



Banking modernization strategy for client lifecycle management in the AWS Cloud

AWS Prescriptive Guidance



AWS Prescriptive Guidance: Banking modernization strategy for client lifecycle management in the AWS Cloud

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

- Introduction 1**
- Banking modernization challenges 2**
 - Changing customer expectations and the rise of neobanks 2
 - Business problem 2
 - Client lifecycle management use case 3
 - Client onboarding opportunities 4
- The onboarding customer journey 5**
- Serverless onboarding 6**
 - An onboarding solution 6
 - Security and compliance considerations 8
- FAQ 10**
 - How can I be sure that AWS services won't store any personal data? 10
 - Is the architecture compliant with GDPR? 10
 - How can I ensure proper data confidentiality? 10
 - Can I integrate an AWS banking modernization solution with any legacy banking system? 10
 - Can I customize the onboarding dialog? 10
- Next steps 11**
- Resources 12**
 - References 12
- Document history 13**

Banking modernization strategy for client lifecycle management in the AWS Cloud

Patrick Rotzetter and Florent Batard, Amazon Web Services

The financial services (FS) industry is facing major disruption. More and more customers expect and even prefer the fully digital self-service banking services that financial technology companies (fintechs) and neobanks (that is, digital-only banks without physical branches) are delivering as part of the retail customer onboarding process. Traditional financial institutions can't keep pace with fintechs and neobanks. This is because banking processes across most traditional financial institutions are still manual or semi-automatic.

There is no standard framework used to implement traditional banking processes, even though these processes are common to all financial institutions. The lack of such a standard framework is hampering innovation. For example, traditional financial institutions are not in a position to capitalize on one of the latest trends in banking: embedded finance. That is, the ability of nonbanks to offer bank-like services. For more information, see [What the embedded-finance and banking-as-a-service trends mean for financial services](#) from McKinsey & Company. For traditional financial institutions to capitalize on embedded finance, they must seamlessly expose their business capabilities to other businesses.

This strategy defines a modernization approach for implementing scalable and secure standard banking processes by using an API-oriented serverless architecture that's built on AWS Cloud services. To illustrate the challenges and opportunities of this approach, this strategy focuses on a case study of client lifecycle management (CLM). The intended audience for this strategy includes CEOs, CFOs, CIOs, and senior managers in the retail banking and wealth management industry.

Banking modernization challenges

Changing customer expectations and the rise of neobanks

The FS industry is facing a massive change. Customers not only expect a fully digital journey, but also a personalized experience. To meet these customer expectations and compete with neobanks, traditional financial institutions must undergo a massive transformation that requires speed, agility, and innovation. That transformation will require traditional financial institutions to overcome the following near-term challenges:

1. Traditional banks must keep pace with neobanks, which are more agile in nature thanks to the lack of technical and process debt.
2. Traditional financial institutions must develop and provide embedded banking services through banking as a service (BaaS).
3. Traditional financial institutions must develop and enable customizable and flexible customer experiences that integrate machine learning models quickly and cost-effectively.

The key strategic factors to consider for the next few years are to leverage built-in cloud services and to enable time-to-market benefits that result from collaboration with service providers and agile transformation.

Business problem

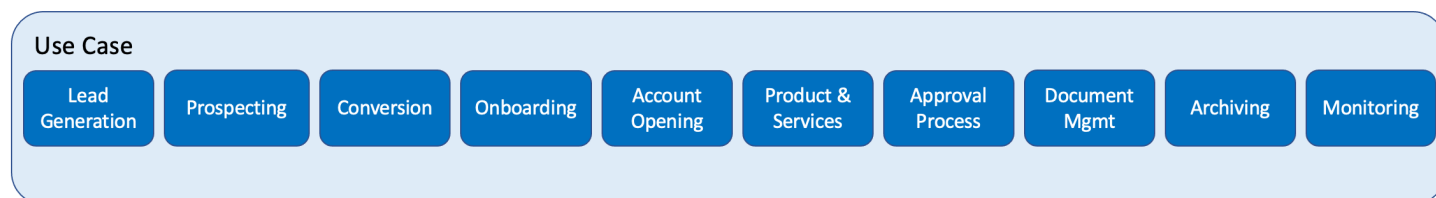
Traditional financial institutions still work with manual or semi-automatic processes. No industry standard is applied for regular banking processes, even though these processes are common to all financial institutions in similar jurisdictions. To overcome these challenges, traditional financial institutions can implement scalable and secure standard processes by using a serverless approach that's built on AWS Cloud services. For example, [Amazon Lex](#) and [Amazon Rekognition](#) provide trained machine learning models that you can integrate into enhanced customer journeys. Previously, such innovation was hampered by complex integration and expensive development processes, but now these innovations are available directly for customer applications through API calls.

AWS Cloud services not only provide additional innovation and speed of integration, but also leverage the latest security capabilities to ensure data confidentiality and privacy. We want to

highlight this modernization approach to banking for the CLM landscape. The client lifecycle is at the heart of the client relationship, which is why it must urgently be modernized and even digitized. As shown in [What the embedded-finance and banking-as-a-service trends mean for financial services](#) from McKinsey & Company, banks who want to offer BaaS will have to make it work, and "Making it work will require new technologies and capabilities because BaaS is usually distributed to clients via APIs and requires strong risk and compliance management of the embedded finance partner." The *Serverless onboarding* section in this strategy explains how you can make such a strategy work.

Client lifecycle management use case

CLM is a set of loosely interdependent and mandatory processes defining the business interaction of a client with its financial institution or insurer. The following figure provides a generalized overview of CLM stages for the banking and insurance industries:



Take into consideration that these processes vary by client segment. For example, commercial client onboarding is different from retail client onboarding. The processes can also vary from industry to industry. Onboarding a banking client is different from onboarding an insurance client. This strategy focuses on the banking retail client experience.

In the diagram above, every box (*Prospecting*, for example) represents a candidate for modernization. You can use a diagram like this to break the use case down into discrete, independent use cases. For example, the *Prospecting* use case can be broken down into constituent use cases. Then, you can map each independent use case to existing capabilities, either from cloud services providers or BaaS providers. This mapping exercise can reveal that document archiving for banking, for instance, could be a service by itself that's provided by an external party. Breaking the client lifecycle management use cases into discrete, independent use cases enables you to design and develop focused services.

This strategy focuses on the example use case of opening a new bank account. For customers, this is often a lengthy, highly-manual, and error-prone process, which results in frustration and dissatisfaction. Today most of onboarding is still happening manually through phone calls or physical touch points. However, research indicates that customers prefer a digital experience.

According to [Digital onboarding in finance: a novel model and related cybersecurity risks](#) from Open Research Europe, 70 percent of customers prefer to contract products digitally.

Client onboarding opportunities

In the post-COVID crisis age, banking and FS customers must face cumbersome and complex processes. In fact, few customers today can complete the entire process of opening a checking account from a mobile device. The majority of customers must follow a manual onboarding process for opening a bank account that is similar to the following:

1. The customer, or end user, must submit proof of identity, a legal ID, and various other data points. In many cases, these data points include documentation for a physical address, employment proof, and more. Often, the customer must visit a branch office to provide this information, or send it to the bank through standard mail.
2. Before customers can even use their banking accounts, they will receive several paper documents. Then, customers must sign and return the documents to the bank (that is, single contracts for each banking product, such as personal accounts, joint accounts, trading accounts, e-banking, credit cards, or security codes).
3. The bank digitizes all the signed documents and links them to the "work-in-progress" account to make the account operational.
4. Depending on the client segment or the applicant's risk assessment, the bank must regularly repeat this process partially or entirely.
5. If the customer's address, civil status, or any other data considered as a "change of circumstance" is updated, the process is initiated again.

Completion of this process can take up to several days or even weeks, and it often requires intervention from and approval by several bank employees from different departments.

The onboarding customer journey

A modernized customer journey for opening an account should include the following features and capabilities:

- The customer can access the onboarding service through an internet website, virtual assistant, or call center. The virtual assistant can be a text-oriented or speech-oriented chatbot, which can be integrated into a webpage or into a messaging application like Messenger or Telegram.
- Virtual assistants can interact with customers by using predefined intents to understand the customer's needs and background. For example, the virtual assistant can ask clients to share their interests, share personal details to open an account, scan their ID and take a selfie, or confirm that they want to proceed with opening an account.
- The chatbot will be multi-lingual.
- Customers can interact with the chatbot using voice or text.
- After you collect all the required information from the customer, then you can send the information to the bank to initiate the account opening process. The customer will receive an email notification confirming that the process started.
- You can close the dialog after the account has been opened.
- Avoid storing customer information in the cloud. This information must be transient and exist only during the customer engagement. After the account opening process is completed, you must delete the data and, if possible, provide some evidence of the deletion. Even if the dialog is interrupted, you must confirm that all personal data is deleted (no caching).
- Avoid storing logs with personally identifiable information (PII).
- Customers can be confident that no PII data is stored. If the services used are serverless, then no data or logs with personal data remain.

Serverless onboarding

This section focuses on the client onboarding use case. It serves as an example of how to reuse serverless patterns to modernize a key customer journey.

An onboarding solution

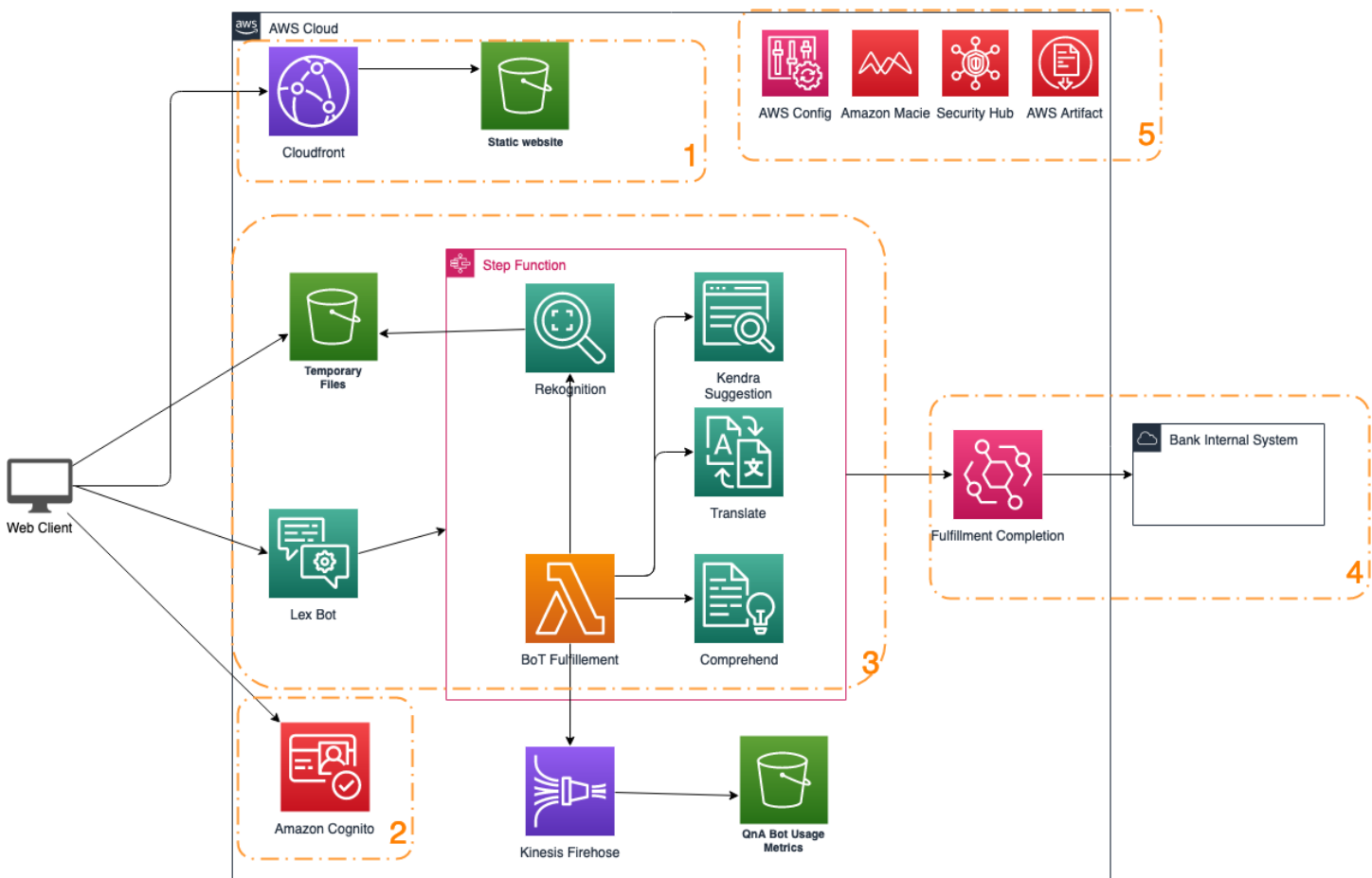
An onboarding solution usually does not come in isolation, but is part of a larger landscape. The components and services required for integration include the following:

1. Lead generation (CRM systems)
2. Due diligence (AML, risk and compliance)
3. Central client data file (core banking systems)
4. Digital signature management (using a third-party tool like DocuSign)
5. Biometrics verification
6. Digital contract management lifecycle
7. Process and operating model automation

Note

The integration of these components with external applications is beyond the scope of this strategy.

The following diagram illustrates a serverless architecture for interacting with customers through chatbots in the AWS Cloud:



Note

The architecture is based on the QnA Bot on AWS solution and the updates to Amazon Rekognition. For more information, see [QnA Bot on AWS](#) in the AWS Solutions Library and [Amazon Rekognition announces updates to its face detection, analysis, and recognition capabilities](#) in the AWS Machine Learning Blog.

The diagram shows the following workflow:

1. Customers access the front-end components to interact with the bot. In this case, the front-end library can be included in existing assets in various ways. For example, you can integrate these components into the corporate website or link them to a major chatting app. These components are delivered in JavaScript and served through static asset management by using Amazon Simple Storage Service (Amazon S3) and Amazon CloudFront.

2. The front-end components embed connections to the back-end components to validate credentials through Amazon Cognito. Once authorized, the client service can communicate with the onboarding solution's service.
3. Customers interact with the service by using an Amazon Lex bot. A predefined scenario for onboarding is run by using AWS Step Functions to call the necessary services to complete the process. You can tailor this scenario to fit the needs of the banking system (for example, to meet photography requirements, translation in multiple languages, and natural language processing to match customer intent). You can use AWS Lambda functions to fulfill the actions of the Amazon Lex bot with the customer and call services depending on the onboarding stage.
4. The output of Step Functions is sent in a batch to the bank internal system through Amazon EventBridge, which uses your preferred method to connect to the Bank Internal System. You can handle this communication either by peering to another AWS account or network peering through a virtual private network (VPN) to the banking system.
5. The entire architecture is designed for security and compliance by using AWS Config, Amazon Macie, AWS Artifact, and AWS Security Hub.

Security and compliance considerations

Maintaining security and handling sensitive information is at the heart of the serverless onboarding approach. Serverless onboarding is designed to be stateless and doesn't store any PII or critical business data. To ensure continuous security and compliance, you must put the following services in place and configure them correctly:

- AWS Config ensures the integrity and confidentiality of the services through rules tailored for compliance. Controls in place will validate overall compliance and prevent drifts in configuration.
- Macie detects and tags any PII throughout the process.
- Security Hub ensures that all services are defined and used within the security scope.
- AWS Artifact provides audit evidence for the compliance of AWS services.

Transactions and non-repudiation evidence can be stored and encrypted by using [customer managed keys](#) or a dedicated hardware security module (HSM) that's sealed for audit purposes. You can also seal data at low cost while ensuring integrity and security by using [Amazon S3 Glacier](#).

Only the operational team should be permitted to access the environment. Any development activities should be streamlined through automatic continuous delivery to prevent any human

access to the production environment. For auditing purposes, an additional role should be provisioned to access artifacts and compliance reports on the platform.

AWS can help ensure that you meet regulatory and compliance standards. For more information, see [FINMA ISAE 3000 Type 2 Report](#) in the AWS documentation. You can also download relevant banking artifacts directly from [AWS Artifact](#), including FINMA Circular 2008/21, FINMA ISAE 3000 Type 2, FINMA Circular 2018/03, and Global Financial Services Regulatory Principles.

FAQ

How can I be sure that AWS services won't store any personal data?

You can use AWS Config to check the compliance of the architecture components in combination with Macie to detect and remediate any personal data that's discovered.

Is the architecture compliant with GDPR?

Yes, as per GDPR Chapter 2, [Article 5](#), the architecture is designed not to retain any data related to the user in the long term but only to processes PII. This is ensured through continuous compliance mechanisms. Additionally, Macie ensures tagging and classification of any PII data that could endanger GDPR compliance.

How can I ensure proper data confidentiality?

Confidentiality is ensured through continuous compliance and security patterns by using Security Hub, AWS Config, and Macie. Confidentiality breaches are immediately detected, remediated, and escalated. We also recommend that you add guardrails at the deployment phase. Guardrails help ensure that no critical change that could endanger confidentiality gets pushed to the architecture.

Can I integrate an AWS banking modernization solution with any legacy banking system?

The solution is designed to be adaptable and can integrate with any legacy banking system that can accept standard API calls.

Can I customize the onboarding dialog?

Yes, you can customize the chatbot dialog to fit your use case and targeted dialog sequence. For more information, see [QnA Bot on AWS](#) in the AWS Solutions Library.

Next steps

There are other CLM use cases beyond client onboarding, and these use cases can be modernized with built-in AWS Cloud services. The following scenarios explore some potential modernization use cases:

1. Integrate product recommendations into banking products and services by using [Amazon Personalize](#) or other custom made recommendations. Although FS products are different from retail products, there is still a high degree of personalization predicted by 2025 according to [Industry Leaders Predict Massive Changes in Banking by 2025](#) from The Financial Brand. Customers might be willing to pay more for highly personalized and contextual services. This can only be implemented with smarter solutions based on machine learning.
2. You can use [Amazon Fraud Detector](#), an online fraud detector based on machine learning, for account monitoring. You can also customize other built-in algorithms to provide standardized account monitoring services (for example, by using anomaly detection).
3. You can have smart assistants that directly engage with customers by using AWS Lex and chatbots based on the latest natural language models.

There is no limit to creative solutions, but you must integrate and deploy such solutions at scale and speed.

Resources

References

- [What the embedded-finance and banking-as-a-service trends mean for financial services](#) (McKinsey & Company)
- [Digital onboarding in finance: a novel model and related cybersecurity risks](#) (Open Research Europe)
- [Industry Leaders Predict Massive Changes in Banking by 2025](#) (The Financial Brand)
- [AWS User Guide to Financial Services Regulations and Guidelines in Switzerland and FINMA workbooks publications](#) (AWS Security Blog)
- [FSI Services Spotlight: Featuring AWS Lambda](#) (AWS blog post)
- [Amazon Rekognition announces updates to its face detection, analysis, and recognition capabilities](#) (AWS Machine Learning Blog)
- [QnA Bot on AWS](#) (AWS Solutions Library)

Document history

The following table describes significant changes to this guide.

Change	Description	Date
Initial publication	—	June 8, 2022