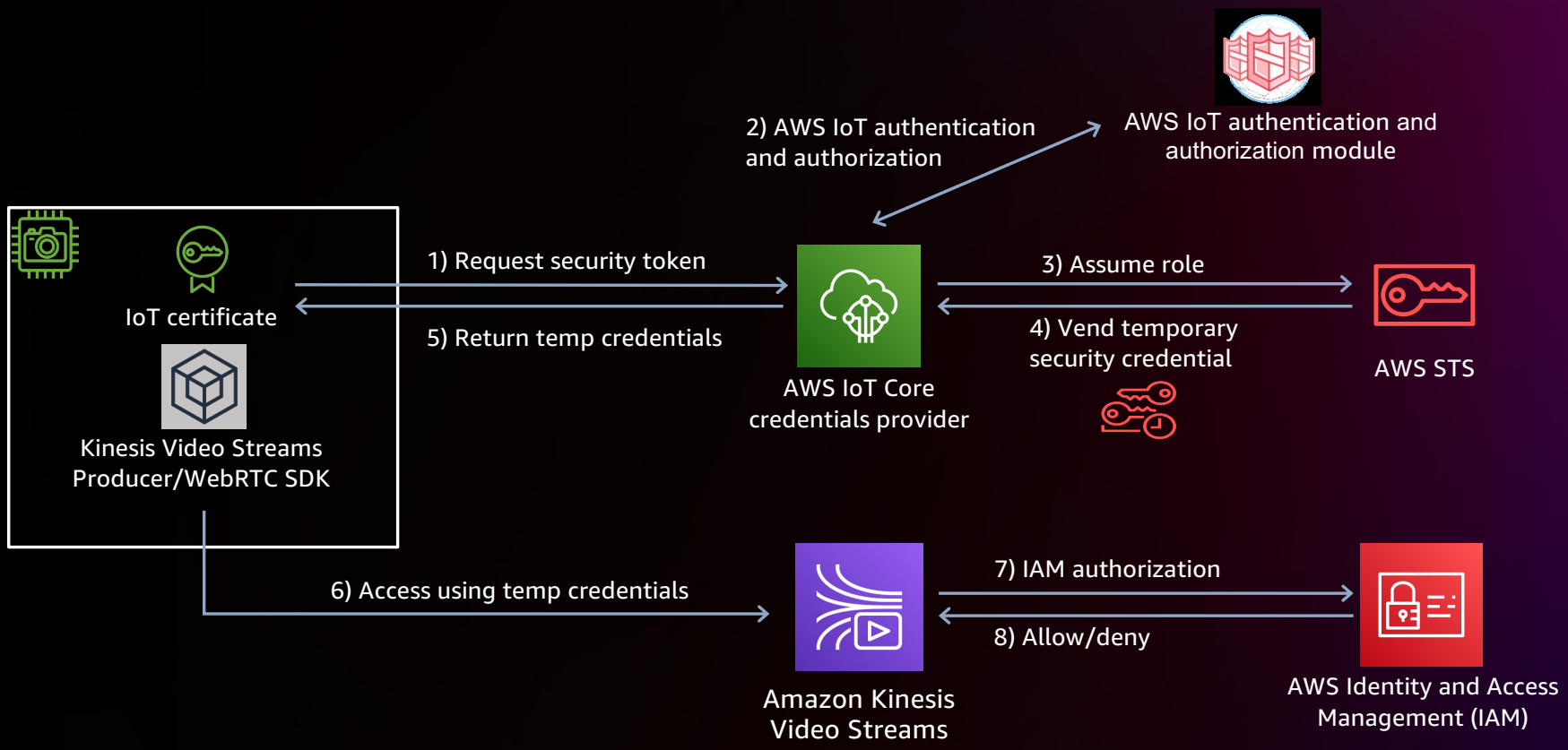
- Open standards project; open source and free to use
- Basis for the WebM format
- Multimedia Container
- Based on the Extensible Binary Meta Language (EBML)
- Supports several video and audio compression formats (codecs)
- MKV is *not* a codec
- Can contain tags
- Detailed explanations available at:
  - https://www.matroska.org/technical/diagram.html
  - https://docs.fileformat.com/video/mkv/

EBML Example



Elements	Content	Position	Size	Data size
EBML head		0	47	35
EBML version	1	12	4	1
EBML read version	1	16	4	1
Maximum EBML ID length	4	20	4	1
Maximum EBML size length	8	24	4	1
Document type	matroska	28	11	8
Document type version	4	39	4	1
Document type read version	2	43	4	1
Segment	size 1319007	47	1,319,019	1,319,007
Seek head		59	72	66
EBML void	size 148	131	157	148
Segment information		288	87	75
Timestamp scale	1000000	306	7	3
Multiplexing application	Lavf57.83.100	313	16	13
Writing application	Lavf57.83.100	329	16	13
Segment UID	0x1b 0xcb 0x36 0xd2 0xdd 0x7c 0xbd 0x24 ...	345	19	16
Duration	00:00:13.346000000	364	11	8
Tracks		375	151	139
Track		393	133	124
Track number	1 (track ID for mkvmerge & mkvextract: 0)	402	3	1
Track UID	1	405	4	1
"Lacing" flag	0	409	3	1
Language	und	412	7	3
Codec ID	V_MPEG4/ISO/AVC	419	17	15
Track type	video	436	3	1
Default duration	00:00:00.033366666 (29.970 frames/fields ...	439	8	4
Video track		447	35	26
Pixel width	960	456	4	2
Pixel height	540	460	4	2
Interlaced	2	464	3	1
Display unit	4	467	4	1
Video color information		471	11	8
Horizontal chroma siting	1	474	4	1
Vertical chroma siting	2	478	4	1
Codec's private data	size 41 (H.264 profile: High @L3.1)	482	44	41
Tags		526	370	358
Cluster		896	495,549	495,537
Cluster		496,445	327,602	327,590
Cluster		824,047	494,965	494,953
Cues		1,319,012	54	42

How to authenticate camera devices



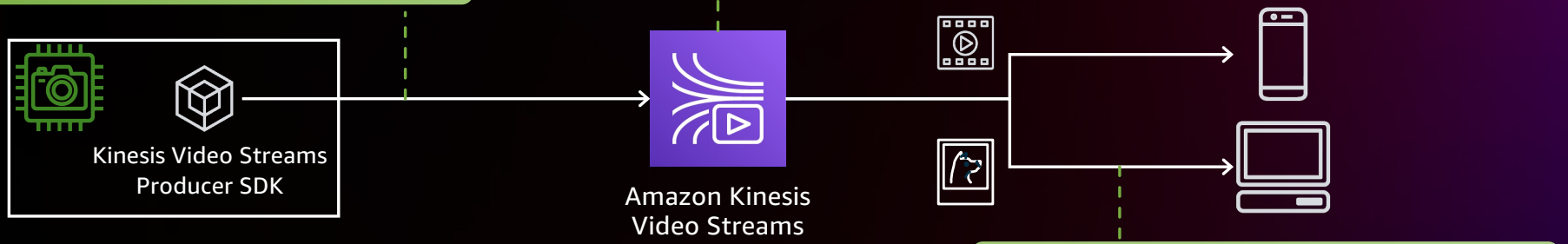
https://docs.aws.amazon.com/iot/latest/developerguide/authorizing-direct-aws.html

Store and index media

Streams

Encrypt and store the media
Set retention for each stream (0~10 years)

Upload media fragments
(1~10 seconds) via HTTPS



Extract or play back media via
HTTPS-based protocols
(APIs, HLS, and MPEG-DASH)

Kinesis Video Streams – Playback APIs

GetDataEndpoint	Returns a data endpoint
GetClip	Create MP4 File
GetDASHStreamingSessionURL GetHLSStreamingSessionURL	Live streaming or playback via HLS / DASH

These APIs have requirements:

The media must contain H.264 or H.265 encoded video and, optionally, AAC or G.711 encoded audio. Specifically, the codec ID of track 1 should be V_MPEG/ISO/AVC (for h.264) or V_MPEGH/ISO/HEVC (for H.265). Optionally, the codec ID of track 2 should be A_AAC (for AAC) or A_MS/ACM (for G.711).

APIs for Media Playback

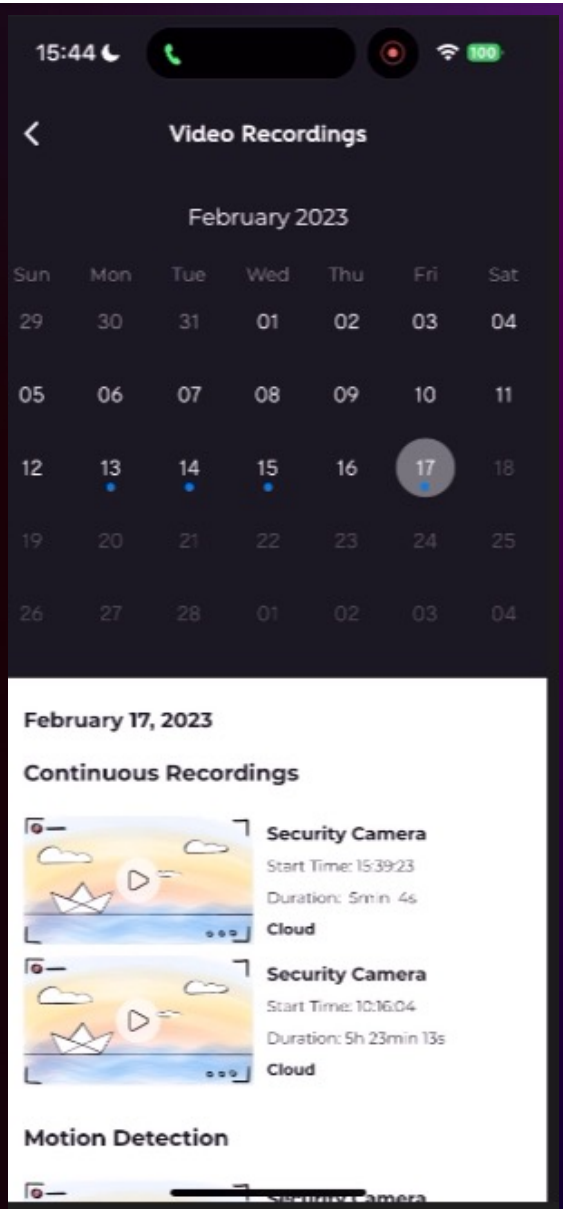
MP4 Clips can be downloaded by calling these APIs

1. GetDataEndpoint
2. GetClip

Alternatively, clips could be played back via HLS or DASH:

- HLS Playback API
- DASH Playback API

- See the following example for reference: https://catalog.us-east-1.prod.workshops.aws/kinesis-video-streams/en-US/3-stream/3-mp4-clip

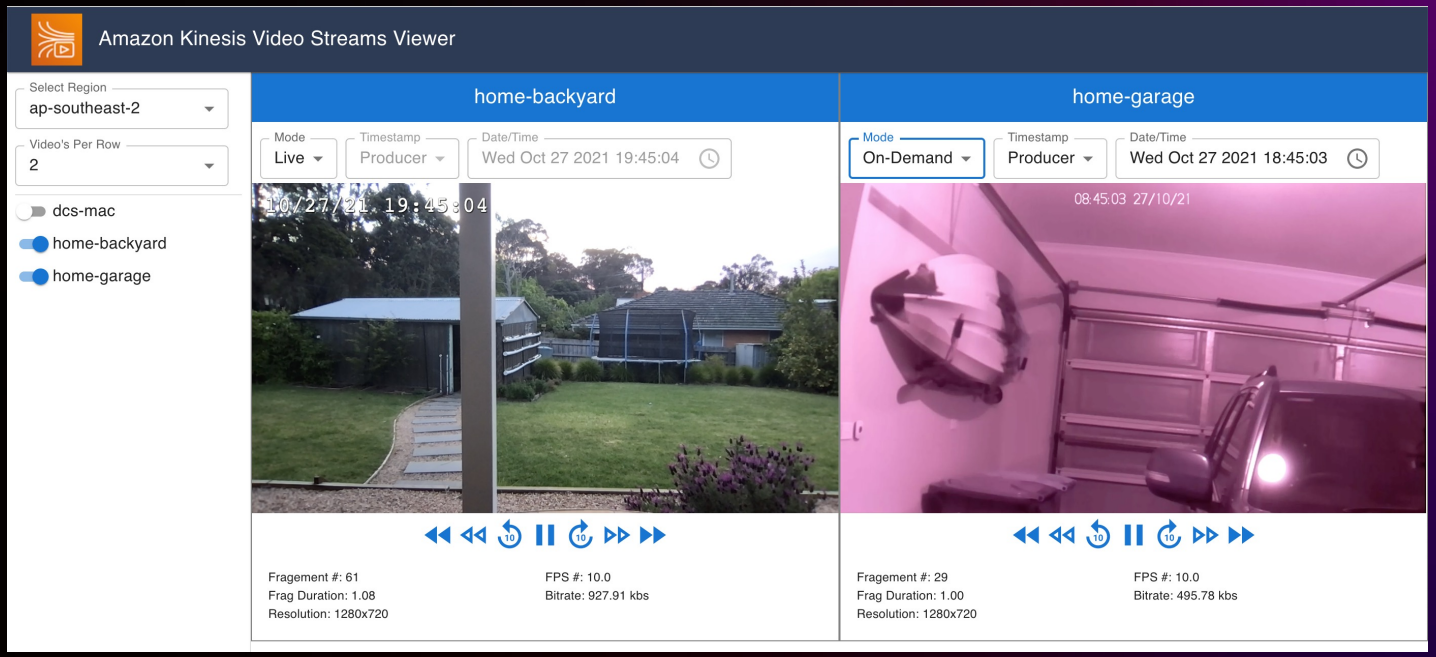


Consume - Play back media

Live and on-demand playback via HLS and DASH protocols

Support for multiple audio and video encoding formats

Open-source sample viewer app is available on GitHub



Kinesis Video Streams – Parser APIs

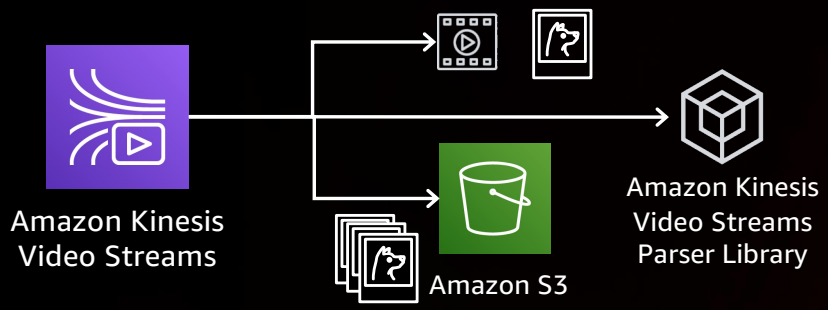
GetDataEndpoint	Returns a data endpoint
GetMedia GetMediaForFragmentList ListFragments	Returns MKV encapsulated media

Consume- Extract and analyze media

Streams

Extract media

- MP4 clip or image frame
- Export images to S3 periodically
- Build custom media processing applications



Analyze media

- Real-time video analysis with Amazon Rekognition Video
- ML inference with Amazon SageMaker
  - Deep Learning AMIs
  - Amazon Kinesis Video Inference template for Sagemaker

Face detection and recognition

Person, pet, and package detection

Image inference with custom ML model

Kinesis Video Streams Parser Library

OPEN SOURCE JAVA LIBRARY THAT MAKES IT EASY TO WORK WITH GETMEDIA OUTPUT

- Get a frame-level object and its associated metadata
- Extract and apply video fragment-specific metadata
- Merge consecutive fragments
- Decode media to JPEG/ PNG (needed for invocation by ML-algos)
- Build into your custom ML or other video-processing applications
- Scale to process 1000’s of streams concurrently