CLOUD ENGINEERING

Cloud Architecture

Ashish Pujari

Lecture Outline

- Architecture Principles
- Cloud Services
- Reference Architectures
- Data Security

ARCHITECTURE PRINCIPLES

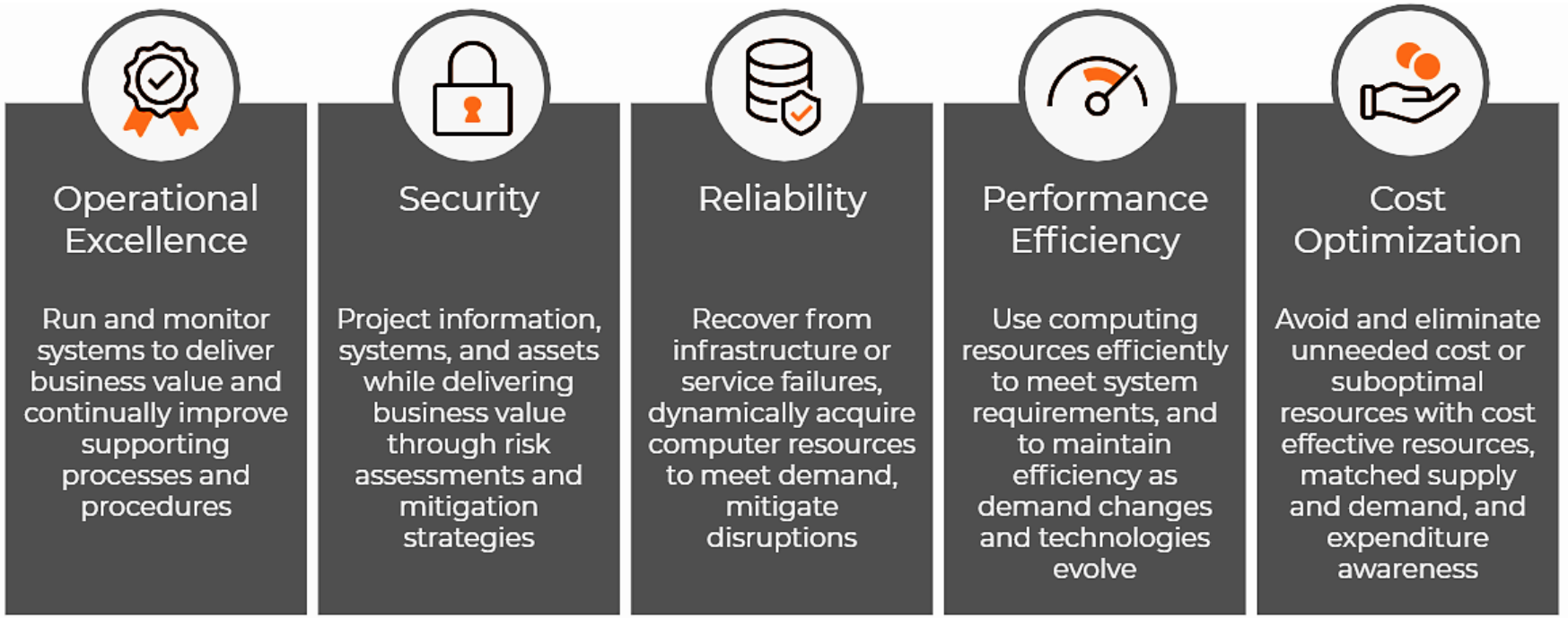
Well Architected Framework

General Design Principles

- Stop guessing your capacity needs
- Test systems at production scale
- Automate to make architectural experimentation easier
- Allow for evolutionary architectures
- Drive architectures using data
- Improve through game days

https://docs.aws.amazon.com/wellarchitected/latest/framework/general-design-principles.html

Well Architected Framework (WAF)



https://aws.amazon.com/architecture/well-architected/

Operational Excellence

- Perform operations as code
- Make frequent, small, reversible changes
- Refine operations procedures frequently
- Anticipate failure
- Learn from all operational failures

https://aws.amazon.com/architecture/well-architected/

Security

- Implement a strong identity foundation
- Enable traceability
- Apply security at all layers
- Automate security best practices
- Protect data in transit and at rest
- Keep people away from data
- Prepare for security events

https://aws.amazon.com/architecture/well-architected/

Reliability

- Automatically recover from failure
- Test recovery procedures
- Scale horizontally to increase aggregate workload availability
- Stop guessing capacity
- Manage change in automation

https://aws.amazon.com/architecture/well-architected/

Performance Efficiency

- Democratize advanced technologies
- Go global in minutes
- Use serverless architectures
- Experiment more often
- Consider mechanical sympathy

https://aws.amazon.com/architecture/well-architected/

Cost Optimization

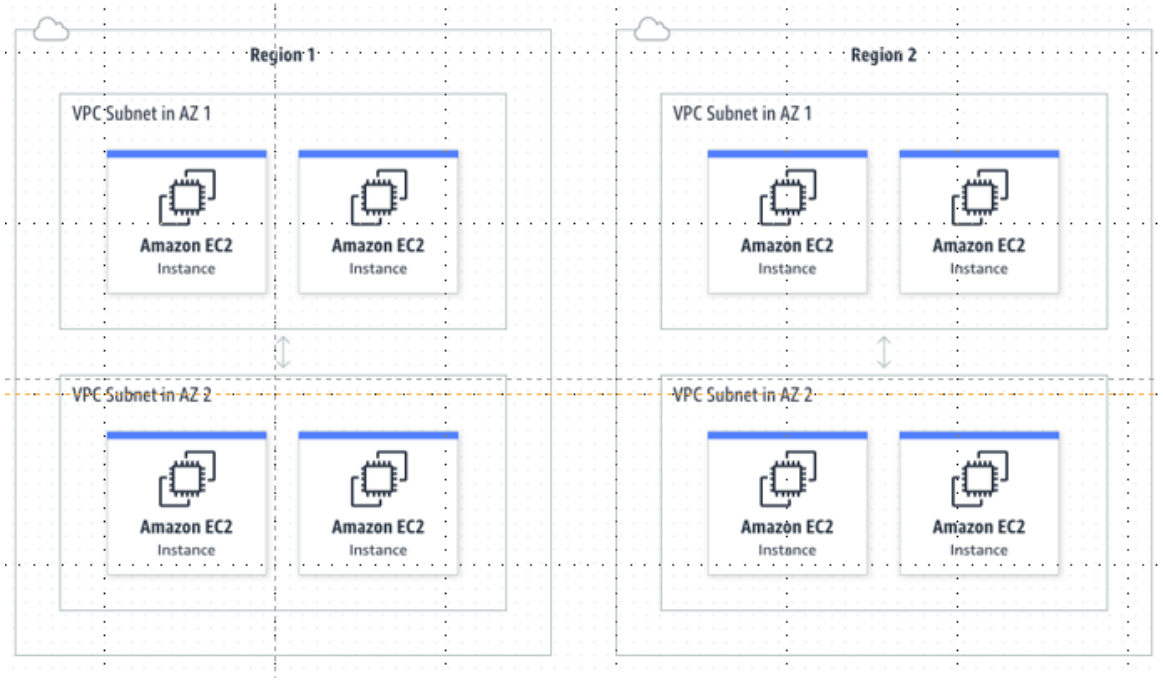
- Implement Cloud Financial Management
- Adopt a consumption model
- Measure overall efficiency
- Stop spending money on undifferentiated heavy lifting
- Analyze and attribute expenditure

https://aws.amazon.com/architecture/well-architected/

CLOUD SERVICES

Virtual Private Cloud (VPC)

- Foundational service that lets you launch cloud resources in a logically isolated virtual network that you define.



API Services

- Fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure Application Programming Interfaces (APIs) at any scale

Serverless Functions

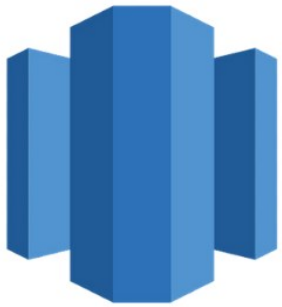
- Serverless event-driven compute service that lets users run code for most types of application or backend service without provisioning or managing servers

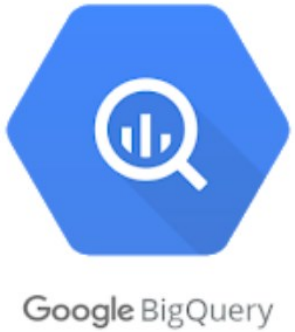
Queuing and Pub Sub

- *Pub/sub* messaging can be used to enable event-driven architectures, or to decouple applications in order to increase performance, reliability and scalability.

Cloud Data Warehouse (DW)

- Fully-managed petabyte-scale data warehouse services that makes it easy to analyze data using standard SQL and existing Business Intelligence (BI) tools.





Business Intelligence (BI) Services

- BI tools enable you to create powerful visualizations and dashboards so that managers and leaders can make data-driven decisions.

Big Data Services

- Managed services for storing and processing huge amounts of data (big data)
- Support for open-source tools such as Apache Spark, Hadoop, and Hive.

Machine Learning (ML) Services

- Fully-managed ML/AI services that makes it easy to build, train, and deploy ML models at scale
- Support Model Governance, Pipelines, Deployment, Registry, Monitoring, etc.

Amazon SageMaker

Azure Machine Learning

vertex.ai

Cloud AutoML Vision

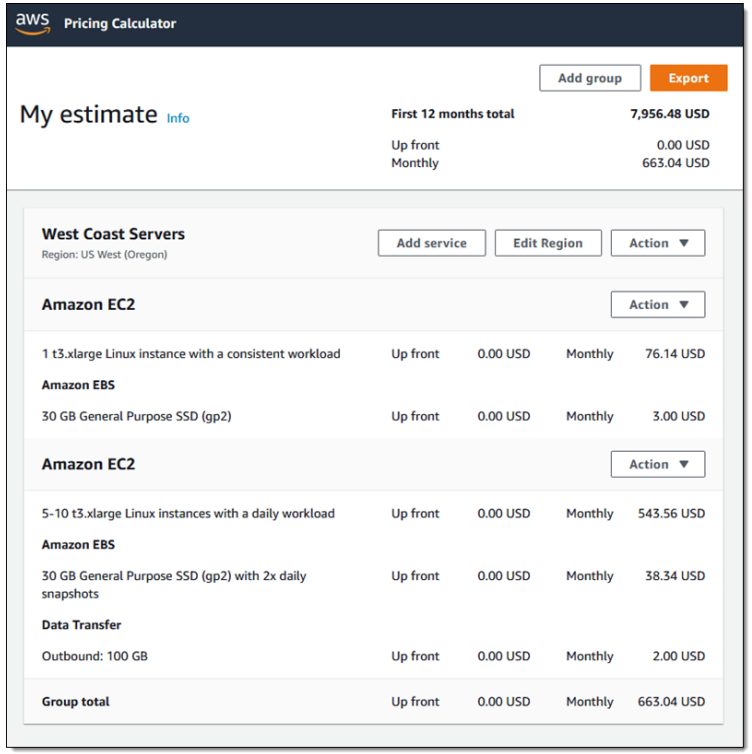
Data Exchange/Marketplaces

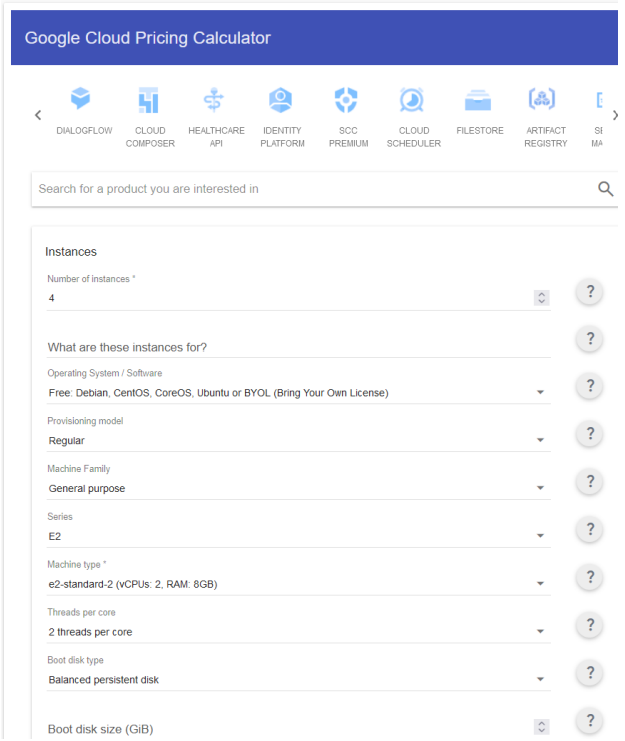
- Online location or store that facilitates the buying and selling of data between producers and consumers across organizational boundaries

AWS Data Exchange

Analytics Hub

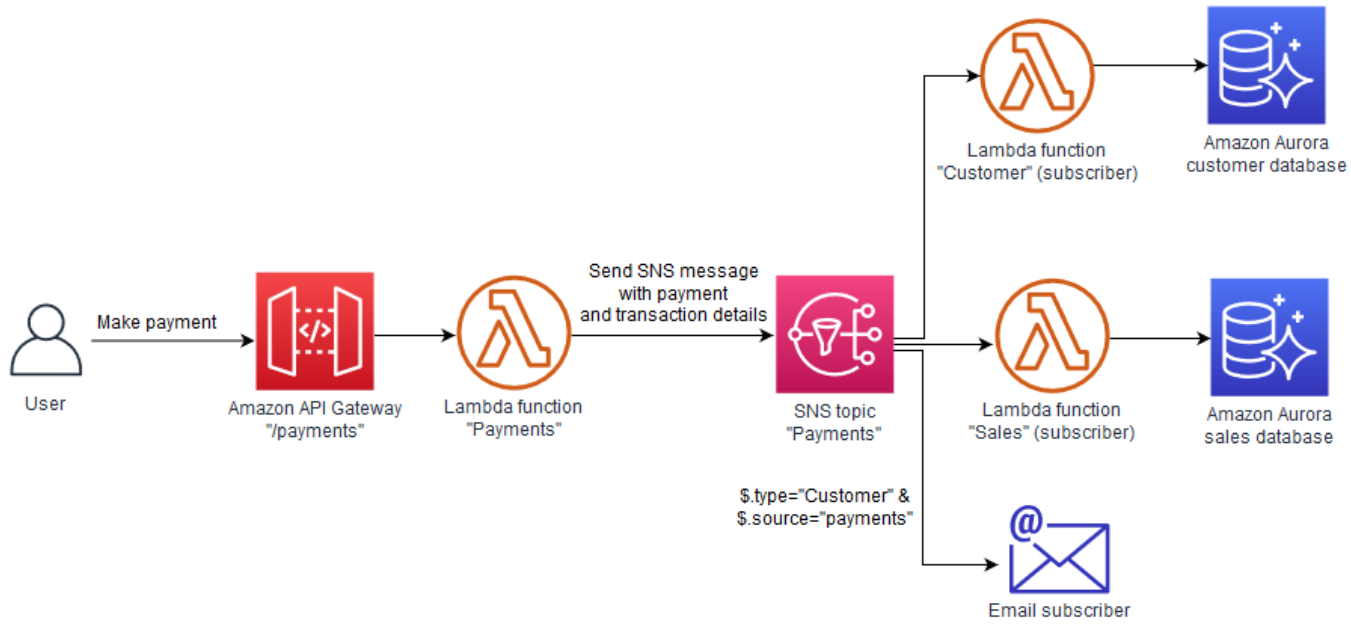
Pricing Calculators



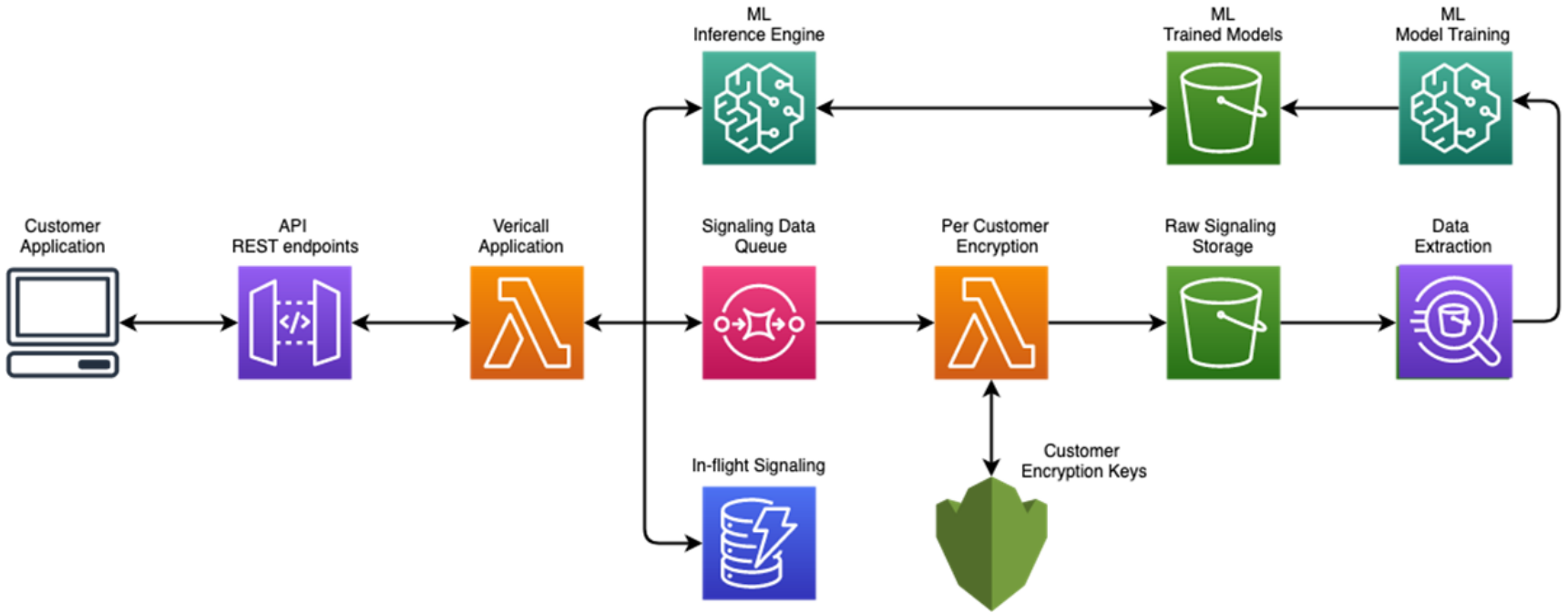


REFERENCE ARCHITECTURES

Pub/Sub Architecture

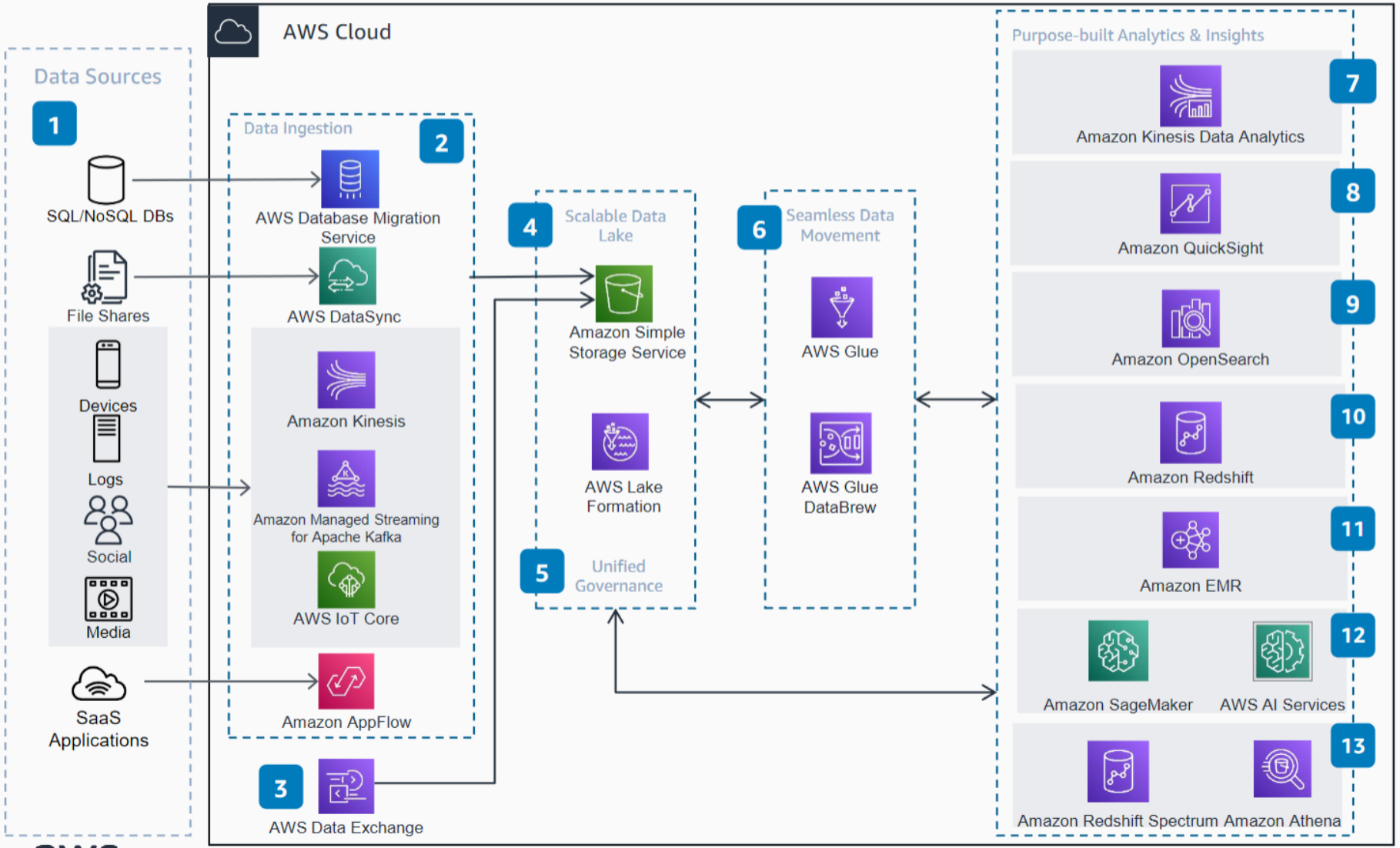


AWS ML Pipeline



Modern Data Analytics Reference Architecture on AWS

This architecture enables customers to build data analytics pipelines using a Modern Data Analytics approach to derive insights from the data.



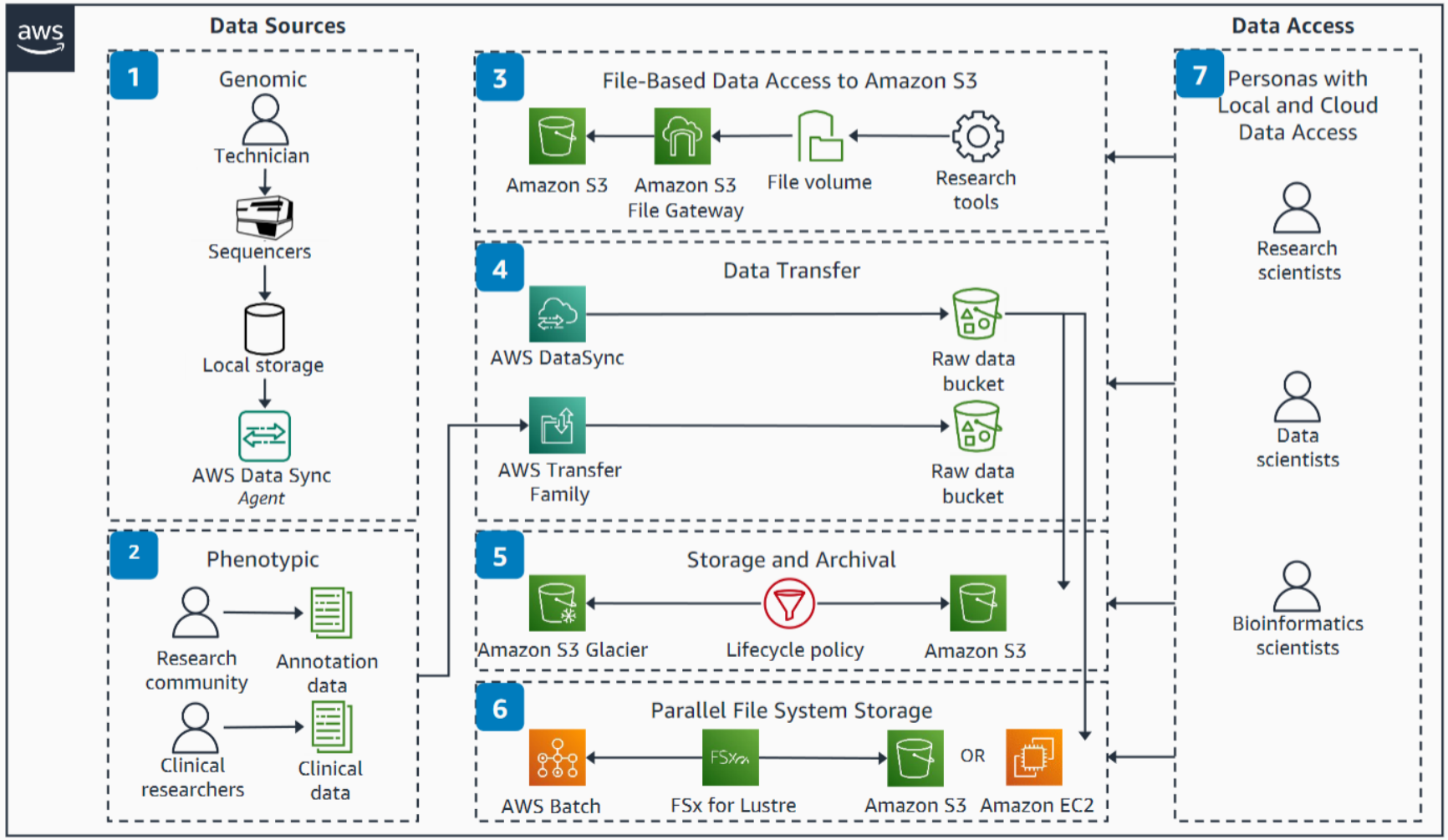
1. Data is collected from multiple data sources across the enterprise, SaaS applications, edge devices, logs, streaming media, and social networks.
2. Based on the type of the data source, **AWS Database Migration Service, AWS DataSync, Amazon Kinesis, Amazon Managed Streaming for Apache Kafka, AWS IoT Core**, and **Amazon AppFlow** are used to ingest the data into a Data Lake in AWS.
3. **AWS Data Exchange** is used for integrating third-party data into the Data Lake.
4. **AWS Lake Formation** is used to build the scalable data lake, and **Amazon S3** is used as the data lake storage.
5. **AWS Lake Formation** is also used to enable unified governance to centrally manage the security, access control, and audit trails.
6. **AWS Glue** and **AWS Glue DataBrew** are used to catalog, transform, enrich, move, and replicate data across multiple data stores and the data lake.
7. **Amazon Kinesis Data Analytics** is used to transform and analyze streaming data in real time.
8. **Amazon QuickSight** provides machine learning-powered business intelligence.
9. **Amazon OpenSearch** can be used operational analytics.
10. **Amazon Redshift** is used as a Cloud Data Warehouse.
11. **Amazon EMR** provides the cloud big data platform for processing vast amounts of data using open source tools.
12. **Amazon SageMaker** and **AWS AI services** can be used to build, train and deploy machine learning models, and add intelligence to your applications.
13. **Amazon Redshift Spectrum** and **Amazon Athena** enable interactive querying, analyzing, and processing capabilities.

Reviewed for technical accuracy May 12, 2022

AWS Reference Architecture

Genomics Data Transfer, Data Access Patterns, Storage, and Archival

Transferring genomics data to the cloud and providing data access using AWS services.



1. A technician loads a sample on a sequencer. The sample is sequenced and written to a folder on local storage on-premises. An **AWS DataSync** task is setup to sync the data from local storage to a bucket in **Amazon Simple Storage Service (Amazon S3)**.
2. Research scientists and clinical researchers can upload annotation and clinical data files to **Amazon S3** with **AWS Transfer Family** by FTP, SFTP, or FTPS.
3. Researchers on-premises use existing bioinformatics tools with data in **Amazon S3** by NFS or SMB using **Amazon S3 File Gateway**.
4. **AWS DataSync** is used to transfer raw genomics data from on-premises sequencers. **AWS Transfer Family** can be used by research scientists to transfer clinical or annotation data to **S3** buckets.
5. Optimize storage by writing instrument run data to an **S3** bucket configured for infrequent access. Identify your **S3** storage access patterns to optimally configure your **S3** bucket lifecycle policy and transfer data to **Amazon S3 Glacier**.
6. Burst to the cloud from on-premises, or use data already in **Amazon S3**, with **Amazon FSx for Lustre**, to provide a high throughput shared file system across compute clusters running on-premises and on **Amazon Elastic Compute Cloud (Amazon EC2)** instances using **AWS Batch**.
7. Researchers can access their data in **Amazon S3** from on-premises or from within their AWS account.

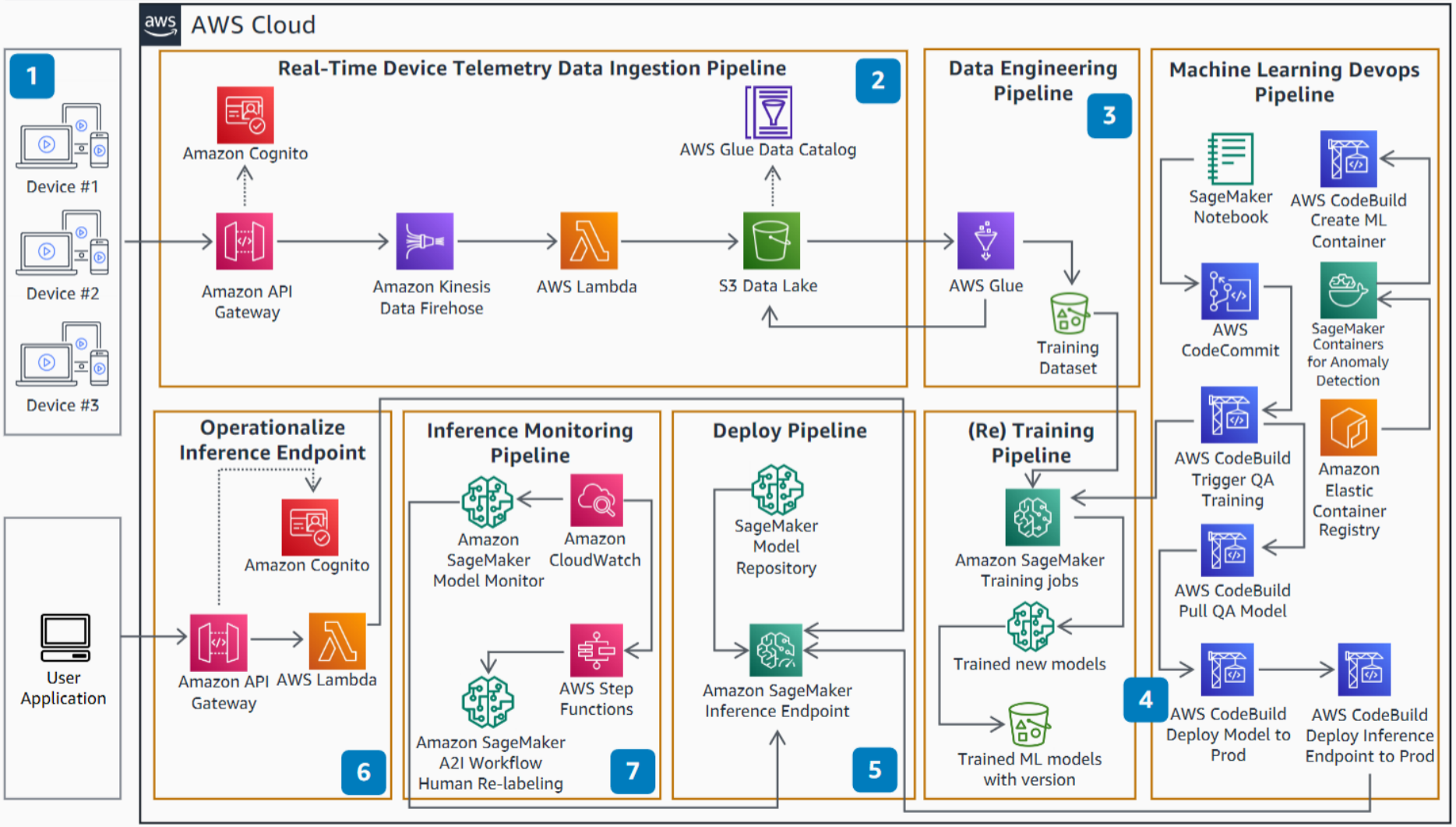
Also refer to: Genomics data transfer, analytics, and machine learning reference architecture

Reviewed for technical accuracy December 7, 2021

AWS Reference Architecture

Build your own Anomaly Detection ML Pipeline

This end-to-end ML pipeline detects anomalies by ingesting real-time, streaming data from various network edge field devices, performing transformation jobs to continuously run daily predictions/inferences, and retraining the ML models based on the incoming newer time series data on a daily basis. Note that Random Cut Forest (RCF) is one of the machine learning algorithms for detecting anomalous data points within a data set and is designed to work with arbitrary-dimensional input.



1. Device telemetry data is ingested from the field devices on a near real-time basis by calls to the API via **Amazon API Gateway**. The requests get authenticated/authorized using **Amazon Cognito**.

2. **Amazon Kinesis Data Firehose** ingests the data in real time, and invokes **AWS Lambda** to transform the data into parquet format. **Kinesis Data Firehose** will automatically scale to match the throughput of the data being ingested.

3. The telemetry data is aggregated on an hourly basis and re-partitioned based on the year, month, date, and hour using **AWS Glue** jobs. The additional steps like transformations and feature engineering are performed for training the Anomaly Detection ML Model using **AWS Glue** jobs. The training data set is stored on **Amazon S3 Data Lake**.

4. The training code is checked in an **AWS CodeCommit** repo which triggers a Machine Learning DevOps (MLOps) pipeline using **AWS CodePipeline**. **CodePipeline** builds the **Amazon SageMaker** training and inference containers, triggers the **SageMaker** training job using the specified training dataset, deploys the trained model in the testing environment, and upon approval, deploys the model into production using **SageMaker** inference endpoints.

5. The ML models generated by training jobs are registered in the **SageMaker** Model Repository. The deploy pipeline selects the best ML model to deploy using **SageMaker** hosting.

6. Classify if the telemetry data is an anomaly or not via HTTP(s) API using **Amazon API Gateway** and **Lambda** functions. The **Lambda** function invokes the **SageMaker** endpoint to predict the anomaly.

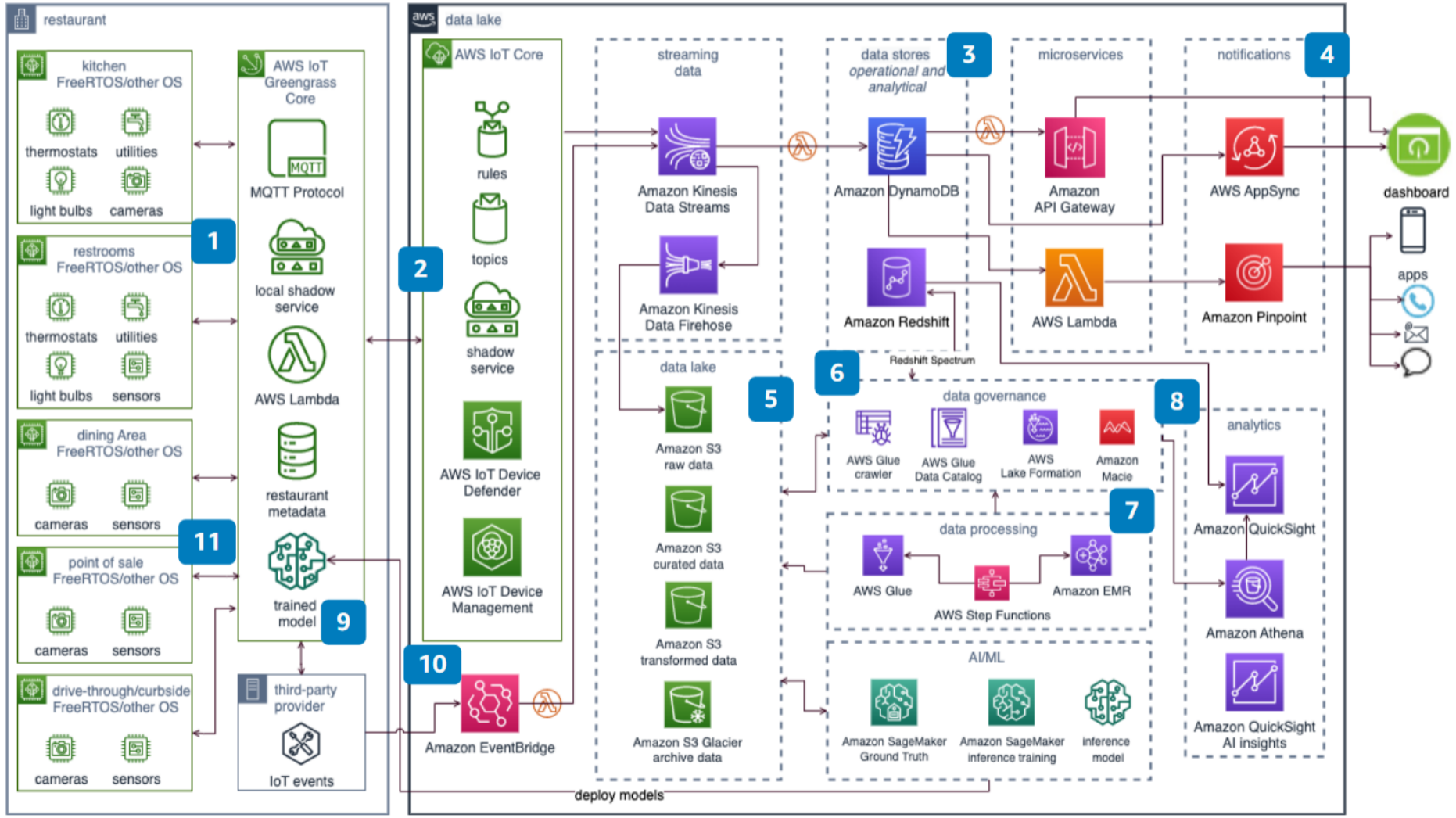
7. The inference quality is monitored using **SageMaker** Model Monitor. The requests with ambiguous prediction scores are sent for re-labeling using **Amazon CloudWatch** events, triggering the **SageMaker** A2I workflow using **AWS Step Functions**.

Reviewed for technical accuracy June 1, 2021

AWS Reference Architecture

Connected Restaurants using IoT and AI/ML

Build smart, connected restaurants that use Internet of Things (IoT) and artificial intelligence/machine learning (AI/ML) capabilities to maintain food quality and safety in the kitchen, preserve products in cold storage, maintain social distancing, manage queue depths, measure and monitor foot and vehicle traffic, and maintain cleanliness and sanitation. Use AWS IoT Greengrass to maintain cost efficiency and improve operability.

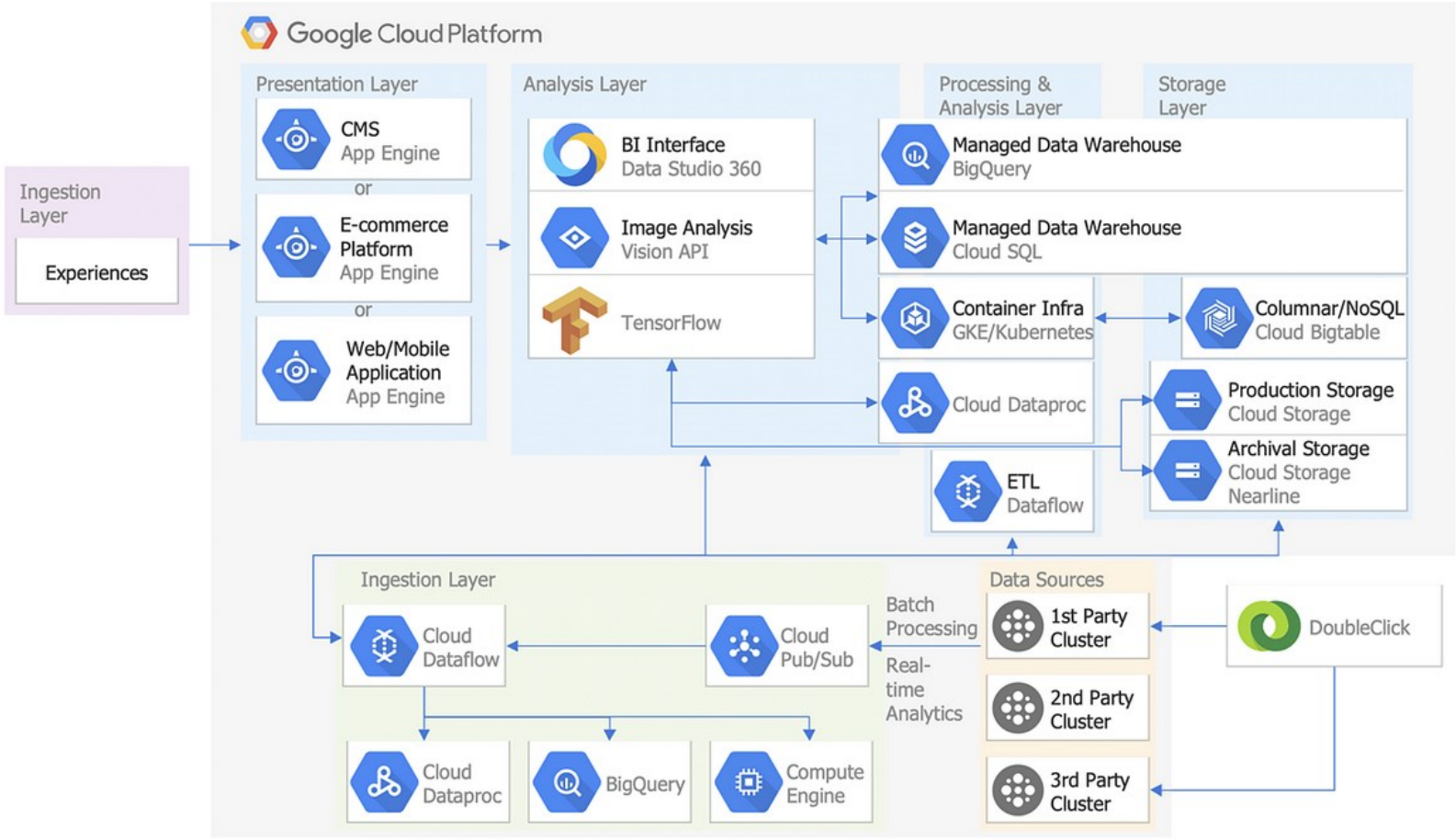


1. Use **AWS IoT Greengrass Core** to connect, publish, and subscribe data using open standard MQTT protocol with Internet of Things (IoT) devices running on **FreeRTOS** and other operating systems (OS').
2. Leverage **AWS IoT Core** to maintain shadows of all IoT devices, connect to **AWS Cloud**, manage devices, update over-the-air (OTA), and secure the devices.
3. Use purpose-built databases such as **Amazon DynamoDB** and serverless architecture to store events, deliver microservices, and generate events for the operational data store.
4. Build a near real-time operational dashboard using microservices and **AWS AppSync**. Deliver alerts to multiple channels using **Amazon Pinpoint**.
5. Build the data lake to store raw data and to create curated processed data in **Amazon Simple Storage Service** (Amazon S3) using **AWS Glue** and **Amazon EMR**.
6. Discover and govern the data in **Amazon S3** using **AWS Glue** crawlers, **AWS Glue Data Catalog**, and **AWS Lake Formation**. Additionally deploy **Amazon Macie** to detect any sensitive data.
7. Use **AWS Glue** jobs and **Amazon EMR** to perform any transformation or enrichment of the data.
8. Use **Amazon Redshift**, **Amazon Athena**, and **Amazon QuickSight** for analytics. Optionally, build data marts in **Amazon Redshift** for heavily used analytics. For one-time requirements, publish the data catalog and use **Amazon Athena** or **Amazon Redshift Spectrum** for direct analysis using the data lake.
9. Use **Amazon SageMaker** to build, train, and deploy inference models. Optionally, deploy edge models on **AWS IoT Greengrass Core**.
10. Use the Facilitate Social Distancing and Queue Depth Management solutions for compliance and enhanced customer experience.
11. Use **Amazon EventBridge** to integrate with third-party providers.

Reviewed for technical accuracy December 21, 2022

AWS Reference Architecture

Digital Marketing



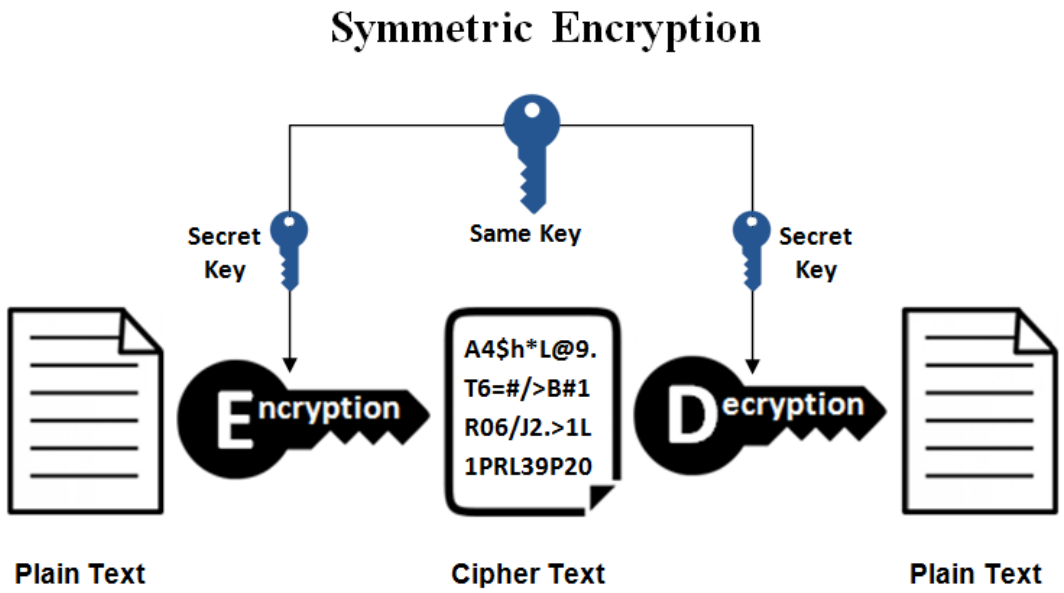
DATA SECURITY

Data Encryption

- Securing digital data on the cloud and computer systems
- One of the most effective methods of data protection
- What data should be encrypted?
  - Personally Identifiable Information (PII)
  - Confidential company data
- Benefits
  - Privacy Protection
  - Cybersecurity - prevents identity theft and ransomware
  - Allows you to securely share and transmit digital data
  - Regulatory compliance
  - Protect remote workers - lost/stolen devices, etc.

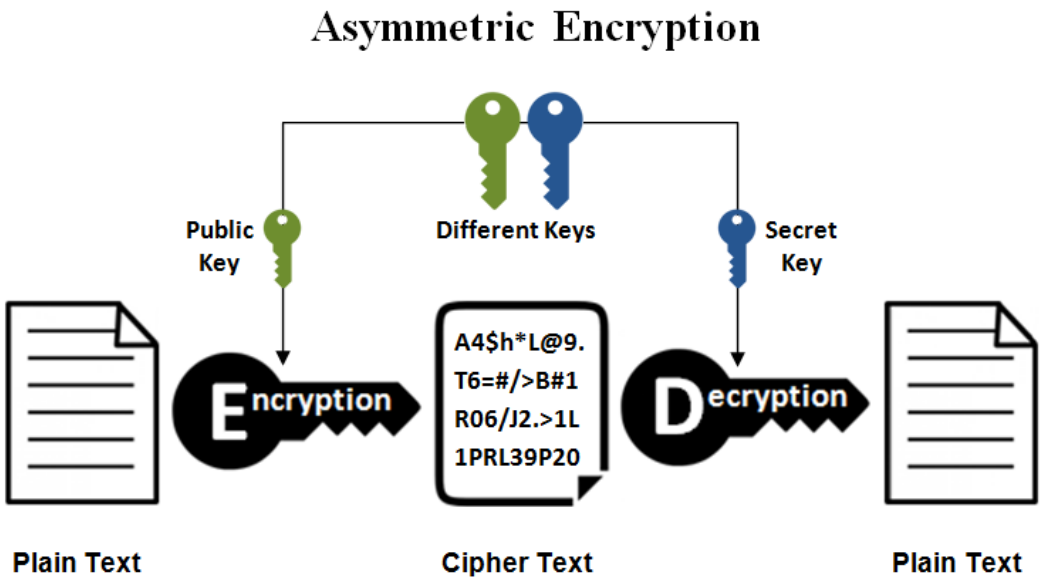
Symmetric Encryption

- Symmetric encryption uses only one secret symmetric key to encrypt the plaintext and decrypt the ciphertext
- Symmetric encryption methods:
  - Data Encryption Standards (DES)
  - Triple DES
  - Advanced Encryption Standard (AES)
  - Twofish

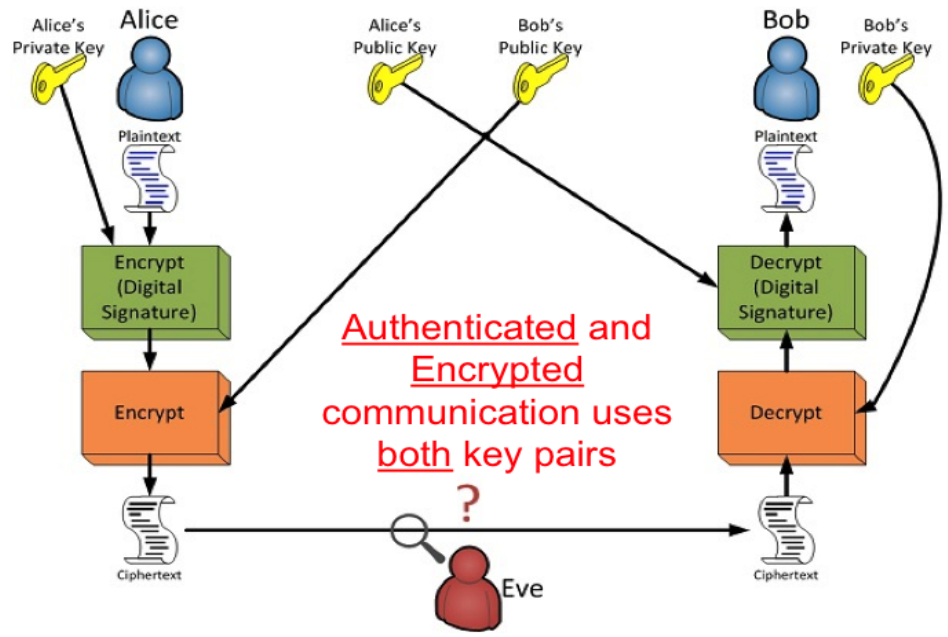


Asymmetric Encryption

- Also known as Public-Key Cryptography, encrypts and decrypts the data using two separate cryptographic asymmetric keys
- These two keys are known as a “public key” and a “private key”
- Asymmetric encryption methods:
  - RSA (Ron Rivest, Adi Shamir, and Leonard Adleman)
  - Public key infrastructure (PKI)



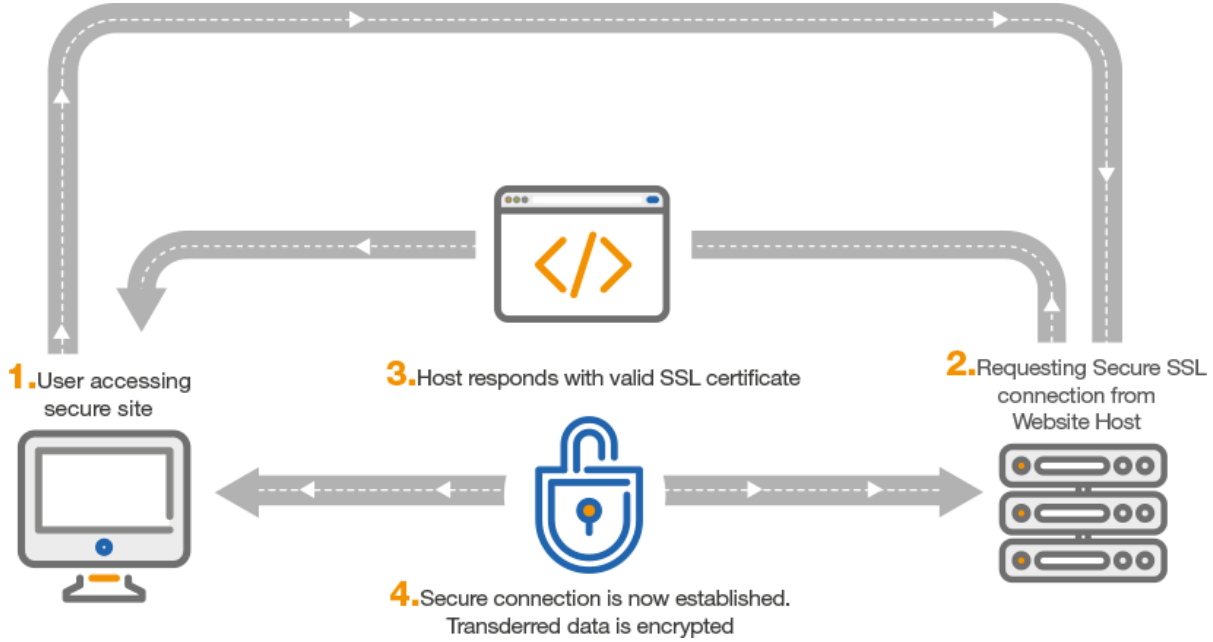
Asymmetric Encryption



If Alice generates a private key and a corresponding public key, then anyone is allowed to know her public key, but Alice must keep her private key secrets.

Secure Sockets Layer (SSL)

- Standard security protocol for establishing encrypted links between a web server and a browser in an online communication



End to End Encryption

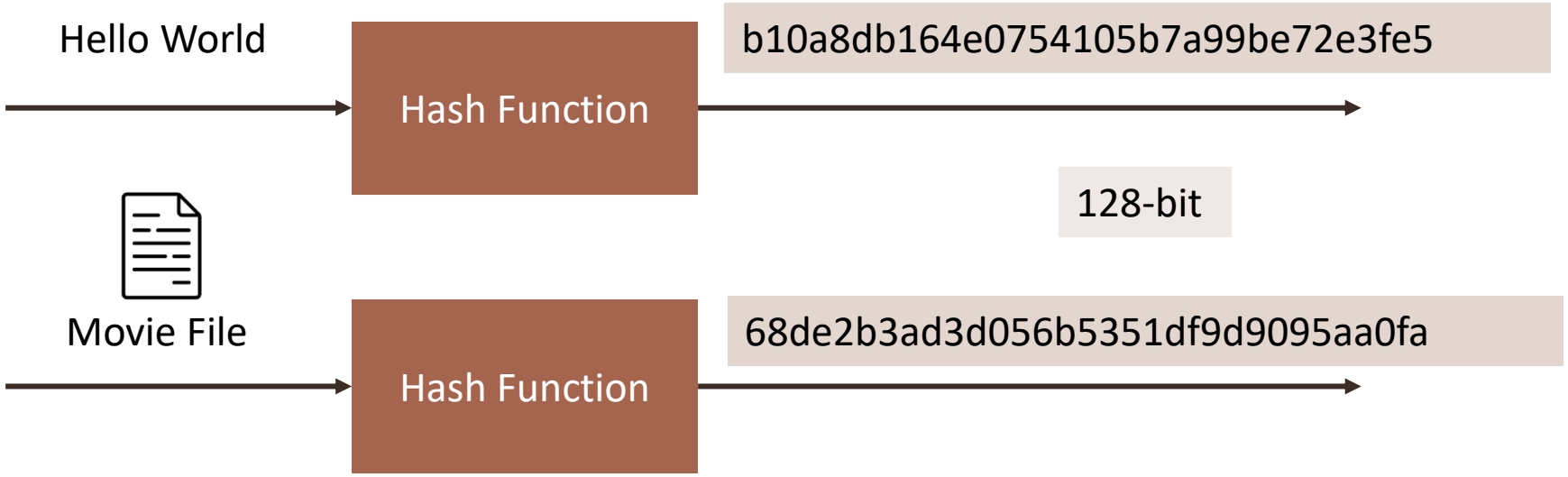
- Sensitive data must be encrypted both at rest and in motion

Stored data (data at rest)

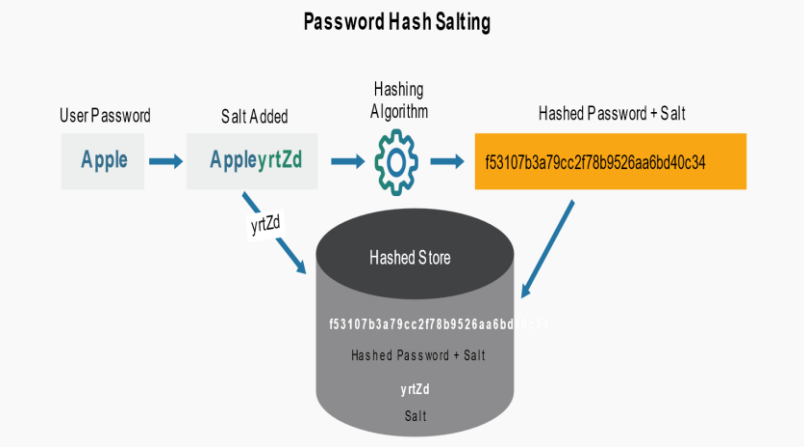
Transmitted data (data in transit)

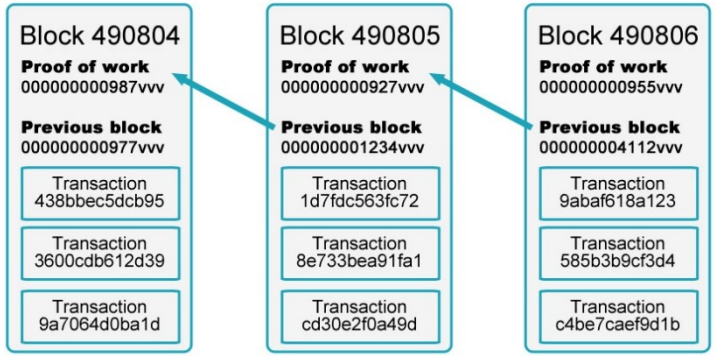
Hash Function

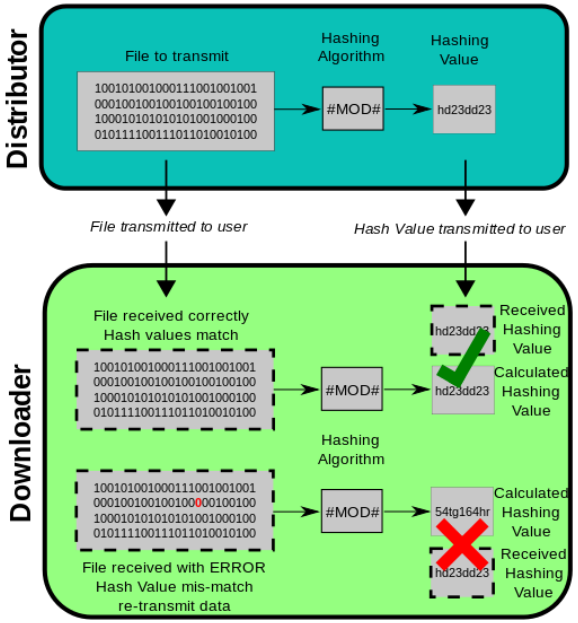
- Used to map data of arbitrary size to fixed-size values



Hashing Applications







https://en.wikipedia.org/wiki/MD5