

# Building improved Data Integrity and Data Exchange Systems with Blockchain and Ledger Technologies

## Amazon Web Services

Presented to: ATIEC 2019

By: Lana Kalashnyk, Principal Blockchain Architect  
Ikalash@amazon.com

Date: ATIEC 2019

*Aviation Information World – Forecasting the Future*



# Let's discuss

- **Blockchain vs a Ledger Technology**
- **Amazon Managed Blockchain**
- **Amazon Quantum Ledger Database**
- **Customer Success Stories**



# Blockchain vs a Ledger

# How do we think about blockchain?

Home > Open Access News > Blockchain News > Blockchain: healthcare's next frontier, or so much hype?

Open Access News Blockchain News

## Blockchain: healthcare's next frontier, or so much hype?

June 25, 2018

Home > Emerging Technology

NEWS ANALYSIS

## Blockchain will be the killer app for supply chain management in 2018

The distributed ledger technology that underpins cryptocurrencies is now poised to disrupt supply chain management – especially in the global shipping industry.

15 JULY 2018 | ARTICLES

## Blockchain Logistics – Changing the World or Just Marketing Hype?

## Blockchain is this year's buzzword – but can it outlive the hype?

The open-source ledger behind bitcoin is touted as revolutionary for everything from banking to health, but the jury is still out



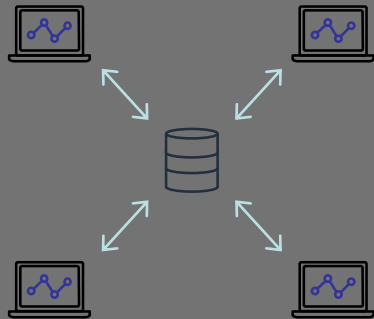
Aviation Information World - Forecasting the Future



# Need for a ledger with centralized trust

•1

•LEDGERS WITH  
CENTRALIZED TRUST



•2

•TRANSACTIONS WITH  
DECENTRALIZED TRUST



•Healthcare

•Verify and track hospital  
equipment inventory



•DMV

•Track vehicle  
title history



•Manufacturers

•Track distribution of a  
recalled product



•HR & Payroll

•Track changes to an  
individual's profile

ATHEC

Aviation Information World - Forecasting the Future

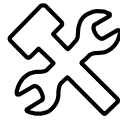




# Challenges customers face



- Resource intensive



- Difficult to manage and scale



- Error prone and incomplete



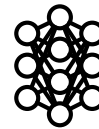
- Impossible to verify

## •Building ledgers with traditional databases

## •Blockchain approaches

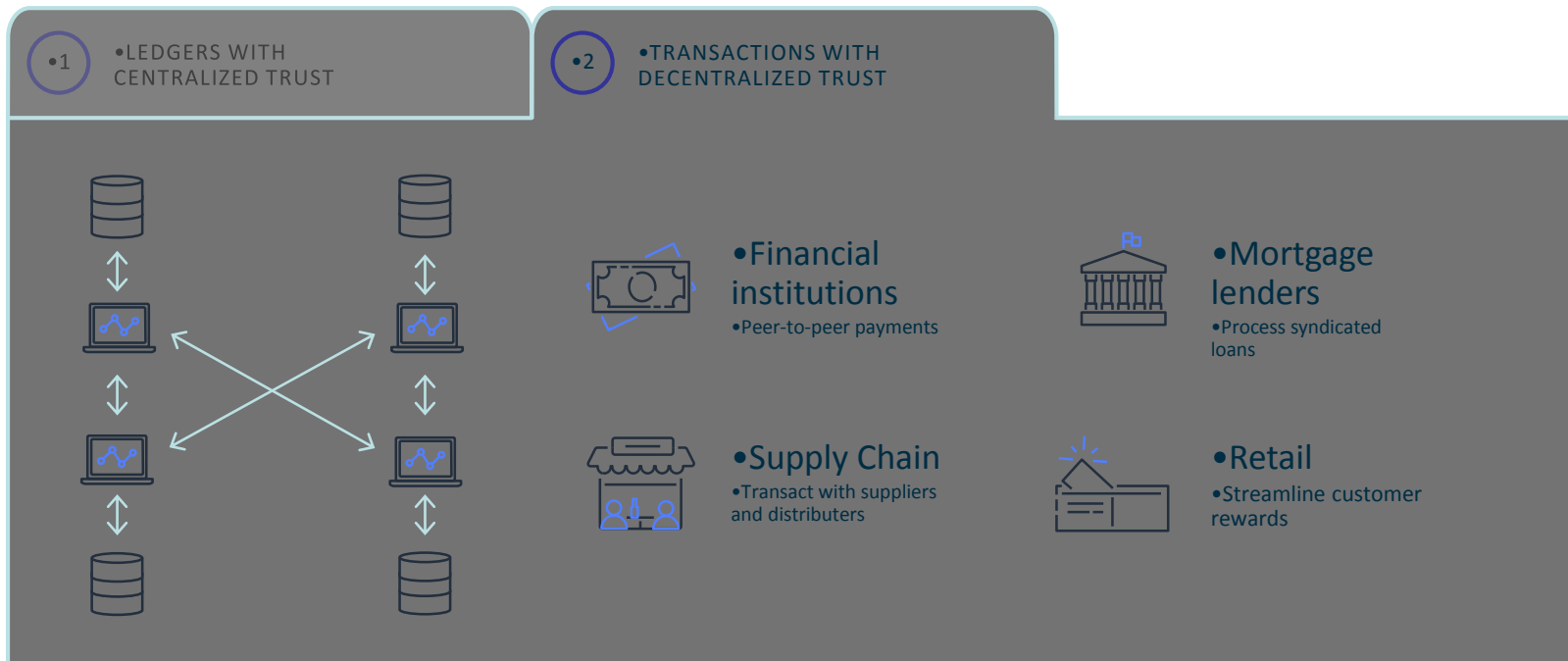


- Designed for a different purpose



- Adds unnecessary complexity

# Need for running transactions with decentralized trust

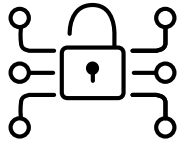


# Complexity of Multi-Party Businesses

- **Multi-party businesses could achieve better outcomes by sharing information, but need:**
- **A way to independently verify transactions**
- **Single, current and accurate view of data with tamper-proof history of transactions**
- **To this end, organizations with multi-party business use:**
- **Central authorities to securely and fairly share data and**
- **Employ costly escrow process for asset transfers.**



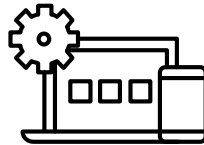
# Blockchain



## What is it

---

Linked transaction data in encrypted, redundant databases, or ledgers, hosted across the Internet



## What it does

---

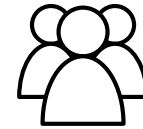
Makes online transactions across enterprises more secure and trustworthy  
Lowers cost by eliminating the need for traditional intermediaries



## How

---

Encrypted, redundant data prevents destruction or falsification of data in any single ledger

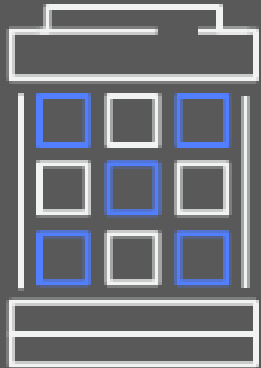


## Who

---

No single entity controls the data, further reducing risk

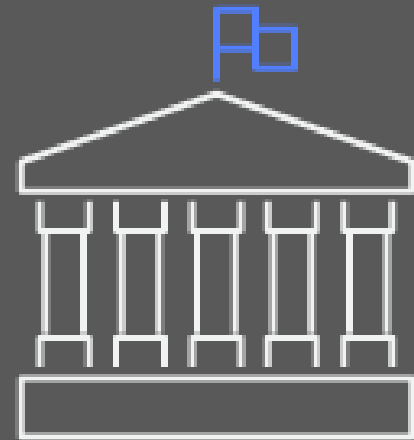
# Perspectives



Business



Technical



Legal

# Typical Distributed Application Stack

**Applications**

**APIs**

Membership

Cryptographic

Events

Transactions

**Consensus**

**Smart Contracts**

**Distributed Ledger**



Aviation Information World - Forecasting the Future



# Permissioned or Permissionless

## •Permissioned

- Users enrolled before transactions
- Identifiable users
- Trace transactions to users

## •Permissionless

- Anonymous
- Anyone can perform transactions
- Commonly restricted to operations on own data



# Blockchain components: “smart contracts”

- Rules embedded in app
- Verified execution of code
- Conditional operators
- Application writes to ledger
- Contract can interact with components outside of the blockchain network (off-chain)

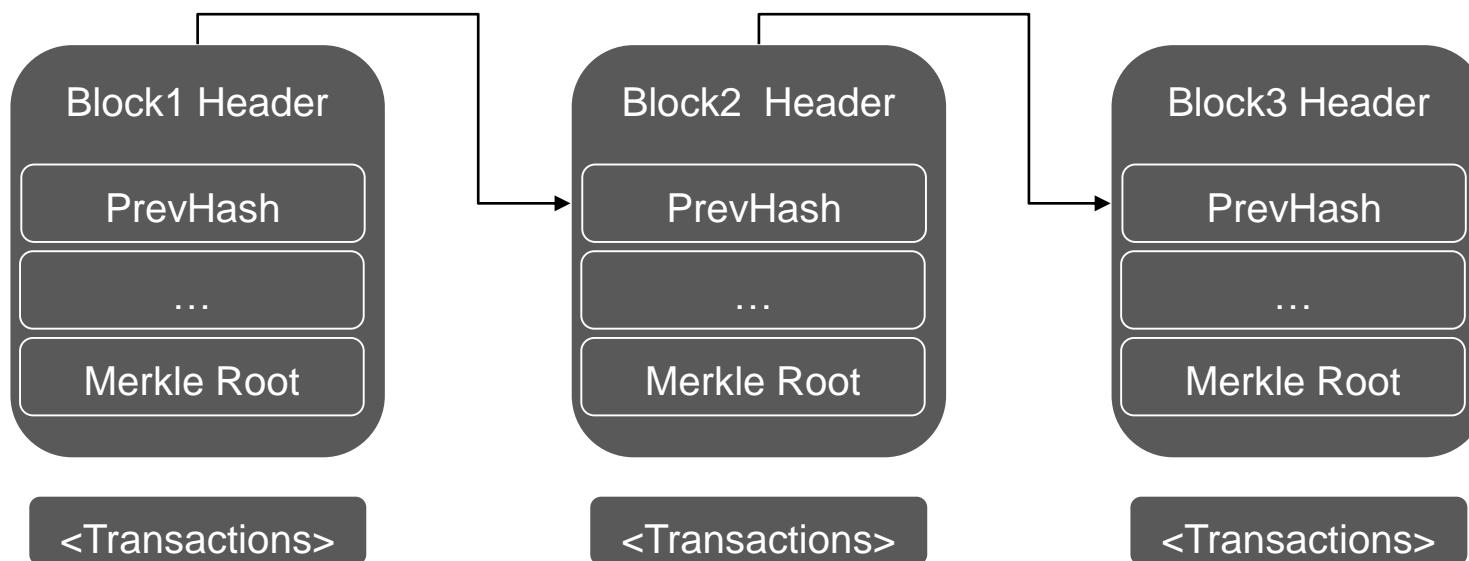




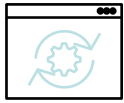
# Consensus Algorithms

- **Proof of Work**
- **Proof of Stake**
- **Proof of Authority**
- **Proof of Elapsed Time (PoET)**
- **Endorsement, Ordering, Validation (PBFT)**

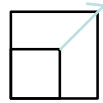
# Blockchain Components: Sample Distributed Ledger



# Challenges with existing blockchain solutions



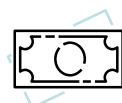
Setup is hard



Hard to scale



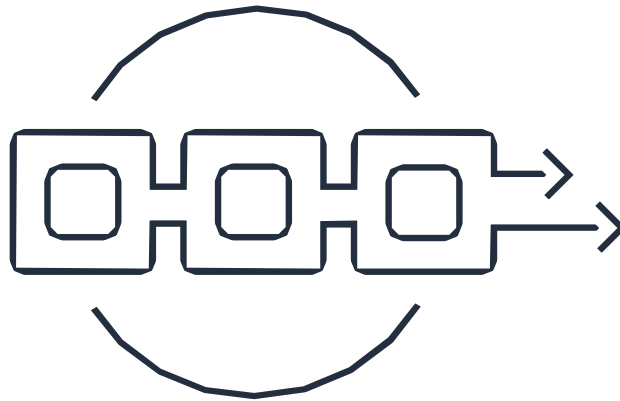
Complicated to  
manage



Expensive

# Amazon Managed Blockchain

# What is Amazon Managed Blockchain?



Amazon Managed Blockchain is a fully managed service that makes it easy to create and manage scalable blockchain networks using popular open source frameworks:

**Hyperledger Fabric and Ethereum**



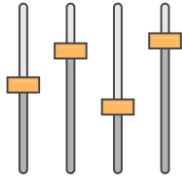
**ATHEC**

Aviation Information World - Forecasting the Future





# Amazon Managed Blockchain



## Fully managed

Create a blockchain network in minutes

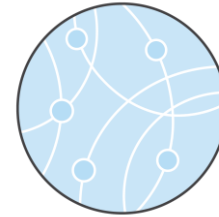
HYPERLEDGER

FABRIC

ethereum

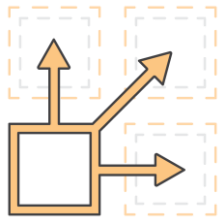
## Open-source variety

Support for two frameworks



## Decentralized

Democratically govern the network



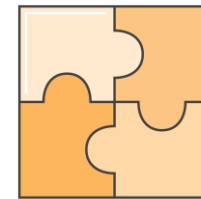
## Reliable & scalable

Backed with Amazon QLDB technology



## Low cost

Only pay for resources used



## Integrated

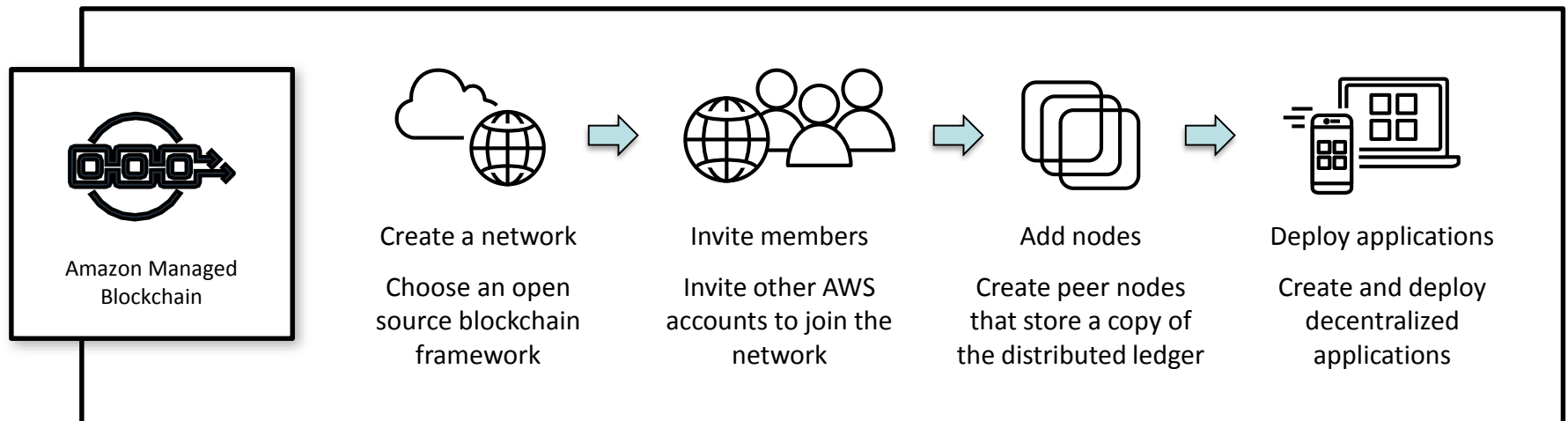
Send data to Amazon QLDB  
for secure analytics

ATHEC

Aviation Information World - Forecasting the Future



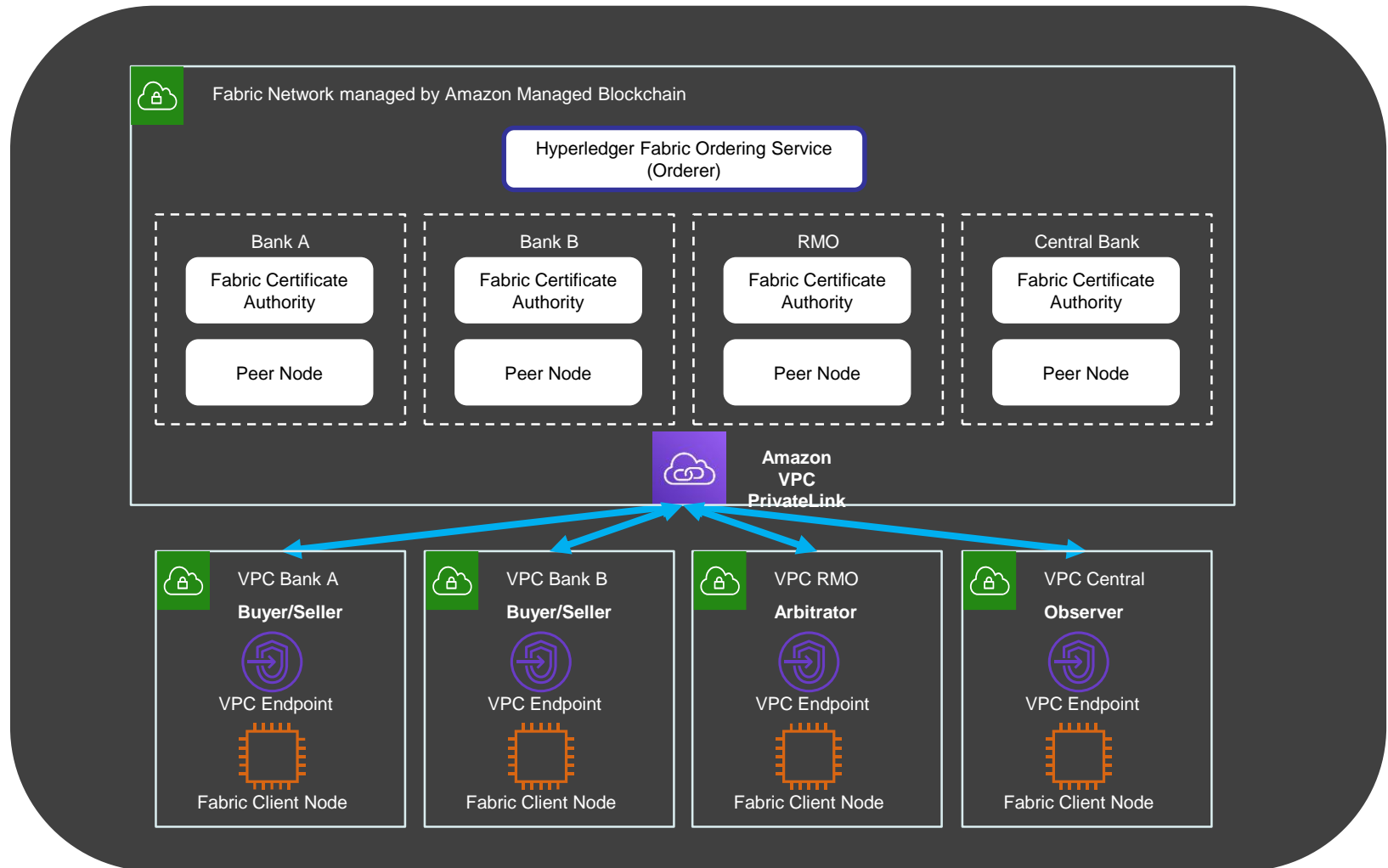
# How Amazon Managed Blockchain works



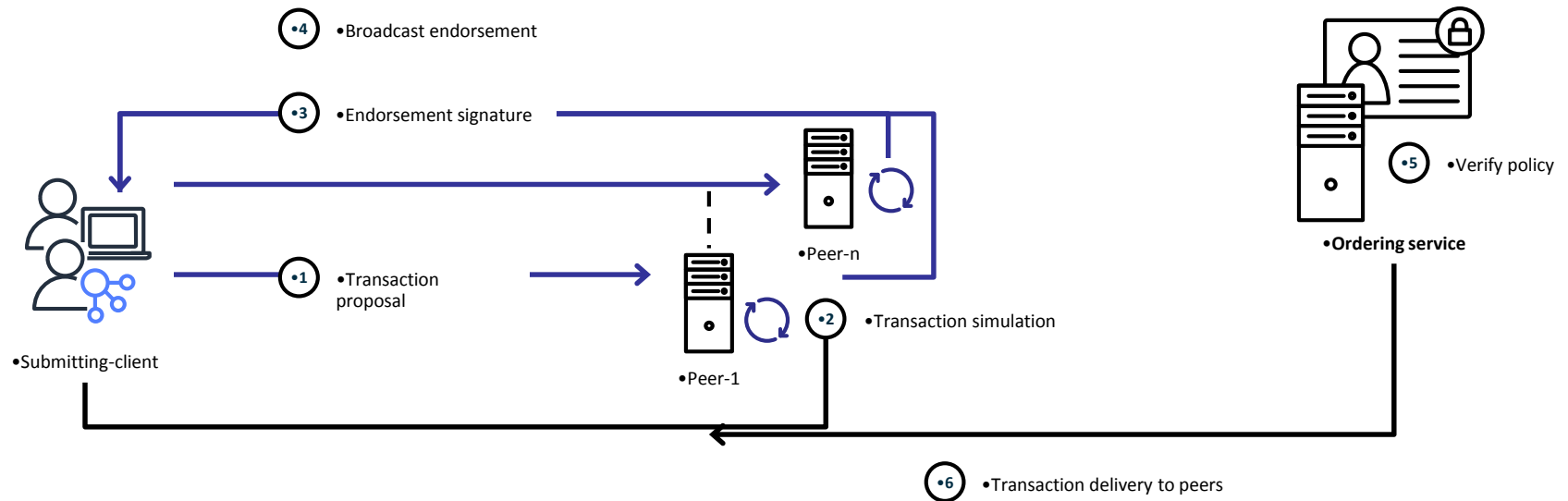
# Who "owns" the network?

- Networks are decentralized and can remain active even after the initial creator leaves
- Inviting members to join
  - Preview: network creator can invite
  - GA: members vote on who to invite and remove
- Network-wide settings
  - GA: members can vote on network-wide settings and configure the actual voting rules (e.g., majority rules or one member decides)
- Each member pays for their resources
- Amazon Managed Blockchain manages shared components like the ordering service and networking settings

# Amazon Managed Blockchain



# Transaction flow with Hyperledger Fabric





# Augmented Hyperledger Fabric

## Ordering service

Core component of a Fabric network to guarantee delivery and order of transactions

Production grade networks using open source will utilize Apache Kafka for this component

Managed Blockchain uses Amazon QLDB technology, increasing durability and reliability

## Certificate authority

Open source uses a “soft” HSM

Managed Blockchain uses AWS Key Management Service (AWS KMS) to secure the Certificate Authority service

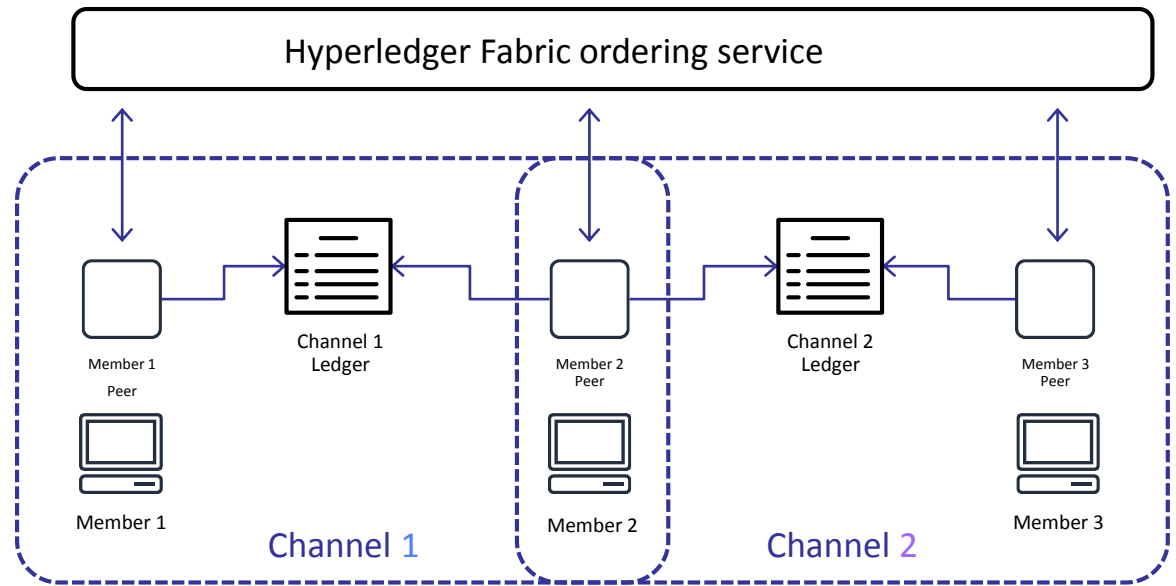


# Channels and private data for access control

**Channels allow isolation of transactions among specific members in the network**

**Create or update a channel with configuration transaction (configtx)**

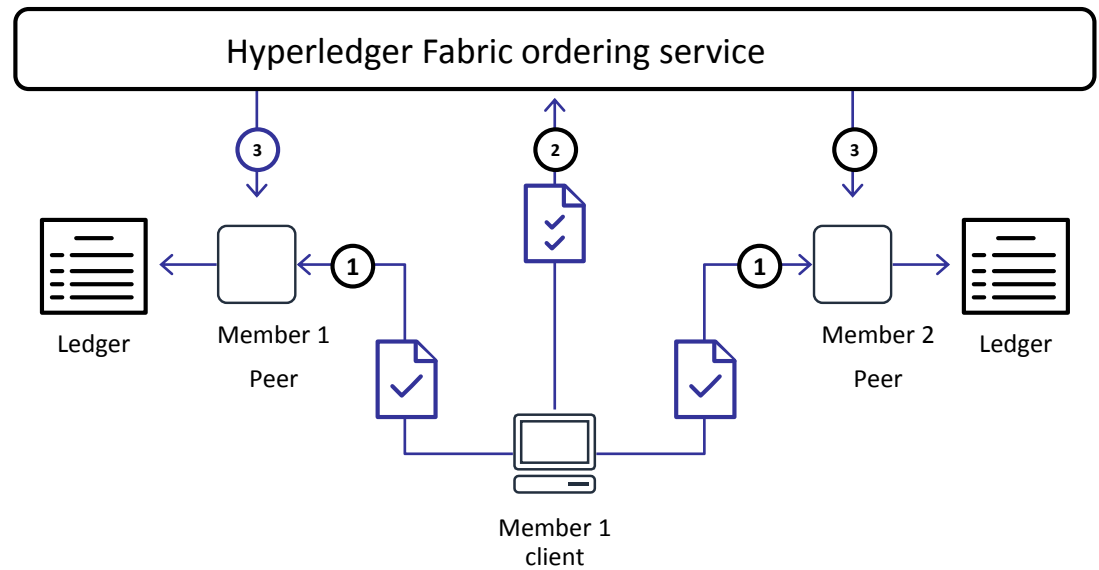
**Private data enables sub-channel access control**



# Endorsement policies

**Endorsement policies allow chaincode to specify which members (or how many) need to validate a transaction before submitting**

**Endorsed transactions then get submitted to the ordering service and assembled into blocks**







# Nestle

## Problem

Nestlé is committed to bringing transparency into the origin and quality of the ingredients used in their products, and wants their customers to have visibility into the end-to-end supply chain for their single origin coffee.

## Solution

AWS Professional Services built supply-chain asset-tracking smart contracts to track single origin coffee from farm to customer on Amazon Managed Blockchain network, and exposed the contracts via a RESTful API. Nestlé's mobile app consumes the API to capture events as the coffee moves through the supply chain.

## Impact

Nestlé and their customers can now track the high quality single origin coffee from farmer to customer. Nestlé now has a platform they can expand to and trace the provenance of other products from their brand portfolio.

# Sony Music Entertainment Japan - Music Rights

Sony Music Entertainment Japan (SMEJ) is committed to helping musicians and artists by removing undifferentiated heavy lifting such as filing and processing content rights and allow artists to focus more time on producing their work

## Solution

Using SMEJ' system on AWS, participants will be able to share and verify information such as date and time of creation, and the author's details and automatically verify the rights generation of any piece of written work.

## Outcomes

The system is expected to improve productivity while maintaining proper rights processing, creating an environment where new generations of creators can launch hit content.





# Legal & General Reinsurance

## Pension Risk Transfer platform

Legal & General picked Amazon Managed Blockchain for their global Pension Risk Transfer (PRT) ecosystem . This is a single ecosystem capable of driving every stage of the PRT reinsurance value chain including pricing, claims handling, financial reporting and collateral, utilising data dynamically stored on the blockchain

### Solution

With Amazon Managed Blockchain, Legal and General is able to create a solution that addresses not only the greater speeds at which risks are transacted but also drives transparency and security in an increasingly interconnected market. This platform replaces multiple processes and systems traditionally used to support each function, with the added security of blockchain technology.

### Impact

Legal & General' platform enables the Group to provide excellent service to customers in multiple markets at lower costs, redefining the way long term life reinsurance business is sold and managed. All Legal & General Reinsurance clients will eventually be supported on this platform.



# Amazon Managed Blockchain Customers










Aviation Information World - Forecasting the Future



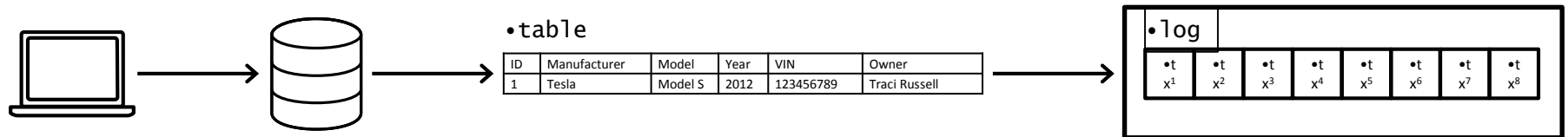
# Amazon Quantum Ledger Database

# Purpose-built databases at AWS

						
•Relational	•Key-value	•Document	•In-memory	•Graph	•Time-series	•Ledger
<ul style="list-style-type: none"> <li>•Referential integrity, ACID transactions, schema-on-write</li> </ul>	<ul style="list-style-type: none"> <li>•High throughput, low-latency reads and writes, endless scale</li> </ul>	<ul style="list-style-type: none"> <li>•Store documents and quickly access querying on any attribute</li> </ul>	<ul style="list-style-type: none"> <li>•Query by key with microsecond latency</li> </ul>	<ul style="list-style-type: none"> <li>•Quickly and easily create and navigate relationships between data</li> </ul>	<ul style="list-style-type: none"> <li>•Collect, store, and process data sequenced by time</li> </ul>	<ul style="list-style-type: none"> <li>•Complete, immutable, and verifiable history of all changes to application data</li> </ul>
<ul style="list-style-type: none"> <li>•Lift and shift, ERP, CRM, finance</li> </ul>	<ul style="list-style-type: none"> <li>•Real-time bidding, shopping cart, social, product catalog, customer preferences</li> </ul>	<ul style="list-style-type: none"> <li>•Content management, personalization, mobile</li> </ul>	<ul style="list-style-type: none"> <li>•Leaderboards, real-time analytics, caching</li> </ul>	<ul style="list-style-type: none"> <li>•Fraud detection, social networking, recommendation engine</li> </ul>	<ul style="list-style-type: none"> <li>•IoT applications, event tracking</li> </ul>	<ul style="list-style-type: none"> <li>•Systems of record, supply chain, healthcare, registrations, financial</li> </ul>

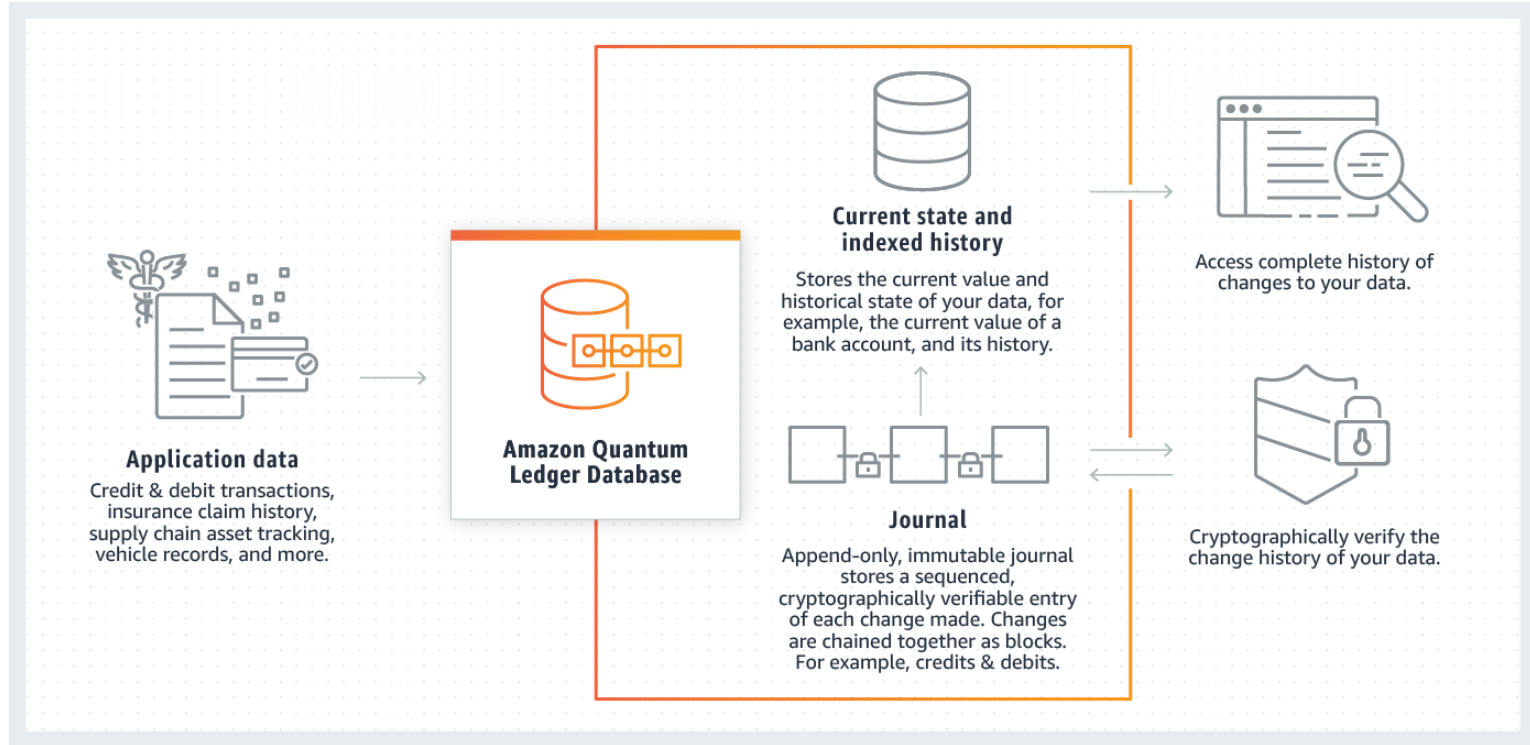
# Traditional database architecture: the log

- Typically an internal implementation
- Used for replicating data
- Difficult, or impossible, to directly access



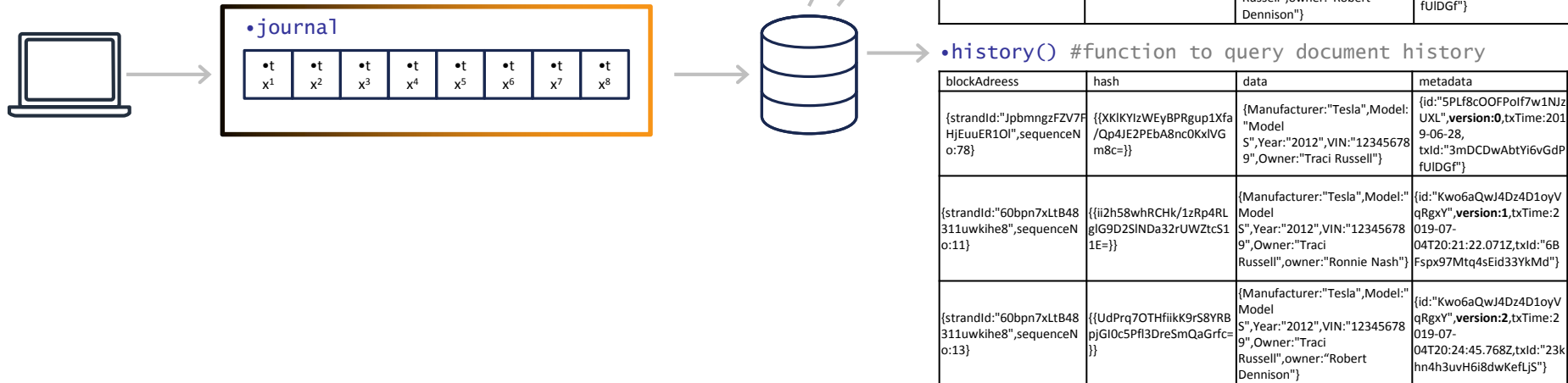


# How Amazon QLDB works

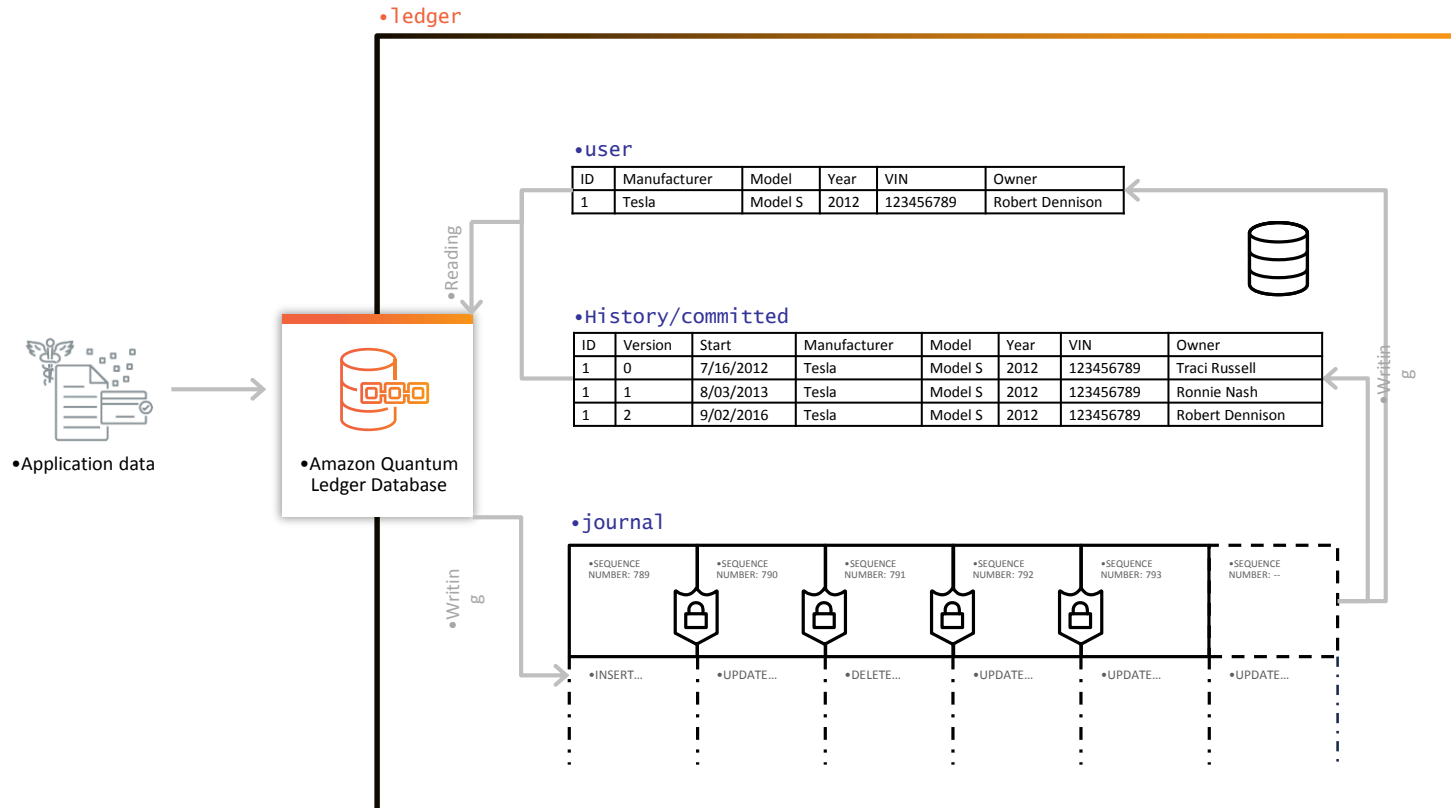


# Amazon QLDB: the journal is the database

- QLDB's journal has structural similarity to a database log
- All writes go to the journal—the journal determines state
- Journal handles concurrency, sequencing, cryptographic verifiability, and availability
- Accessible history of all transactions, document versions, document metadata



# Amazon QLDB: the journal is the database



# Easy to use (SQL)



- **INSERT** INTO cars

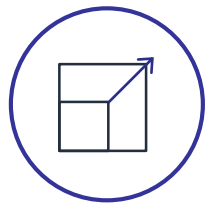
- { 'Manufacturer': 'Tesla',
- 'Model': 'Model S',
- 'Year': 2012,
- 'VIN': 123456789,
- 'Owner': 'Traci Russell'

- **UPDATE** cars **SET** owner = 'Ronnie Nash' **WHERE** VIN = '123456789'

- 

- **SELECT** \* FROM cars

# Serverless, scalable, highly available



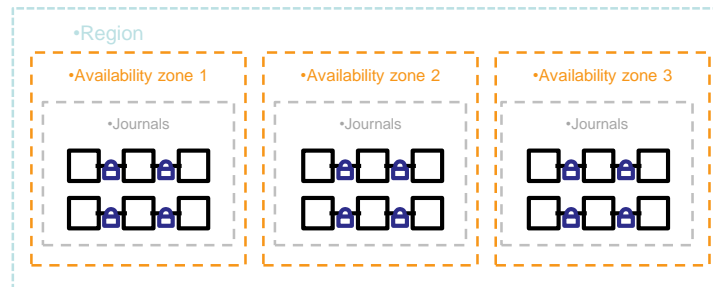
Ledger name

myLedgerName

Create ledger



•Multi-AZ for  
high availability

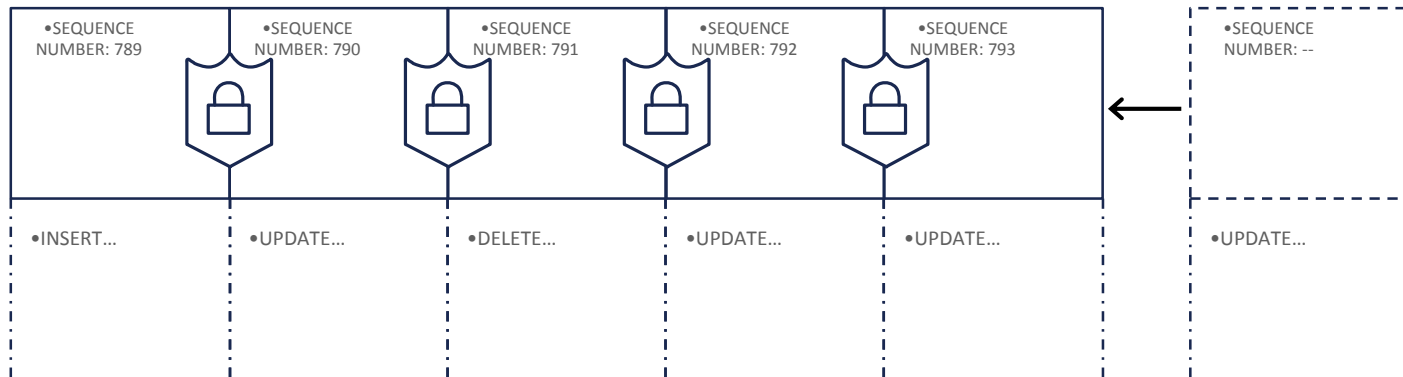


•Multiple copies per AZ  
providing strong durability



# Immutable

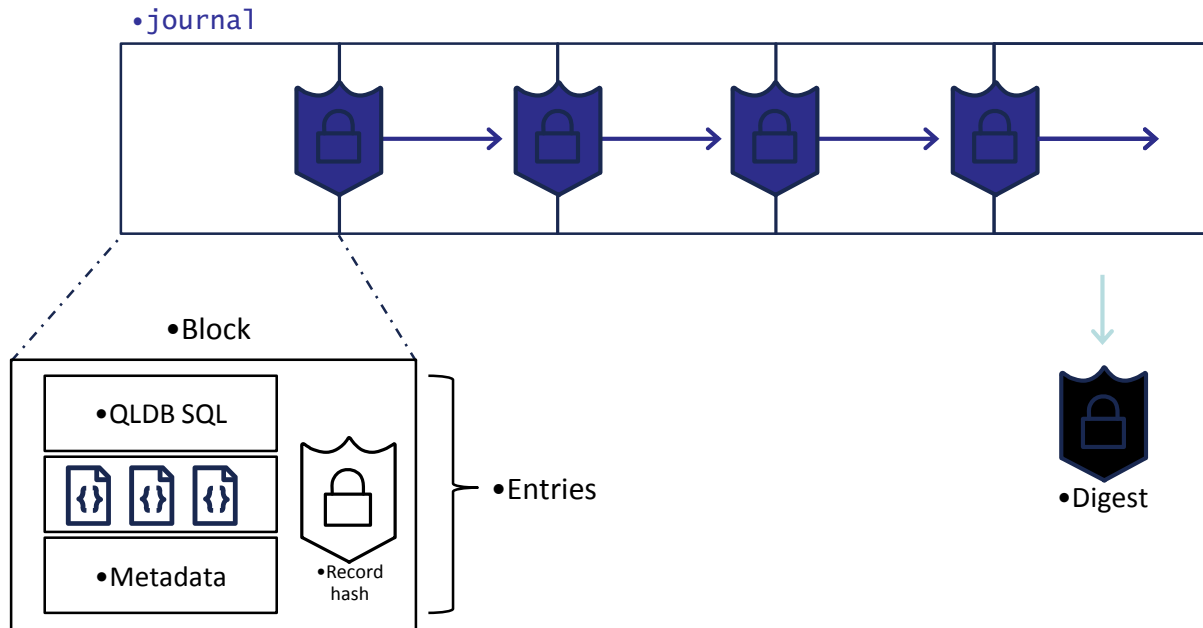
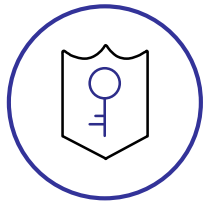
- Records cannot be altered



- The journal is append only and sequenced
- There is no API or other method to alter committed data
- All operations, including deletes, are written to the journal

# Cryptographic verification

- Hash chaining using sha-256



# Amazon QLDB summary

## •Immutable



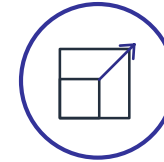
- Append-only, sequenced

## •Cryptographically verifiable



- Hash-chaining provide data integrity

## •Highly scalable



- Serverless, highly available

## •Easy to use



- Familiar SQL operators

## •ACID Transactions



- Fully serializable isolation

## •Journal-first



- The journal is the database



Aviation Information World - Forecasting the Future



# QLDB's data model: Amazon Ion

## data format

```
• vehicle = {  
•   'VIN' :      "KM8SRDHF6EU074761",  
•   'MfgDate' :  "2017-03-01"  
•   'Type' :      "Truck"  
•   'Mfgr' :      "Ford"  
•   'Model' :     "F150"  
•   'Color' :     "Black"  
•   'Specs' : {  
•       'EngSize' : 3.3  
•       'Curbweight': 4878  
•       'HP': 327  
•       'BatterySize': Null  
•   }  
• }
```

• JSON document

• /\* Ion supports comments. \*/

```
• vehicle = {  
•   'VIN' :      "KM8SRDHF6EU074761",  
•   'MfgDate' :  "2017-03-01T"  
•   'Type' :      "Truck"  
•   'Mfgr' :      "Ford"  
•   'Model' :     "F150"  
•   'Color' :     "Black"  
•   'Specs' : {  
•       'EngSize' : 3.3 (decimal)  
•       'Curbweight': 4878 (int)  
•       'HP': 327 (int)  
•       'BatterySize': NULL.int  
•   }  
• }
```

• Ion document

• <https://github.com/amzn/ion-java>



Aviation Information World - Forecasting the Future

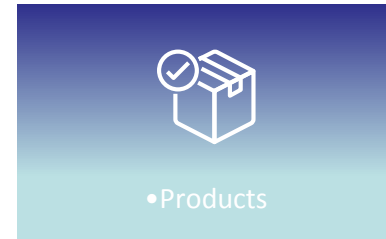
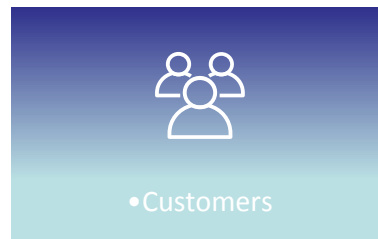
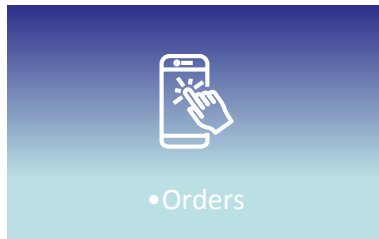






# QLDB's data model: e-commerce data model using Ion

Assume 3 tables



•CREATE TABLE Orders

•CREATE TABLE Customers

•CREATE TABLE Products

# Ledger: Order-System

Nested document structure enables optimal queries and data access



- INSERT INTO orders

- {

- 'order-id' : 100056,

- 

- customer' : {

- 'document-id' : 'some-value',

- 'customer-id': 1000,

- 'first-name' : 'Mike',

- 'last-name' : 'Labib',

- 'address' : '126 Brampton Lane',

- 'city' : 'Chicago',

- 'state' : 'IL'

- },

- 'order-date' : 2019-04-30T,

- 'order-details' : {

- 'item' : {

- 'document-id' : 'some-value',

- 'product-id' : 346211 ,

- 'product-description' : '3 pair socks',

- 'product-color' : 'blue',

- 'price' : 15.00,

- 'quantity' : 2



Aviation Information World - Forecasting the Future



# Ledger: Order-System

## •query

filter on nested doc

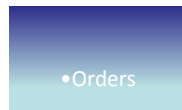
- SELECT o.order-details from orders o
- WHERE o.customer.customer-id = 1000
- AND o.order-id = 100056

Nested document query  
(customer within orders)

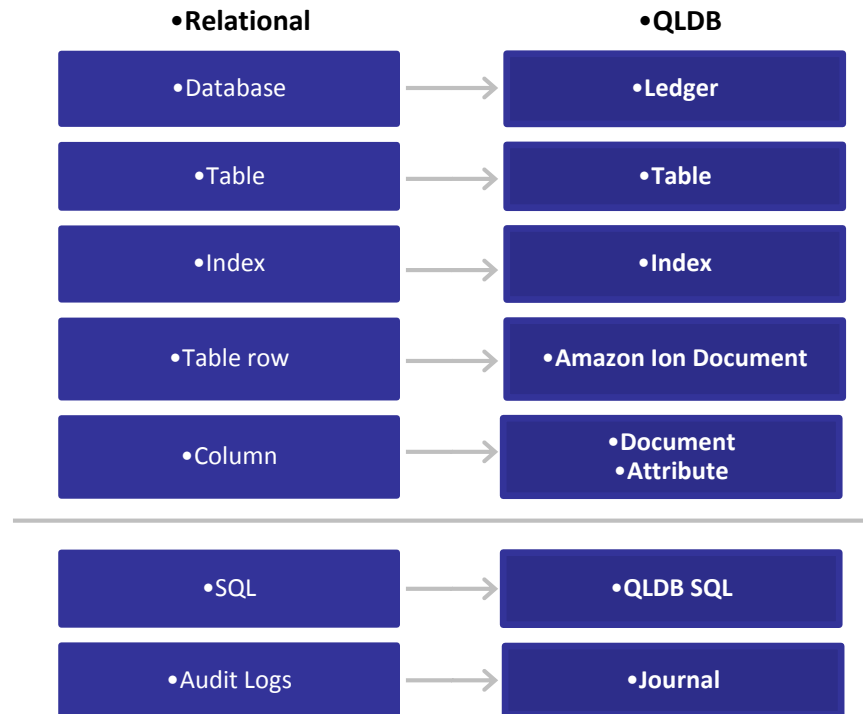


## •result

- { item:
- { 'product-id': 346211,
- 'product-description': '3 pair socks',
- 'product-color': 'blue',
- 'price': 15.00,
- 'quantity': 2
- }
- }



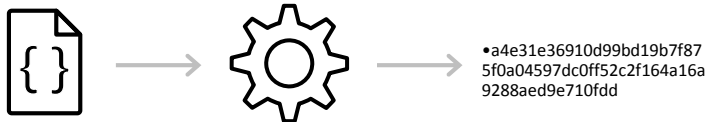
# Mapping constructs between RDBMS & QLDB



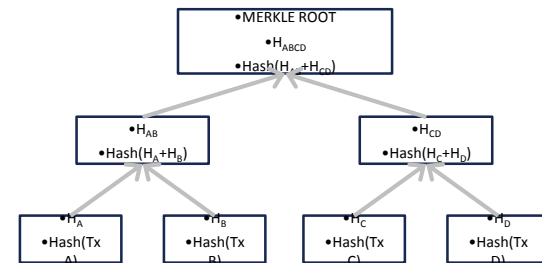
# Deeper look at cryptographic verifiability

- Four basic steps to seeing how QLDB's verifiability works

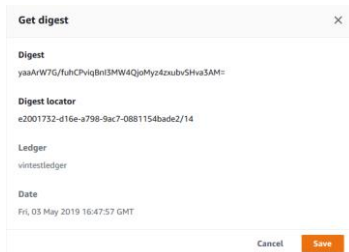
- SHA256: Unique Signature of a document



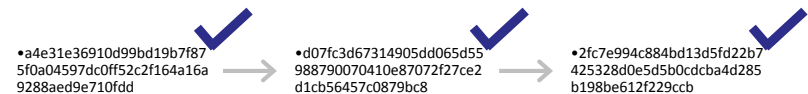
- Merkle Trees: Chaining past hashes together



- Digest: Periodic hash covering all history



- Proof: A chain of hashes that links a document to its digest





# How it works

```
• INSERT INTO cars <<
• { 'Manufacturer': 'Tesla',
•   'Model': 'Model S',
•   'Year': '2012',
•   'VIN': '123456789',
•   'Owner': 'Traci Russell' }
• >>
```

• journal

• J

```
• INSERT cars      • H(T1)
• ID: 1
• Manufacturer: Tesla
• Model: Model S
• Year: 2012
• VIN: 123456789
• Owner: Traci Russell
•
• Metadata: {
•   Date: 07/16/2012
• }
```

• cars

• C

ID	Manufacturer	Model	Year	VIN	Owner

• history()

• H

ID	Version	Start	Manufacturer	Model	Year	VIN	Owner

# How it works

```
•INSERT INTO cars <<
• { 'Manufacturer': 'Tesla',
•   'Model': 'Model S',
•   'Year': '2012',
•   'VIN': '123456789',
•   'Owner': 'Traci Russell' }
• >>
```

•cars

ID	Manufacturer	Model	Year	VIN	Owner
1	Tesla	Model S	2012	123456789	Traci Russell

•history()

ID	Version	Start	Manufacturer	Model	Year	VIN	Owner
1	1	7/16/2012	Tesla	Model S	2012	123456789	Traci Russell

•journal

•J

```
•INSERT cars      •H(T1)
•ID:1
•Manufacturer: Tesla
•Model: Model S
•Year: 2012
•VIN: 123456789
•Owner: Traci Russell
•
•Metadata: {
•Date:07/16/2012
•}
```



Aviation Information World - Forecasting the Future



# How it works

```
•UPDATE cars SET owner = 'Ronnie Nash' WHERE
VIN = '123456789'
```

•cars

ID	Manufacturer	Model	Year	VIN	Owner
1	Tesla	Model S	2012	123456789	•Ronnie Nash

•history()

ID	Version	Start	Manufacturer	Model	Year	VIN	Owner
1	1	7/16/2012	Tesla	Model S	2012	123456789	Traci Russell
1	2	8/03/2013	Tesla	Model S	2012	123456789	Ronnie Nash

•journal

•H(T<sub>1</sub>)

```
•INSERT cars
•ID:1
•Manufacturer: Tesla
•Model: Model S
•Year: 2012
•VIN: 123456789
•Owner: Traci Russell
•
Metadata: {
  •Date:07/16/2012
  •}
```

•H(T<sub>2</sub>)

```
•UPDATE cars
•ID:1
•Owner: Ronnie Nash
•
Metadata: {
  •Date:08/03/2013
  •}
```



```
•Date:07/16/2012
•}
```



Aviation Information World - Forecasting the Future



# How it works

•DELETE FROM cars WHERE VIN = '123456789'

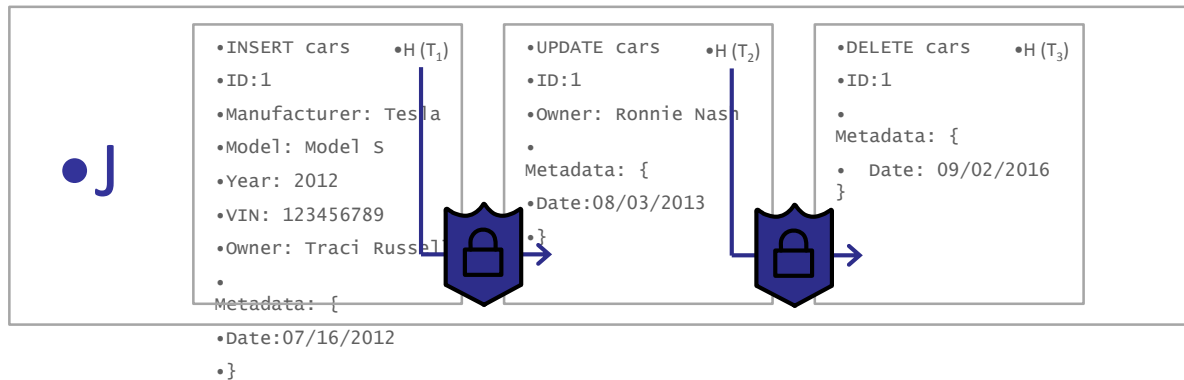
•cars

ID	Manufacturer	Model	Year	VIN	Owner
1	Tesla	Model S	2012	123456789	Ronnie Nash

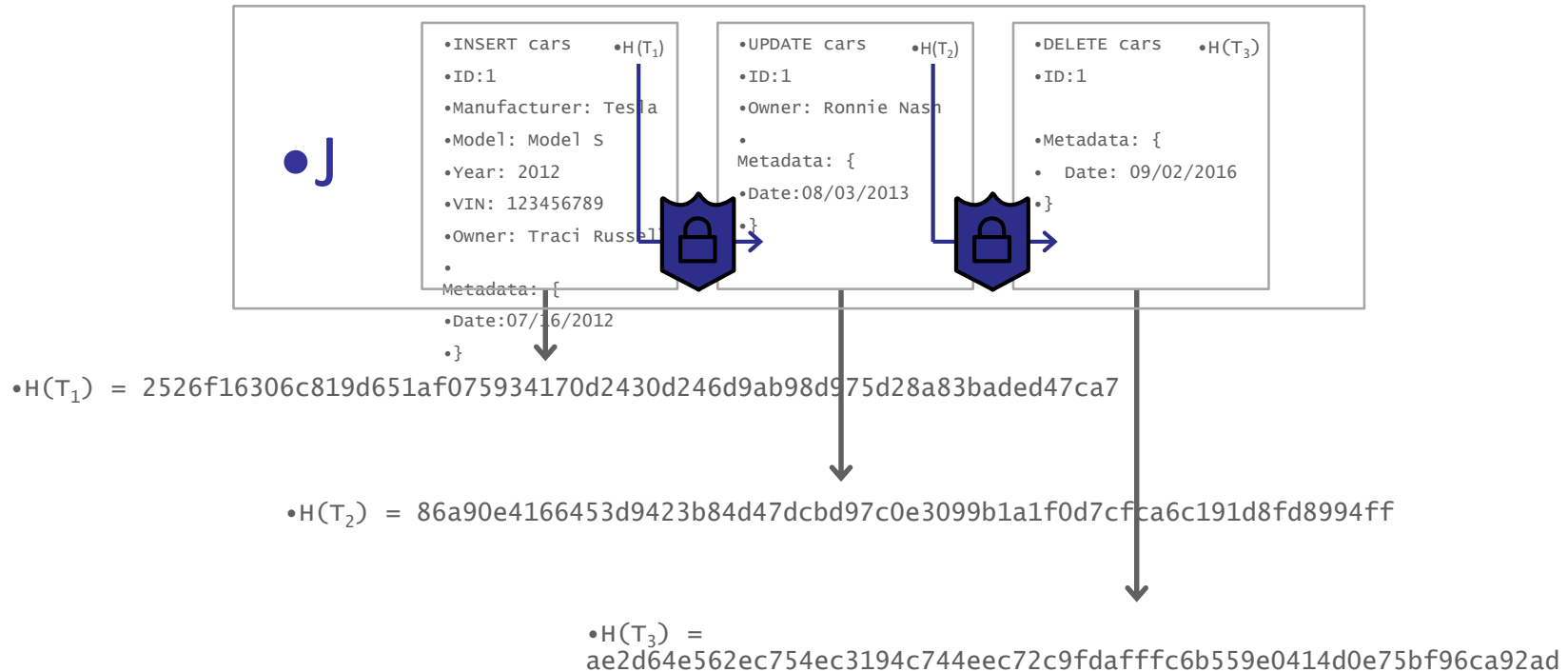
•history()

ID	Version	Start	Manufacturer	Model	Year	VIN	Owner
1	1	7/16/2012	Tesla	Model S	2012	123456789	Traci Russell
1	2	8/03/2013	Tesla	Model S	2012	123456789	Ronnie Nash
1	3	9/02/2016	Deleted				

•journal

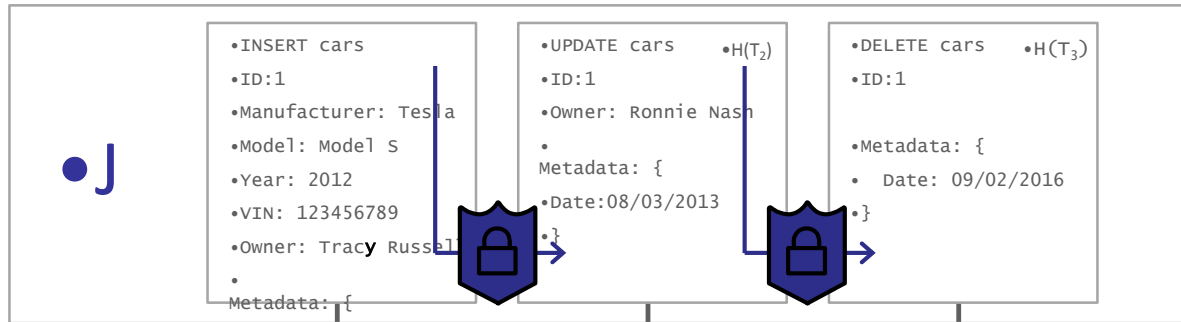


# A digest is a hash value at a point in time





# Changing committed data breaks the chain



$H(T_1) = 2526f16306c819d651af075934170d2430d246d9ab98d575d28a83badcd47ca7$

$H(T_1) = 25d0b44e6e8878151646ffc1fea4eb85c3e4bf4baec212a9fcf67b6d5a81e01a$

$H(T_2) = 86a90e4166453d9423b84d47dcbd97c0e3099b1a1f0d7cfea6c191d8fd8994fff$

$H(T_2) = a90a9898c7e4b1aab19c705b554afd9e0bf6539bb0346df19be362ff63001098$

$H(T_3) = a92d64e562ec754ec3194c744eec72c9fdafffc6b559e0414d0e75bf96ca92adc6268578a24dbe0c7cfba07bd967411a35462b8c875d42f1991faad02c0ac93c$

# Why does immutability and verifiability matter?

- Reduce risk: ensure safeguarding of critical system-of-record applications where data loss could be expensive.
- Improve data tracking: helps you or any parties that have access to the system to quickly and accurately track data's entire lineage, improving efficiency in tracking the source of issues (e.g., manufacturing defects, maintain supply network data hygiene)
- Auditability: helps reduce downtime caused due to audit and compliance issues, saving hundreds of productivity hours for your team
- Reduce implementation effort: building immutability and verifiability in a traditional way is time consuming, complex, and expensive



# Blockchain Success Stories on AWS



ATHEC

Aviation Information World - Forecasting the Future



# What's next ?

To learn more about our services

## Amazon Managed Blockchain

- Amazon Managed Blockchain : <https://aws.amazon.com/managed-blockchain>

## Amazon QLDB

- Amazon QLDB webpage: <https://aws.amazon.com/qldb>

## Amazon Blockchain Partners

- APN Blockchain Partners Spotlight:  
<https://aws.amazon.com/partners/spotlights/blockchain-partner-spotlight/>

