Software Engineering

Cloud Computing and Cloud Software Architecture

1142SE07
MBA, IM, NTPU (M5010) (Spring 2026)
Wed 2, 3, 4 (9:10-12:00) (B3F17)

https://meet.google.com/ish-gzmy-pmo

Min-Yuh Day, Ph.D,
Professor and Director

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2026-04-22

Syllabus

Week	Date	Subject/Topics
1	2026/02/25	Introduction to Software Engineering
2	2026/03/04	Software Products and Project Management: Software product management and prototyping with Generative AI and Agentic AI
3	2026/03/11	Agile Software Engineering: Agile methods, Scrum, and Extreme Programming
4	2026/03/18	Case Study on Software Engineering I
5	2026/03/25	Features, Scenarios, and Stories
6	2026/04/01	Software Architecture: Architectural design, System decomposition, and Distribution architecture

Syllabus

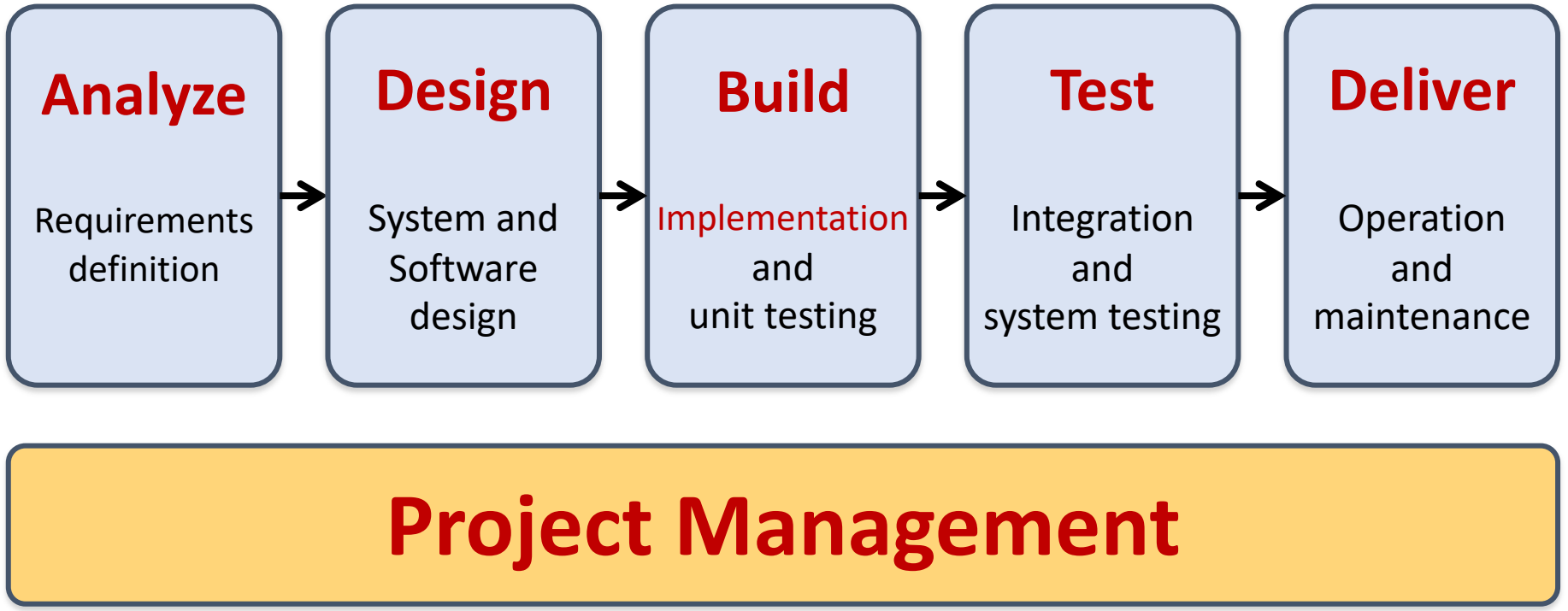
Week	Date	Subject/Topics
7	2026/04/08	Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service
8	2026/04/15	Midterm Project Report
9	2026/04/22	Cloud Computing and Cloud Software Architecture
10	2026/04/29	Microservices Architecture, RESTful services, Service deployment
11	2026/05/06	Case Study on Software Engineering II
12	2026/05/13	Security and Privacy; Reliable Programming; Testing: Functional testing, Test automation, Test-driven development, and Code reviews

Syllabus

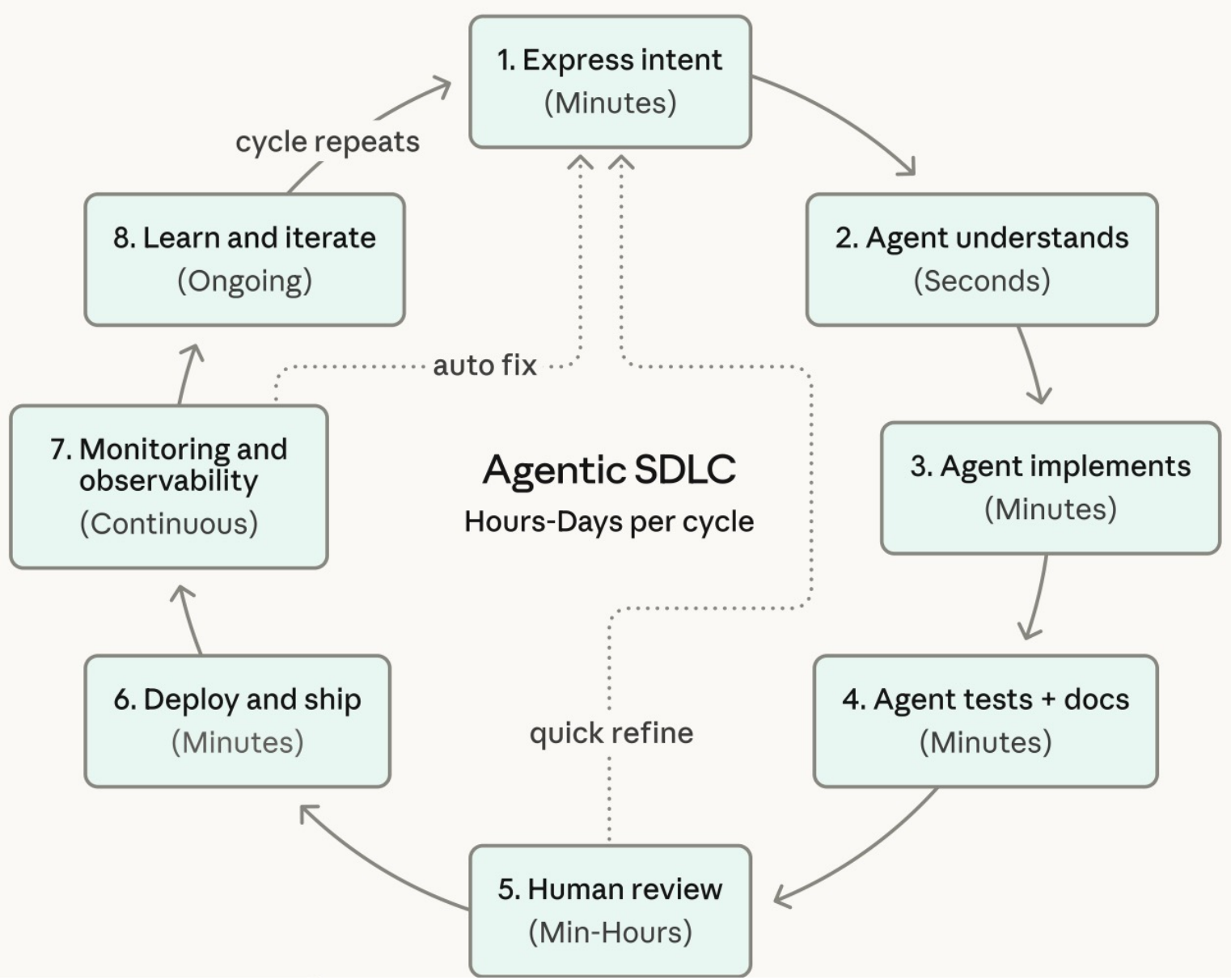
Week	Date	Subject/Topics
13	2026/05/20	Industry Practices of Software Engineering
14	2026/05/27	DevOps and Code Management: Code management and DevOps automation
15	2026/06/03	Final Project Report I
16	2026/06/10	Final Project Report II

Cloud Computing and Cloud Software Architecture

Software Engineering
and
Project Management



Agentic Coding Software Development Lifecycle



Source: Anthropic (2026), 2026 Agentic CodingTrends Report: How coding agents are reshaping software development, https://resources.anthropic.com/hubfs/2026%20Agentic%20Coding%20Trends%20Report.pdf

Information Management

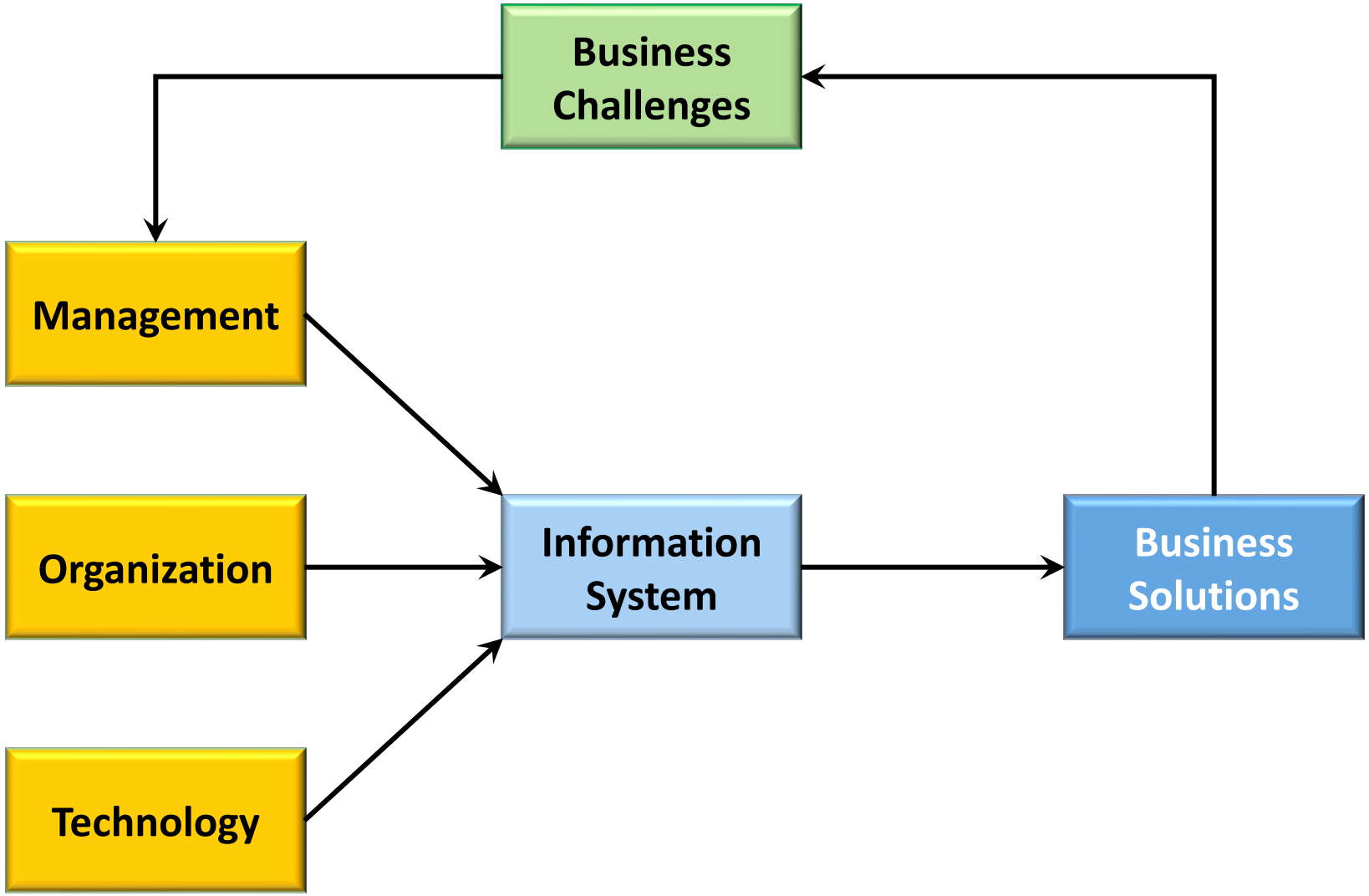
Management Information Systems (MIS)

Information Systems

Information Management (MIS)
Information Systems

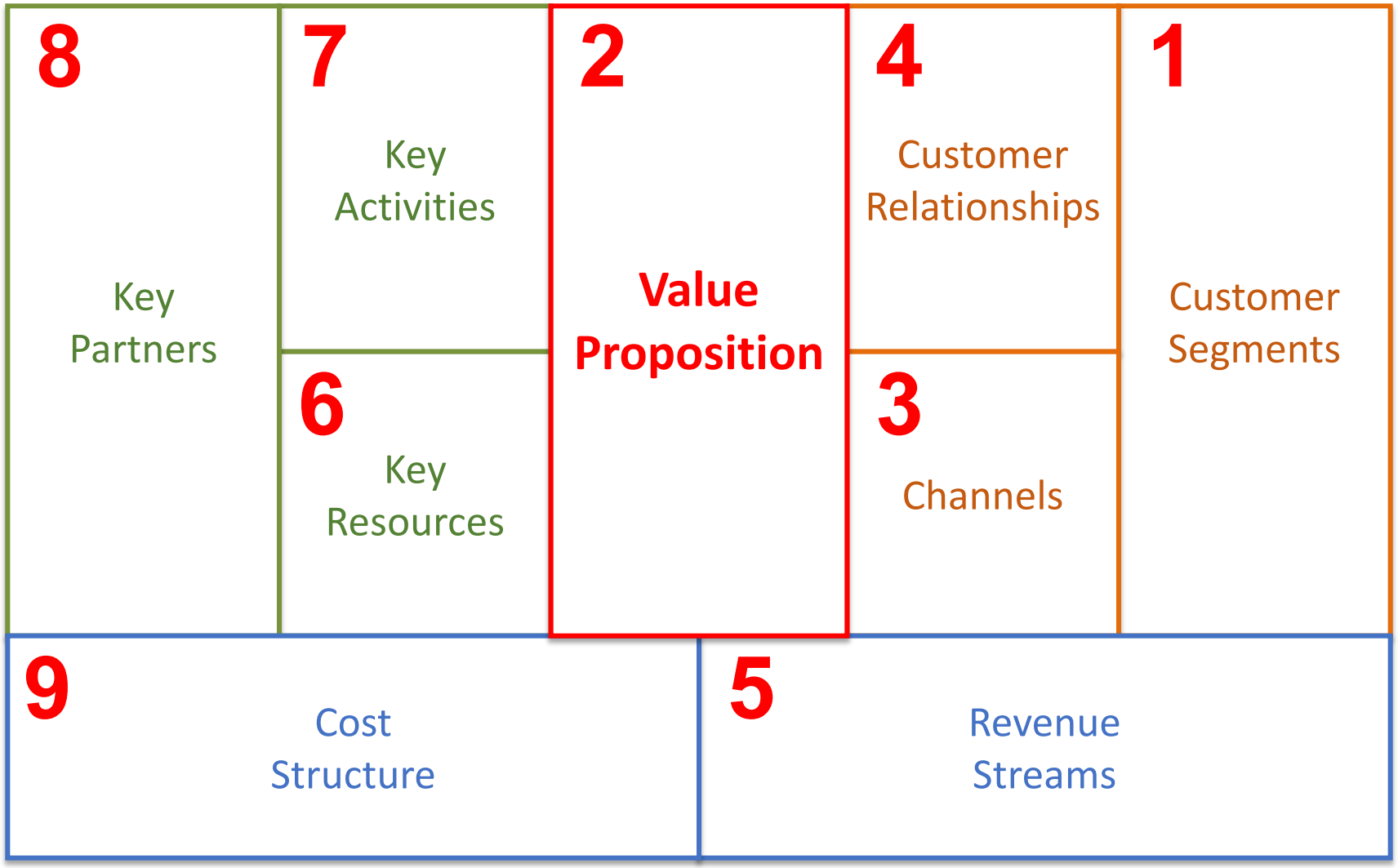
Source: Kenneth C. Laudon & Jane P. Laudon (2014), Management Information Systems: Managing the Digital Firm, Thirteenth Edition, Pearson.

Fundamental MIS Concepts



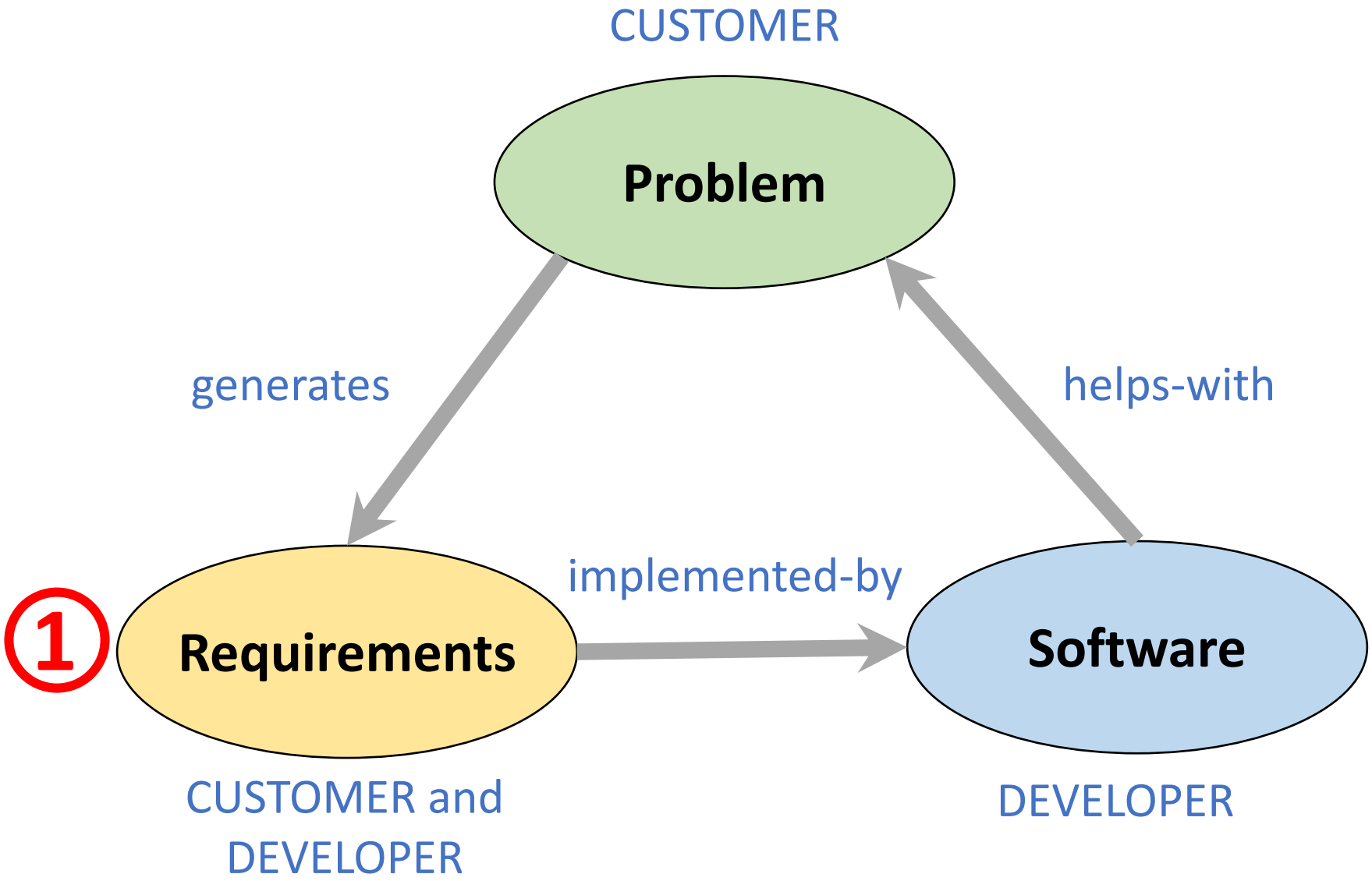
Source: Kenneth C. Laudon & Jane P. Laudon (2014), Management Information Systems: Managing the Digital Firm, Thirteenth Edition, Pearson.

Business Model



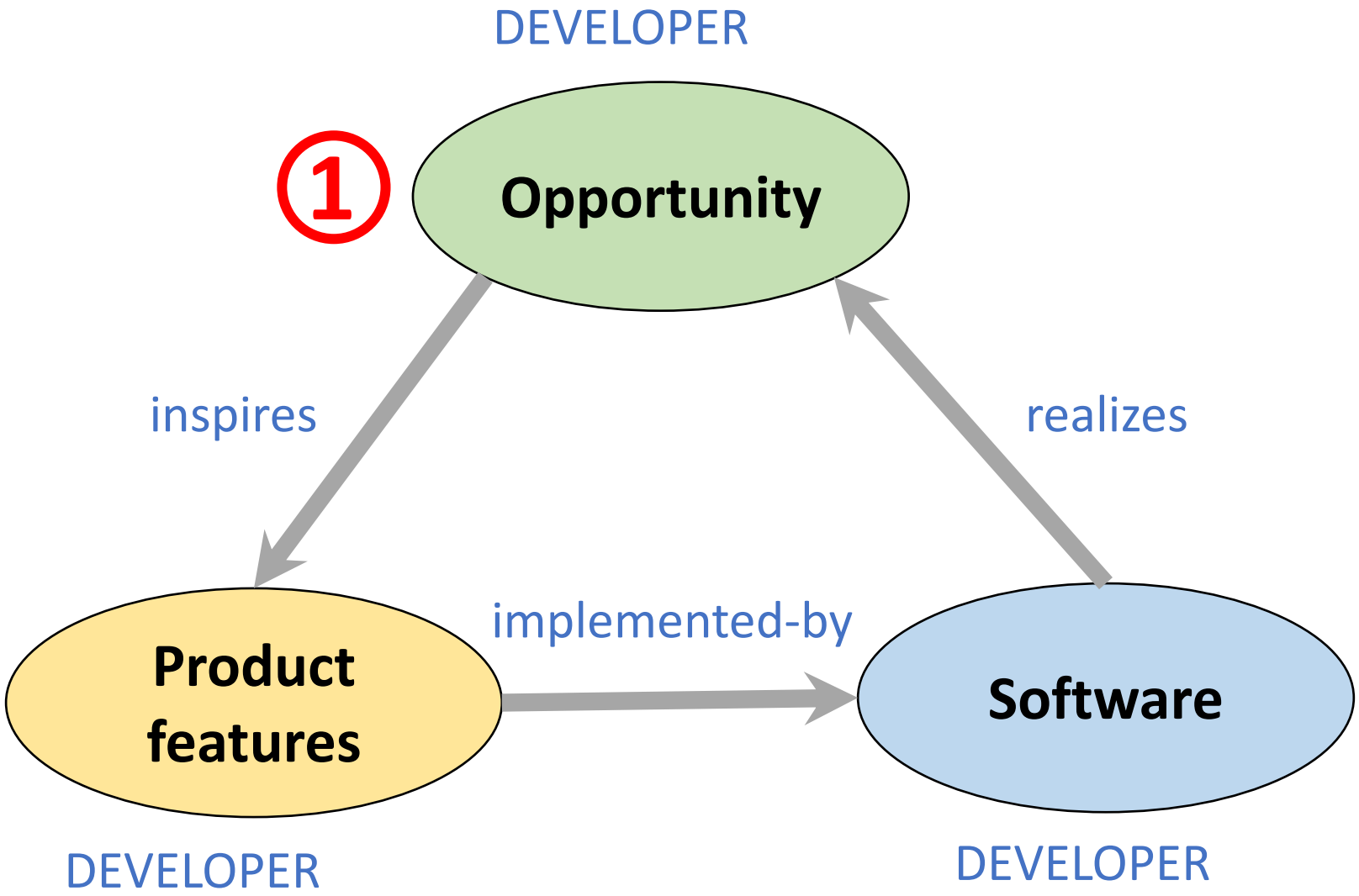
Source: Alexander Osterwalder & Yves Pigneur, Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers, Wiley, 2010.

Project-based software engineering



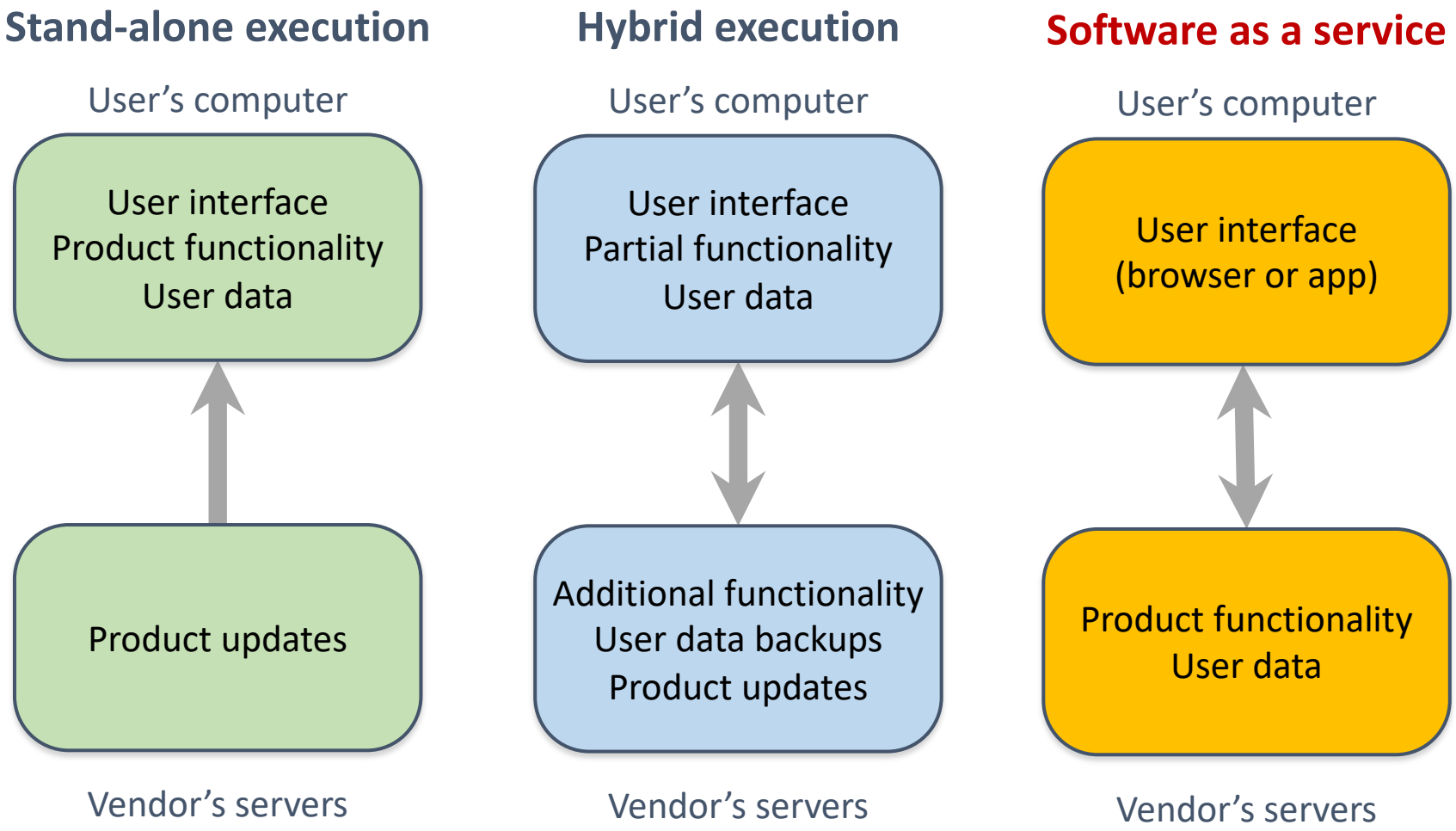
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product software engineering



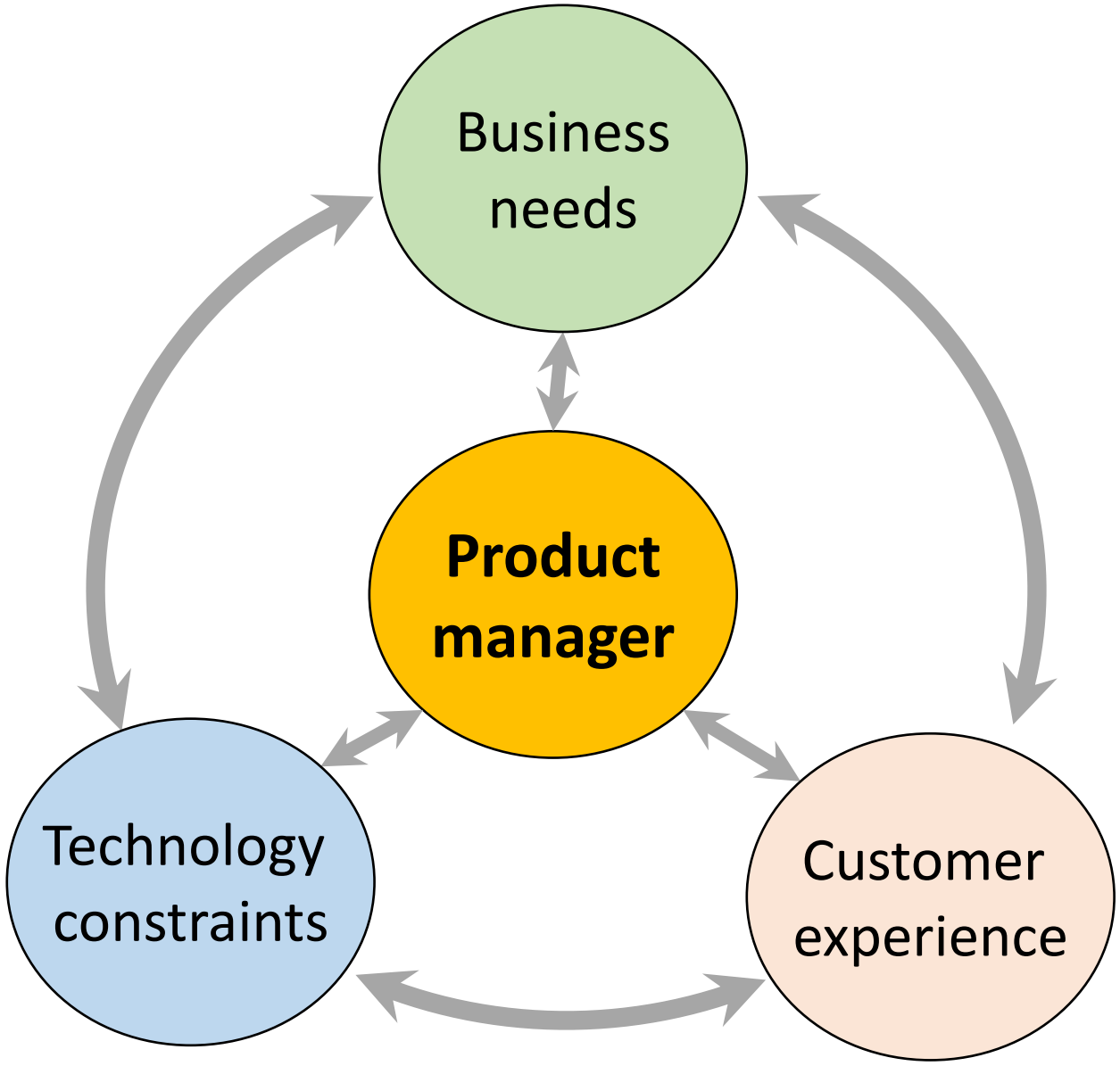
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software execution models



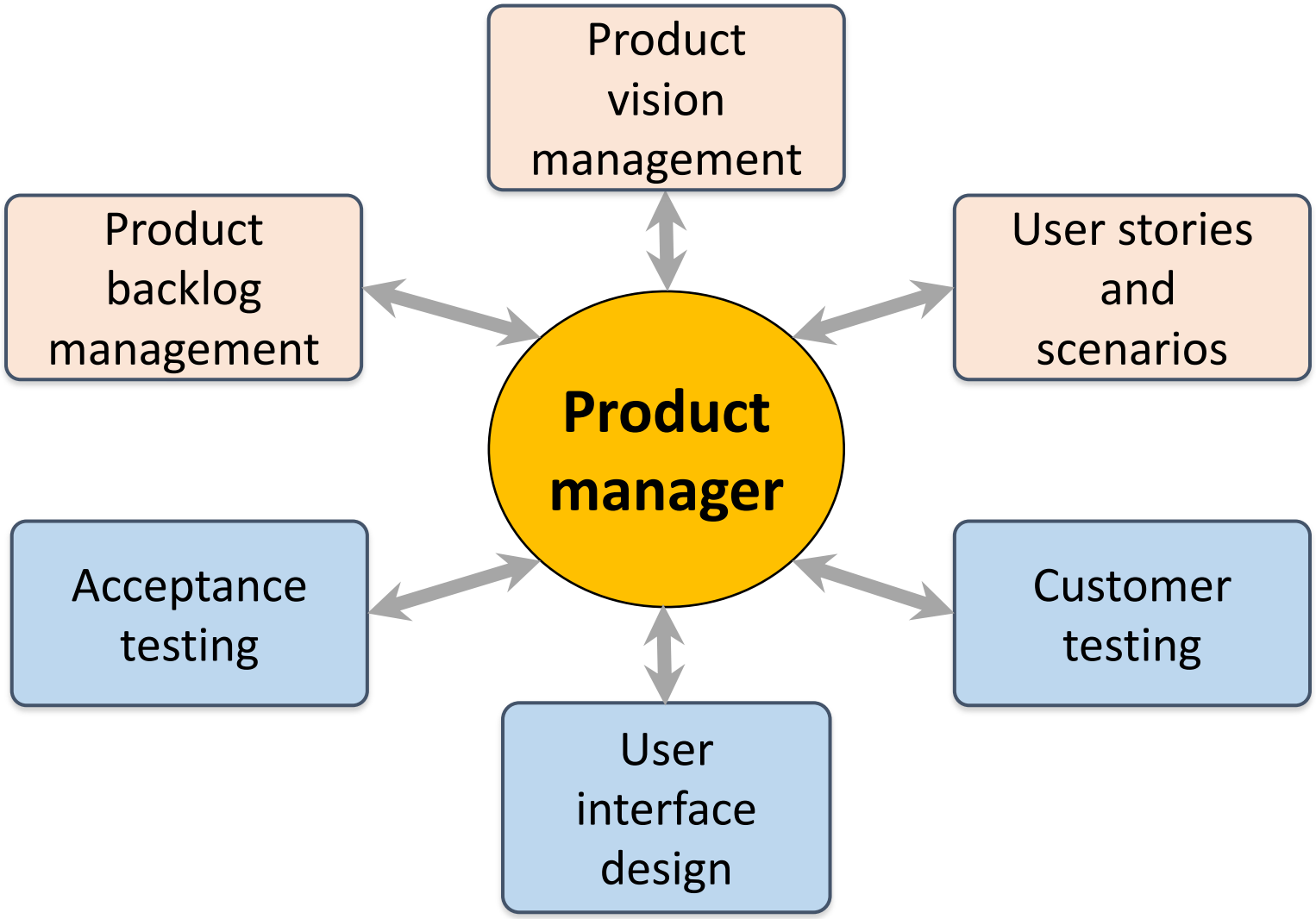
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product management concerns



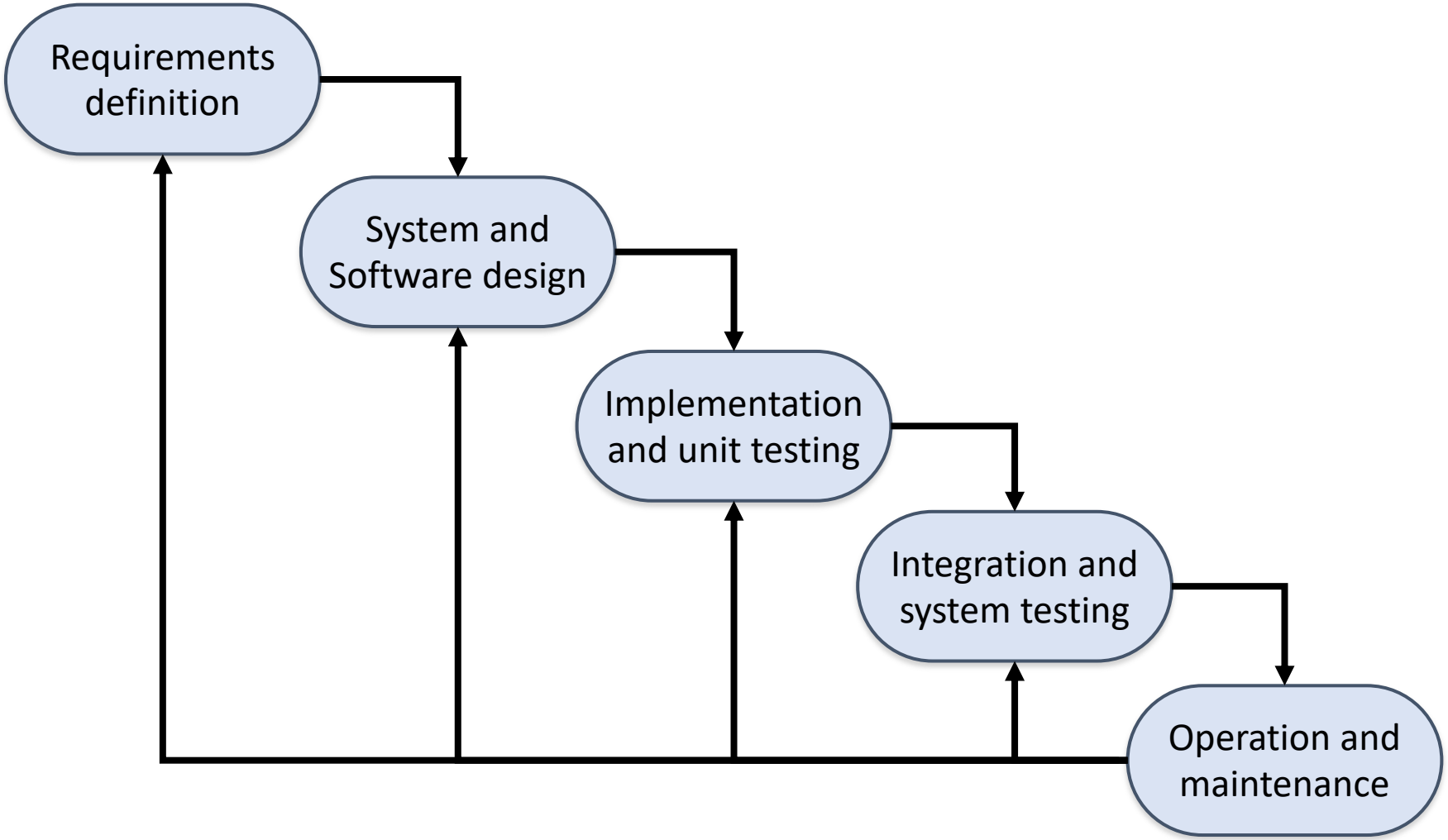
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Technical interactions of product managers



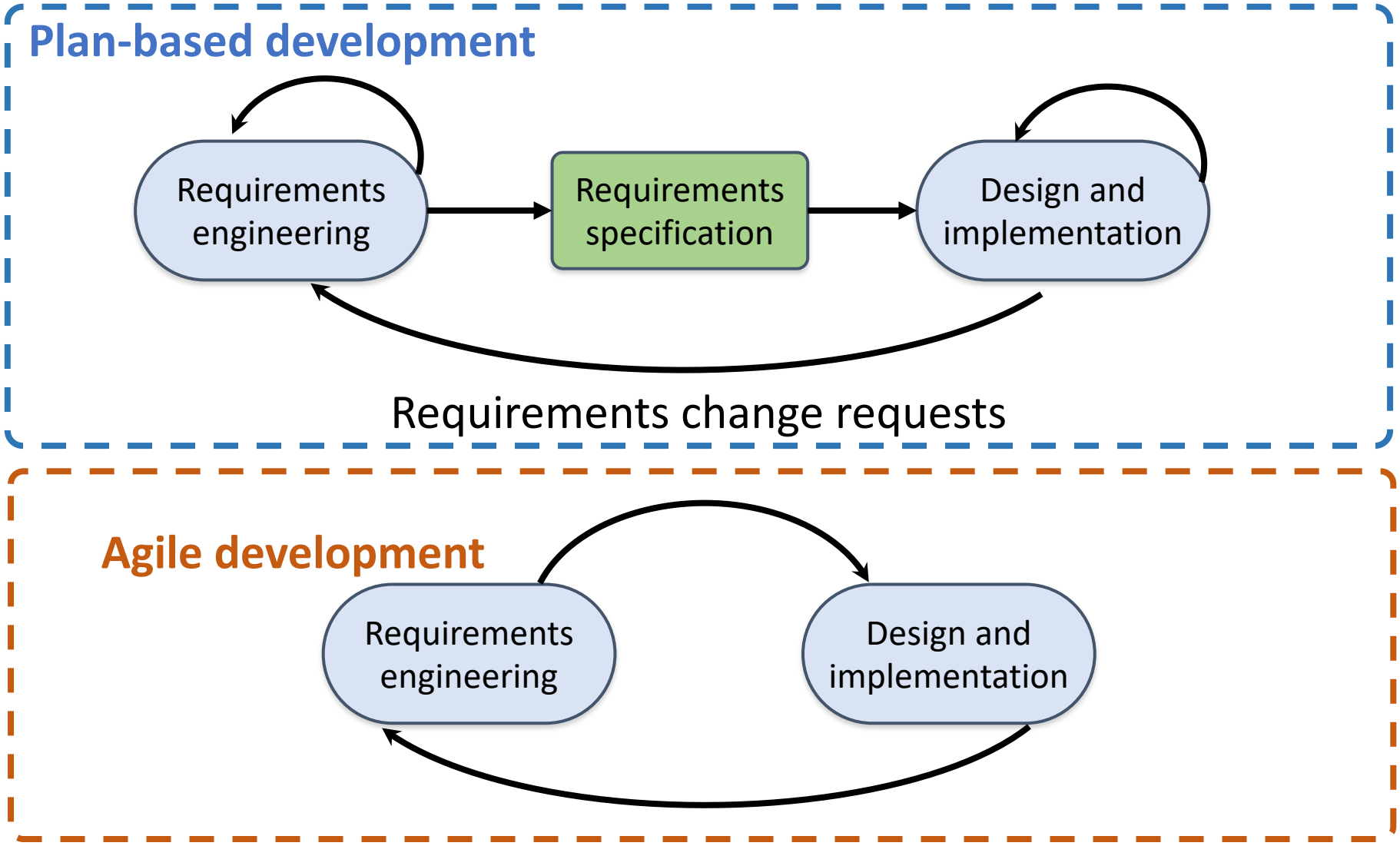
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software Development Life Cycle (SDLC)
The waterfall model



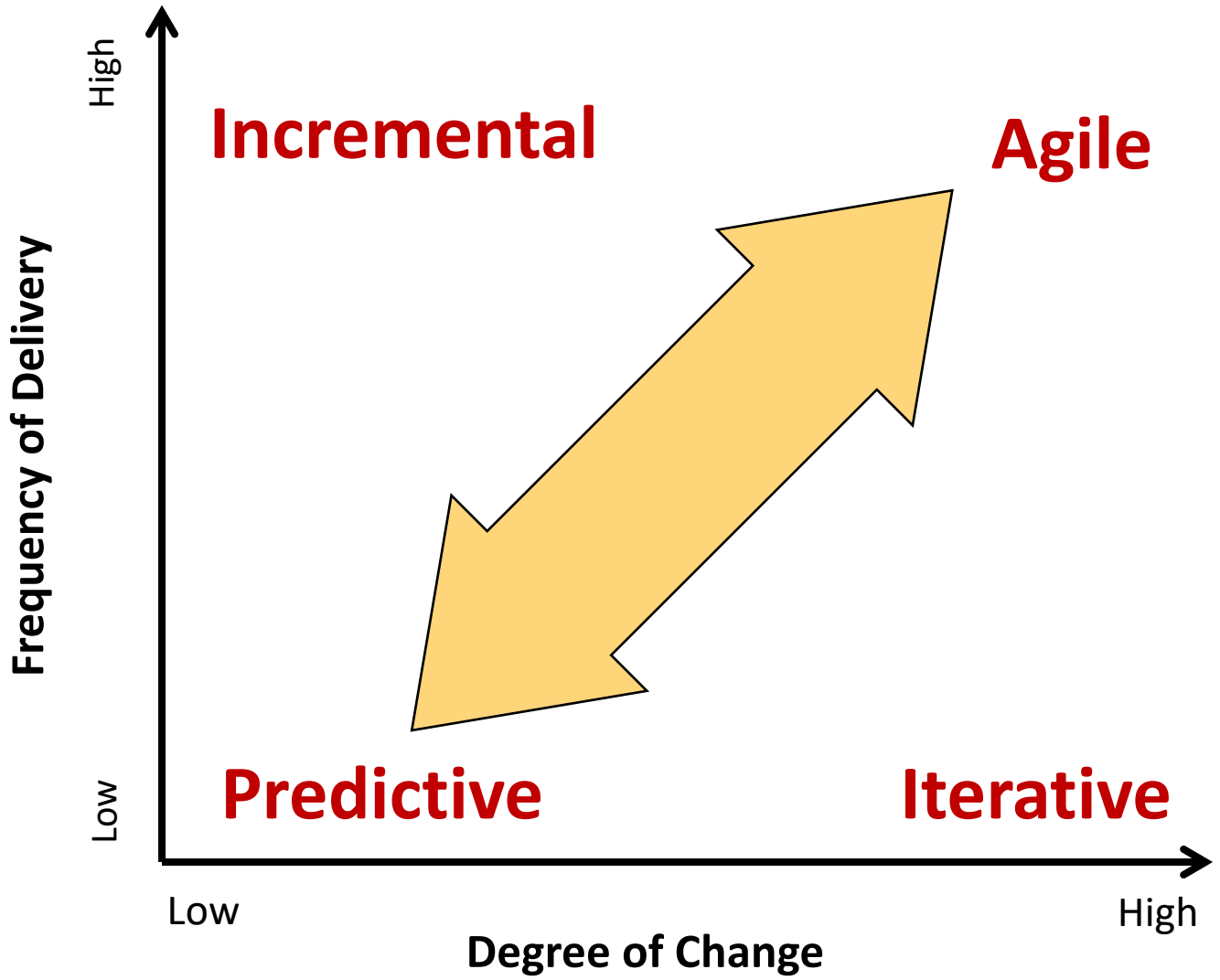
Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

Plan-based and Agile development



Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

The Continuum of Life Cycles

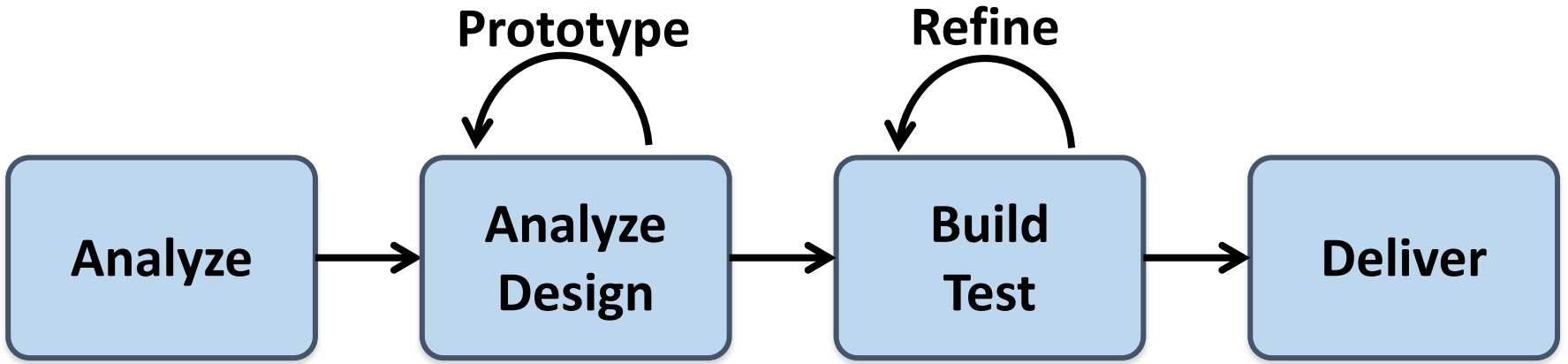


Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Predictive Life Cycle

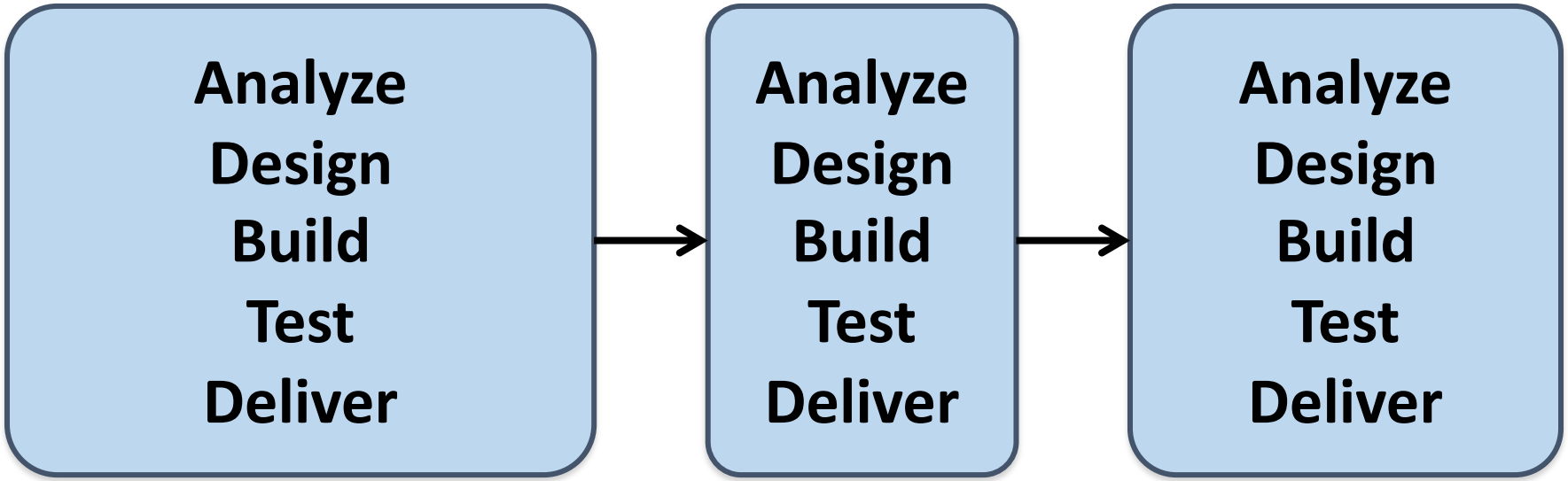
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iterative Life Cycle



Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

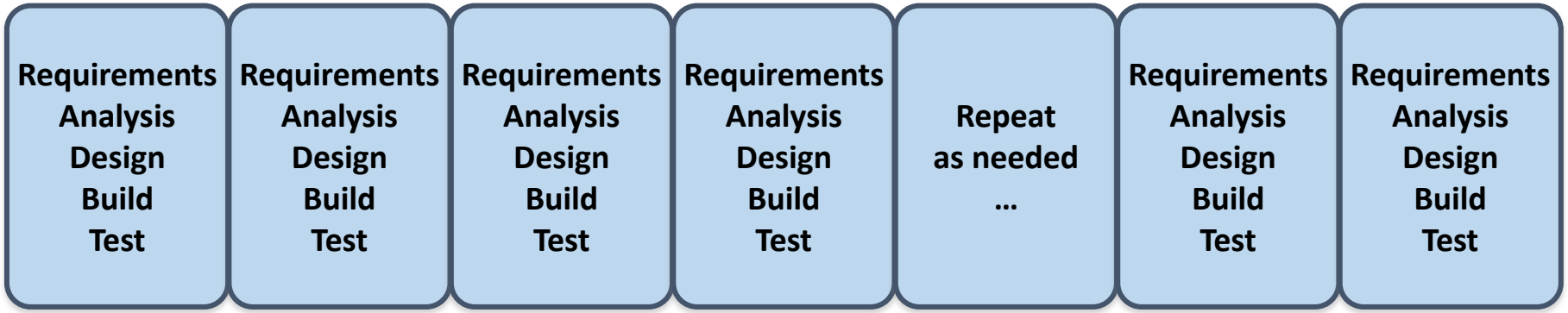
A Life Cycle of Varying-Sized Increments



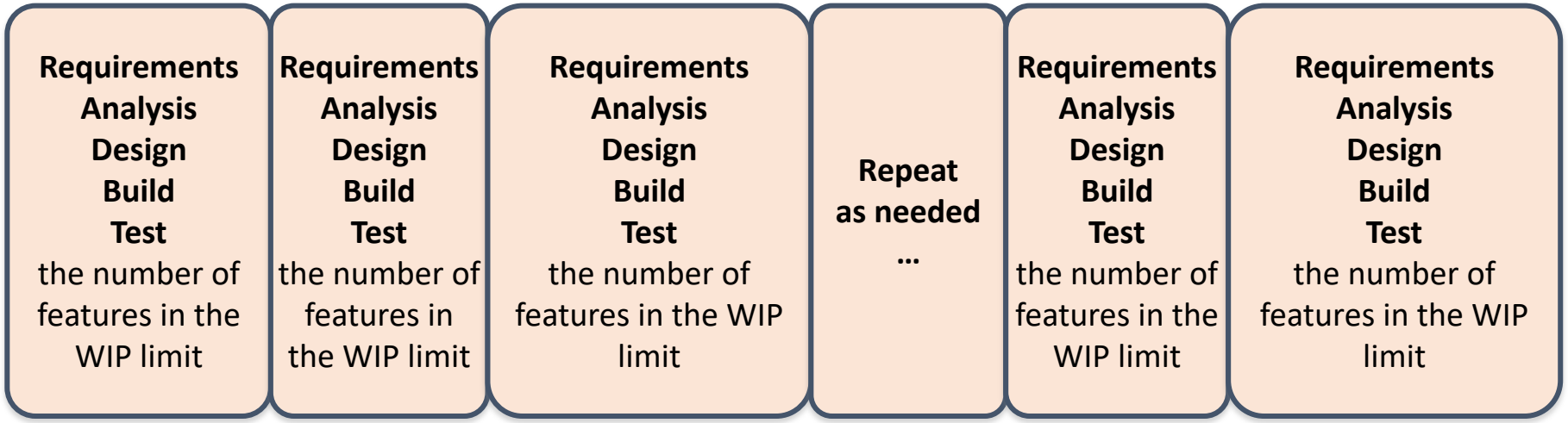
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iteration-Based and Flow-Based Agile Life Cycles

Iteration-Based Agile

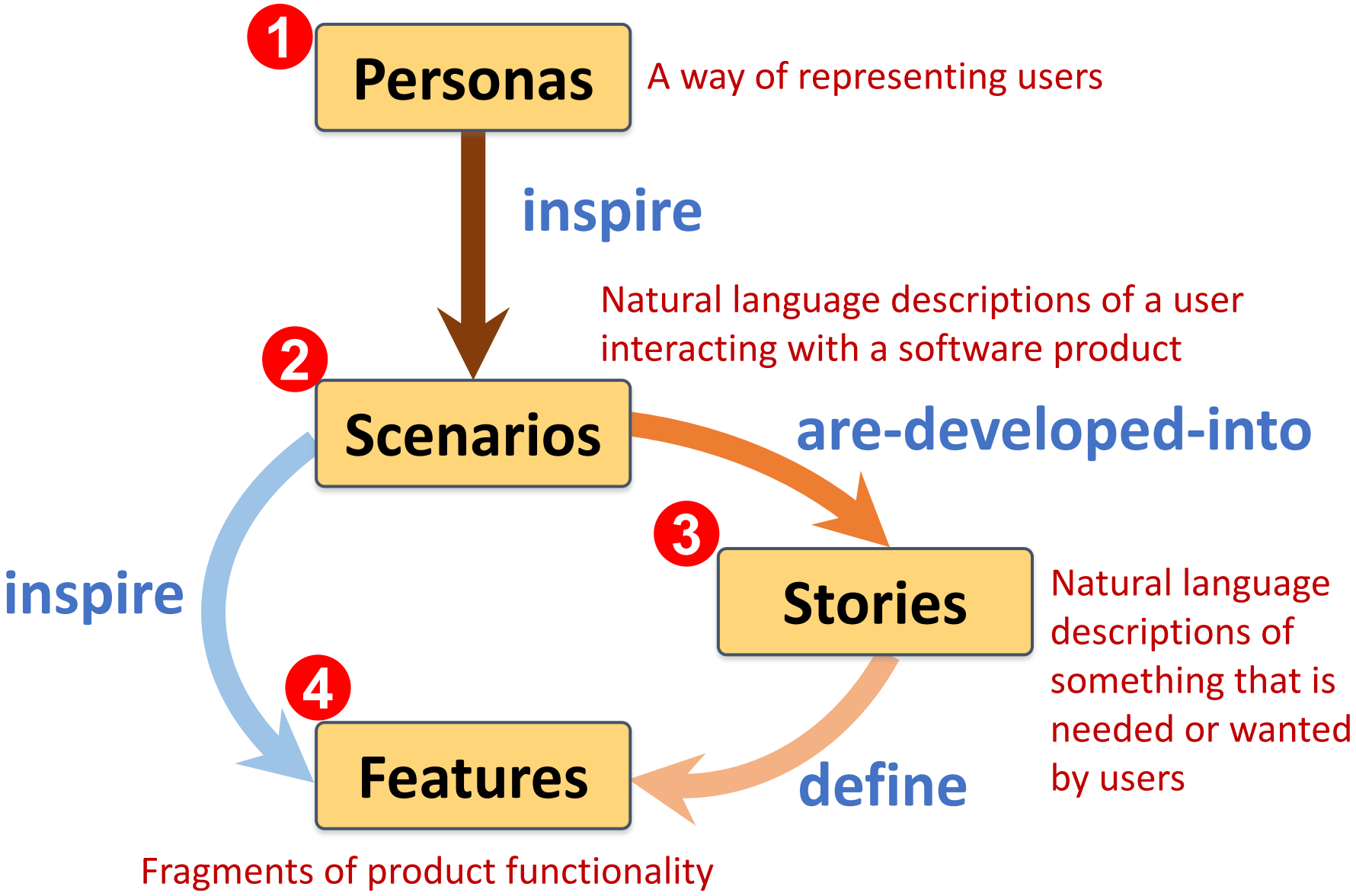


Flow-Based Agile



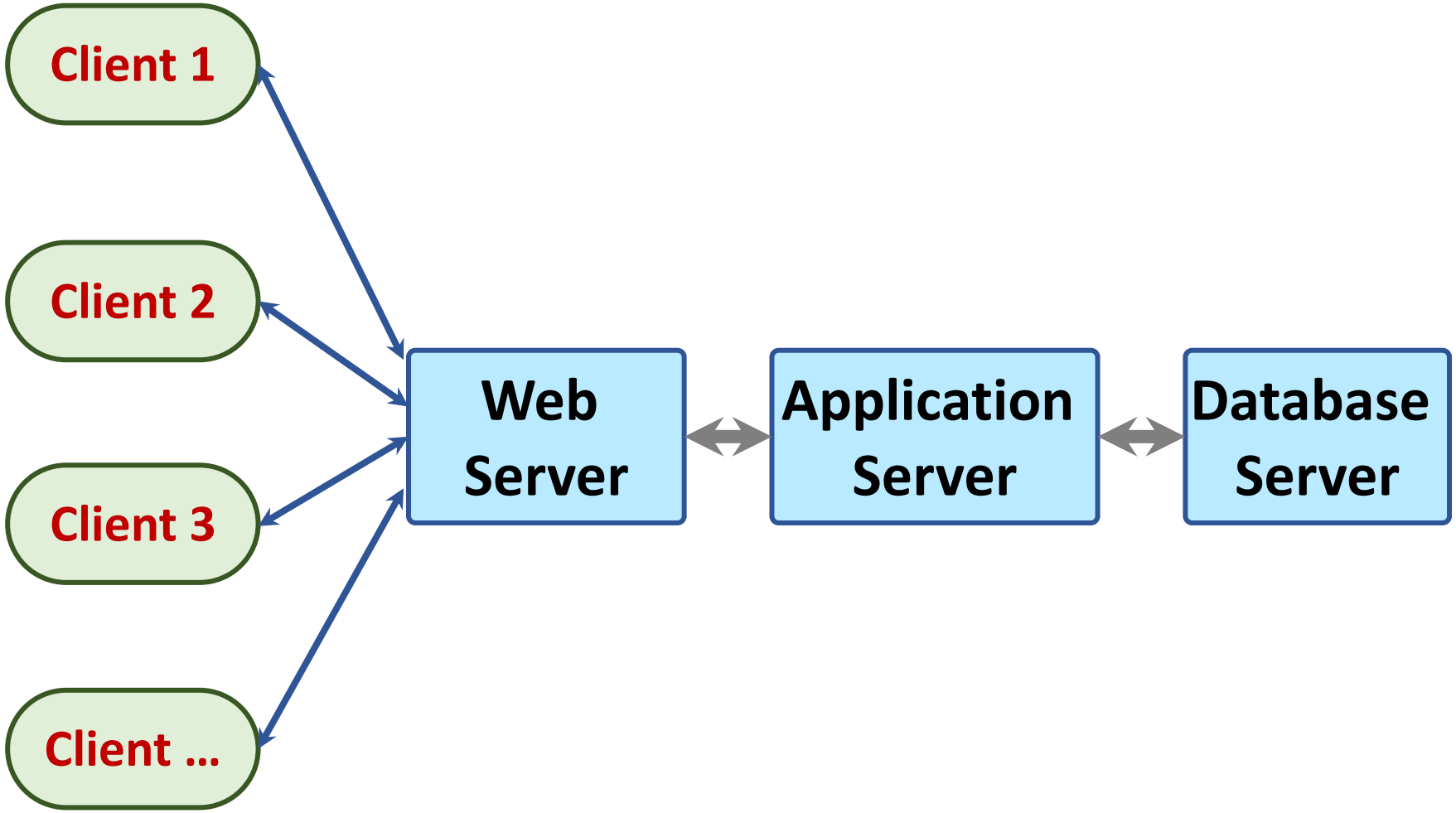
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

From personas to features



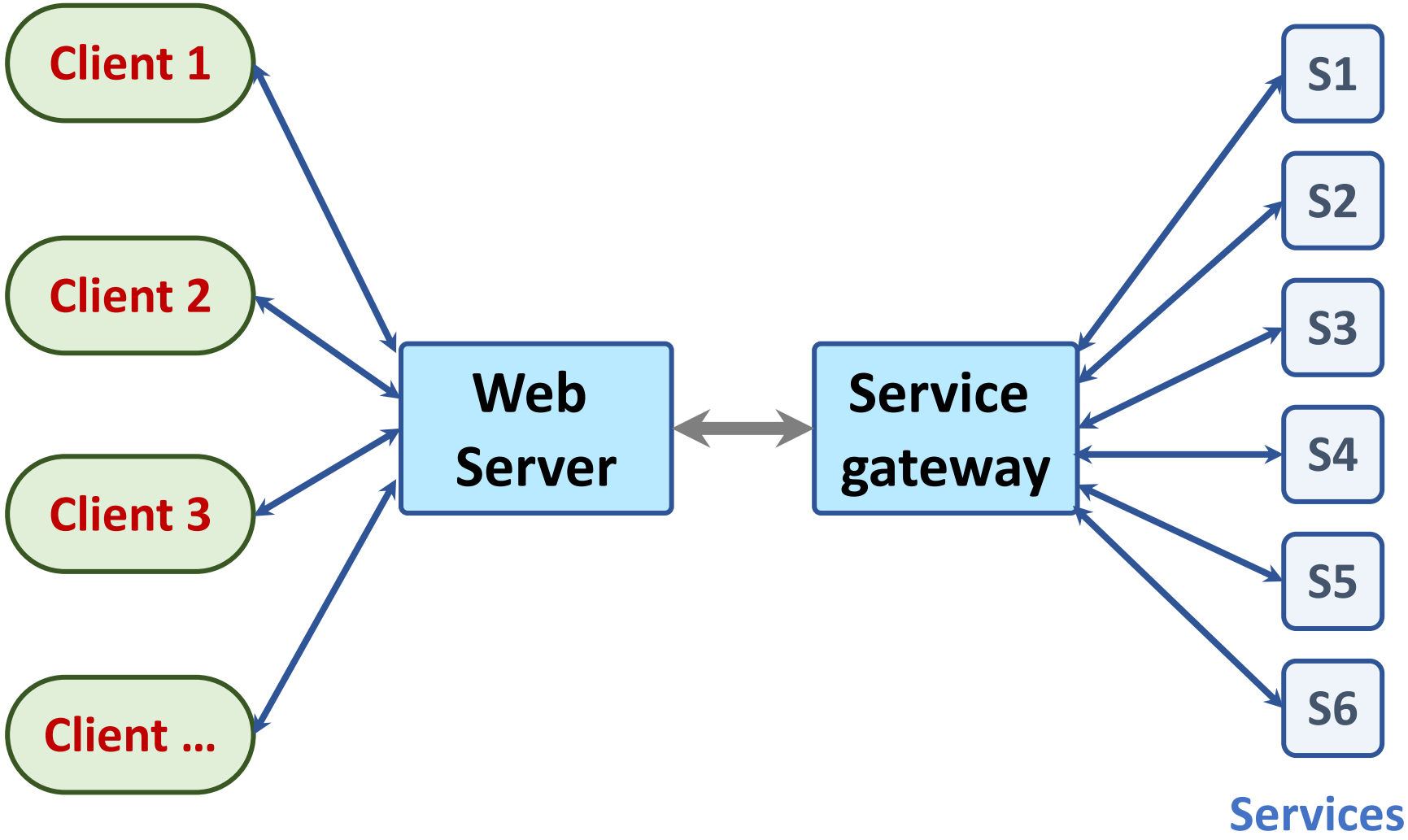
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Multi-tier client-server architecture

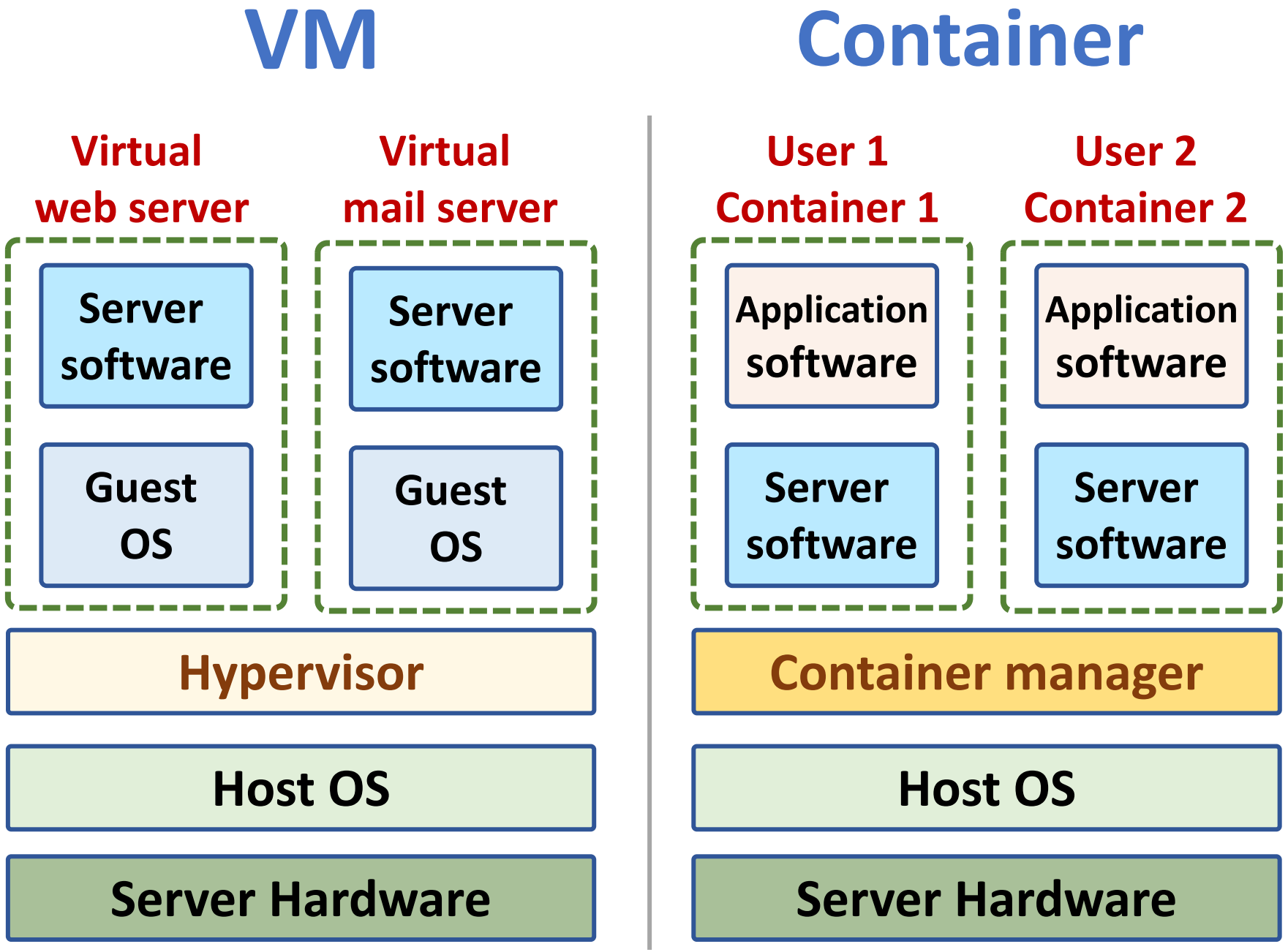


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Service-oriented Architecture

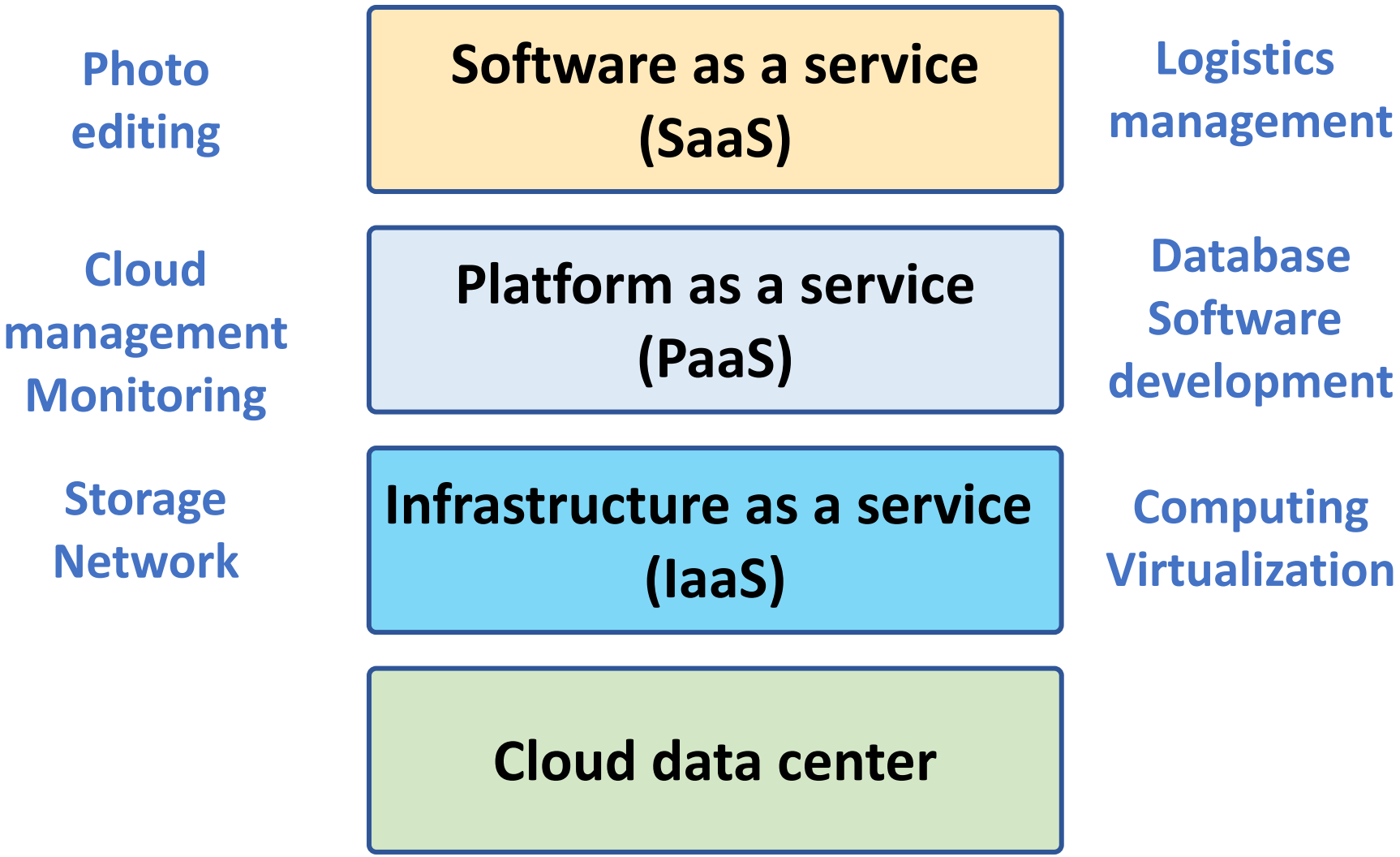


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.



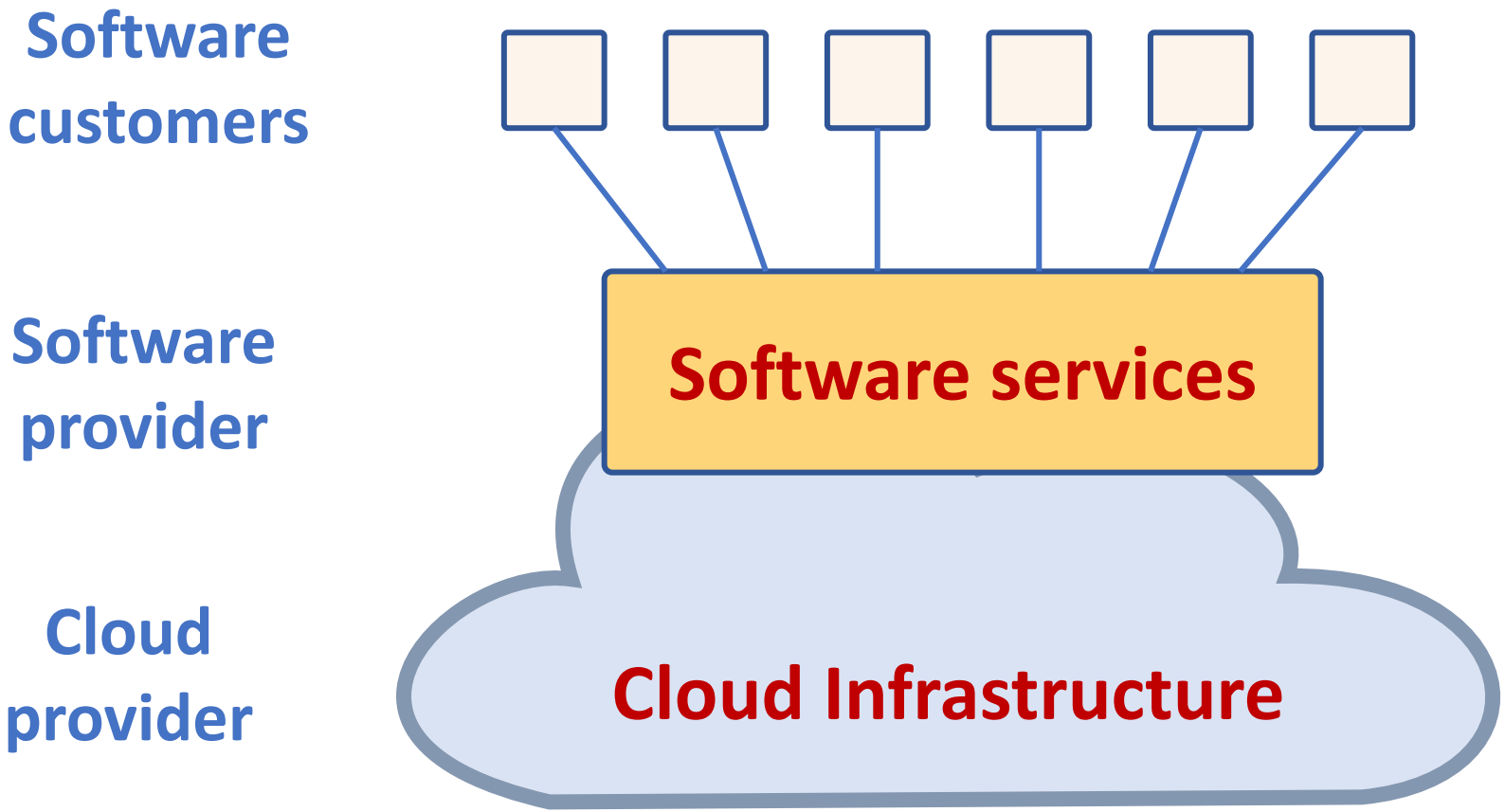
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Everything as a service



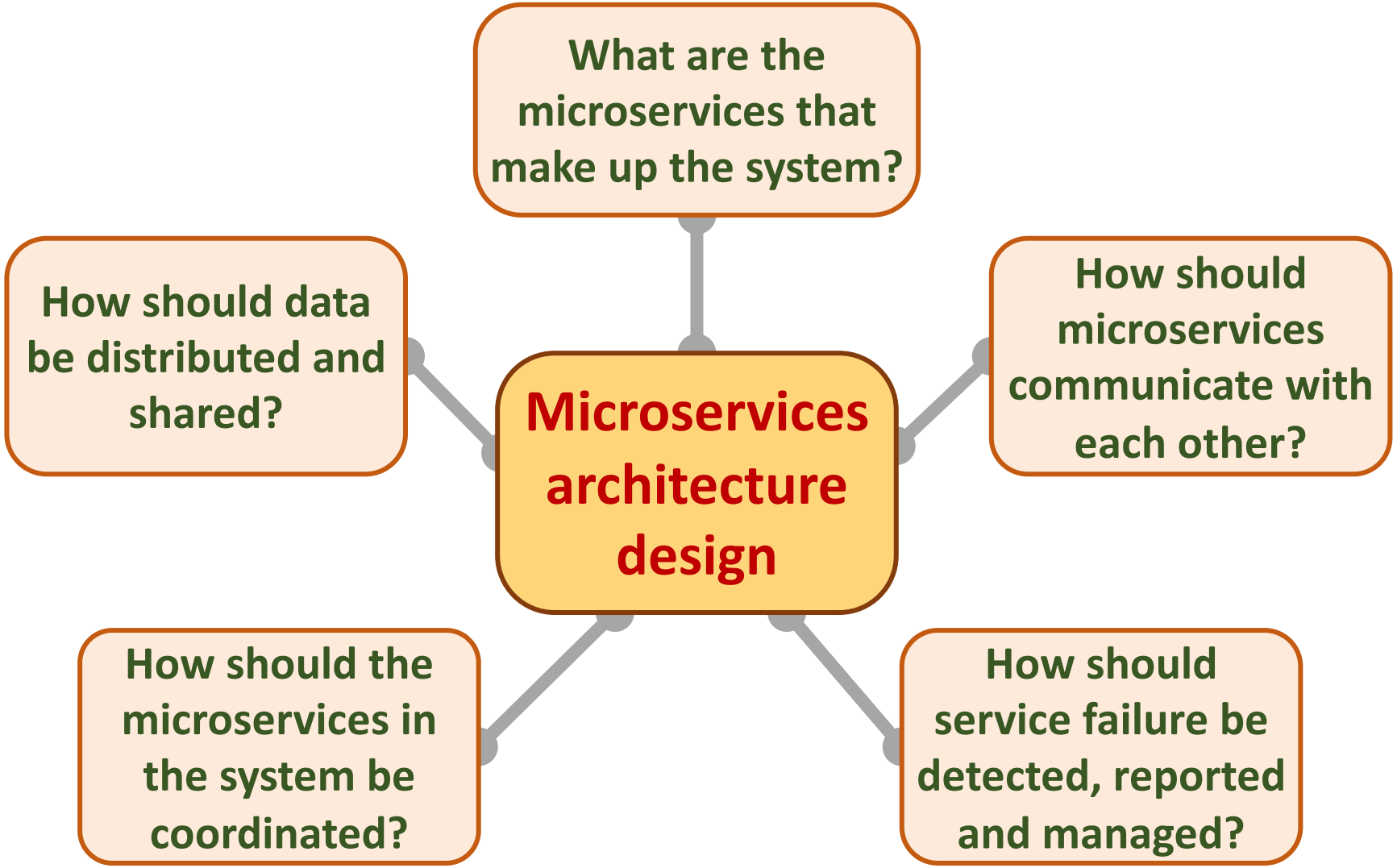
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software as a service



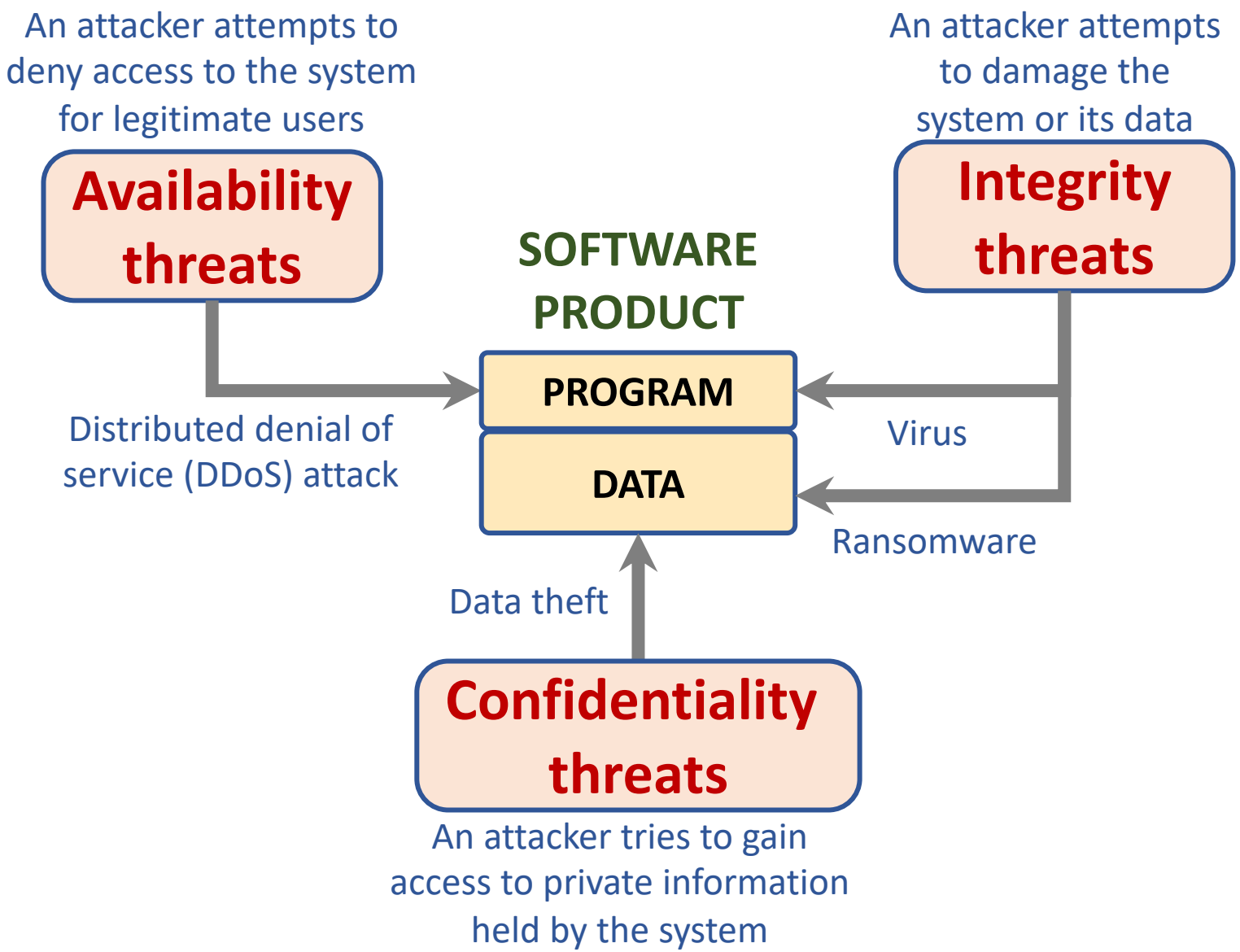
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Microservices architecture – key design questions



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Types of security threat

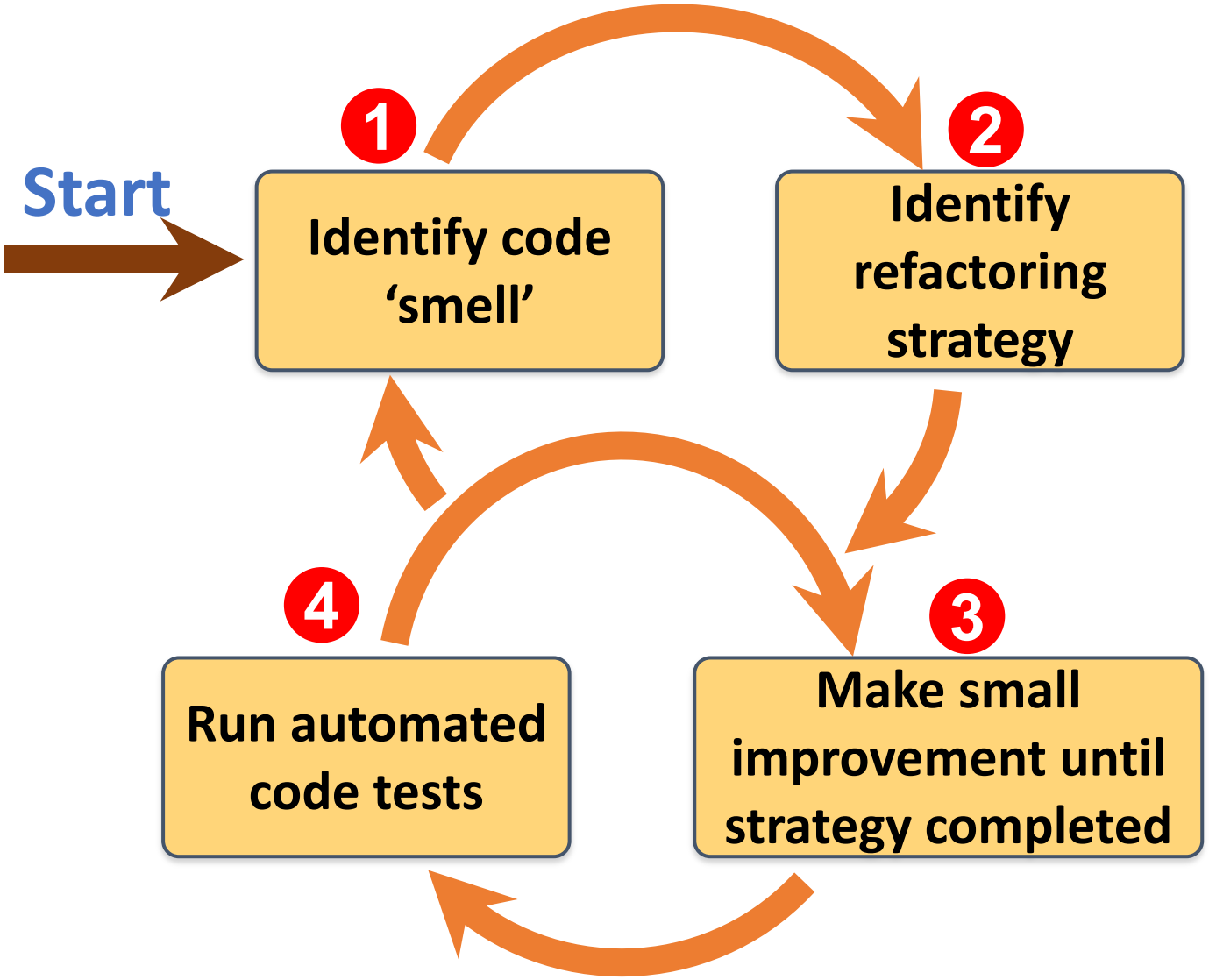


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software product quality attributes

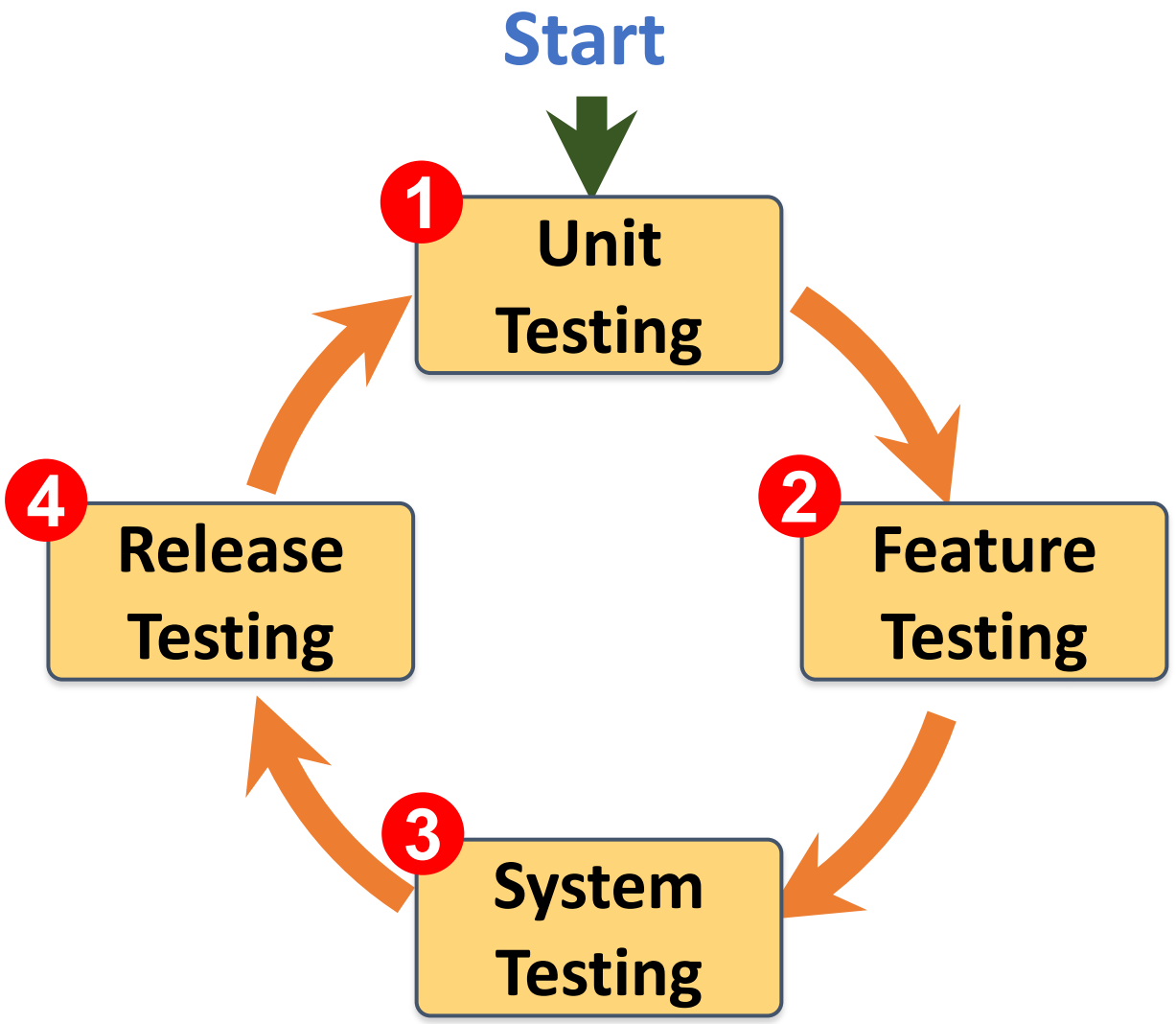
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

A refactoring process



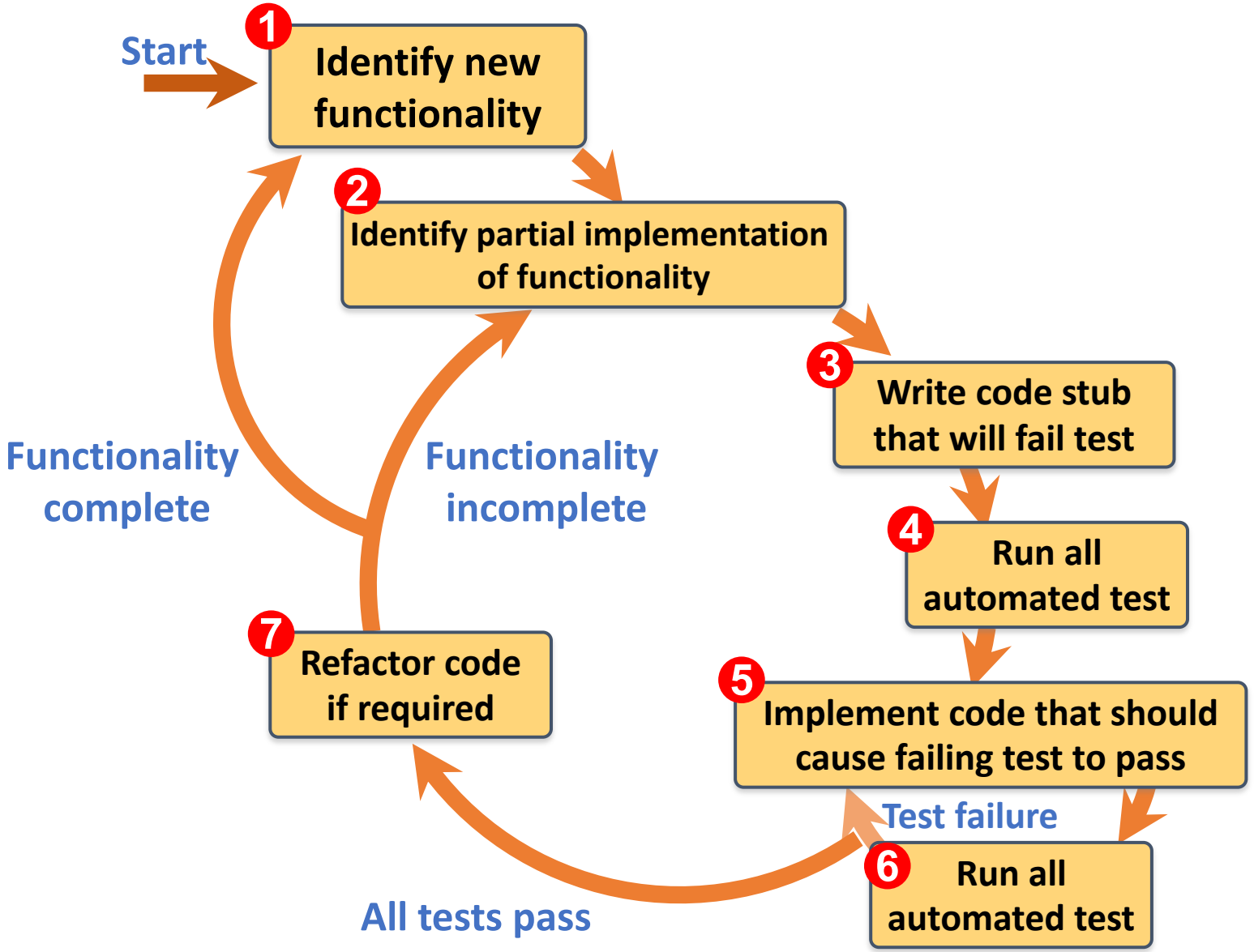
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Functional testing



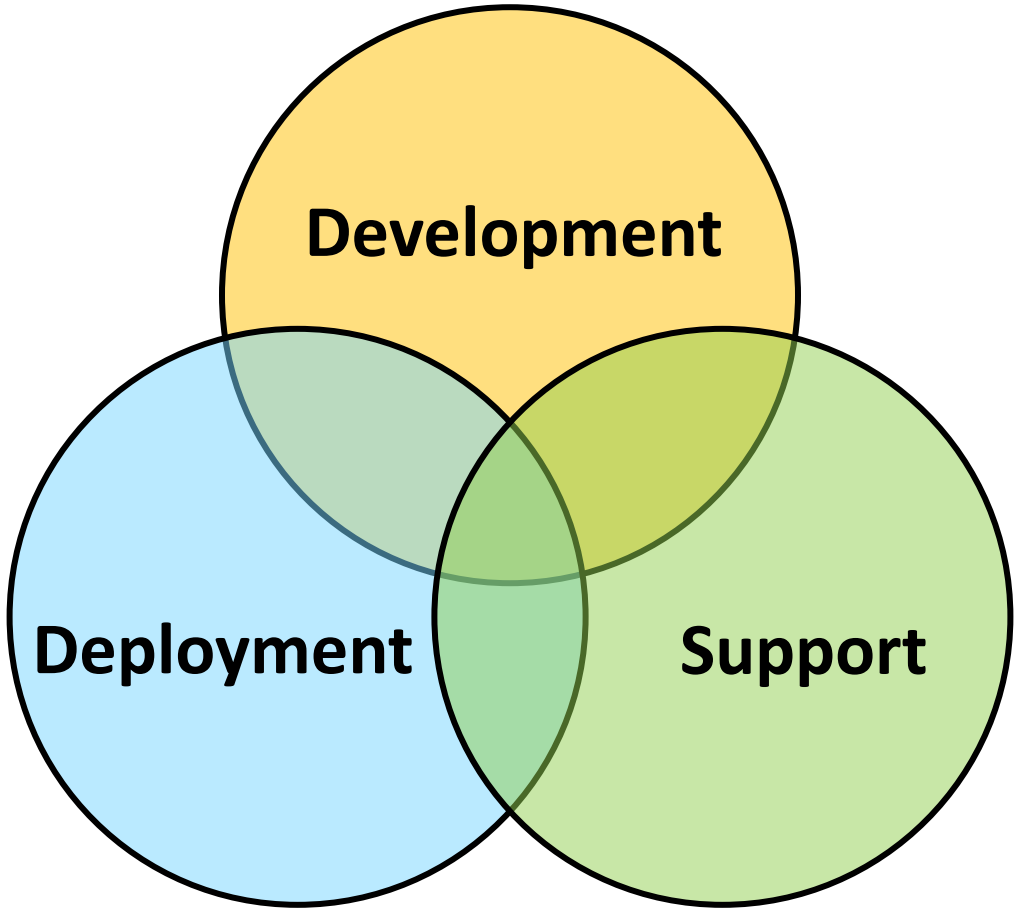
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Test-driven development (TDD)



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

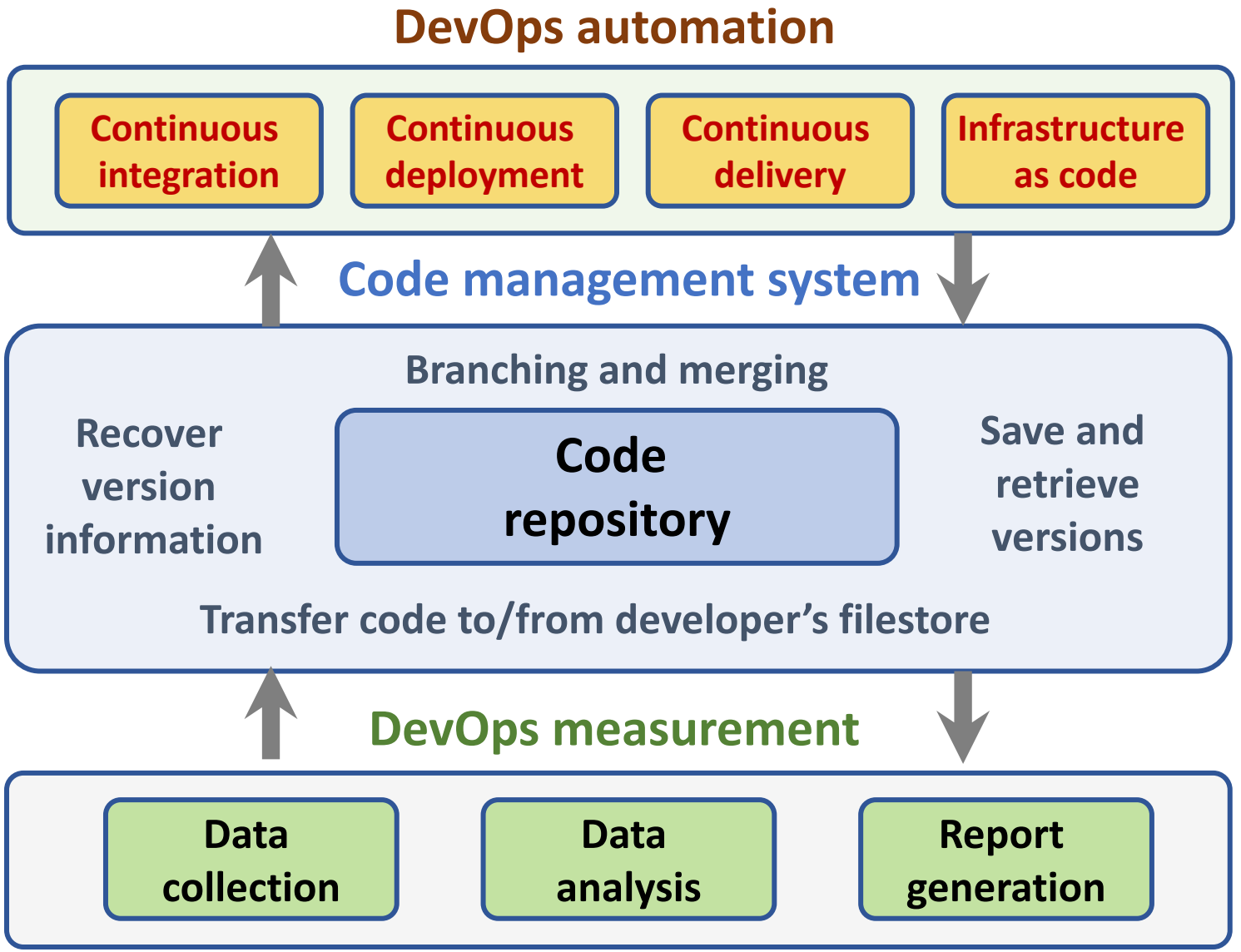
DevOps



Multi-skilled DevOps team

Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Code management and DevOps



Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Cloud Computing and Cloud Software Architecture

Cloud Software Architecture for Agentic AI

Cloud Software Architecture for Agentic AI

Microservices and Serverless Architecture

Containers (Docker, Kubernetes)
Serverless platforms (AWS Lambda, Google Cloud Functions)

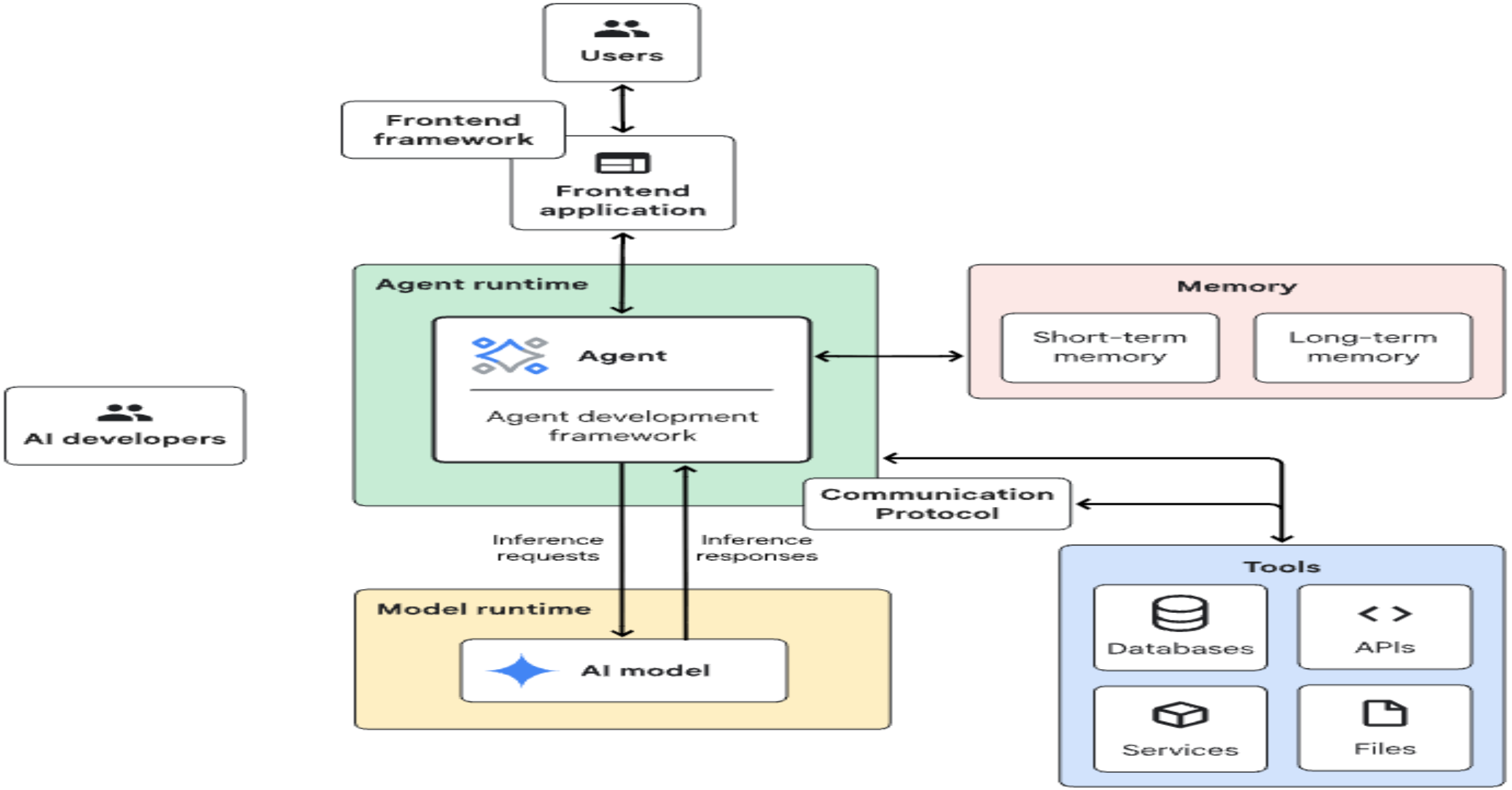
APIs and Tooling Integration via MCP

Agents access tools (e.g., databases, APIs, CRMs, payment gateways)
using Model Context Protocol (MCP)
Enhances tool-using behavior of LLM agents

Tools and Frameworks

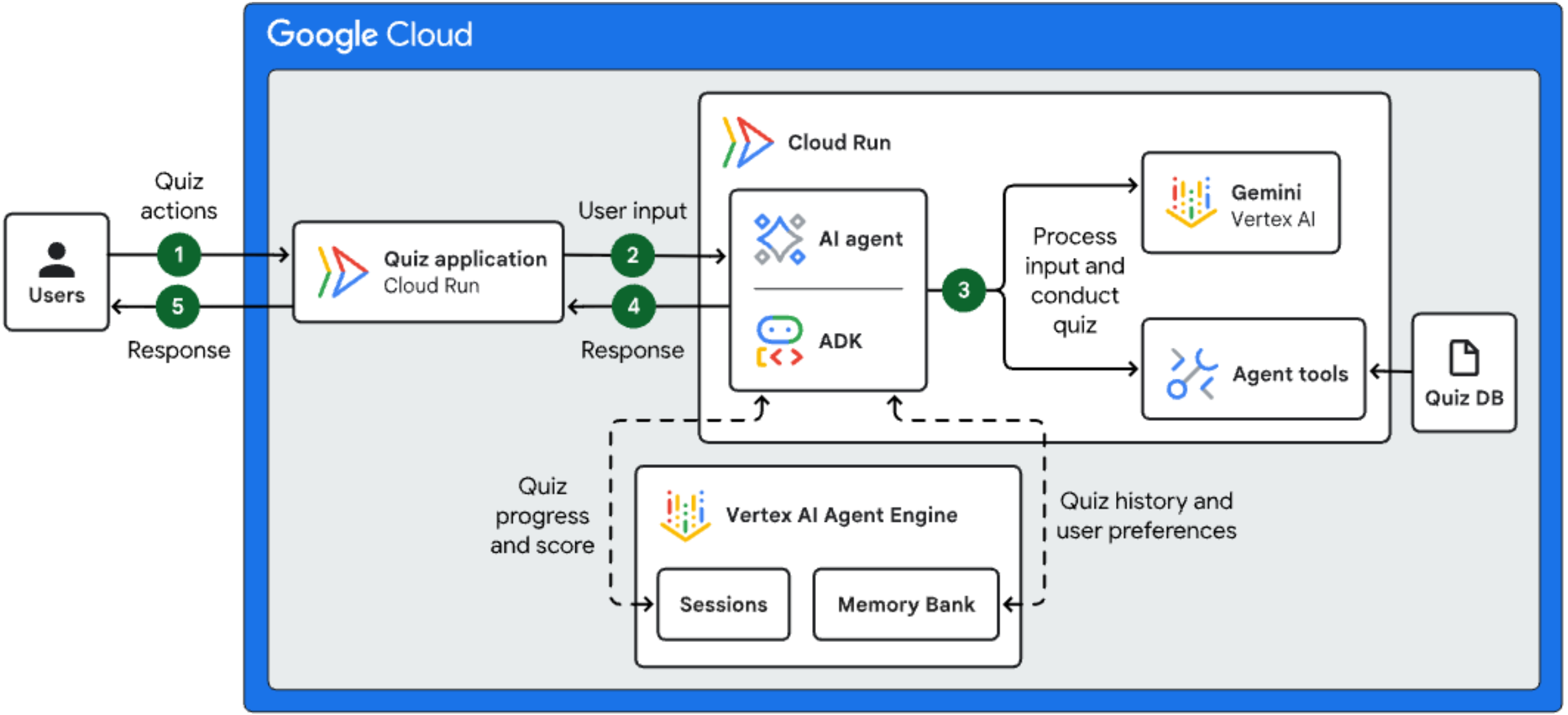
LangChain, AutoGen, CrewAI: for orchestrating LLM agents
Anthropic's MCP, Google's A2A: communication protocols
Vector DBs (Pinecone, Weaviate): for agent memory

Agentic AI System Architecture



Source: https://docs.cloud.google.com/architecture/choose-agentic-ai-architecture-components

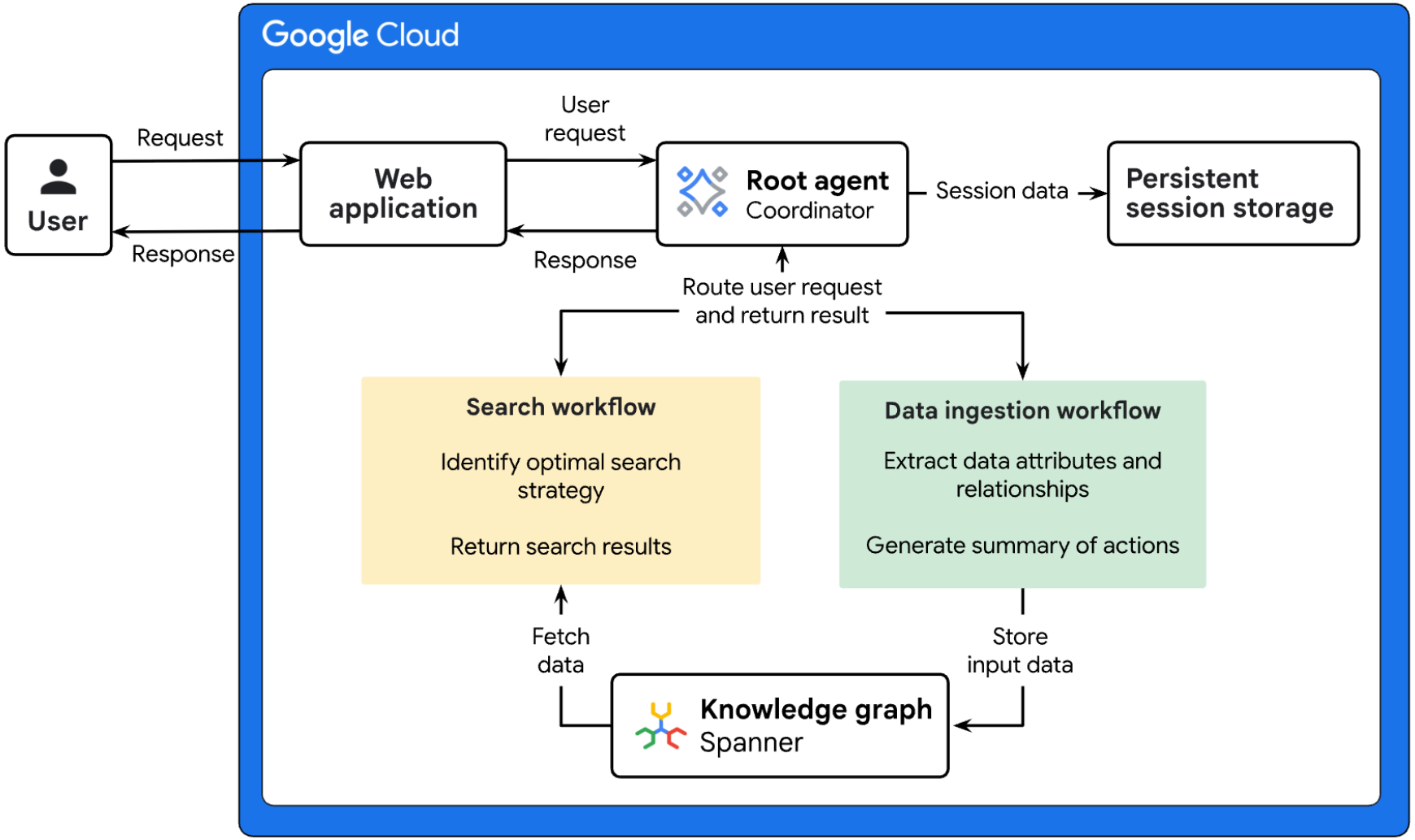
Agentic AI Cloud Architecture: Interactive Learning Agent



Source: https://docs.cloud.google.com/architecture/agentic-ai-interactive-learning

Agentic AI Cloud Architecture:

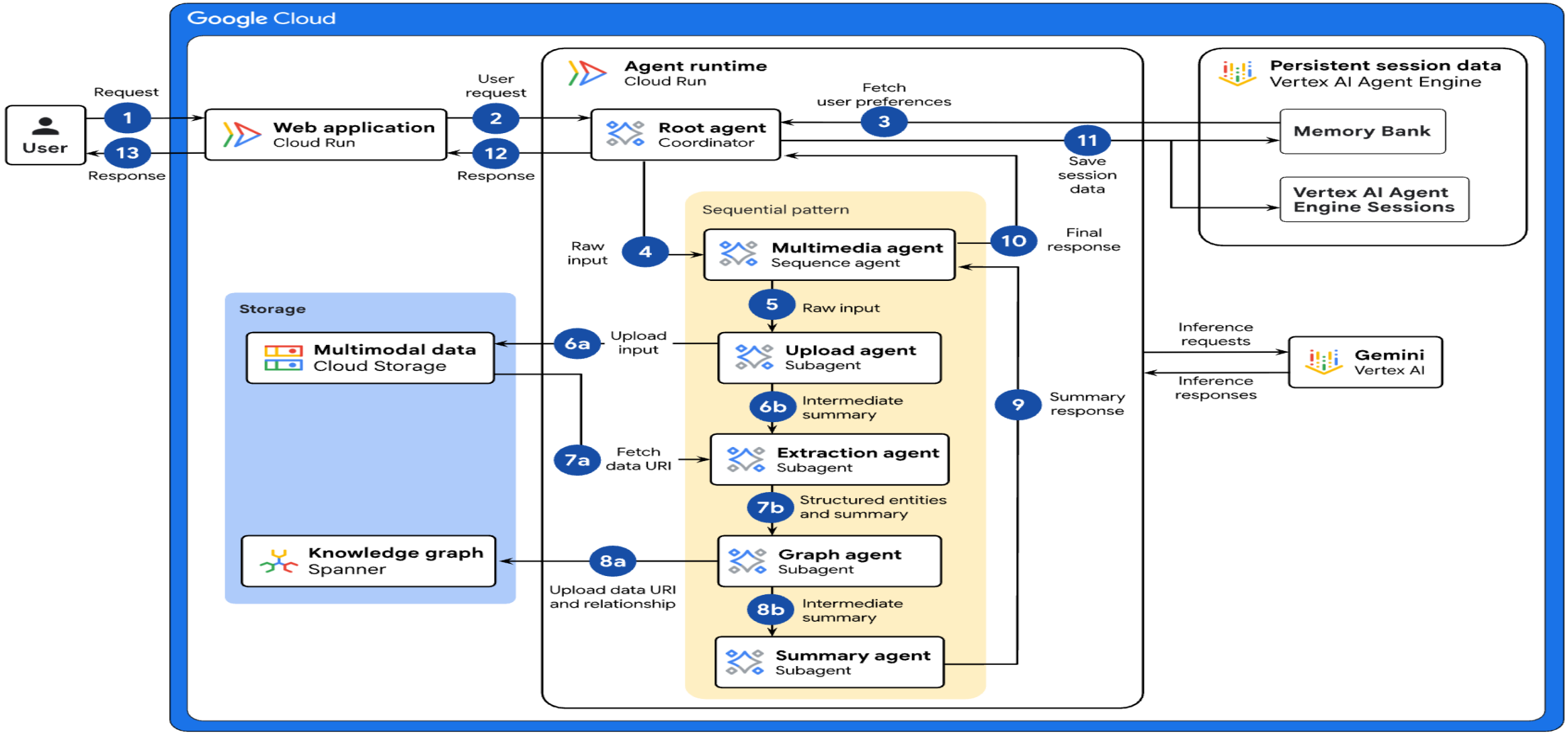
Multi-agent AI System for Multimodal GraphRAG Resource Orchestration



Source: https://docs.cloud.google.com/architecture/agentic-ai-multimodal-graph-rag-resource-orchestration

Agentic AI Cloud Architecture:

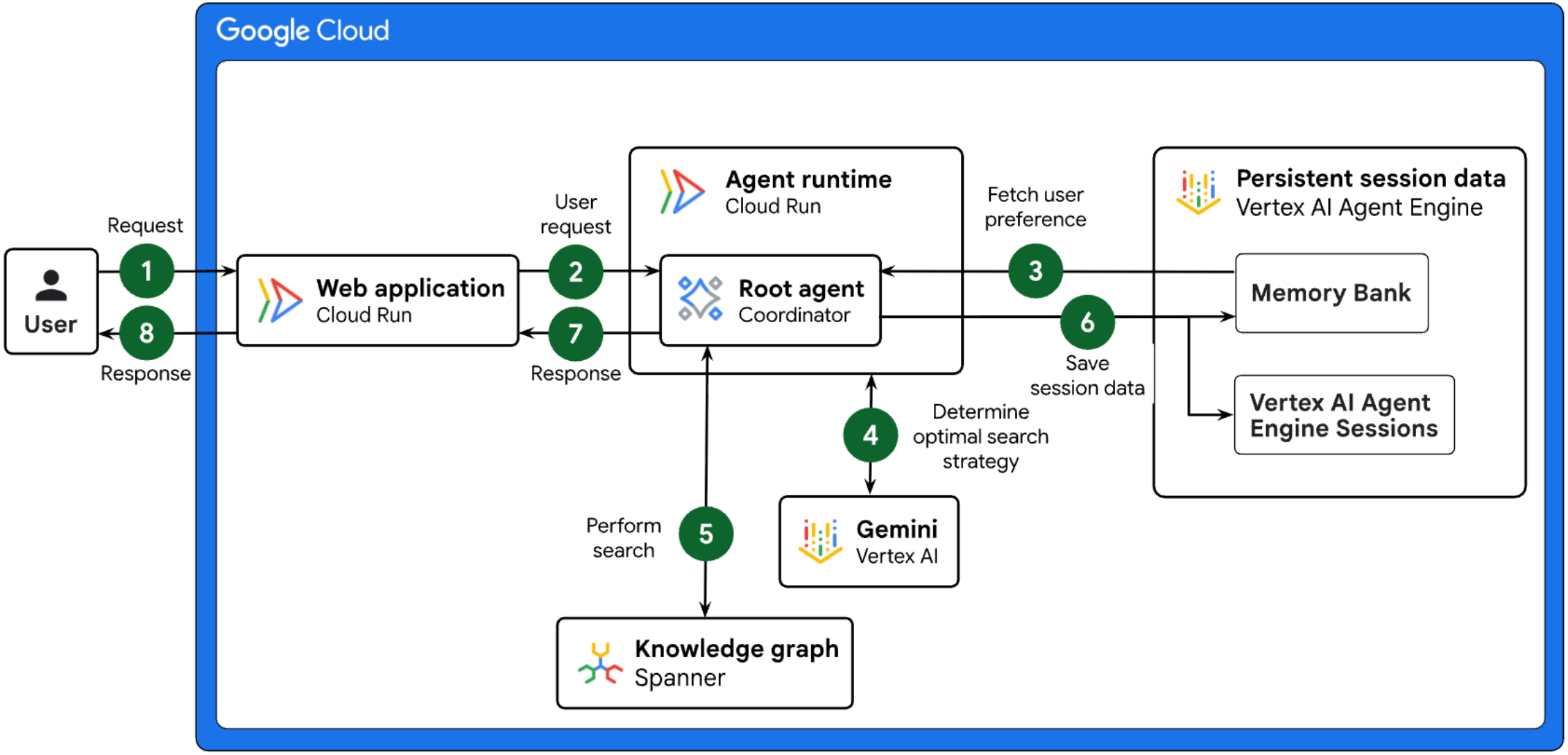
Data Ingestion Workflow for Multimodal GraphRAG Resource Orchestration



Source: https://docs.cloud.google.com/architecture/agentic-ai-multimodal-graph-rag-resource-orchestration

Agentic AI Cloud Architecture:

Search Workflow for Multimodal GraphRAG Resource Orchestration

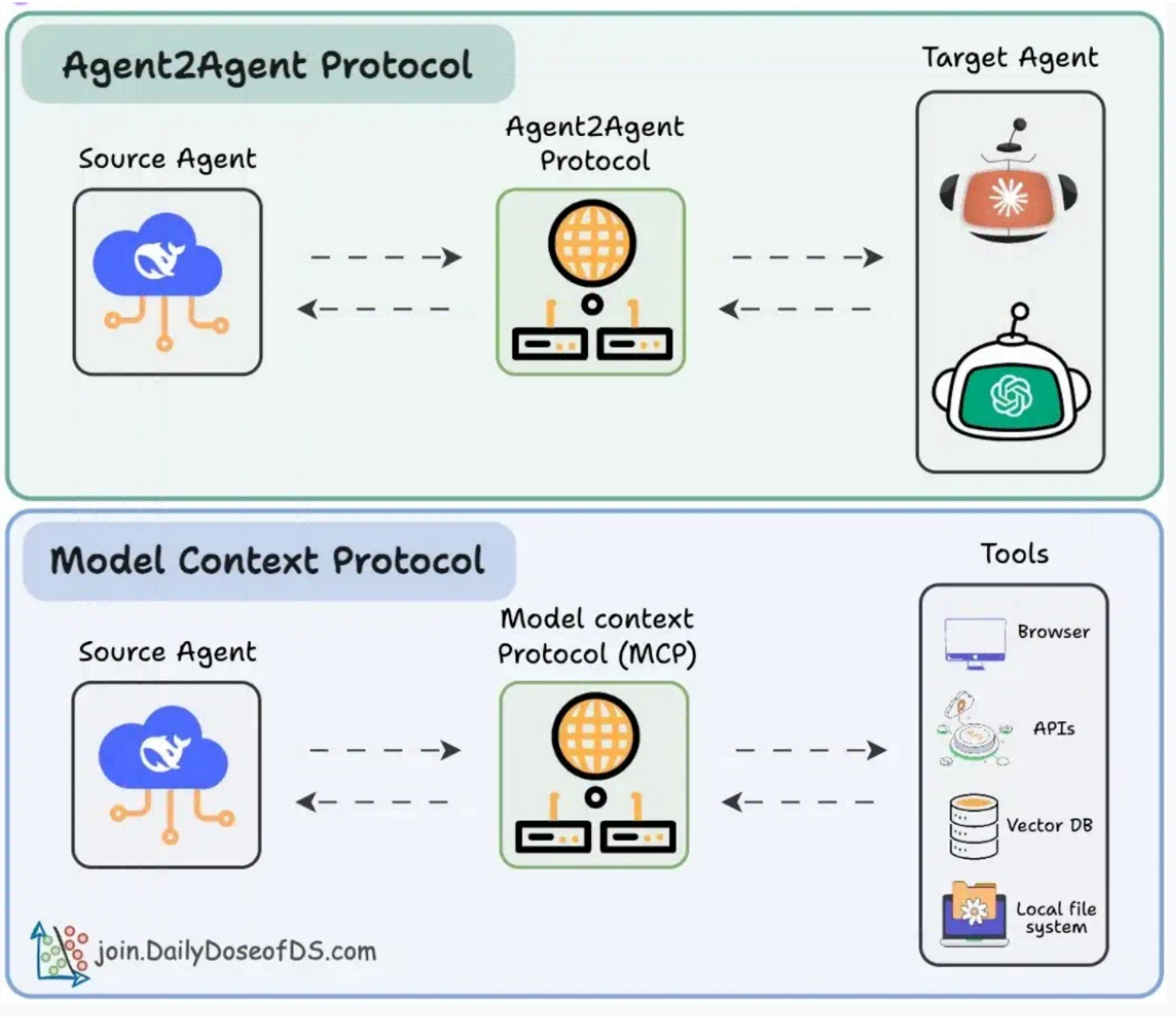


Source: https://docs.cloud.google.com/architecture/agentic-ai-multimodal-graph-rag-resource-orchestration

Agentic applications require both A2A and MCP

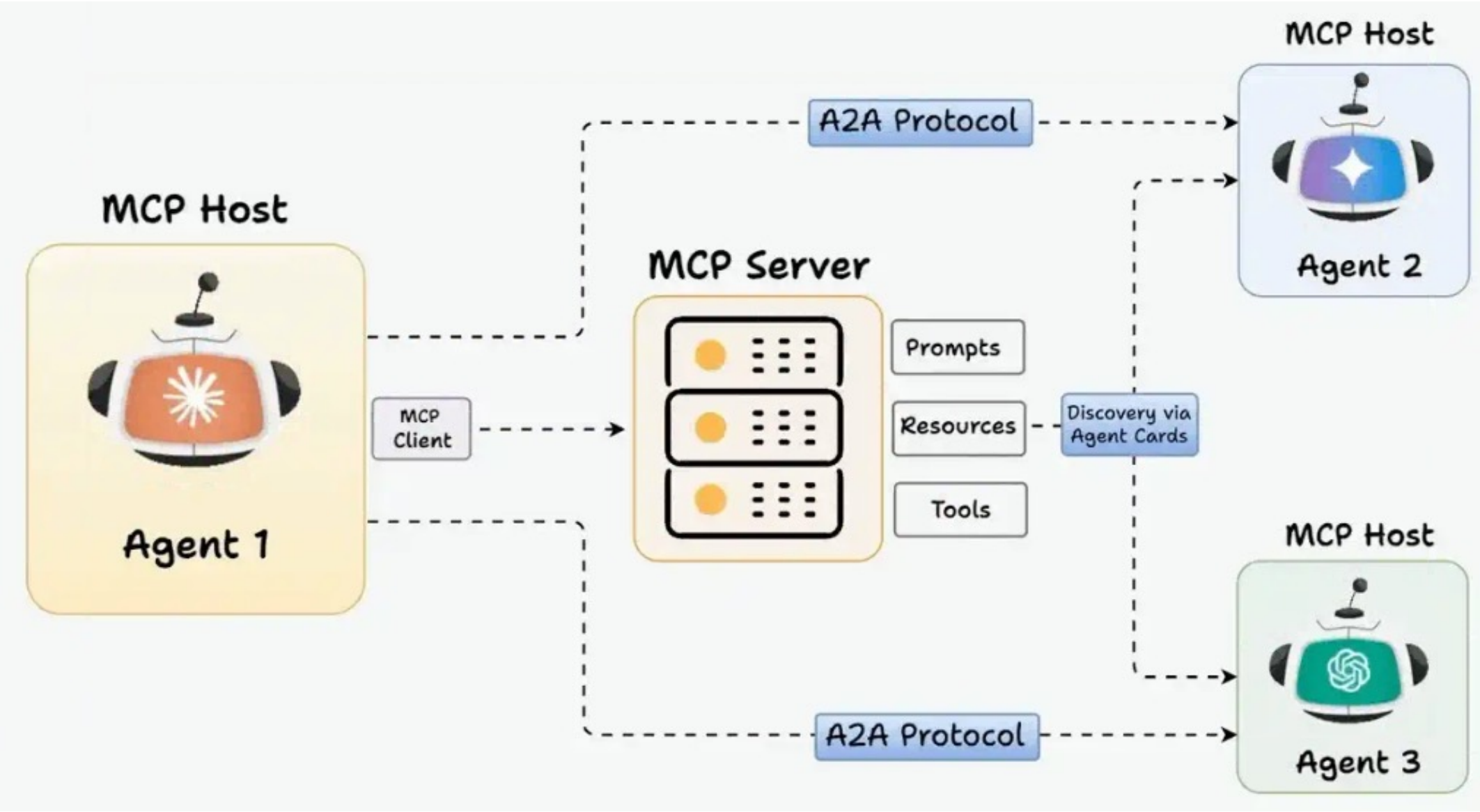
A2A allows agents to connect with other agents and collaborate in teams.

MCP provides agents with access to tools



Source: https://blog.dailydoseofds.com/p/a-visual-guide-to-agent2agent-a2a

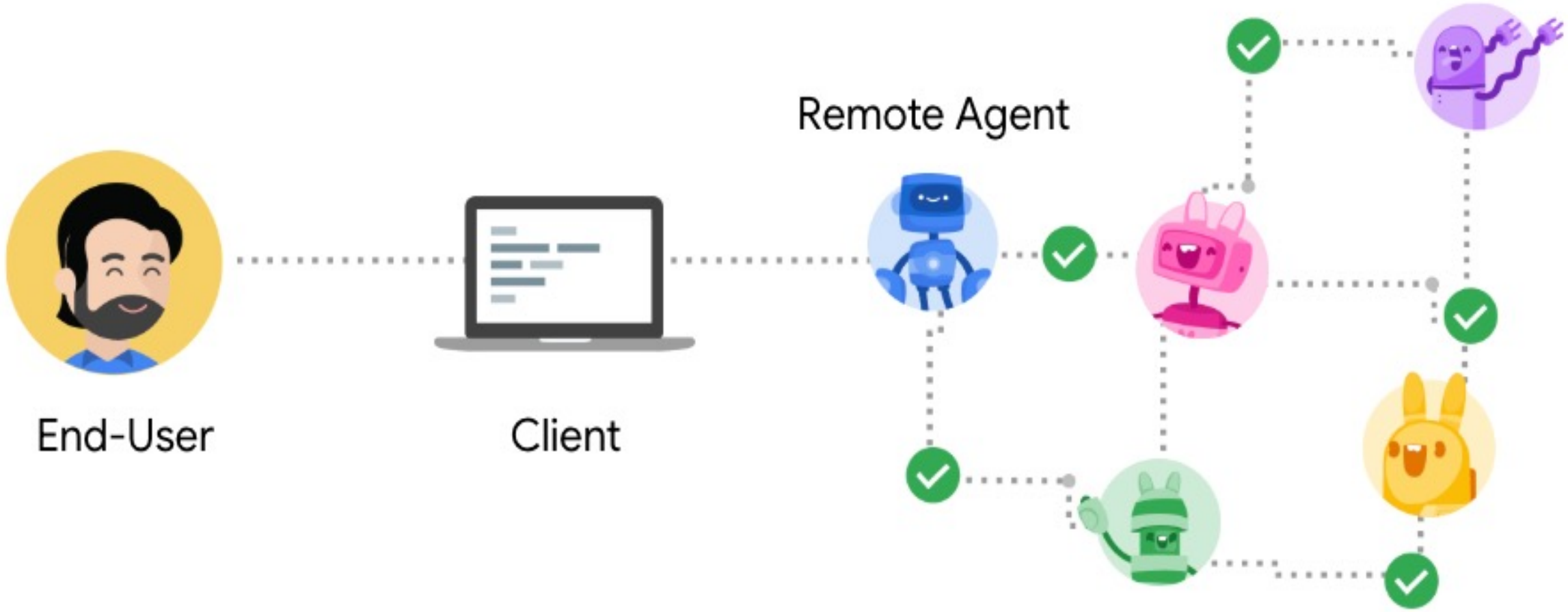
MCP and A2A Protocol for AI Agents



Source: https://blog.dailydoseofds.com/p/a-visual-guide-to-agent2agent-a2a

Agent2Agent Protocol (A2A)

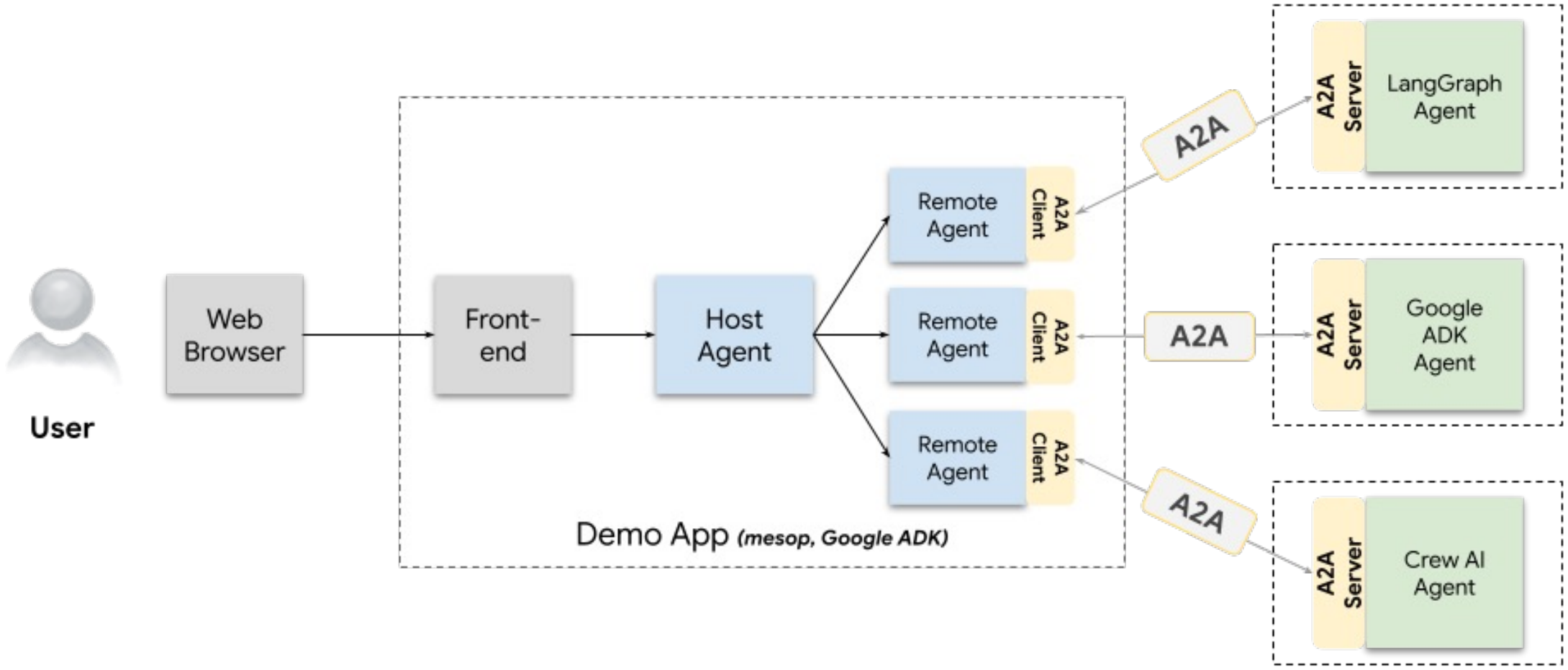
An open protocol enabling Agent-to-Agent interoperability, bridging the gap between opaque agentic systems



Source: https://google.github.io/A2A/

A2A Demo Web App

Agents talking to other agents over A2A



Source: https://github.com/google/A2A/blob/main/demo/README.md

A2A

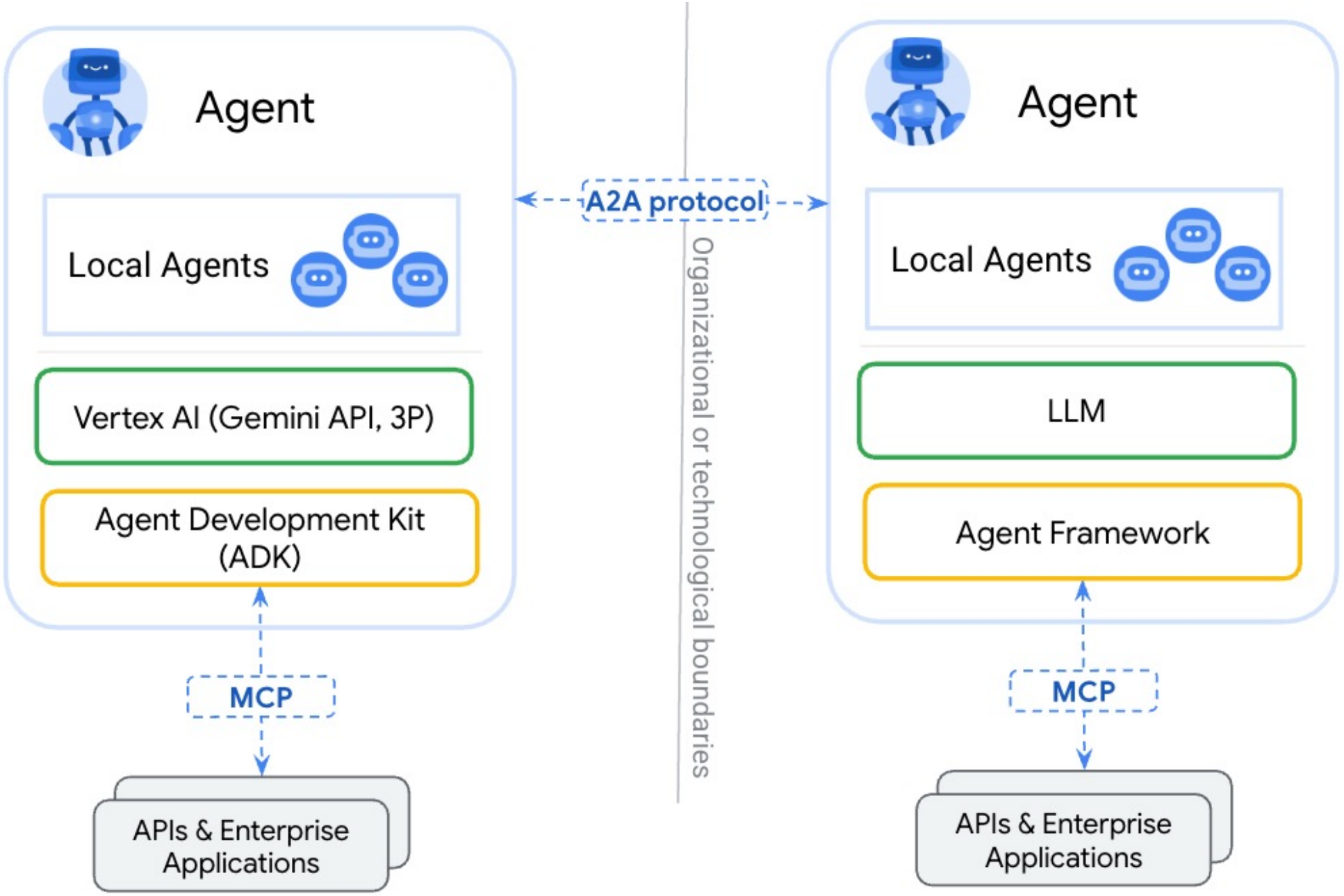
(Agent2Agent Protocol)

for agent-agent collaboration

MCP

(Model Context Protocol)

for tools and resources



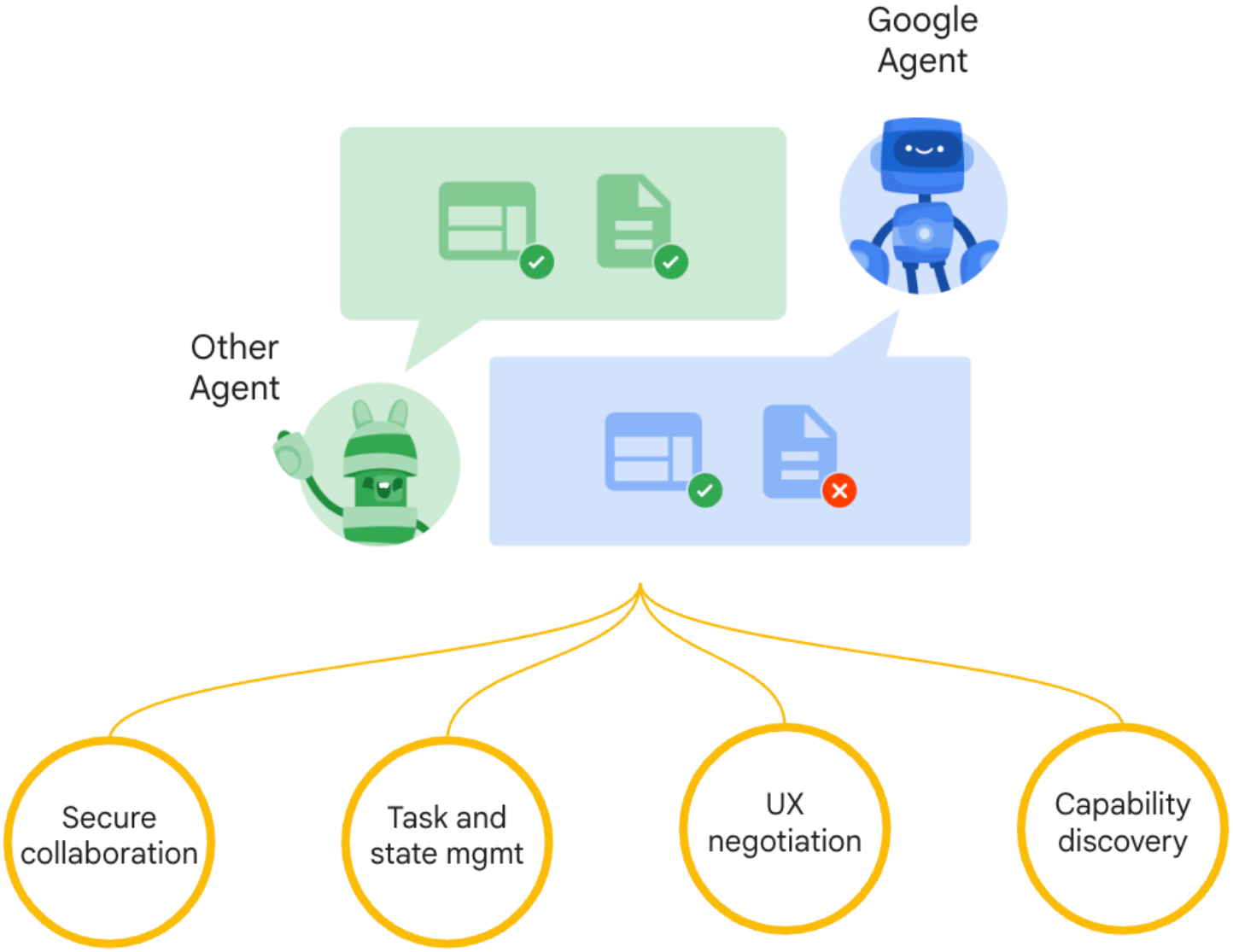
Source: https://google.github.io/A2A/

Google A2A (Agent2Agent Protocol)

Seamless Agent Collaboration

Simplifies Enterprise Agent Integration

Supports Key Enterprise Requirements

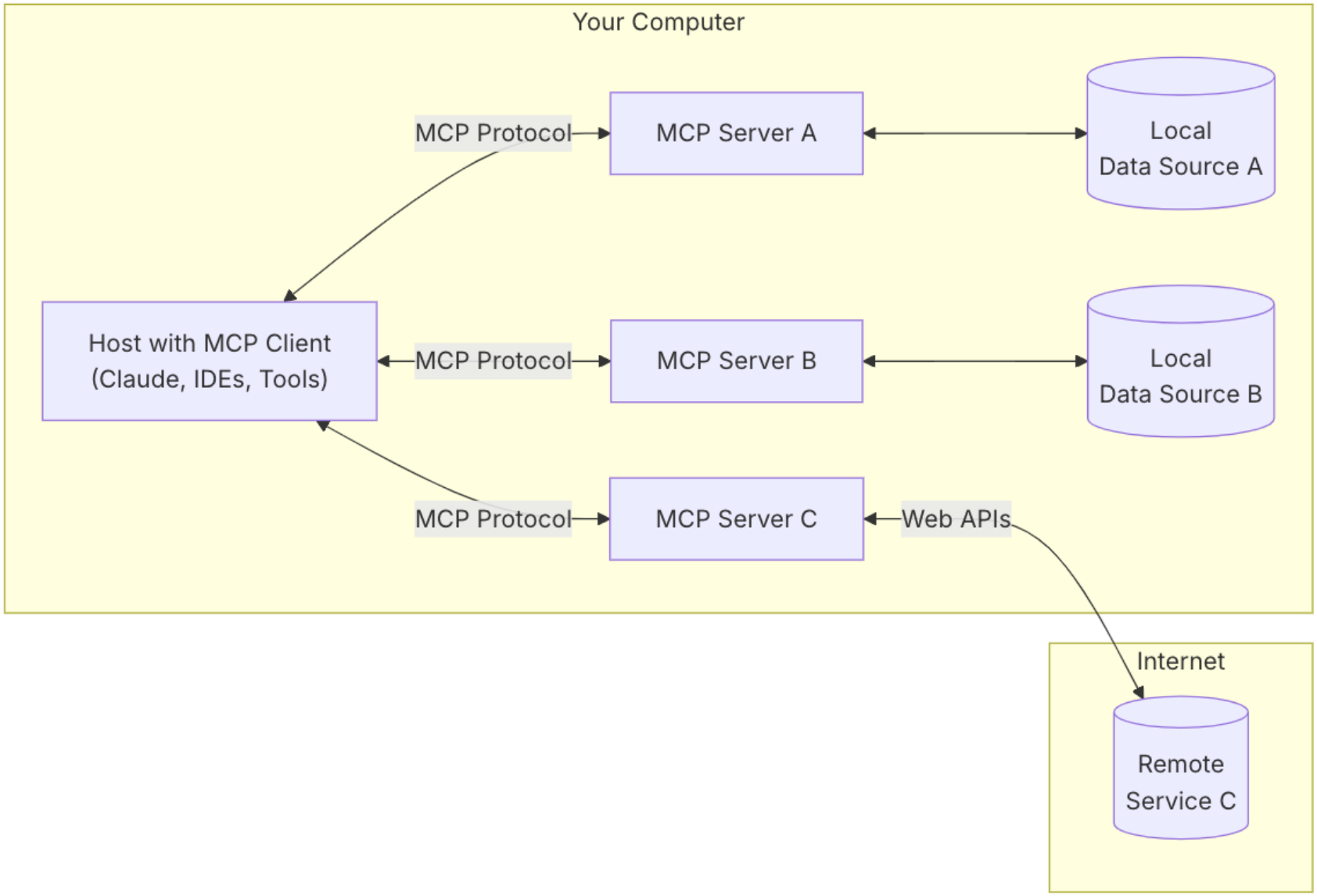


Source: https://google.github.io/A2A/

MCP (Model Context Protocol)

MCP is a open protocol that standardizes how applications provide context to LLMs.

MCP: USB-C port for AI applications.



Source: https://modelcontextprotocol.io/introduction

MCP and A2A

- MCP (Model Context Protocol) for tools and resources
  - Connect agents to tools, APIs, and resources with structured inputs/outputs.
  - Google ADK supports MCP tools. Enabling wide range of MCP servers to be used with agents.
- A2A (Agent2Agent Protocol) for agent-agent collaboration
  - Dynamic, multimodal communication between different agents without sharing memory, resources, and tools
  - Open standard driven by community.
  - Samples available using Google ADK, LangGraph, Crew.AI

Source: https://google.github.io/A2A/

Web Application with AWS Core Services

AWS Products and Services


Analytics


Application Integration


AR & VR


AWS Cost Management


Blockchain


Business Applications


Compute


Customer Engagement


Database


Developer Tools


End User Computing


Game Tech


Internet of Things


Machine Learning


Management & Governance


Media Services


Migration & Transfer


Mobile


Networking & Content Delivery


Quantum Technologies


Robotics


Satellite


Security, Identity & Compliance


Storage

Source: https://aws.amazon.com/

AWS Compute

Amazon EC2
Virtual servers in the cloud

Amazon EC2 Auto Scaling
Scale compute capacity to meet demand

Amazon Elastic Container Registry
Store and retrieve docker images

Amazon Elastic Container Service
Run and manage docker containers

Amazon Elastic Kubernetes Service
Run managed Kubernetes on AWS

Amazon Lightsail
Launch and manage virtual private servers

AWS Batch
Run batch jobs at any scale

AWS Elastic Beanstalk
Run and manage web apps

AWS Fargate
Run containers without managing servers or clusters

AWS Lambda
Run code without thinking about servers

AWS Outposts
Run AWS infrastructure on-premises

AWS Serverless Application Repository
Discover, deploy, and publish serverless applications

AWS Wavelength
Deliver ultra-low latency applications for 5G devices

VMware Cloud on AWS
Build a hybrid cloud without custom hardware

Source: https://aws.amazon.com/

AWS Database

Amazon Aurora
High Performance Managed Relational Database

Amazon DynamoDB
Managed NoSQL Database

Amazon DocumentDB (with MongoDB compatibility)
Fully managed document database

Amazon ElastiCache
In-memory Caching System

Amazon Managed Apache Cassandra Service
Managed Cassandra-compatible database

Amazon Neptune
Fully Managed Graph Database Service

Amazon Quantum Ledger Database (QLDB)
Fully managed ledger database

Amazon RDS
Managed Relational Database Service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB

Amazon RDS on VMware
Automate on-premises database management

Amazon Redshift
Fast, Simple, Cost-effective Data Warehousing

Amazon Timestream
Fully managed time series database

AWS Database Migration Service
Migrate Databases with Minimal Downtime

Source: https://aws.amazon.com/

AWS Storage

Amazon Simple Storage Service (S3)
Scalable Storage in the Cloud

Amazon Elastic Block Store (EBS)
EC2 block storage volumes

Amazon Elastic File System (EFS)
Fully managed file system for EC2

Amazon FSx for Lustre
High-performance file system integrated with S3

Amazon FSx for Windows File Server
Fully managed Windows native file system

Amazon S3 Glacier
Low-cost Archive Storage in the Cloud

AWS Backup
Centralized backup across AWS services

AWS Snow Family
Physical devices to migrate data into and out of AWS

AWS Storage Gateway
Hybrid Storage Integration

CloudEndure Disaster Recovery
Highly automated disaster recovery

Source: https://aws.amazon.com/

AWS Networking & Content Dilivery

Amazon VPC
Isolated Cloud Resources

Amazon API Gateway
Build, Deploy, and Manage APIs

Amazon CloudFront
Global Content Delivery Network

Amazon Route 53
Scalable Domain Name System

AWS PrivateLink
Securely Access Services Hosted on AWS

AWS App Mesh
Monitor and control microservices

AWS Cloud Map
Application resource registry for microservices

AWS Direct Connect
Dedicated Network Connection to AWS

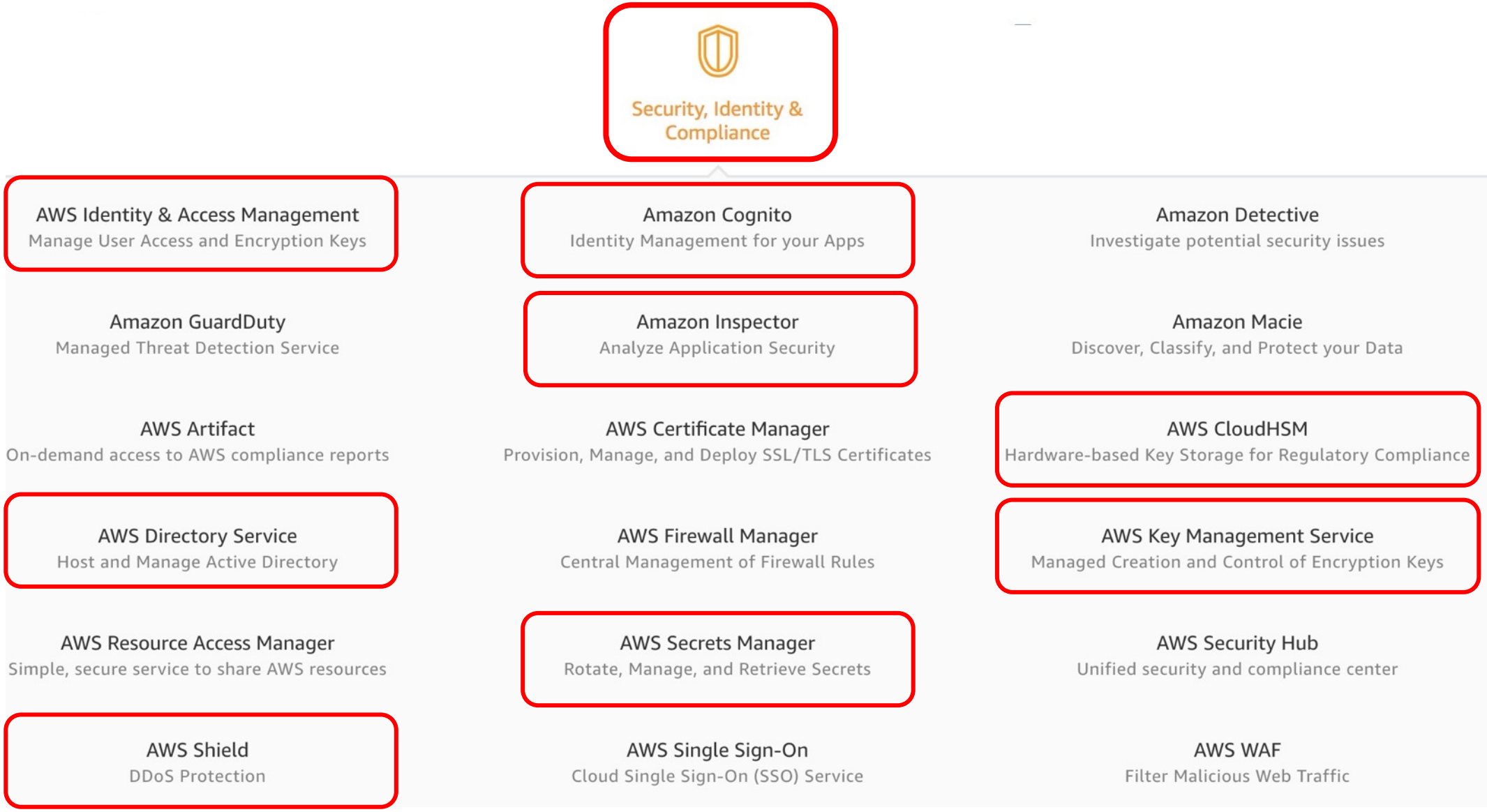
AWS Global Accelerator
Improve application availability and performance

AWS Transit Gateway
Easily scale VPC and account connections

Elastic Load Balancing
Distribute incoming traffic across multiple targets

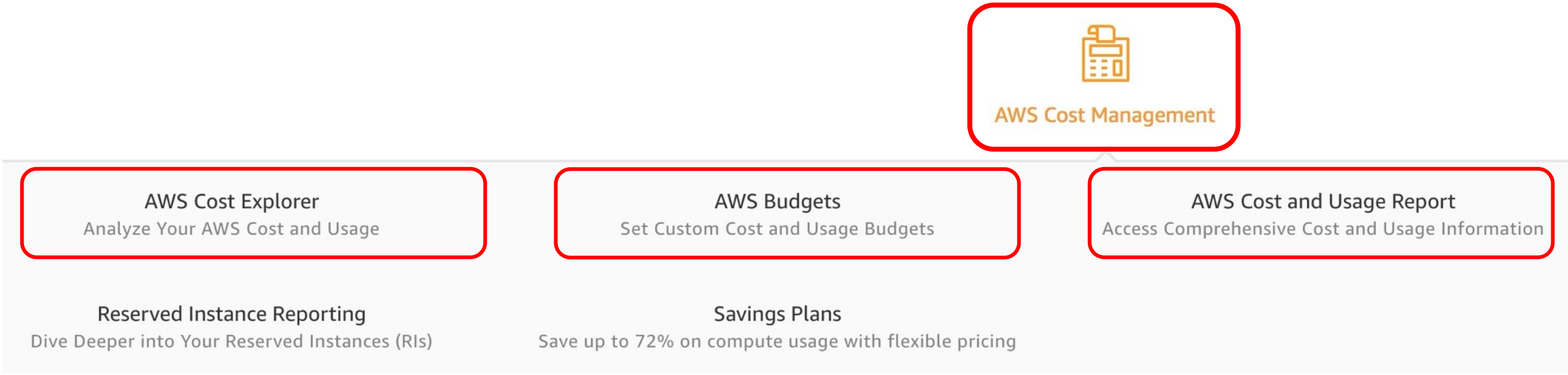
Source: https://aws.amazon.com/

AWS Security, Identity & Compliance



Source: https://aws.amazon.com/

AWS Cost Management



Source: https://aws.amazon.com/

AWS Services

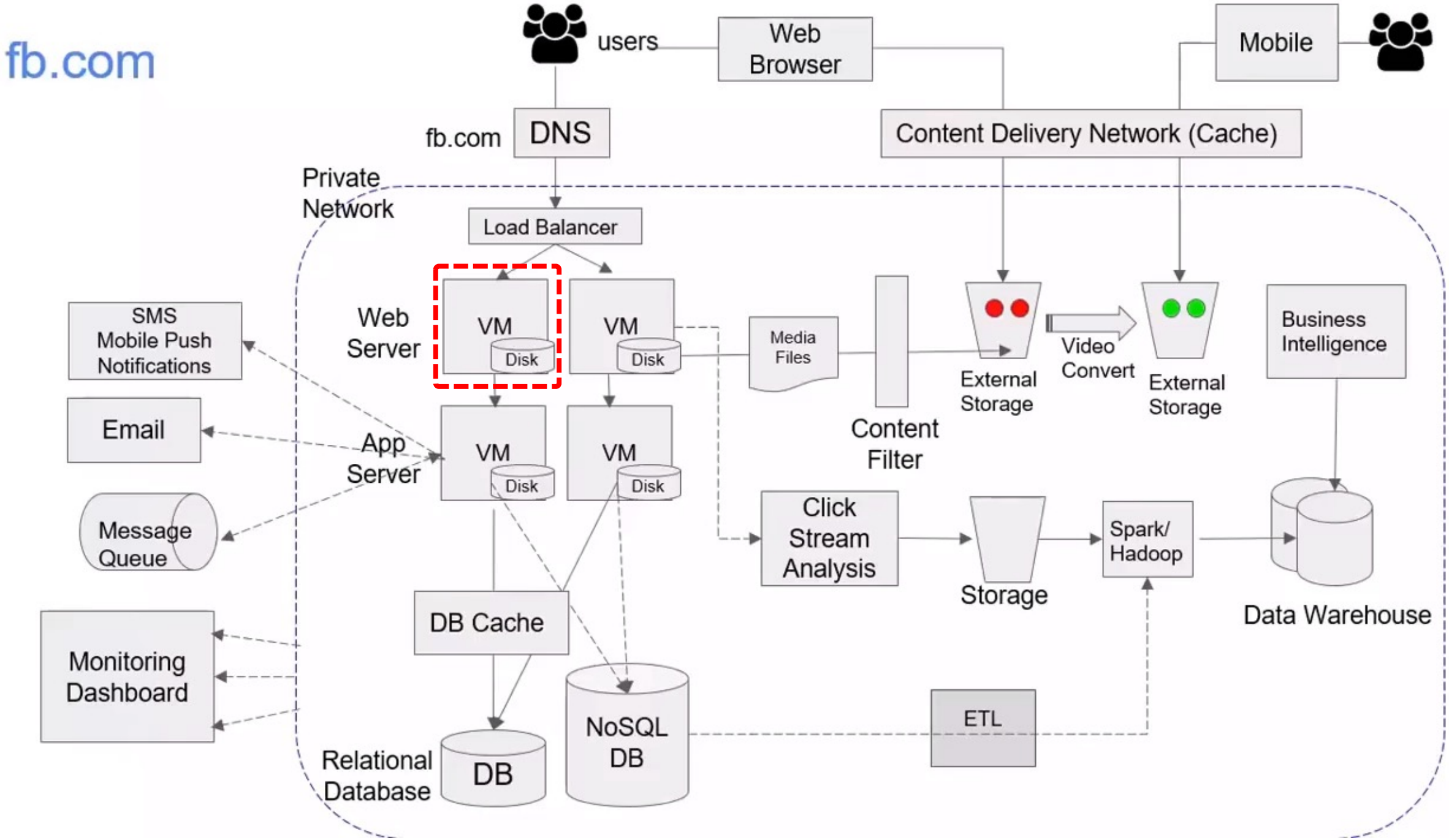
- Amazon EC2
  - Virtual servers in the cloud
- Amazon Simple Storage Service (S3)
  - Scalable storage in the cloud
- Amazon Aurora
  - High performance managed relational database
- Amazon DynamoDB
  - Managed NoSQL database
- Amazon RDS
  - Managed relational database service for MySQL, PostgreSQL, Oracle, SQL Server, and MariaDB

Source: https://aws.amazon.com/

AWS Services

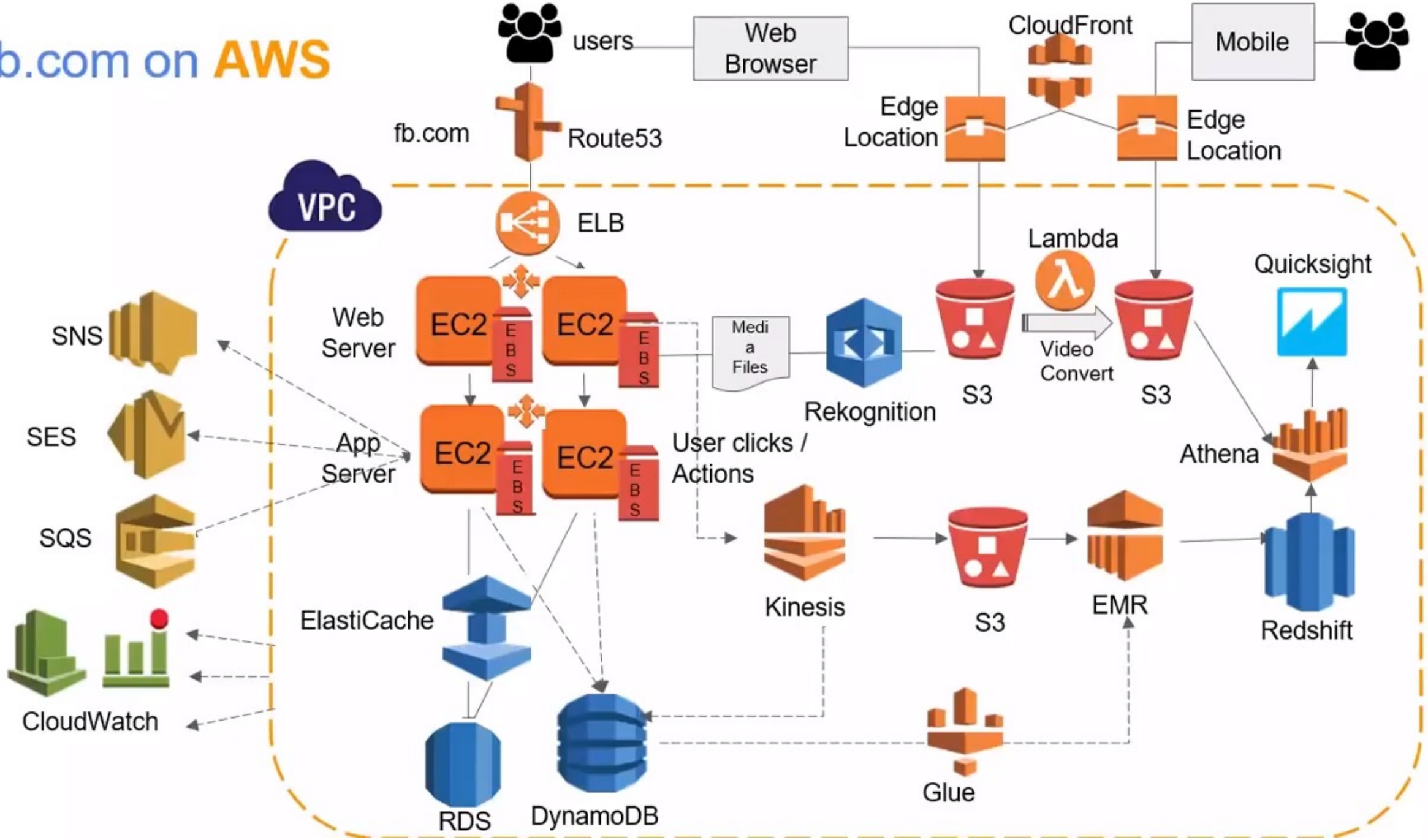
- AWS Lambda
  - Run code without thinking about servers
- AWS Elastic Beanstalk
  - Run and manage web apps
- Amazon VPC
  - Isolated cloud resources
- Amazon Lightsail
  - Launch and manage virtual private servers
- Amazon SageMaker
  - Build, train, and deploy machine learning models at scale

Source: https://aws.amazon.com/



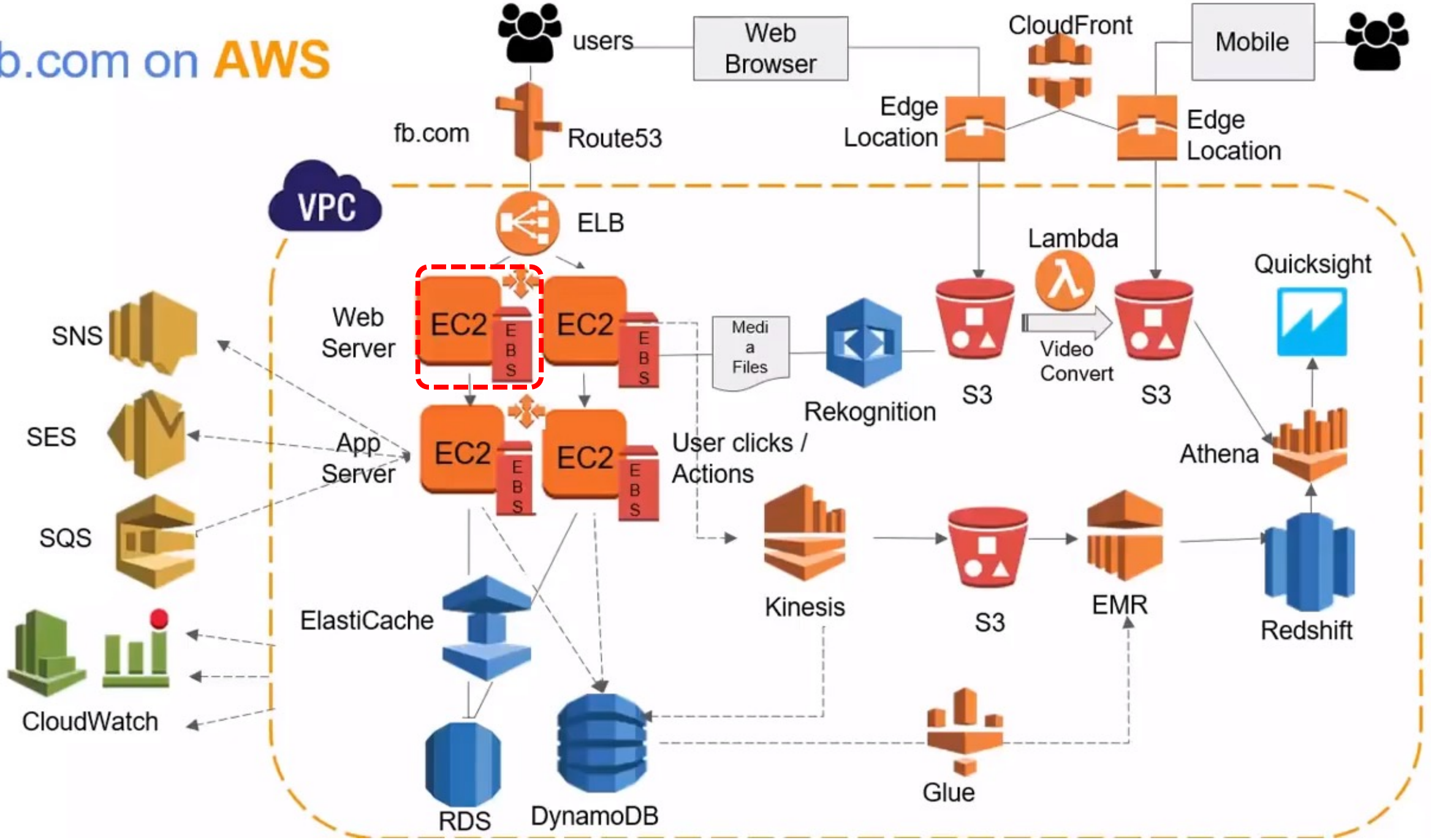
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

fb.com on AWS

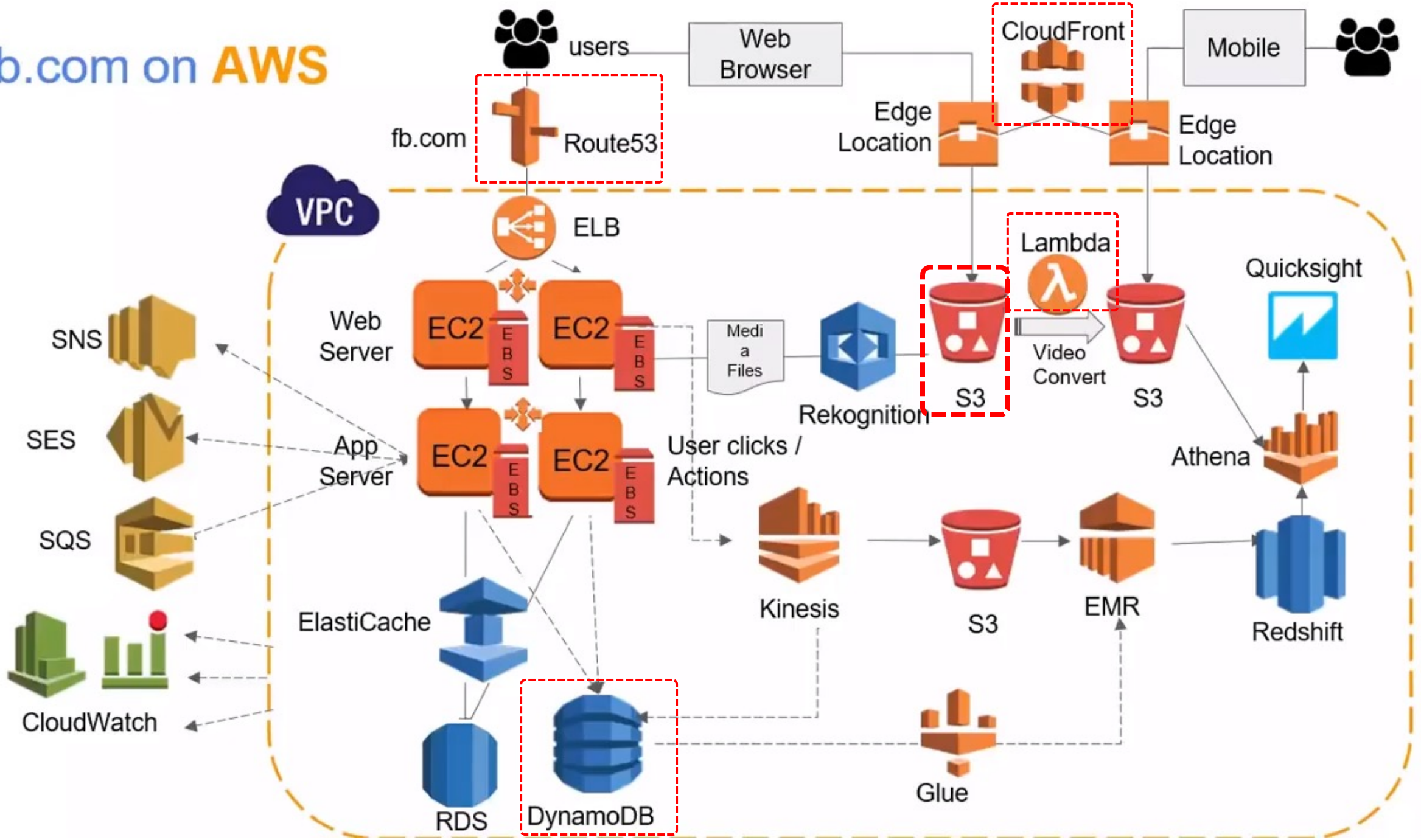


Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

fb.com on AWS

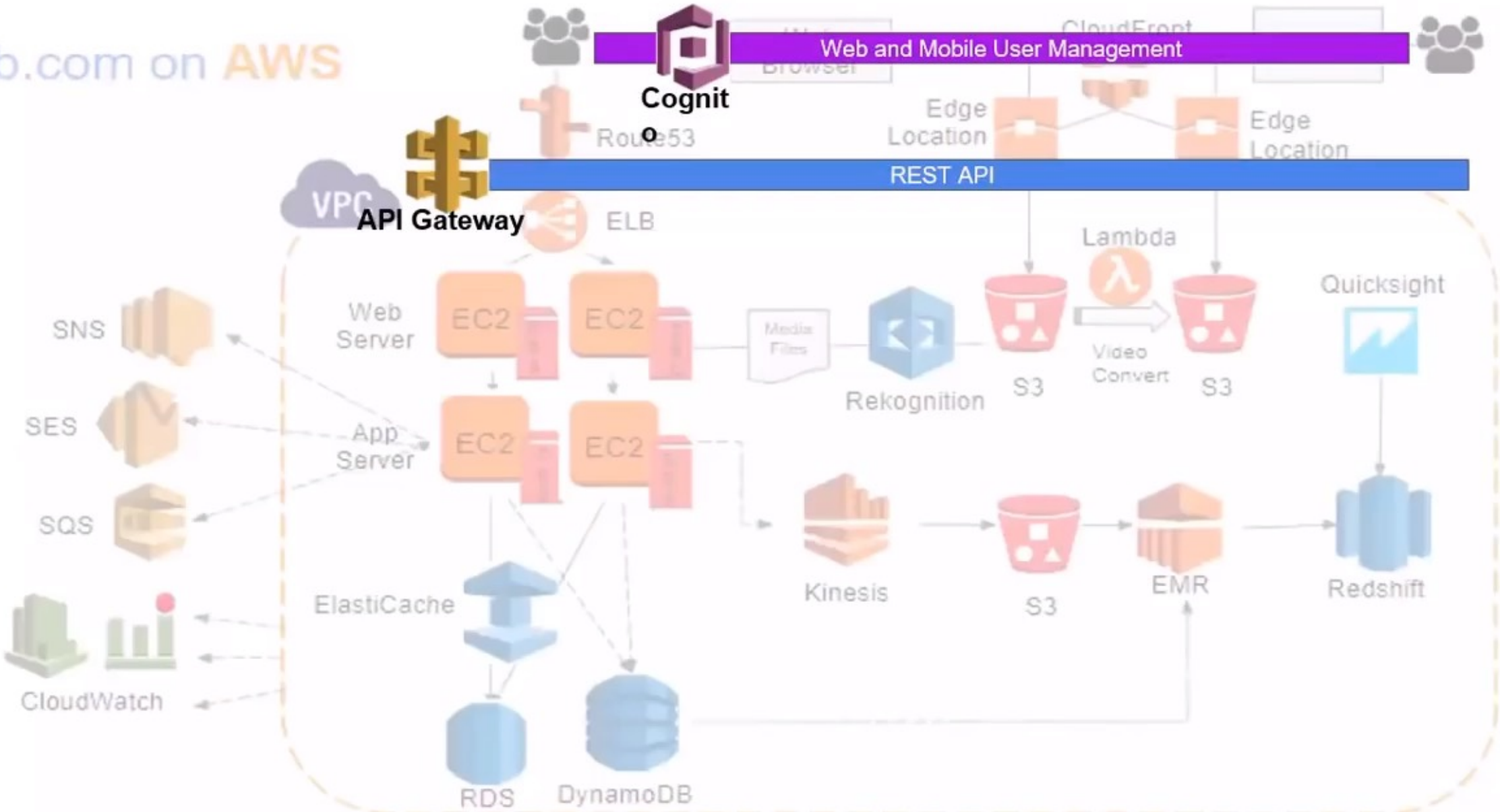

users
Web Browser
CloudFront
Mobile
fb.com
Route53
Edge Location
Edge Location
VPC
ELB
Lambda
Quicksight
Web Server
EC2
EBS
EC2
EBS
Media Files
Rekognition
S3
Video Convert
S3
SNS
SES
SQS
App Server
EC2
EBS
EC2
EBS
User clicks / Actions
Athena
Kinesis
S3
EMR
Redshift
ElastiCache
CloudWatch
RDS
DynamoDB
Glue

fb.com on AWS



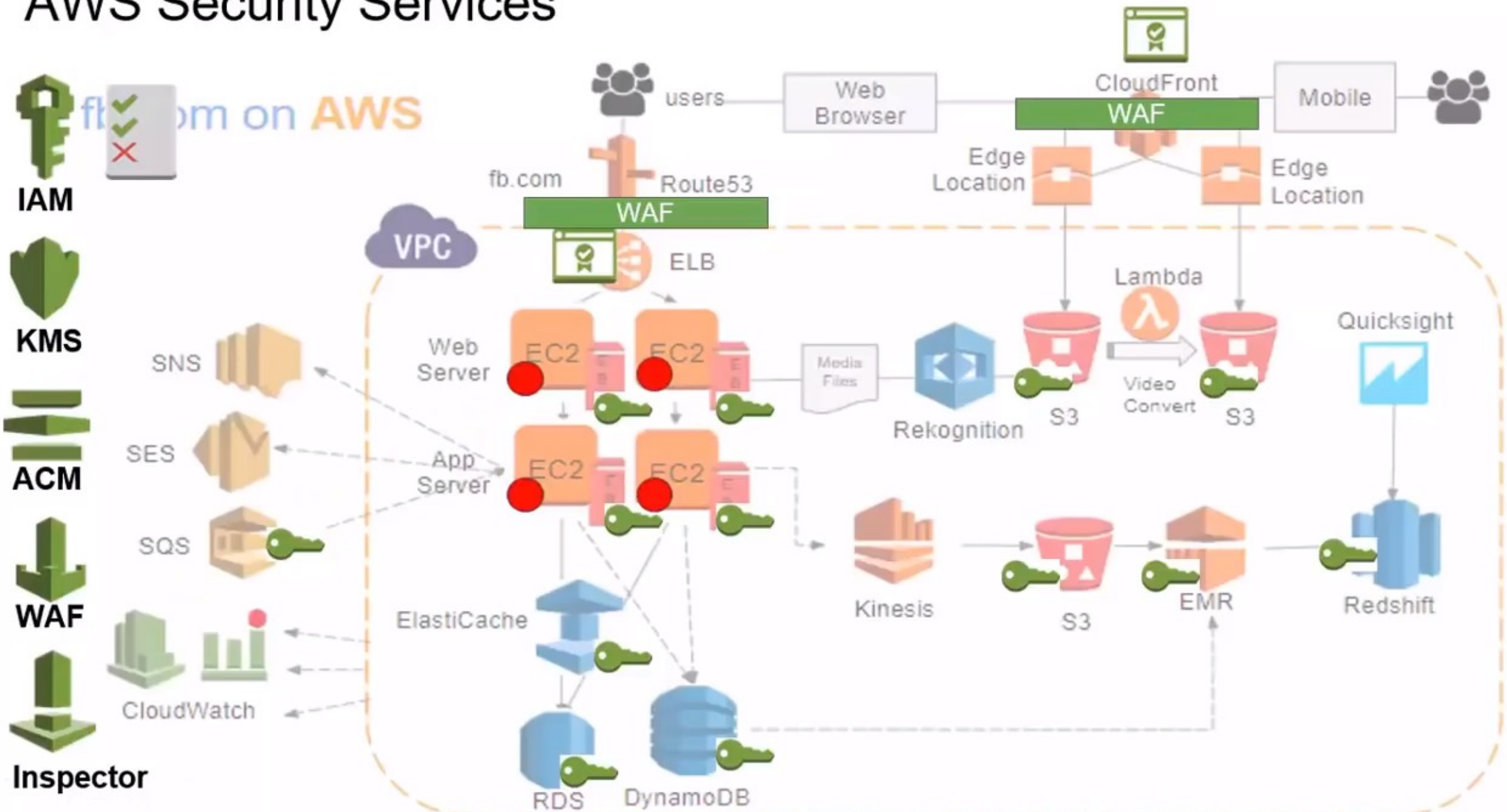
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

AWS Application Services



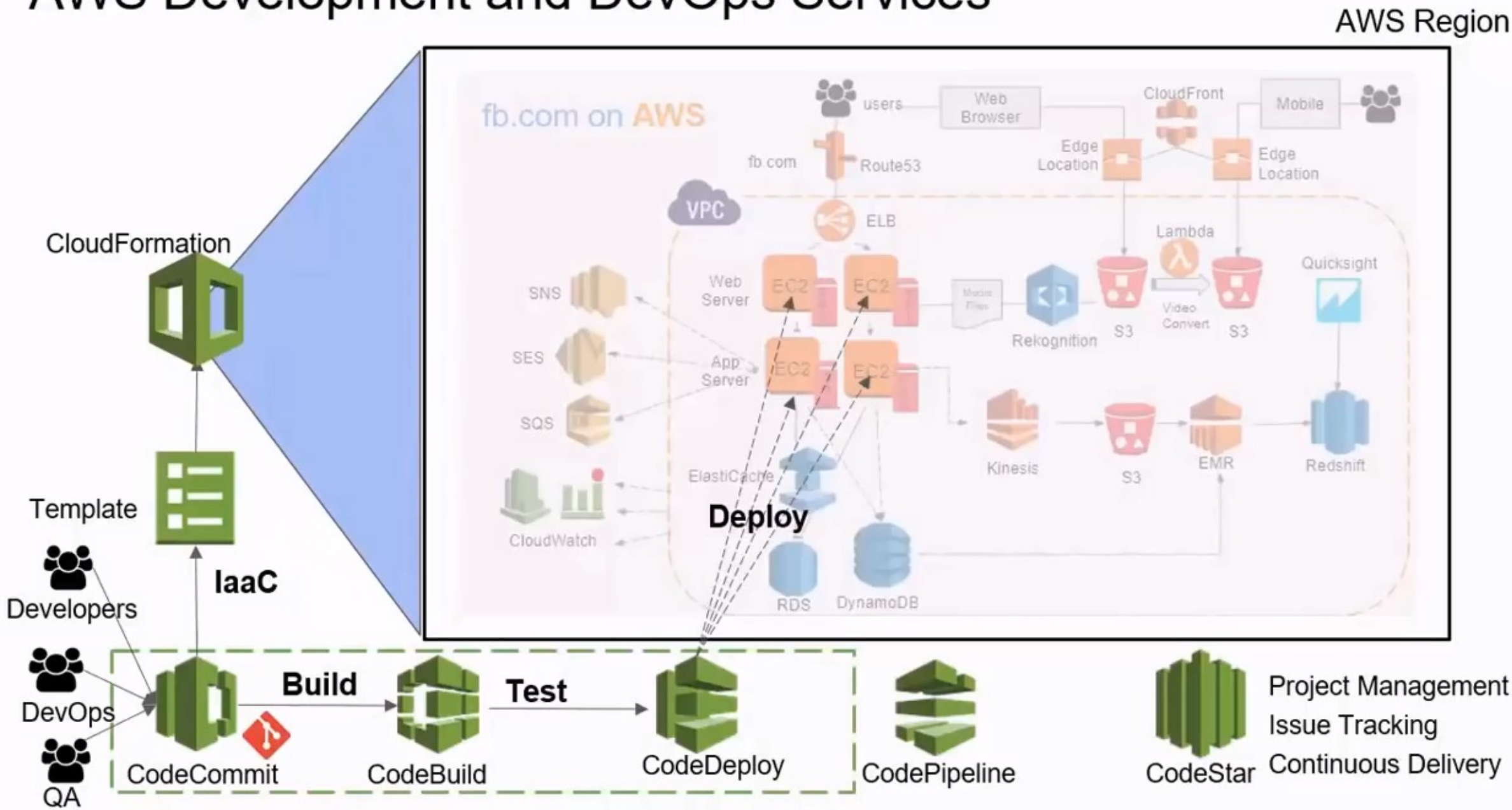
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

AWS Security Services



Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

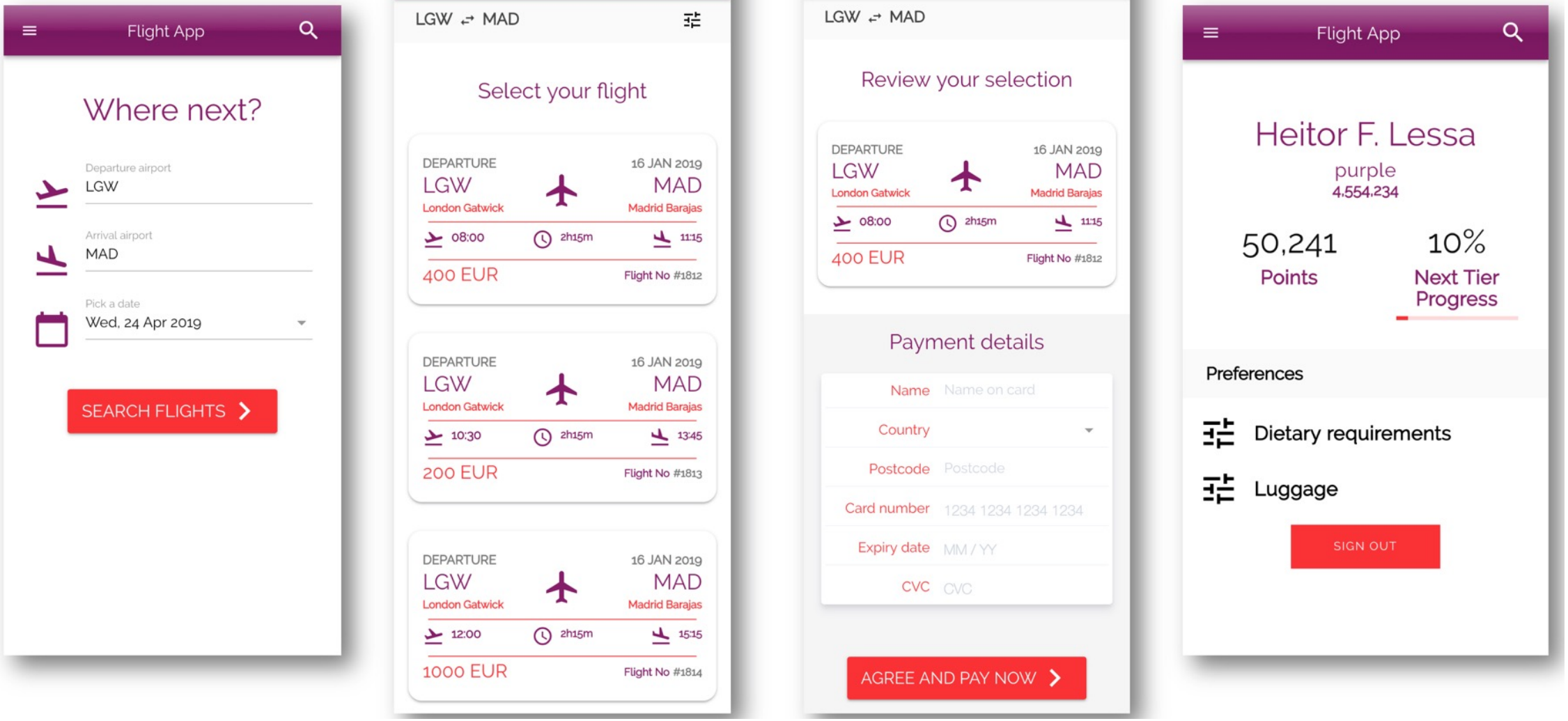
AWS Development and DevOps Services



Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

AWS Serverless Architecture

AWS Serverless Airline Booking



Source: https://github.com/aws-samples/aws-serverless-airline-booking

AWS Serverless Airline Booking Stack

UI/UX


Quasar framework


Vue.js


AWS Amplify

Stripe Elements

Data/Lang


Amazon DynamoDB


Python

Typescript

JavaScript

API/Auth


AWS AppSync


Amazon API Gateway


Amazon Cognito

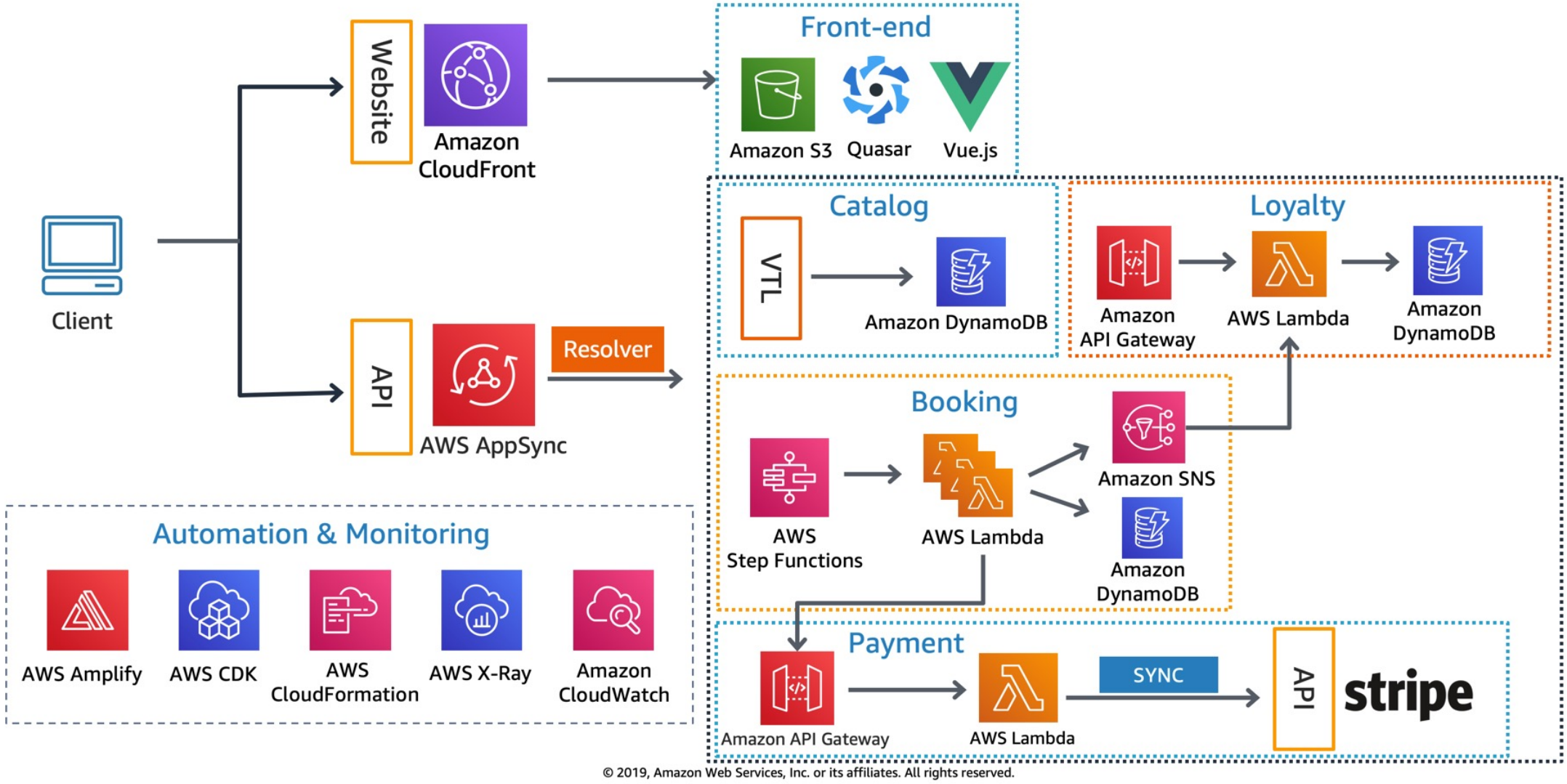
Messaging


Amazon SNS


AWS Step Functions

Source: https://github.com/aws-samples/aws-serverless-airline-booking

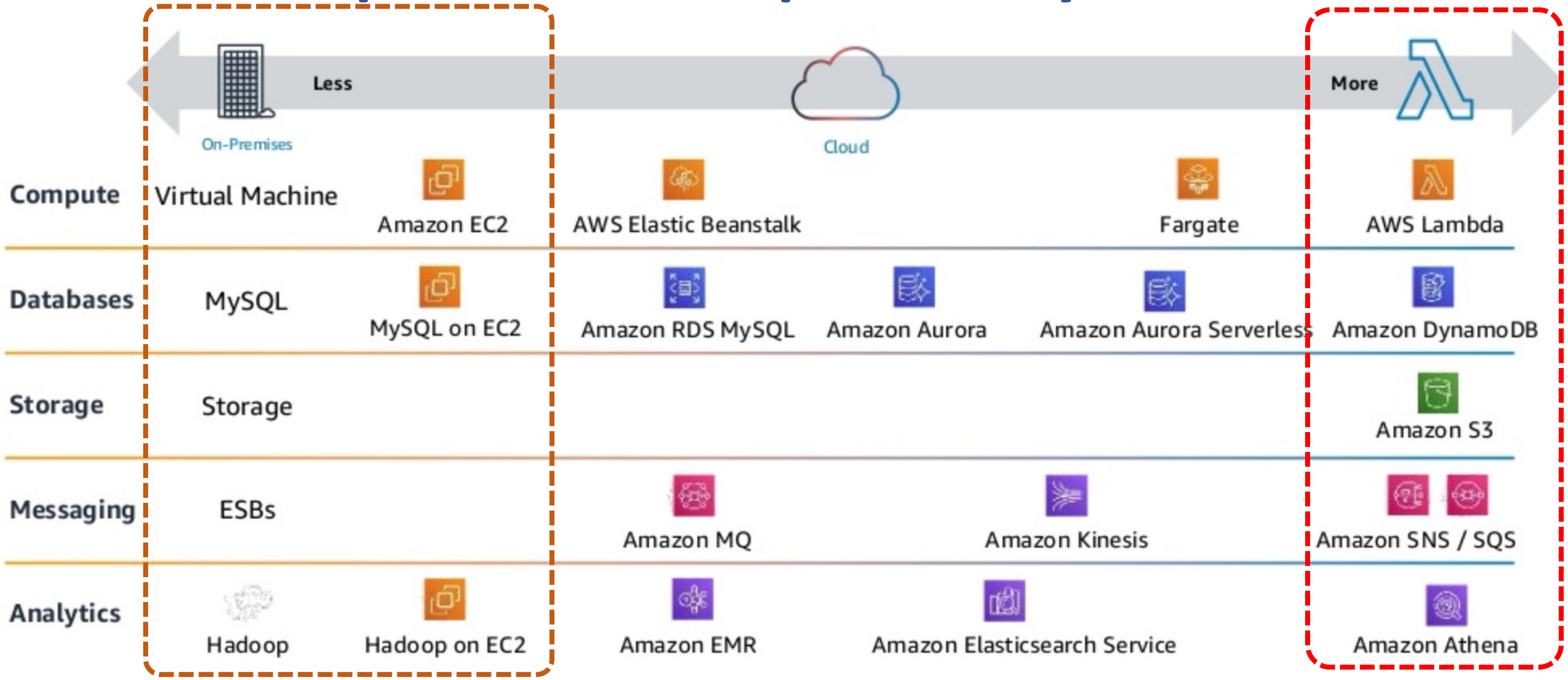
AWS Serverless Airline Booking
High level infrastructure architecture



Source: https://github.com/aws-samples/aws-serverless-airline-booking

AWS Serverless Architecture

AWS Operational Responsibility Models



Source: Heitor Lessa (2019), How to build a full stack serverless airline ticketing web app, https://www.youtube.com/watch?v=MyoOeHTp2pg

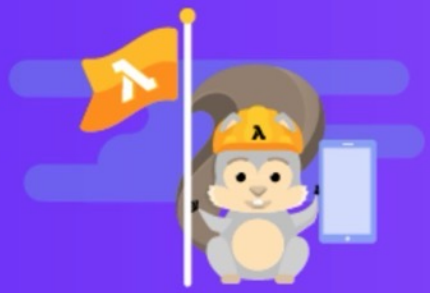
Build a Serverless Web Application

Build a Serverless Web Application

Projects on AWS:

Build a Serverless Web Application

with AWS Lambda, Amazon API Gateway, Amazon S3, Amazon DynamoDB, and Amazon Cognito



Introduction | 1 Host a static website | 2 Manage users | 3 Build a serverless backend | 4 Deploy a RESTful API | 5 Terminate resources

Overview

In this tutorial, you'll create a simple serverless web application that enables users to request unicorn rides from the Wild Rydes fleet. The application will present users with an HTML based user interface for indicating the location where they would like to be picked up and will interface on the backend with a RESTful web service to submit the request and dispatch a nearby unicorn. The application will also provide facilities for users to register with the service and log in before requesting rides.

AWS Experience: Beginner

Time to complete: 2 hours

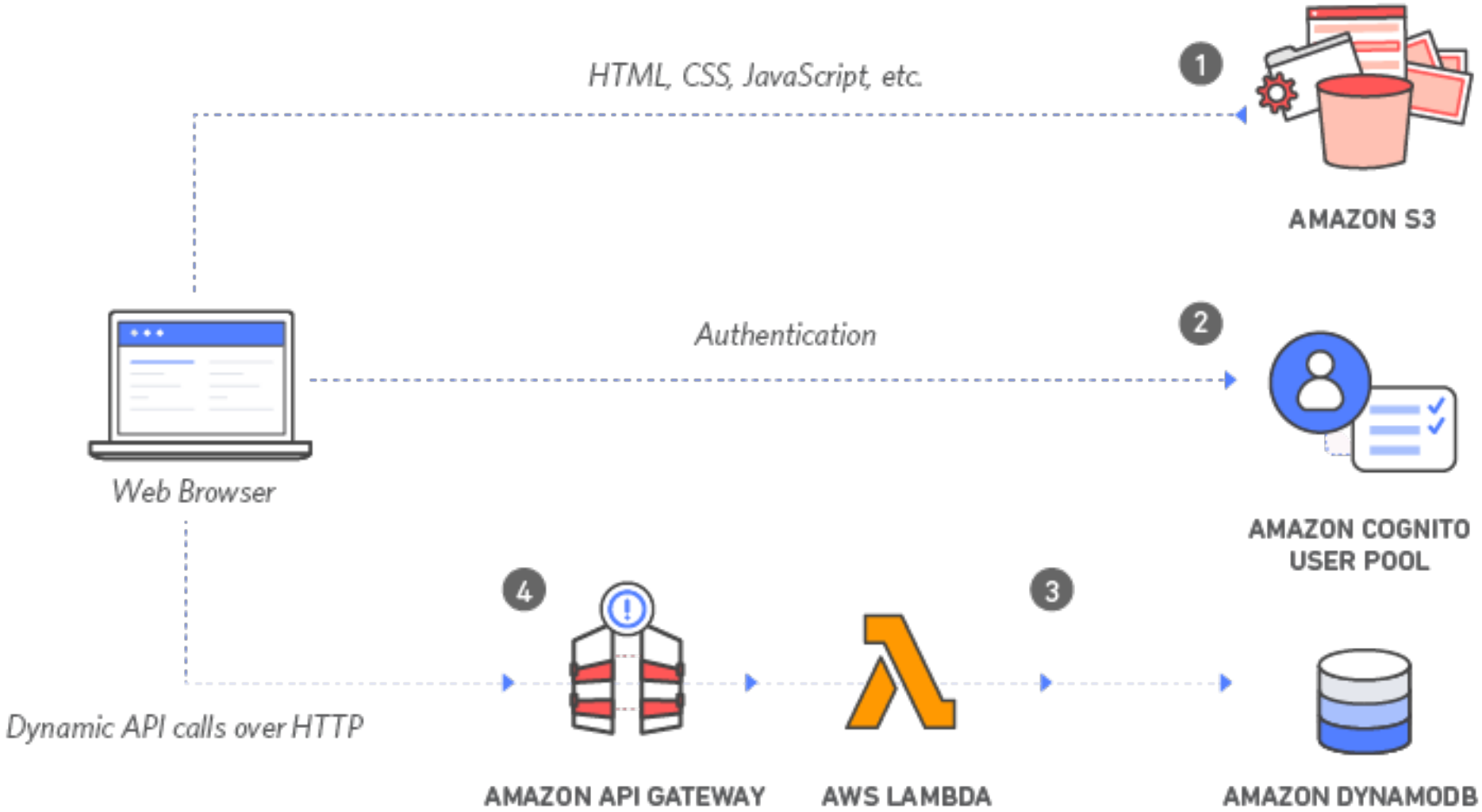
Cost to complete: Each service used in this architecture is eligible for the AWS Free Tier. If you are outside the usage limits of the Free Tier, completing this tutorial will cost you less than $0.25*.

Application Architecture

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

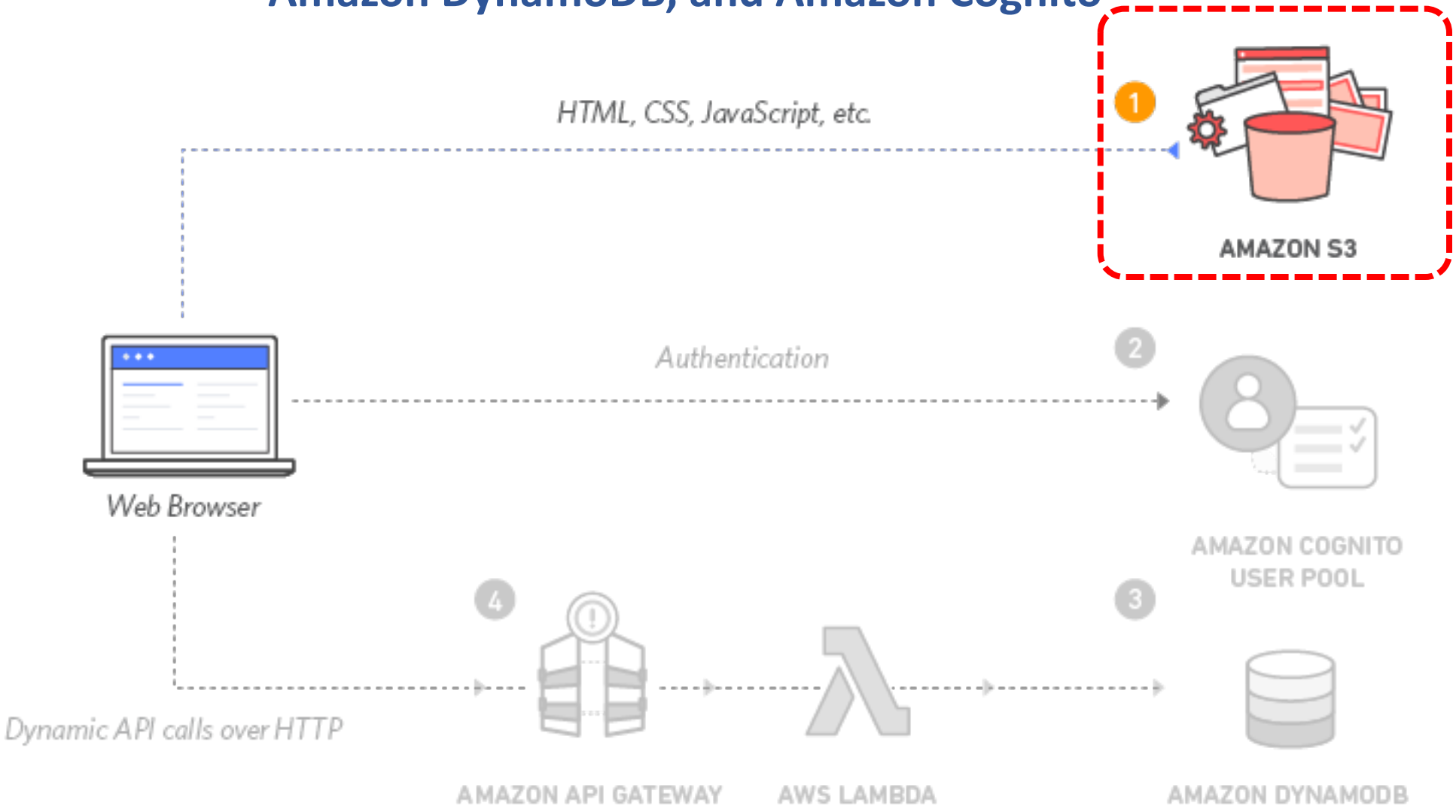


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

1



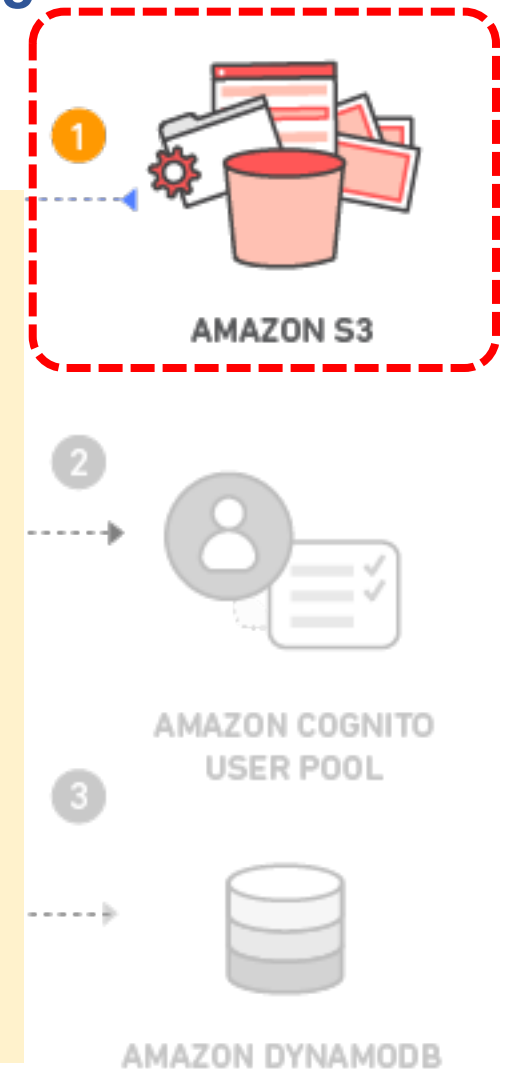
Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

1

HTML, CSS, JavaScript, etc.



Static Web Hosting

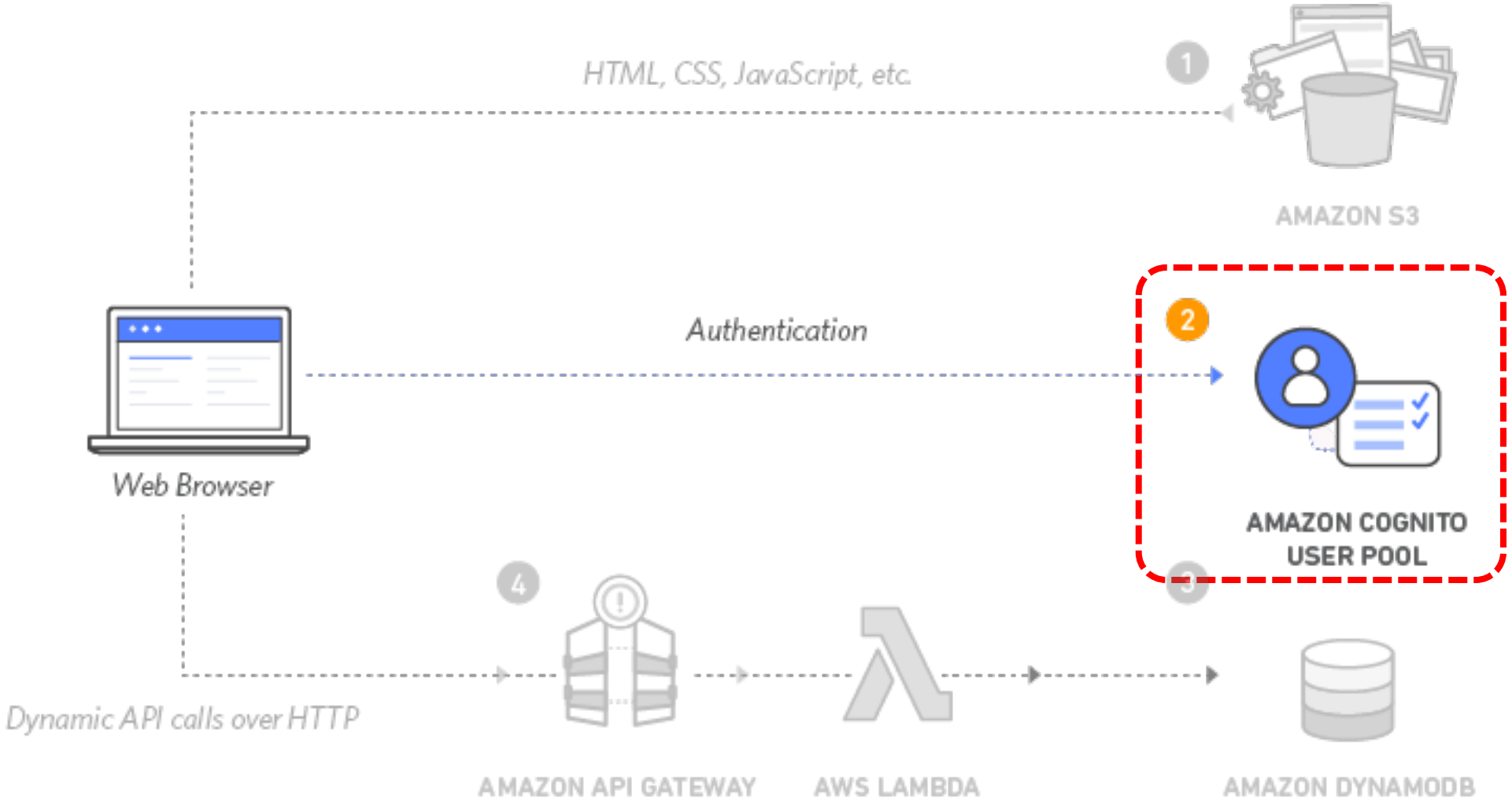
Amazon S3 hosts static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser.

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

2



Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

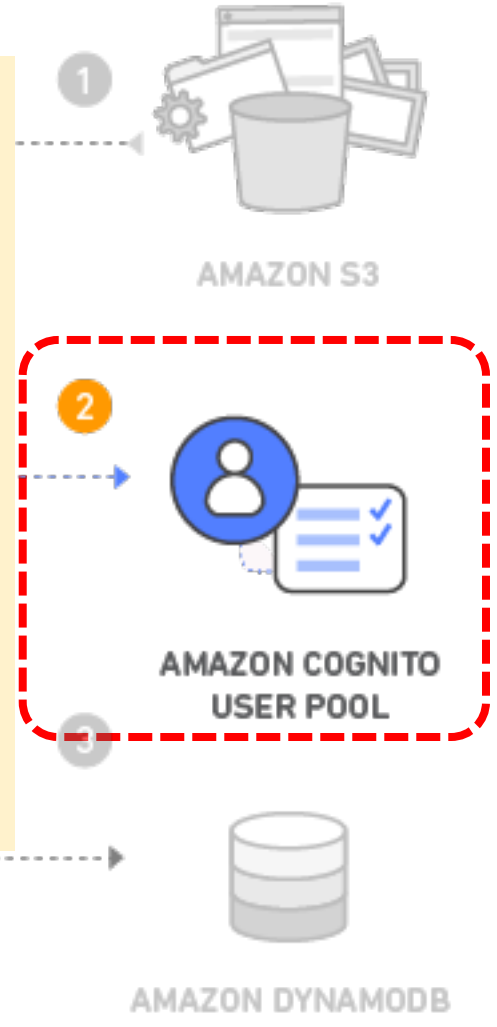
Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

2

User Management

Amazon Cognito provides user management and authentication functions to secure the backend API.

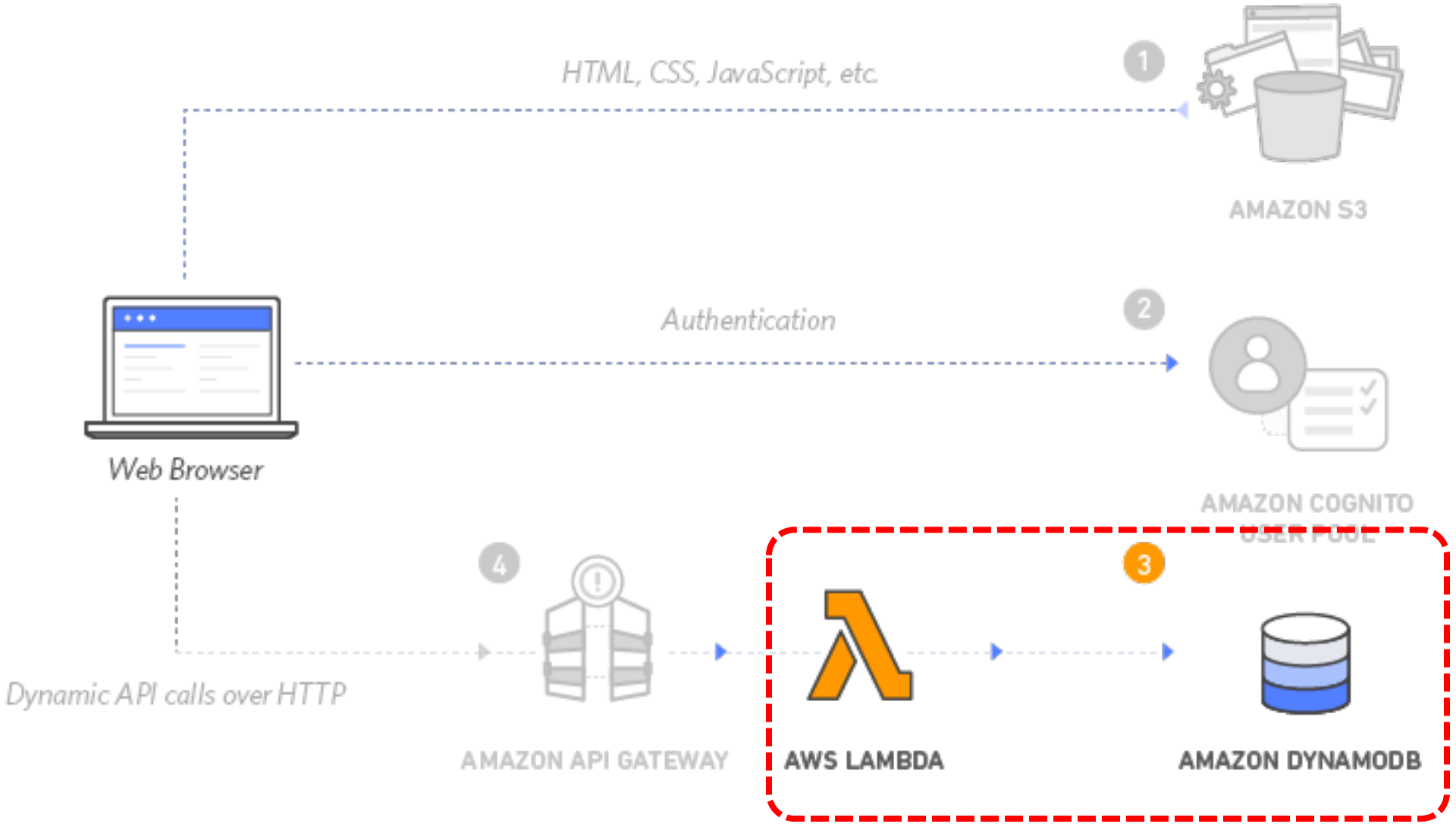


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

3



Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

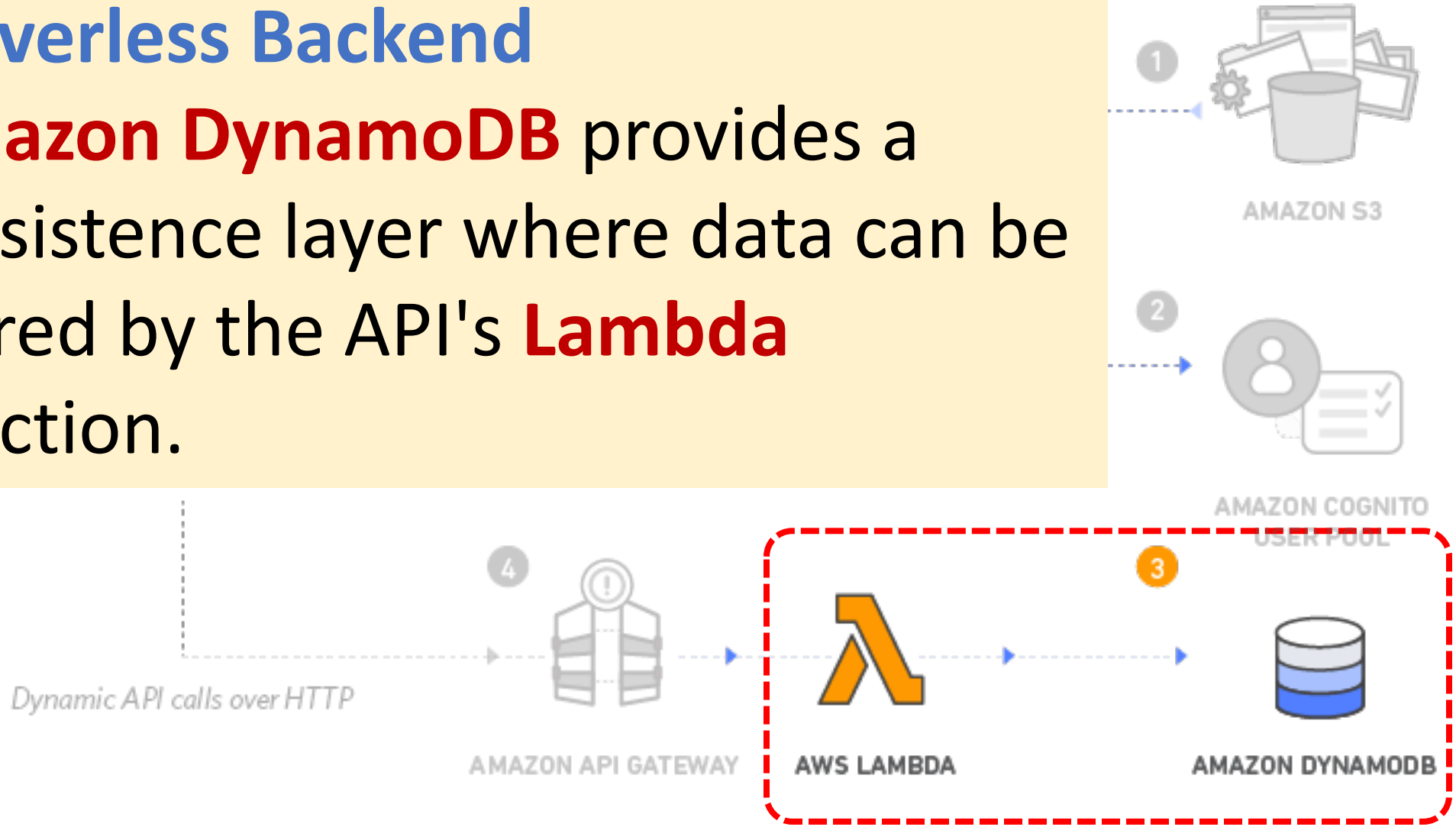
Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

3

Serverless Backend

Amazon DynamoDB provides a persistence layer where data can be stored by the API's **Lambda** function.

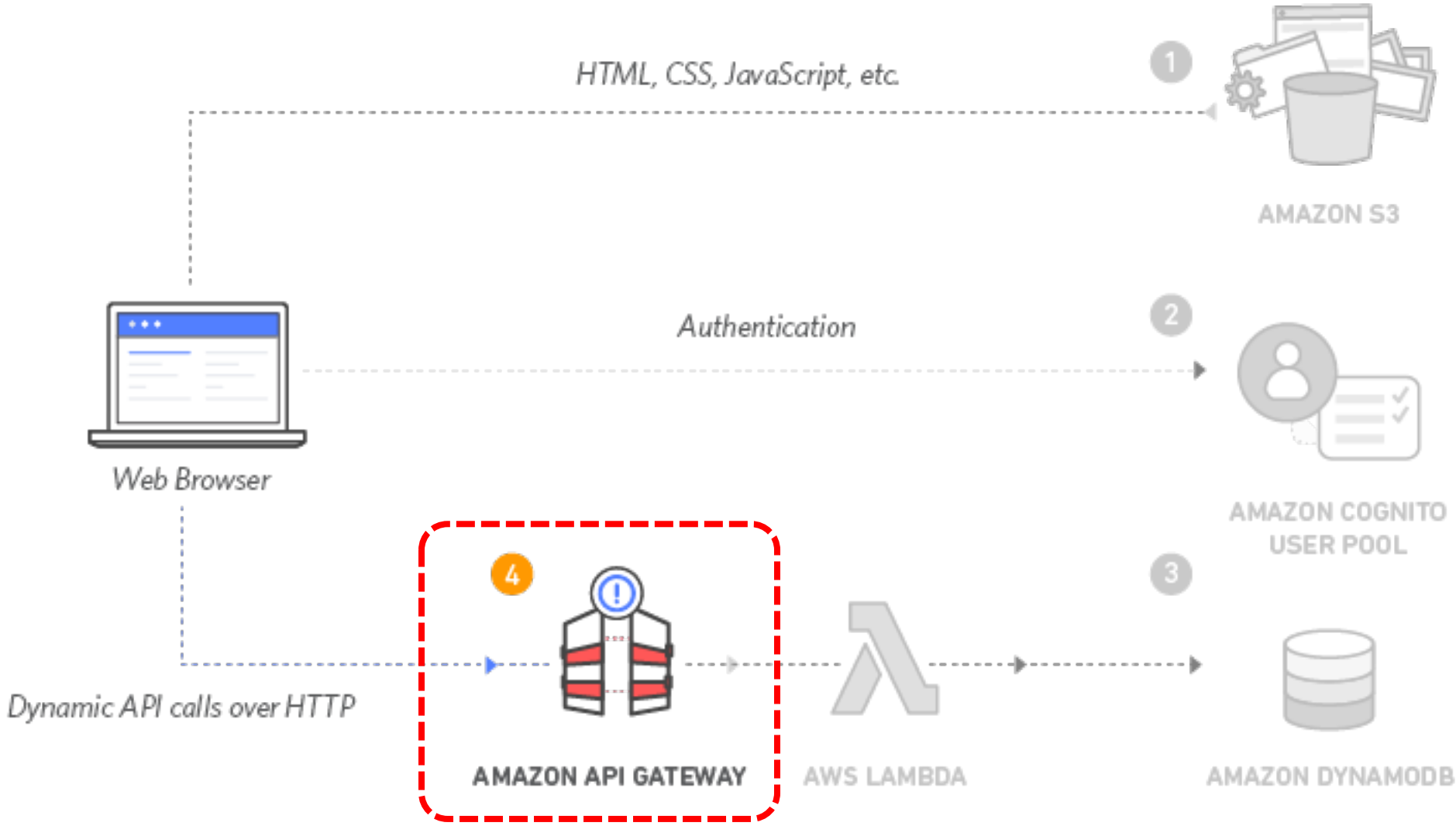


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

4



Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

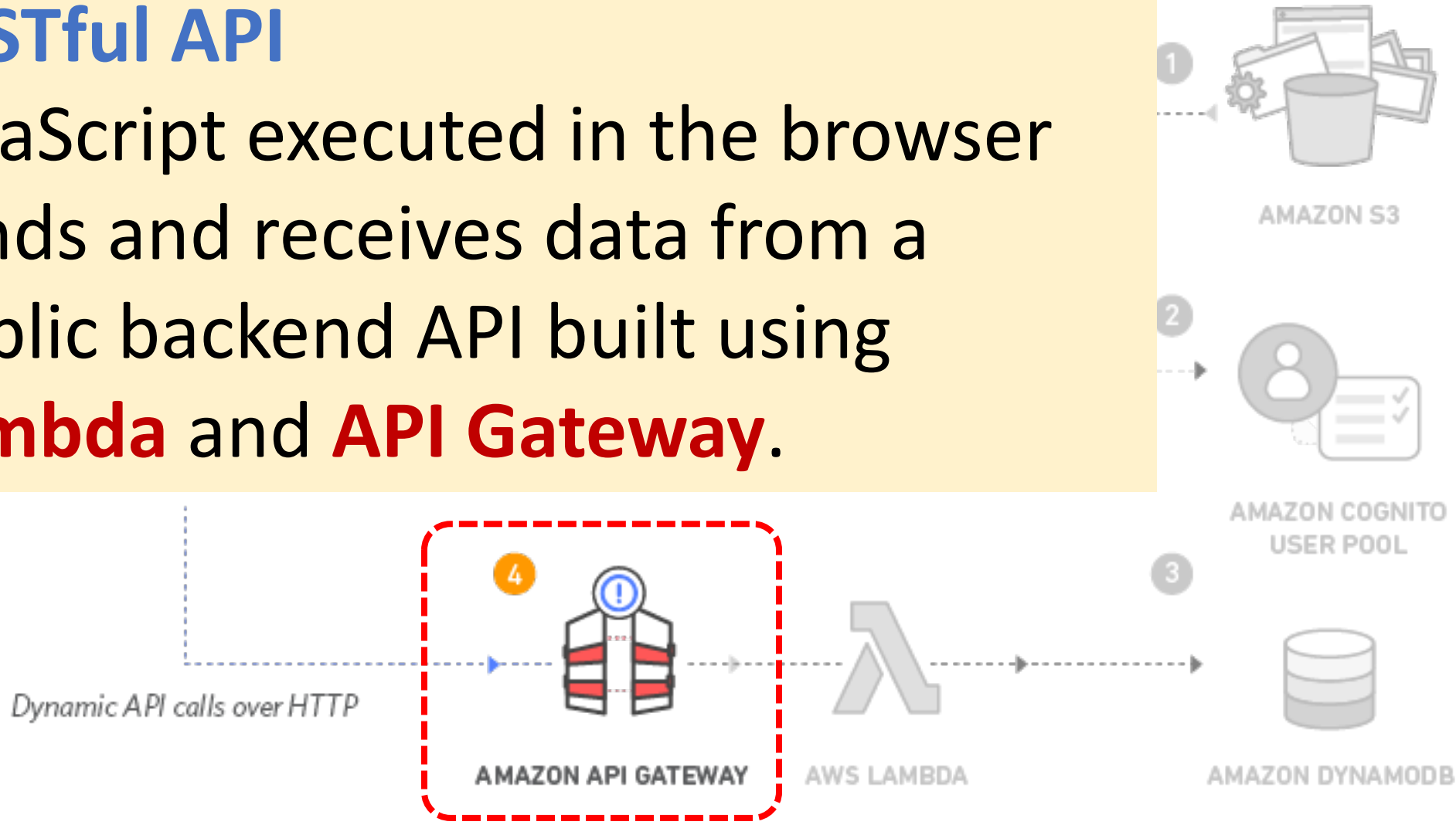
Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

4

RESTful API

JavaScript executed in the browser sends and receives data from a public backend API built using **Lambda** and **API Gateway**.



Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

5

Terminate resources

Resource Cleanup

You will terminate an **Amazon S3** bucket, an **Amazon Cognito** User Pool, an **AWS Lambda** function, an **IAM** role, a **DynamoDB** table, a **REST API**, and a **CloudWatch** Log.

It is a best practice to delete resources you are no longer using to avoid unwanted charges.

Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

References

- Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.
- Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.
- Titus Winters, Tom Manshreck, and Hyrum Wright (2020), Software Engineering at Google: Lessons Learned from Programming Over Time, O'Reilly Media.
- Project Management Institute (2021), A Guide to the Project Management Body of Knowledge (PMBOK Guide) – Seventh Edition and The Standard for Project Management, PMI.
- Project Management Institute (2017), A Guide to the Project Management Body of Knowledge (PMBOK Guide), Sixth Edition, Project Management Institute.
- Project Management Institute (2017), Agile Practice Guide, Project Management Institute.
- Nesreen Otoum and Nuha Elkhalili.(2026) "Methods and Techniques of Agentic Software Engineering: A Systematic Literature Review." IEEE Access 14 (2026): 7443-7465.
- Lakshana Iruni Assalaarachchi, Zainab Masood, Rashina Hoda, and John Grundy. (2026) "Toward Agentic Software Project Management: A Vision and Roadmap." arXiv preprint arXiv:2601.16392 (2026).
- NVIDIA DLI (2026), Building RAG Agents with LLMs, https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-15+V1
- NVIDIA DLI (2026), Generative AI with Diffusion Models, https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-14+V1
- NVIDIA DLI (2026), Building Agentic AI Applications with LLMs, https://learn.nvidia.com/courses/course-detail?course_id=course-v1:DLI+S-FX-41+V1