軟體工程
(Software Engineering)
雲端運算與雲軟體架構
(Cloud Computing and Cloud Software Architecture)

1101SE07
MBA, IM, NTPU (M6131) (Fall 2021)
Thu 11, 12, 13 (19:25-22:10) (209)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021/11/18

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/09/23	軟體工程概論 (Introduction to Software Engineering)
2	2021/09/30	軟體產品與專案管理：軟體產品管理，原型設計 (Software Products and Project Management: Software product management and prototyping)
3	2021/10/07	敏捷軟體工程：敏捷方法、Scrum、極限程式設計 (Agile Software Engineering: Agile methods, Scrum, and Extreme Programming)
4	2021/10/14	功能、場景和故事 (Features, Scenarios, and Stories)
5	2021/10/21	軟體工程個案研究 I (Case Study on Software Engineering I)
6	2021/10/28	軟體架構：架構設計、系統分解、分散式架構 (Software Architecture: Architectural design, System decomposition, and Distribution architecture)

課程大綱 (Syllabus)

國立臺北大學 National Taipei University

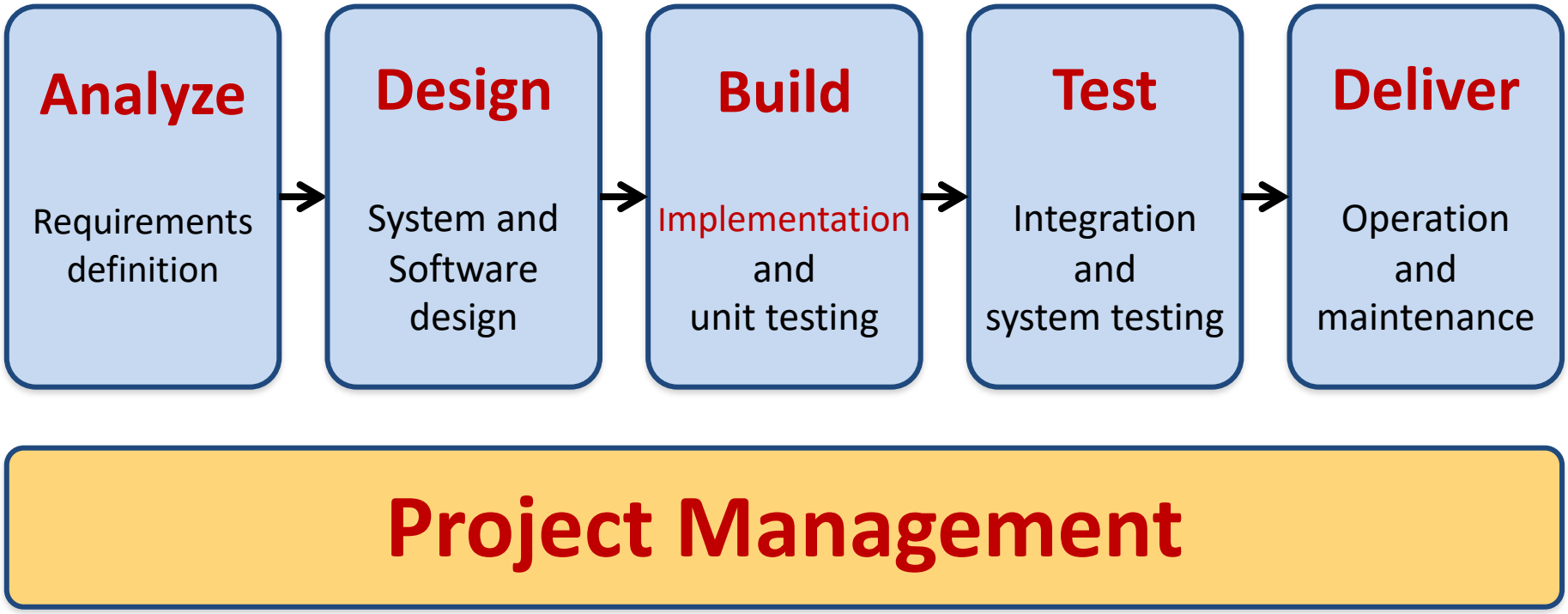
週次 (Week)	日期 (Date)	內容 (Subject/Topics)
7	2021/11/04	基於雲的軟體：虛擬化和容器、軟體即服務 (Cloud-Based Software: Virtualization and containers, Everything as a service, Software as a service)
8	2021/11/11	期中報告 (Midterm Project Report)
9	2021/11/18	雲端運算與雲軟體架構 (Cloud Computing and Cloud Software Architecture)
10	2021/11/25	微服務架構：RESTful服務、服務部署 (Microservices Architecture, RESTful services, Service deployment)
11	2021/12/02	軟體工程產業實務 (Industry Practices of Software Engineering)
12	2021/12/09	軟體工程個案研究 II (Case Study on Software Engineering II)

課程大綱 (Syllabus)

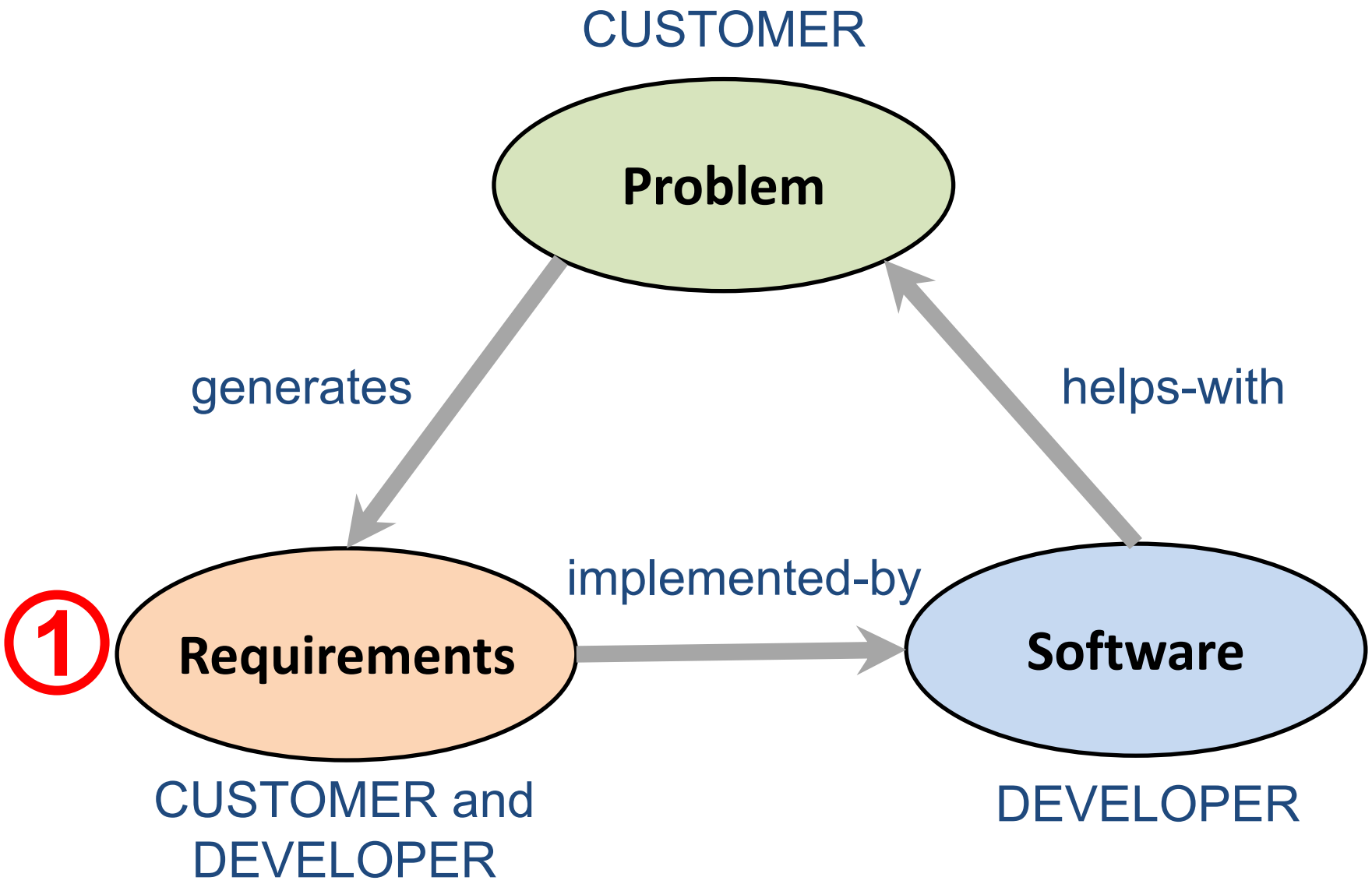
國立臺北大學 National Taipei University

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
13	2021/12/16	安全和隱私 (Security and Privacy); 可靠的程式設計 (Reliable Programming)
14	2021/12/23	測試：功能測試、測試自動化、 測試驅動的開發、程式碼審查 (Testing: Functional testing, Test automation, Test-driven development, and Code reviews); DevOps和程式碼管理：程式碼管理和DevOps自動化 (DevOps and Code Management: Code management and DevOps automation)
15	2021/12/30	期末報告 I (Final Project Report I)
16	2022/01/06	期末報告 II (Final Project Report II)
17	2022/01/13	學生自主學習 (Self-learning)
18	2022/01/20	學生自主學習 (Self-learning)

Software Engineering and Project Management

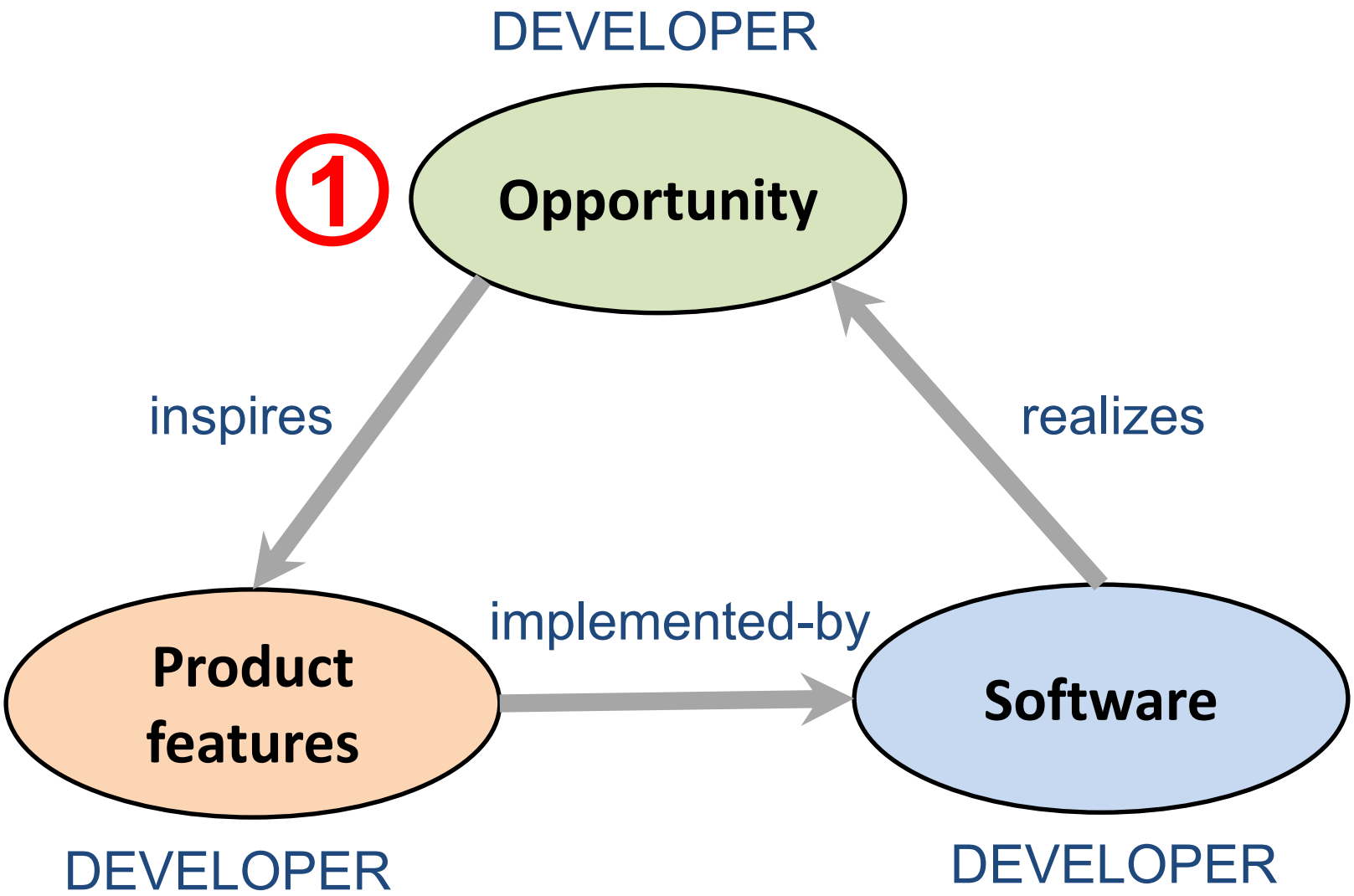


Project-based software engineering



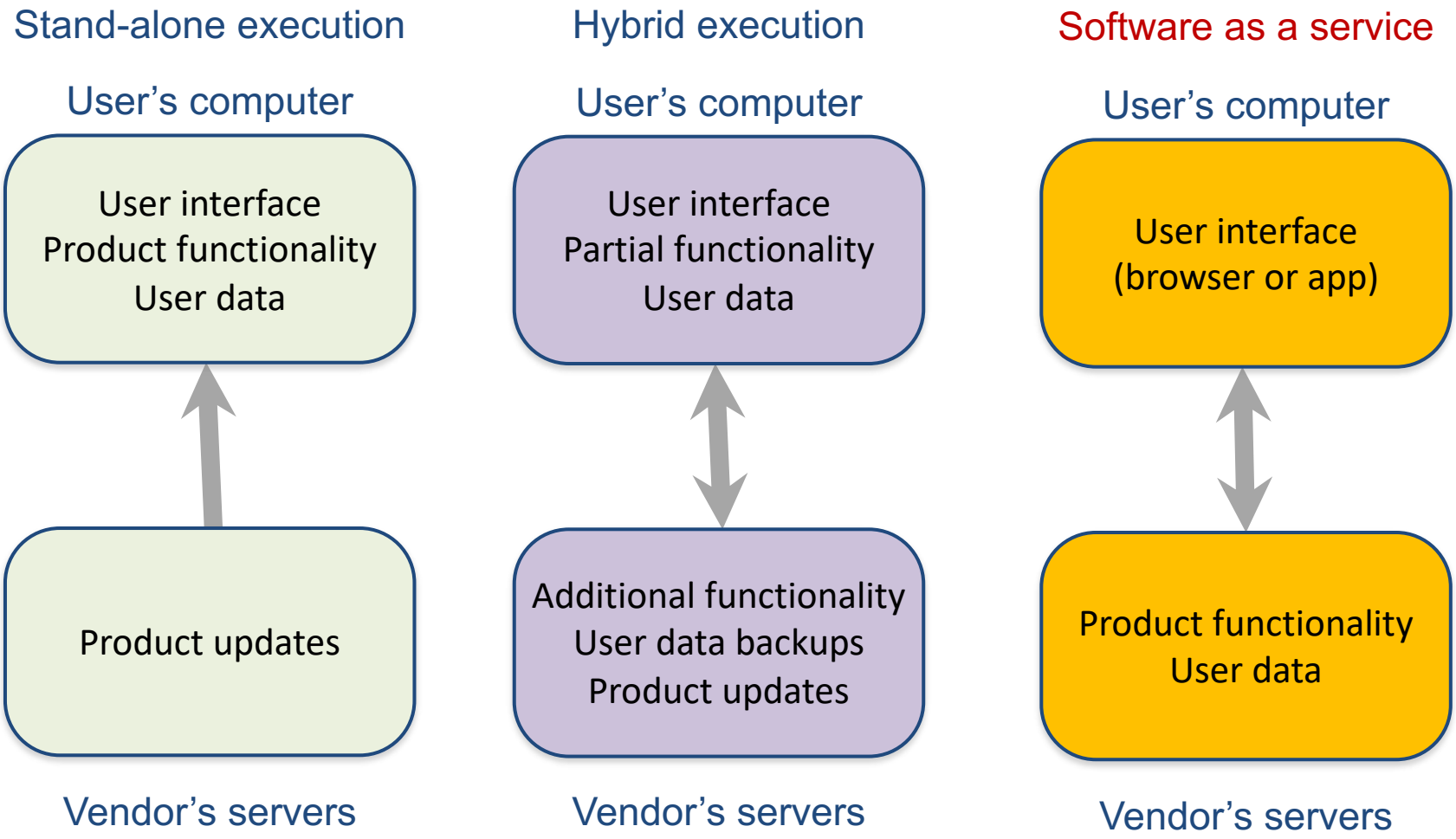
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product software engineering



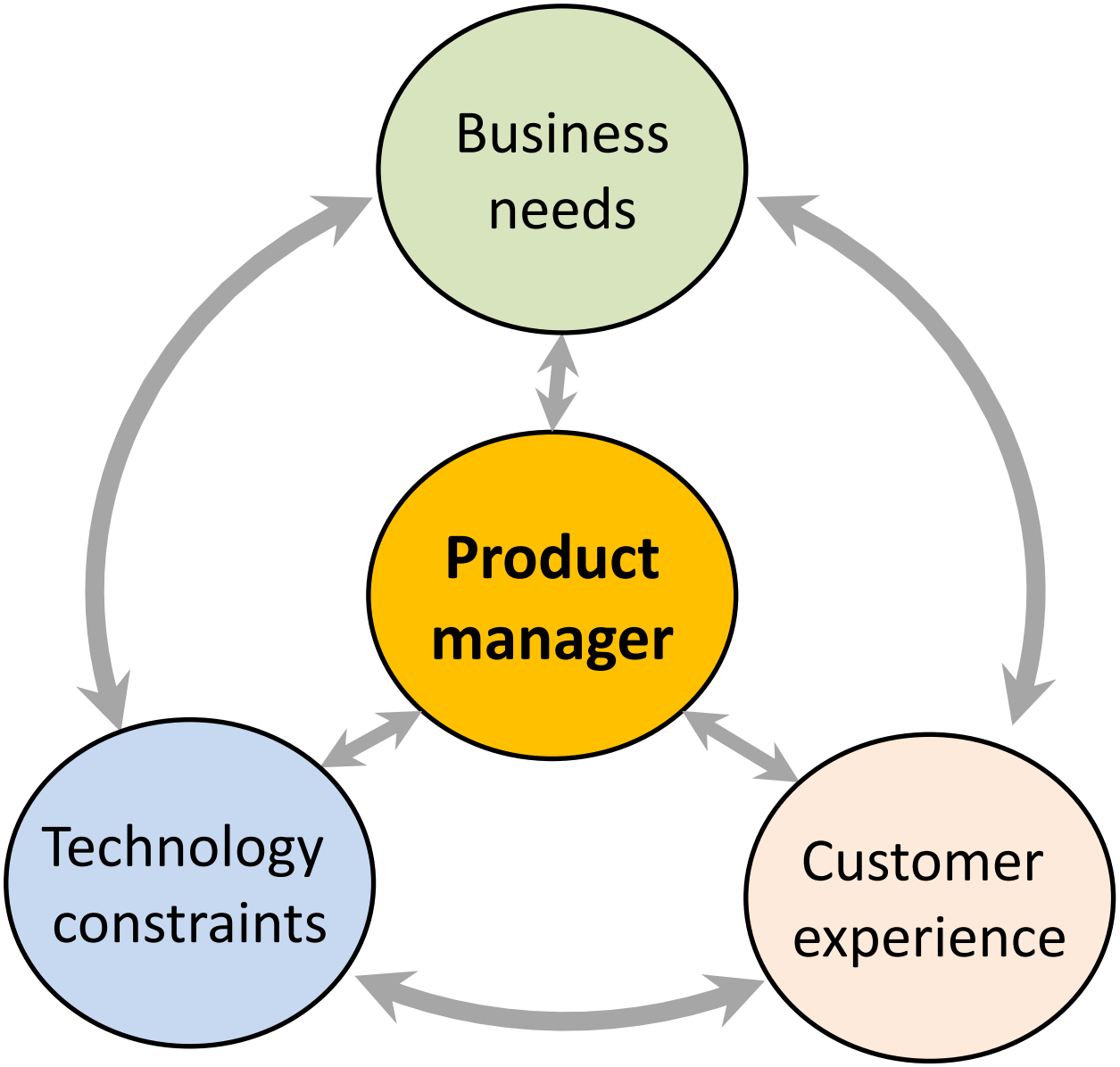
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software execution models



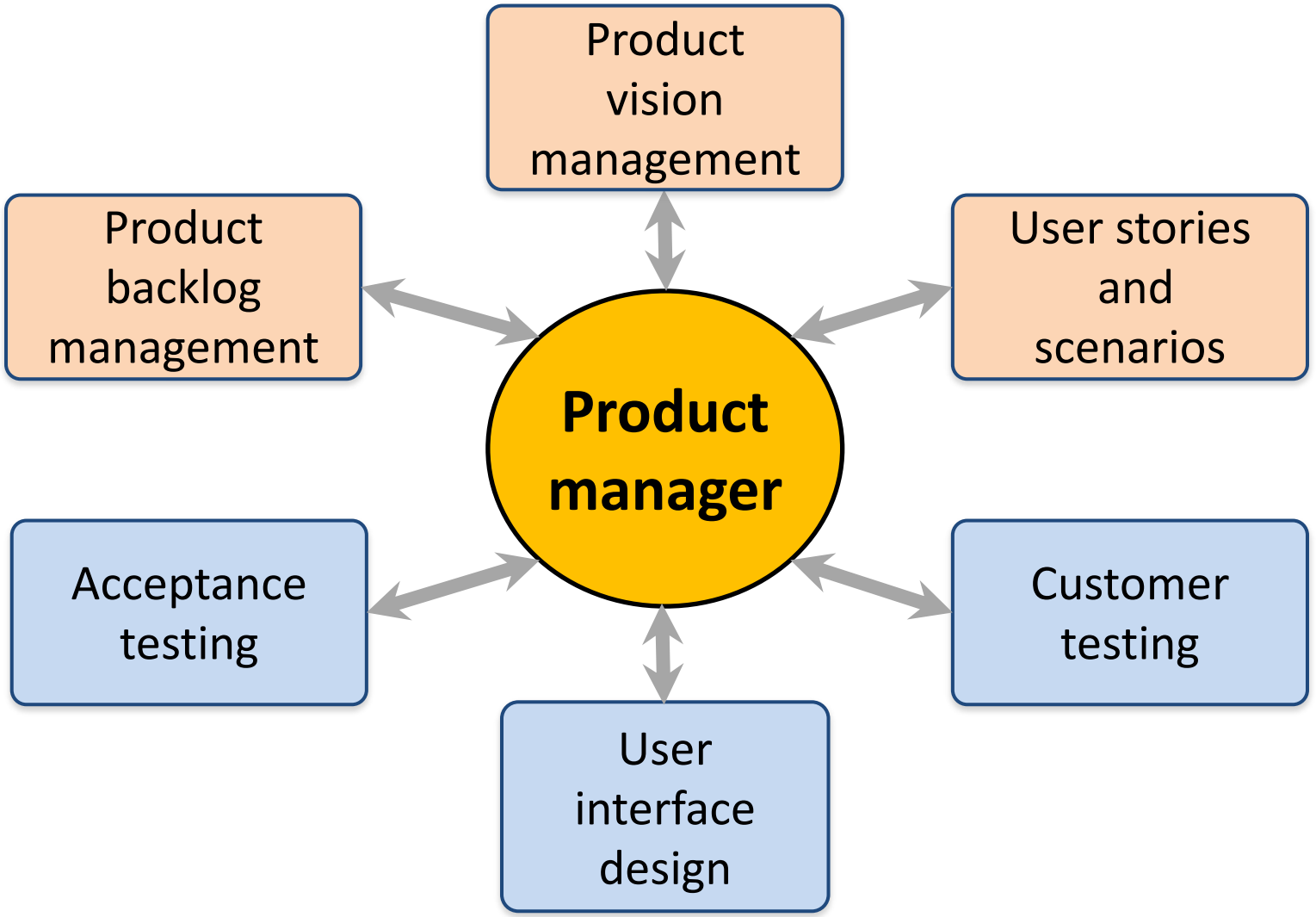
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Product management concerns



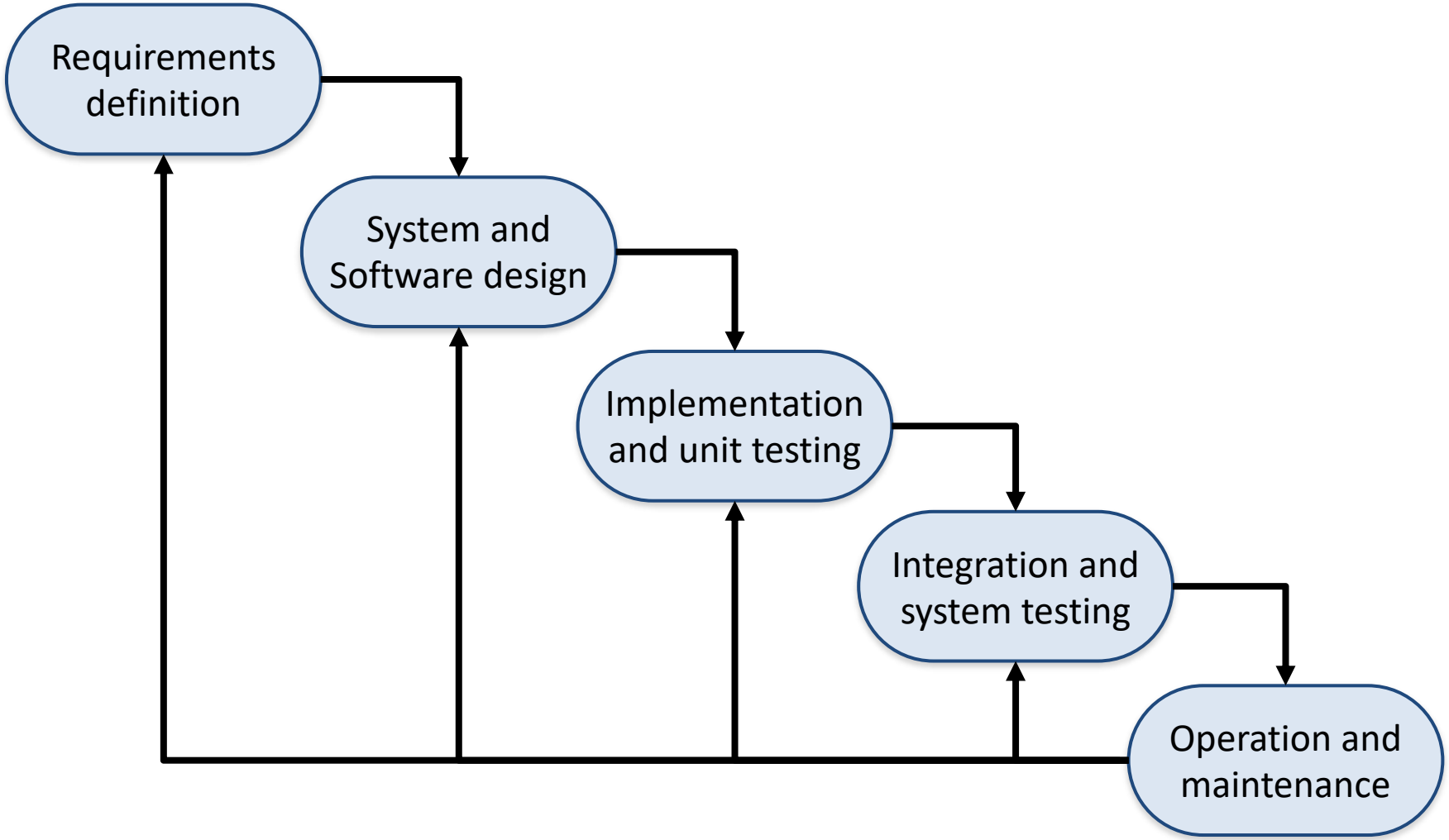
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Technical interactions of product managers



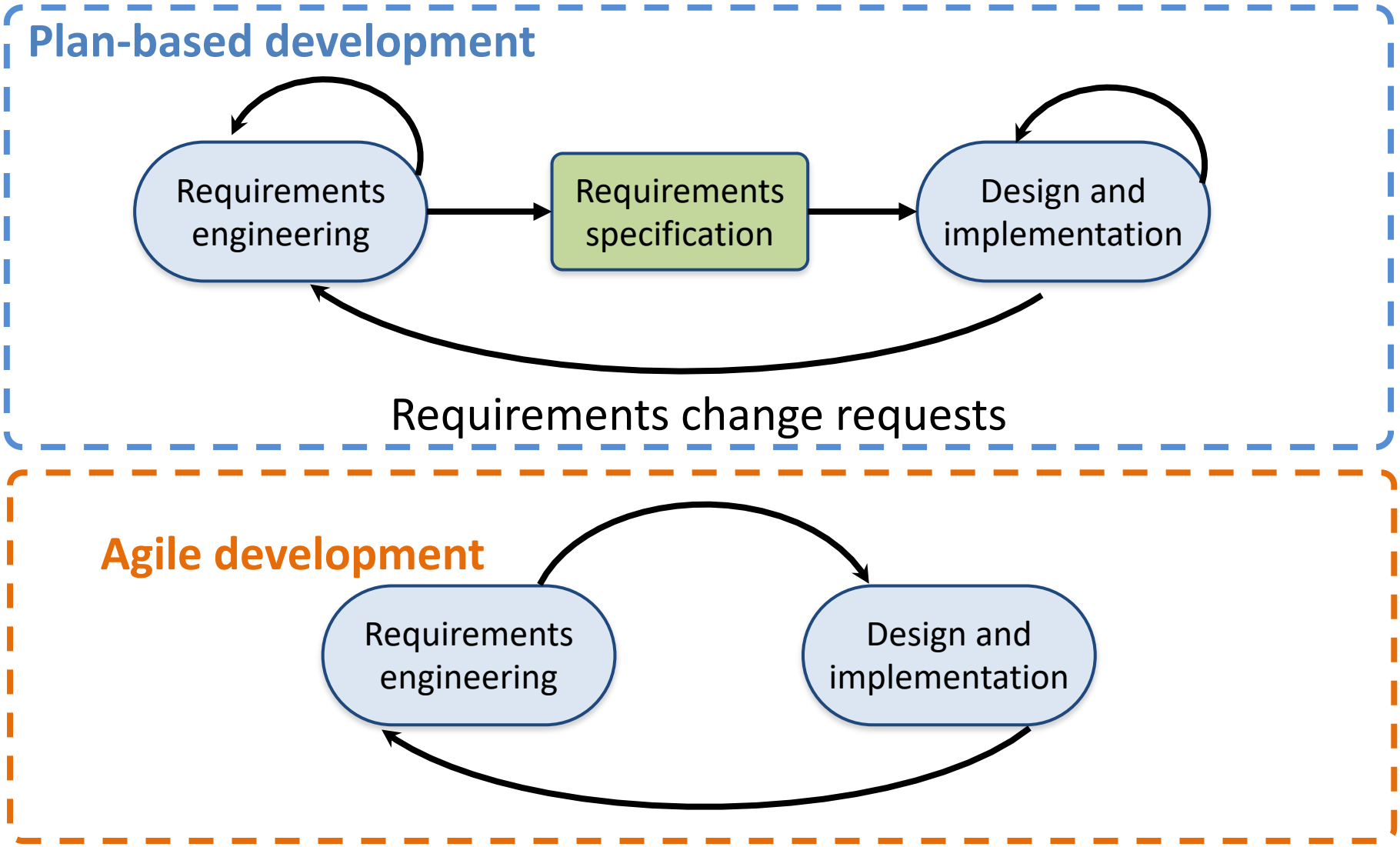
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software Development Life Cycle (SDLC)
The waterfall model



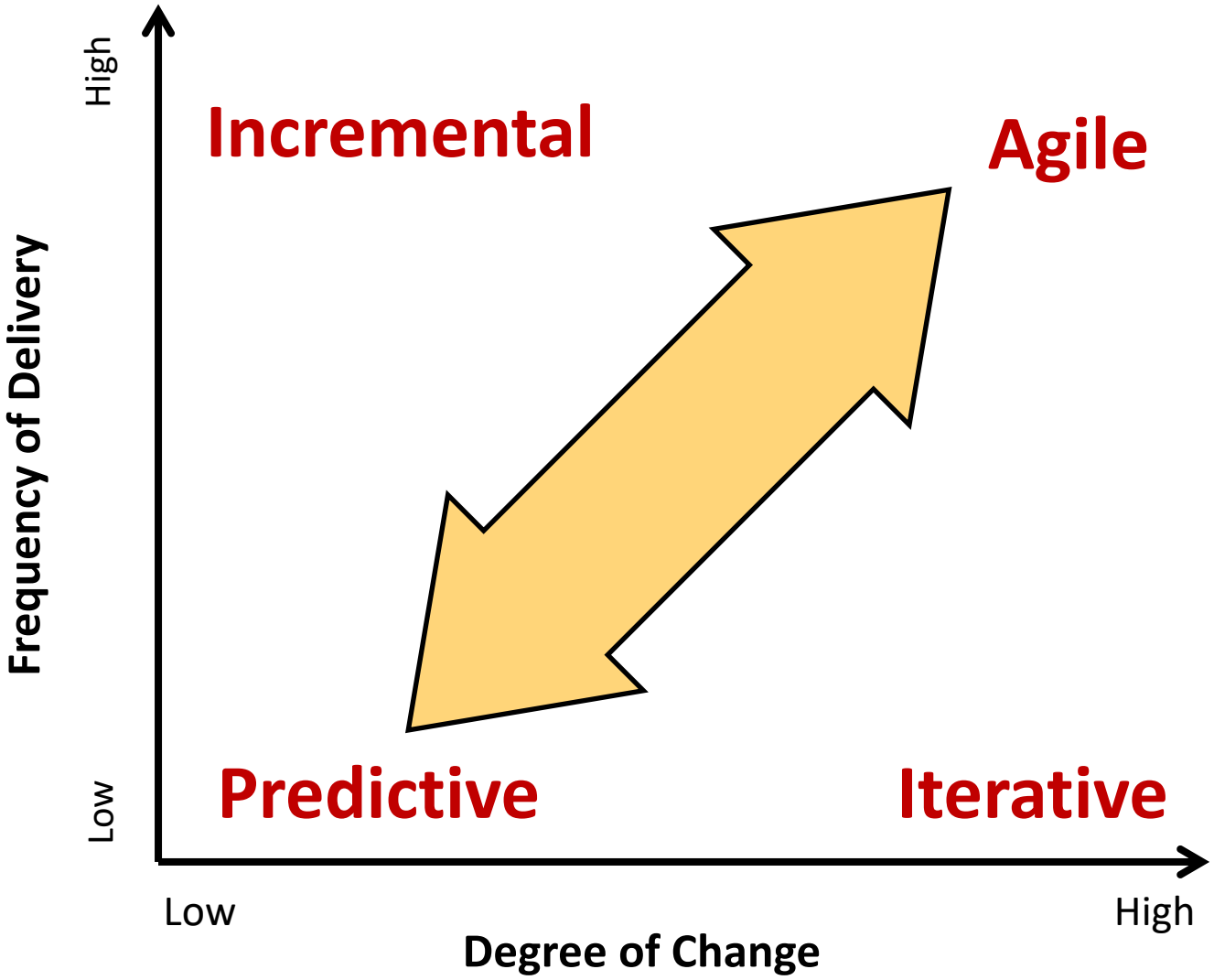
Source: Ian Sommerville (2015), Software Engineering, 10th Edition, Pearson.

Plan-based and Agile development



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

The Continuum of Life Cycles

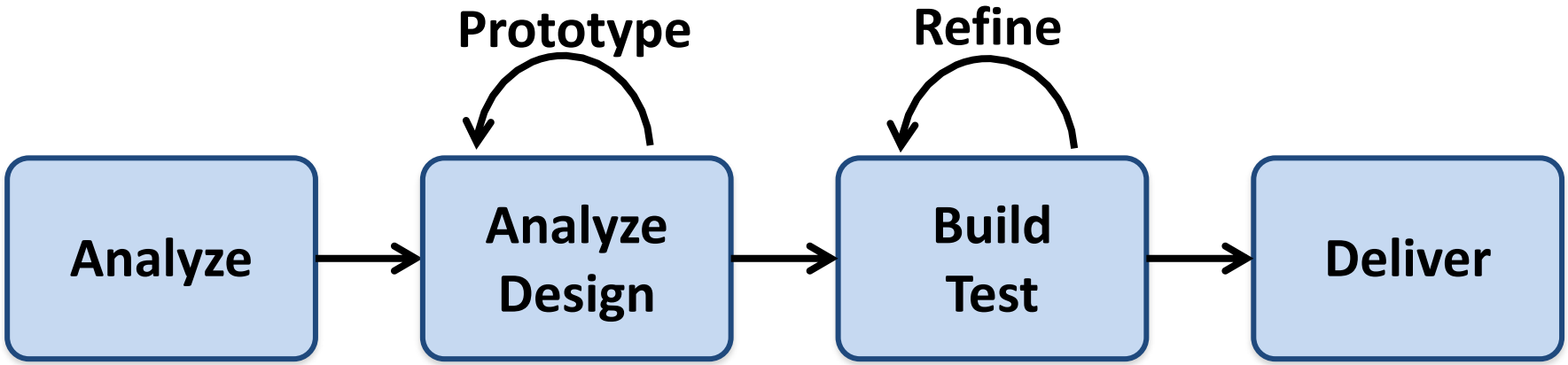


Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Predictive Life Cycle

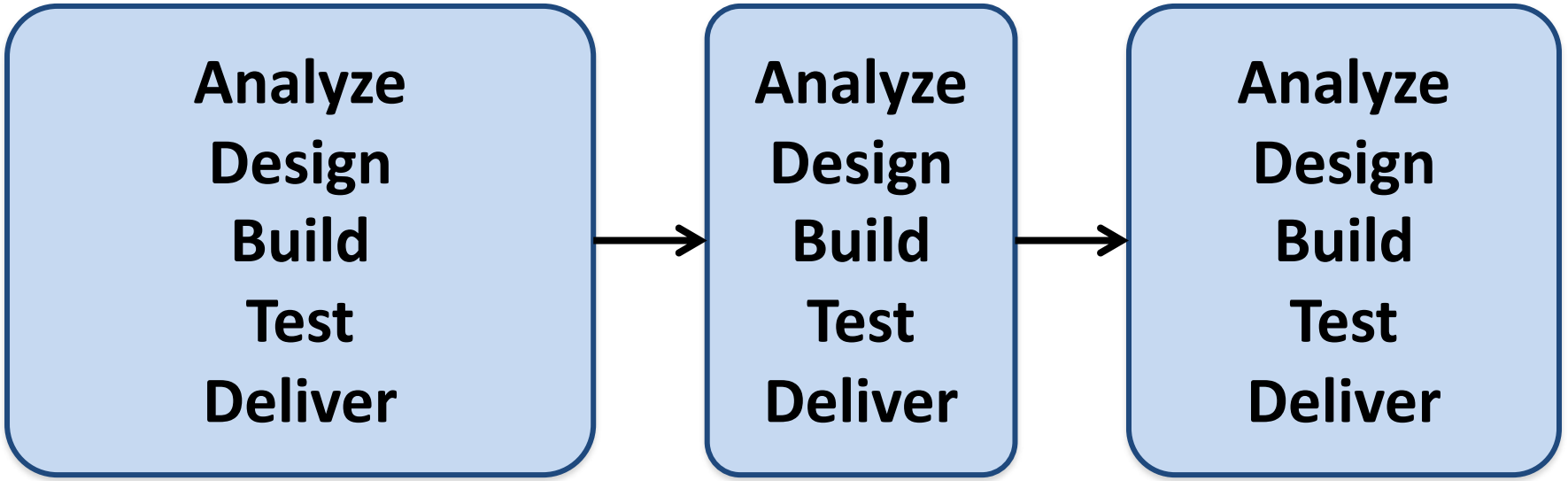
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iterative Life Cycle



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

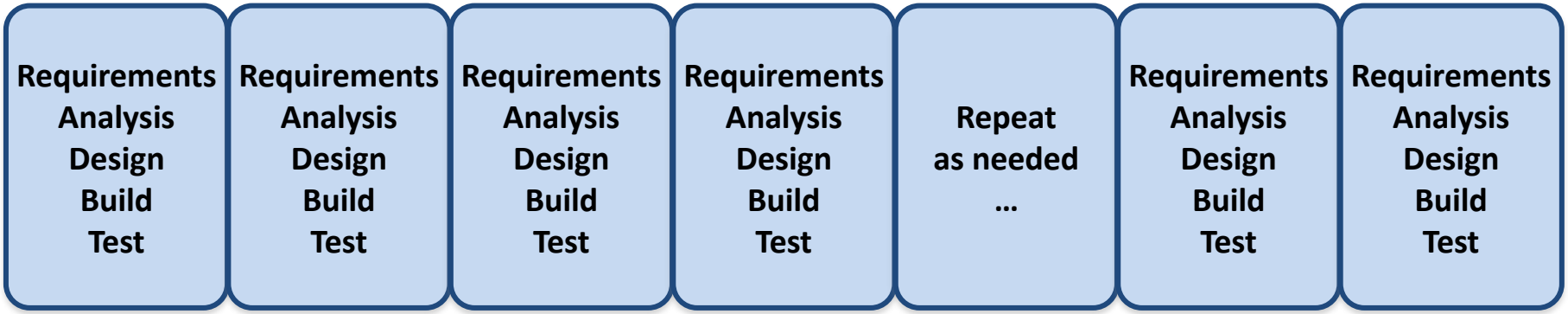
A Life Cycle of Varying-Sized Increments



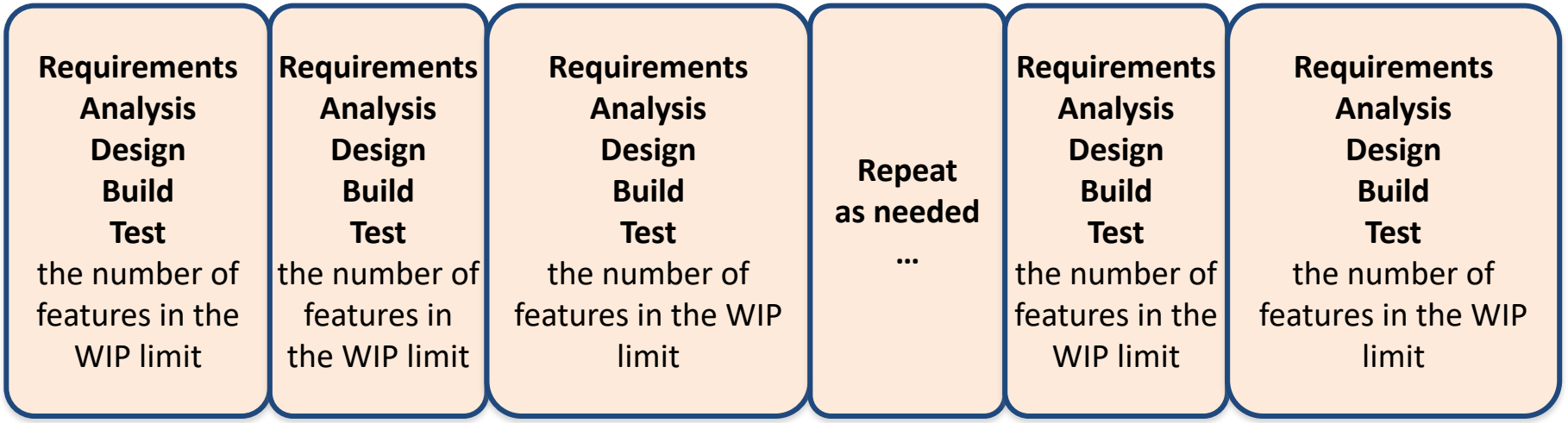
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

Iteration-Based and Flow-Based Agile Life Cycles

Iteration-Based Agile

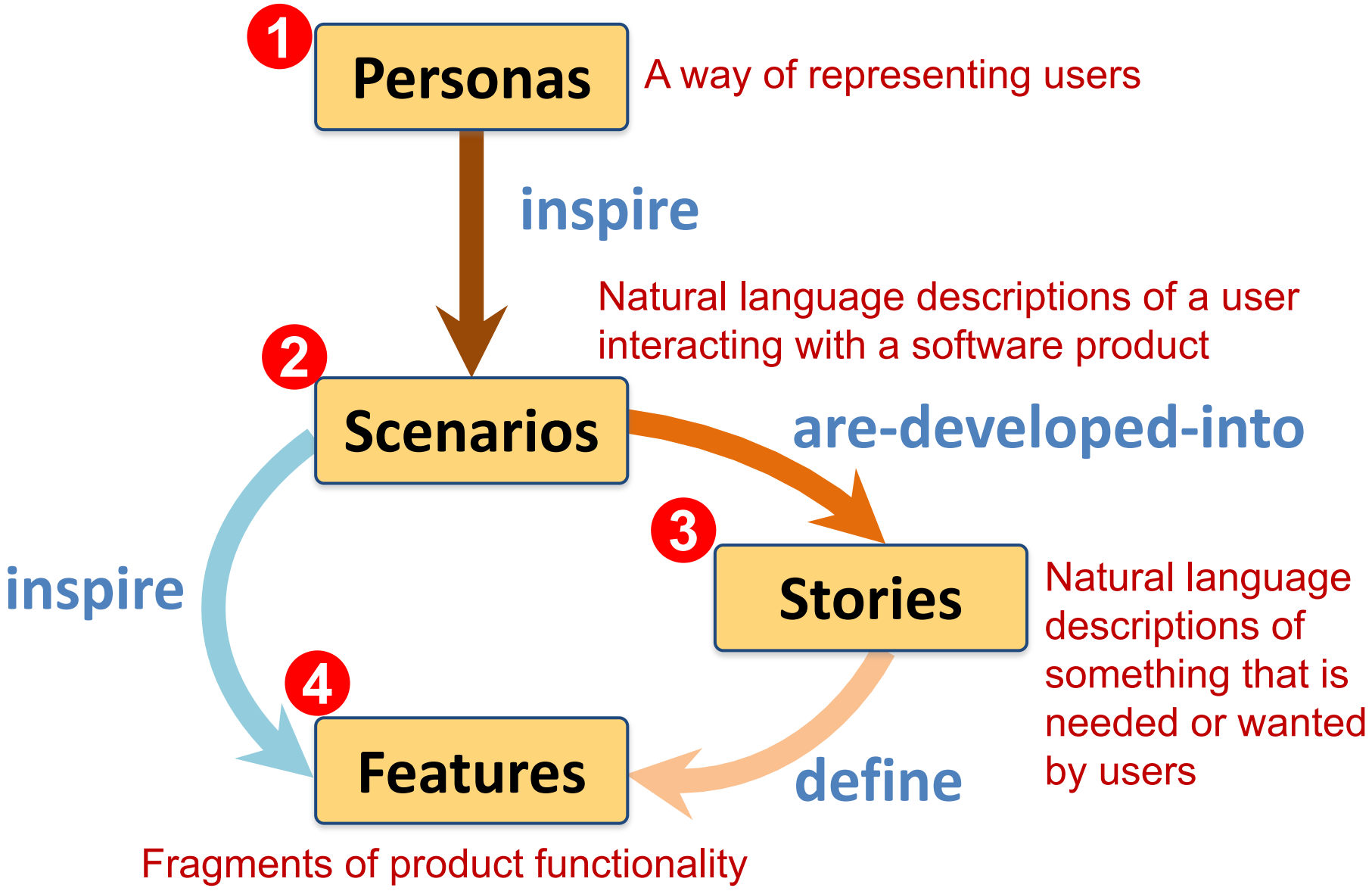


Flow-Based Agile



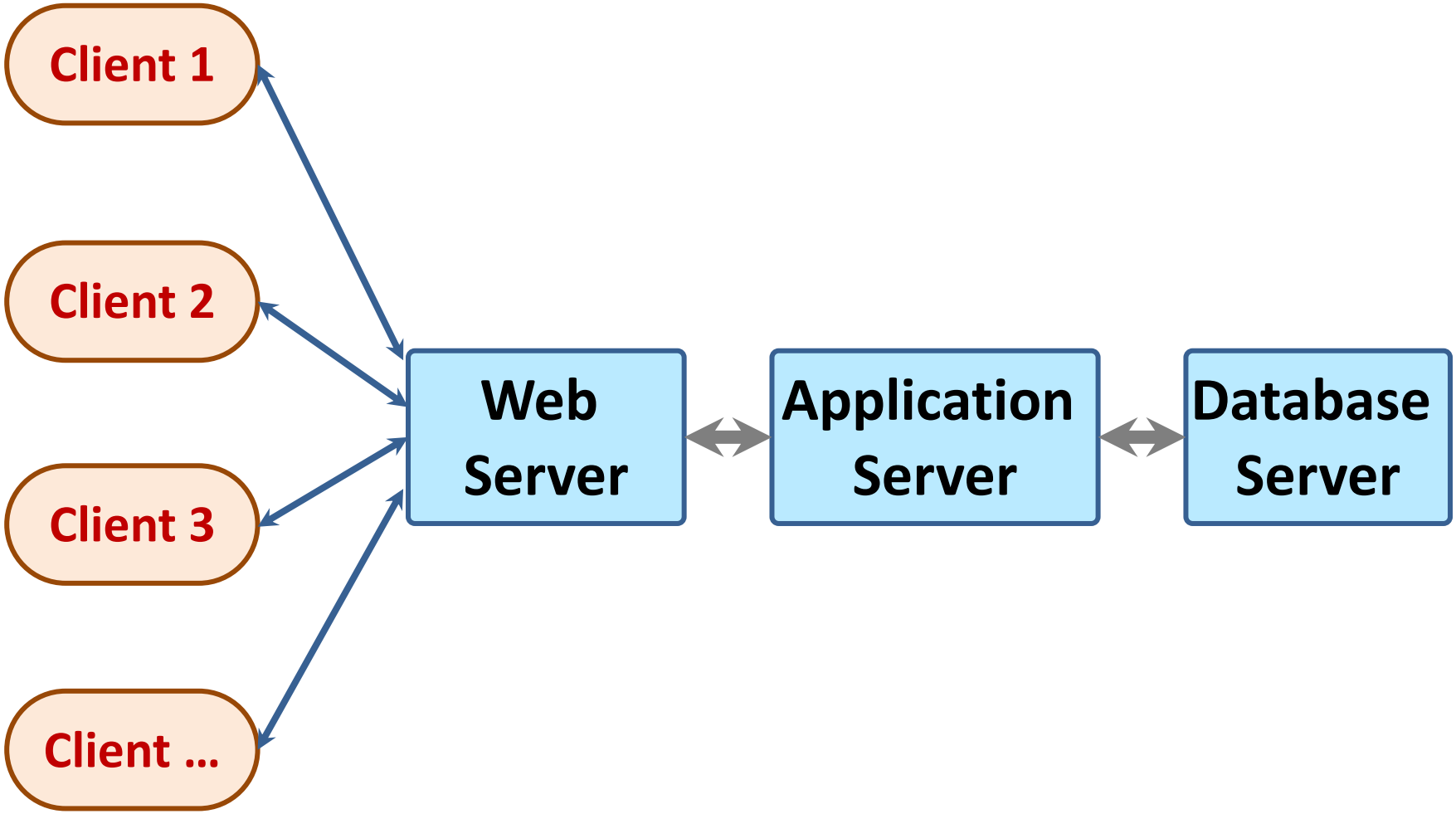
Source: Project Management Institute (2017), Agile Practice Guide, Project Management Institute

From personas to features



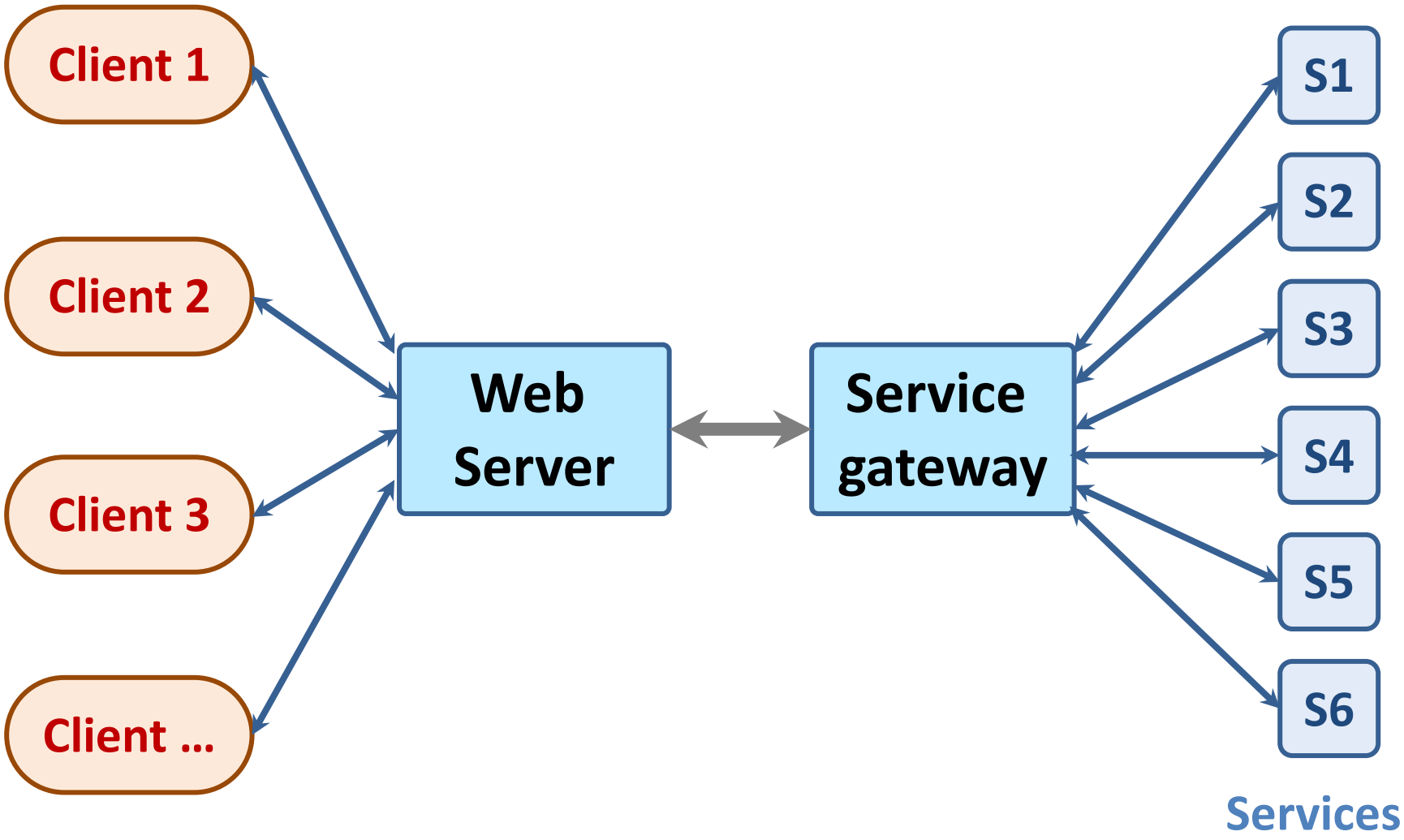
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Multi-tier client-server architecture

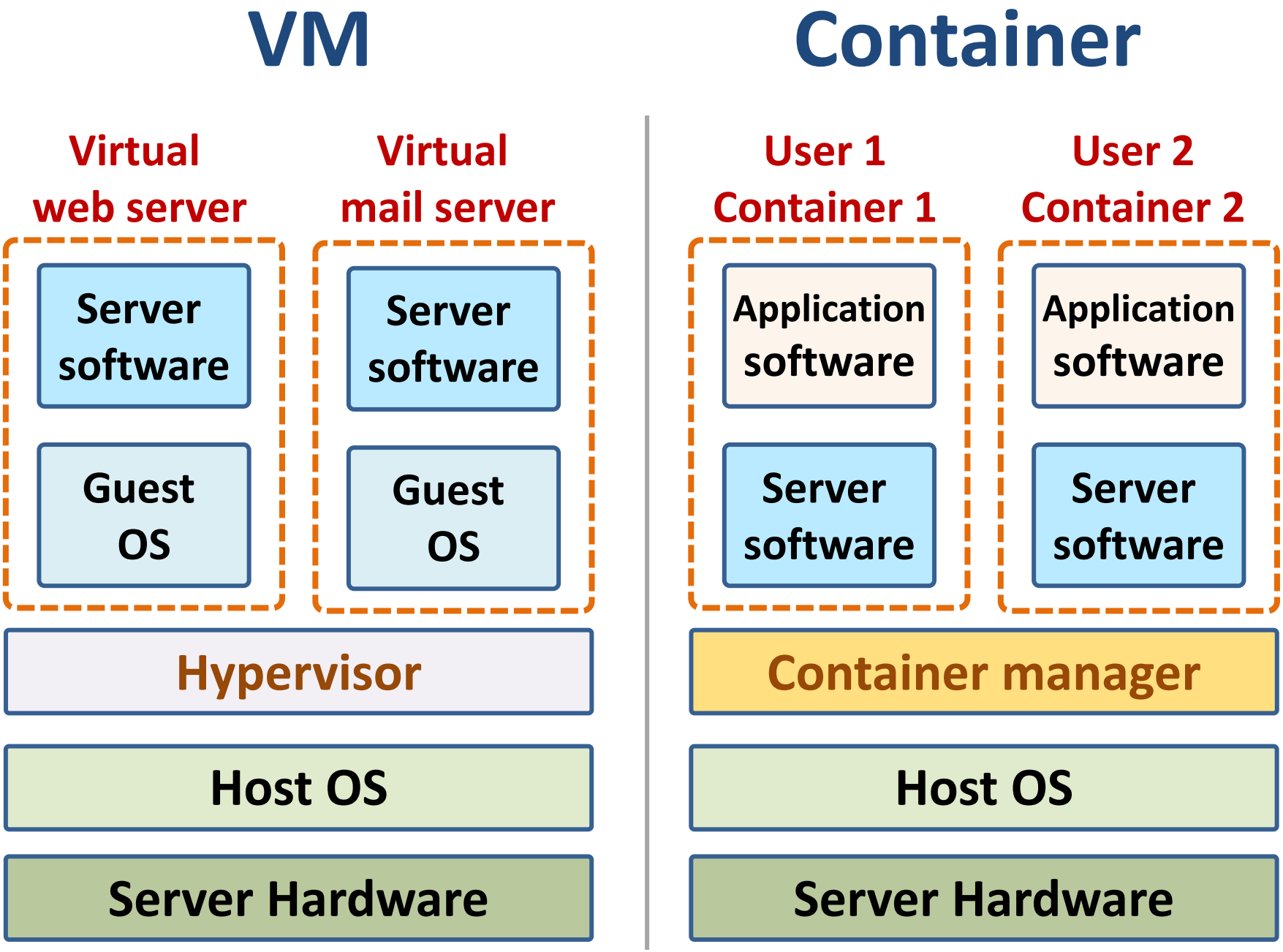


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Service-oriented Architecture

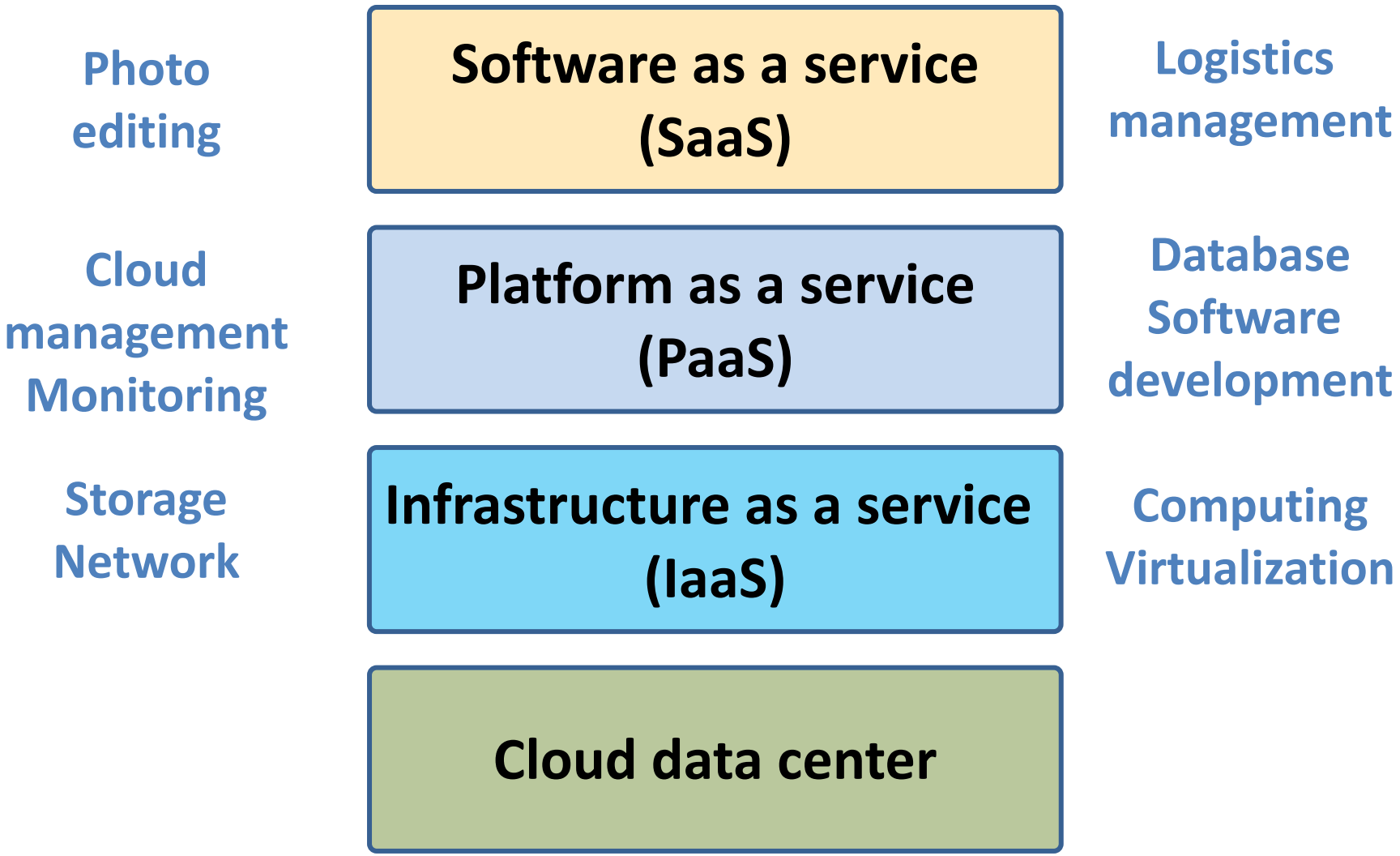


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.



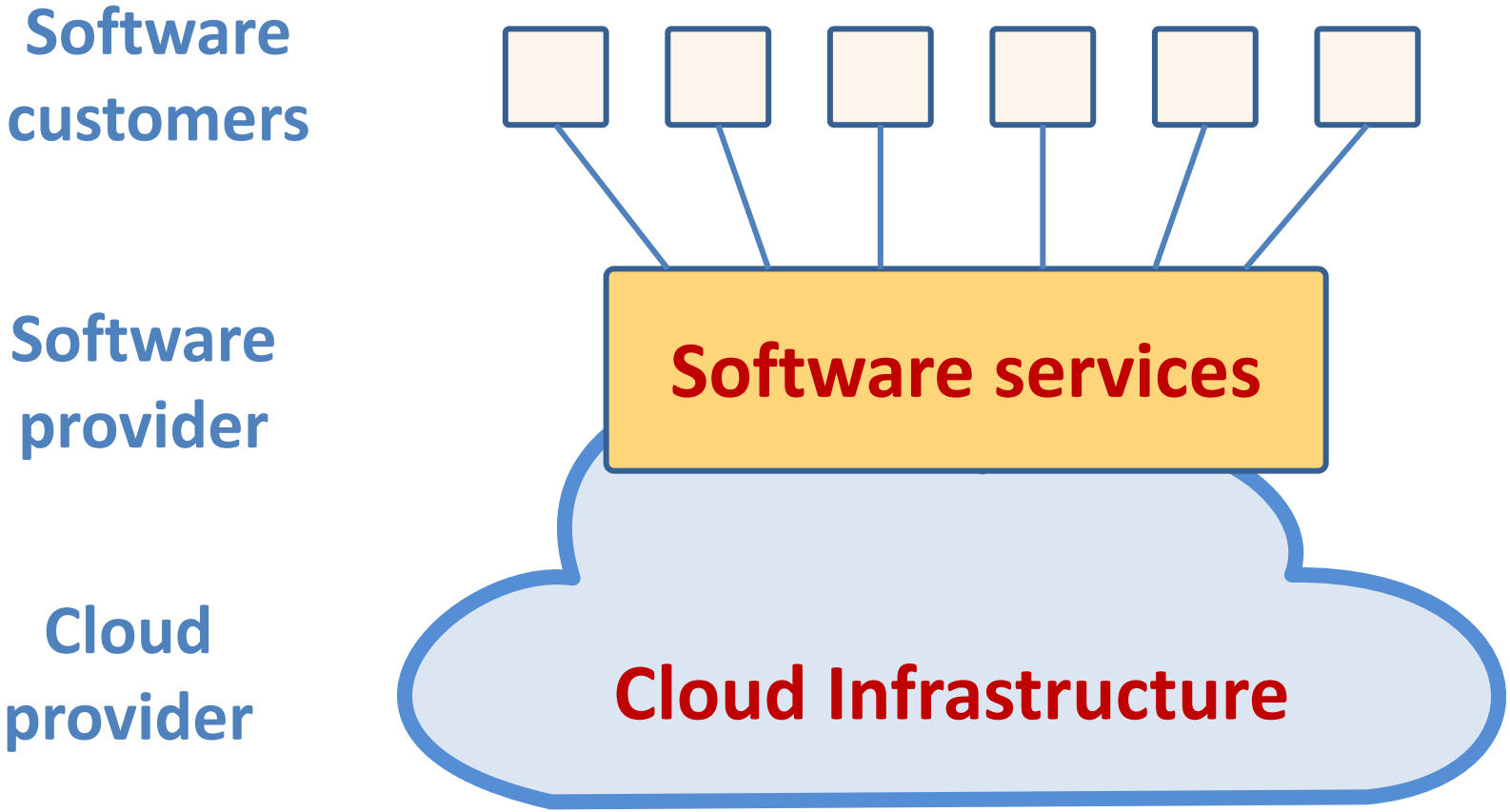
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Everything as a service



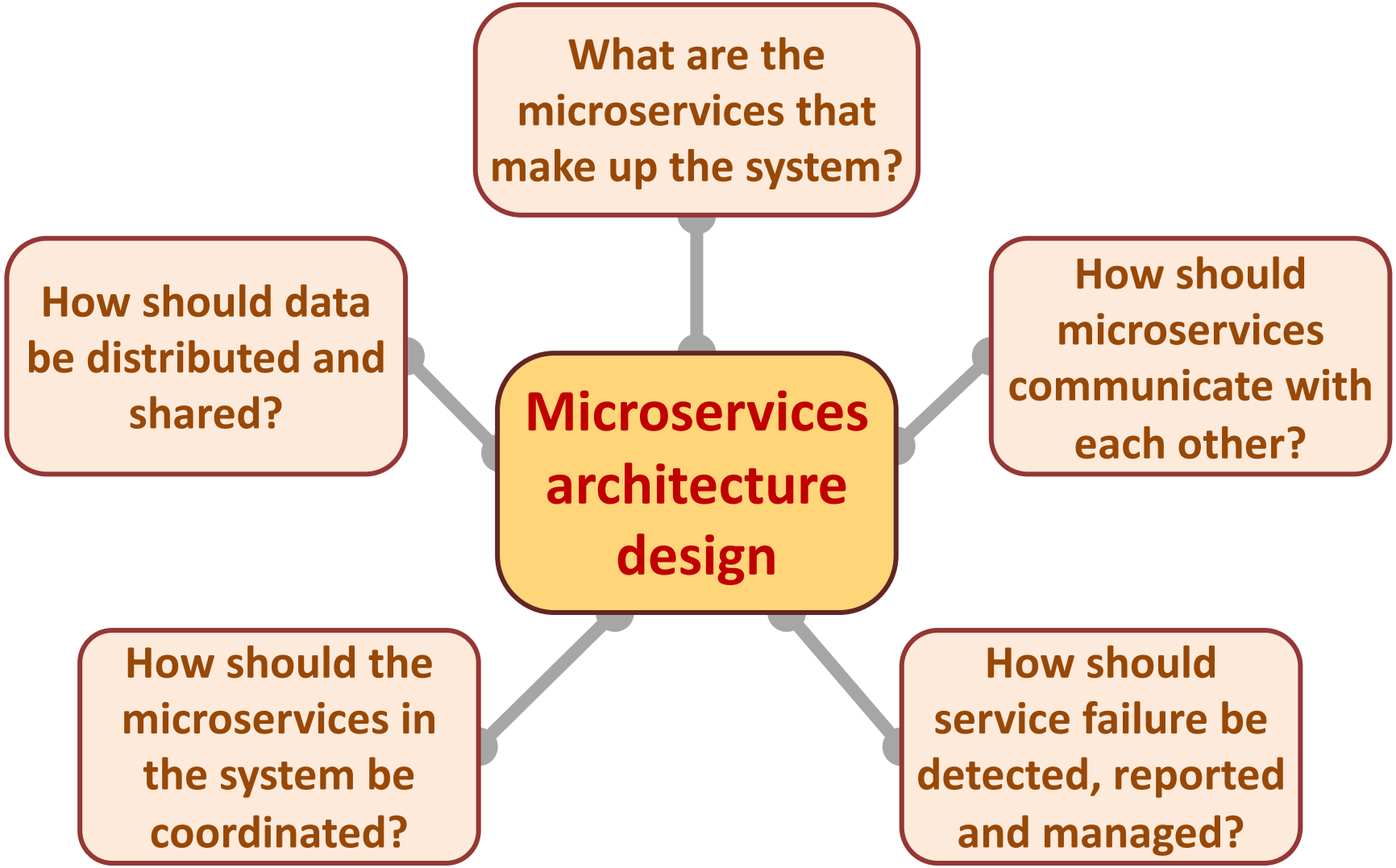
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software as a service



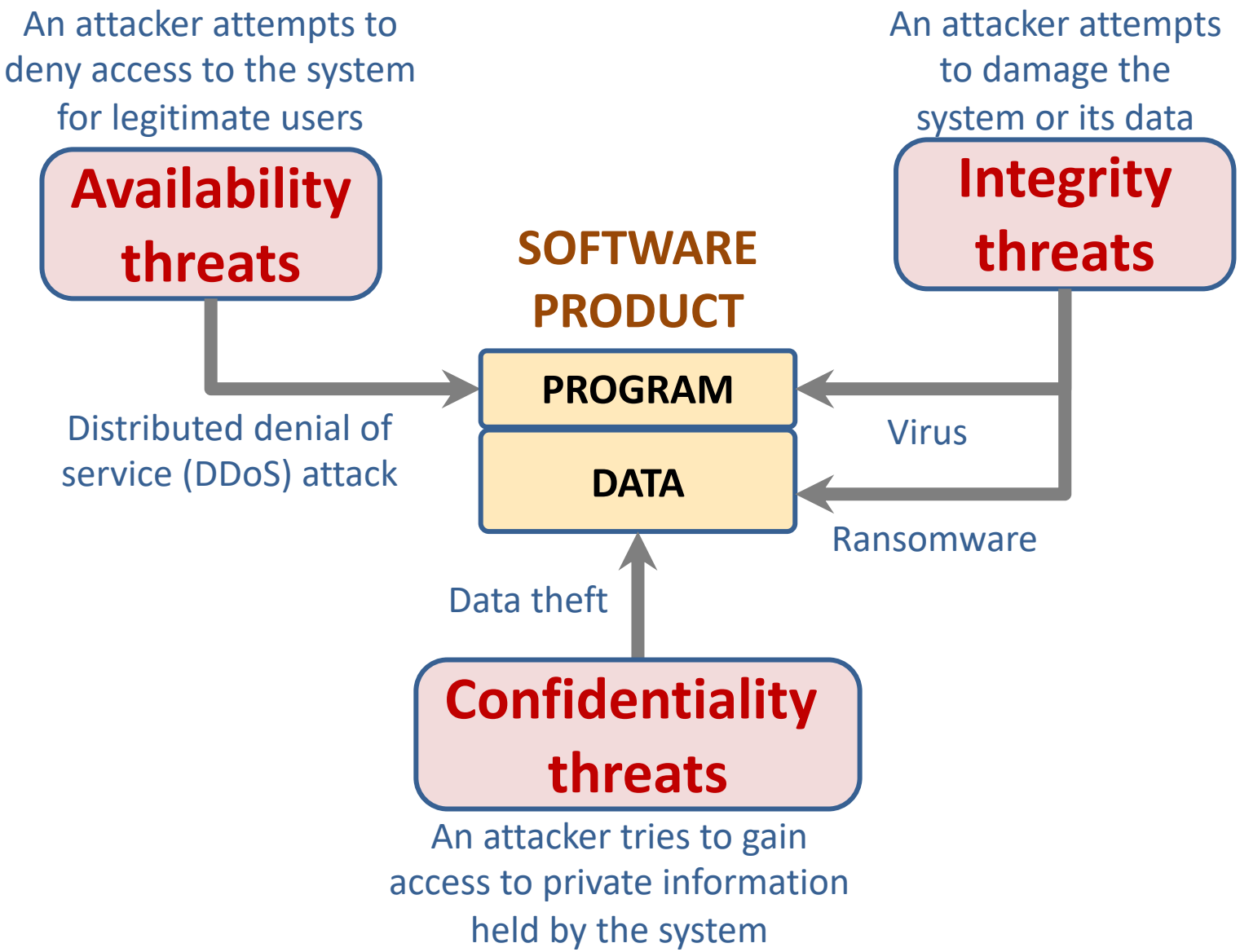
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Microservices architecture – key design questions



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

Types of security threat

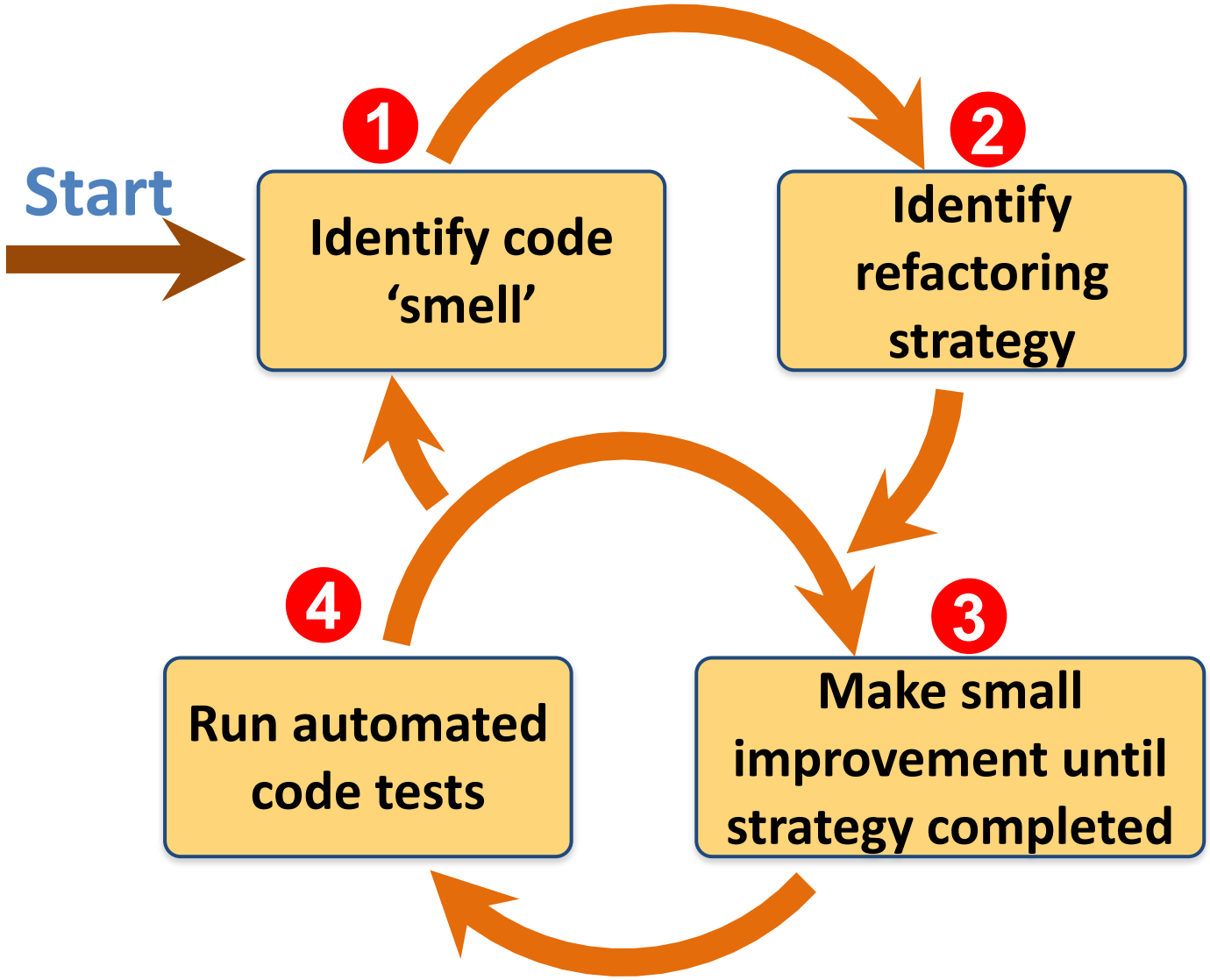


Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Software product quality attributes

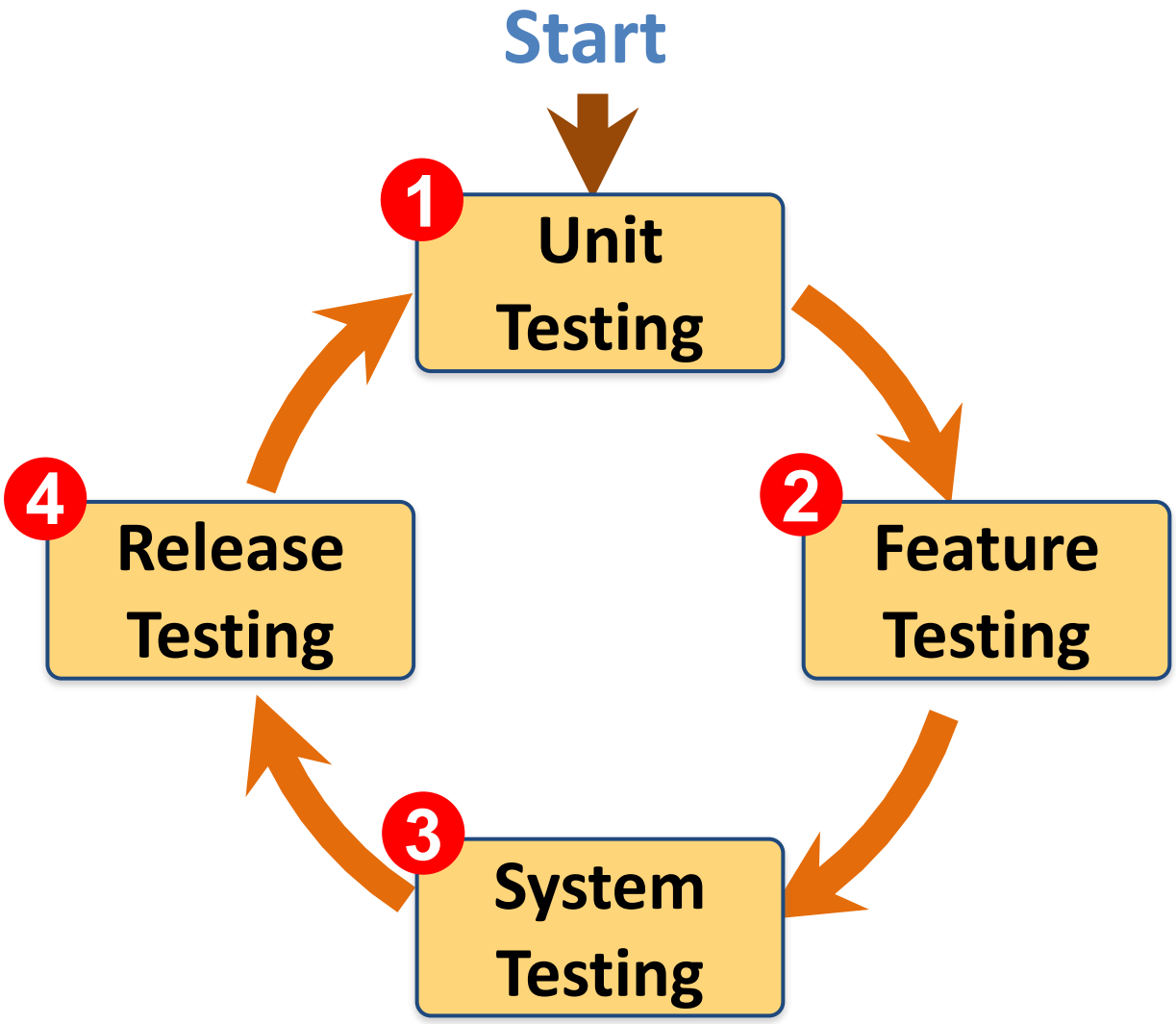
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

A refactoring process



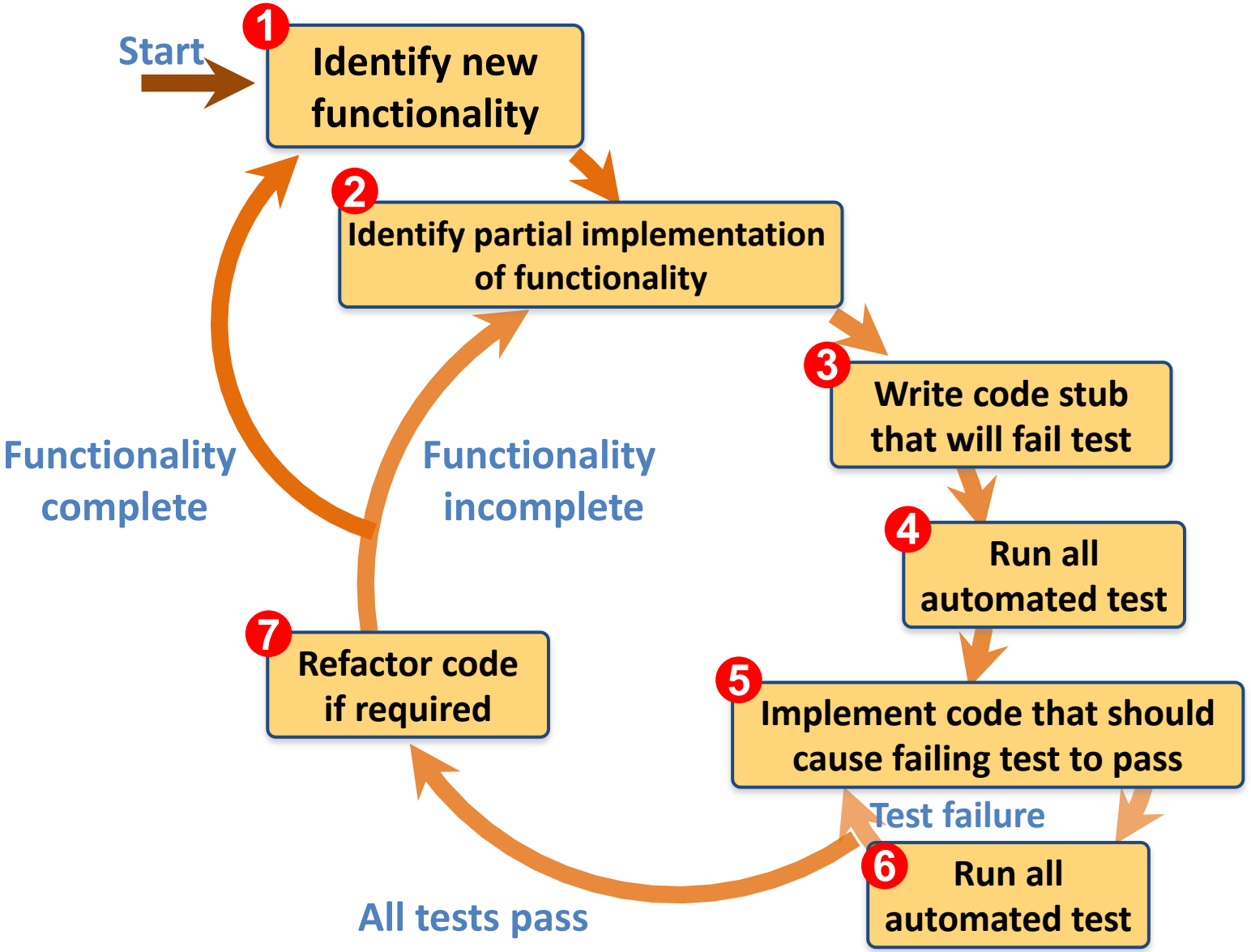
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Functional testing



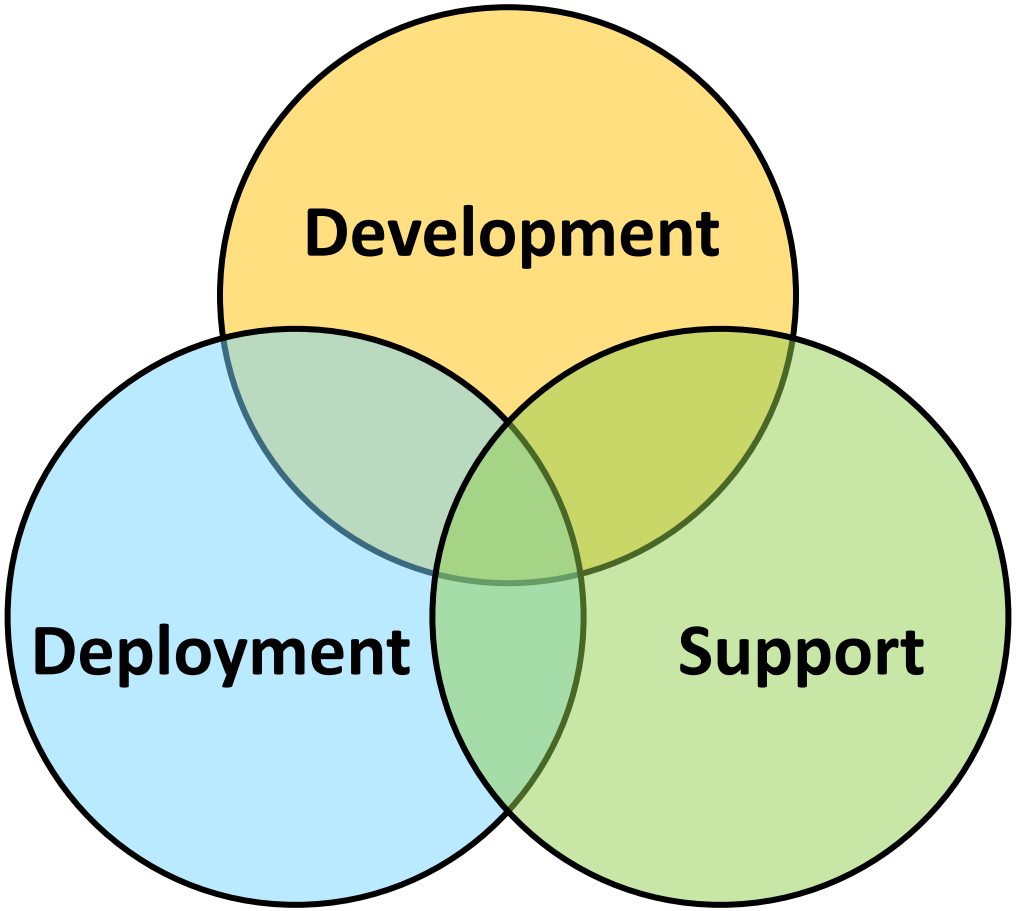
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Test-driven development (TDD)



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

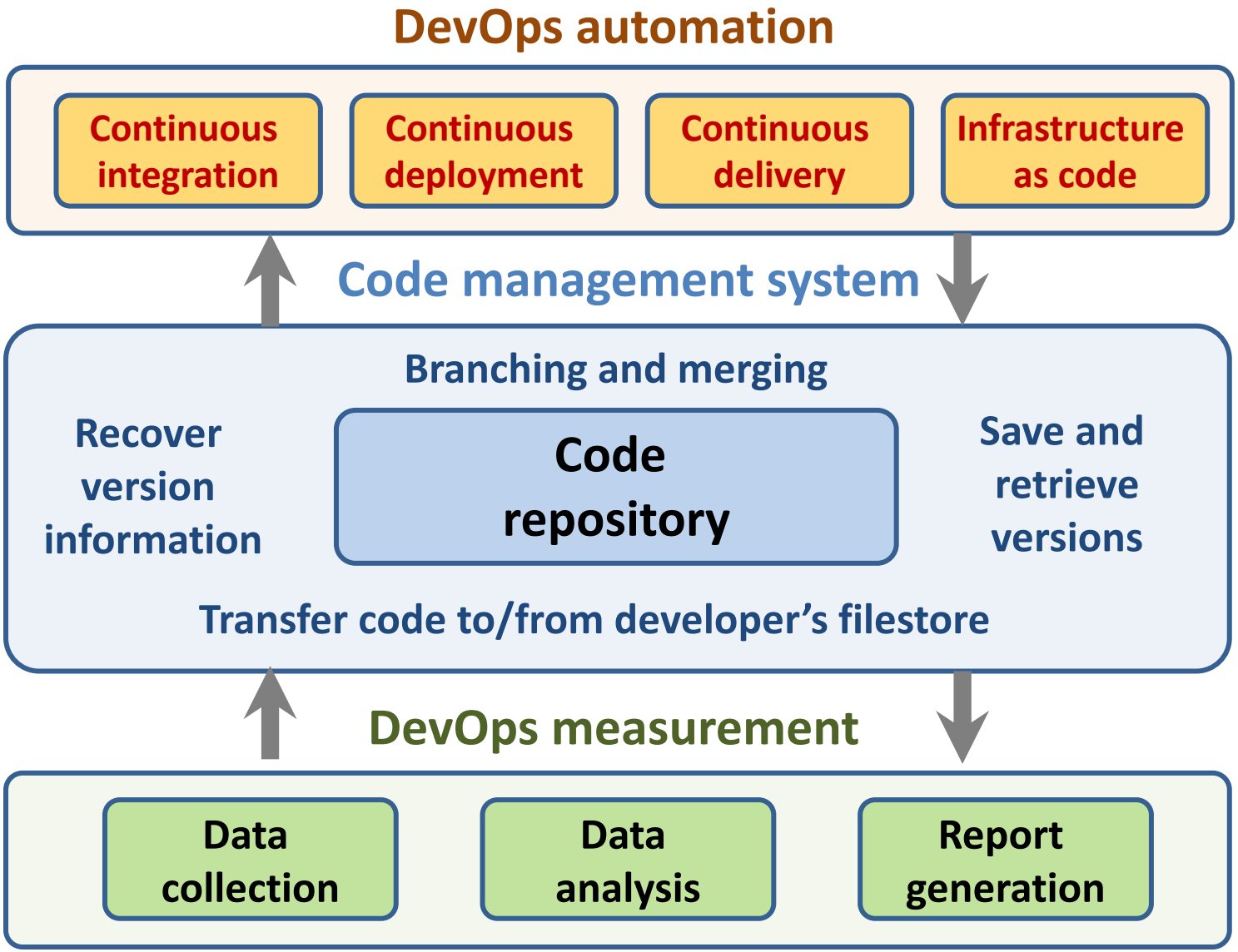
DevOps



Multi-skilled DevOps team

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

Code management and DevOps



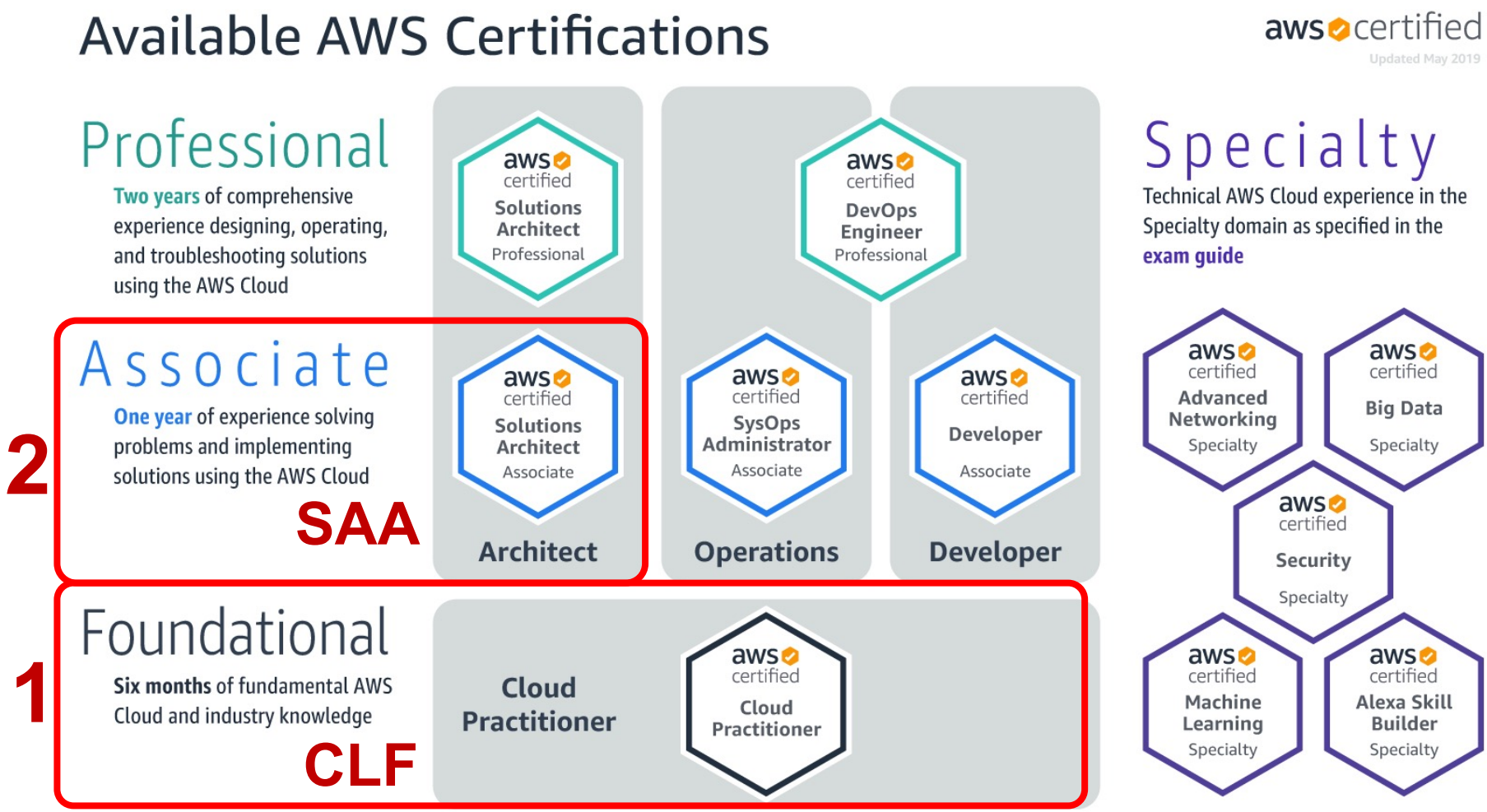
Source: Ian Sommerville (2019), Engineering Software Products: An Introduction to Modern Software Engineering, Pearson.

Cloud Computing and Cloud Software Architecture

Outline

- Cloud Computing and Cloud Software Architecture
- AWS Certified Cloud Practitioner (CLF-C01)
- AWS Certified Solutions Architect – Associate (SAA-C02)
- Web Application with AWS Core Services
- AWS Serverless Architecture
- Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

AWS Certifications



https://aws.amazon.com/certification/

AWS Certifications

AWS Certified Cloud Practitioner

- This certification provides individuals in a larger variety of cloud and technology roles with a way to validate their AWS Cloud knowledge and enhance their professional credibility.
- This exam covers four domains, including cloud concepts, security, technology, and billing and pricing.

https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Solutions Architect – Associate

- This certification validates your ability to effectively demonstrate knowledge of how to architect and deploy secure and robust applications on AWS technologies.
- This exam is for anyone with at least one year of hands-on experience designing available, cost-efficient, fault-tolerant, and scalable and distributed systems on AWS.

https://aws.amazon.com/certification/certified-solutions-architect-associate/

AWS Academy and Certifications

- **AWS Academy Cloud Foundations (ACF)**
  - **AWS Certified Cloud Practitioner (CLF-C01)**
  - https://aws.amazon.com/certification/certified-cloud-practitioner/

1

- **AWS Academy Cloud Architecting (ACA)**
  - **AWS Certified Solutions Architect – Associate (SAA-C02)**
  - https://aws.amazon.com/certification/certified-solutions-architect-associate/

2

https://aws.amazon.com/training/awsacademy/

AWS Certified Cloud Practitioner (CLF-C01)

Domain	% of Examination
Domain 1: Cloud Concepts	26%
Domain 2: Security and Compliance	25%
Domain 3: Technology	33%
Domain 4: Billing and Pricing	16%
TOTAL	100%

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Solutions Architect – Associate (SAA-C02)

Domain	% of Examination
Domain 1: Design Resilient Architectures	30%
Domain 2: Design High-Performing Architectures	28%
Domain 3: Specify Secure Applications and Architectures	24%
Domain 4: Design Cost-Optimized Architectures	18%
TOTAL	100%

Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Certified Cloud Practitioner (CLF-C01)

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Cloud Practitioner (CLF-C01)

- **Domain 1: Cloud Concepts**
  - 1.1 Define the AWS Cloud and its value proposition
  - 1.2 Identify aspects of AWS Cloud economics
  - 1.3 List the different cloud architecture design principles

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Cloud Practitioner (CLF-C01)

- **Domain 2: Security and Compliance**
  - 2.1 Define the AWS shared responsibility model
  - 2.2 Define AWS Cloud security and compliance concepts
  - 2.3 Identify AWS access management capabilities
  - 2.4 Identify resources for security support

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Cloud Practitioner (CLF-C01)

- **Domain 3: Technology**
  - 3.1 Define methods of deploying and operating in the AWS Cloud
  - 3.2 Define the AWS global infrastructure
  - 3.3 Identify the core AWS services
  - 3.4 Identify resources for technology support

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Cloud Practitioner (CLF-C01)

- **Domain 4: Billing and Pricing**
  - 4.1 Compare and contrast the various pricing models for AWS
  - 4.2 Recognize the various account structures in relation to AWS billing and pricing
  - 4.3 Identify resources available for billing support

Source: https://aws.amazon.com/certification/certified-cloud-practitioner/

AWS Certified Solutions Architect – Associate (SAA-C02)

Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Certified Solutions Architect – Associate (SAA-C02)

- **Domain 1: Design Resilient Architectures**
  - 1.1 Design a multi-tier architecture solution
  - 1.2 Design highly available and/or fault-tolerant architectures
  - 1.3 Design decoupling mechanisms using AWS services
  - 1.4 Choose appropriate resilient storage

Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Certified Solutions Architect – Associate (SAA-C02)

- **Domain 2: Design High-Performing Architectures**
  - 2.1 Identify elastic and scalable compute solutions for a workload
  - 2.2 Select high-performing and scalable storage solutions for a workload
  - 2.3 Select high-performing networking solutions for a workload
  - 2.4 Choose high-performing database solutions for a workload

Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Certified Solutions Architect – Associate (SAA-C02)

- **Domain 3: Design Secure Applications and Architectures**
  - 3.1 Design secure access to AWS resources
  - 3.2 Design secure application tiers
  - 3.3 Select appropriate data security options

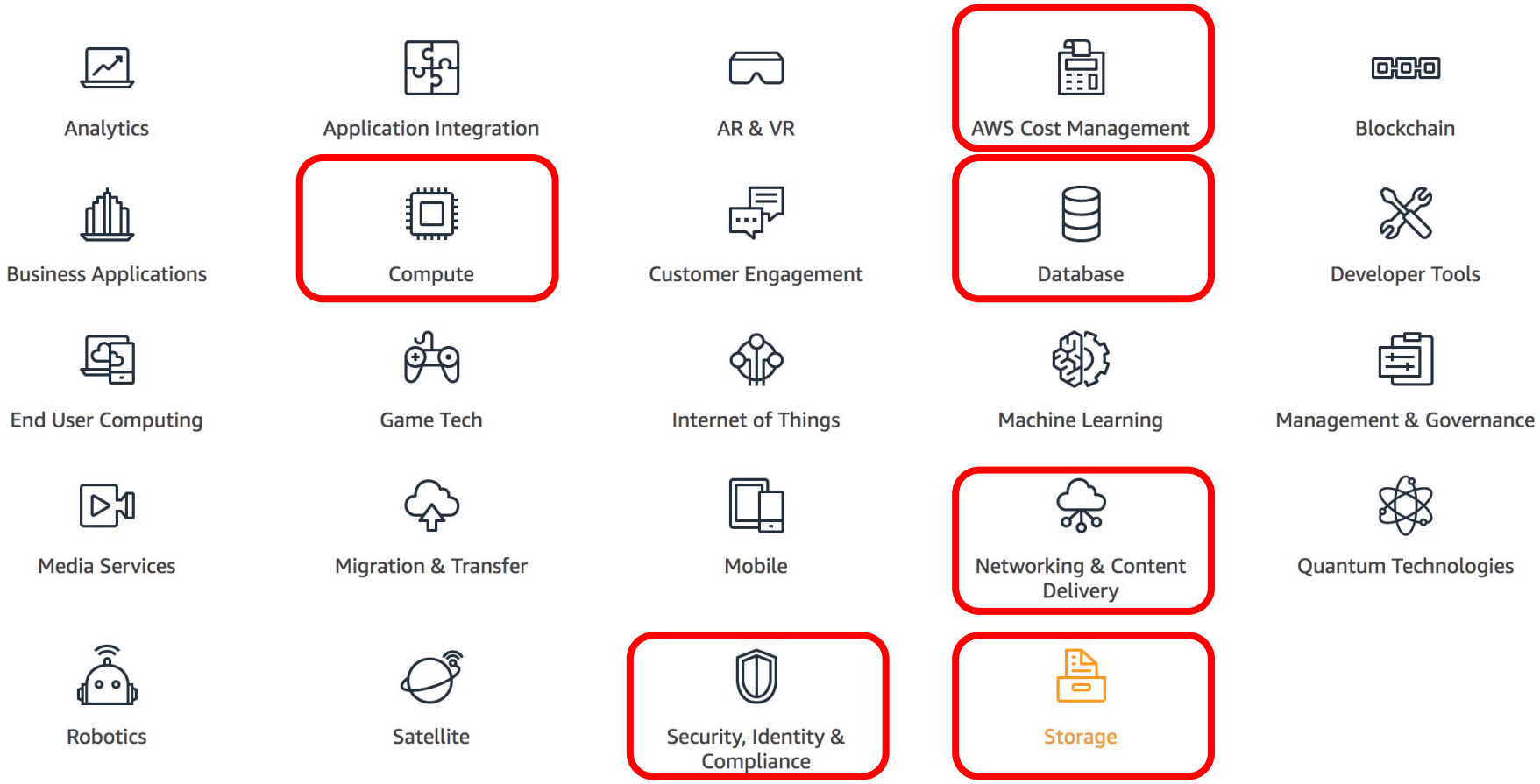
Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Certified Solutions Architect – Associate (SAA-C02)

- **Domain 4: Design Cost-Optimized Architectures**
  - 4.1 Identify cost-effective storage solutions
  - 4.2 Identify cost-effective compute and database services
  - 4.3 Design cost-optimized network architectures

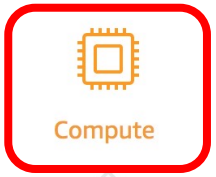
Source: https://aws.amazon.com/certification/certified-solutions-architect-associate

AWS Products and Services



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

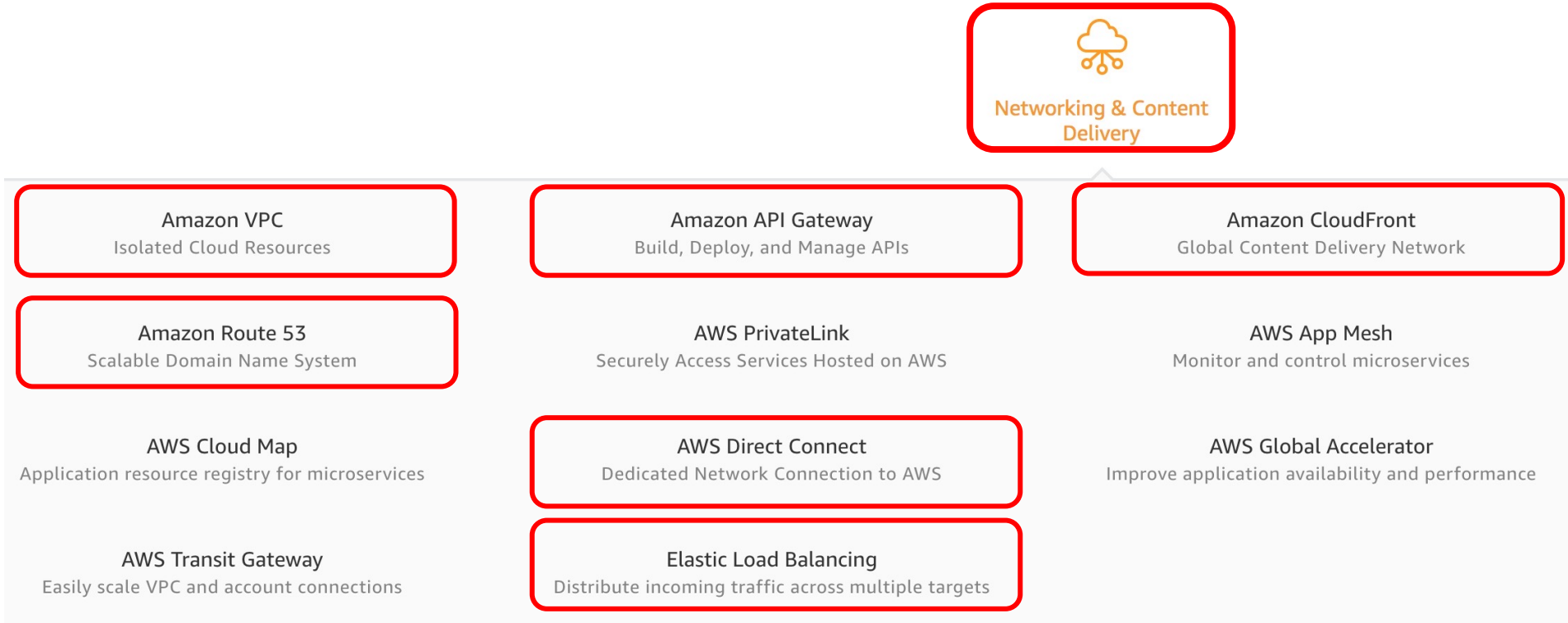
AWS Snow Family
Physical devices to migrate data into and out of AWS

AWS Storage Gateway
Hybrid Storage Integration

CloudEndure Disaster Recovery
Highly automated disaster recovery

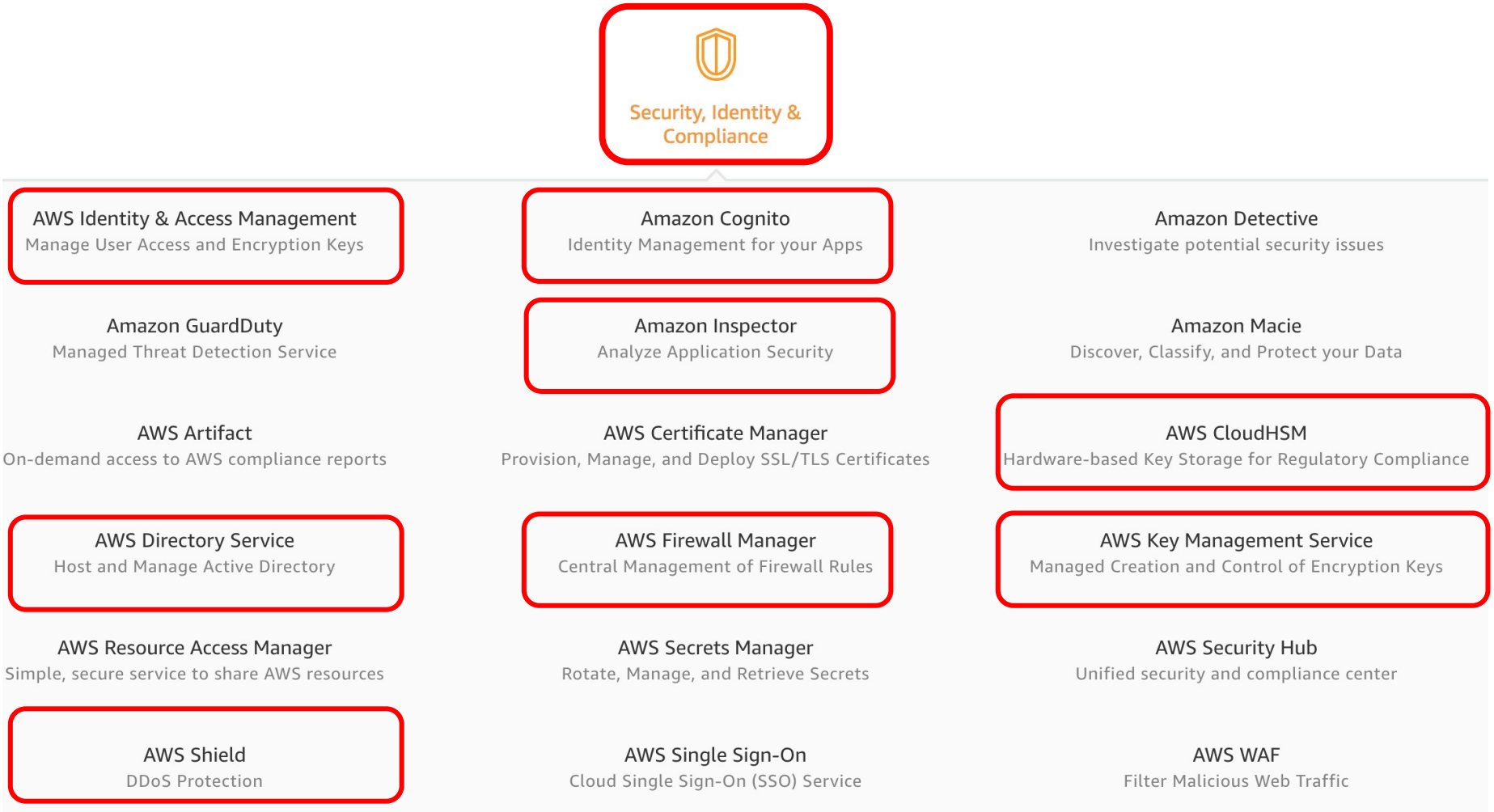
Source: https://aws.amazon.com/

AWS Networking & Content Delivery



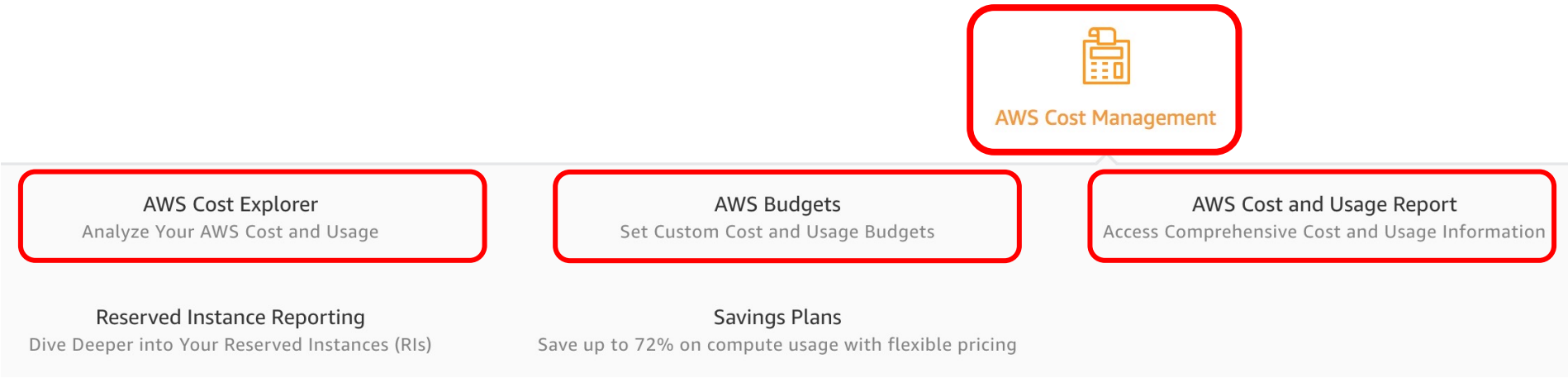
Source: https://aws.amazon.com/

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/

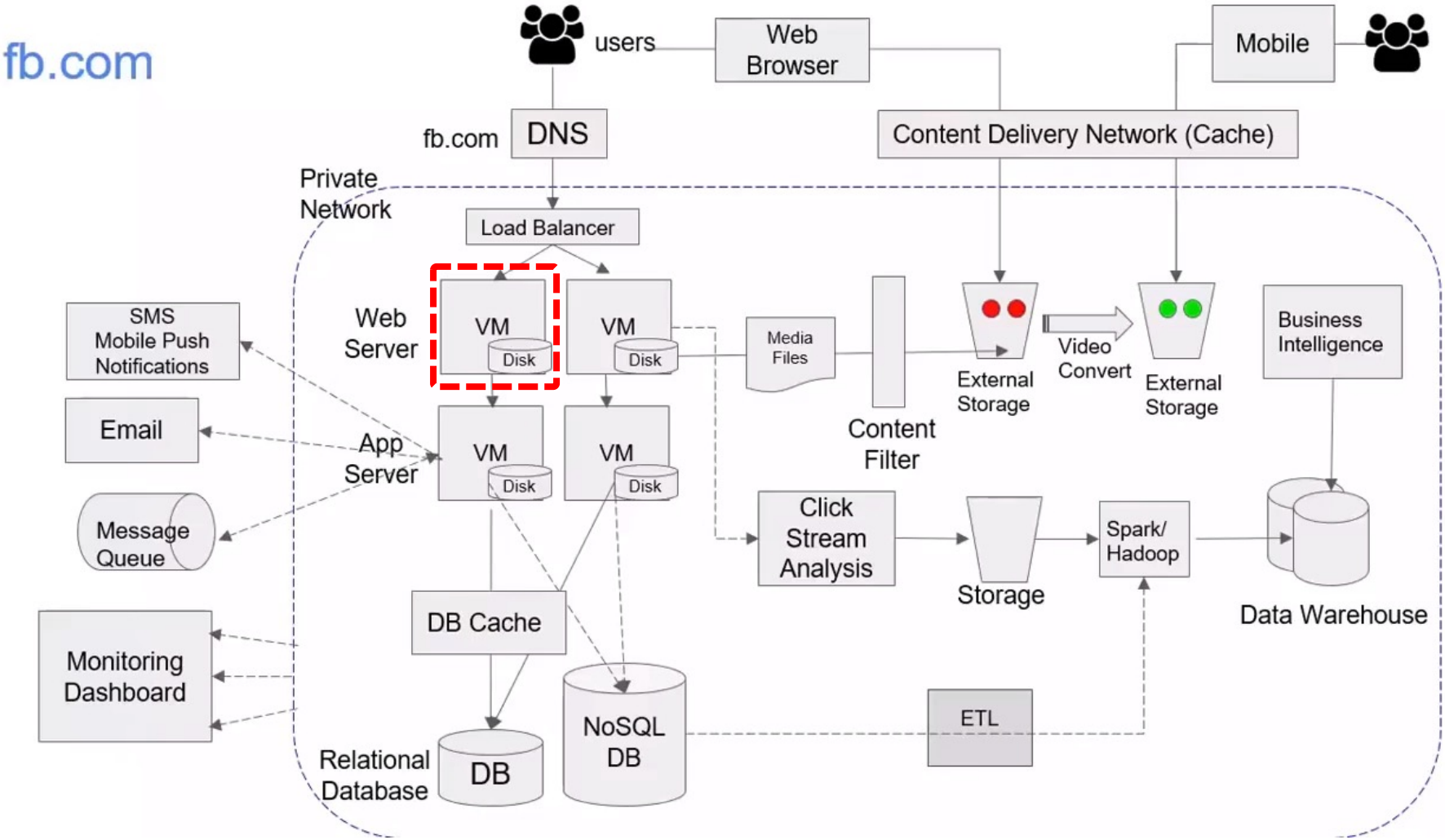
AWS Services

- 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/

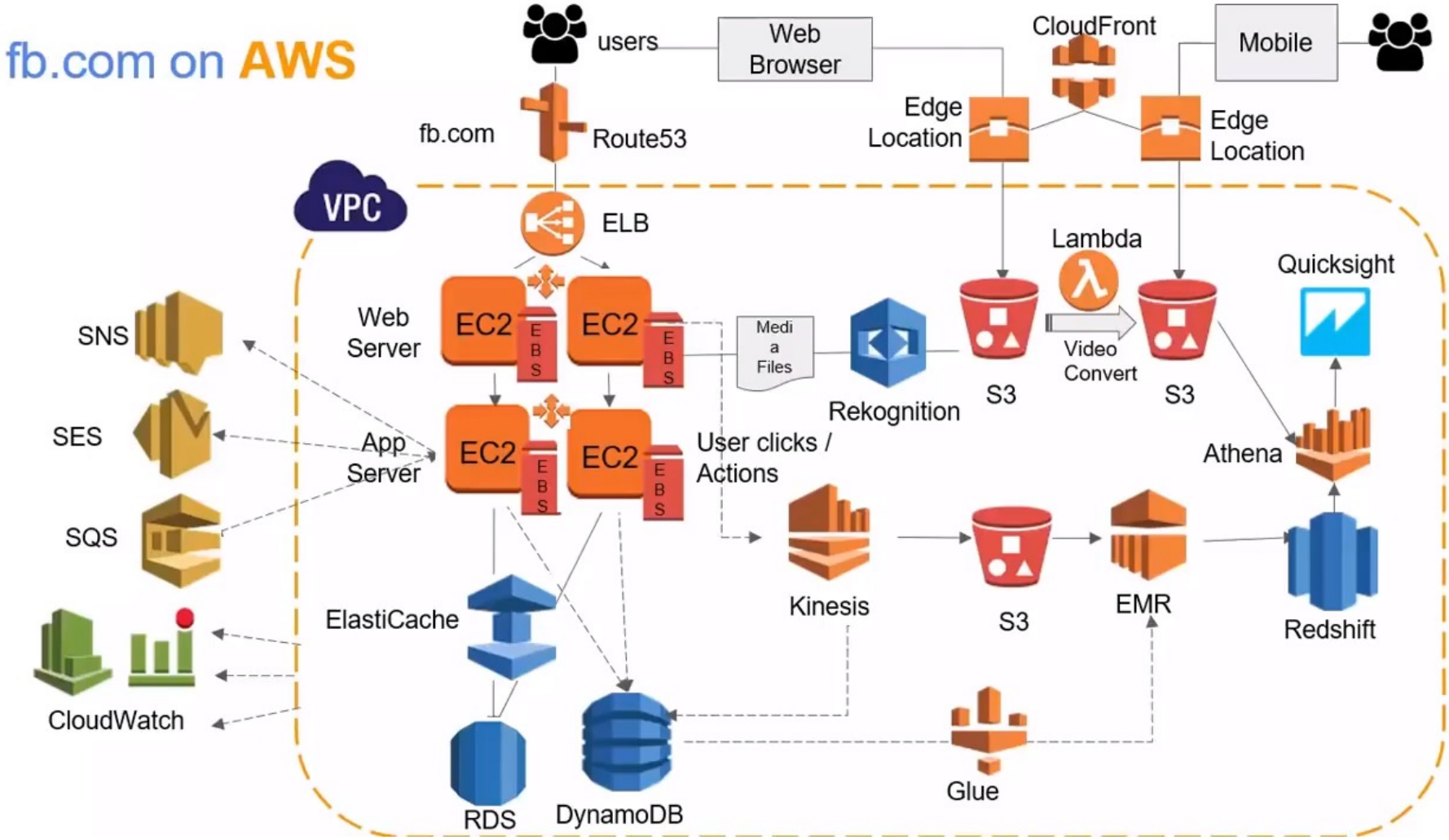
Web Application with AWS Core Services

Software Architecture



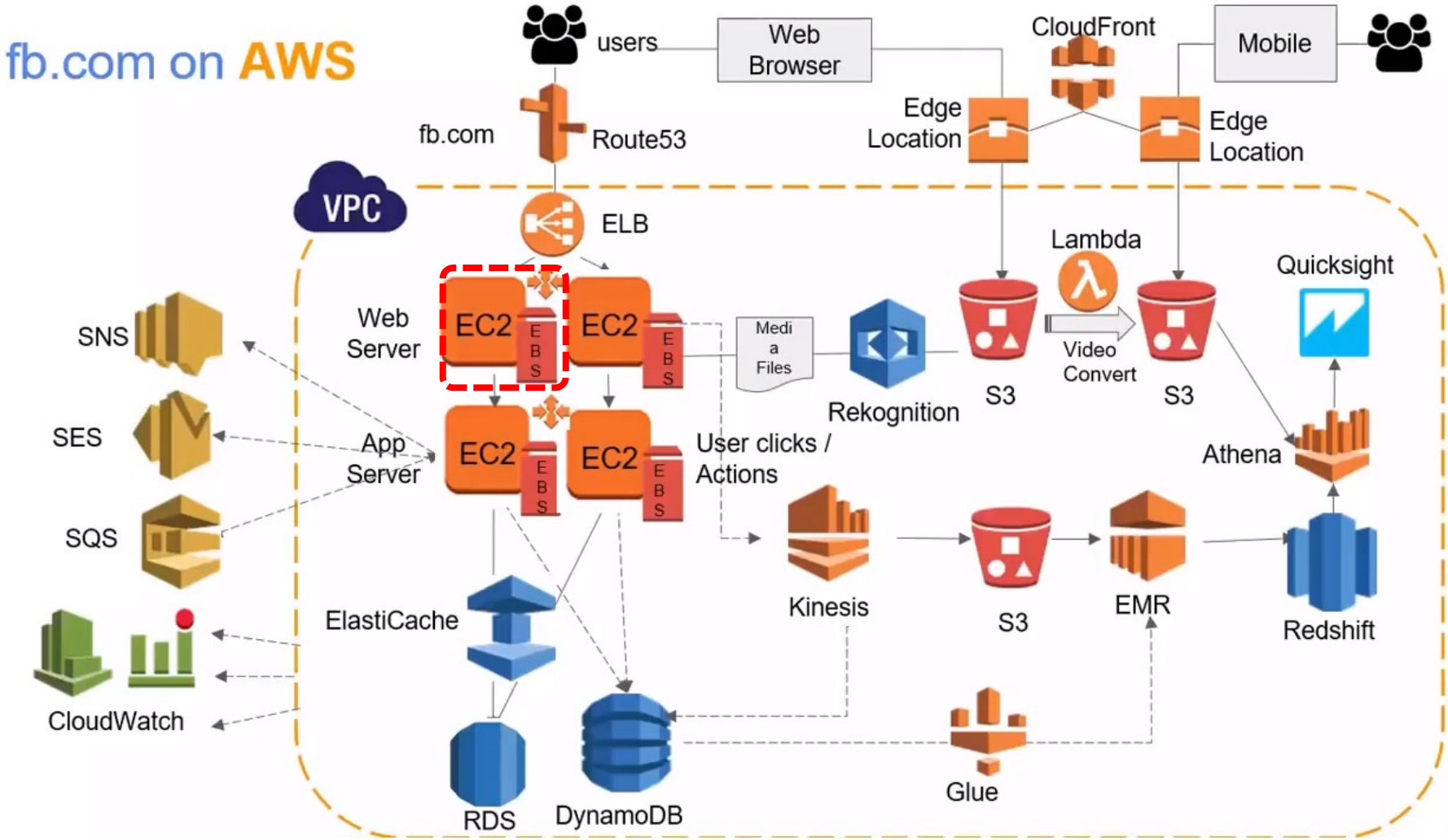
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

Cloud Software Architecture



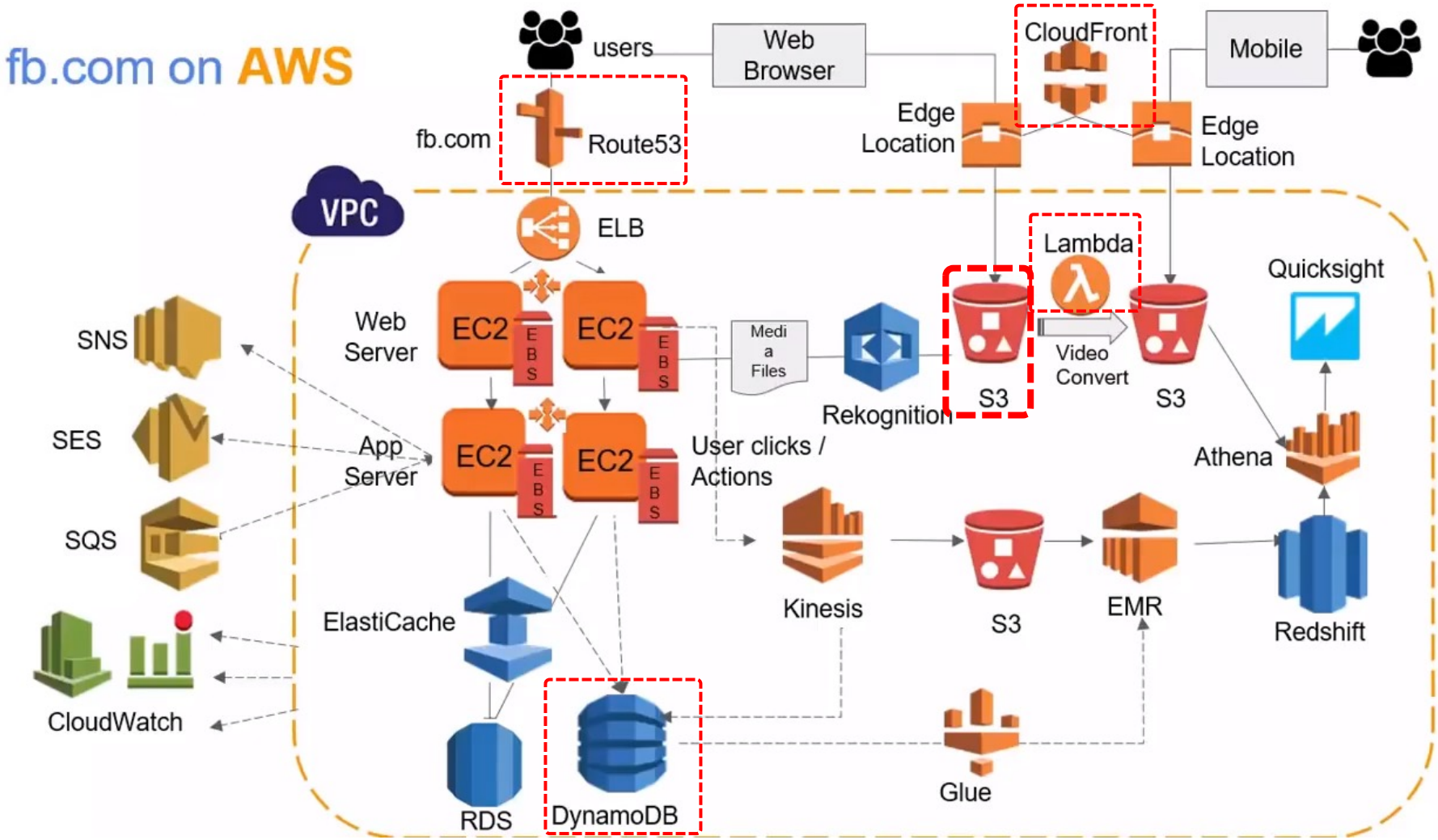
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

Cloud Software Architecture



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

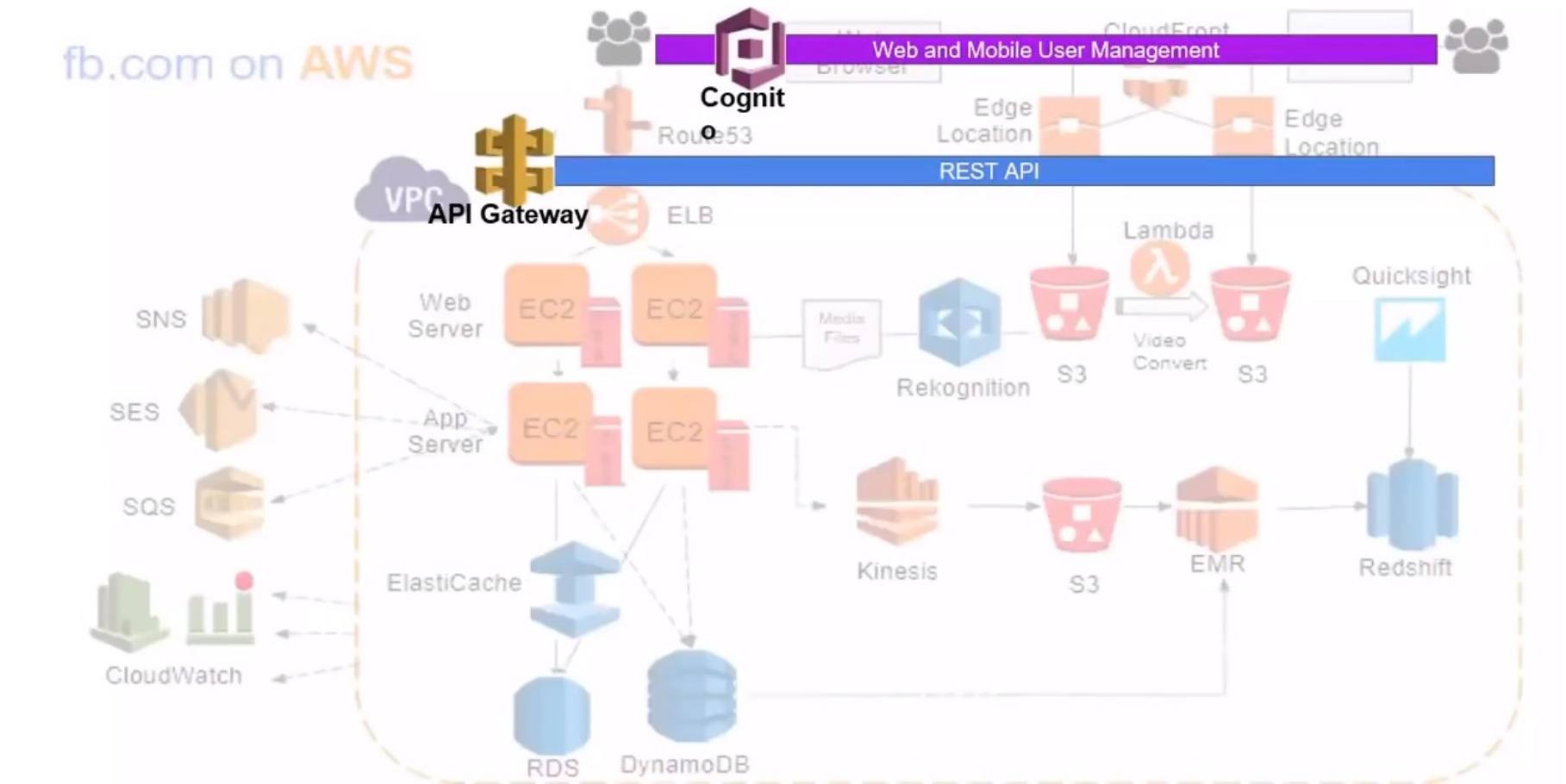
Cloud Software Architecture



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

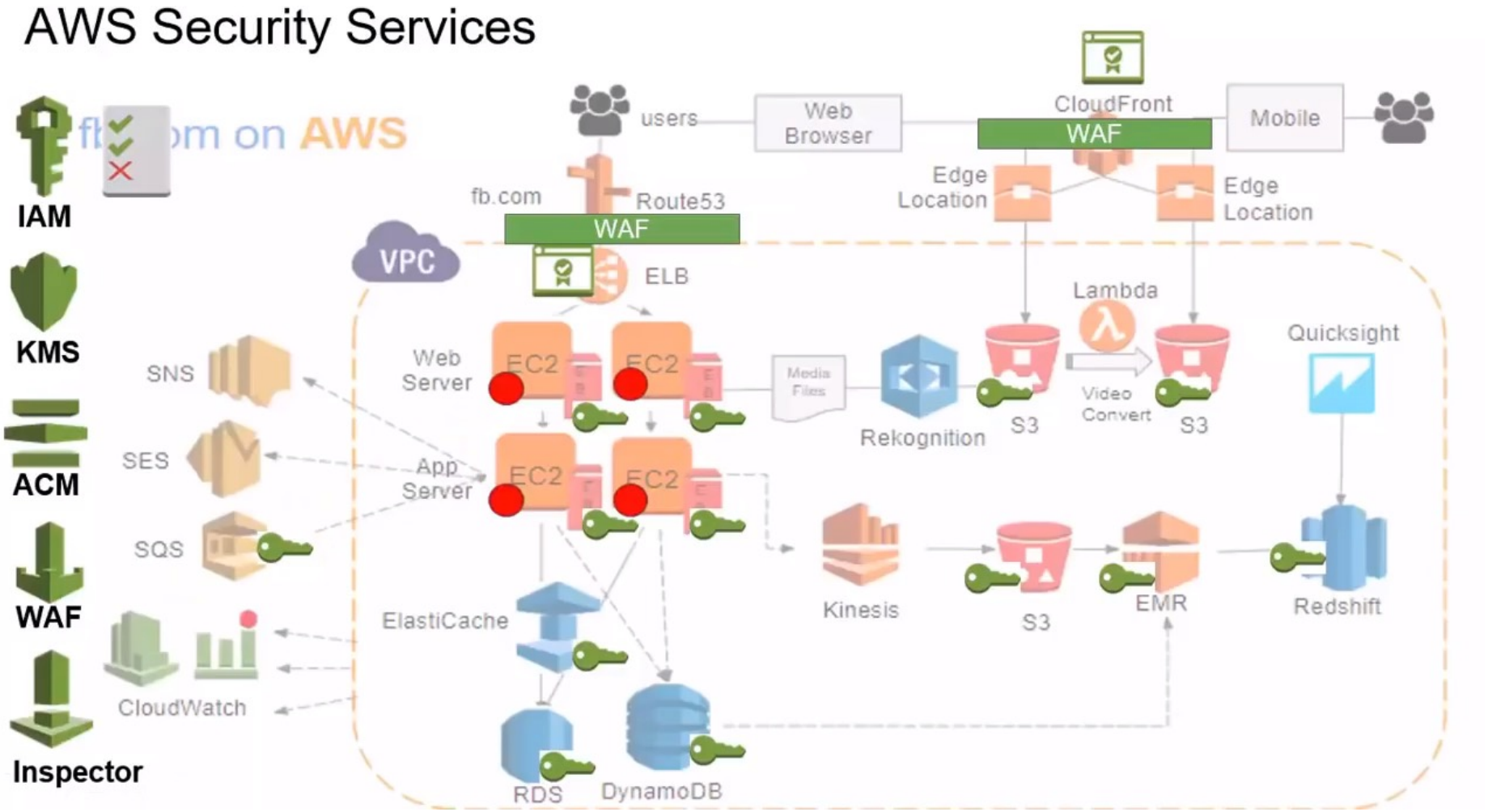
Cloud Software Architecture

AWS Application Services



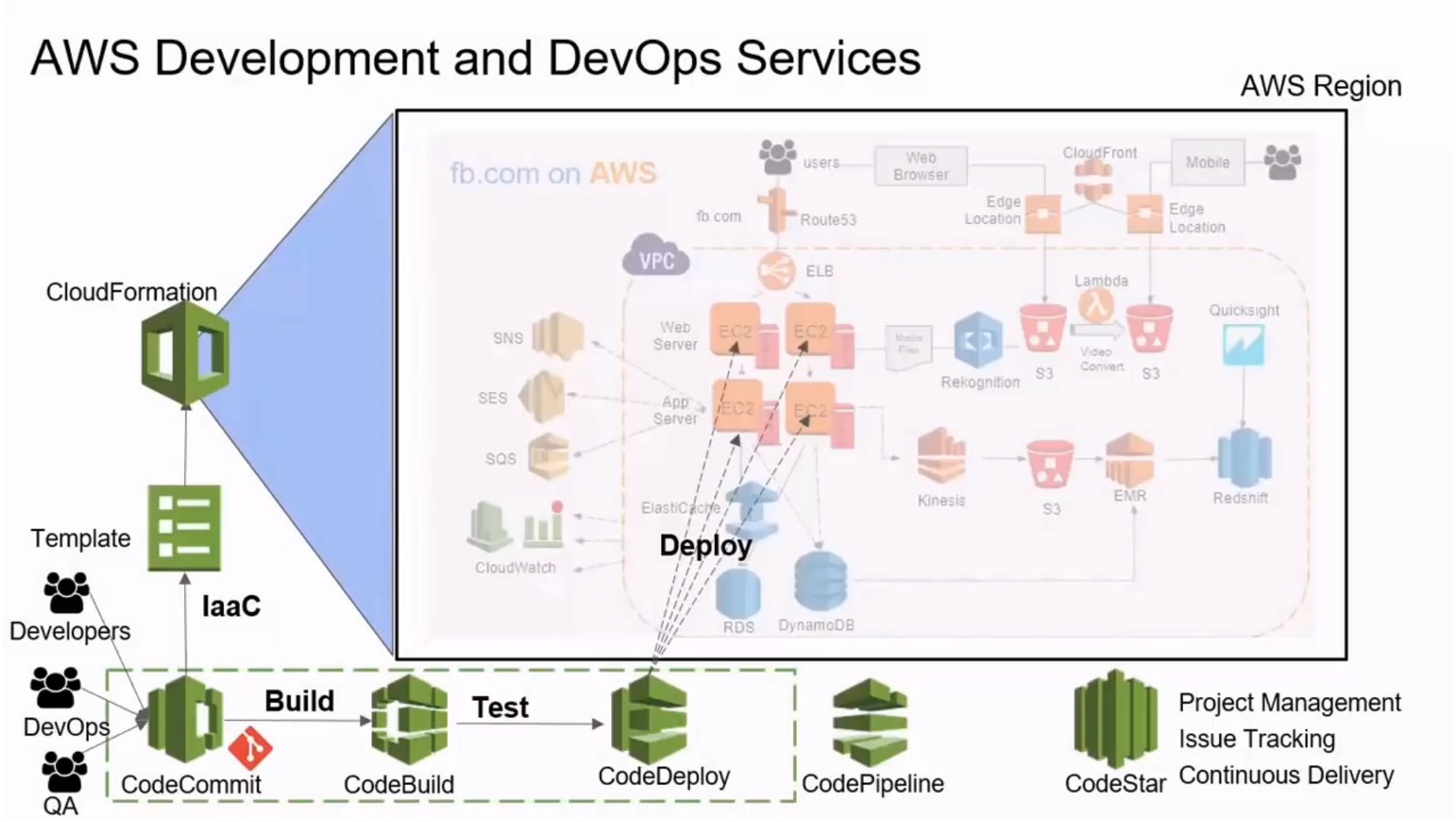
Source: AWS Training Center (2019), Introduction to AWS Services, https://youtu.be/Z3SYDTMP3ME

Cloud Software Architecture



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

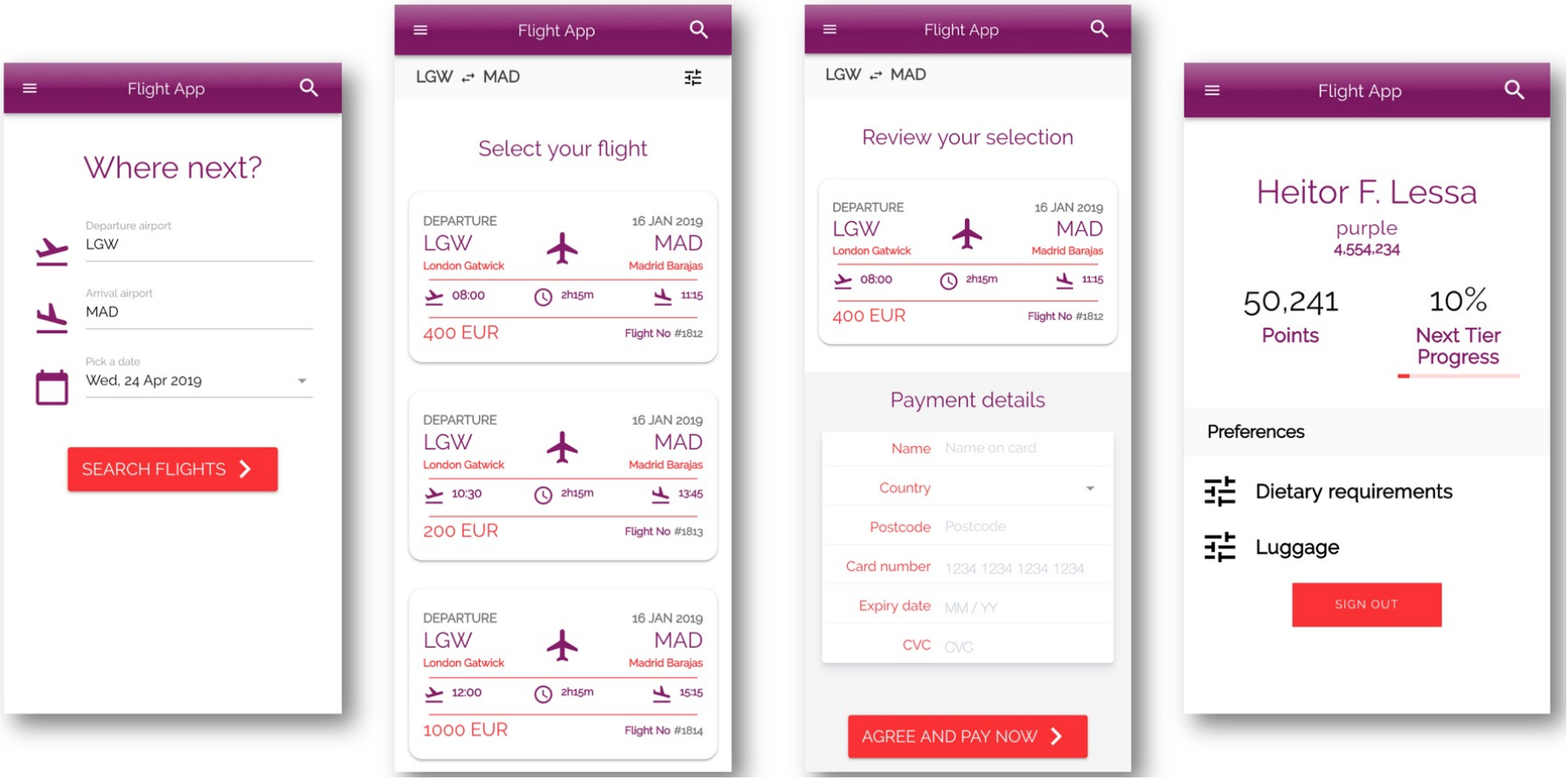
Cloud Software Architecture



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

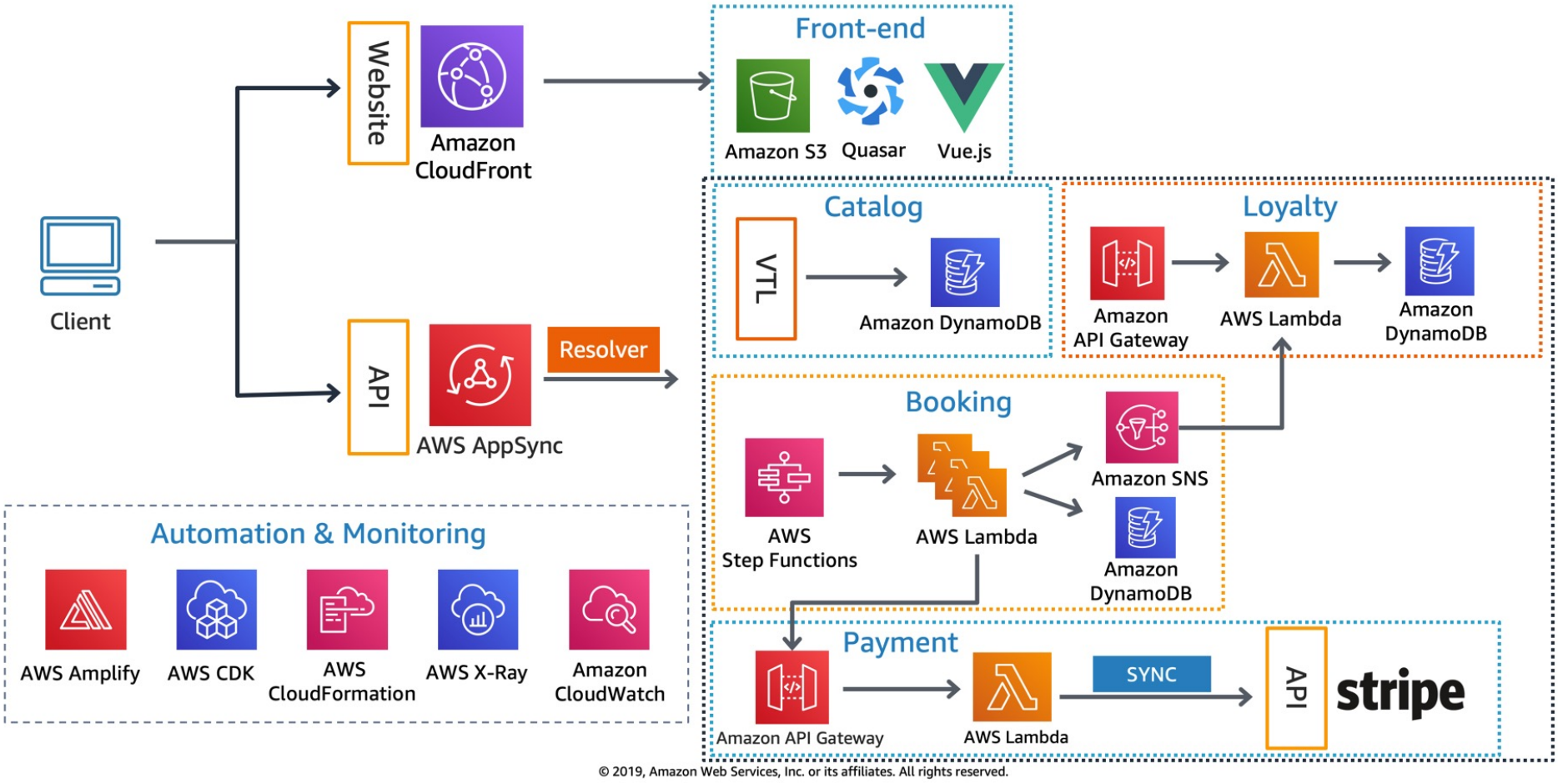

Amazon SNS


AWS Step Functions

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

AWS Serverless Airline Booking

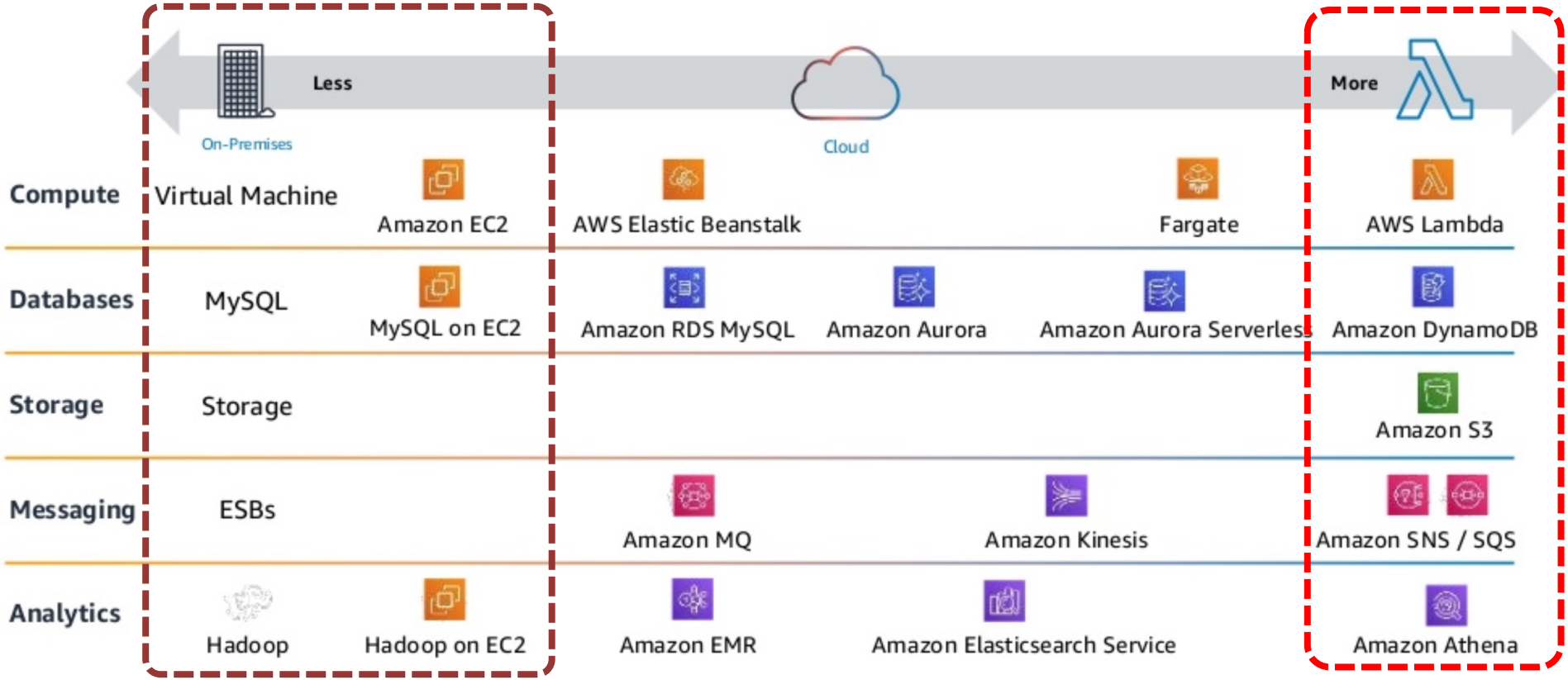
High level infrastructure architecture



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

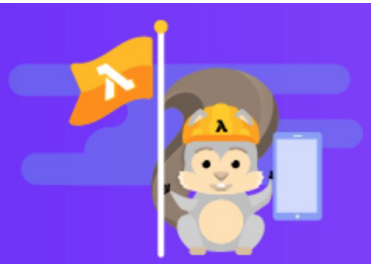
Build a Serverless Web Application

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

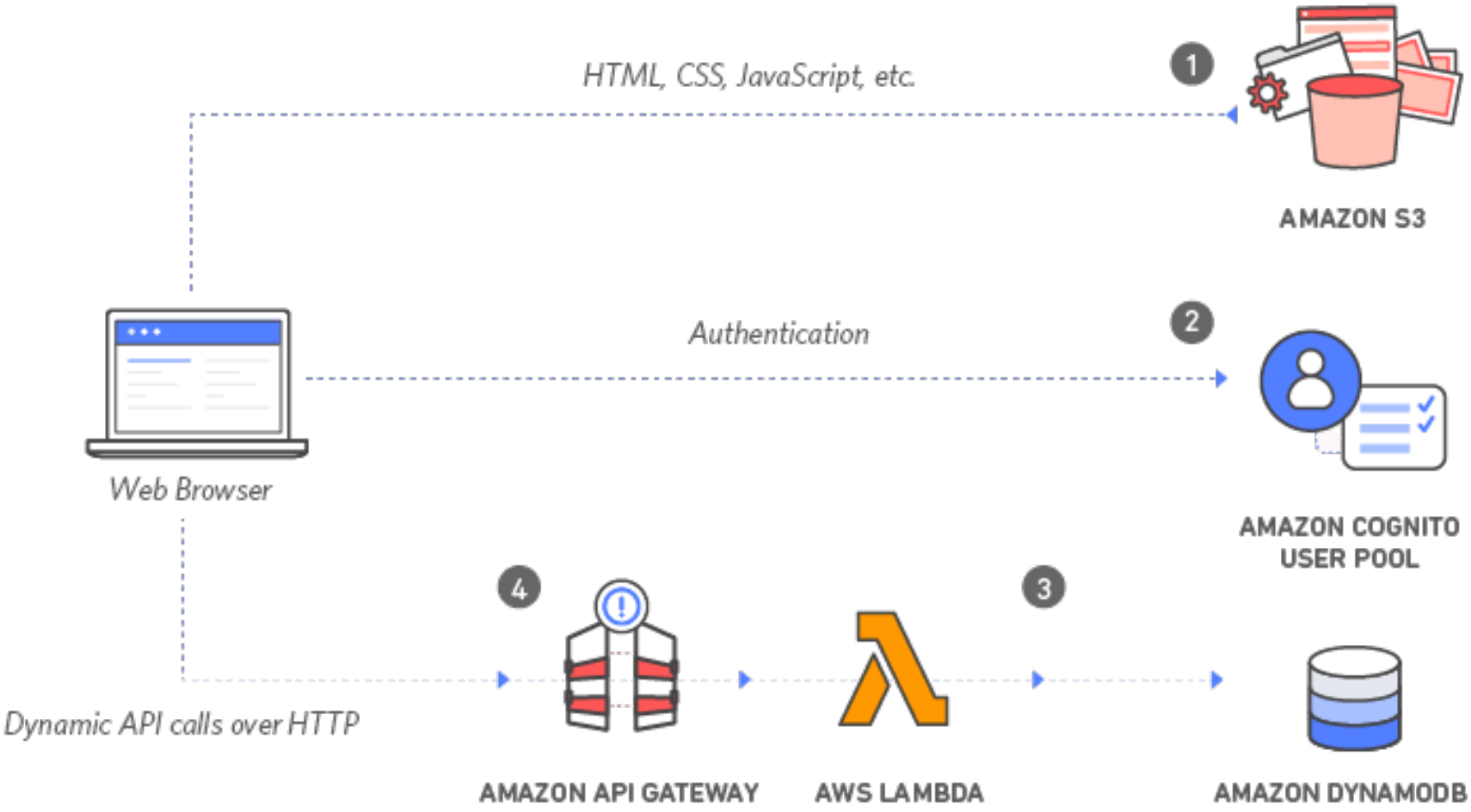
Cost to complete: Each service used in this architecture is eligible for the AWS Free Tier. If you are outside the usage limits of the Free Tier, completing this tutorial will cost you less than $0.25*.

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

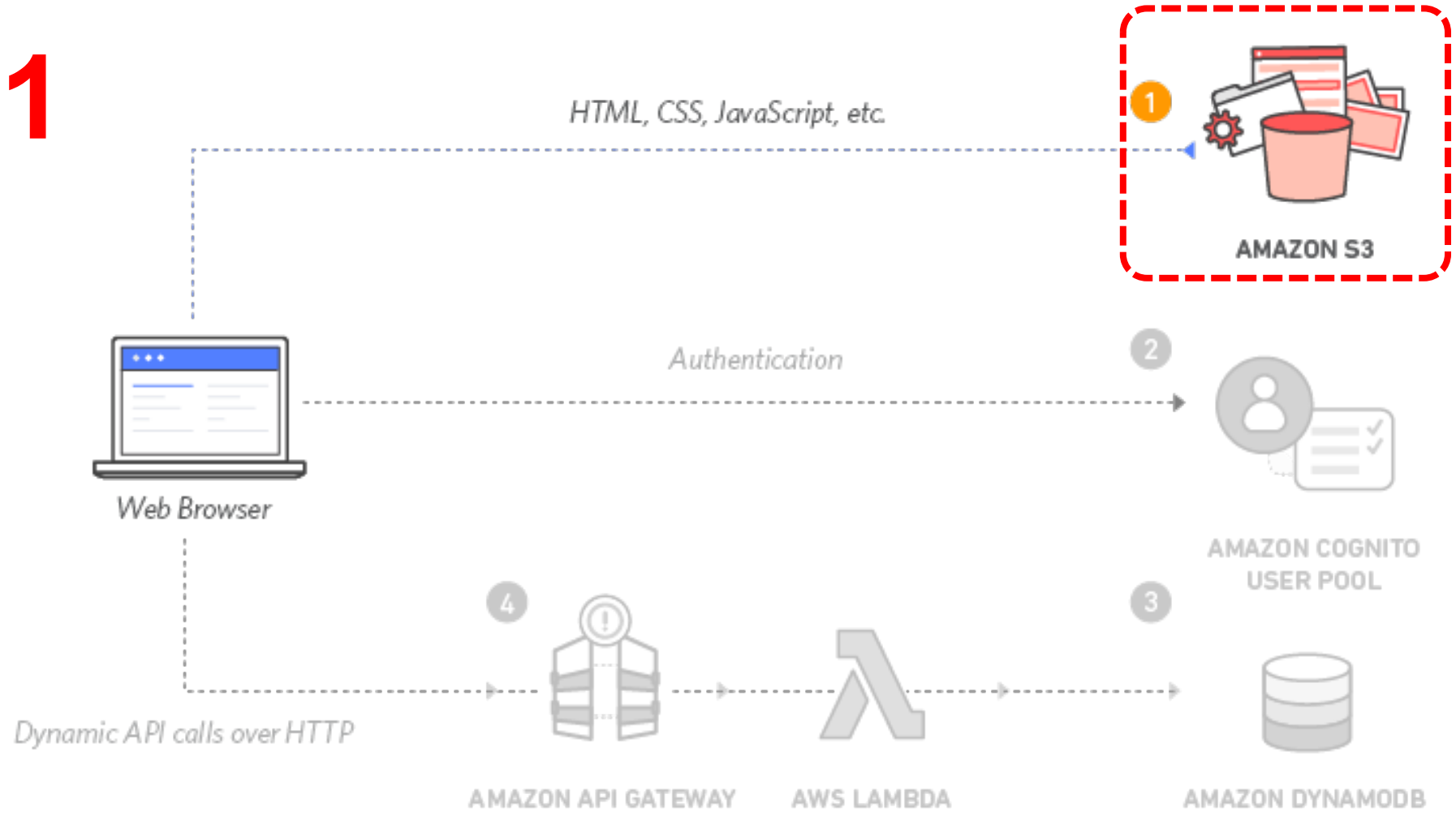


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

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

1

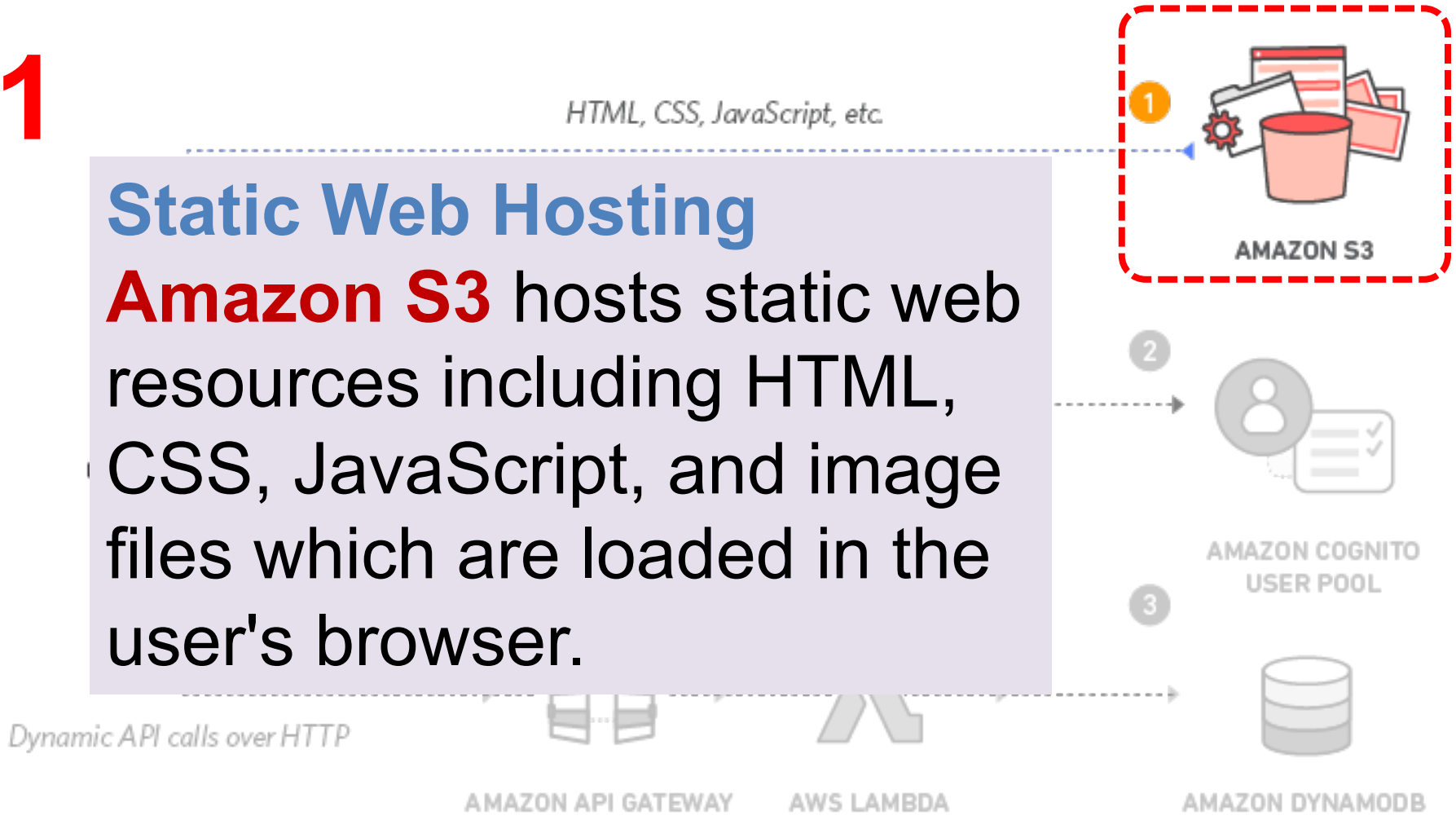


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

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

1



Static Web Hosting

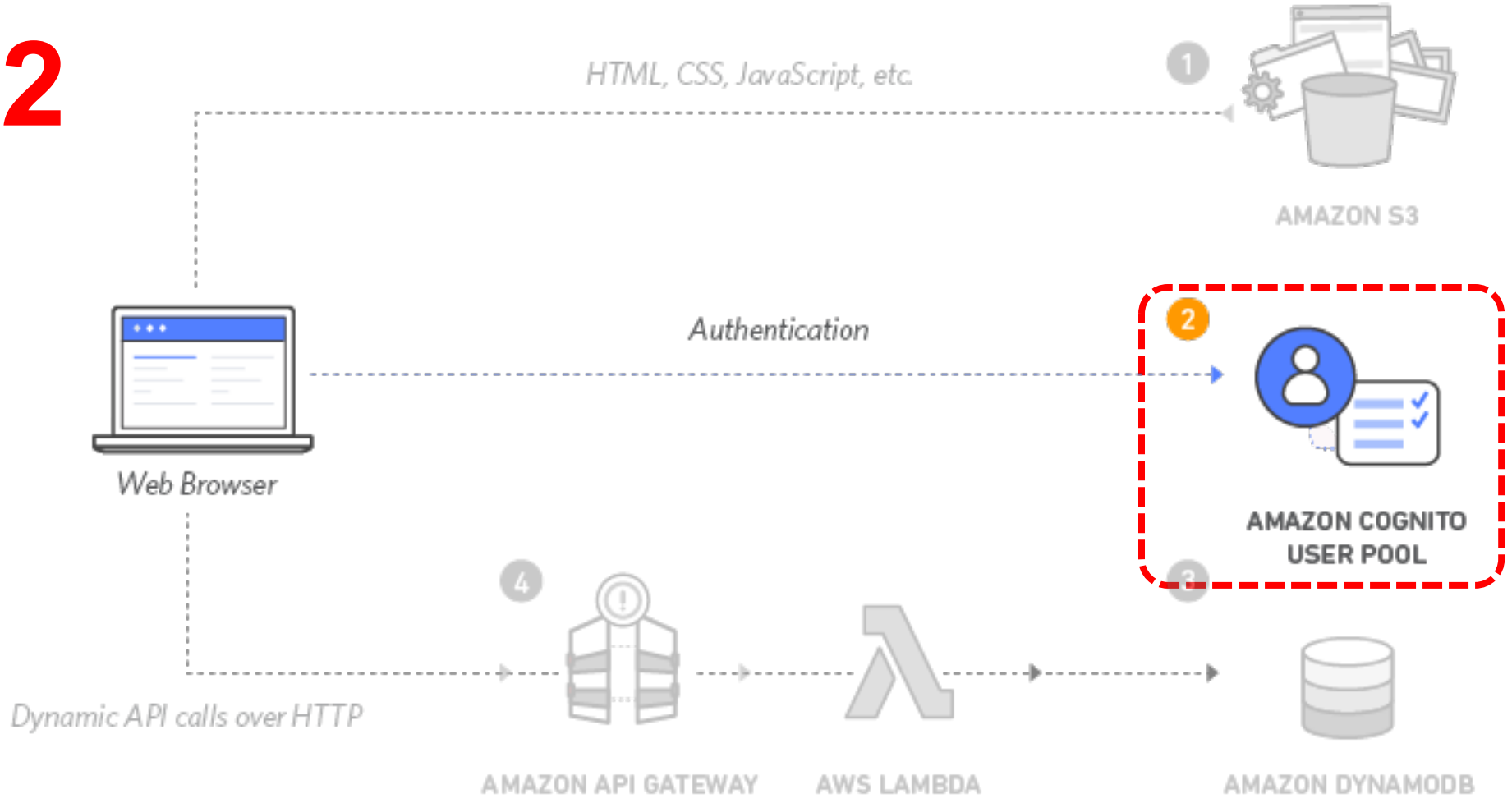
Amazon S3 hosts static web resources including HTML, CSS, JavaScript, and image files which are loaded in the user's browser.

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

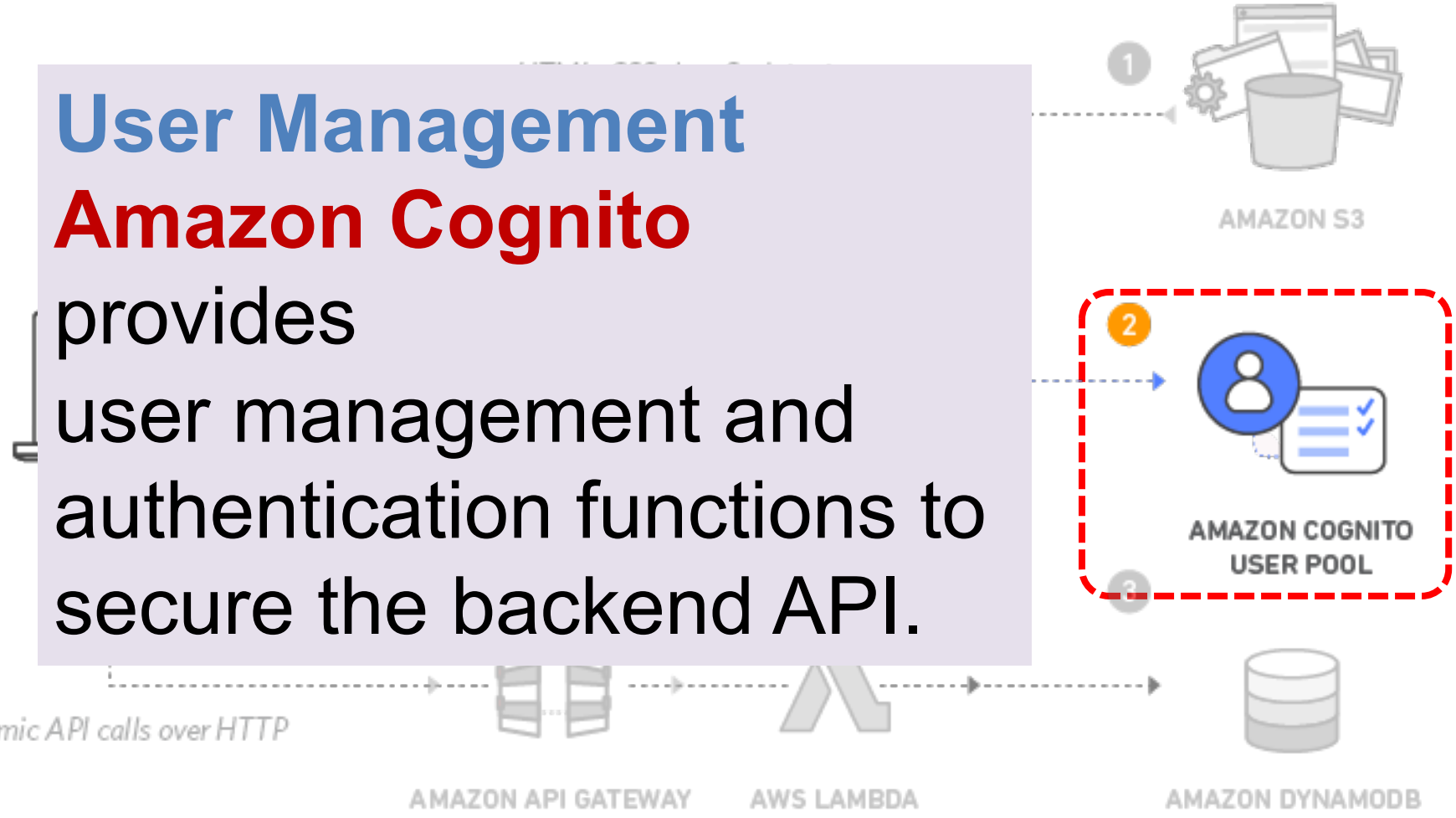
with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

2

User Management

Amazon Cognito

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

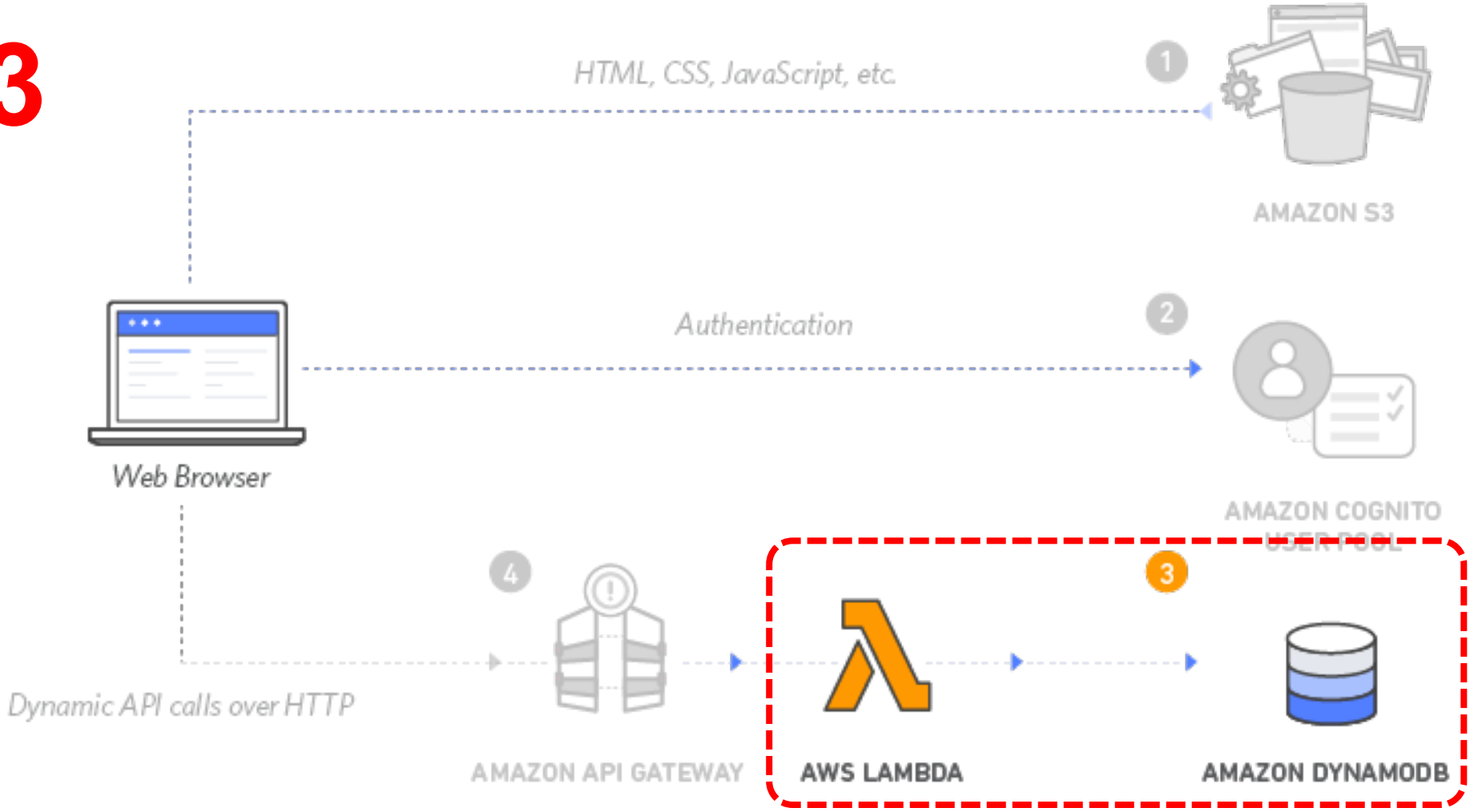


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

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/

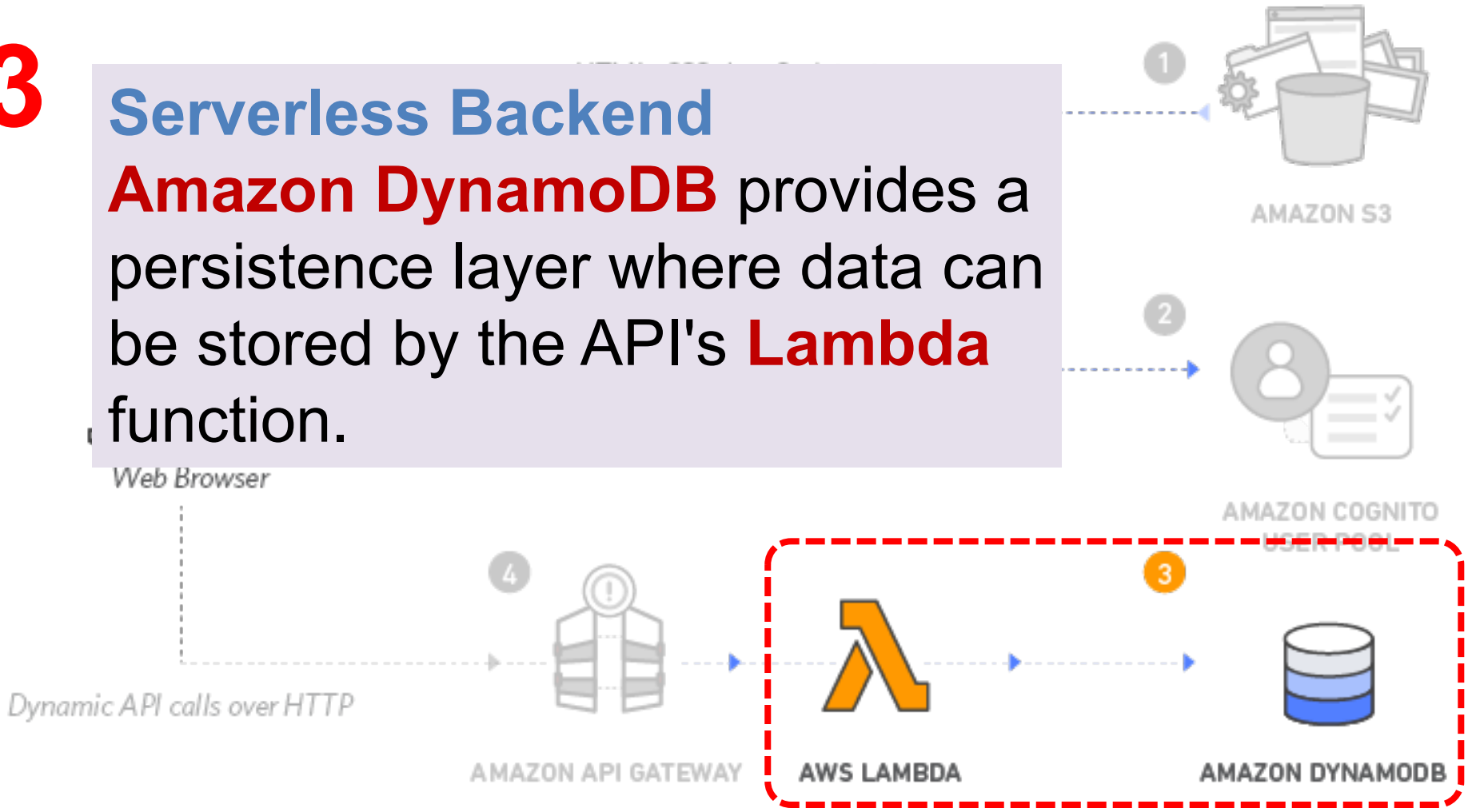
Build a Serverless Web Application

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.

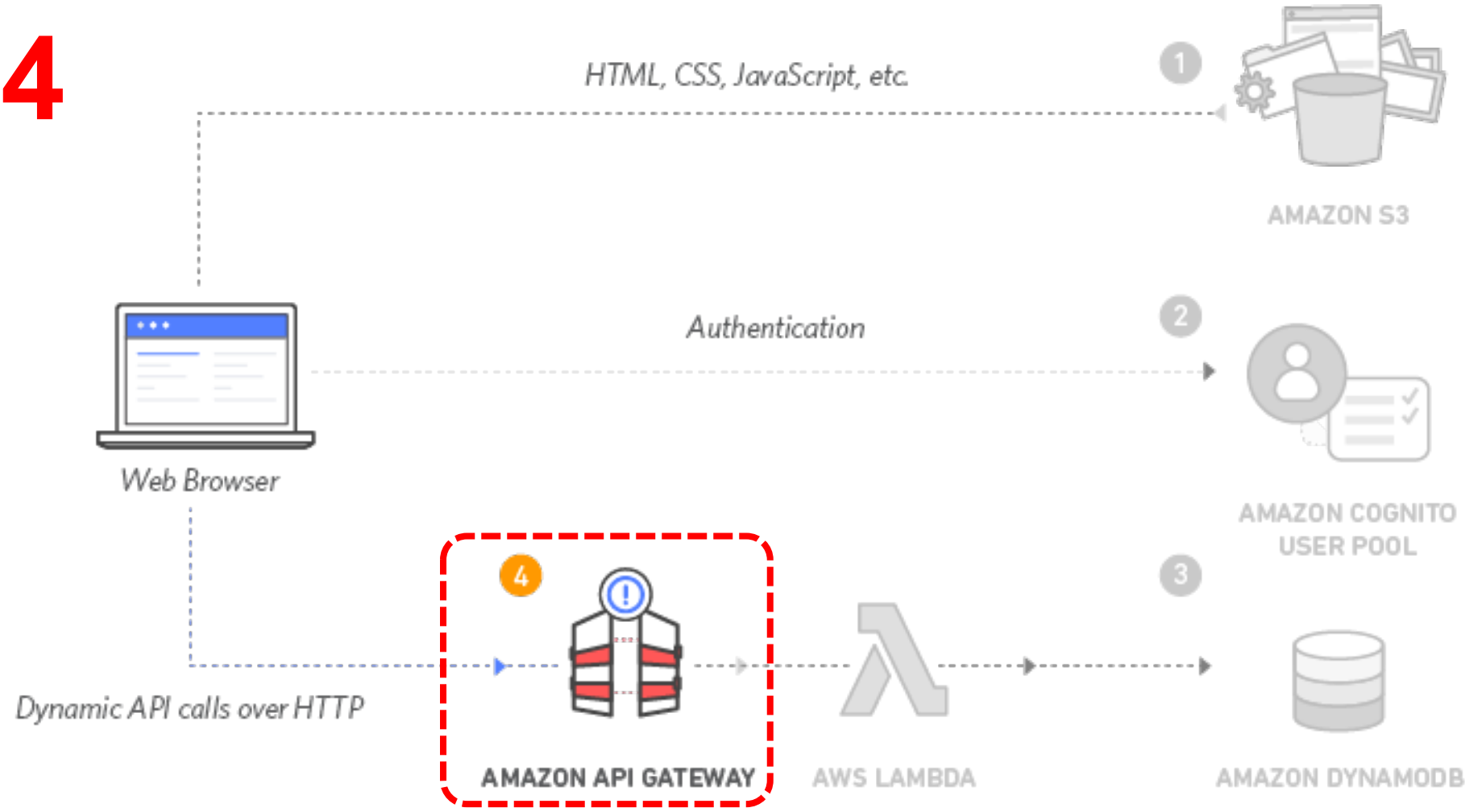


Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

Build a Serverless Web Application

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

4



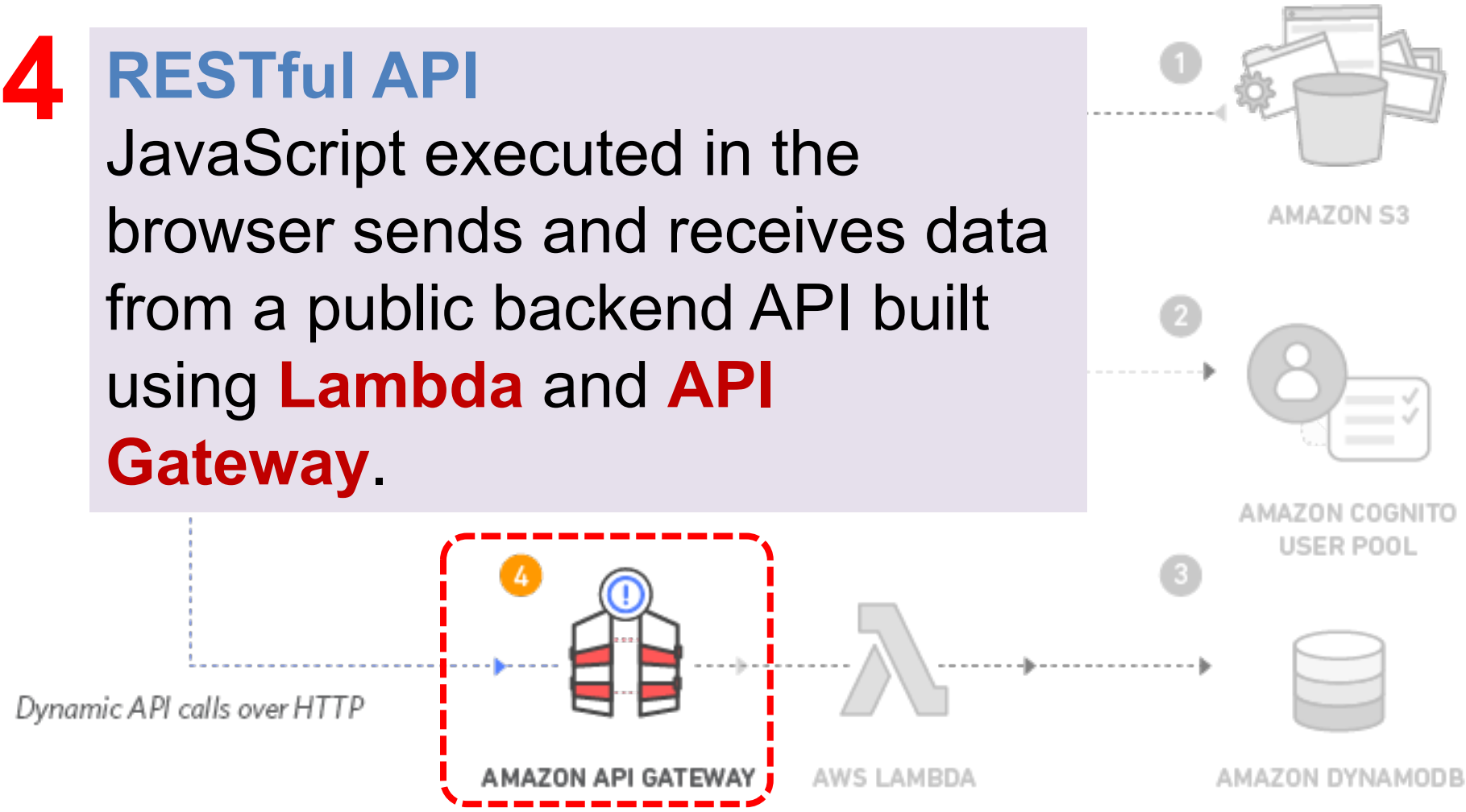
Source: https://aws.amazon.com/getting-started/projects/build-serverless-web-app-lambda-apigateway-s3-dynamodb-cognito/

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

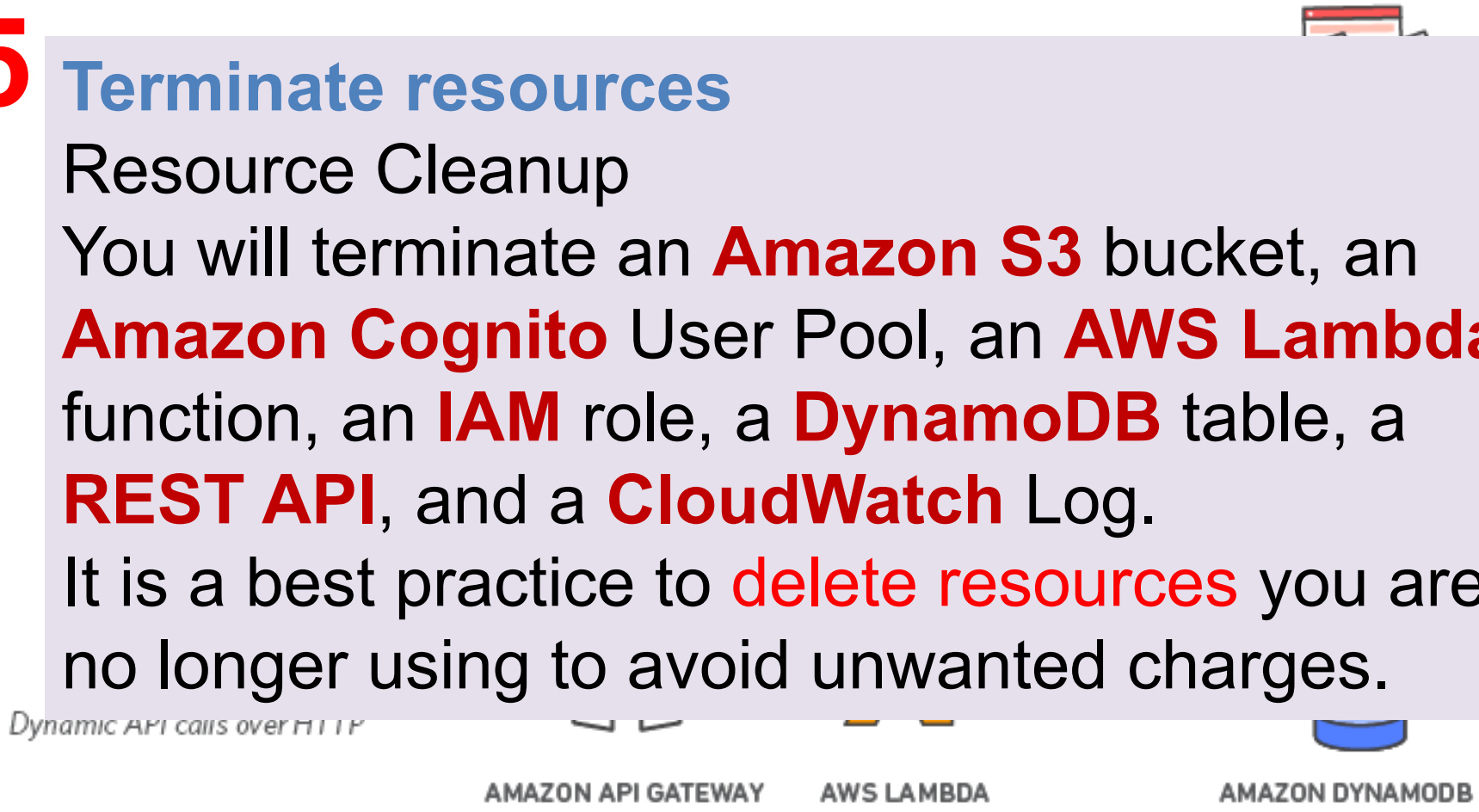
5

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/

Summary

- Cloud Computing and Cloud Software Architecture
- AWS Certified Cloud Practitioner (CLF-C01)
- AWS Certified Solutions Architect – Associate (SAA-C02)
- Web Application with AWS Core Services
- AWS Serverless Architecture
- Build a Serverless Web Application with Amazon S3, AWS Lambda, Amazon API Gateway, Amazon DynamoDB, and Amazon Cognito

References

- Ben Piper and David Clinton (2019), **AWS Certified Solutions Architect Study Guide: Associate SAA-C01 Exam**, 2 edition, Sybex, 2019
- **AWS Cloud Practitioner Essentials (Second Edition)**
  - https://aws.amazon.com/training/course-descriptions/cloud-practitioner-essentials/
- **AWS Certified Cloud Practitioner**
  - https://aws.amazon.com/certification/certified-cloud-practitioner/
- **AWS Certified Solutions Architect – Associate**
  - https://aws.amazon.com/certification/certified-solutions-architect-associate/
- **AWS Academy Cloud Foundations (AWS ACF)**, AWS Academy
- **AWS Academy Cloud Architecting (AWS ACA)**, AWS Academy

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