Building System of Record Applications with Amazon Quantum Ledger Database (QLDB)

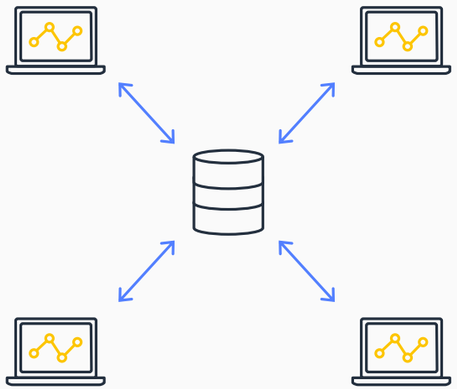
Philip Simko
Product Manager
AWS

Michael Labib
Principal Solutions Architect
AWS

The Customer's Problem – Need for a Ledger

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

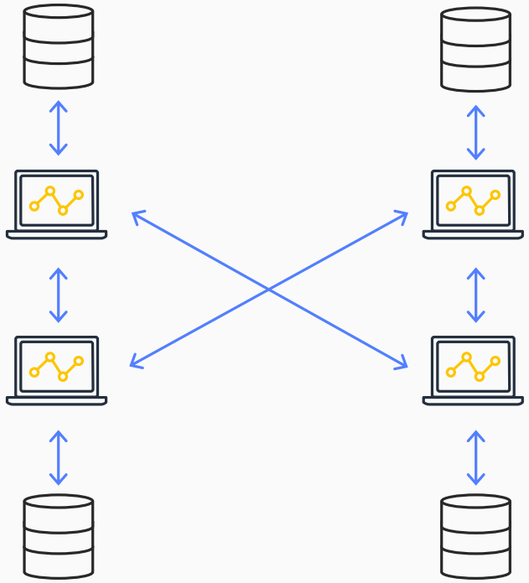
HR & payroll

Track changes to an individual's profile

The Customer's Problem – Need for a Ledger

1 LEDGERS WITH CENTRALIZED TRUST

2 TRANSACTIONS WITH DECENTRALIZED TRUST



Financial institutions

Peer-to-peer payments

Mortgage lenders

Process syndicated loans

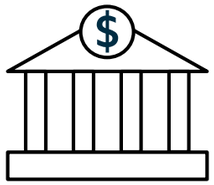
Supply chain

Transact with suppliers and distributers

Retail

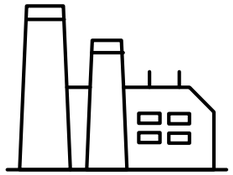
Streamline customer rewards

Ledgers have been around for a while



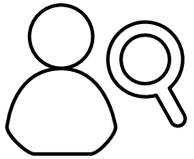
Banking & finance

Keeping track of transactions, trades, and accounts



Manufacturing

Recording components used in manufacturing



Ownership

Maintaining records of asset ownership

Introduction to Amazon QLDB

Fully managed ledger database with a central trusted authority

Purpose-built databases at AWS

Relational

Referential integrity, ACID transactions, schema-on-write

Lift and shift, ERP, CRM, finance

Key-value

High throughput, low-latency reads and writes, endless scale

Real-time bidding, shopping cart, social, product catalog, customer preferences

Document

Store documents and quickly access querying on any attribute

Content management, personalization, mobile

In-memory

Query by key with microsecond latency

Leaderboards, real-time analytics, caching

Graph

Quickly and easily create and navigate relationships between data

Fraud detection, social networking, recommendation engine

Time-series

Collect, store, and process data sequenced by time

IoT applications, event tracking

Ledger

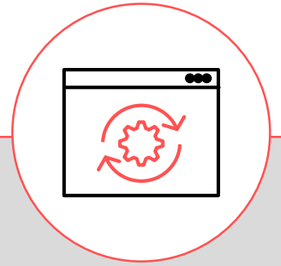
Complete, immutable, and verifiable history of all changes to application data

Systems of record, supply chain, healthcare, registrations, financial

Amazon QLDB Features

Fully managed ledger database
Track and verify history of all changes made to your application's data

Immutable

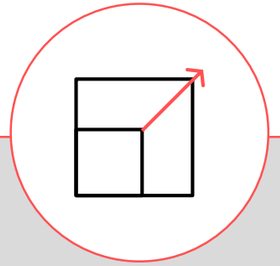


Maintains a sequenced record of all changes to your data, which cannot be deleted or modified; you have the ability to query and analyze the full history

Cryptographically verifiable

Uses cryptography to generate a secure output file of your data's history

Highly scalable



Executes 2–3X as many transactions as ledgers in common blockchain frameworks

Easy to use

Easy to use, letting you use familiar database capabilities like SQL APIs for querying the data

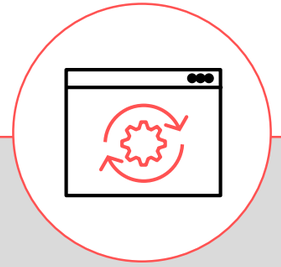
Immutability and Verifiability Matter

Many industries have auditory and compliance requirements that rely on immutability and verifiability

Reduce Risk

Safeguard critical system-of record applications

Improve Tracking

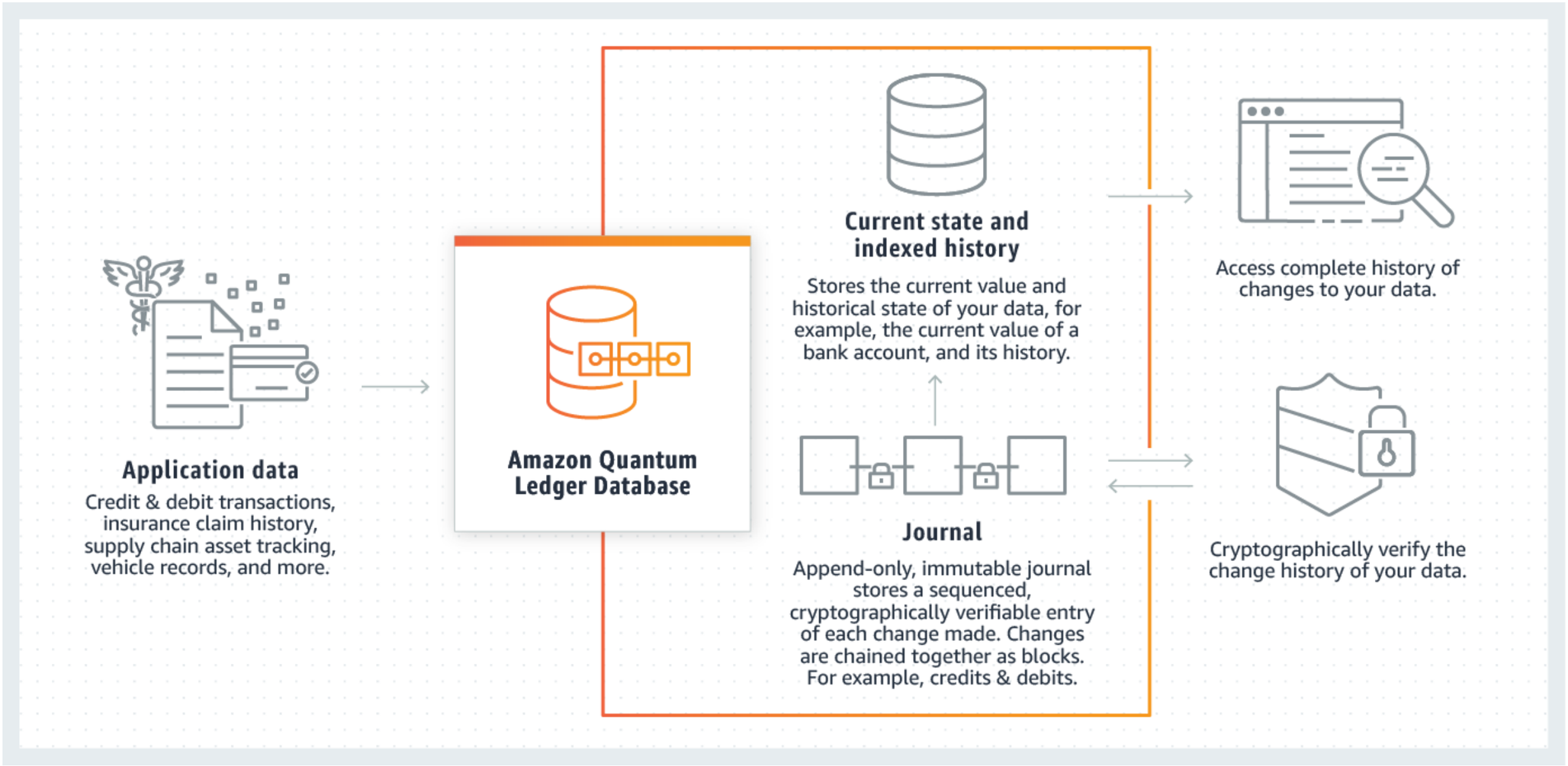


Accurately track data lineage

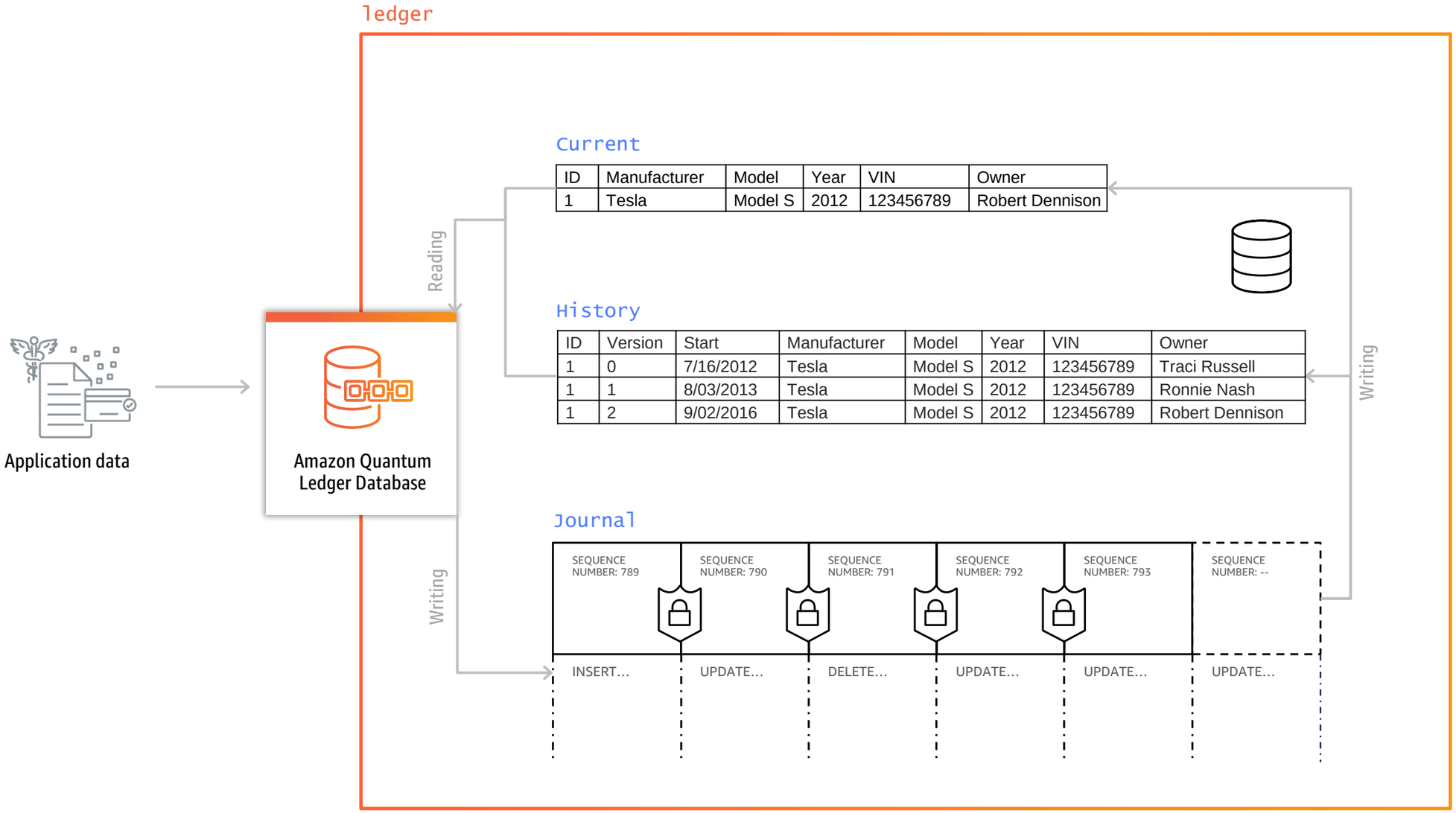
Provide Auditability

Reduce downtime cause by audits

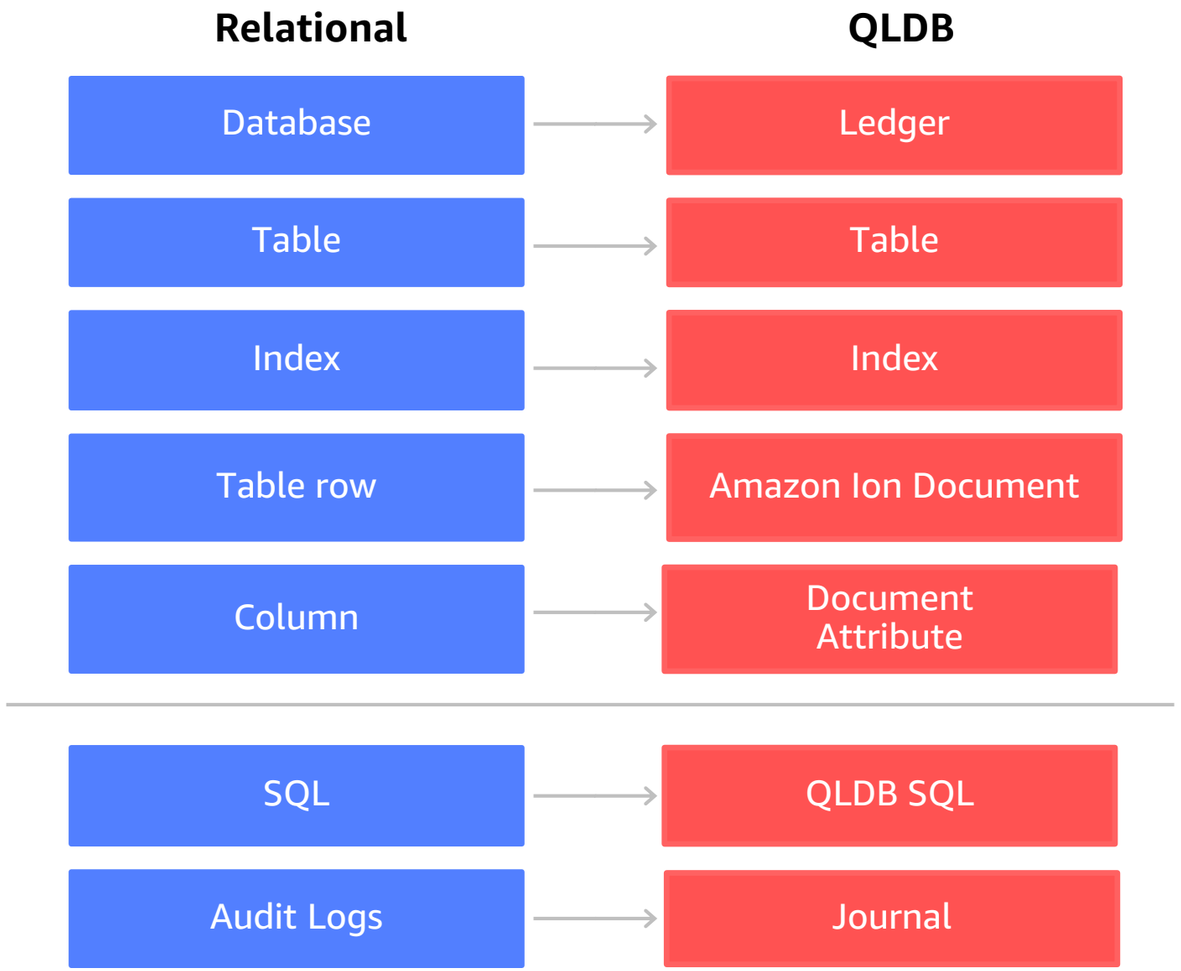
How Amazon QLDB works



Amazon QLDB - Journal First Database



Ease of Use – Mapping to a Relational Database



Immutability

DMV Scenario

1 Tracy buys a car on Aug 3, 2013

Journal

INSERT cars $H(T_1)$
ID:1
Manufacturer: Tesla
Model: Model S
Year: 2012
VIN: 123456789
Owner: Traci Russell

Metadata: {
Date:07/16/2012
}

Current

ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase
1	0	Tesla	Model S	2012	123456789	Traci Russell	8/3/2013

History

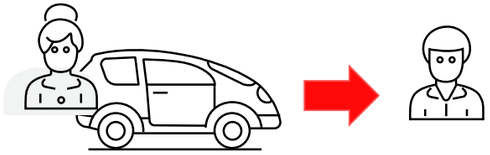
ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase
1	0	Tesla	Model S	2012	123456789	Traci Russell	8/3/2013

Immutability

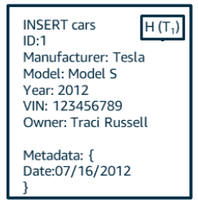
DMV Scenario

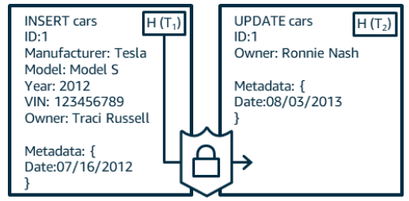
1 Tracy buys a car on Aug 3, 2013

2 Tracy sells car to Ronnie on Sept 10, 2014



Journal





Current

ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase
1	1	Tesla	Model S	2012	123456789	Ronnie Nash	9/10/2014

History

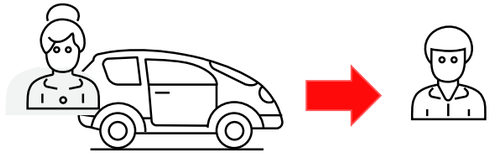
ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase
1	0	Tesla	Model S	2012	123456789	Traci Russell	8/3/2013
1	1	Tesla	Model S	2012	123456789	Ronnie Nash	9/10/2014

Immutability

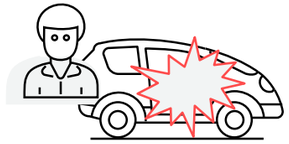
DMV Scenario

1. Tracy buys a car on Aug 3, 2013

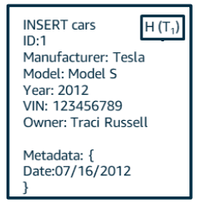
2. Tracy sells car to Ronnie on Sept 10, 2014

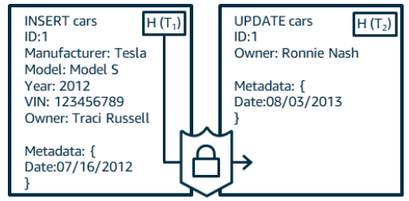


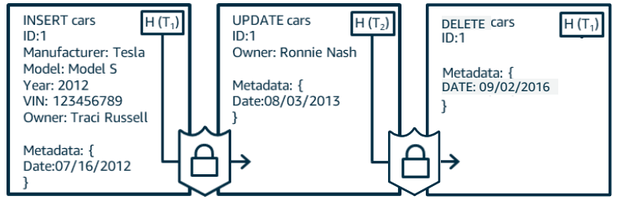
3. Ronnie's car gets in an accident and gets totaled



Journal







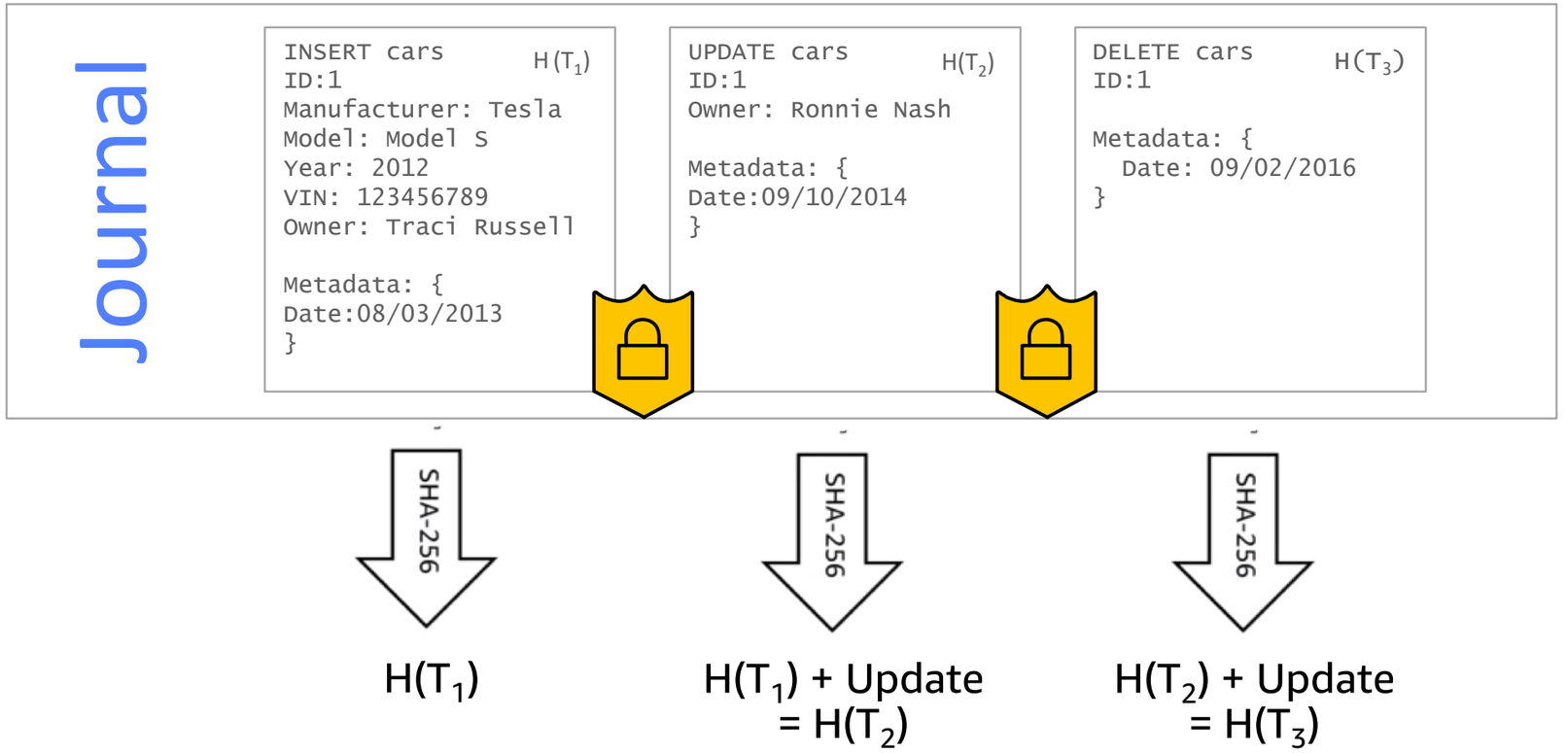
Current

ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase

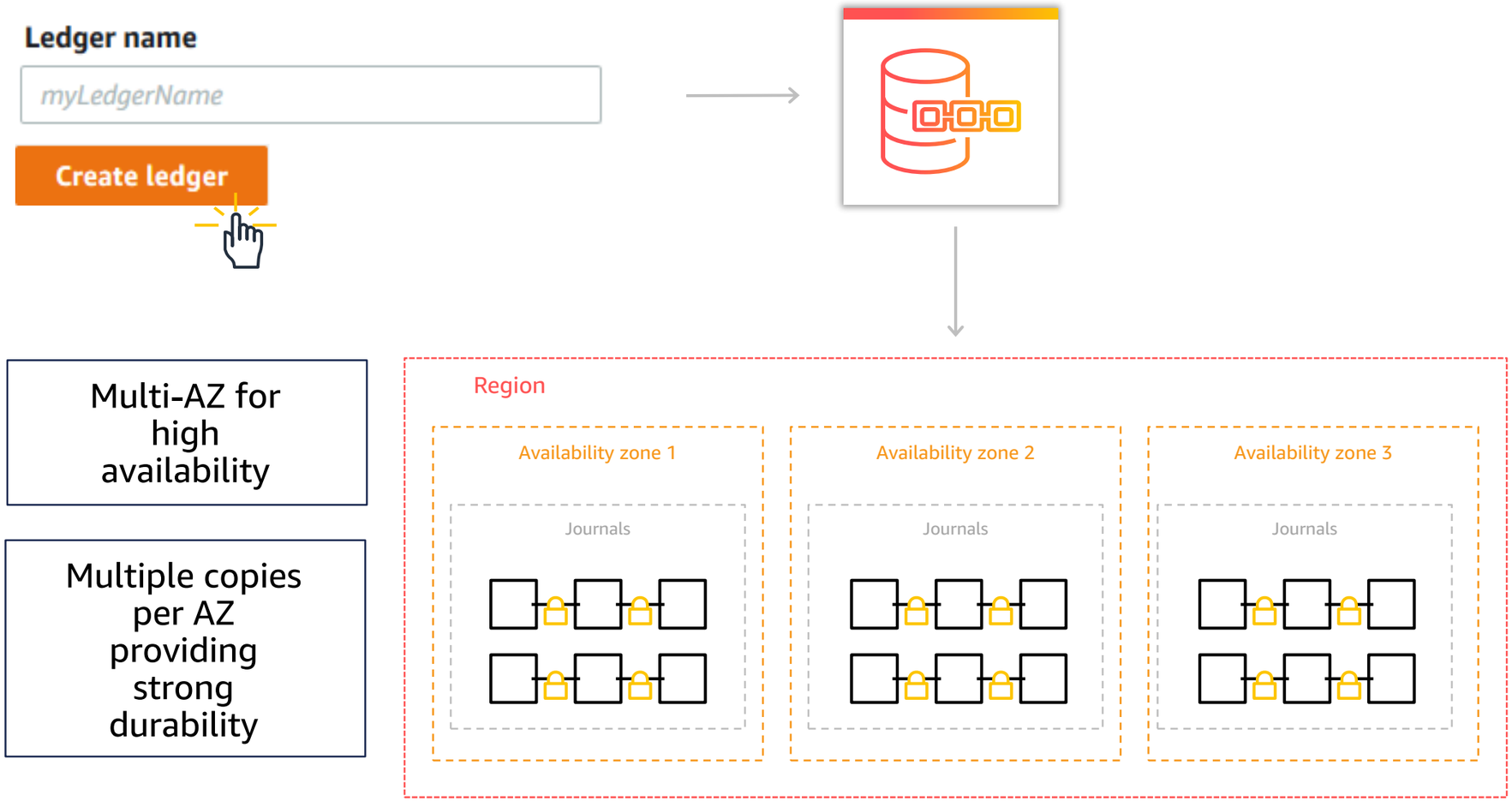
History

ID	Version	Manufacturer	Model	Year	VIN	Owner	Date of Purchase
1	0	Tesla	Model S	2012	123456789	Traci Russell	8/3/2013
1	1	Tesla	Model S	2012	123456789	Ronnie Nash	9/10/2014
1	2	Deleted					

Cryptographic Verifiability



Highly Scalable - Serverless



Multi-AZ for high availability

Multiple copies per AZ providing strong durability

Easy to use - Amazon Ion & SQL-like APIs

Amazon Ion

```
/* 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
                }
}
```

QLDB 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
```

ACID Transactions

HIGHEST TO LOWEST

Isolation Level	Potential Issues
Serializable	-
Snapshot Isolation	Potential write skew
Repeatable read	Phantom reads
Read committed	Phantom reads/non-repeatable reads
Read uncommitted	Phantom reads/non-repeatable reads/dirty reads

Amazon QLDB Features

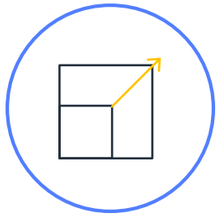
Immutable

Append-only

Cryptographically verifiable

Hash-chaining for data integrity

Highly scalable



Serverless

Easy to use

Flexible document model & familiar SQL operators

ACID Transactions

Fully serializable isolation

Journal-first

The journal is the database

Challenges with Current Ledger Approaches

Traditional Database

Resource intensive

Difficult to manage and scale

Error prone and incomplete

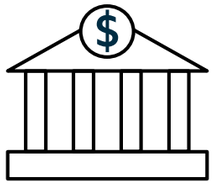
Impossible to verify

Blockchain

Designed for a different purpose

Adds unnecessary complexity

Common customer use cases

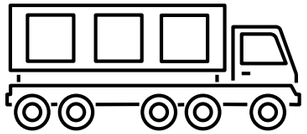


Banking & finance

Keeping track of transactions, trades and accounts

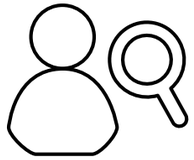
E-Commerce

Where's my stuff?



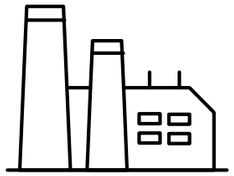
Transport & logistics

Tracking transportation of goods



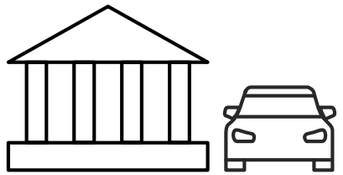
HR & payroll

Tracking changes to an individual's profile



Manufacturing

Recording components used in manufacturing



Government

Tracking vehicle title history

Amazon QLDB Demo

Mike Labib
Principal Solutions Architect – Amazon QLDB

Thank you!

Phil Simko
Product Manager – Amazon QLDB

Mike Labib
Principal Solutions Architect – Amazon QLDB

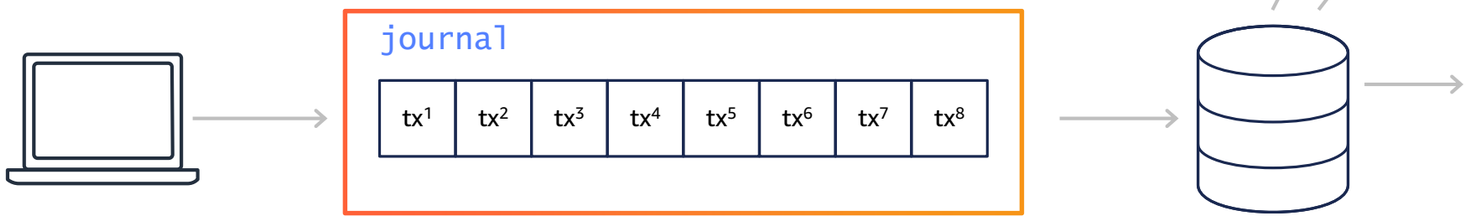
Deep Dive into Amazon QLDB

Traditional Database Architecture - The Log

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

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



User #standard user data, the default

ID	Manufacturer	Model	Year	VIN	Owner
1	Tesla	Model S	2012	123456789	Robert Dennison

Committed #includes metadata

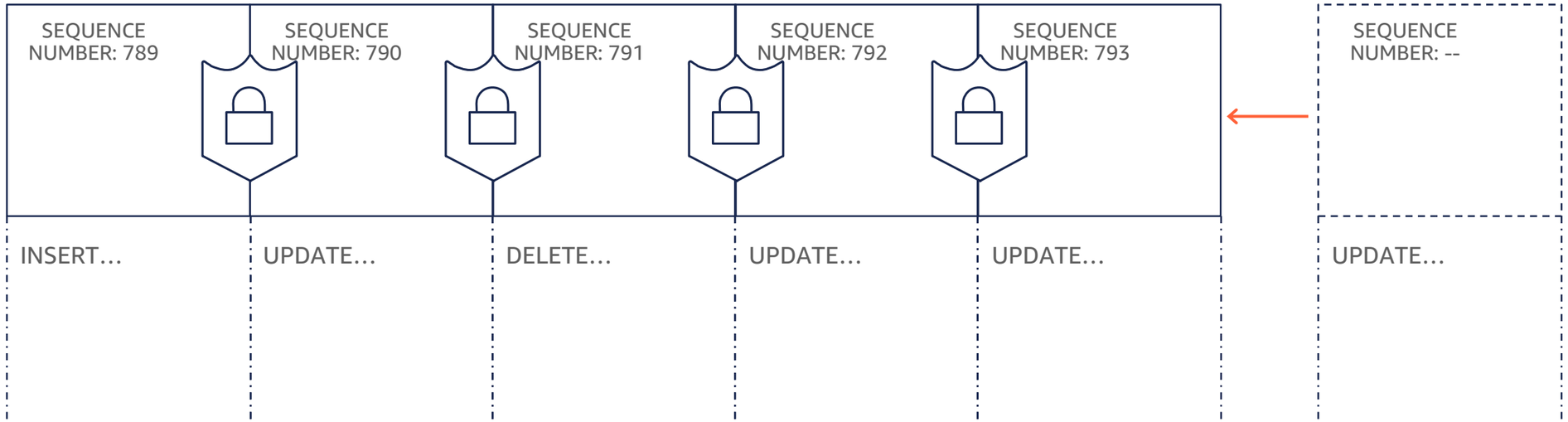
blockAddress	hash	data	metadata
{strandId:"JpbmngzFZV7FHjEuuER1Ol",sequenceNo:78}	{{XKlKYIzWEyBPRgup1Xfa/Qp4JE2PEbA8nc0KxlVGm8c=}}	{FirstName:“Traci",LastName:“Russell",DOB:1963-08-19T00:00:00.000Z,GovId:"LEIS26LL",GovIdType:"Driver License"}	{id:"5PLf8cOOFPoIf7w1NJzUXL",version:0,txTime:2019-06-28,txId:"3mDCDwAbtYi6vGdPfUlDGf"}

history() #function to query document history

blockAdreess	hash	data	metadata
{strandId:"JpbmngzFZV7FHjEuuER1Ol",sequenceNo:78}	{{XKlKYIzWEyBPRgup1Xfa/Qp4JE2PEbA8nc0KxlVGm8c=}}	{Manufacturer:"Tesla",Model:"Model S",Year:"2012",VIN:"123456789",Owner:"Traci Russell"}	{id:"5PLf8cOOFPoIf7w1NJzUXL",**version:0**,txTime:2019-06-28,txId:"3mDCDwAbtYi6vGdPfUlDGf"}
{strandId:"60bpn7xLtB48311uwkihe8",sequenceNo:11}	{{ii2h58whRCHk/1zRp4RLglG9D2SlNDa32rUWZtcS11E=}}	{Manufacturer:"Tesla",Model:"Model S",Year:"2012",VIN:"123456789",Owner:"Traci Russell",owner:"Ronnie Nash"}	{id:"Kwo6aQwJ4Dz4D1oyVqRgxY",**version:1**,txTime:2019-07-04T20:21:22.071Z,txId:"6BFspx97Mtq4sEid33YkMd"}
{strandId:"60bpn7xLtB48311uwkihe8",sequenceNo:13}	{{UdPrq7OTHfiikK9rS8YRBpjGI0c5Pfl3DreSmQaGrfc=}}	{Manufacturer:"Tesla",Model:"Model S",Year:"2012",VIN:"123456789",Owner:"Traci Russell",owner:“Robert Dennison"}	{id:"Kwo6aQwJ4Dz4D1oyVqRgxY",**version:2**,txTime:2019-07-04T20:24:45.768Z,txId:"23khn4h3uvH6i8dwKefLjS"}

The Journal and Immutability

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

QLDB's Data Model - Amazon Ion

JSON Document

```
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
            }
}
```

Ion 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
            }
}
```

https://github.com/amzn/ion-java

QLDB's Data Model – Query

QLDB 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
```

SQLStatements

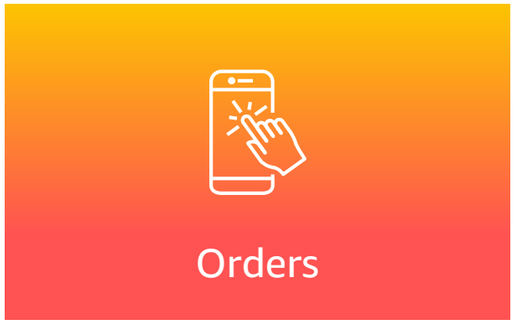
CREATE INDEX - CREATE TABLE - DELETE
DROP TABLE - FROM - INSERT - SELECT
UPDATE

SQL Functions

CAST - CHAR_LENGTH - CHARACTER_LENGTH
COALESCE - DATE_ADD - DATE_DIFF
EXISTS – EXTRACT – LOWER – SIZE - NULLIF
SUBSTRING - TO_STRING - TO_TIMESTAMP
TRIM – UPPER - UTCNOW

Data Model Example - Retail

Assume 3 tables



CREATE TABLE Orders

CREATE TABLE Customers

CREATE TABLE Products

Data Model Example - Retail

How best to model this ?

- Flexible document schema leveraging Amazon ION


```
INSERT INTO customers
{
    'customer-id': 1000,
    'first-name': ‘John',
    'last-name': ‘Doe',
    'membership': true,
    'address': ‘123 First Street’
    'city': ‘Seattle',
    'state': ‘WA'
}
```


```
INSERT INTO products
{
   'product-id': 346211,
   'product-description': 'socks',
   'product-color': 'blue',
   'price': 5.00,
   'active': true,
   'external-sku': 'Ak3234211'
}
```

Data Model Example - Retail

Nested document structure enables optimal queries and data access


```
INSERT INTO orders
{
    'order-id' : 100056,

    'customer' :   {
                'customer-id': 1000,
                'first-name' : ‘John',
                'last-name'  : ‘Doe',
                'address'    : ‘123 First Street',
                'city'       : ‘Seattle',
                'state'      : ‘WA'
                      },

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

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

                      },
    'total' : 55.00
}
```

Data Model Example - Retail

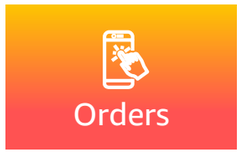
query

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

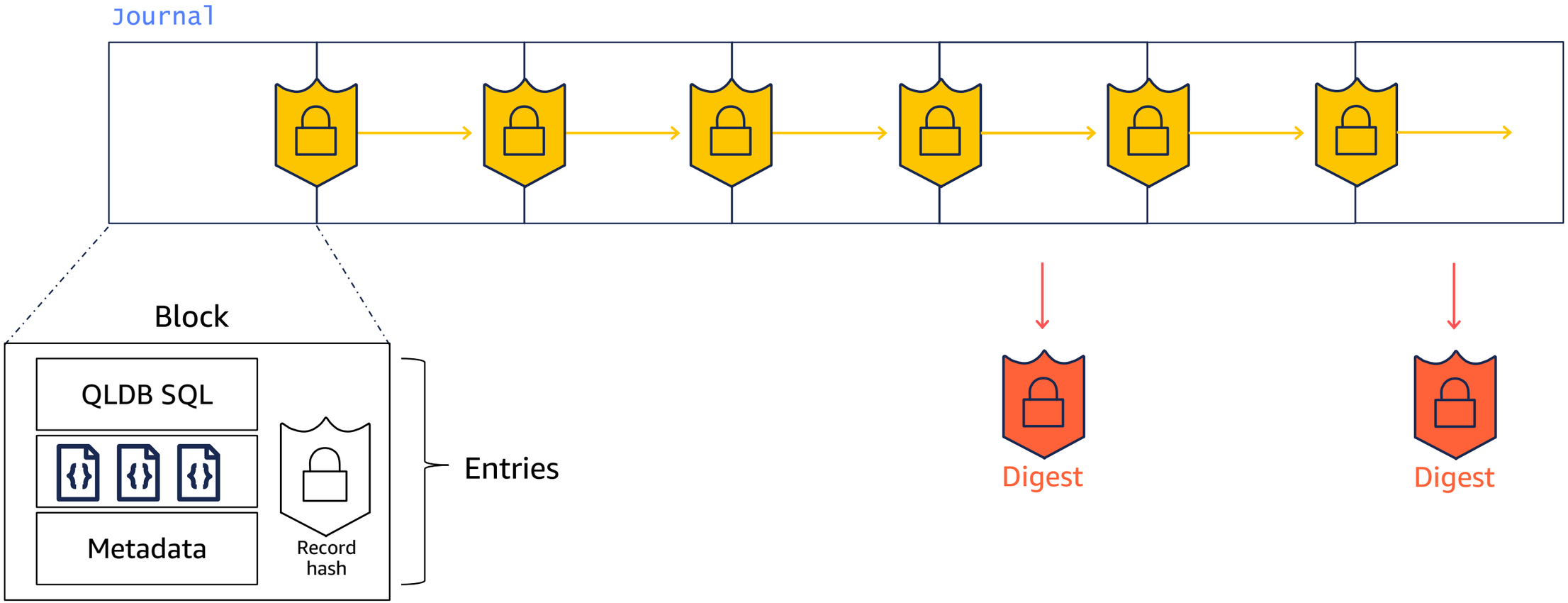
result

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

Nested document query
(**customer within orders**)



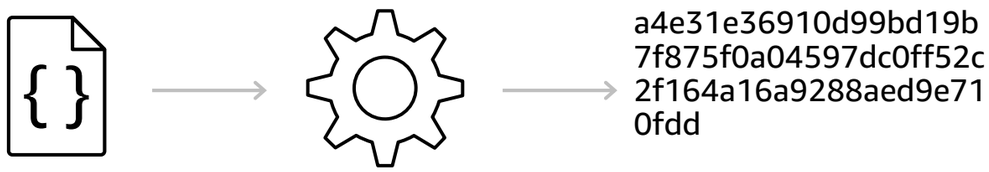
Cryptographic Verification



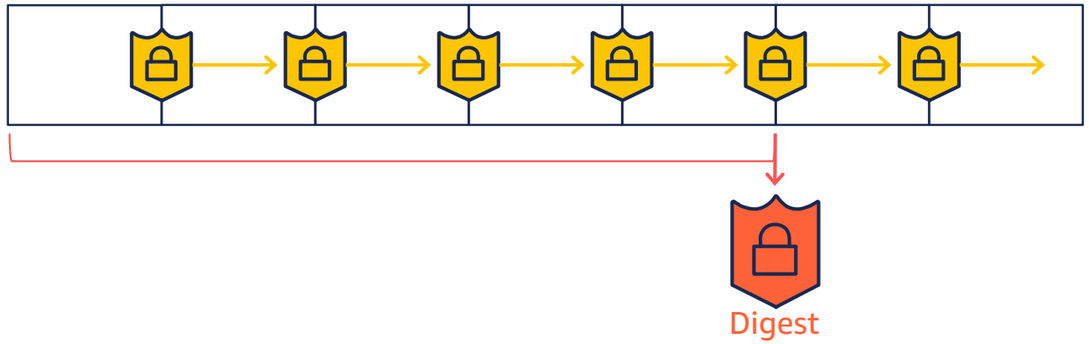
Cryptographic Verifiability

Four basic steps to seeing how QLDB's verifiability works

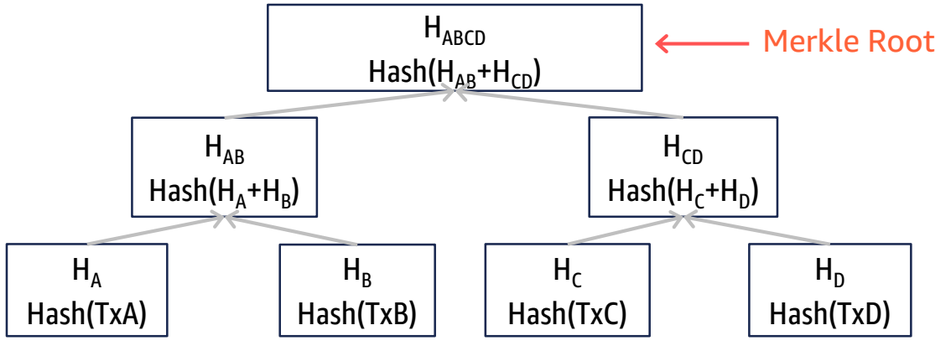
SHA256: Unique Signature of a document



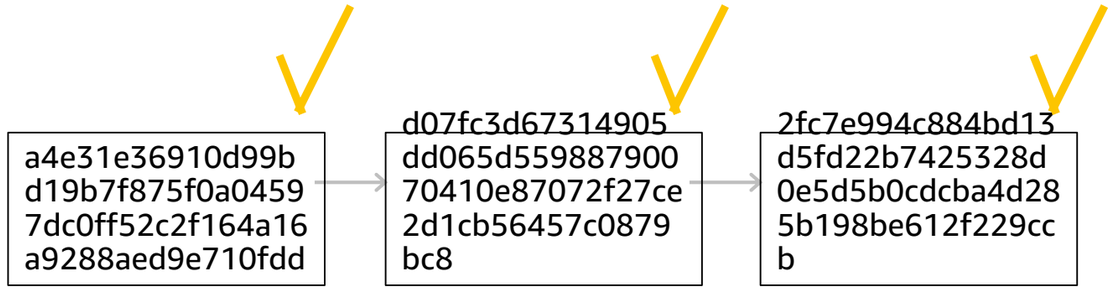
Digest: Periodic hash covering all history



Merkle Trees: Chaining past hashes together



Proof: A chain of hashes linking a document to its digest



Cryptographic Verifiability – SHA-256

```
vehicle = {
            'VIN' :       "KM8SRDHF6EU074761",
            'Type':       "Truck"
            'Model':      "F150"
            'Specs': {
                  'EngSize' : 3.3
                  'CurbWeight': 4,878
                  'HP': 327
            }
}
```

SHA-256

a4e31e36910d99bd19b7f
875f0a04597dc0ff52c2f16
4a16a9288aed9e710fdd

```
vehicle = {
            'VIN' :       "KM8SRDHF6EU074761",
            'Type':       "Truck"
            'Model':      "F150"
            'Specs': {
                  'EngSize' : 3.3
                  'CurbWeight': 4,879
                  'HP': 327
            }
}
```

SHA-256

19318457408920af2d2cb
eacd90c7afe0fbd7f6ff316
972c8f656c8bbc402dd1

Cryptographic Verifiability – SHA-256

```
vehicle = {
            ‘VIN’ :       “KM8SRDHF6EU074761”,
            ‘Type’:       “Truck”
            ‘Model’:      “F150”
            ‘Specs’: {
                  ‘EngSize’ : 3.3
                  ‘CurbWeight’: 4,878
                  ‘HP’: 327
            }
}
```

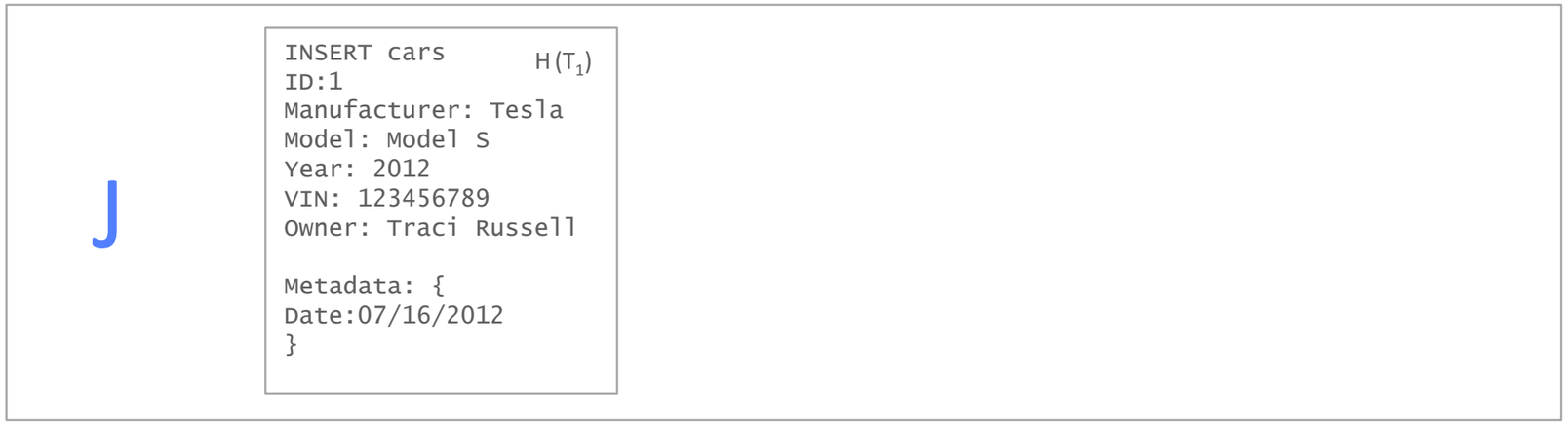
SHA-256

a4e31e36910d99bd19b7f
875f0a04597dc0ff52c2f16
4a16a9288aed9e710fdd

SHA-256

19318457408920af2d2cb
eacd90c7afe0fbd7f6ff316
972c8f656c8bbc402dd1

Hashing and Chaining Transactions



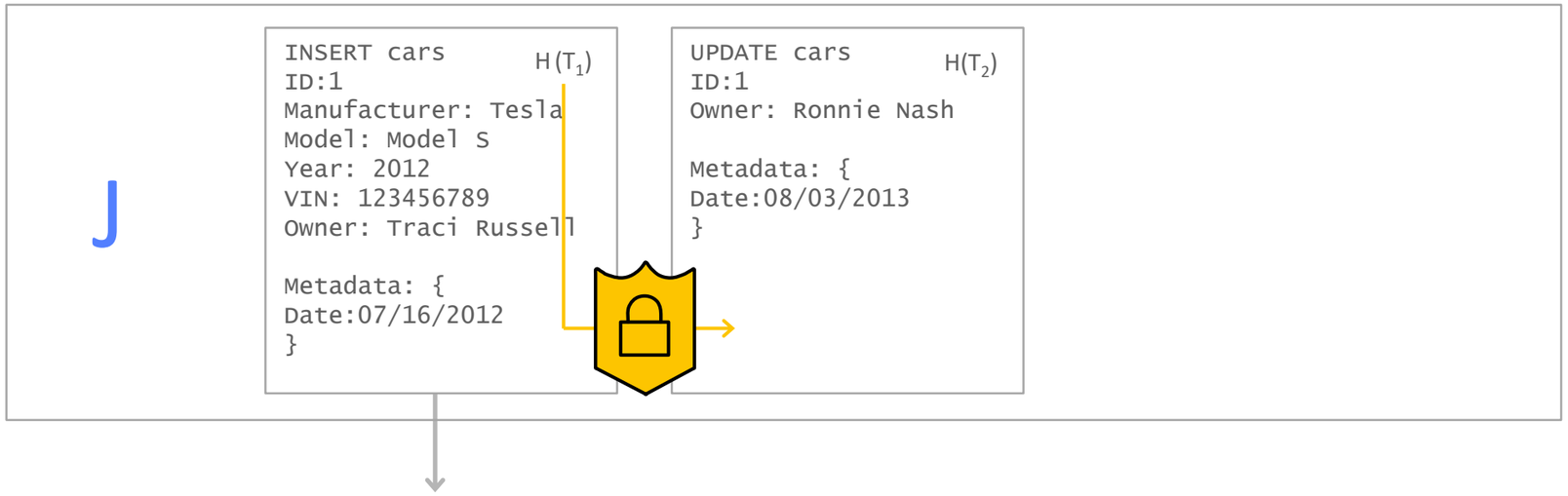

```
INSERT cars
ID:1
Manufacturer: Tesla
Model: Model S
Year: 2012
VIN: 123456789
Owner: Traci Russell

Metadata: {
Date:07/16/2012
}
```

SHA-256

$H(T_1)$ = 2526f16306c819d651af075934170d2430d246d9ab98d975d28a83baded47ca7

Hashing and Chaining Transactions



$H(T_1)$ = 2526f16306c819d651af075934170d2430d246d9ab98d975d28a83baded47ca7

$H(T_1)$ +

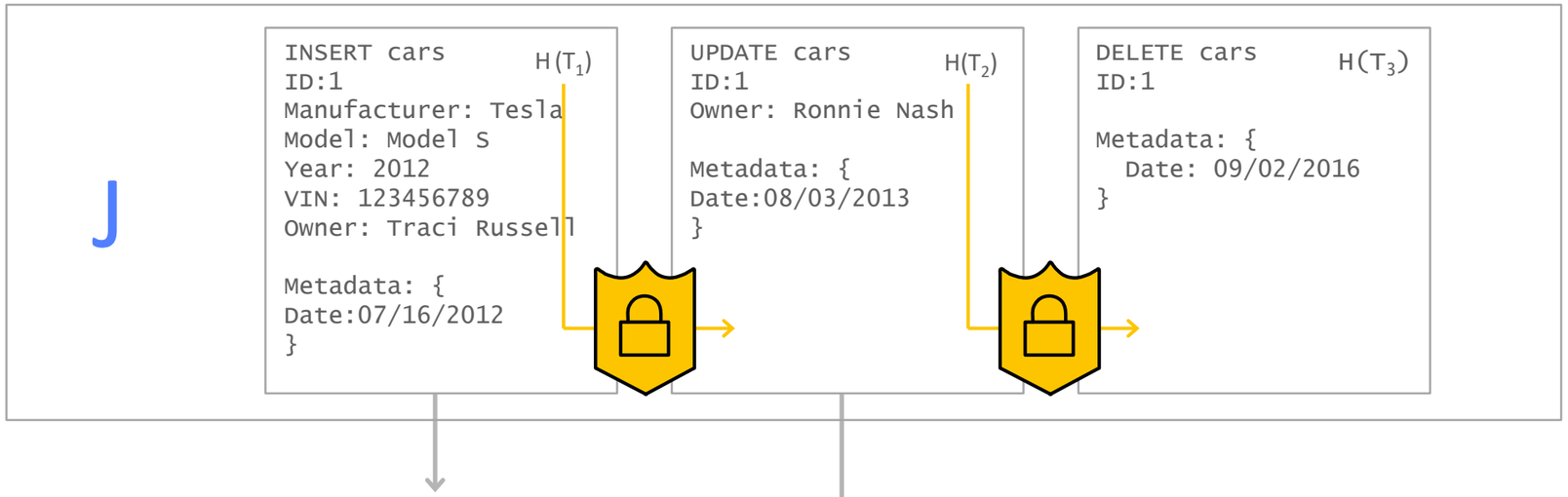
```
UPDATE cars
ID:1
Owner: Ronnie Nash

Metadata: {
Date:08/03/2013
}
```

SHA-256

$H(T_2)$ = 86a90e4166453d9423b84d47dcbd97c0e3099b1a1f0d7cfca6c191d8fd8994ff

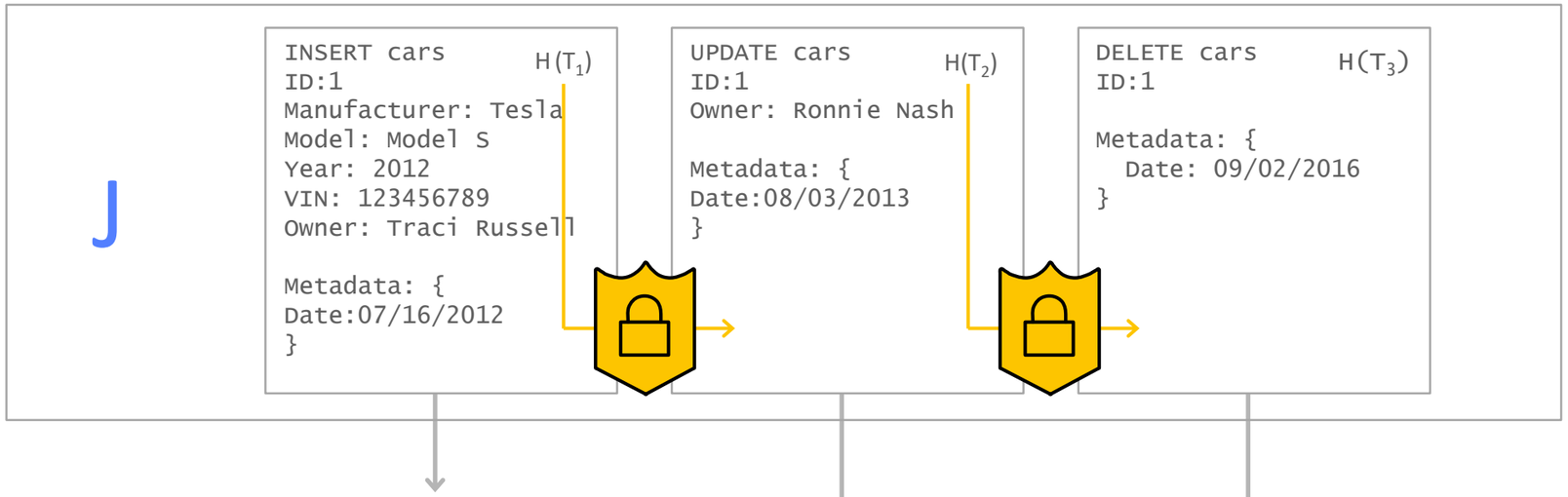
Hashing and Chaining Transactions



$H(T_1)$ = 2526f16306c819d651af075934170d2430d246d9ab98d975d28a83baded47ca7

$H(T_2)$ = 86a90e4166453d9423b84d47dcbd97c0e3099b1a1f0d7cfca6c191d8fd8994ff

Hashing and Chaining Transactions

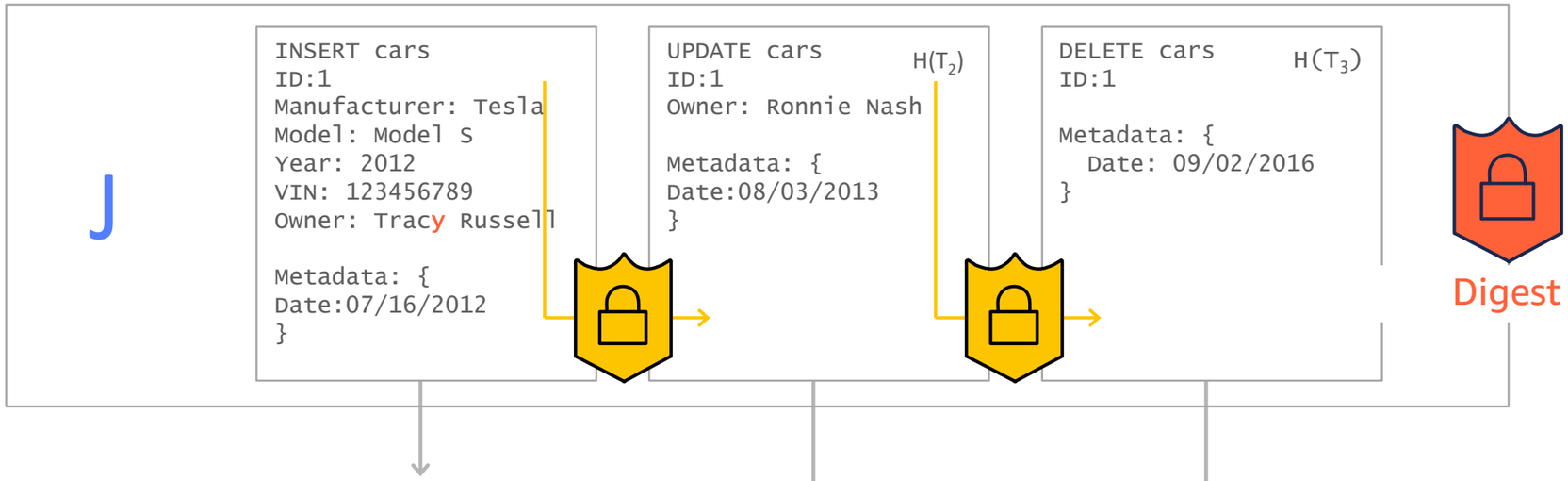


$H(T_1)$ = 2526f16306c819d651af075934170d2430d246d9ab98d975d28a83baded47ca7

$H(T_2)$ = 86a90e4166453d9423b84d47dcbd97c0e3099b1a1f0d7cfca6c191d8fd8994ff

$H(T_3)$ = ae2d64e562ec754ec3194c744eec72c9fdafffc6b559e0414d0e75bf96ca92ad

Digests & Data Changes



$H(T_1)$ = ~~2526f16306c819d651af075934170d2430d246d9ab98d975d28a83baded47ca7~~

$H(T_1)$ = 25d0b44e6e8878151646ffc1fea4eb85c3e4bf4baec212a9fcf67b6d5a81e01a

$H(T_2)$ = ~~86a90e4166453d9423b84d47dcbd97c0e3099b1a1f0d7cfca6c191d8fd8994ff~~

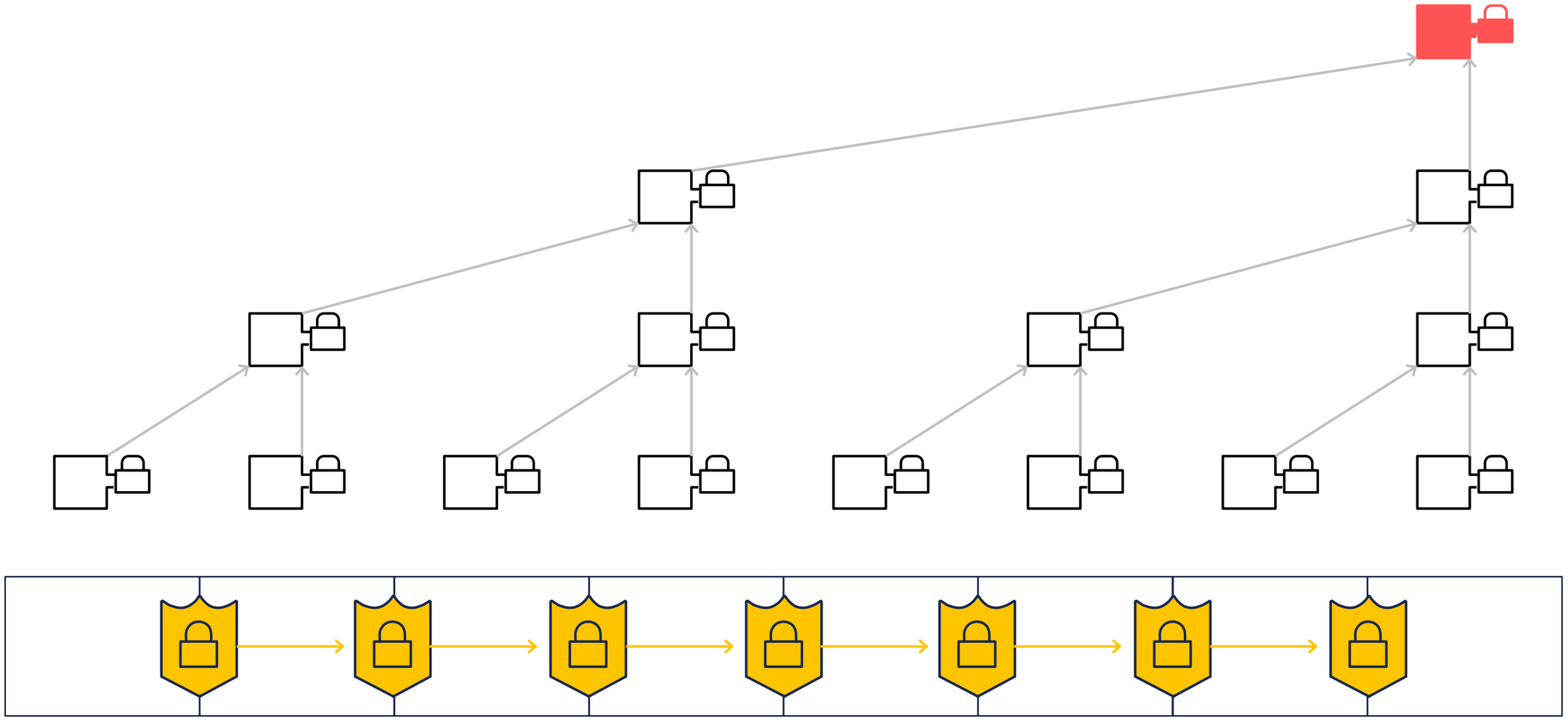
$H(T_2)$ = a90a9898c7e4b1aab19c705b554afd9e0bf6539bb0346df19be362ff63001098

$H(T_3)$ = ~~ae2d64e562ec754ec3194c744eec72c9fdafffc6b559e0414d0e75bf96ca92ad~~

$H(T_3)$ = c6268578a24dbe0c7cfba07bd967411a35462b8c875d42f1991faad02c0ac93c

Cryptographic Verifiability – The Digest

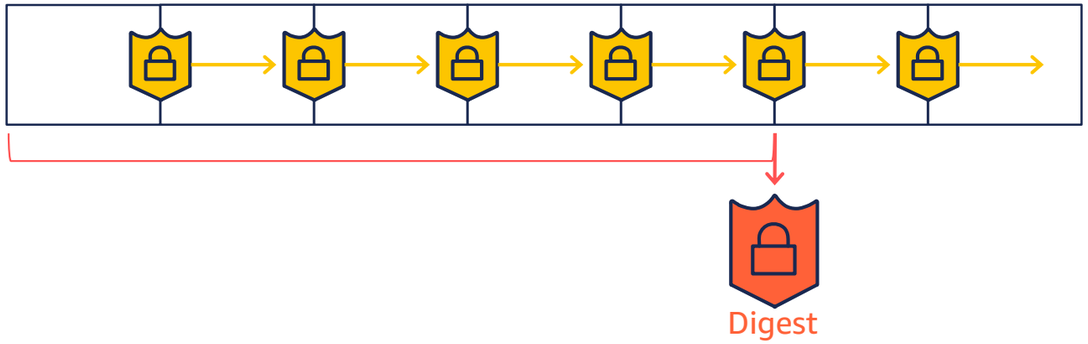
Your ledger's Merkle tree root at a point in time



Cryptographic Verifiability - Recalculations

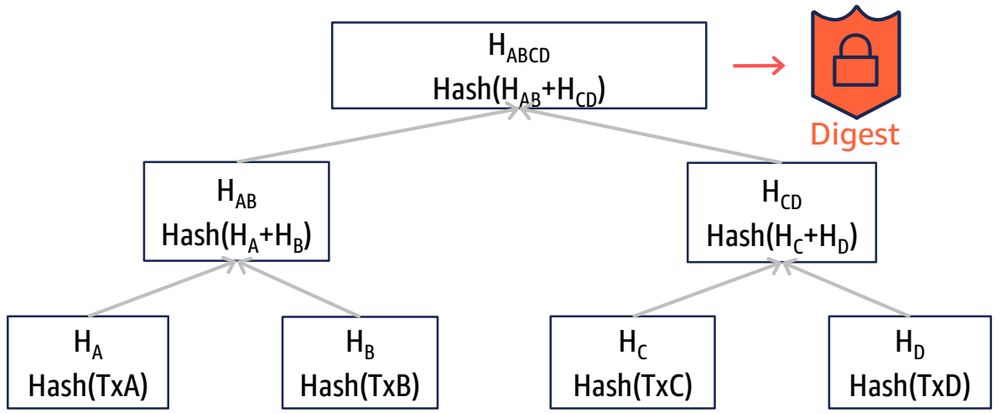
Linear

10,000,000 Transactions = 10,000,000 hashes

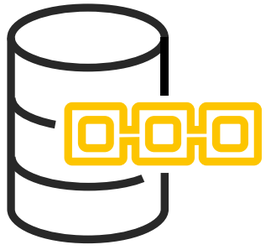


Merkle

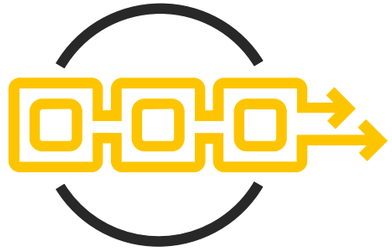
10,000,000 Transactions = 24 hashes



Blockchain at AWS



Amazon Quantum Ledger Database



Amazon Managed Blockchain

Thank you!

Nate Welshons
Global BDM - QLDB