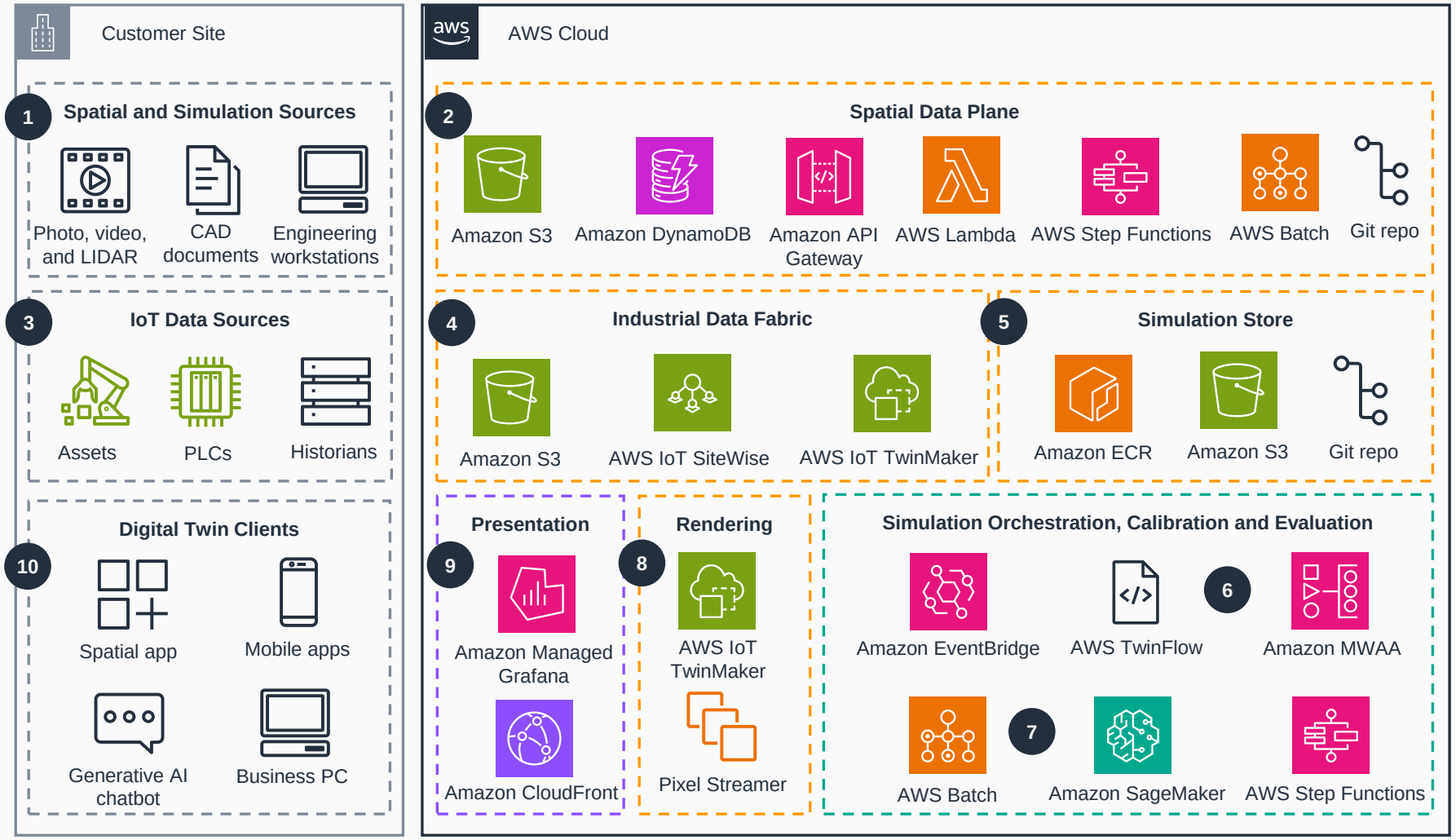
Guidance for Digital Twin Framework on AWS

High-Level Overview

This architecture diagram consists of three integrated modules that address key stages of workforce safety and compliance and create the Digital Twin framework: IoT, spatial compute, and simulation components. This slide includes Steps 1-6.



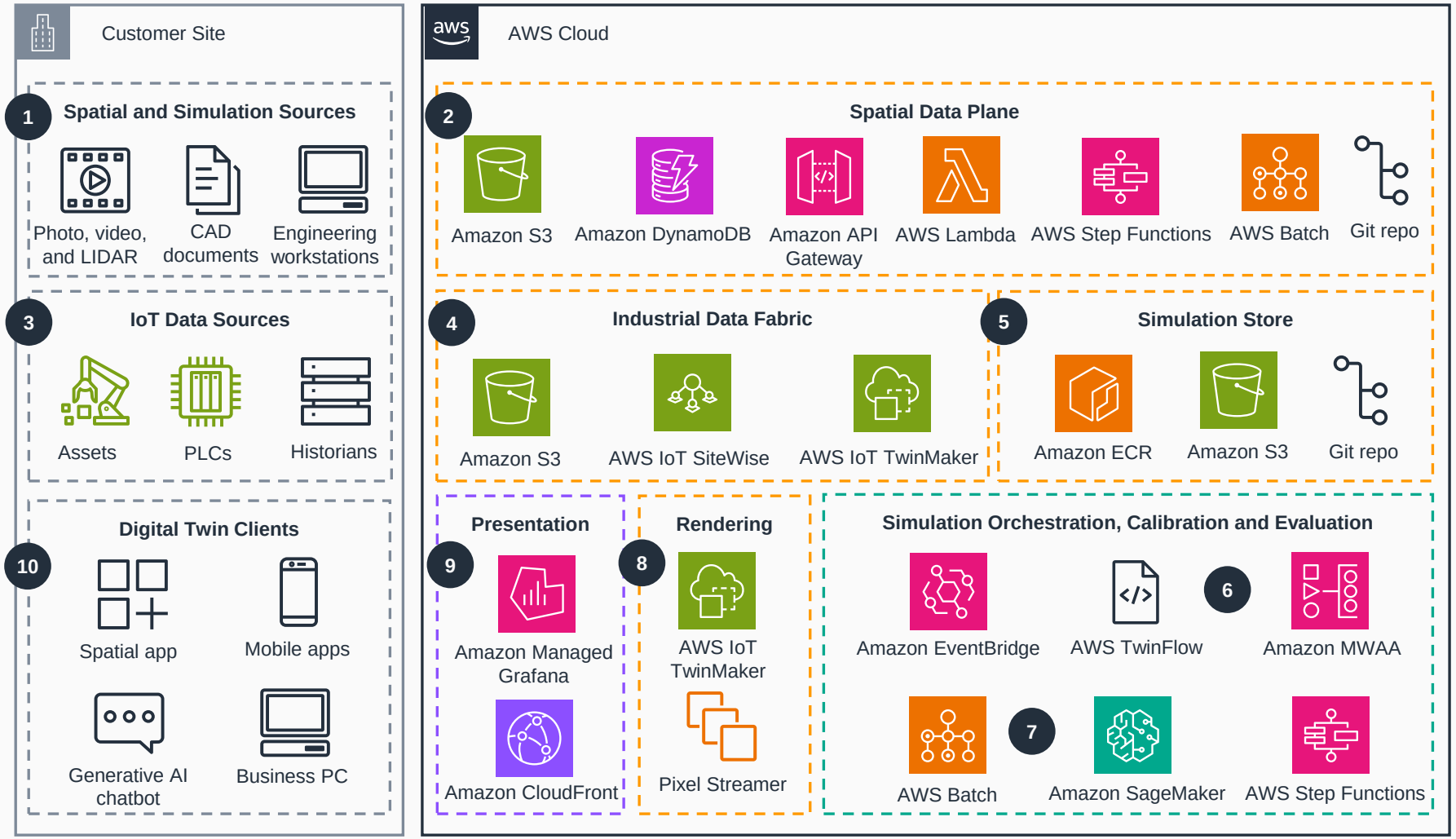
1. Spatial data for the Digital Twin Framework is ingested into a spatial data plane using **Visual Asset Management System (VAMS)** or similar digital asset managers.
2. **Amazon API Gateway** manages the storage of file data in **Amazon Simple Storage Service (Amazon S3)** and metadata in **Amazon DynamoDB**.
3. IoT data to fuel the Digital Twin Framework is ingested using a preferred Industrial Data Fabric solution.
4. **Amazon S3** provides object storage, while **AWS IoT SiteWise** and **AWS IoT TwinMaker** provide structure and semantics to the IoT data.
5. Simulation and AI/ML models created by engineers are stored in AWS. The simulation runtime environments are containerized and stored in **Amazon Elastic Container Registry (Amazon ECR)**. Configuration files and model weights are stored in **Amazon S3**, and the simulation source code is stored in Git repositories.
6. Simulations workflows are managed with orchestrators like **TwinFlow**, **Amazon Managed Workflows for Apache Airflow (Amazon MWAA)**, and **AWS Step Functions**. Simulations and AI/ML models are evaluated in **AWS Batch** or **Amazon SageMaker** using **Amazon EventBridge** events.

AWS Reference Architecture

Guidance for Digital Twin Framework on AWS

High-Level Overview

This architecture diagram consists of three integrated modules that address key stages of workforce safety and compliance and create the Digital Twin framework: IoT, spatial compute, and simulation components. This slide includes Steps 7-10.



7 Periodically, **EventBridge** will initiate **AWS Batch** processes that re-calibrate the simulation or model with updated data stored in **Amazon S3**, **AWS IoT SiteWise**, or other database solutions such as **Amazon Timestream**.

8 Real-time 3D rendering of the digital twin can be done in the user's browser using **AWS IoT TwinMaker** or can be rendered in the cloud and displayed in the user's browser with WebRTC video streams.

9 Users view digital twins with **Amazon Managed Grafana** or with custom dashboards built with IoT App Kit and hosted with **Amazon CloudFront**.

Additionally, users can interact with virtual workstations using **NICE DCV** remote desktop software or using cloud-hosted applications with **Amazon AppStream 2.0**.

10 Users may opt to view digital twins on mobile devices and AR/VR headsets. Users may also choose to integrate generative AI chatbots into their dashboards.

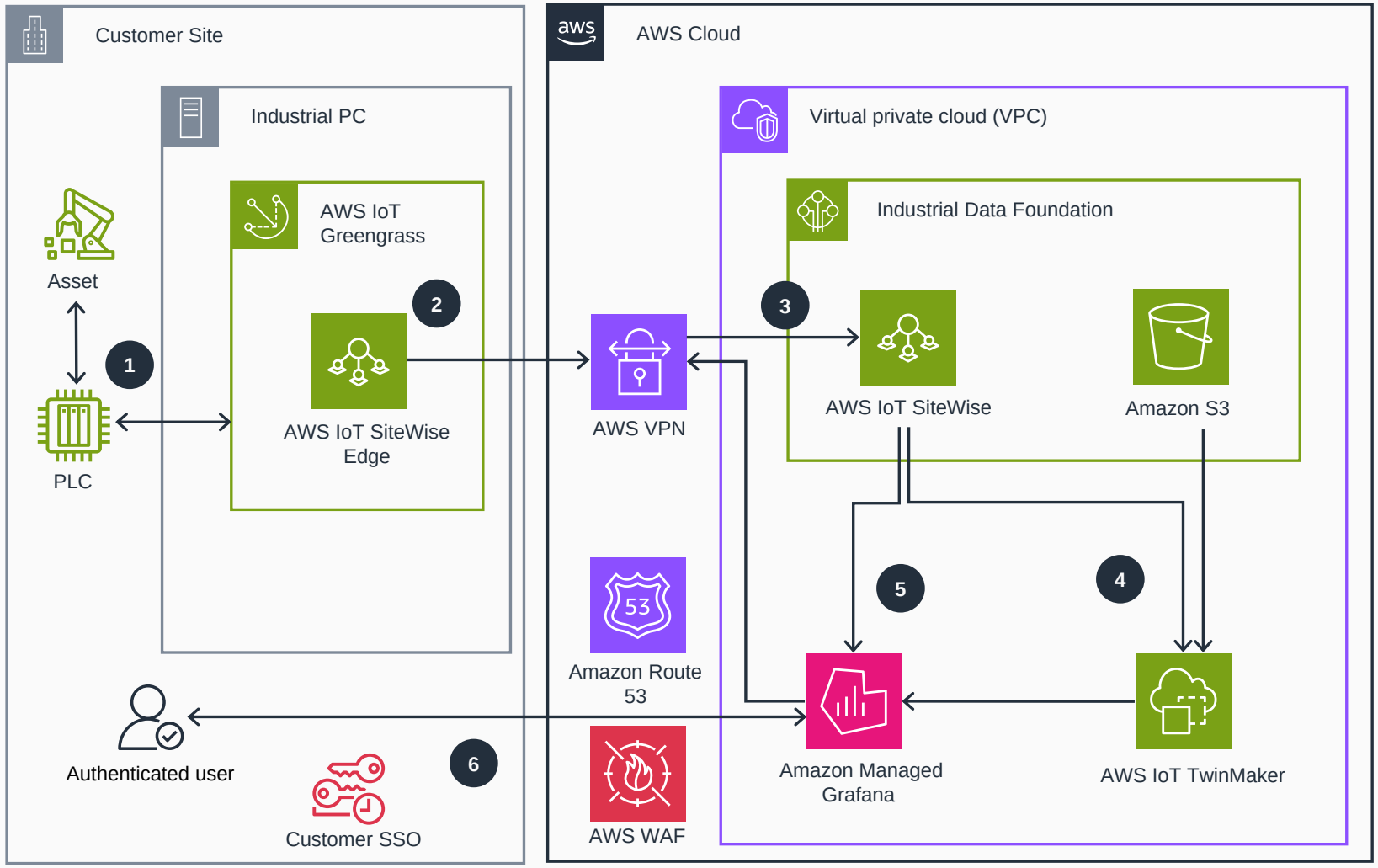
Reviewed for technical accuracy September 4, 2024

AWS Reference Architecture

Guidance for Digital Twin Framework on AWS

IoT Data

This architecture diagram shows how to connect IoT data to the digital twin.



1. The asset connects to an industrial personal computer (PC) at the customer's location through a programmable logic controller (PLC).
2. This PC runs **AWS IoT Greengrass** edge runtime and **AWS IoT SiteWise Edge**. **AWS IoT SiteWise Edge** collects the asset telemetry data and pushes it over a secure connection, such as a virtual private network (VPN), to **AWS IoT SiteWise**.
3. **AWS IoT SiteWise** organizes assets in a hierarchy, computes key performance indicators (KPIs), and stores time-series data in different storage tiers.
4. The data in **AWS IoT SiteWise**, **Amazon S3**, and other data sources is connected to **AWS IoT TwinMaker** to build knowledge graphs and 3D digital twins.
5. **Amazon Managed Grafana** provides operational dashboards along with 3D digital twins leveraging **AWS IoT TwinMaker** and **AWS IoT SiteWise** plug-ins for Grafana.
6. Users, authenticated through their company's single sign-on (SSO) federation, can safely observe the dashboards over the public internet.

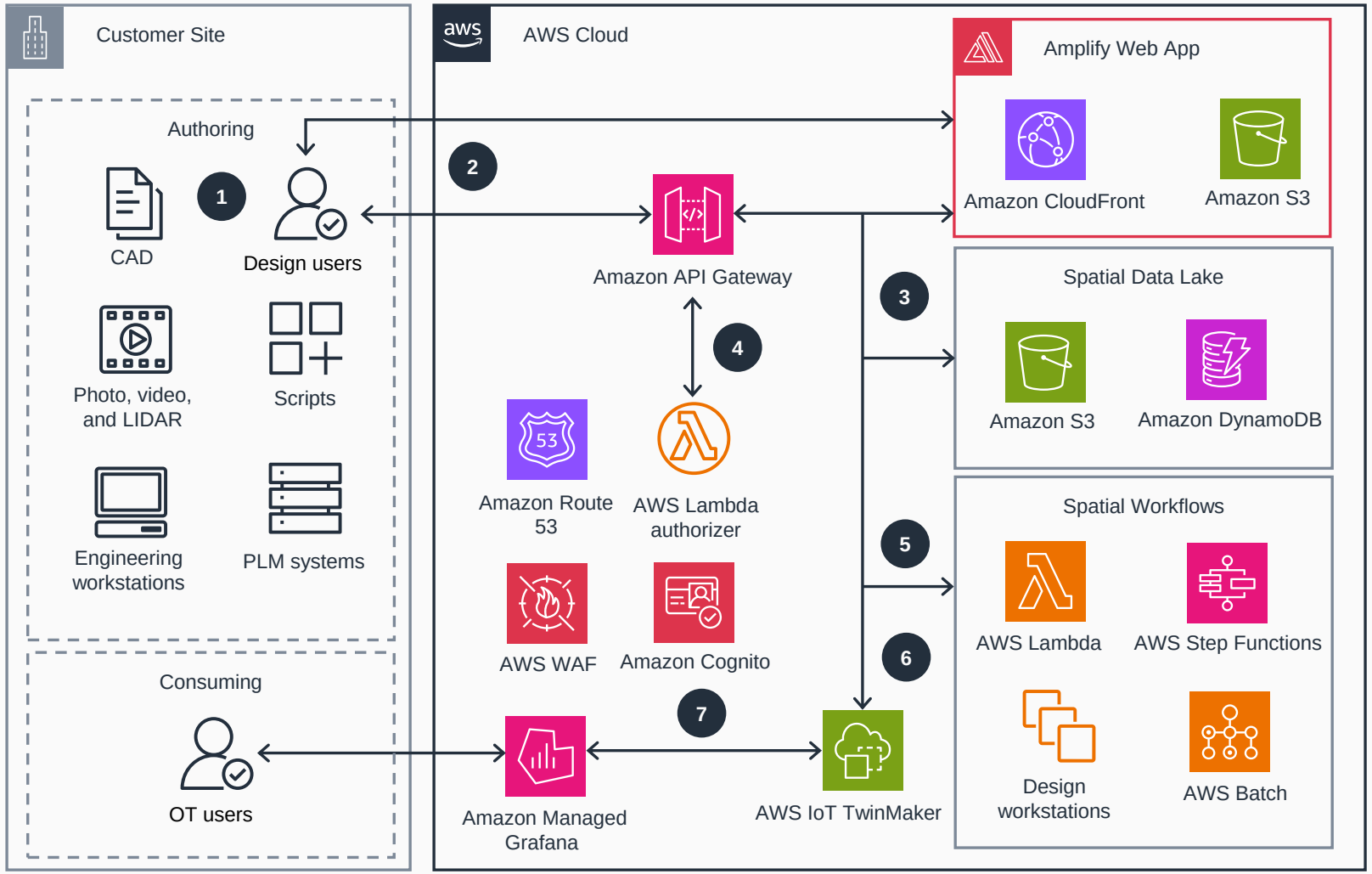
Reviewed for technical accuracy September 4, 2024

AWS Reference Architecture

Guidance for Digital Twin Framework on AWS

Spatial Data Plane

This architecture diagram shows how to create the spatial component of a digital twin, including the ingestion and processing of data into real-time 3D assets.



1. Spatial assets such as computer aided design (CAD) files, photos, videos, and light detection and ranging (LIDAR) point clouds accumulate on-premises in engineering workstations and product lifecycle management (PLM) systems.
2. The design user uploads spatial assets content to the cloud through a custom web application hosted with **Amazon CloudFront** and **Amazon S3**. The web app uses **API Gateway** to integrate with the spatial data plane. The design user may also write custom automation scripts that interact with **API Gateway** directly.
3. **API Gateway** manages uploads of large binary files to **Amazon S3** and stores location, tags, versions, and other metadata in **DynamoDB** tables.
4. **API Gateway** uses **Amazon Cognito** and **AWS Lambda** authorizers to manage role-based access controls (RBAC) and asset-based access controls (ABAC).
5. **API Gateway** also manages the pipelines and workflows that convert source assets into real-time 3D assets used in rendering the digital twin.
6. **AWS IoT TwinMaker** provides an interactive real-time 3D scene component, binding spatial data to industrial IoT data.
7. **Amazon Managed Grafana** provides dashboards that include 3D context alongside charts and graphs, leveraging the **AWS IoT TwinMaker** plug-in for Grafana. It alerts the operational technology (OT) user when there is an exception.

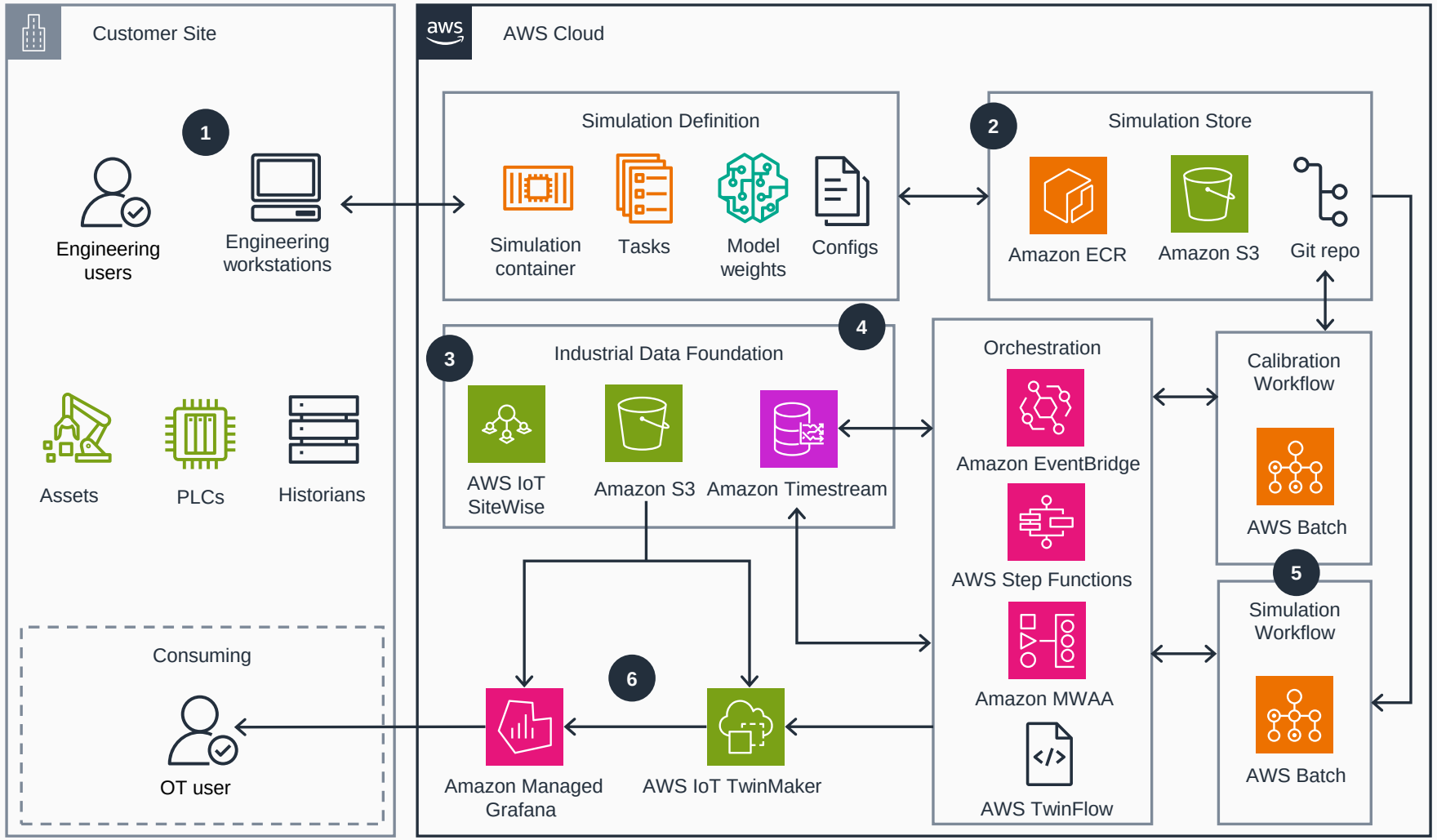
Reviewed for technical accuracy September 4, 2024

AWS Reference Architecture

Guidance for Digital Twin Framework on AWS

Building and Orchestrating Simulation Twins

This architecture diagram shows how to simulate a digital twin.



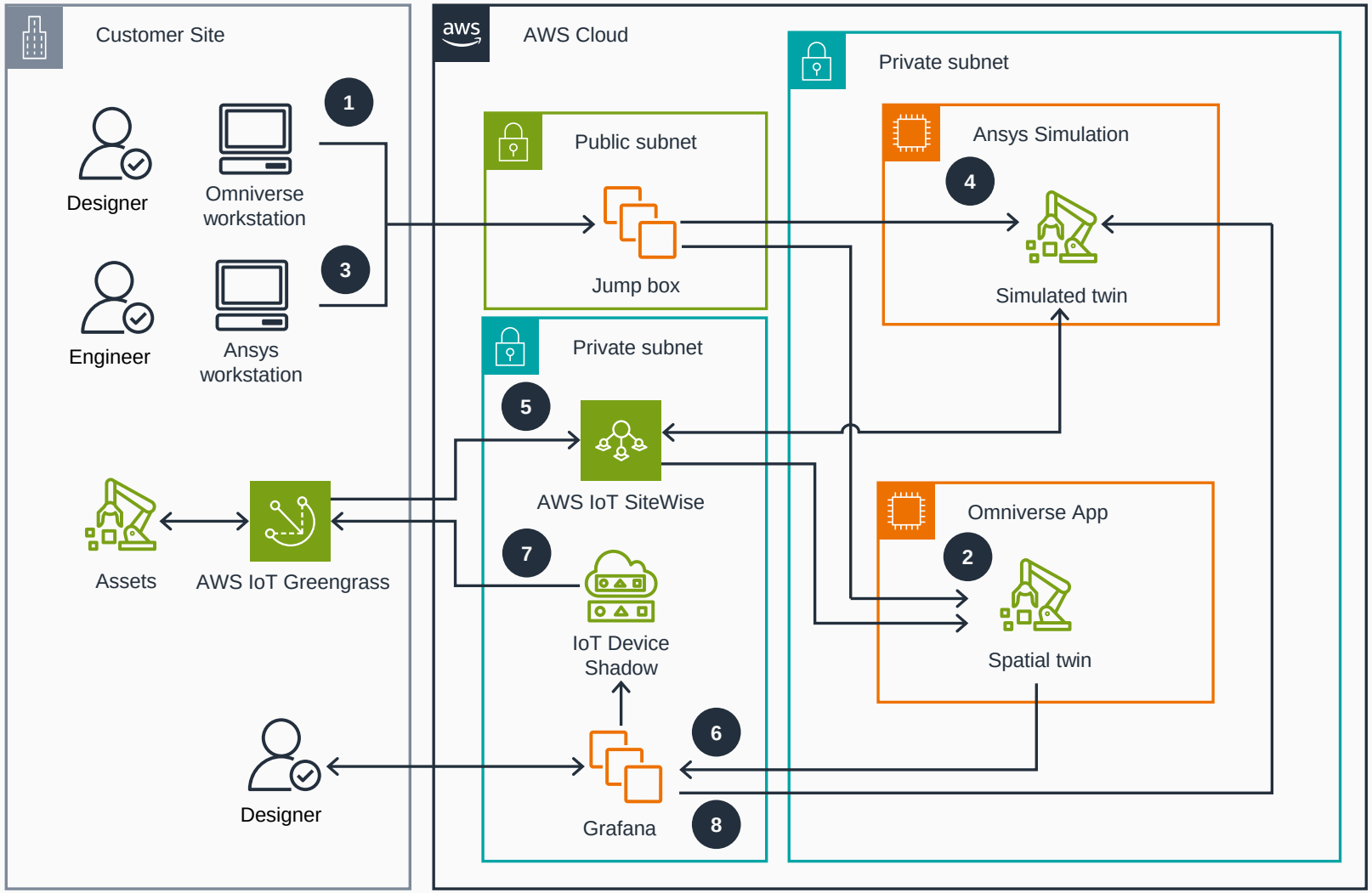
1. An engineer or scientist builds the simulation twin from a local workstation.
2. The engineer builds containerized simulation runtime stored in **Amazon ECR**. Model weights, configuration files, and task definitions are stored in **S3** buckets or Git repositories.
3. The simulations run on data ingested from the industrial edge into an industrial data foundation, commonly a mix of **AWS IoT SiteWise, Amazon S3,** and if real-time capabilities are needed, **Amazon Timestream**.
4. Orchestration of static data, IoT data, models, configuration, execution of simulations, and assessment of calibration requirements is performed by one of many offerings, such as **EventBridge**, **Step Functions**, **Amazon MWAA**, and opensource **TwinFlow**.
5. Periodically, the simulations are re-calibrated to align the predictions with current results and data. Orchestration can invoke **AWS Batch** workflows that update the simulations stored in **Amazon ECR** or model weights stored in **Amazon S3**.
6. The simulations run on demand as scheduled by an orchestrator and feed results back to the industrial data foundation to be presented to users by **AWS IoT TwinMaker** and an **Amazon Managed Grafana** dashboard.

AWS Reference Architecture

Guidance for Digital Twin Framework on AWS

Product Design Example

This architecture diagram shows a specific implementation of the Digital Twin framework used for product design.



1. A design engineer uses a local workstation to author an Omniverse application for the spatial twin of the device.
2. CAD assets are converted to universal scene description (USD) assets. Using a jump box, the CAD assets are installed as an Omniverse application on a GPU instance in a private subnet.
3. The simulation engineer uses a local workstation to author an Ansys Twin Builder Reduced Order Model of the asset.
4. The model is installed using a jump box on a GPU instance in a private subnet.
5. The IoT data from the asset is ingested to **AWS IoT SiteWise** by **AWS IoT Greengrass** edge runtime. The IoT data is consumed by both the simulation twin and the spatial twin.
6. The spatial twin renders into a WebRTC video stream consumed by an iframe within the dashboard of a self-hosted Grafana instance.
7. The industrial designer can control the physical asset from the dashboard. Widgets in the dashboard will initiate changes to an AWS IoT Device Shadow, read by an **AWS IoT Greengrass** component that sends commands to the asset.
8. Similarly, the designer can control the simulation asset. Widgets in the dashboard will initiate the simulation twin, which will send results to **AWS IoT SiteWise** and render into the Omniverse spatial twin video stream on the Grafana dashboard.

Reviewed for technical accuracy September 4, 2024

AWS Reference Architecture