Central bank digital currency (part 2): Technology options and performance criteria

2021

Table of contents

Introduction 3

Central bank digital currency (CBDC) architectural considerations: A summary 4

Key CBDC technology options 5

Technical performance criteria 7

- Security 7
- Instant settlement and throughput 8
- Availability and resilience 9
- Scalability 11
- Interoperability, flexibility, and adaptability 11

Example reference architecture 12

Conclusion 17

References 18

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Introduction

In "Central bank digital currency (part 1): Objectives and architectural considerations," we looked at reasons for deploying a CBDC and how those objectives influence architectural decisions.

Part 2 of the series builds on those concepts. Here we review technology options available to store and process CBDC data, consider key technical performance criteria for a CBDC solution, and provide a detailed reference architecture to deliver on objectives. The guidance will help inform central banks' ongoing exploration of this emerging area, building on existing research into the opportunities and success criteria.[1,2]

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

CBDC architectural considerations: A summary

Direct CBDC versus two-tier architectures with payment service providers (PSPs)

There are two broad architectural options to support a general purpose CBDC: direct and two-tier. A direct architecture allows transactions to take place through a direct connection between individual CBDC users and the central bank (or a central services provider acting on the central bank's behalf).

A two-tier architecture requires CBDC users to carry out transactions through PSPs, such as commercial banks. This allows central banks to focus on their core competences, leaving the commercial sector to focus on innovation and customer service. So we recommend the two-tier approach.

Account-based versus token-based models and offline operation

Central banks need to decide whether to implement their CBDC solution with an account-based or token-based digital currency.

In account-based models, the CBDC system maintains records of user account balances. With a two-tier approach, users access their CBDC through PSPs that authenticate their identity and maintain account records.

Token-based models can operate in a similar way, but by recording ownership of each CBDC token. Alternatively, tokens can operate purely as bearer instruments without requiring online validation of user identity or CBDC ownership. This bearer instruments approach supports offline operability of CBDC. However, in an account-based solution, offline operability could be provided by an additional PSP.

Clearing and settlement models and network architecture

Central banks currently use various models to support real-time gross settlement (RTGS) and fast payment systems. The example reference architecture detailed on page 12 supports direct settlement using a hub network with no pre-clearing between PSPs.

Learn more in "CBDC (part 1): Objectives and architectural considerations."

Read now

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Key CBDC technology options

"Which technology should we use for keeping records of CBDC ownership and executing transactions?"

This is a critical CBDC design question. There are three overarching options to choose from:

- Distributed ledger using blockchain technology
- Centralized databases
- Distributed bearer instrument solutions[3]

Here, we focus on the first two options because they're well understood and documented across the industry.

Distributed ledger using blockchain technology

Central banks like China,[4] Sweden,[5] and Cambodia[6] are investigating blockchain technologies for their CBDC pilot programs.

With this technology option, the blockchain technology distributes the ledger across multiple nodes and records transactions as cryptographically linked blocks in a chain. This provides reliability and auditability because these records can't be subsequently changed. These permissioned blockchain solutions give the central bank control over the system and limit participation to approved PSPs. They can sometimes offer a lower cost of operation compared to public blockchain technologies. And they can also manage much larger transaction volumes.

Some blockchain technologies can embed code to facilitate smart contracts and programmable money. However, there are alternative ways to deliver this functionality in solutions based on either blockchain or centralized databases through additional PSP services.

The key benefit of a distributed ledger solution is the ability to create a system where each participant has a share of operational control. This allows them to transact without a central governing authority or intermediary. However, this also presents a challenge, because central banks need the ability to exercise ultimate control over the entire CBDC system to provide stability.

Centralized databases

This technology option leverages the relational database approach widely used in commercial banking systems. In the past, it was common for commercial banks to use databases like DB2 and Oracle. Now, they are increasingly turning to newer services like Amazon Aurora to reduce costs and access advanced features.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Since CBDCs built with distributed ledger technologies use a cryptographically assured chain of records, they offer one major advantage over their relational database counterparts: immutability. If central banks prefer a relational database approach without sacrificing immutability, we recommend Amazon Quantum Ledger Database (Amazon QLDB), which offers centralized control with additional auditability, immutability, and reliability. As with blockchain solutions, Amazon QLDB prevents CBDC system users from making retrospective ledger changes. (See page 12 for a reference architecture using Amazon QLDB.)

INTRODUCTION
CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY
KEY CBDC TECHNOLOGY OPTIONS
TECHNICAL PERFORMANCE CRITERIA
EXAMPLE REFERENCE ARCHITECTURE
CONCLUSION
REFERENCES

Technical performance criteria

For a CBDC solution to be successful, it must provide a robust and usable infrastructure for executing payments. A group of central banks, together with the Bank for International Settlements,[7] set out the following key criteria for CBDC systems:

- Security
- Instant settlement and throughput
- Availability and resilience
- Scalability
- Interoperability, flexibility, and adaptability

Let's look at each in more detail.

Security

> **"Both the infrastructure and participants of a CBDC system should be extremely resistant to cybersecurity attacks and other threats. This should also include ensuring effective protection from counterfeiting."**
>
> Bank for International Settlements

Both blockchain and centralized database approaches can use cryptographical techniques to secure messages for processing transactions. They also rely on PSPs to apply robust authentication methods that prevent unauthorized access.

Critical security requirements for the online aspects of a two-tier architecture include:

- Physical security of the IT facilities used to deliver the CBDC solution
- Processes for managing system configurations
- Strict access controls for performing administrative actions

Centralized databases have fewer security vulnerabilities than distributed blockchain solutions because they have a more limited physical footprint, have fewer entry points for cybersecurity attacks, and make it easier to maintain a consistent IT environment.

Offline operation generally presents security risks, and these are similar for blockchain and centralized database solutions. For example, counterfeiting is a risk with CBDC solutions that use a token-based bearer instrument approach.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Instant settlement and throughput

> **"Instant or near-instant final settlement should be available to end users of the system. The system should be able to process a very high number of transactions."**
>
> Bank for International Settlements

There are many factors affecting transaction processing time and throughput. It's therefore not possible to generalize about what can be achieved with either blockchain or centralized database solutions. However, key factors impacting performance include:

- The number of messages participants must exchange to process the transaction
- The complexity of the steps the system follows for settlement

With centralized databases, as few as two messages could be required to settle a transaction. Blockchain solutions are more complex because of their distributed architecture; the number of required messages depends on the specific blockchain technology. When settlement requires a consensus across multiple nodes in the blockchain network, more messages are required. This increases transaction time and potentially limits throughput. One way to reduce message volume and increase throughput is to combine transactions in batches (although this could potentially increase transaction time).

Blockchain solutions can also involve more complex functions to settle transactions. In some cases, they must step through each transaction in the chain to prove CBDC record validity. Token-based models can also involve more complexity than account-based models, depending on the number of tokens involved.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Availability and resilience

> **"A CBDC system should be extremely resilient to operational failure and disruptions, natural disasters, electrical outages, and other issues. There should be some ability for end users to make offline payments if network connections are unavailable. End users of the system should be able to make payments 24/7, 365."**
>
> Bank for International Settlements

CBDC systems must be highly resilient. They should be available and fully functional, ideally without any service interruption—even if one or more components fail or aren't available.

The reference architecture on page 12 incorporates the Committee on Payments and Market Infrastructures core principles for systemically important payment systems[8] and the AWS Well-Architected Framework pillars (operational excellence, security, reliability, performance efficiency, and cost optimization). It provides:

- Multi-data center and multi-AWS Regional failover capabilities
- Auto scaling to increase and decrease service capacity based on demand
- Automated data backup and restore
- Tiered architectures that can scale up and down independently
- Loosely coupled applications so failures don't impact the entire solution

High availability

We designed the reference architecture on page 12 with a centralized database approach. It uses AWS Cloud multi-Availability Zone infrastructure to deliver high availability and resilience. Every AWS Region is made up of at least two and usually three or more Availability Zones (AZs). AZs are physically separated in different flood plains, located in different fault zones, have redundant Internet providers, and are on separate power grids where available. The underlying Amazon QLDB technology storing the CBDC ledger and other system components are designed to operate in a multi-AZ configuration. By designing CBDC solutions with AWS Regional and multi-AZ configurations, you can build highly fault-tolerant architectures that recover automatically, with zero downtime, from many failure scenarios.

In a blockchain solution, each network node relies on its own, high-availability solution that may be standardized or variable depending on who's managing the node. Different blockchain technologies support this architecture in different ways. Some solutions can continue operating even if some nodes are unavailable. These provide additional resilience and reduce reliance on a centralized service provider.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Disaster recovery

Even with a highly resilient and available architecture, central banks need to protect their CBDC solution from complete system outages—and be able to recover it to a functional state in another location.

Many centralized databases like Aurora support disaster recovery through a multi-AWS Region architecture. This means transactions automatically replicate in a secondary AWS Region so service can be restored with a minimal recovery point objective (RPO). With Aurora, the RPO can be under a minute if there is a catastrophic failure.

With Amazon QLDB, all completed transactions are streamed asynchronously to a secondary AWS Region and updated in a secondary Amazon QLDB. If there's a catastrophic event, the entire architecture infrastructure is recreated in minutes in the secondary AWS Region. It points to the existing secondary Amazon QLDB, and transaction processing can resume. Disaster recovery for Amazon DynamoDB is configured using the DynamoDB Global Tables feature, which asynchronously moves items from your primary AWS Region to your secondary AWS Region.

Different blockchain solutions are at different maturity levels in terms of disaster recovery functionality. For some solutions, a single node failure won't necessarily affect the whole system's performance. Other solutions need to develop disaster recovery capabilities further.

Offline capabilities

If there's a disaster, limited offline transactions could continue through CBDC devices like smartphones. Such offline capability would be dependent on the ability for these devices to be locked down under strict security protocols.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Scalability

> **"To accommodate the potential for large future volumes, a CBDC system should be able to expand."**
>
> Bank for International Settlements

We designed the reference architecture on page 12 using AWS managed services, all of which scale to meet your business needs with little or no manual intervention.

For example, AWS Lambda allows you to deploy and run code without having to provision or operate servers. Lambda can run one or even thousands of concurrent invocations of the same function. DynamoDB is a a key-value data store that scales to storage needs with single millisecond response times for retrieving data. Amazon QLDB is a managed service database that allows for cryptographically verifiable data chaining. With both database services, there are no servers to provision. Simply configure your read/write needs and start storing data.

Interoperability, flexibility, and adaptability

> **"The system needs to offer sufficient interaction mechanisms with private sector digital payment systems and arrangements to allow easy flow of funds between systems. A CBDC system should be flexible and adaptable to changing conditions and policy imperatives."**
>
> Bank for International Settlements

A PSP ecosystem is an important part of CBDC system design. Our reference architecture gives you an example of how to implement one. We designed it to be interoperable with existing private sector digital payment systems using APIs. This approach allows for additional functionality, like smart contracts and offline payments. PSPs can provide these services and integrate them with the rest of the ecosystem, using standardized APIs and processes the central bank defines.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

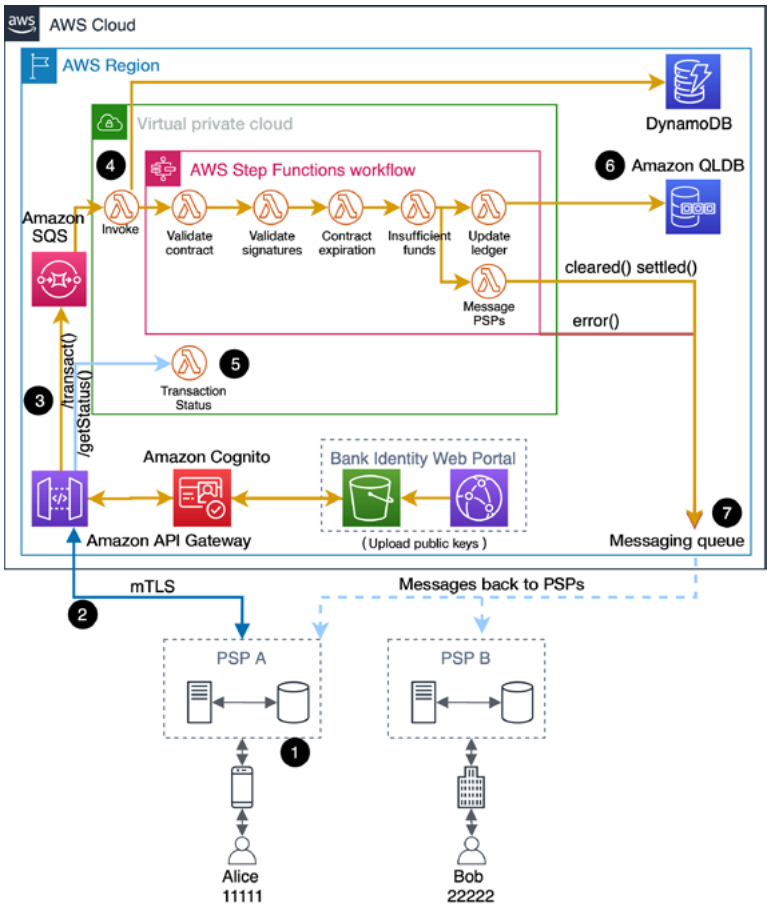
EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Example reference architecture

This reference architecture assumes an account-based model and uses a centralized database solution. Amazon QLDB powers the CBDC core ledger. It's a direct settlement architecture without a pre-clearing process between PSPs before settlement and uses a hub network model. The central bank controls everything shown within the AWS Region.



Transaction flow

Stage 1

The transaction begins with Alice (the payor) requesting PSP A to make a payment to Bob (the recipient). Alice provides evidence of her identity, which PSP A verifies through an established payment processing method. At this stage, PSP A may also carry out anti-money laundering (AML) or fraud detection checks. PSP A then creates a CBDC payment message to capture the request.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

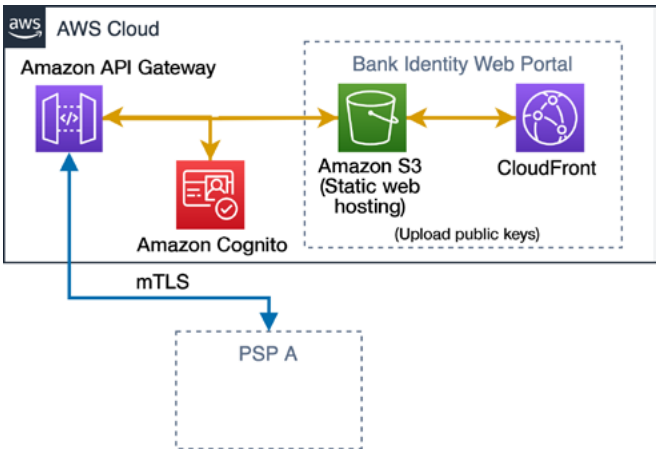
REFERENCES

Stage 2

PSP A digitally signs the payment message through asymmetric encryption using its private key. Then it sends the signed request to the CBDC transaction API Gateway.

The API Gateway authenticates PSP A's identity using information in the Bank Identity Web Portal. For additional security, Amazon API Gateway supports a mutual TLS protocol to authenticate the identities of both PSP A and the CBDC API Gateway.

Each affiliated PSP can access the CBDC Bank Identity Web Portal (powered by Amazon Simple Storage Services (Amazon S3) and Amazon CloudFront) through a secure "Know Your Customer" process.



Using the portal, authorized personnel operating on a PSP's behalf can log in, update contact information, and upload the PSP's public key for verifying digital signatures.

Stage 3

Once the CBDC API Gateway has received and processed the request, the message enters a queueing system using Amazon Simple Queue Service (Amazon SQS). This allows the system to cope with transaction volume changes and manage throughput if there are sudden activity spikes.

Stage 4

The system creates an entry in DynamoDB for tracking transaction status. Then, it checks the proposed transaction's validity.

It sends the message to an AWS Step Functions engine, a highly scalable serverless state machine. The Step Functions execute each state as a separate Lambda function, drawing on compute resources as required. Each function validates one aspect of the payment message to confirm that:

- The request is new and hasn't been processed before—by validating the incoming message's contract_id
- The incoming message's digital signature is a valid digital signature for PSP A
- The CBDC account details are valid

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

- PSP A is authorized to process the transaction on Alice's behalf and that PSP B is authorized to process the transaction on Bob's behalf
- The transaction's expiration time (specified in the payment message) hasn't been exceeded
- Alice has sufficient funds in her account

Note: You can replace Lambda functions with other compute technologies like container-based microservices or Amazon Elastic Compute Cloud (Amazon EC2) instances.

Stage 5

At any time during the process, PSP A can check the transaction status by submitting the transaction id to the getStatus() API. The getStatus() API is backed by a Lambda function that reads the transaction id status from DynamoDB. It returns a status of "started," "running," "error," "completed," or "not found."

Stage 6

The CBDC core ledger records the transaction request by updating the balances in both Alice's and Bob's accounts.

Stage 7

The process places a message for PSP A and PSP B confirming settlement in a messaging queue. The message also confirms the transaction details as per the initiating payment message and includes the updated CBDC account balances.

This is the first communication PSP B receives about the transaction. PSP B then informs Bob through agreed channels, such as a mobile app or messaging system.

At this point, the transaction is final and settlement is complete. Bob can immediately use the CBDC Alice has transferred into his account.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

CBDC payment message structure

Here's an example payment message structure. It's based on the minimum information required to execute the transaction, so it only has details about each CBDC account, the PSPs involved, a valid time stamp, and a digital signature to validate the request with PSP A during Stage 3.


```
{
    "contract_id":"23948hjkwehrkjw24ea7u24hwe23s4",
    "contract_info":
    {
        "amount":100.00,
        "debit":
        {
            "debit_account":11111111,
            "debit_psp_id":"1231232-5"
        },
        "credit":
        {
            "credit_account":22222222,
            "credit_psp_Id":"7485768-A"
        },
        "time_stamp":"2020-08-18T08:00:00+00:00",
        "valid_until_time":"2020-08-18T08:30:00+00:00"
    },
    "init_psp_sig":"982yuywy23234wer2uherjh82233p2o"
}
```


contract_id = UUID
amount = amount to be debited
debit_account = user account number to be debited
debit_psp_id = the PSP's identification number for the account being debited
credit_account = user account number to be credited
credit_psp_id = the PSP's identification number for the account being credited
time_stamp = timestamp UTC time when the contract is created
valid_until_time = future timestamp UTC for the contract to expire
init_psp_sig = digital signature of the 'contract_info' from the initiating PSP

Again, this message structure shows minimal information. It demonstrates how to carry out CBDC transactions while preserving a degree of anonymity for Alice. In this example, Alice's identity isn't in the payment message and there's no need for PSP B to pass it on to Bob once the transaction is executed. In a full implementation, the central bank may decide to include additional fields with detailed information ISO 20022 specifies for payment messages.

CBDC core ledger structure

The core ledger could also contain minimal CBDC account information, like owner identities, the PSPs responsible for managing each account, and the current balances. Subject to public policy and privacy considerations, it's possible to store additional information in the core ledger. You may want to do this to help analyze economic conditions or prevent unlawful activity.

INTRODUCTION
CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY
KEY CBDC TECHNOLOGY OPTIONS
TECHNICAL PERFORMANCE CRITERIA
EXAMPLE REFERENCE ARCHITECTURE
CONCLUSION
REFERENCES

PSP roles

This reference architecture assumes a single PSP is involved for each Alice and Bob. PSP A (acting on Alice's behalf) carries out each key transaction function:

- Capturing the details of the transaction Alice wants to make
- Authenticating Alice's identity
- Carrying out prevention of unlawful activity checks, such as AML or fraud detection
- Managing the CBDC account on Alice's behalf
- Providing payment services to execute the transaction

In reality, different PSPs can provide different functions so long as the CBDC system receives the payment message with PSP A's digital signature. PSP A could rely on other PSPs to carry out the other functions based on a set of APIs and security protocols defined by central bank. This is where a PSP ecosystem comes in.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

Conclusion

As we've seen, distributed ledgers using blockchain and centralized databases are the two main technology options for recording CBDC ownership and processing payments. Each approach has strengths and weaknesses in terms of their key technical performance characteristics.

Our reference architecture uses centralized databases and cloud technologies. Centralized databases are relatively simple to implement and can potentially provide key functionality often associated with blockchain solutions, such as offline transactions and smart contracts through PSP overlay services. When designed properly, cloud-enabled solutions can help central banks meet key operational requirements (such as high availability and resilience), support a PSP ecosystem, and provide robust security.

Further work is required to address issues with using blockchain solutions for a general-purpose CBDC. This may involve taking advantage of new developments in blockchain technologies or different implementation approaches. Both blockchain and centralized database technology options will benefit from additional research into key technical performance requirements, such as meeting transaction throughput during peak periods, improving security, and achieving the required operational resilience and availability.

Looking to take the next step?

No matter where you are in your journey to the cloud we are here to help. For more information, browse our technical resources to expand your knowledge of the AWS Cloud or contact us to connect with an AWS expert.

INTRODUCTION

CBDC ARCHITECTURAL CONSIDERATIONS: A SUMMARY

KEY CBDC TECHNOLOGY OPTIONS

TECHNICAL PERFORMANCE CRITERIA

EXAMPLE REFERENCE ARCHITECTURE

CONCLUSION

REFERENCES

References

1 Bank for International Settlements, 9 October 2020: Central bank digital currencies: foundational principles and core features.
2 Bank for International Settlements, 24 June 2020: Annual economic report: Central banks and payments in the digital era.
3 eCurrency Mint Limited, 2017: The case for digital legal tender: The macroeconomic policy implications of digital fiat currency.
4 Binance Research, August 2019: First look: China's central bank digital currency.
5 Sveriges Riksbank, February 2020: The Riksbank's e-krona pilot.
6 Hyperledger.org, August 2020: The National Bank of Cambodia boosts financial inclusion with Hyperledger Iroha.
7 Bank for International Settlements, 9 October 2020: Central bank digital currencies: foundational principles and core features.
8 Bank for International Settlements, January 2001: Core principles for systemically important payment systems.