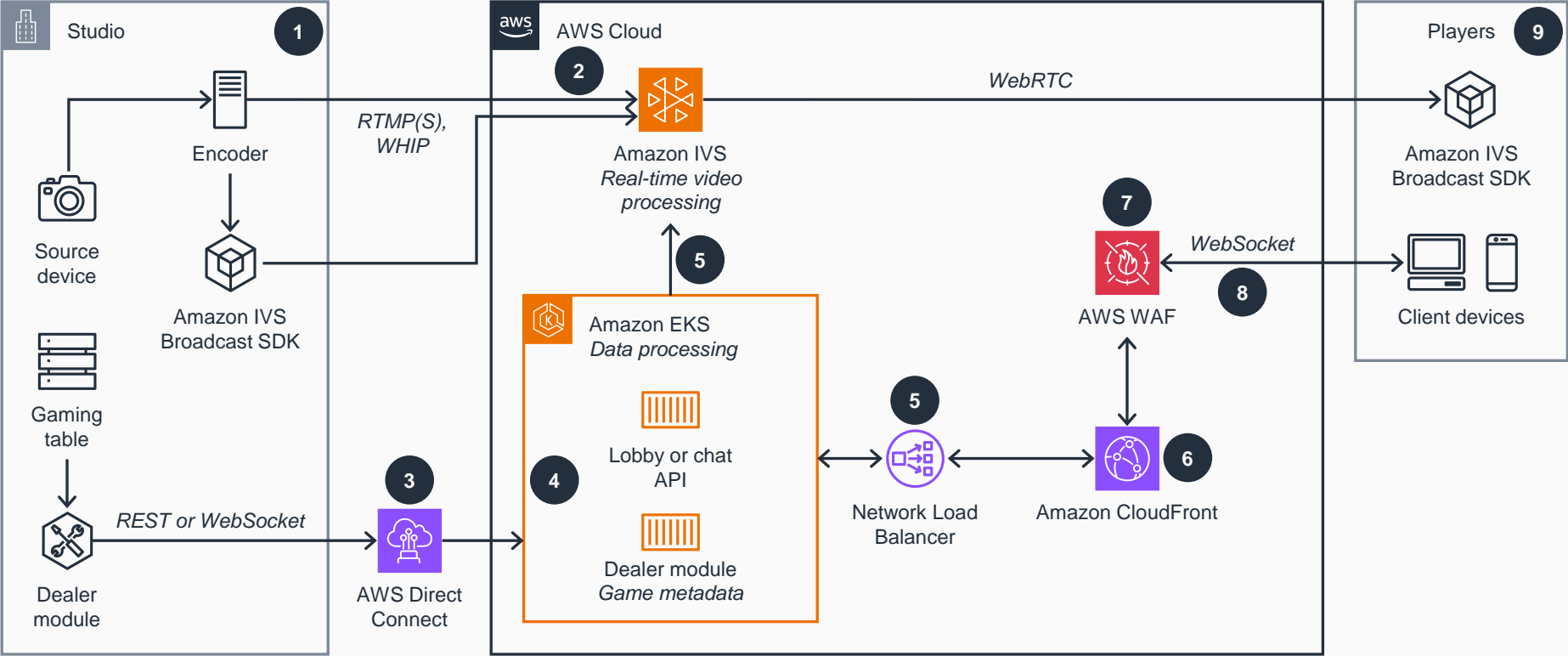


Guidance for Optimizing Livestreaming for Betting and Gaming on AWS

Less than 300 millisecond latency

This architecture diagram shows how to build a betting and gaming architecture to deliver video and data streams with lower than 300ms latency using Amazon Interactive Video Service (Amazon IVS).

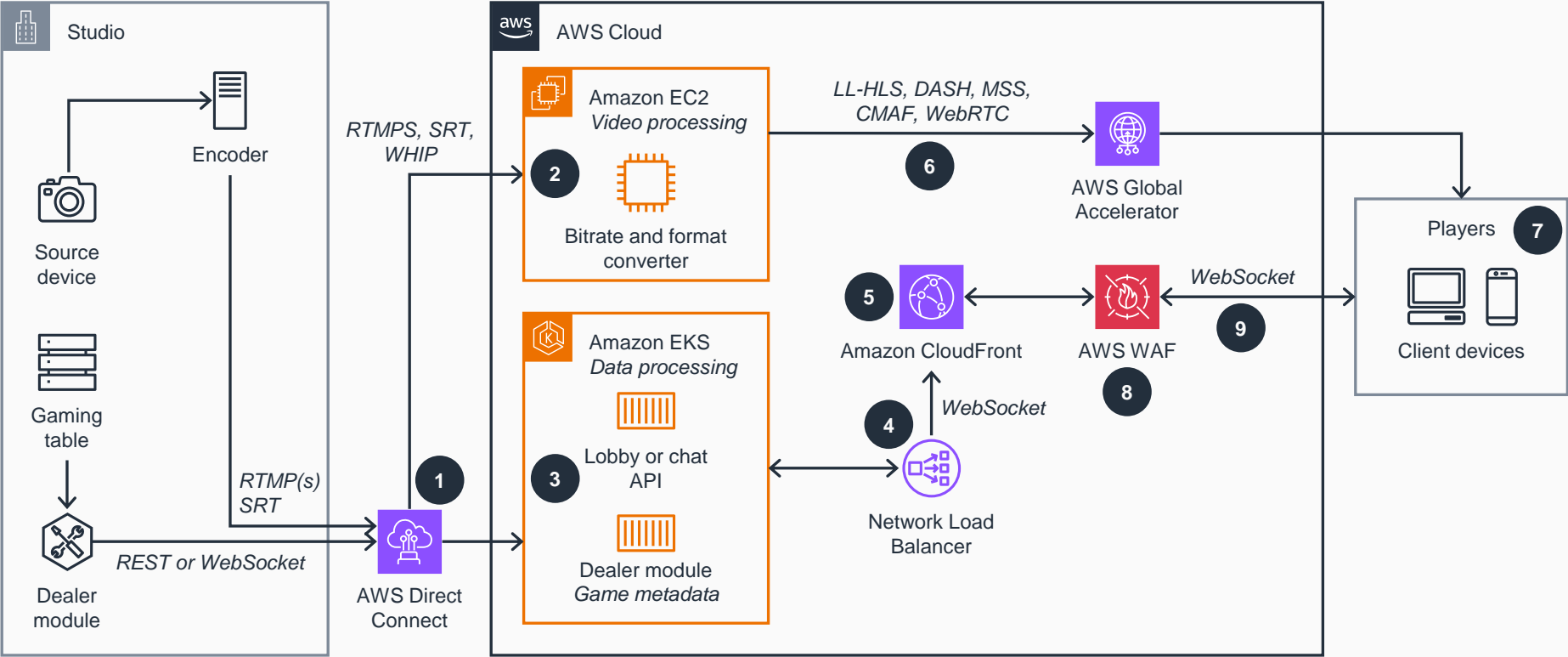
- 1 Live video is captured from mobile or desktop source devices integrated with the **Amazon Interactive Video Service (Amazon IVS) Real-Time Streaming** broadcast SDK. It is then encoded and sent as an input to an **Amazon IVS** stage. An encoder device or an application like OBS Studio can also send video.
- 2 **Amazon IVS** receives the encoded video through Real Time Messaging Protocol (RTMP) over a TLS/SSL connection (RTMPS), WebRTC-HTTP ingestion protocol (WHIP), or WebRTC.
- 3 Game metadata is sent to a REST API layer for additional storage and processing and to a player. **AWS Direct Connect** uses a dedicated connection for a secure and low-latency data (REST or WebSocket) transfer between your on-premises studio and AWS.
- 4 **Amazon Elastic Kubernetes Service (Amazon EKS)** processes REST APIs and WebSockets for game metadata, lobby or chat, and API functionality. It handles players' API calls and maintains WebSocket connections to players for dealer module messages.
- 5 **Network Load Balancer** offers ultralow latencies for latency-sensitive applications to process API communication between players and the application. The timed metadata is sent to the **Amazon IVS** endpoint.
- 6 **Amazon CloudFront** acts as an endpoint for inbound data flow and customer API requests during the video stream.
- 7 **AWS WAF** protects the endpoints and APIs from distributed denial of service (DDoS) attacks.
- 8 Players maintain WebSocket connections to the backend API for lobby or chat and game action calls.
- 9 The **Amazon IVS** player on the client device receives the video stream and metadata. The video and metadata are sent to client applications running the **Amazon IVS** broadcast SDK through WebRTC. Your viewers can watch live streams globally through the **Amazon IVS** content delivery network. The **Amazon IVS** broadcast SDK optimizes performance, reducing the impact on your app and on user devices.



Guidance for Optimizing Livestreaming for Betting and Gaming on AWS

Sub 2 second latency

This architecture diagram shows how to build a betting and gaming architecture to deliver video and data streams with sub 2 second latency using Amazon IVS.



- 1 **Direct Connect** uses a dedicated connection for secure and low-latency data (REST or WebSocket) and video (RTMPS or SRT) transfer between your on-premises studio and AWS.
- 2 Applications based on **Amazon Elastic Compute Cloud (Amazon EC2)**, like Nginx and Ffmpeg, process video and transcode it into adaptive bitrate streaming-supported HLS, LL-HLS, CMAF, and DASH protocols. **Amazon EC2** is chosen over containers to reduce any potential network impact on video latency.¹
- 3 **Amazon EKS** processes REST APIs and WebSockets for game metadata, lobby or chat, and API functionality. It handles players' API calls and maintains a WebSocket connection to players for dealer module messages.
- 4 **Network Load Balancer** offers ultralow latencies for latency-sensitive applications and distributes the traffic for API metadata services to enable scaling and route traffic to healthy endpoints.
- 5 **CloudFront** distribution delivers your live stream to viewers with low latency and high transfer speeds by using http-based protocols like HLS, LL-HLS, DASH, and CMAF. It also acts as an endpoint for customer API requests during the video stream.
- 6 **AWS Global Accelerator** distributes LL-HLS, DASH, maximum segment size (MSS), WebRTC, or CMAF video traffic to consumers.²
- 7 The consumer receives the data and video streams through public endpoints and performs synchronization using video timestamps.
- 8 **AWS WAF** protects the endpoints and APIs from DDoS attacks.
- 9 Players maintain WebSocket connections to the backend API for lobby or chat and game action calls.

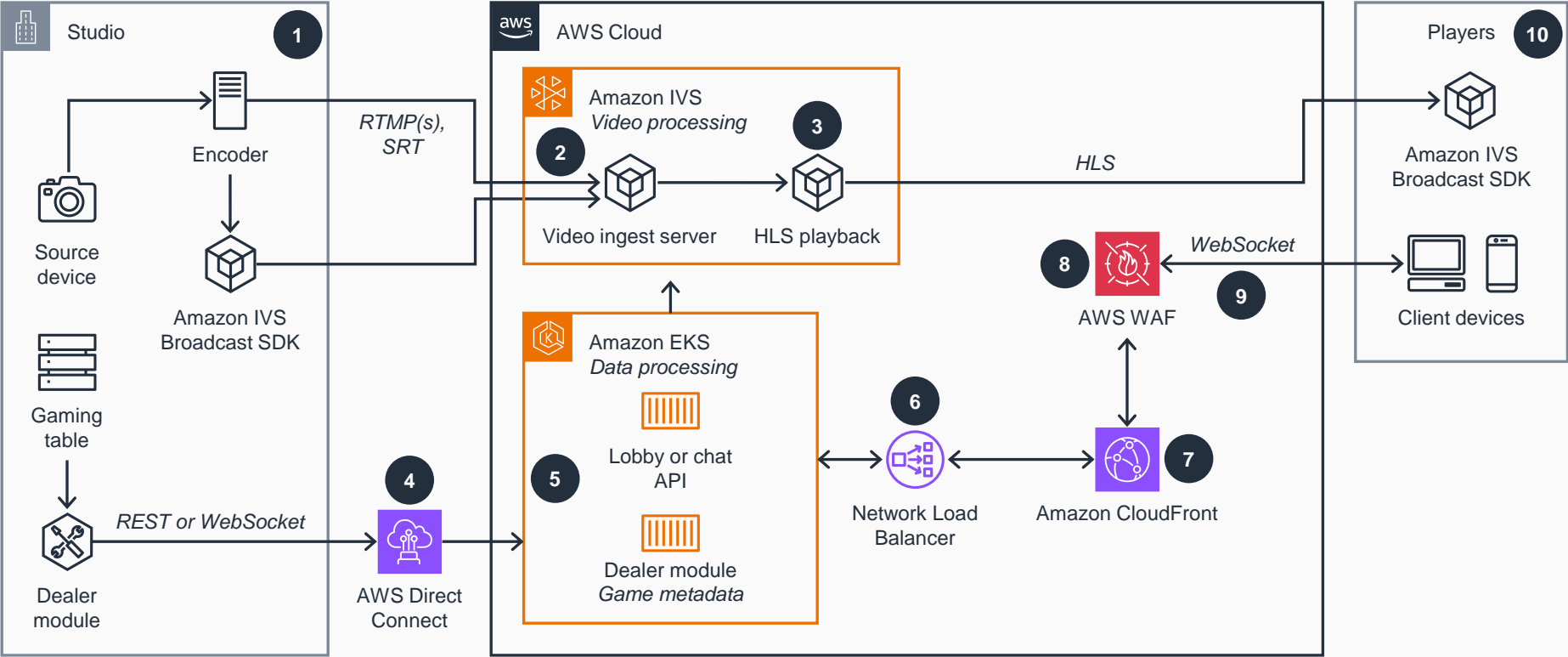
¹ Run Open Source FFMPEG at Lower Cost and Better Performance on a VT1 Instance for VOD Encoding Workloads
² Improving Real-Time Communication (RTC) Client Experience with AWS Global Accelerator

Guidance for Optimizing Livestreaming for Betting and Gaming on AWS

2–5 second latency

This architecture diagram shows how to build a betting and gaming architecture to deliver video and data streams with 2-5 second latency using Amazon IVS.

- 1 Live video is captured from mobile or desktop source devices integrated with the **Amazon IVS** broadcast SDK. It is then encoded and sent as an input to an **Amazon IVS** video ingest server. An encoder device or an application like OBS Studio can also send video.
- 2 **Amazon IVS** handles video processing (ingesting and transcoding) and synchronizes the metadata with the audio and video frames. As a result, all viewers get the metadata at the same time, relative to the stream.
- 3 **Amazon IVS** HTTP Live Streaming (HLS) playback delivers the live stream and metadata to the **Amazon IVS** player on a client's device through the HLS protocol.
- 4 **Direct Connect** uses a dedicated connection for a secure and low-latency data (REST or WebSocket) transfer between your on-premises studio and AWS.
- 5 **Amazon EKS** processes REST APIs and WebSockets for game metadata, lobby or chat, and API functionality. It handles players' API calls and maintains WebSocket connections to players for dealer module messages.
- 6 **Network Load Balancer** offers ultralow latencies for latency-sensitive applications to process API communication between players and the application. The timed metadata is sent to the **Amazon IVS** endpoint.
- 7 **CloudFront** acts as an endpoint for inbound data flow and customer API requests during the video stream.
- 8 **AWS WAF** protects endpoints and APIs from DDoS attacks.
- 9 Players maintain WebSocket connections to the backend API for lobby or chat and game action calls.
- 10 The **Amazon IVS** player on the client device receives the video stream and metadata. Your viewers can watch live streams globally through the **Amazon IVS** content delivery network. The **Amazon IVS** broadcast SDK optimizes performance, reducing the impact on your app and on user devices.



Guidance for Optimizing Livestreaming for Betting and Gaming on AWS

3–5 second latency

This architecture diagram shows how to build a betting and gaming architecture to deliver video and data streams with 3-5 second latency using Amazon IVS.

- 1 Live video is captured from mobile or desktop source devices, encoded, and sent as an input to **AWS Elemental MediaLive**.
- 2 **Direct Connect** uses a dedicated connection for secure and low-latency data (REST or WebSocket) and video (RTMPS or Secure Reliable Transport (SRT)) transfer between your on-premises studio and AWS.
- 3 The video is ingested to **MediaLive** through the provided endpoint with the following protocols supported: URL_PULL, RTMP_PUSH, RTMP_PULL, and RTP_PUSH.
- 4 **AWS Elemental MediaPackage** ingests the **MediaLive** adaptive bitrate output and packages the live stream into HLS, Dynamic Adaptive Streaming over HTTP (DASH), and Common Media Application Format (CMAF) protocols. These are then delivered to three **MediaPackage** custom endpoints.¹
- 5 **Amazon EKS** processes the REST APIs and WebSockets for game metadata, lobby or chat, and API functionality. It handles players' API calls and maintains WebSocket connections to players for dealer module messages.
- 6 **Network Load Balancer** offers ultralow latencies for latency-sensitive applications and distributes the traffic for API metadata services to enable scaling and route traffic to healthy endpoints.
- 7 **CloudFront** acts as an endpoint for inbound data flow and customer API requests during the video stream. A **CloudFront** distribution uses the **MediaPackage** custom endpoints as its origin and delivers your live stream to viewers using http-based protocols such as HLS, low-latency HLS (LL-HLS), DASH, and CMAF.
- 8 **AWS WAF** protects endpoints and APIs from DDoS attacks.
- 9 Players maintain WebSocket connections to the backend API for lobby or chat and game action calls.

¹ Live Streaming on AWS

