



Modern Cloud-Native Payment Systems

Payment System Interface Modernization on AWS



Payment System Interface Modernization on AWS: Modern Cloud-Native Payment Systems

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Payment System Interface Modernization on AWS **i**

Payment System Interface Modernization on AWS Diagram 1

Further reading 2

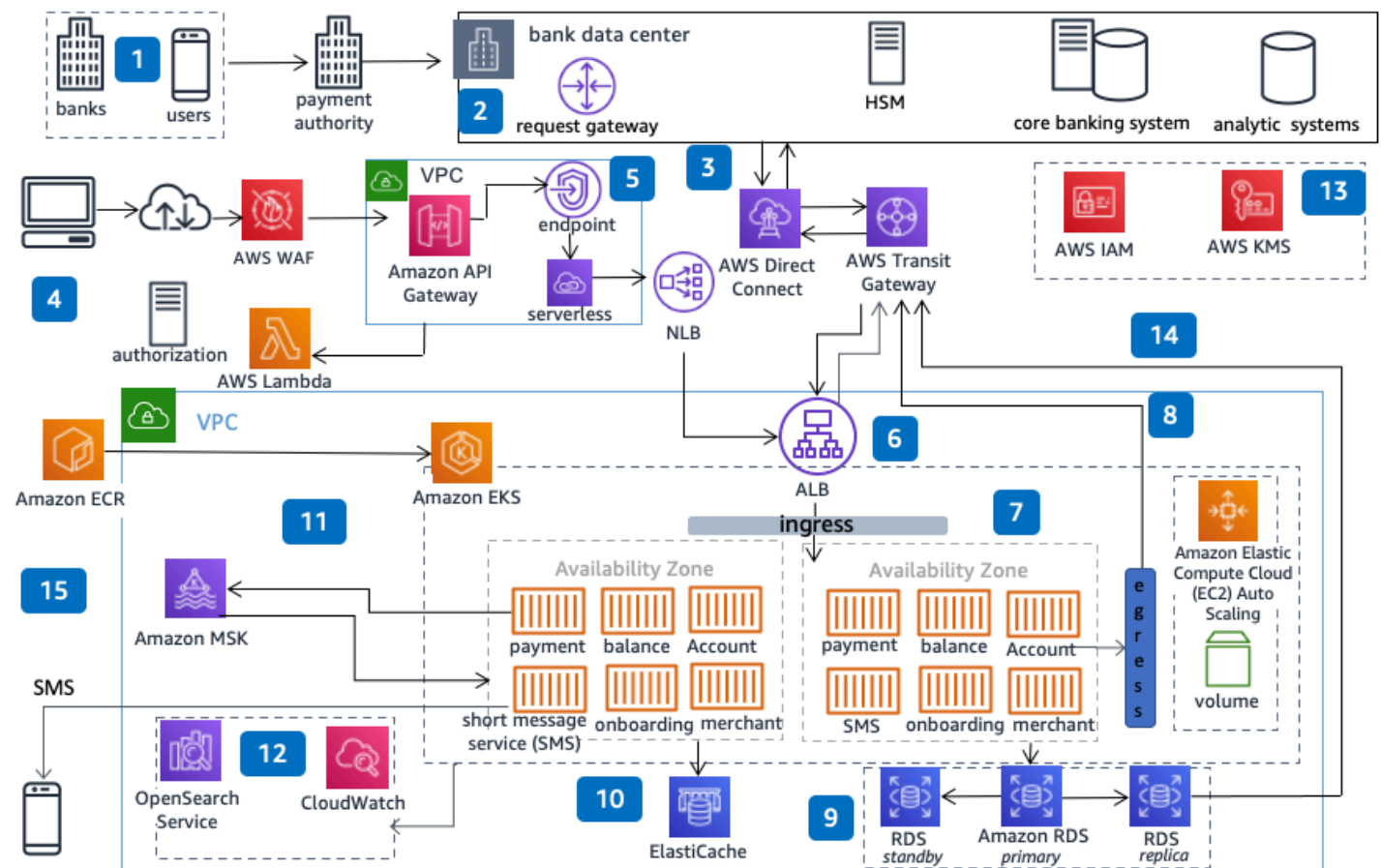
Diagram history 3

Payment System Interface Modernization on AWS: Modern Cloud-Native Payment Systems

Publication date: April 6, 2022 ([Diagram history](#))

This architecture shows you how to build a microservices-based payment system to handle scale and optimized performance with improved container-based deployment, and using API and event-based models for handling different channels in payment.

Payment System Interface Modernization on AWS Diagram



1. Users initiate transactions to payment interface system using mobile apps integrated with the payment system, or through bank-provided apps.
2. The request receiver accepts the request at the bank data center.

3. The request is routed to payment interface services in the AWS network from the bank using **Direct Connect**.
4. A request from a merchant or other client arrives directly from the internet.
5. The request passes through **AWS WAF** and is received through the **Amazon API Gateway**, which routes requests to services using a VPC endpoint and **Network Load Balancer (NLB)**.
6. In AWS, the request for payment interface services is received by the **Amazon Elastic Kubernetes Service (Amazon EKS)** cluster's **Application Load Balancer (ALB)** ingress controller.
7. Ingress rules redirect the request to a targeted microservice for purposes such as payment, balance, virtual payment address (VPA), merchant, account, and so on.
8. Payment interface services reach the bank hardware security model (HSM), core banking systems hosted inside the bank using **AWS Direct Connect**.
9. Payment interface services perform create, read, update, and delete (CRUD) operations in the **Amazon Relational Database Service (Amazon RDS)** unified payment interface (UPI) database.
- 10 Payment interface services maintain sessions in **Amazon ElastiCache (Redis OSS)**.
- 11 Payment interface services communicate with each other with an event-driven model using topics in **Amazon Managed Streaming for Apache Kafka (Amazon MSK)**.
- 12 Payment interfaces microservices observability using **Amazon OpenSearch Service** and **Amazon CloudWatch**.
- 13 Use **AWS Identity and Access Management (IAM)**, **AWS Key Management Service (AWS KMS)**, and **AWS Secrets Manager** to ensure the roles-based access, securing credentials, and data.
- 14 Payment interface near real-time transaction data is consumed by other systems in the bank DC using an **Amazon Relational Database Service (Amazon RDS)** replica over **Direct Connect**.
- 15 Payment interface microservices container images are maintained inside **Amazon Elastic Container Registry (Amazon ECR)**.

Further reading

For additional information, refer to

- [AWS Architecture Icons](#)
- [AWS Architecture Center](#)
- [AWS Well-Architected](#)

Diagram history

To be notified about updates to this reference architecture diagram, subscribe to the RSS feed.

Change	Description	Date
Initial publication	Reference architecture diagram first published.	April 6, 2022

Note

To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.