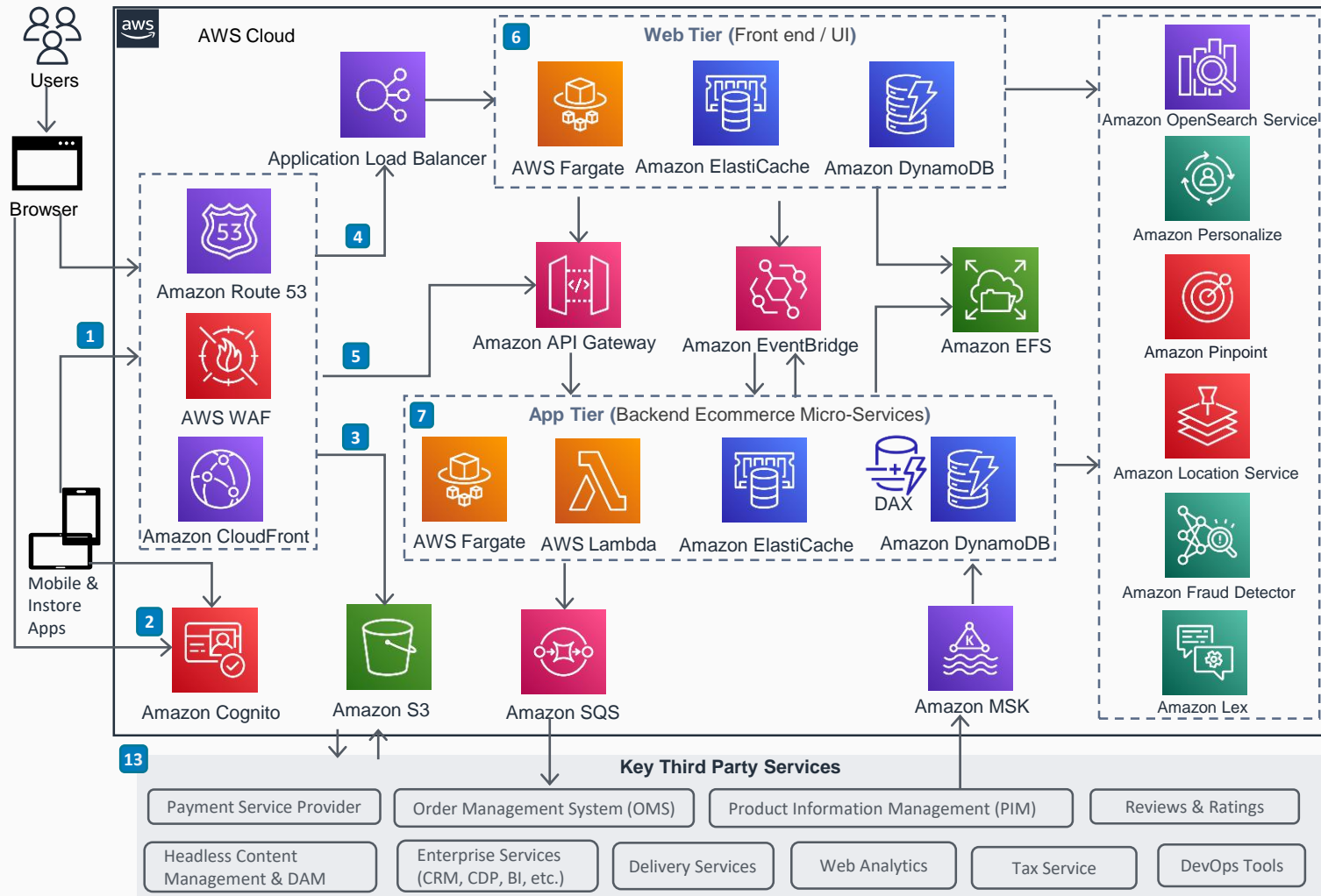


# Guidance for Web Store on AWS

The architecture provides a pattern to build an ecommerce web application on AWS with decoupled UI and back-end service layers, and using AWS native services to implement the core capabilities such as search, personalization, marketing, fraud detection, customer authentication, location services and chat bots to deliver an optimal customer experience. We start by looking at the data flowing into the Web store and then look at how it is used.

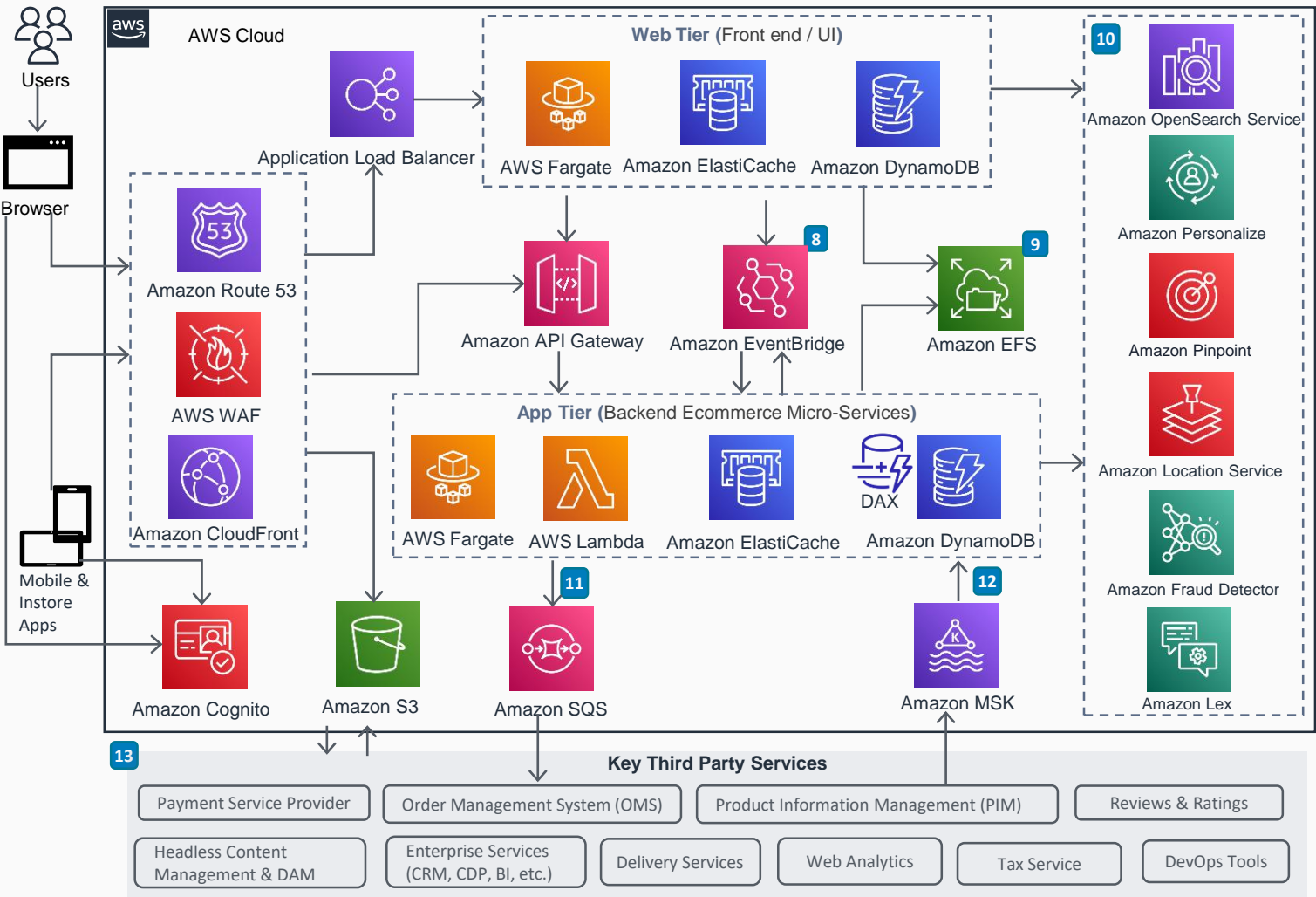


- 1 Customers access the web application through different channels. **Amazon Route 53**, the Domain Name System (DNS) enables front-end clients to resolve the website hostname to the AWS content delivery network (**Amazon CloudFront**). **Amazon CloudFront** takes care to route the web requests to origin servers, caches the static content & assets served from Amazon S3 and Origin servers. It also secures the application traffic using **AWS WAF** (a web application firewall), which helps protect the application against common exploits and bots.
- 2 The web application uses **Amazon Cognito** to perform authentication (user sign-up, sign-in) and authorization of backend APIs.
- 3 **Amazon Simple Storage Service (Amazon S3)** is a highly available and durable object storage service that stores and serves the static assets (images and videos).
- 4 **Application Load Balancer (ALB)** serves the front-end 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 micro-services to access data and execute the business logic. These micro-services are exposed as Restful APIs for consumption by 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** (serverless compute service). This Web Tier uses **Amazon ElastiCache** to cache static content and orchestrated backend API responses; and **Amazon DynamoDB** table to persist the user sessions and frontend application configurations (e.g. Feature flags).
- 7 **eCommerce Backend Services (App Tier)** is a set of stateless Restful micro-services built to access the data and also execute specific business logic (such as OrderMx for cart and checkout as well as PaymentMx for handling payments). These micro-services are deployed on the serverless compute services. **AWS Fargate** and **AWS Lambda**. **Amazon DynamoDB** in the App Tier provide the ecommerce application data store. It holds products, customer and customer transaction data (such as orders and shopping carts). **DynamoDB DAX** caches the database query results, while **Amazon ElastiCache** caches the transformed response of individual microservices.



# Building the rest of the Web Store

In the second part of this architecture, we show how the serverless event bus is used by both Web and App Tiers to emit events that will be consumed asynchronously by the micro-services in the App Tier and/or other supported sources to perform specific actions. It also provides detail on the set of AWS services that deliver the core ecommerce business capabilities and explains where they fit.



8

**Amazon EventBridge** is a serverless event bus used by both Web and App Tiers to emit events that will be consumed asynchronously by the micro-services in the App Tier and/or other supported sources to perform specific actions. **As an example**, a Customer Consent sign-up action on the front-end triggers an event to **Amazon EventBridge**, which in response invokes multiple backend micro-services to execute independent business logic and update isolated applications/datastores such as **DynamoDB**, **Amazon Pinpoint** as well as third-party CMS and marketing systems.

9

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

10

A set of AWS services delivering core ecommerce business capabilities. **Amazon Open Search** for Intelligent search and filtering of products, **Amazon Personalize** for AI/ML powered product and offer recommendations, **Amazon Pinpoint** for marketing campaigns and push notifications, **Amazon Location Services** for Maps, store locator, delivery tracking, etc. **Amazon Fraud Detector** to detect fraudulent transactions (such as malicious attempts of customer login and payment), **Amazon Lex** for AI/ML powered chatbot.

11

**Amazon Simple Queue Service (Amazon SQS)** first in, first out (FIFO) is used to publish the order messages for the orders placed by customers using the eCommerce application, to the Order Management System (OMS) for processing and fulfillment.

12

**Amazon Managed Streaming for Apache Kafka (MSK)** is used to perform the ETL (Extract, Transform and Load) activities at scale (such as importing data feeds into eCommerce data stores. These include data feeds such as product/catalog data from the PIM, near real-time inventory and order status updates from Supply Chain Systems.

13

Some of the **key third party services** and applications, which integrate with the ecommerce application to deliver business capabilities.