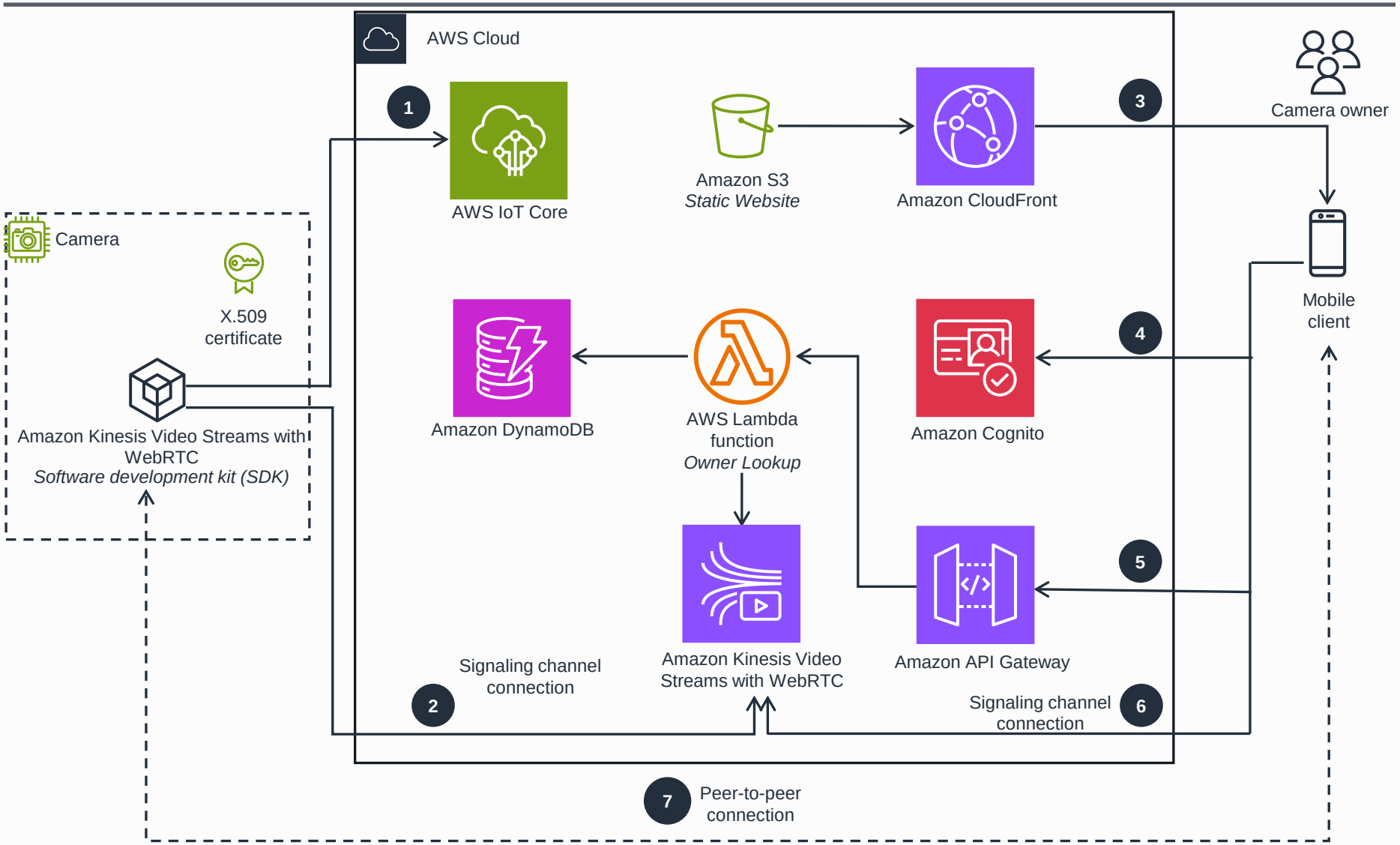
Guidance for Deploying Smart Cameras Using Amazon Kinesis Video Streams with WebRTC

Live View Smart Home Surveillance Cameras

This architecture diagram shows how use Kinesis Video Streams for WebRTC, allowing for the low-latency, bi-directional, peer-to-peer streaming of video from home surveillance cameras.



1. The camera authenticates to **AWS IoT Core** using the Credentials Provider.
2. The **Amazon Kinesis Video Streams** software development kit (SDK) opens signaling to the signaling channel, awaiting viewer requests**.**
3. **Amazon CloudFront** is the content delivery network (CDN) for the static single-page application (SPA) hosted in **Amazon Simple Storage Service** (Amazon S3).
4. The camera owner logs in to the mobile client. **Amazon Cognito** and **Amazon API Gateway** manage user access for the mobile client.
5. The mobile client calls an API to look up the cameras belonging to the owner. The owner then selects a specific camera to initiate real-time video streaming.
6. The mobile client initiates a connection to the **Kinesis Video Streams** signaling channel, a WebSocket connection that the camera is listening to for viewer connection requests. The viewer and the camera negotiate connection details by sending messages to each other over the signaling channel.
7. The mobile client and the camera establish a peer-to-peer connection. The **Kinesis Video Streams** client streams video data through the WebRTC peer connection, and the camera owner views the video stream in the mobile client. Optionally, the WebRTC data channel can be utilized for bi-directional command and control messages.

   If a direct, peer-to-peer connection cannot be established, video data can be relayed through the **Kinesis Video Streams** Traversal Using Relays around NAT (TURN) server.

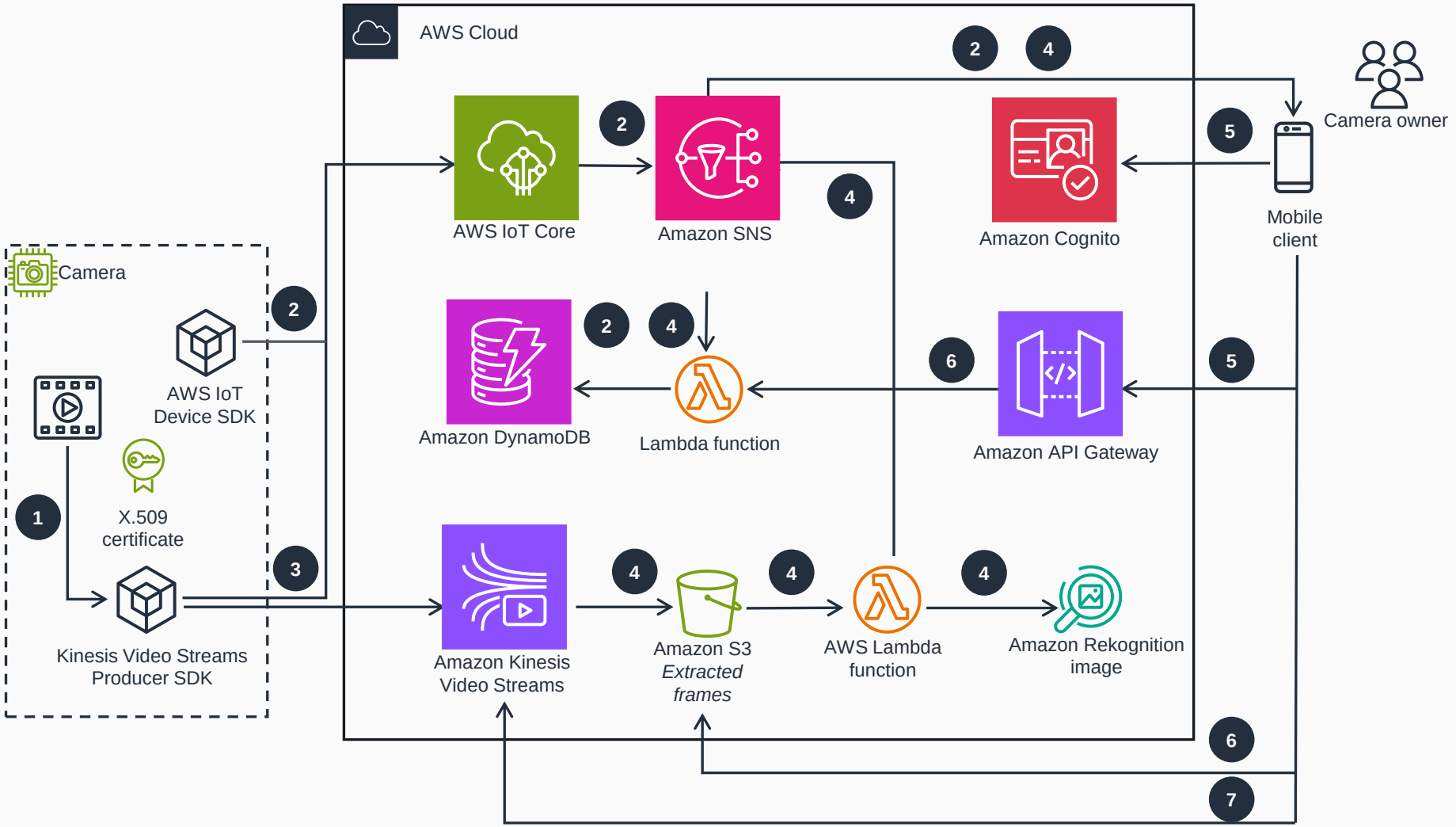
Reviewed for technical accuracy Month Day, 2022

AWS Reference Architecture

Guidance for Deploying Smart Cameras Using Amazon Kinesis Video Streams with WebRTC

Smart Home Surveillance Camera

This architecture diagram shows how to stream video from smart home cameras to Kinesis Video Streams for storage, playback, and analytics. It also shows how to send push notifications to mobile application users to notify them of events.



1. The camera detects an event and begins recording video and audio using local storage.
2. The camera application publishes an event to **AWS IoT Core**. A rule is configured in Rules for IoT Core with an action to publish an **Amazon Simple Notification Service** (Amazon SNS) message, triggering a push notification to the end user's mobile application. An **AWS Lambda** function subscribes to the **Amazon SNS** topic to save events in an **Amazon DynamoDB** table.
3. The **Kinesis Video Streams** Producer SDK uses the X.509 certificate to authenticate to **AWS IoT Core** using the credentials provider. It then streams media to a stream belonging to the camera for storage, playback, and analytics.
4. **Kinesis Video Streams** is configured with the **Amazon S3** delivery feature to automatically extract still images from the uploaded media to deliver to an **Amazon S3** bucket of your choice. An **Amazon S3** trigger is configured to invoke a **Lambda** function, which will invoke an **Amazon Rekognition** image to perform object detection and trigger notifications in **Amazon SNS**.
5. The camera owner logs in to the mobile client. **Amazon Cognito** and **API Gateway** manage user access for the mobile client.
6. The mobile client application invokes an API to retrieve a list of cameras and view a list of events. The events list view uses the images stored in **Amazon S3** as preview thumbnails for viewing the stored media.
7. The mobile application user selects an event to replay and uses Kinesis Video Streams playback to replay the stored video and audio.

Reviewed for technical accuracy Month Day, 2022

AWS Reference Architecture