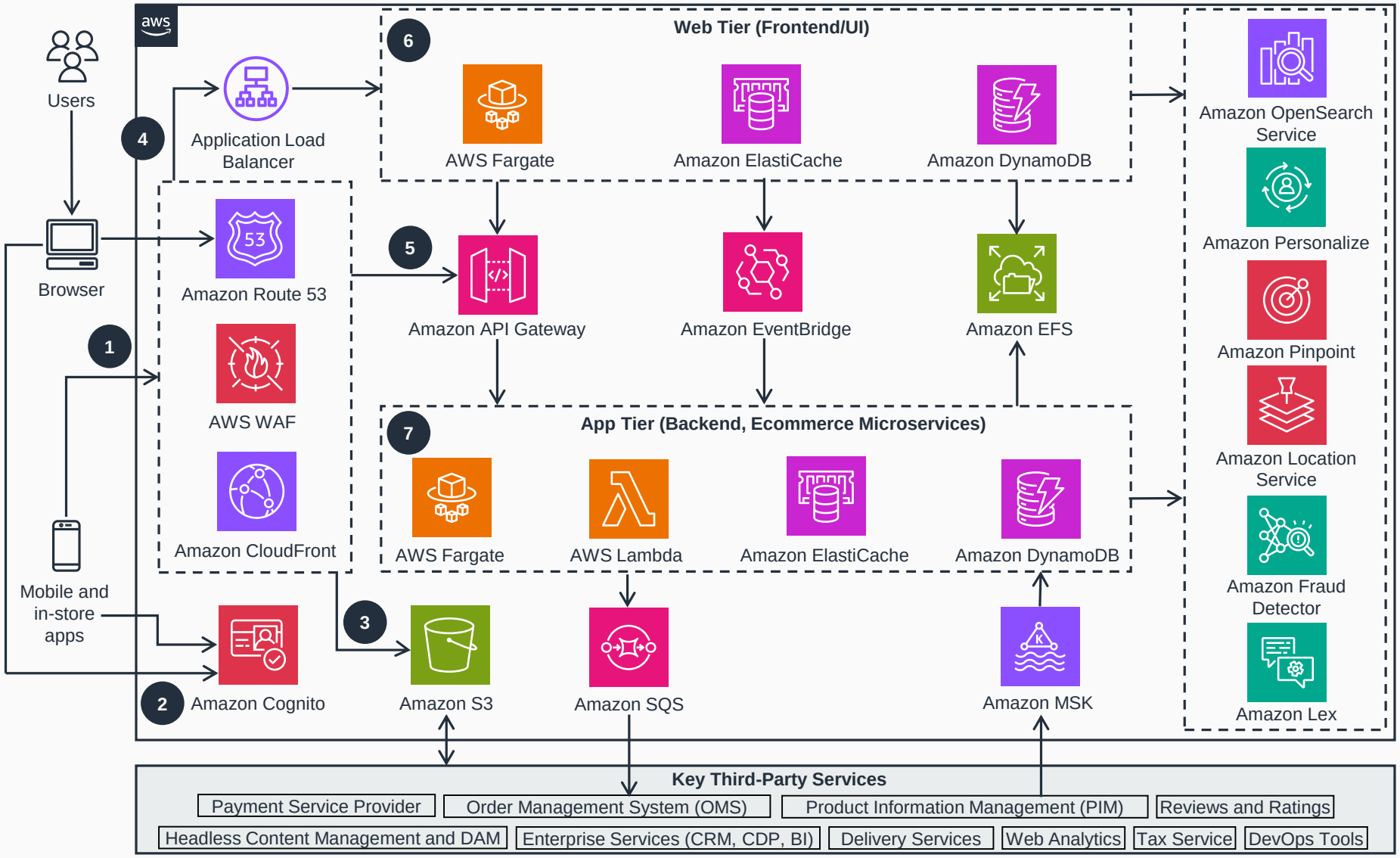
Guidance for Web Store on AWS

This architecture diagram shows how to build an ecommerce web application on AWS. This slide shows steps 1-7.



Key Third-Party Services

Payment Service Provider | Order Management System (OMS) | Product Information Management (PIM) | Reviews and Ratings

Headless Content Management and DAM | Enterprise Services (CRM, CDP, BI) | Delivery Services | Web Analytics | Tax Service | DevOps Tools

1. Customers access the web application through different channels. **Amazon Route 53** enables frontend clients to resolve the website hostname to **Amazon CloudFront**, which routes the web requests to origin servers and caches the static content and assets served from **Amazon Simple Storage Service (Amazon S3)** and origin servers. It also secures the application traffic using **AWS WAF (Web Application Firewall)**, which helps protect the application against common exploits and bots.

2. The web application uses **Amazon Cognito** to perform authentication and authorization of backend APIs.

3. **Amazon S3** is a highly available and durable object storage service that stores and serves the static assets, such as images and videos.

4. **Application Load Balancer** serves the frontend web requests by automatically distributing the incoming traffic across multiple Web Tier targets, deployed in multiple Availability Zones.

5. **Amazon API Gateway** is a fully managed service that interfaces the backend microservices to access data and execute the business logic. These microservices are exposed as restful APIs for consumption by the Web Tier and the mobile app.

6. Ecommerce Frontend/Web Tier is a headless and responsive web UI, built on your choice of frontend technologies (like ReactJS, VueJS, AngularJS, NodeJS, etc.) and deployed on **AWS Fargate**. This Web Tier uses **Amazon ElasticCache** to cache static content and orchestrated backend API responses. The **Amazon DynamoDB** table persists the user sessions and frontend application configurations.

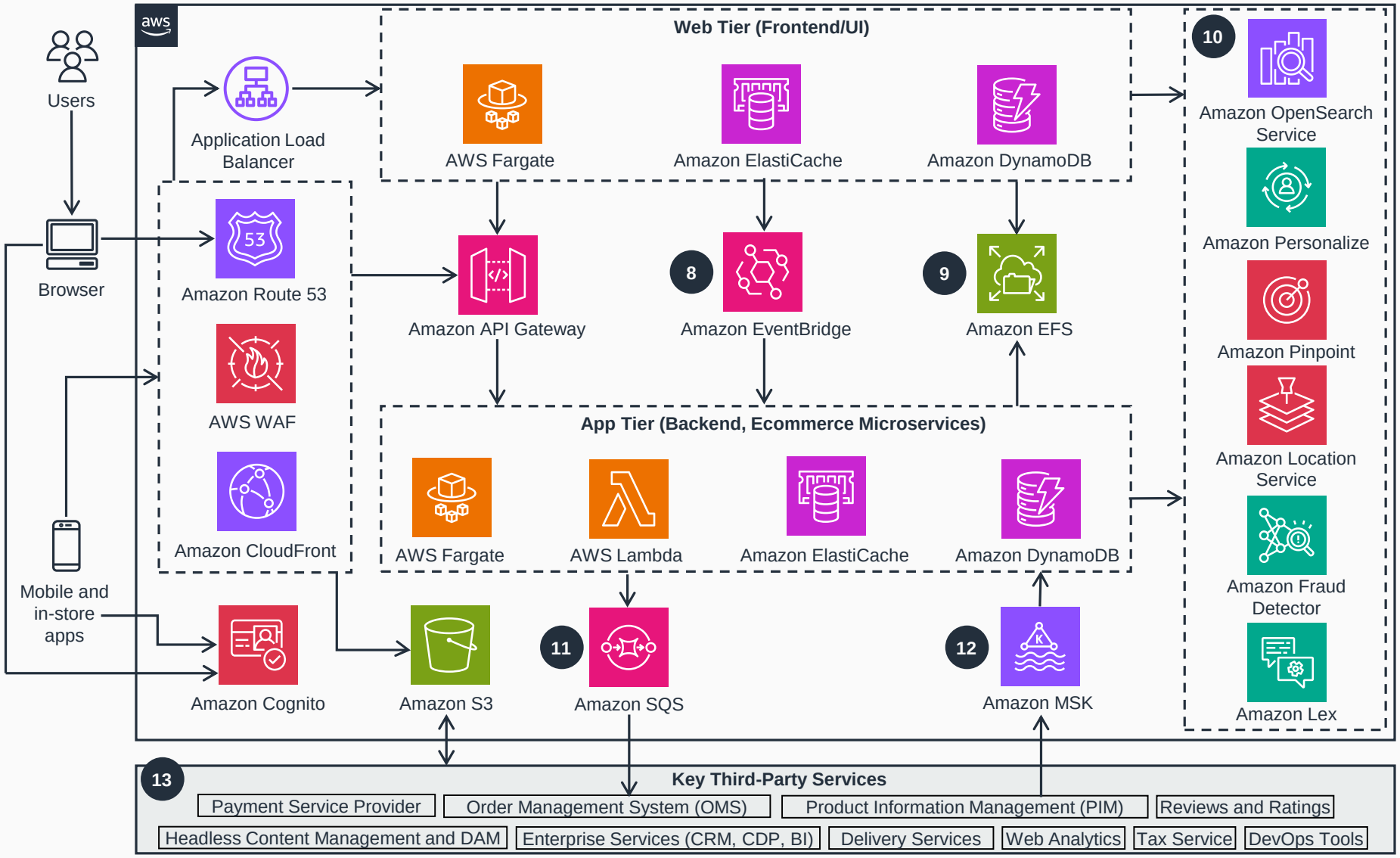
7. Ecommerce backend services (App Tier), a set of stateless restful microservices, are built to access the data and execute specific business logic, such as OrderMx for cart and checkout and PaymentMx for handling payments. These microservices are deployed on **Fargate** and **Lambda**. **DynamoDB** in the App Tier provides the ecommerce application data store, storing data on products, customers, and customer transactions. **DynamoDB Accelerator (DAX)** caches the database query results, while **ElastiCache** caches the transformed response of individual microservices.

Reviewed for technical accuracy January 19, 2024

AWS Reference Architecture

Guidance for Web Store on AWS

This architecture diagram shows how to build an ecommerce web application on AWS. This slide shows steps 8-13.



Reviewed for technical accuracy January 19, 2024

AWS Reference Architecture

8 **Amazon EventBridge** is a serverless event bus used by both the Web and App Tiers to emit events that will be consumed asynchronously by the microservices in the App Tier and other supported sources to perform specific actions. As an example, a Customer Consent sign-up action on the frontend invokes an event to **EventBridge.** In response, **Eventbridge** invokes multiple backend microservices to execute independent business logic and update isolated applications and data stores such as **DynamoDB**, **Amazon Pinpoint**, and third-party customer management system (CMS) and marketing systems.

9 Both the Web and App Tiers use **Amazon Elastic File System (Amazon EFS)** to share common code and files such as properties and configurations, JavaScript, CSS, and JSON templates.

10 A set of AWS services delivers core ecommerce business capabilities. **Amazon Open Search Service** provides intelligent search and filtering of products. **Amazon Personalize** offers artificial intelligence and machine learning (AI/ML)-powered product recommendations. **Amazon Pinpoint** supports marketing campaigns and push notifications. **Amazon Location Service** provides maps, a store locator, and delivery tracking. **Amazon Fraud Detector** detects fraudulent transactions, such as malicious attempts of customer log-in and payment. **Amazon Lex** deploys an AI/ML-powered chatbot.

11 **Amazon Simple Queue Service (Amazon SQS)** first in, first out (FIFO) publishes the order messages for the orders placed by customers using the ecommerce application. These are published to the Order Management System (OMS) for processing and fulfillment.

12 **Amazon Managed Streaming for Apache Kafka (Amazon MSK)** performs extract, transform, and load activities (ETL) at scale, such as importing data feeds into ecommerce data stores. These include product and catalog data from product information management (PIM), near real-time inventory, and order status updates from supply chain systems.

13 Key third-party services integrate with the ecommerce application to deliver business capabilities.