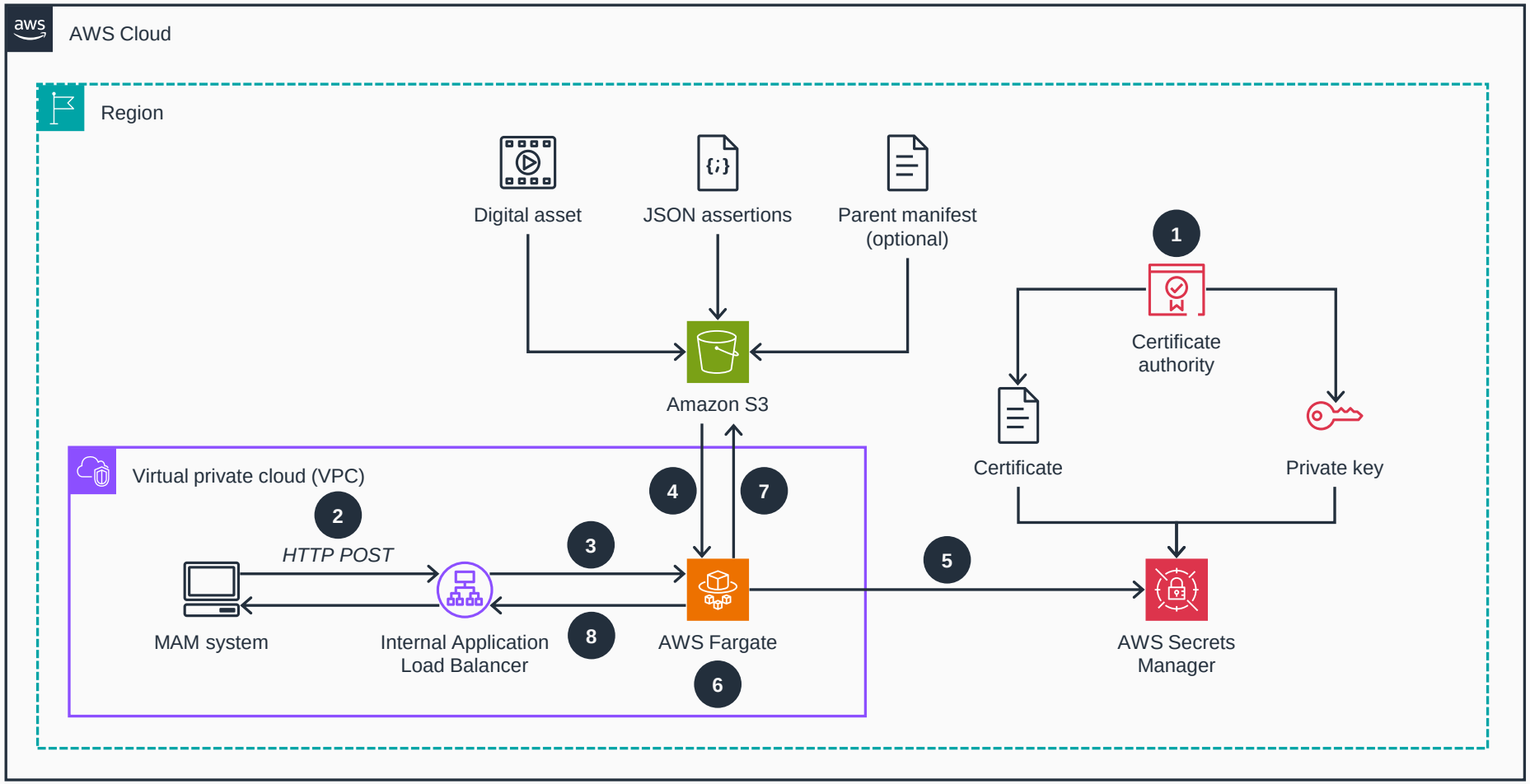
# Guidance for Media Provenance with C2PA on AWS

## AWS Fargate

This architecture diagram shows how you can generate a Coalition for Content Provenance and Authenticity (C2PA) manifest sidecar file for a media workload in your AWS account using AWS Fargate. This also works with AWS Lambda, as shown on the next slide.




1. The certificate and private key are obtained from the certificate authority.
2. The media asset management (MAM) system sends an HTTP POST to the internal Application Load Balancer. The request parameters include presigned URLs for a digital asset, a JSON assertions file, and a parent C2PA manifest (if applicable), stored in **Amazon Simple Storage Service (Amazon S3)**. The caller can also provide the JSON assertions in the request body rather than by URL.
3. The Application Load Balancer forwards the POST request to an **AWS Fargate** task running a FastAPI application.
4. The **Fargate** task uses the presigned URLs to download the digital asset, JSON assertions file, and parent C2PA manifest from **Amazon S3** to its attached ephemeral storage.
5. **Fargate** retrieves your digital certificate and private key from **AWS Secrets Manager** and stores the values in environmental variables.
6. Using the open-source C2PA tool, **Fargate** creates a C2PA manifest and generates the signature block by retrieving the digital certificate and private key values from environmental variables.
7. **Fargate** uploads the generated C2PA manifest sidecar to the **Amazon S3** bucket.
8. **Fargate** returns a presigned URL to the MAM system for the C2PA manifest stored in **Amazon S3**.

*Reviewed for technical accuracy September 10, 2024*

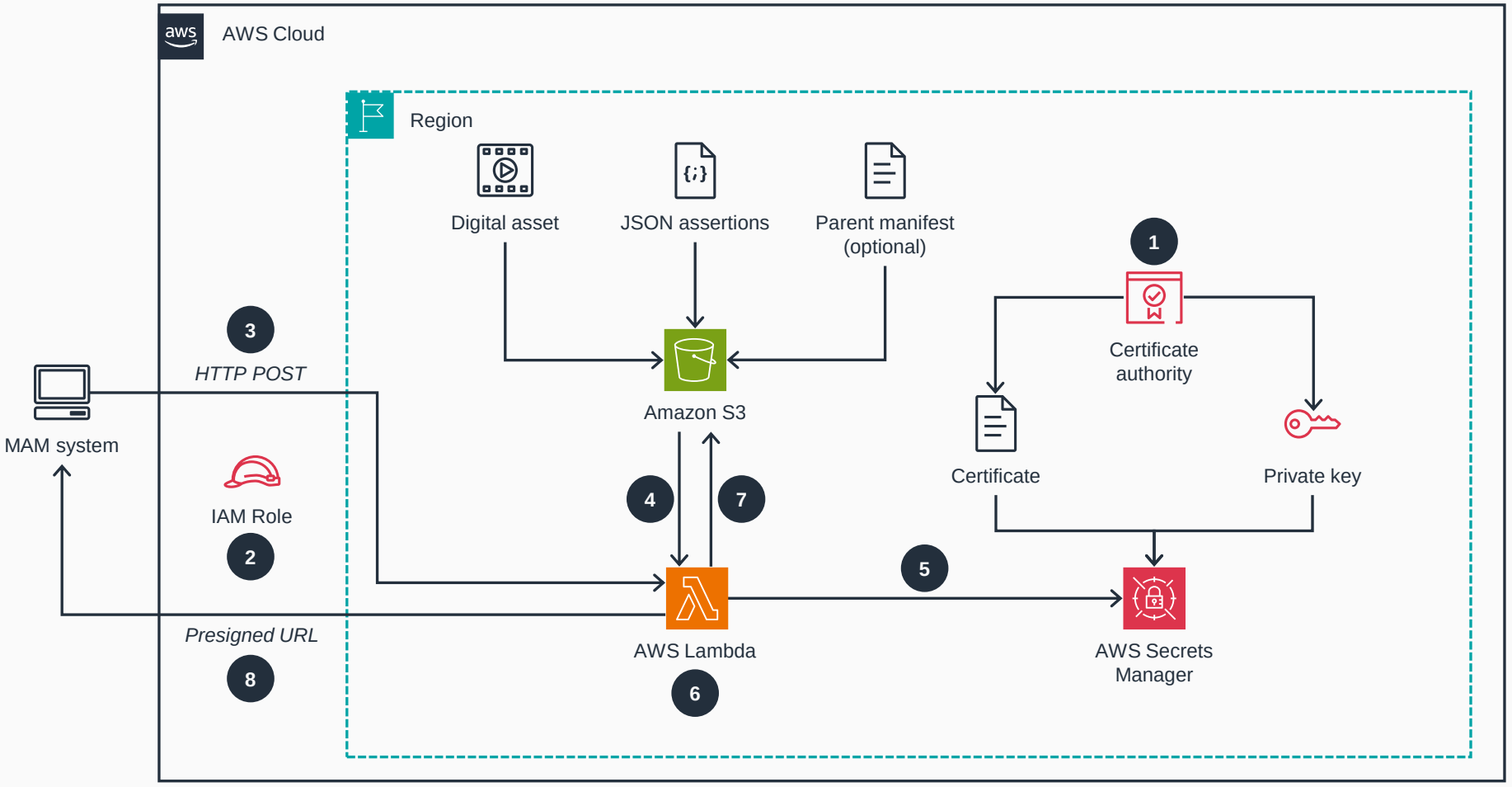


**AWS Reference Architecture**

# Guidance for Media Provenance with the Content Authenticity Initiative (C2PA) on AWS

## AWS Lambda

This architecture diagram shows how you can generate a C2PA manifest sidecar file for a media workload in your AWS account using AWS Lambda.




1. The certificate and private key are obtained from the certificate authority.
2. The MAM system assumes an **AWS Identity and Access Management (IAM)** role to support the invocation of an **AWS Lambda** function.
3. The MAM system sends a POST request to the **Lambda** function URL. The request parameters include presigned URLs for a digital asset, a JSON assertions file, and a parent C2PA manifest (if applicable), stored in **Amazon S3**. The caller can also provide the JSON assertions in the request body rather than by URL.
4. **Lambda** uses the presigned URLs to download the digital asset, JSON assertions file, and parent C2PA manifest from **Amazon S3** to its attached ephemeral storage.
5. **Lambda** retrieves your digital certificate and private key from **Secrets Manager** and stores the values in environmental variables.
6. Using the open-source C2PA tool, **Lambda** creates a C2PA manifest and generates the signature block by retrieving the digital certificate and private key values from environmental variables.
7. **Lambda** uploads the generated C2PA manifest sidecar to the **Amazon S3** bucket.
8. **Lambda** returns a presigned URL to the MAM system for the C2PA manifest stored in **Amazon S3**.

*Reviewed for technical accuracy September 10, 2024*


**AWS Reference Architecture**