



Architecture diagrams for financial services

Open Banking on AWS



Open Banking on AWS: Architecture diagrams for financial services

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

Overview **1**

Open Banking overview diagram 1

Further reading 2

Diagram history 2

Networking and security **4**

Open Banking networking and security diagram 4

Further reading 5

Diagram history 5

Services and operations **7**

Open Banking services and operations diagram 7

Further reading 8

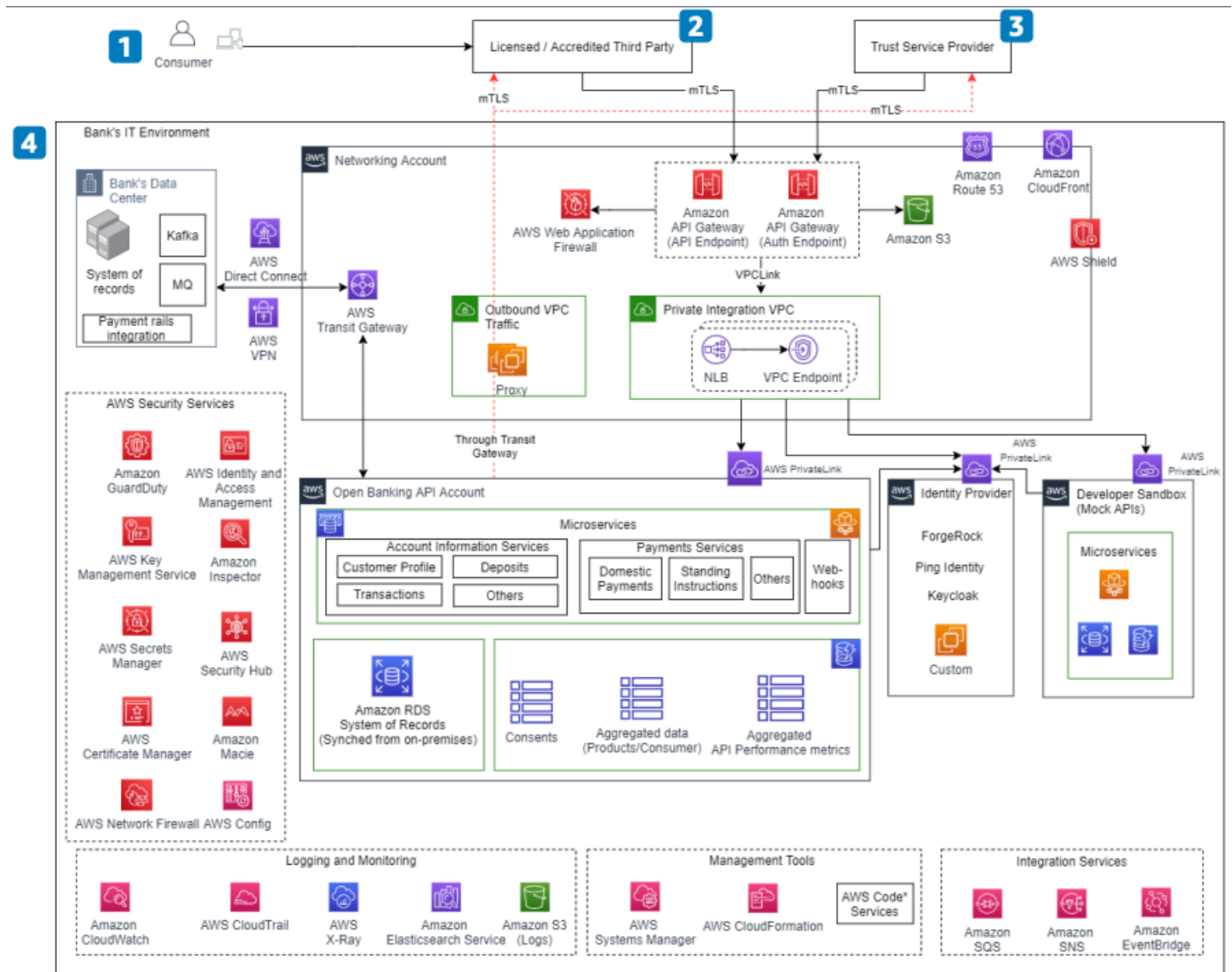
Diagram history 9

Open Banking on AWS: Overview

Publication date: **September 7, 2021** ([Diagram history](#))

With this architecture, you can open APIs for authorized third parties and implement Open Banking regulations. Third parties access consumer data (account balances, transactions, statements) or initiate payment submissions with customer consent. This supports use cases such as spend analysis, credit decisioning, and ecommerce payments.

Open Banking overview diagram



The following steps describe the Open Banking ecosystem participants:

1. Access the licensed or accredited third-party application and provide consent for the third party to access consumer data or initiate a payment submission request.
2. Use third parties as authorized institutions that provide value-added services on top of regular banking needs (accounts information, payments). This approach supports use cases such as spend analysis, credit decisioning, and ecommerce payments.
3. Validate the authenticity of banks and third parties through a Trust Service Provider (TSP) authorized by a supervisory government body. The TSP issues digital certificates to third parties.
4. Connect the bank IT environment, including AWS and data center components.

Further reading

For additional information, see the following resources:

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

Diagram history

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

Change	Description	Date
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021

** RSS subscription requirement**

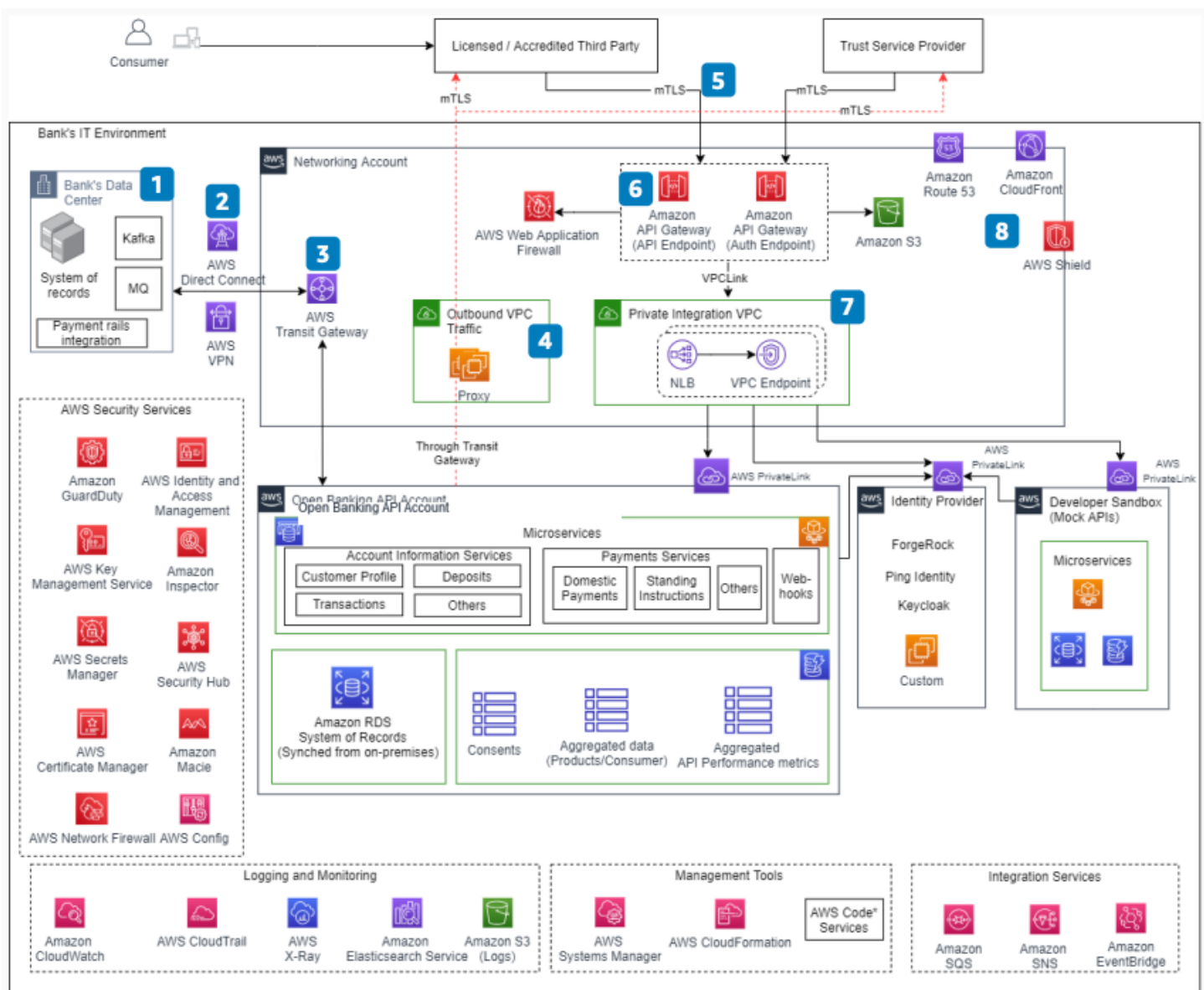
To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Open Banking on AWS: Networking and security

Publication date: **September 7, 2021** ([Diagram history](#))

With this architecture, you can connect on-premises core banking systems to AWS and expose Open Banking APIs securely. The solution uses [AWS Direct Connect](#) for connectivity, [AWS Transit Gateway](#) for network hub management, and [Amazon API Gateway](#) for API management with [AWS WAF](#) integration.

Open Banking networking and security diagram



The following steps describe the networking and security components for this architecture:

1. Send and receive all new and updated transactions between core banking systems and AWS by using streaming technologies such as Apache Kafka and message queue (MQ) mechanisms.
2. Connect the bank data center to AWS by using a combination of AWS Direct Connect and [AWS Site-to-Site VPN](#). Use two diverse AWS Direct Connect connections for maximum resiliency.
3. Use AWS Transit Gateway as the central hub on AWS to manage interconnectivity between workloads running in different AWS accounts. Share the AWS Direct Connect and VPN connection with other workloads in the bank.
4. Provide secure outbound access from an outbound Amazon VPC through a proxy.
5. Authenticate accredited third parties and provide access tokens through mutual TLS (mTLS) for transport layer security.
6. Expose Open Banking APIs and Authorization APIs through Amazon API Gateway. Integrate AWS WAF with Amazon API Gateway for web protection.
7. Store public certificates of clients in [Amazon Simple Storage Service](#) as a trust store for validating requests. Validate the authenticity of third parties against a Trust Service Provider (TSP).
8. Connect Amazon API Gateway to the private subnets hosting microservices in other AWS accounts through a private integration Amazon VPC and AWS PrivateLink.
9. Provide traffic management and domain name resolution by using [Amazon Route 53](#). Deliver static data through [Amazon CloudFront](#) with [AWS Shield](#) protection.

Further reading

For additional information, see the following resources:

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

Diagram history

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

Change	Description	Date
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021

 RSS subscription requirement

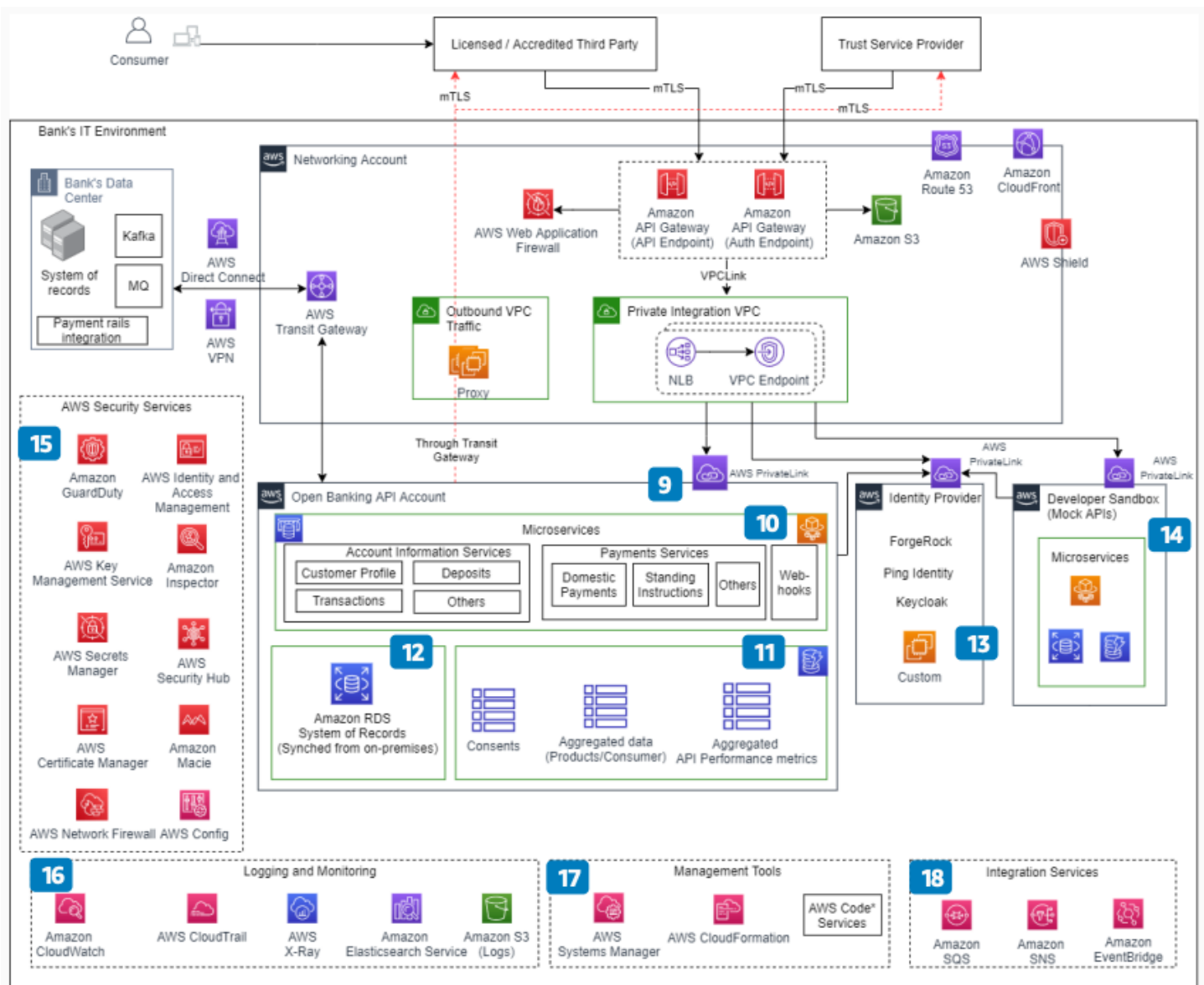
To subscribe to RSS updates, you must have an RSS plugin enabled for the browser you are using.

Open Banking on AWS: Services and operations

Publication date: September 7, 2021 ([Diagram history](#))

With this architecture, you can implement Open Banking API microservices, manage identity, and monitor security posture. The solution uses [Amazon Elastic Container Service](#) with [AWS Fargate](#) for containerized services, [Amazon DynamoDB](#) for consent storage, and [Amazon GuardDuty](#) for threat detection.

Open Banking services and operations diagram



The following steps describe the services and operational components for this architecture:

1. Connect securely between VPCs and services hosted on AWS or on-premises by using [AWS PrivateLink](#).
2. Implement Open Banking API specifications for Account Information and Payments services by using multiple container-based microservices hosted on Amazon ECS with AWS Fargate. Cache customer account information by using [Amazon ElastiCache](#). Host webhooks for payment status in this layer.
3. Store consumer consents, aggregated data, and API performance metrics in DynamoDB.
4. Hold a copy of the system of record in [Amazon RDS](#). Synchronize data in near real time from the bank core system.
5. Implement the identity provider (IdP) for OAuth 2.0 in a separate AWS account so that other workloads in the bank can consume it securely.
6. Provide a separate developer sandbox for the third party to integrate with the bank AWS environment and build their products.
7. Monitor for malicious activity and unauthorized behavior by using GuardDuty. Get a comprehensive view of security alerts and security posture across AWS accounts by using [AWS Security Hub CSPM](#).
8. Collect logs from all services in [Amazon Simple Storage Service](#). Analyze and monitor logs by using [Amazon OpenSearch Service](#).
9. Manage configuration by using AWS Systems Manager. Deploy environments by using [AWS CloudFormation](#). Use [Amazon EventBridge](#), [Amazon Simple Queue Service](#), and [Amazon Simple Notification Service](#) for notification capability between services.

Further reading

For additional information, see the following resources:

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

Diagram history

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

Change	Description	Date
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021
Initial publication	Reference architecture diagram first published.	September 7, 2021

RSS subscription requirement

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