Building Real-time OTA Updates with an Object Oriented DB (Realm/Device SDK)

Arnaldo Vera, Industry Solutions MongoDB

11 October 2023 | COVESA Fall AMM 2023Troy, MI

Agenda

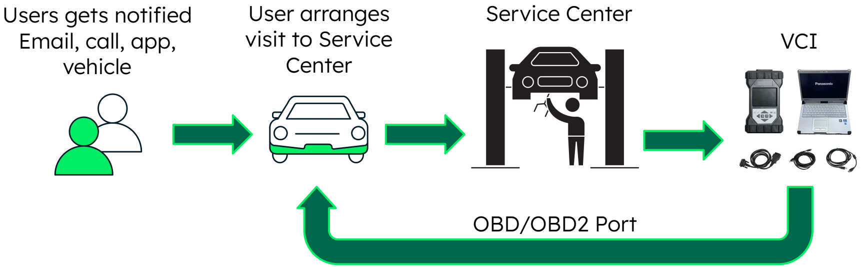
- Traditional updates vs Over-the-Air (OTA)
- Challenges of OTA updates
- Robust SOTA system
- Logical Concepts for robust SOTA system
- Our proposed solution
- Other Possibilities besides MongoDB
- Realm in AUTOSAR

Software updates

Typical Design and Challenges

Software Update Methods

Traditional



Users gets notified Email, call, app, vehicle

User arranges visit to Service Center

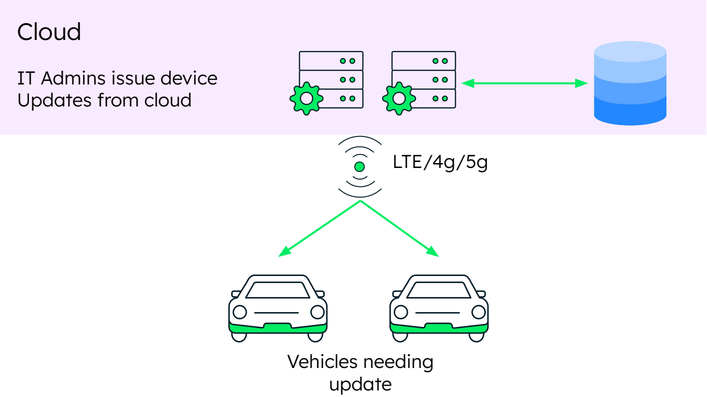
Service Center

VCI

OBD/OBD2 Port

- Distribution takes A LOT of time/resources:
- No Guarantees for Security Patches
- Not Scalable
- In some cases, tracking/tracing is lost.
- Highly inconvenient for the user.
- Unofficial Service Centers might not be able to update.
- Human error involved

Over-the-Air (OTA)



Cloud

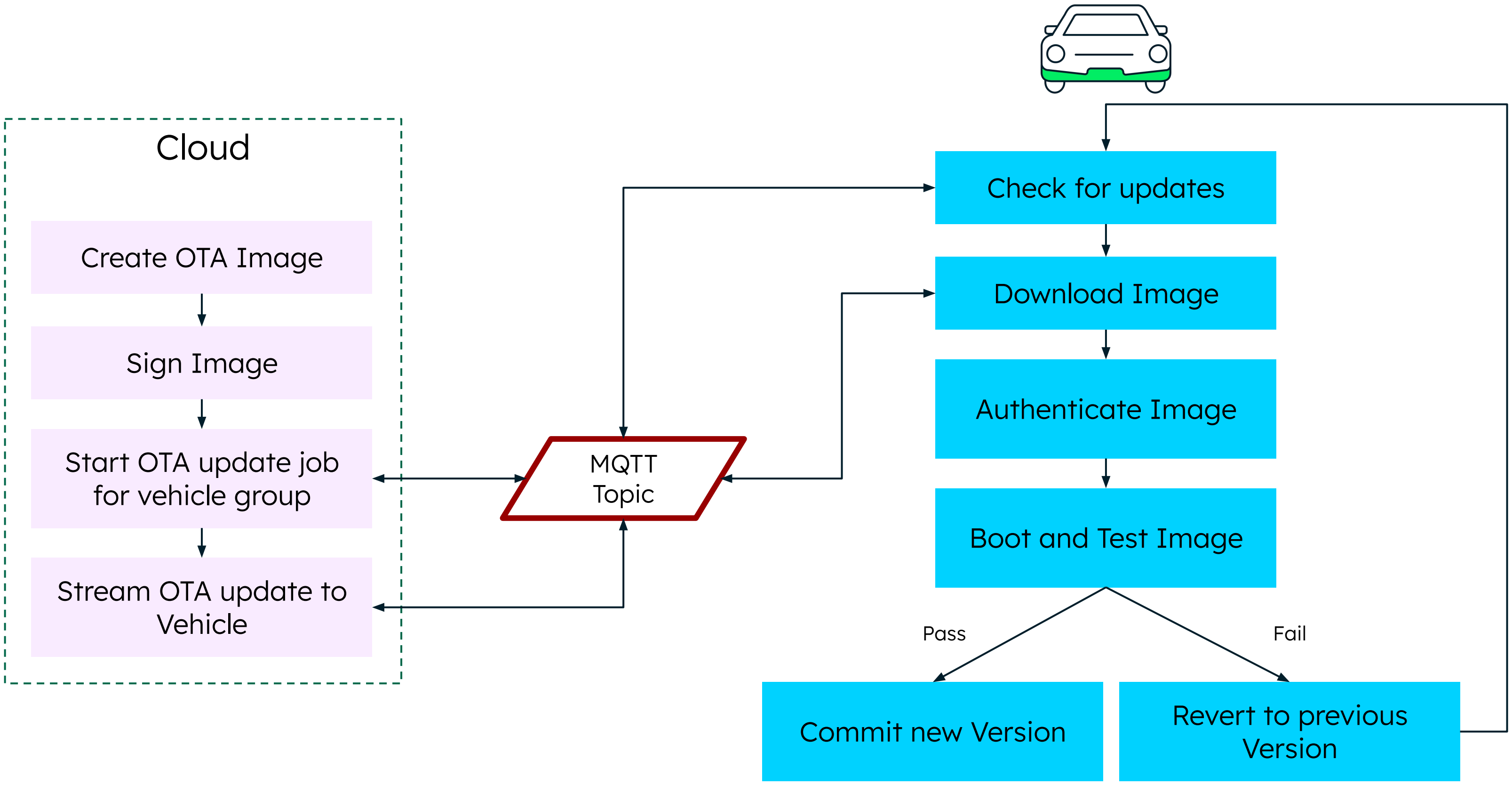
IT Admins issue device Updates from cloud

LTE/4g/5g

Vehicles needing update

- Better Control over updates on a fleet.
- Much better user experience.
- Much less human error.
- Distribution is done digitally.
- Silent or user-approved updates
- Instant update for security patches
- Digitalization of Firmware/Software versioning/history

Typical SOTA updates flow



Cloud

Create OTA Image

Sign Image

Start OTA update job for vehicle group

Stream OTA update to Vehicle

MQTT Topic

Check for updates

Download Image

Authenticate Image

Boot and Test Image

Pass

Fail

Commit new Version

Revert to previous Version

Source: https://www.ti.com/lit/wp/sway021/sway021.pdf?ts=1696937685448

Challenges

Very High Network Costs

$/GB data transfer over the Telco Network can become extremely expensive at Scale

Bandwidth limitations affect updates

Some updates require a few GBs, with slow network, vehicles can stay blocked for many hours

Data Transmission Reliability

In case of connection lost, retry mechanisms are needed. Redownload data packages

Heterogeneous Vehicle systems

Different configurations and update plans→ updates are not just 1 to 1, sometimes one ECU has to update, then that triggers another update in another ECU.. etc.

Costly infrastructure to develop and maintain

Managing retries, connectivity, authentication, filtering, scalability, and high availability.

Heavy investment in Security

Field Level encryption, at rest and in flight. Prevent man-in-the-middle attacks.

The Result of these Challenges

OEMs are reluctant to deploy updates

$/GB data transfer over the Telco Network can become extremely expensive at Scale

Deployed updates stale and affect users

Some updates require vehicles to stay blocked for many hours.
User satisfaction decreases

Costs are increased instead of decreased due to R&D

With all the infrastructure cost, internal Engineering hours for building and maintaining the system. TCO end up increasing.

Requirements for robust SOTA system

Compressed and Efficient Data Transfer

Lower $/GB data transfer over the Telco Network as much as possible.

Offline-first Paradigm and User Notifications

Notify users reliably and decrease retries as much as possible.

Reliable/Robust Communication Protocol

Conflict resolution, retry mechanisms., reliable.

Flexible Data Model

Keep a library of current updates for all models, years, packages and configurations.
Keep track of historical updates.

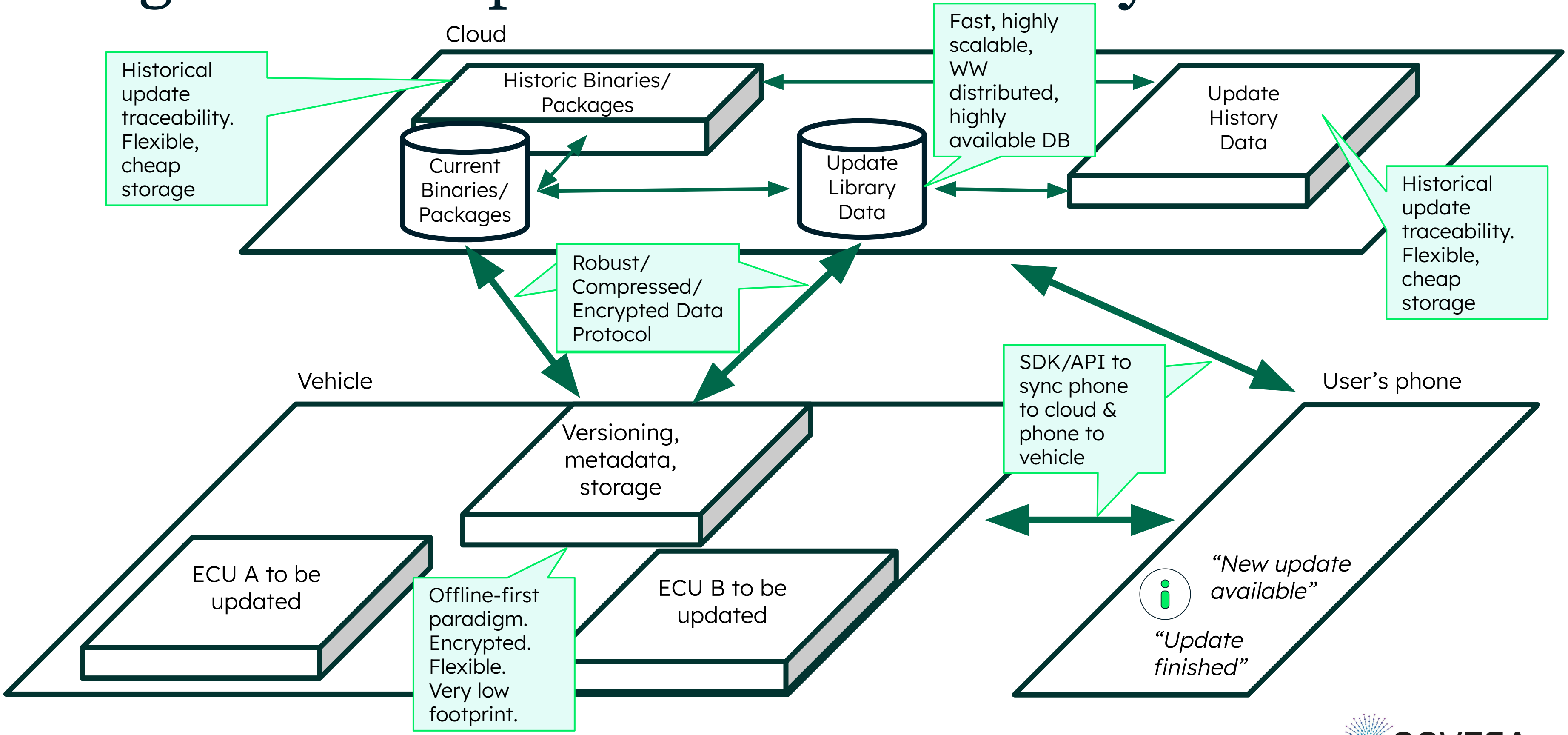
Flexible System, easy to develop/maintain.

Reduce as much as possible data piping work for engineers. Otherwise, fast, error-reduced deployments won’t be possible

Encrypted and Secured system

Authentication, authorization, encryption at rest, and in flight.

Logical Concepts for robust SOTA system



Historical update traceability. Flexible, cheap storage

Fast, highly scalable, WW distributed, highly available DB

Historical update traceability. Flexible, cheap storage

Robust/ Compressed/ Encrypted Data Protocol

SDK/API to sync phone to cloud & phone to vehicle

Offline-first paradigm. Encrypted. Flexible. Very low footprint.

Our proposed solution

Device SDK, Device Sync and the Document Model

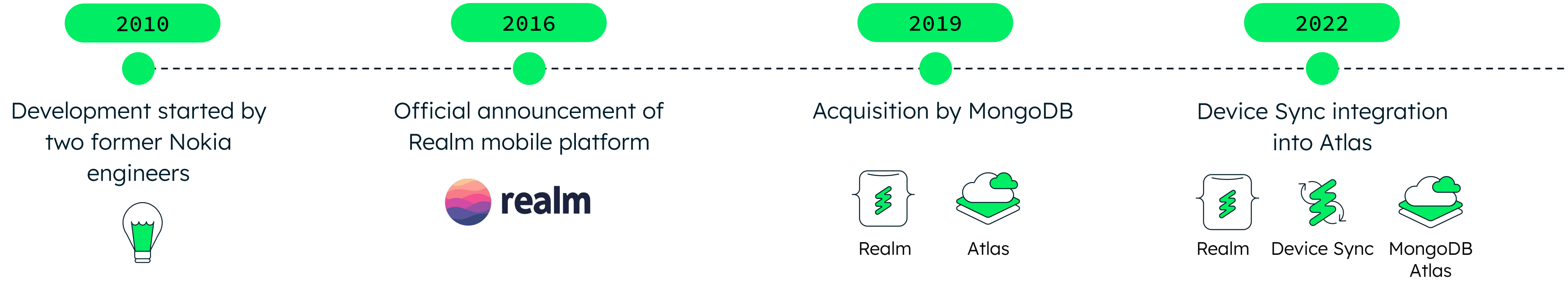
Realm (Device SDK) - Embedded, Object Oriented DB

Offline first paradigm

- Usage: **100k+** developers; **65%** of Fortune 1000; **2B+** app installs
- **Apache 2.0** license
- C++ / .NET / Node.js / Swift / Kotlin / Flutter / React Native / Java …

For developers

- Designed and built for **resource constrained environments**
- Just objects, with **native code paradigms**
- **Live objects** update automatically
- The class definitions ***are*** the database schema



2010

Development started by two former Nokia engineers

2016

Official announcement of Realm mobile platform

2019

Acquisition by MongoDB

2022

Device Sync integration into Atlas

MongoDB®

Document Model / Object Oriented Data Platform

Flexible

Scalable

Always On

Freedom to run anywhere

Documents Are Objects

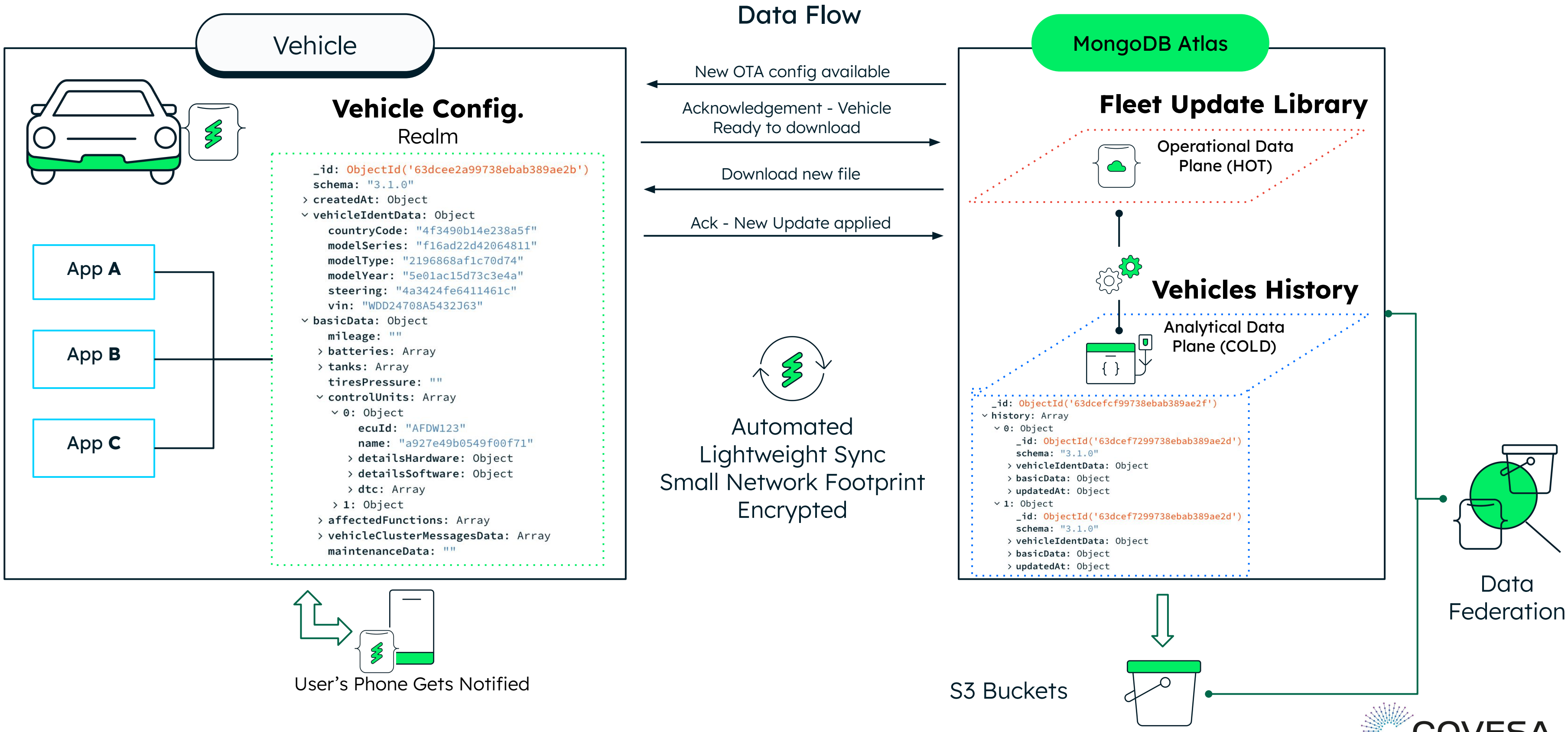
Related data contained in a single, rich document

```
{
  "_id" : ObjectId("5ad88534e3632e1a35a58d00"),
  "name" : {
    "first" : "John",
    "last" : "Doe" },
  "address" : [
    { "location" : "work",
      "address" : {
        "street" : "16 Hatfields",
        "city" : "London",
        "postal_code" : "SE1 8DJ"},
      "geo" : { "type" : "Point", "coord" : [
         -0.109081, 51.5065752]}},
+   {...}
  ],
  "dob" : ISODate("1977-04-01T05:00:00Z"),
  "retirement_fund" : NumberDecimal("1292815.75")
}
```

How it works

Technical capabilities

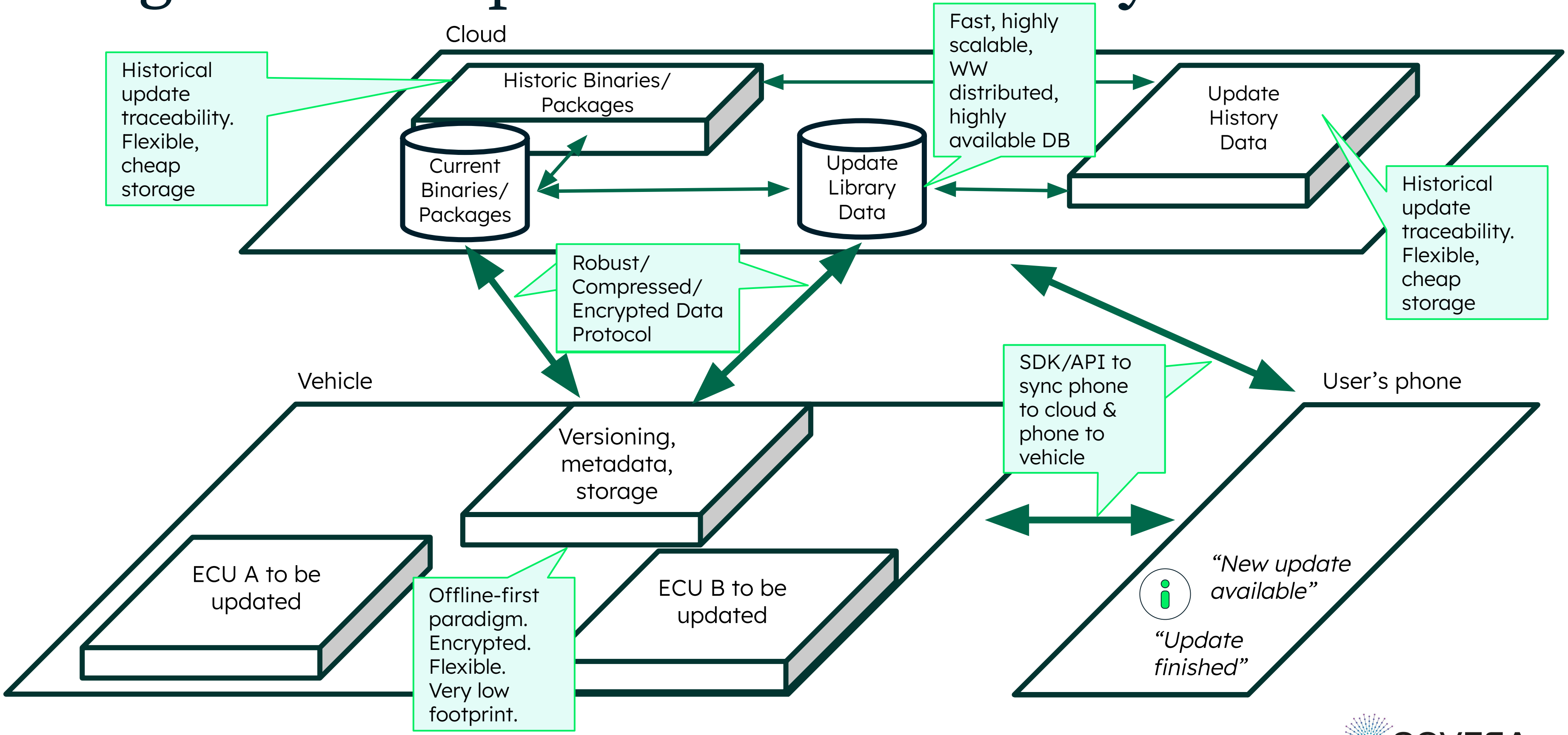
How Atlas Device SDK Facilitates SOTA updates



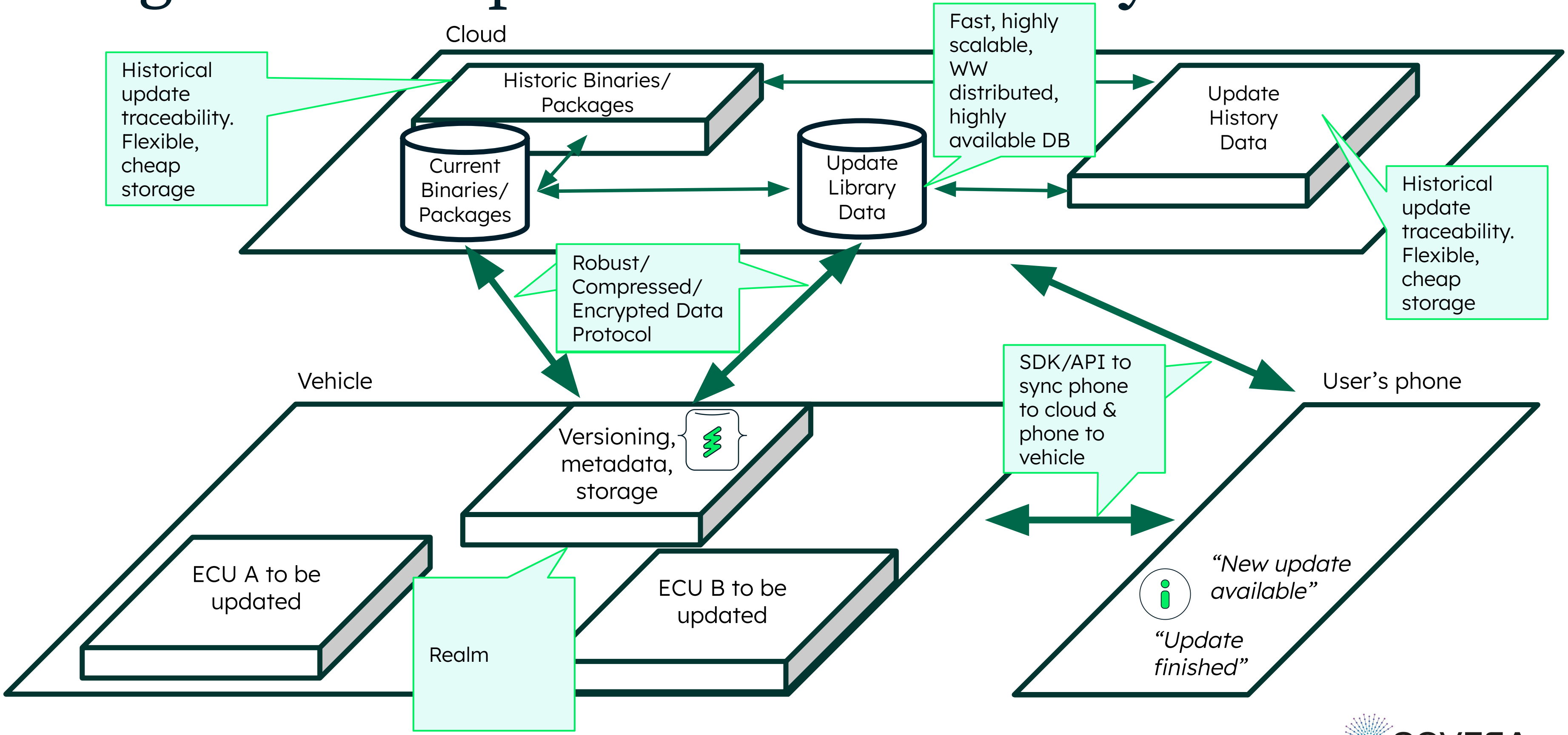
```
  _id: ObjectId('63dcee2a99738ebab389ae2b')
  schema: "3.1.0"
› createdAt: Object
⌄ vehicleIdentData: Object
    countryCode: "4f3490b14e238a5f"
    modelSeries: "f16ad22d42064811"
    modelType: "2196868af1c70d74"
    modelYear: "5e01ac15d73c3e4a"
    steering: "4a3424fe6411461c"
    vin: "WDD24708A5432J63"
⌄ basicData: Object
    mileage: ""
  › batteries: Array
  › tanks: Array
    tiresPressure: ""
  ⌄ controlUnits: Array
    ⌄ 0: Object
        ecuId: "AFDW123"
        name: "a927e49b0549f00f71"
      › detailsHardware: Object
      › detailsSoftware: Object
      › dtc: Array
    › 1: Object
  › affectedFunctions: Array
  › vehicleClusterMessagesData: Array
    maintenanceData: ""
```

```
  _id: ObjectId('63dcefcf99738ebab389ae2f')
⌄ history: Array
  ⌄ 0: Object
      _id: ObjectId('63dcef7299738ebab389ae2d')
      schema: "3.1.0"
    › vehicleIdentData: Object
    › basicData: Object
    › updatedAt: Object
  ⌄ 1: Object
      _id: ObjectId('63dcef7299738ebab389ae2d')
      schema: "3.1.0"
    › vehicleIdentData: Object
    › basicData: Object
    › updatedAt: Object
```

Logical Concepts for robust SOTA system


Cloud
Historic Binaries/ Packages
Current Binaries/ Packages
Update Library Data
Update History Data
Historical update traceability. Flexible, cheap storage
Fast, highly scalable, WW distributed, highly available DB
Historical update traceability. Flexible, cheap storage
Robust/ Compressed/ Encrypted Data Protocol
Vehicle
Versioning, metadata, storage
ECU A to be updated
ECU B to be updated
Offline-first paradigm. Encrypted. Flexible. Very low footprint.
SDK/API to sync phone to cloud & phone to vehicle
User's phone
"New update available"
"Update finished"

Logical Concepts for robust SOTA system



Cloud

Historical update traceability. Flexible, cheap storage

Historic Binaries/ Packages

Current Binaries/ Packages

Update Library Data

Fast, highly scalable, WW distributed, highly available DB

Update History Data

Historical update traceability. Flexible, cheap storage

Robust/ Compressed/ Encrypted Data Protocol

SDK/API to sync phone to cloud & phone to vehicle

Vehicle

Versioning, metadata, storage

ECU A to be updated

Realm

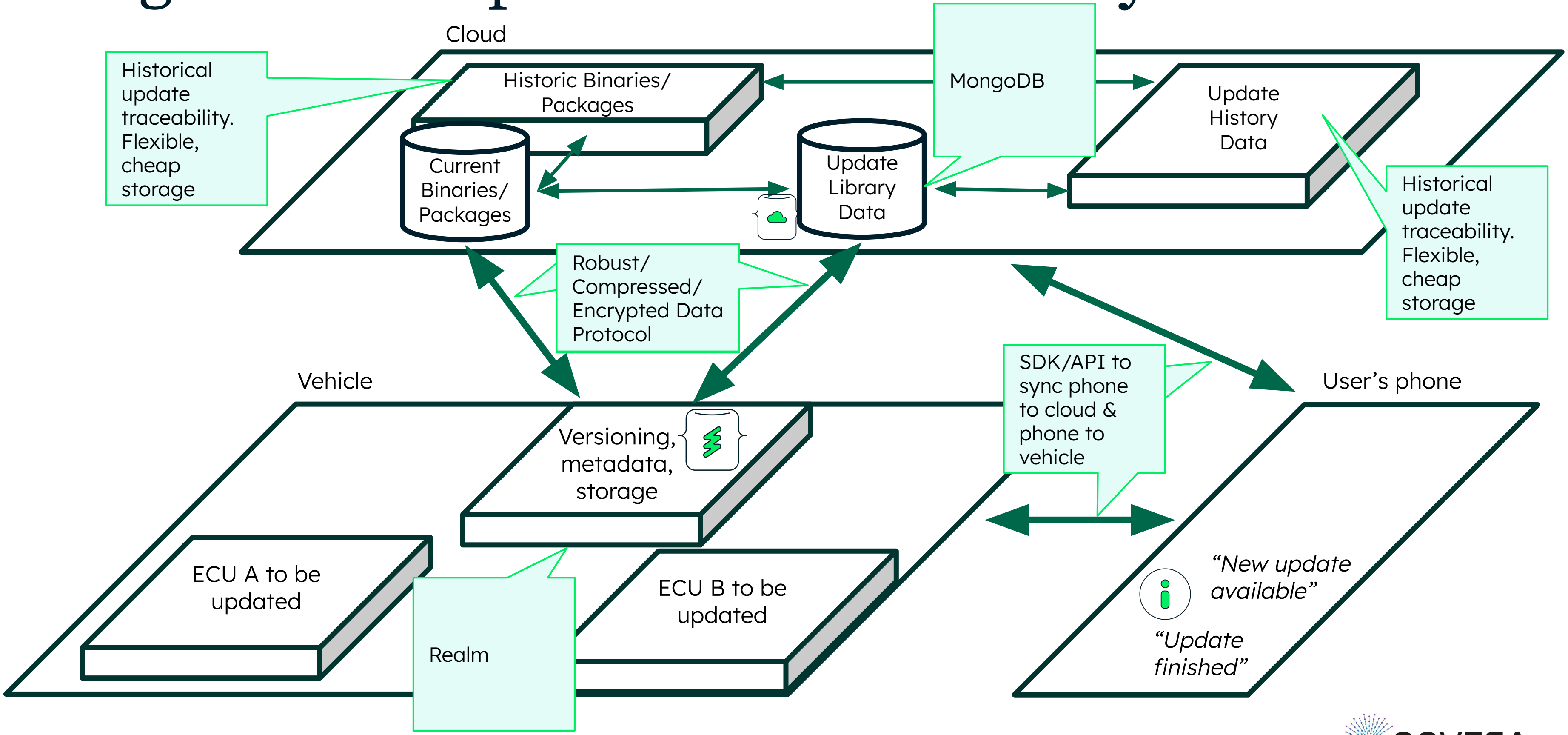
ECU B to be updated

User’s phone

“New update available”

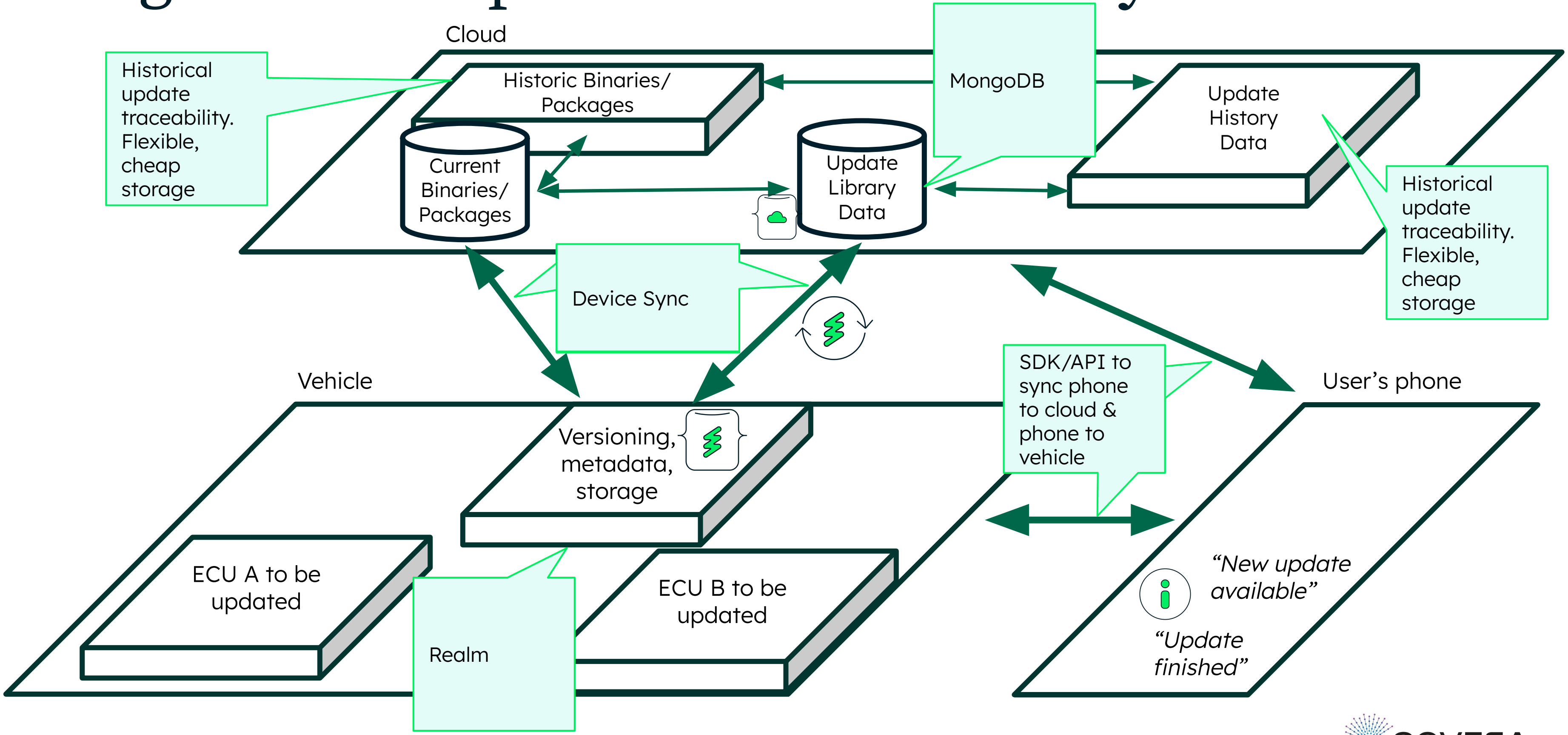
“Update finished”

Logical Concepts for robust SOTA system

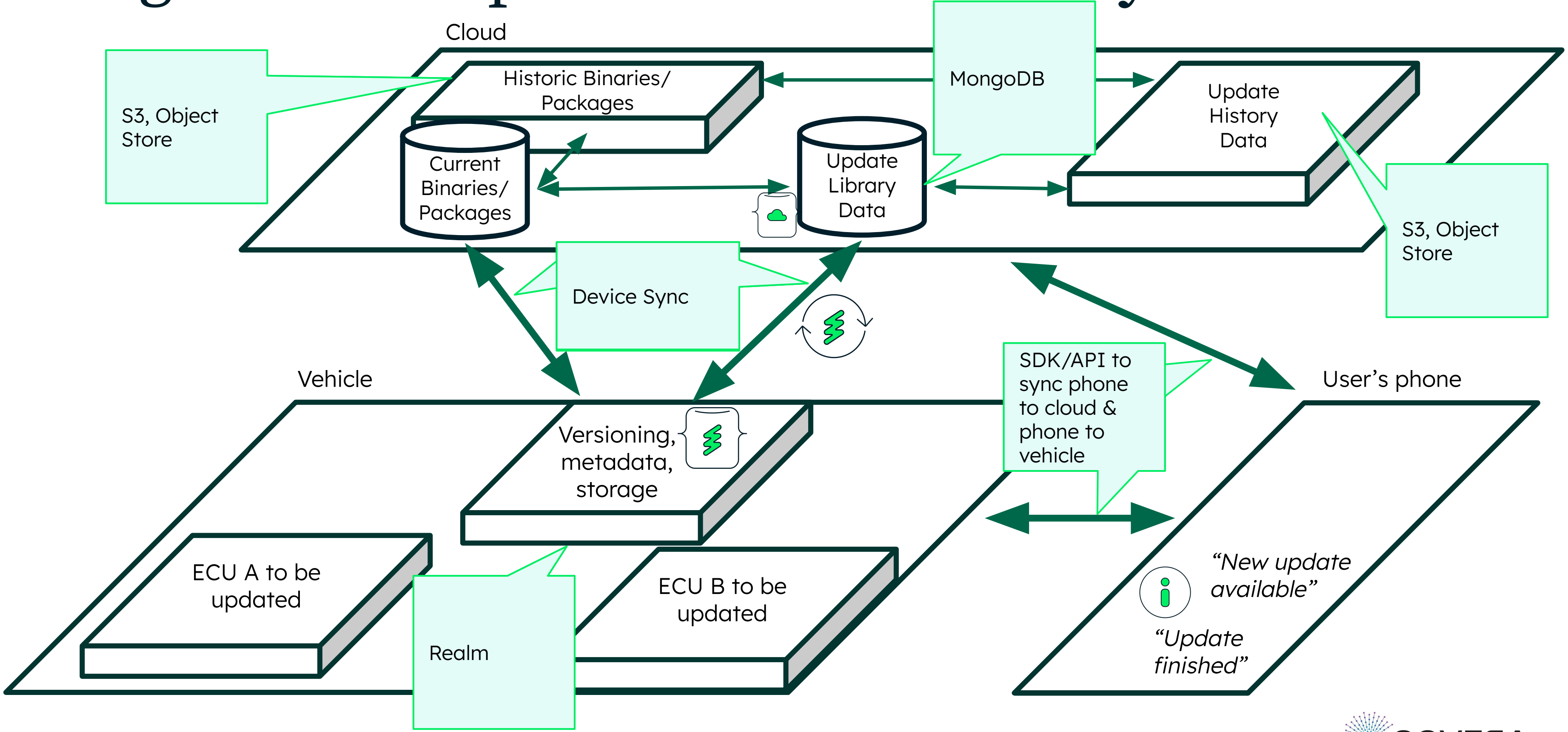


Cloud
Historical update traceability. Flexible, cheap storage
Historic Binaries/ Packages
MongoDB
Update History Data
Current Binaries/ Packages
Update Library Data
Historical update traceability. Flexible, cheap storage
Robust/ Compressed/ Encrypted Data Protocol
Vehicle
SDK/API to sync phone to cloud & phone to vehicle
User's phone
Versioning, metadata, storage
ECU A to be updated
ECU B to be updated
Realm
"New update available"
"Update finished"

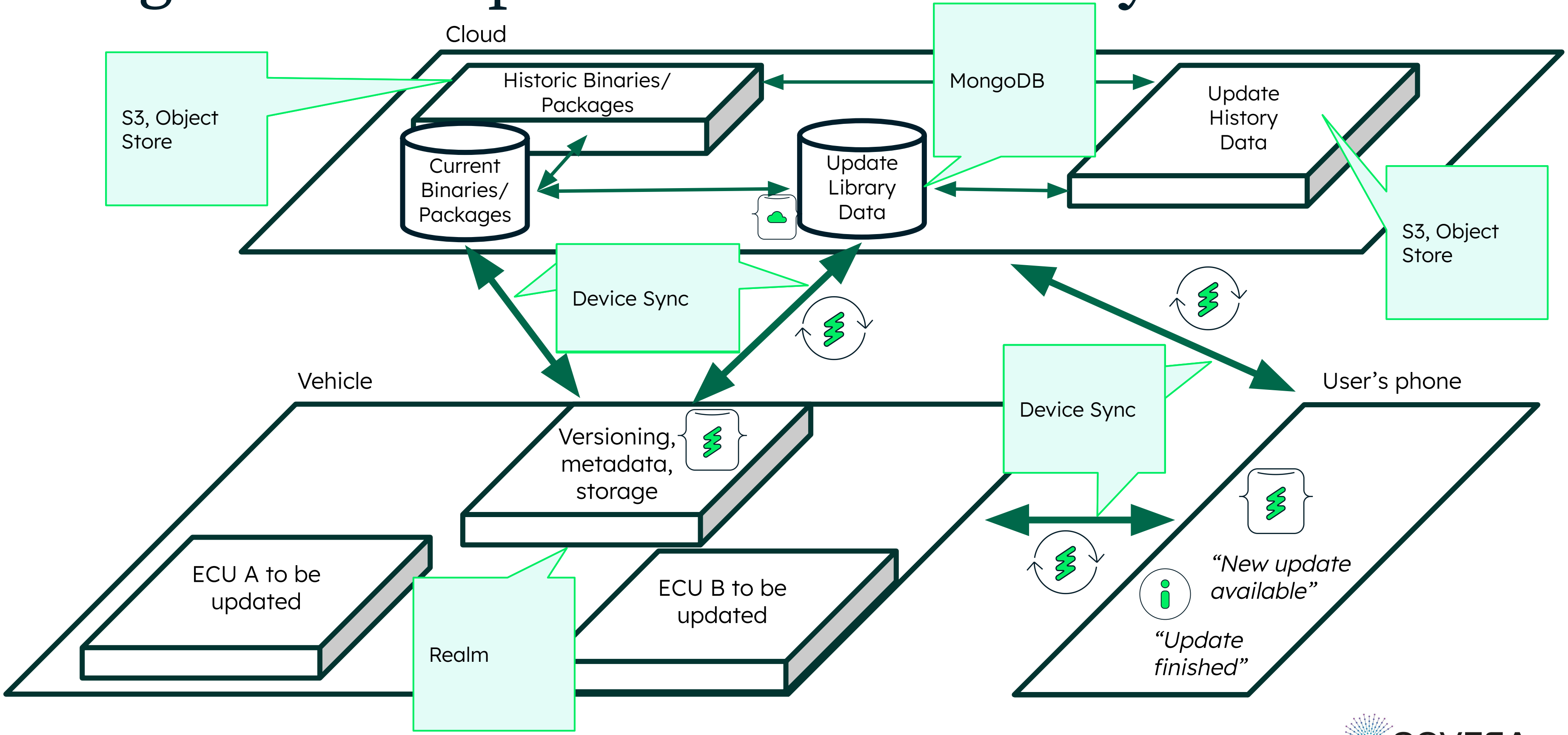
Logical Concepts for robust SOTA system


Cloud
Historical update traceability. Flexible, cheap storage
Historic Binaries/ Packages
MongoDB
Update History Data
Current Binaries/ Packages
Update Library Data
Historical update traceability. Flexible, cheap storage
Device Sync
Vehicle
SDK/API to sync phone to cloud & phone to vehicle
User's phone
Versioning, metadata, storage
ECU A to be updated
ECU B to be updated
Realm
"New update available"
"Update finished"

Logical Concepts for robust SOTA system

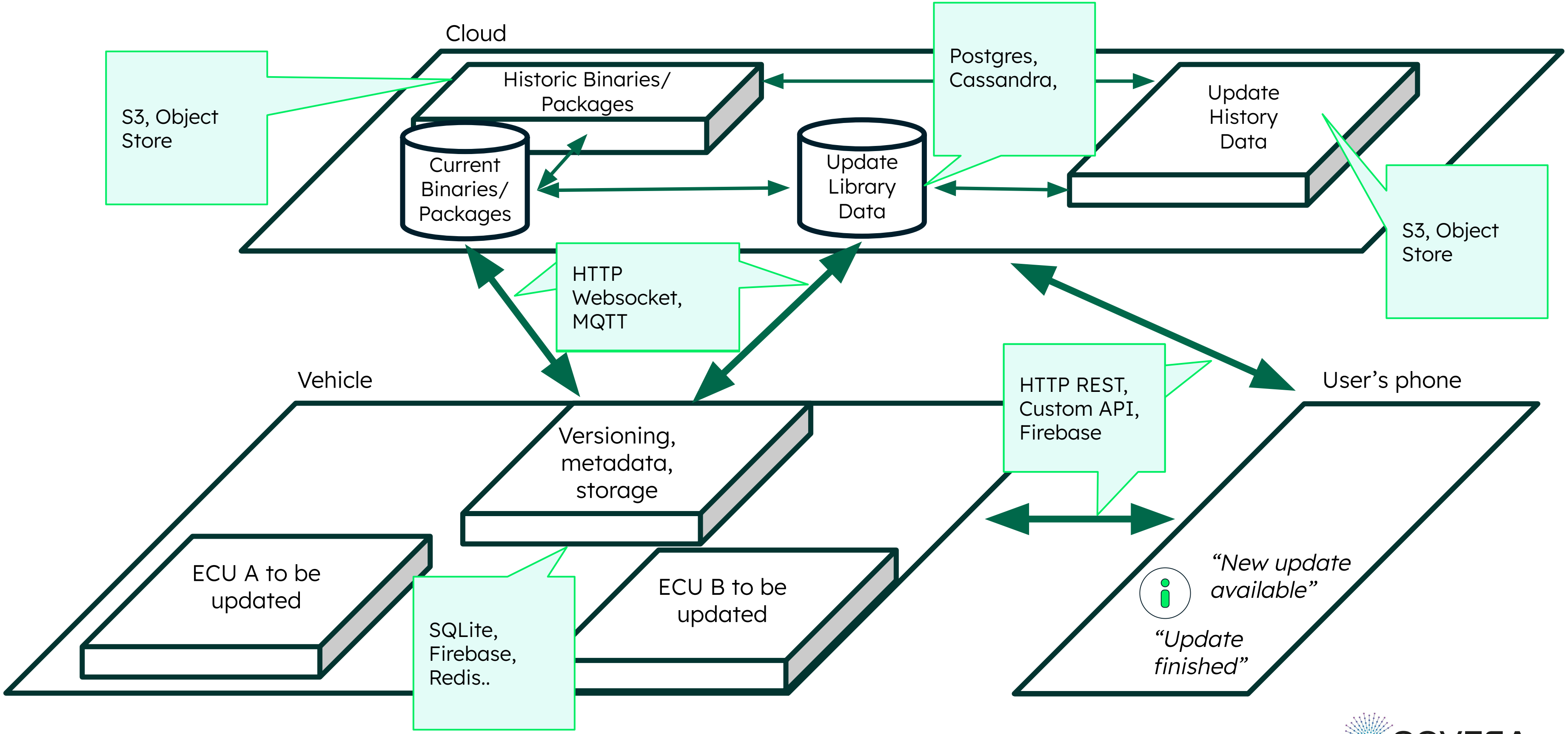


Logical Concepts for robust SOTA system



COVESA

Other Possibilities



COVESA

Accelerating the future of connected vehicles

Curious? -> Reach out

arnaldo.vera@mongodb.com

humza.akhtar@mongodb.com