AWS re:Invent

DAT375

How to make your data history immutable and verifiable

Marc Bowes

Principal Engineer, Amazon QLDB

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws

Agenda

Challenges faced with traditional RDBMS systems

Write Ahead Logs

The QLDB architecture

How best to take advantage of QLDB

Let’s build a bank

The schema of our bank

`Accounts`

Account name	Balance	Since
John	$100	2019
Mary	$250	2004

`Transactions`

From	To	Amount	Date
John	Mary	$50	Thursday

Our bank

What are some use cases?
Transfer money, view balance and statements, receive alerts

What are the entities?
Accounts, transactions

What are the different teams / departments?
Core banking application, marketing, alerts, fraud

The OLTP perspective

Accounts

Account name	Balance	Since
John	$100	2019
Mary	$250	2004
New customer	$0	NOW()

Transactions

From	To	Amount	Date
John	Mary	$50	Thursday
Mary	Joe	$50	NOW()

Registration of new users

Recording transactions

Updating balances

The marketing perspective

Accounts

Account name	Balance	Since
John	$100	2019
Mary	$250	2004
New customer	$0	NOW()

Transactions

From	To	Amount	Date
John	Mary	$50	Thursday

New customers

Retention

The fraud / alerts perspective

Accounts

Account name	Balance	Since
John	$100	2019
Mary	$250	2004

Transactions

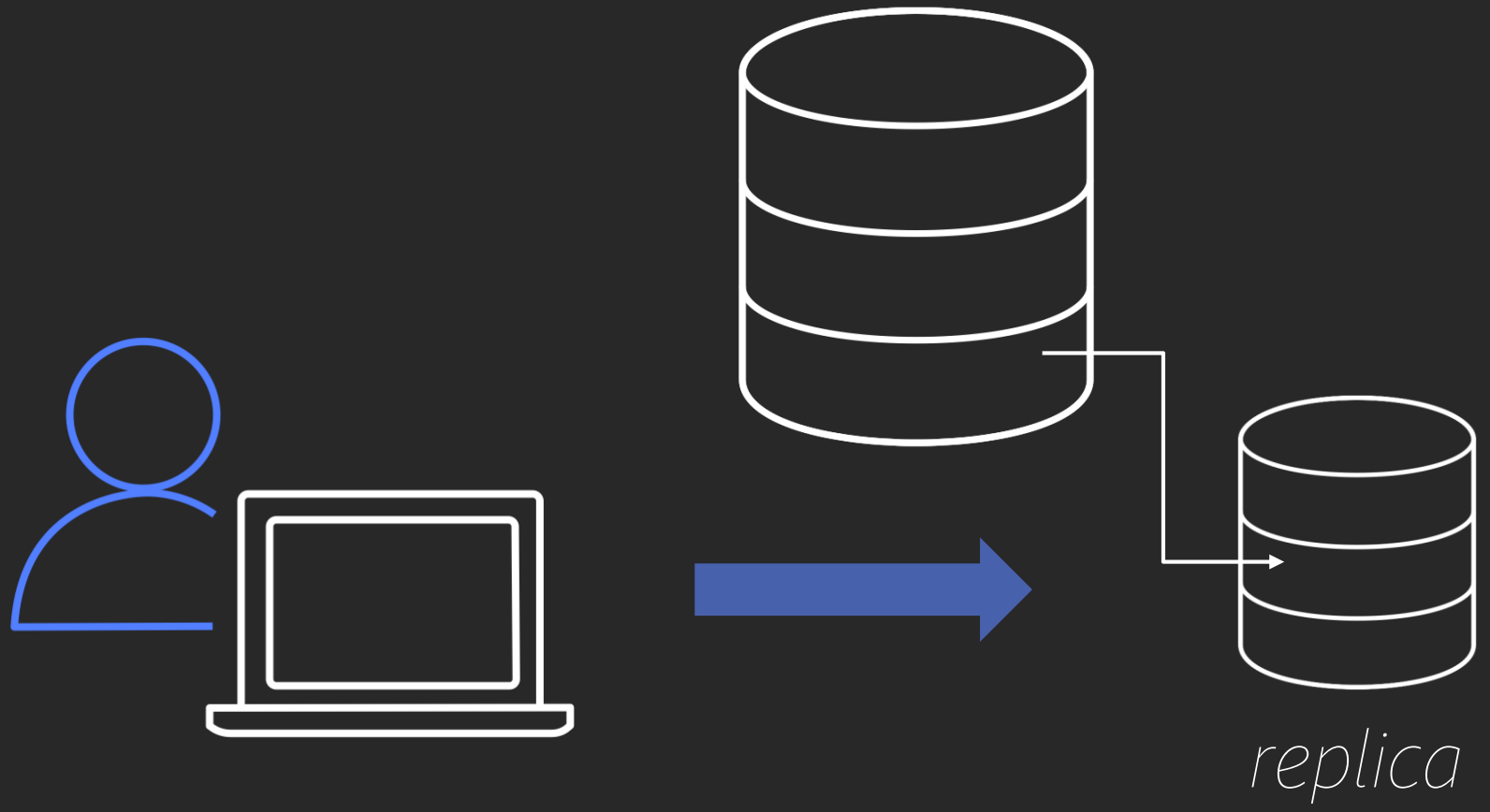
From	To	Amount	Date
John	Mary	$50	Thursday
John	Mary	$50	Thursday
John	Mary	$50	Thursday
John	Mary	$50	Thursday

Suspicious transactions

Need access to all transactions

Real time

Direct SQL access



Direct SQL access

- Setting up replicas is expensive
- Permissions can get complicated
- Some queries may not have the required indexes
- The replica may not have enough retention
- Queries can impact each other
- Creates dependencies on schema

Snapshots



Snapshots

- Arbitrary retention
- Indirection over schema
- Harder to consume
- Still need to setup dedicated replicas and hosts to run the code
- Storage can get expensive
- Can miss changes
- Not real time

Events

Accounts

Account name	Balance	Since
John	$100	2019
Mary	$250	2004
New customer	$0	NOW()



Events

- Need to generate the event
  - Application code
  - Database triggers
- Real time
- Hard to decide what to put in the event
- Ordering and de-duplication is really hard
- Consumer often doesn’t want events

Audit tables

`Transactions`

Id	From	To	Amount	Date
1	John	Mary	$50	Thursday
2	John	Mary	$50	Thursday
3	John	Mary	$50	Thursday
4	John	Mary	$50	Thursday

- Needs to be maintained
  - Application
  - Database triggers
- Unbounded growth
- Indexes are problematic

Common themes

All of the solutions require work and have ongoing cost

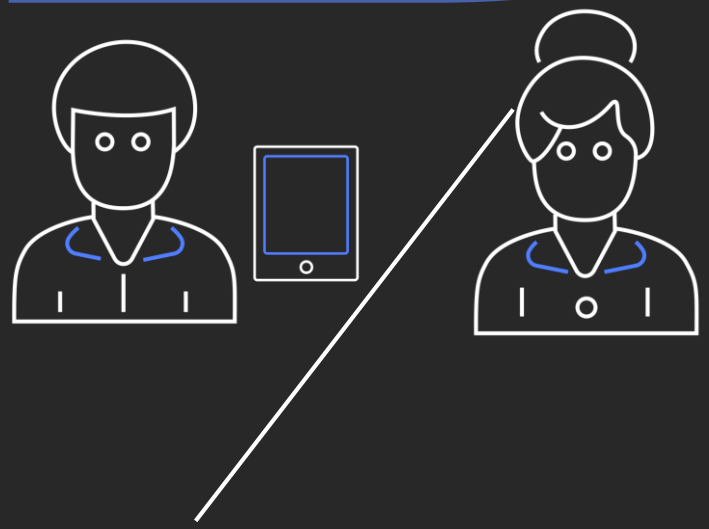
None solve for all use cases

All require foresight

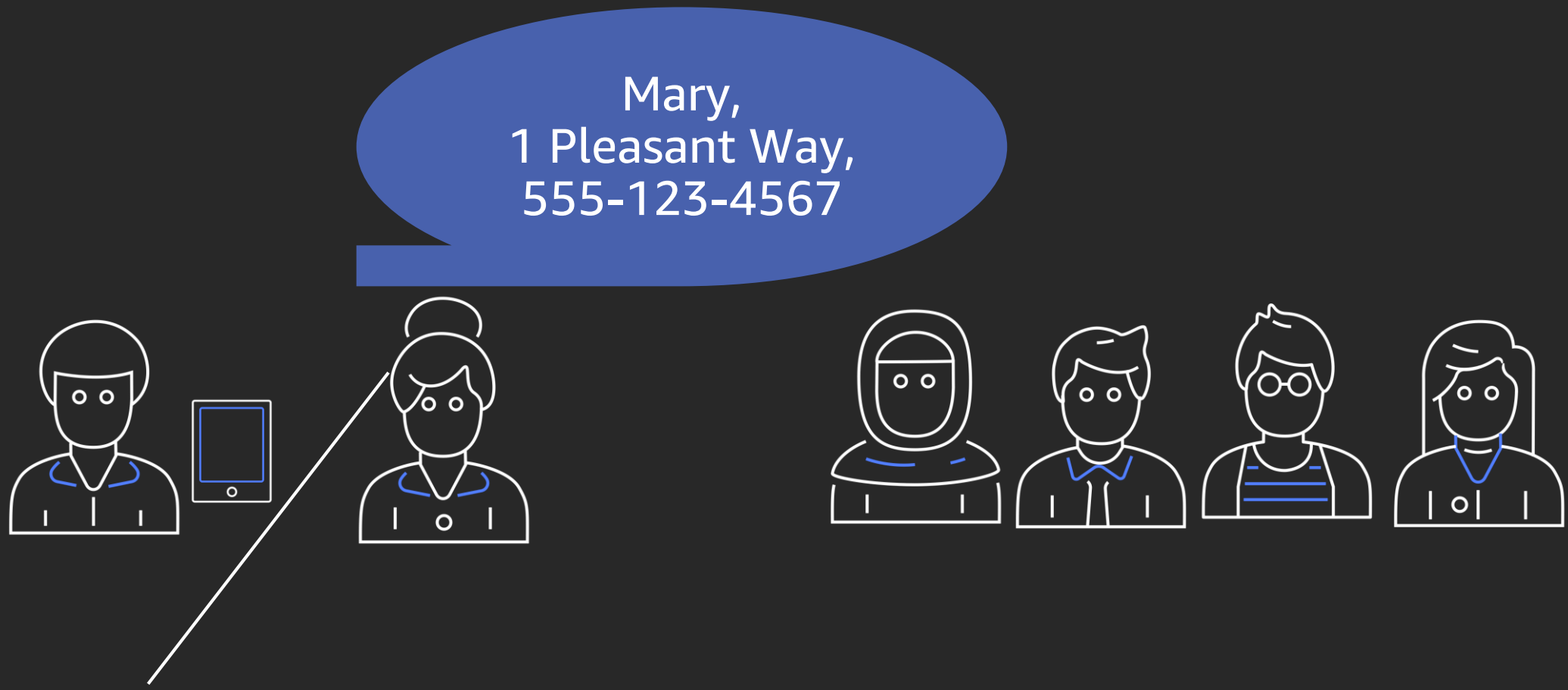
Lack of integrity

Performance overhead

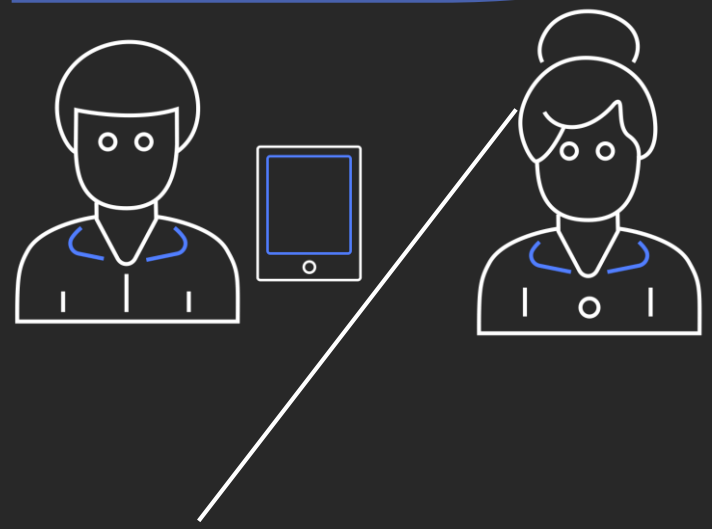
Name,
address and
phone number?





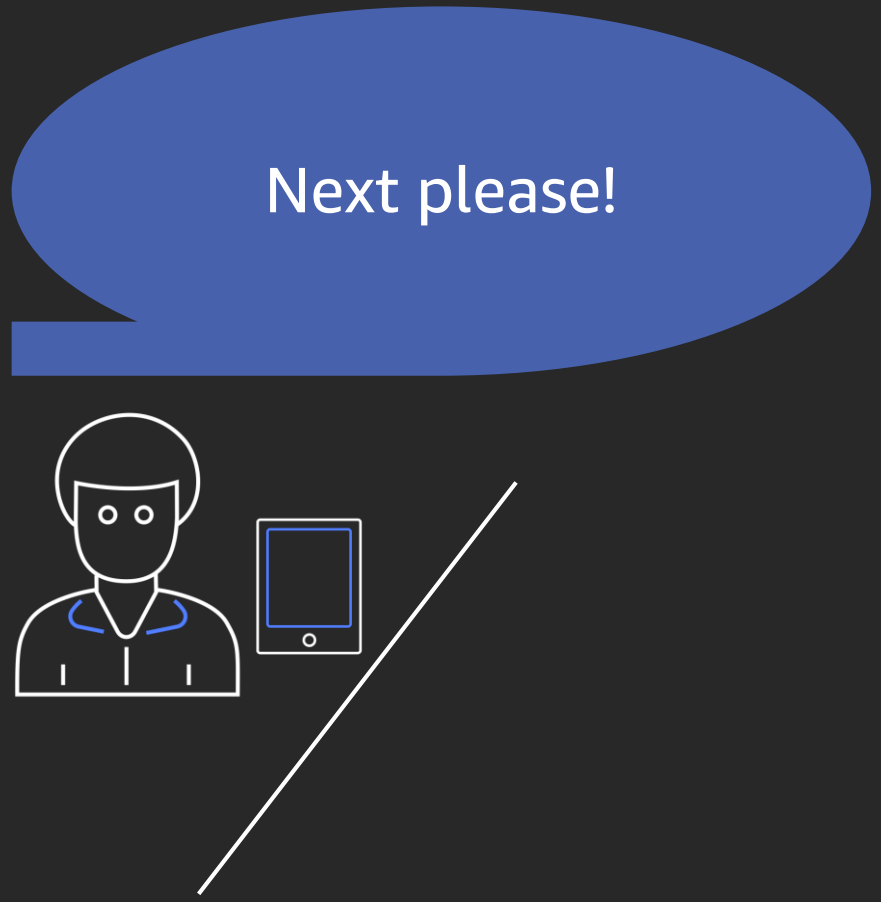

Mary,
1 Pleasant Way,
555-123-4567

You’re all set





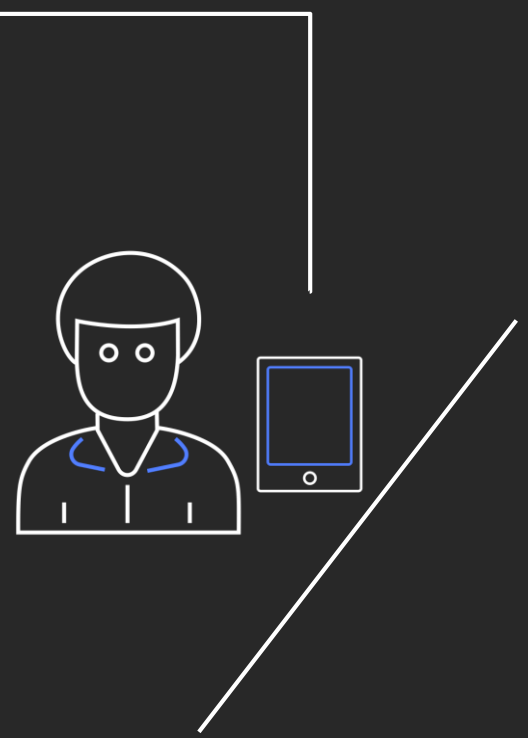
Write Ahead Logs



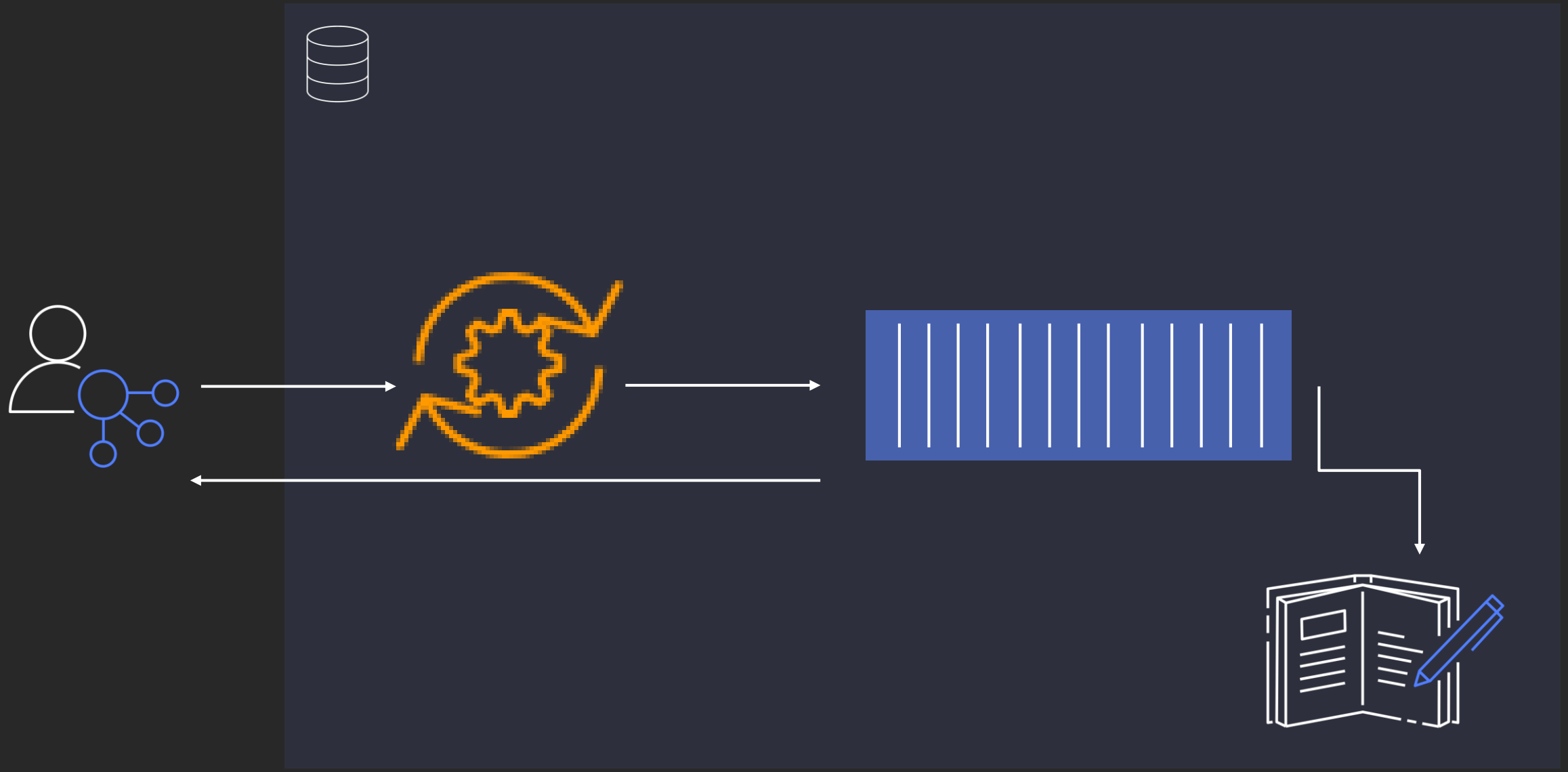




Take all the new address information
And build a phone book!



Write Ahead Logs



Write Ahead Logs

First write down the changes

Then acknowledge them

Later, organize the data

Database implementation detail

Writing to disk is slow

Sequential writes are faster

All changes are in the WAL

The QLDB Journal

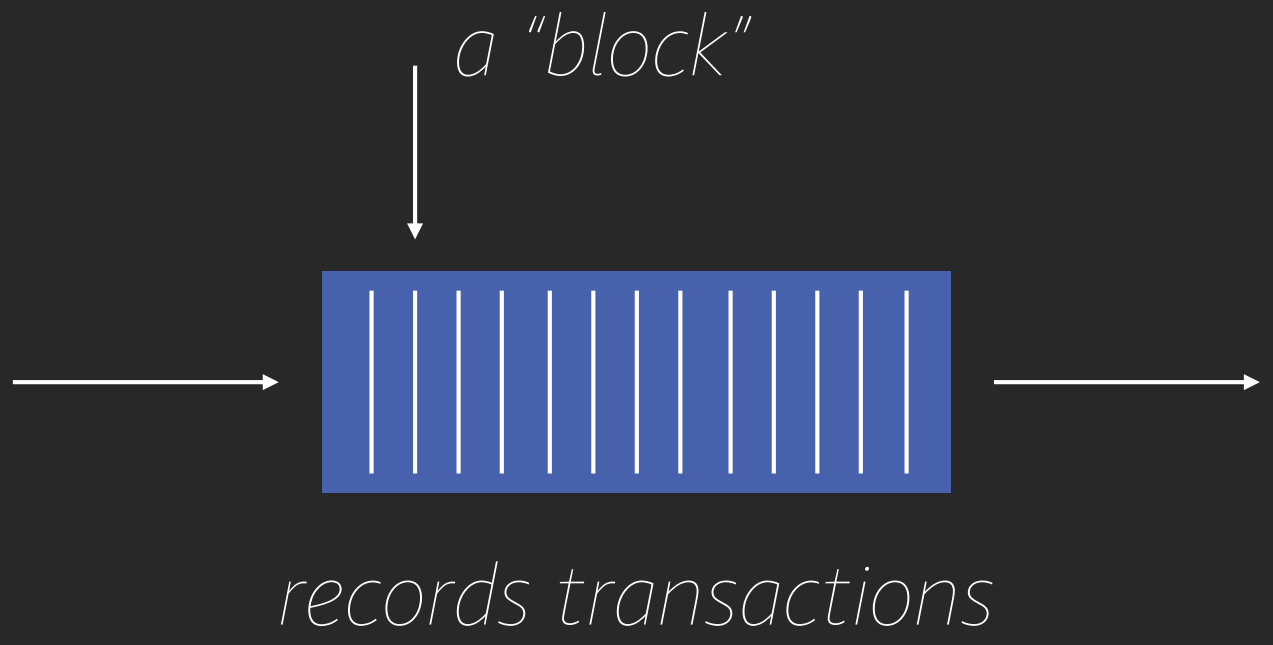
Supercharged log

Paxos at the bottom

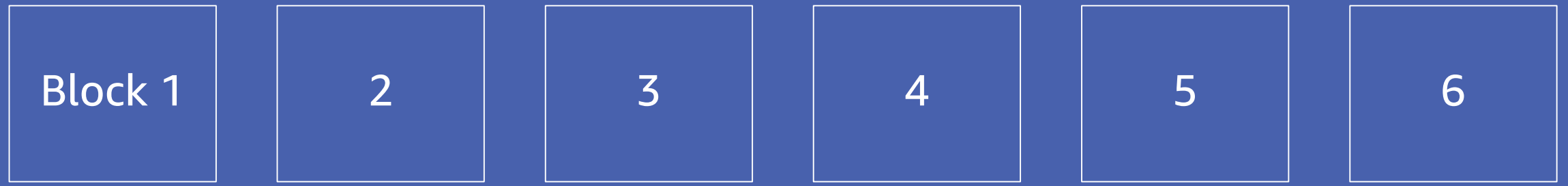
Infinite retention

Stable APIs

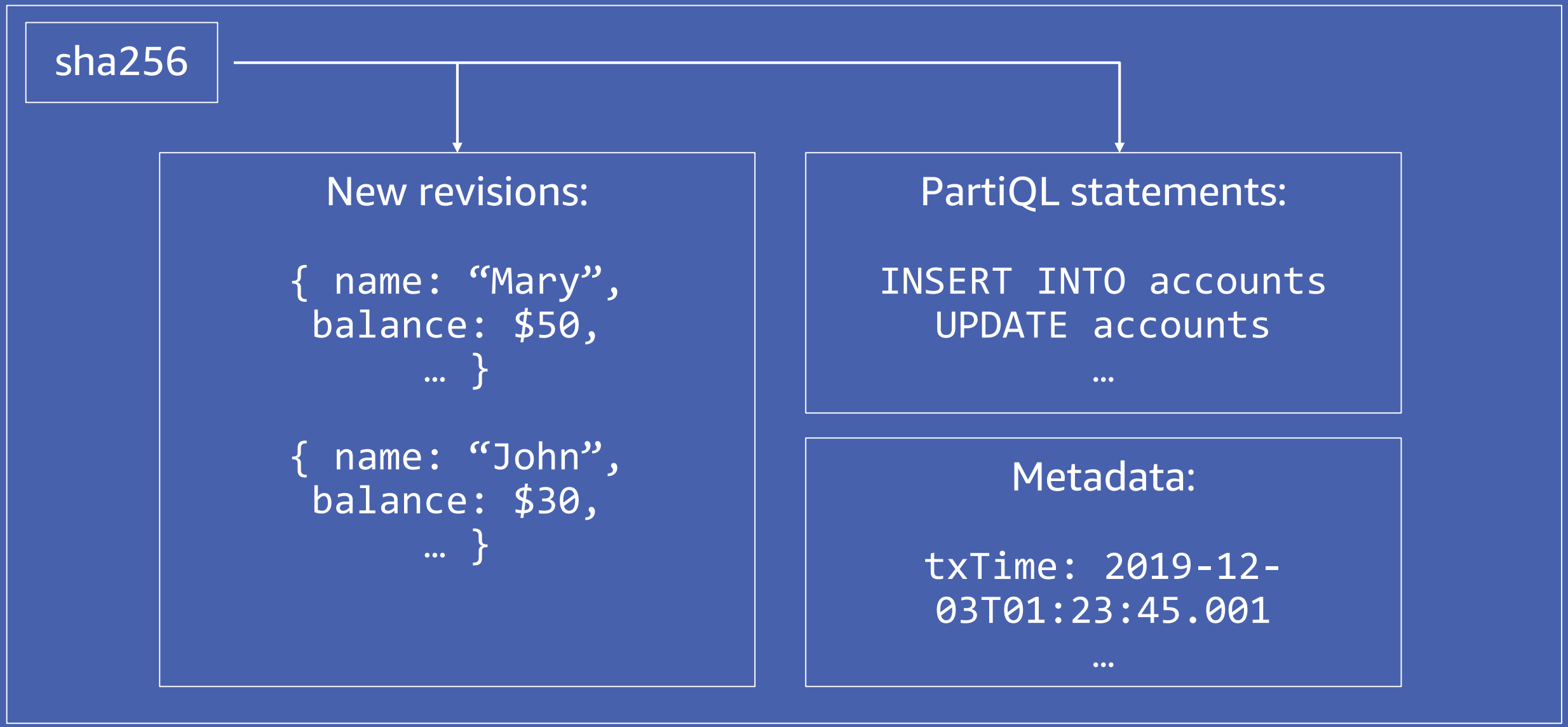
Highly available



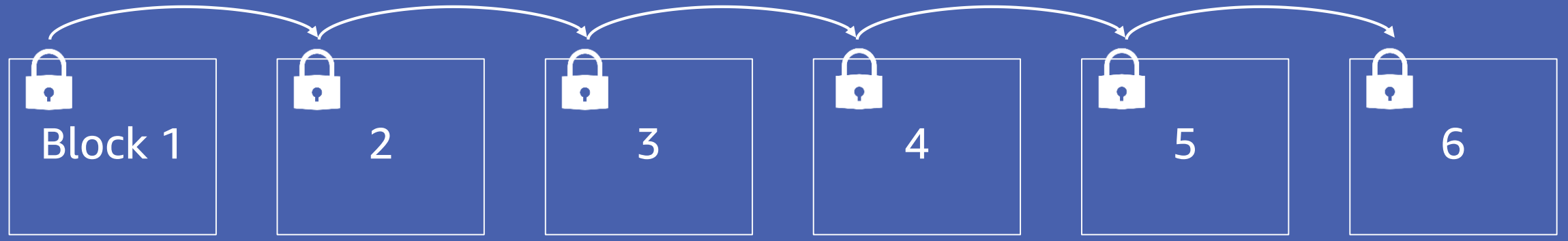
Blocks in the QLDB Journal



A Block in the QLDB Journal

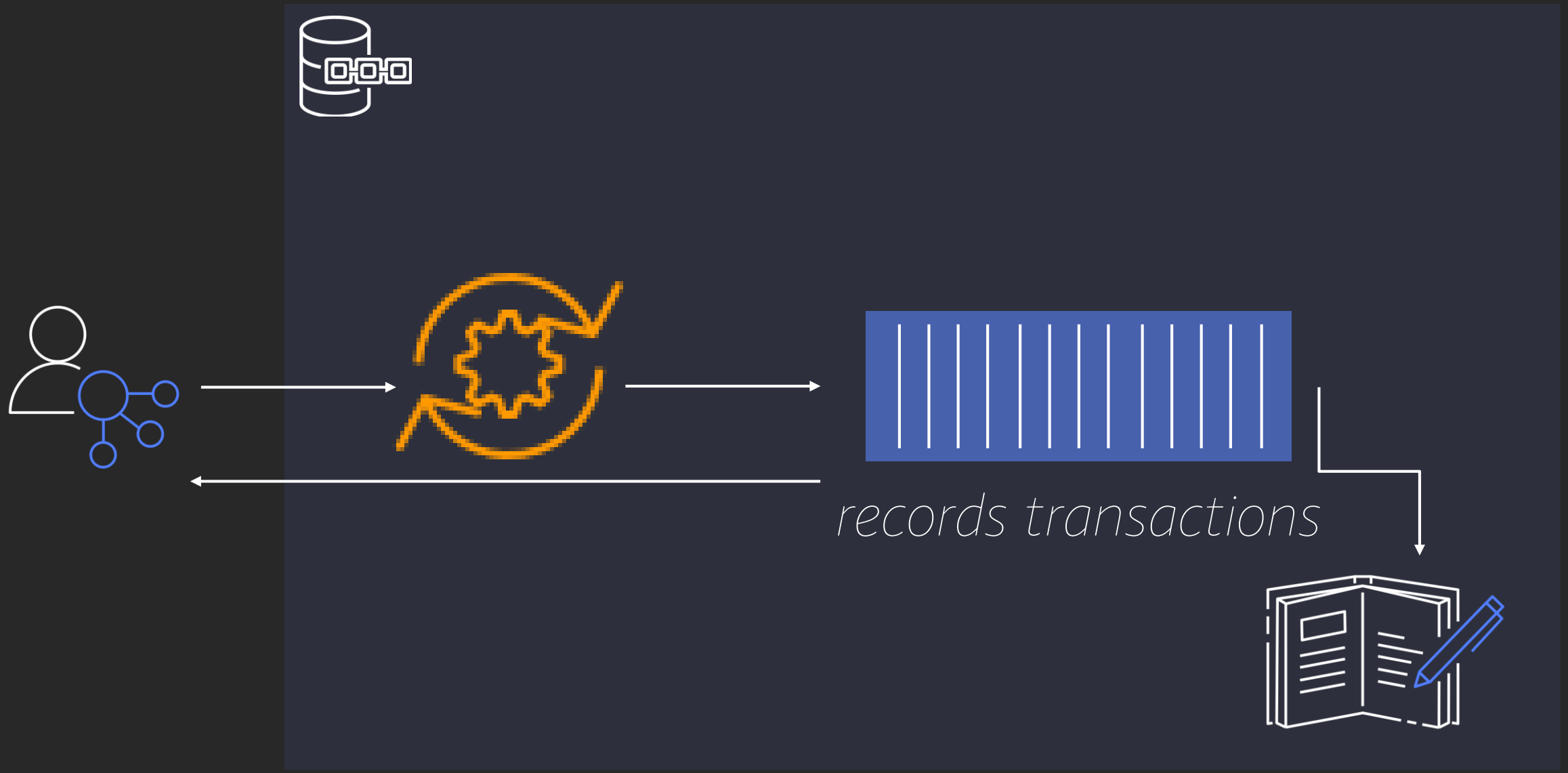


Blocks in the QLDB Journal

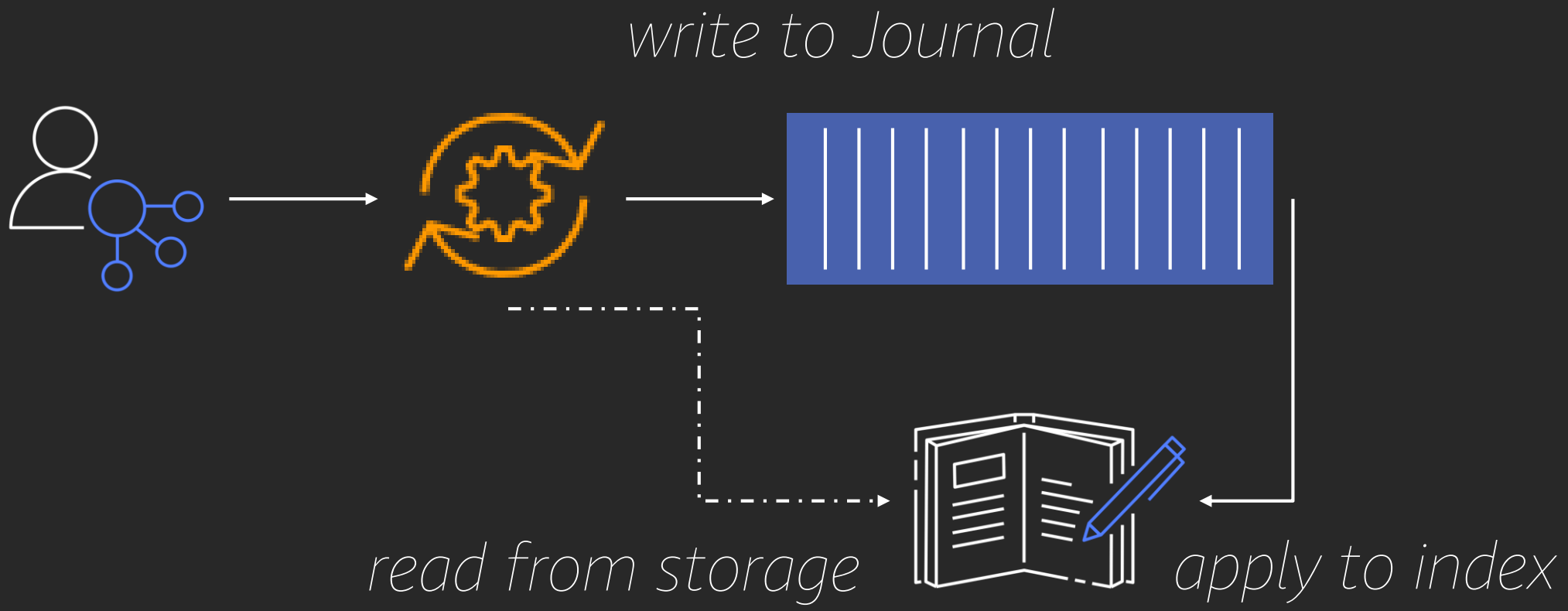


The Journal

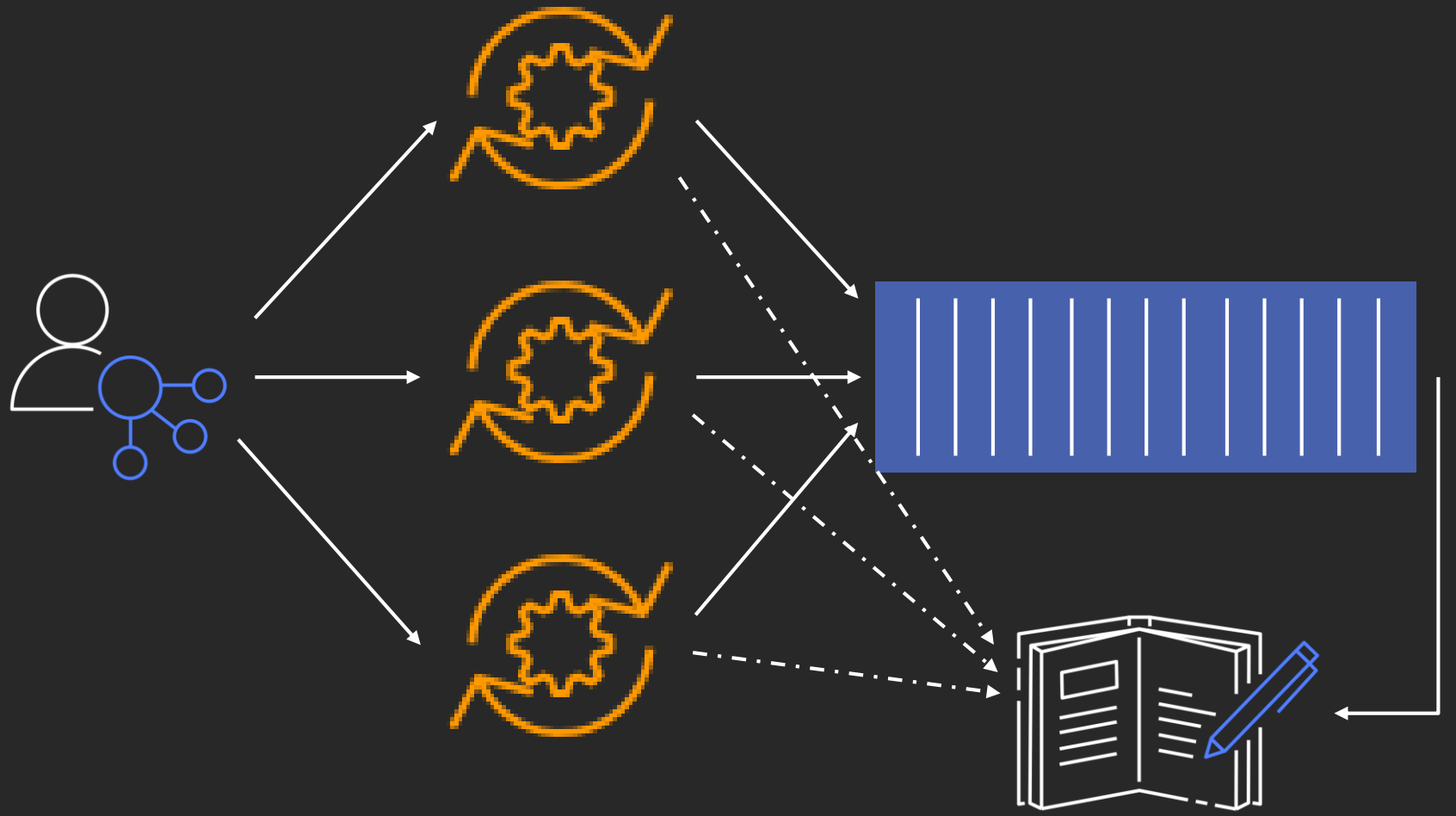
The QLDB architecture



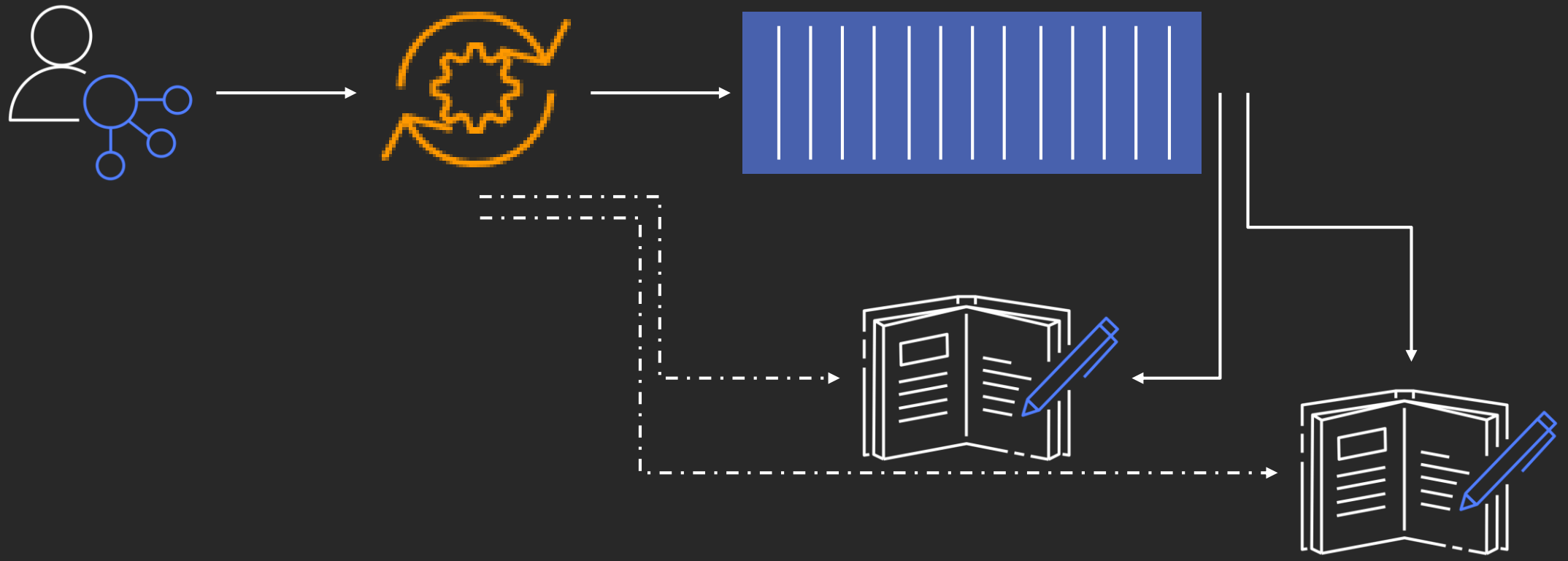
The QLDB architecture



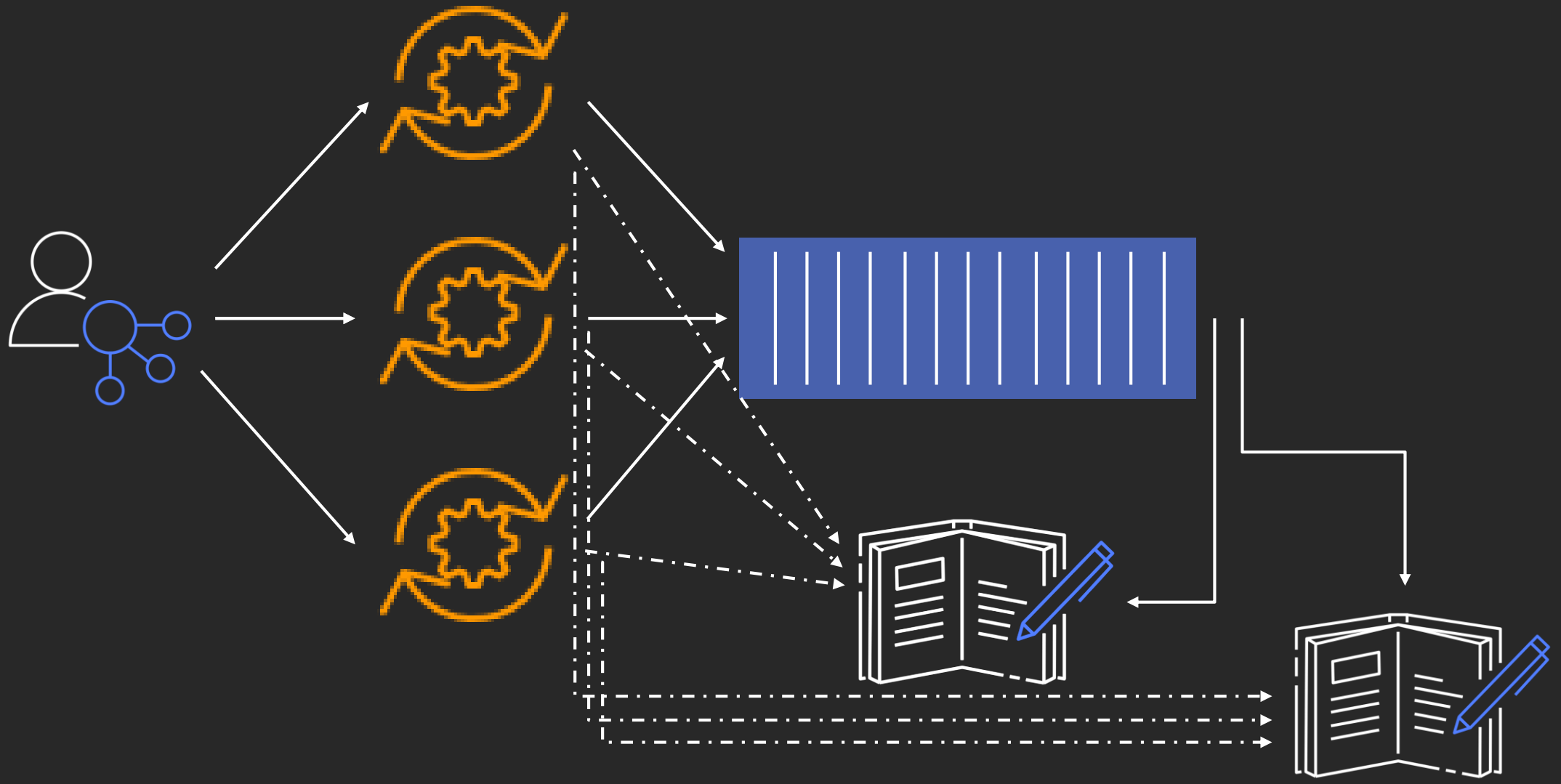
The QLDB architecture – scale out



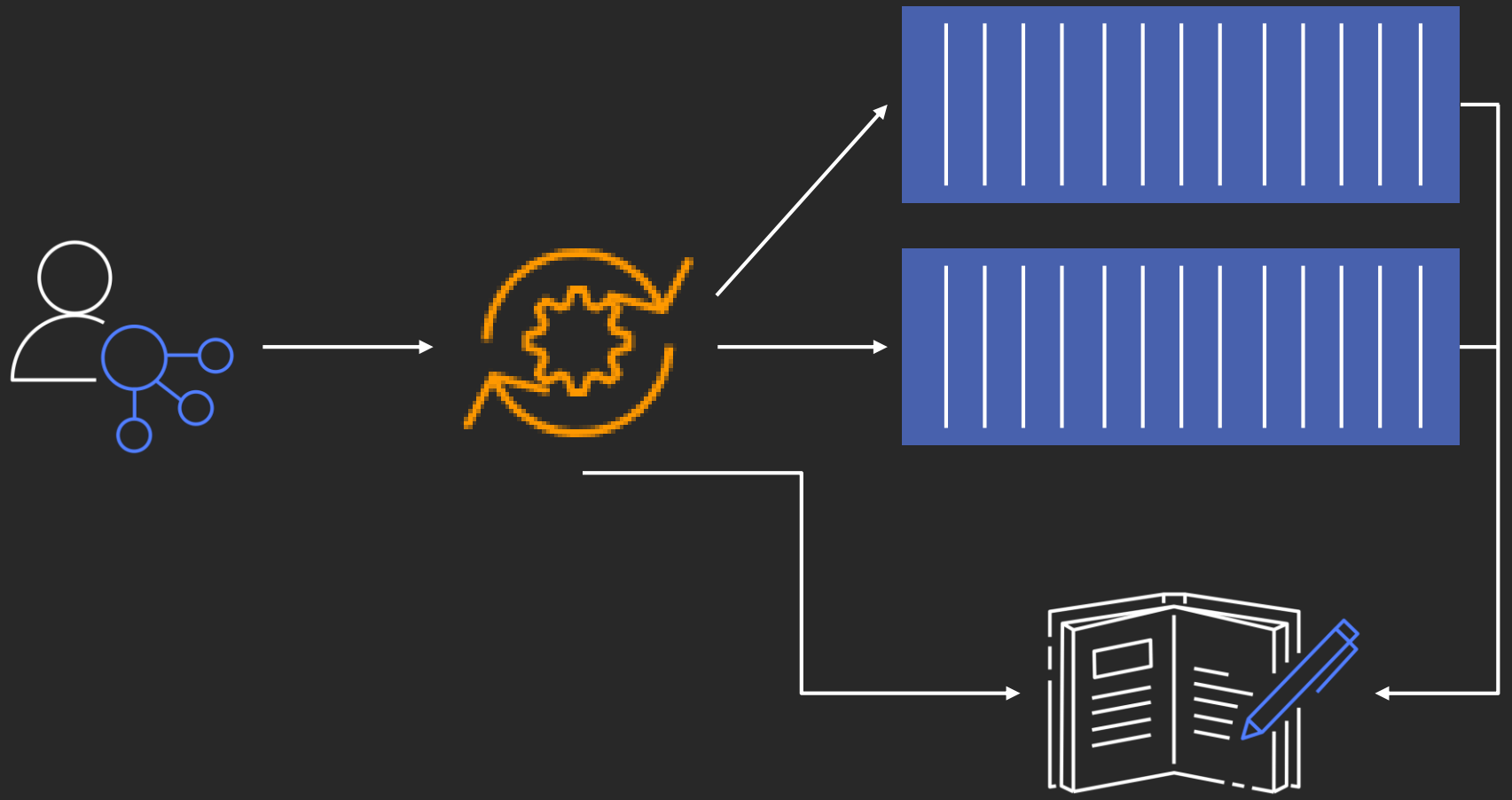
The QLDB architecture – read scaling



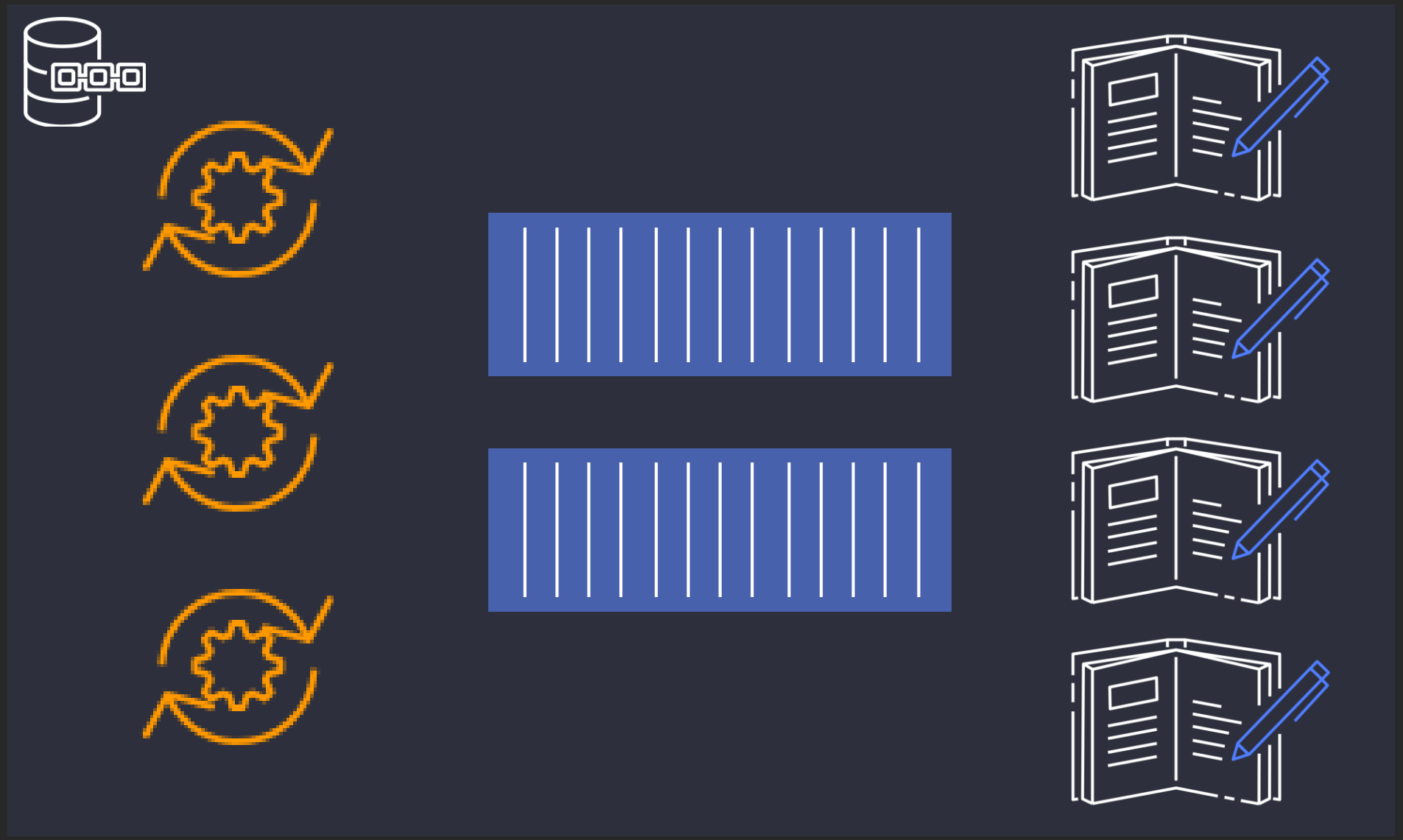
The QLDB architecture – read scaling



The QLDB architecture – write scaling



The QLDB architecture



How best to take advantage of QLDB

Bank setup

```
CREATE TABLE accounts
CREATE INDEX ON accounts (name)

CREATE TABLE transactions
CREATE INDEX ON transactions (id)
CREATE INDEX ON transactions (from)
CREATE INDEX ON transactions (to)
```

Register some accounts

```
INSERT INTO accounts <<
 {‘name’:’Mary’,’balance’:0},
 {‘name’:’John’,’balance’:0}
>>
```

Transfer money

```
SELECT balance FROM accounts
WHERE name IN [‘Mary’,’John’]

UPDATE accounts SET balance = ? WHERE name = ?

INSERT INTO transactions <<
 {‘from’:’John’,to:’Mary’,amount:50,date:2019-12-02T}
>>
```

Concurrency

1. Read John’s balance
2. Update John’s balance to 0
3. Increment Mary’s balance

1. Read John’s balance
2. Update John’s balance to 0
3. Increment Richard’s balance

John will have $0

Mary and Richard will each gain $50

Concurrency with locks

1. SELECT … FOR UPDATE
2. Read John’s balance
3. Update John’s balance to 0
4. Increment Mary’s balance

1. SELECT … FOR UPDATE
2. Wait for lock!
3. Read John’s balance
4. Fail – not enough money

John will have $0

Only Mary will gain $50

Optimistic concurrency control

1. Read John’s balance
2. Update John’s balance to 0
3. Increment Mary’s balance
4. Commit

1. Read John’s balance
2. Update John’s balance to 0
3. Increment Richard’s balance
4. Commit will be rejected (OCC)
5. Retry, not enough money

John will have $0

Only Mary will gain $50

History

```
SELECT id FROM accounts BY id
WHERE name = ‘Maria’

SELECT h.data.name, h.metadata.txTime
FROM history(accounts) AS h
WHERE h.metadata.id = ?
```

Exports

Amazon QLDB > Create export job

Create export job

Choose the date range of journal blocks that you want to export from a ledger. QLDB will write the data into an S3 bucket. Learn more

Journal blocks

Ledger

Choose a ledger

Start date and time (UTC)

YYYY/MM/DD *hh:mm:ss*

End date and time (UTC)

YYYY/MM/DD *hh:mm:ss*

Write journal blocks to S3

Destination for journal blocks

Enter a destination in Amazon S3 where your journal blocks will be stored. Amazon S3 is object storage built to store and retrieve any amount of data from anywhere. **Learn more**

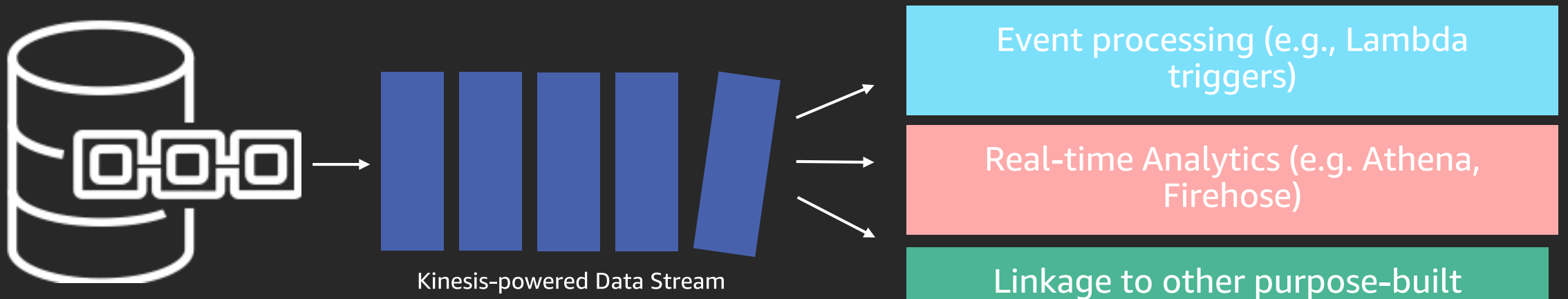
s3://bucket/prefix **Browse S3**

Format: s3://bucket/prefix. For example, if you specify s3://carzz/ledgerexport1/ and QLDB exports a file named 2019/03/01/33001724-5d16-d430-c1d1-524d97ca7f88.1-3.ion, the S3 console shows a folder named ledgerexport1 with the object 2019/03/01/33001724-5d16-d430-c1d1-524d97ca7f88.1-3.ion

S3 encryption

- (●) Bucket default encryption
- ( ) AES-256
  Use server-side encryption with Amazon S3 managed keys (SSE-S3)
- ( ) AWS-KMS
  Use server-side encryption with AWS KMS managed keys (SSE-KMS)

Announcing Private Preview of QLDB Streaming



To get started, email: qldb-outbound@amazon.com

Related Breakouts

BLC203 – Why you need a ledger database: BMW, DVLA, and Sage discuss their use cases (Breakout)

Wednesday, Dec 4, 11:30 AM - 12:30 PM – MGM, Level 3, Chairman's Ballroom 366

DAT380 – Amazon QLDB: An engineer's deep dive on why this is a game changer (Breakout)

Wednesday, Dec 4, 2:30 PM - 3:30 PM – Venetian, Level 4, Delfino 4005

GPSTEC339 – Integrate Managed Blockchain & Amazon QLDB in your solutions portfolio (Breakout)

12/4, 1:00 PM, Bellagio, Grand Ballroom 5 Green

DAT352 – Building applications on Amazon QLDB (Workshop)

Thursday, Dec 5, 12:15 PM - 2:30 PM – Venetian, Level 4, Delfino 4003

BLC209 – Asset provenance ledger system based on Amazon QLDB: BMW's use case (Chalk talk)

Thursday, Dec 5, 1:00 PM - 2:00 PM – Venetian, Level 3, Murano 3301B

DAT351 – Demo: Building a DMV application with Amazon QLDB (Builders Session)

Tuesday, Dec 3, 4:45 PM - 5:45 PM – Aria, Level 1 West, Bristlecone 4 - Table 7

DAT378 – Demo: Common PartiQL operations in Amazon QLDB (Builders Session)

Wednesday, Dec 4, 11:30 AM - 12:30 PM – Mirage, Grand Ballroom B - Table 9

Meet us at our Booth

- Come talk to our engineering and product teams and get answers to your questions
- Answer some trivia, and win a QLDB shirt!
- Venetian Expo Hall, AWS Village, Blockchain and Ledger Booth



Thank you!

Marc Bowes

@marcbowes

AWS re:Invent

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.

aws



Please complete the session survey in the mobile app.

© 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved.