OVER THE AIR UPDATES

PACCAR ITD

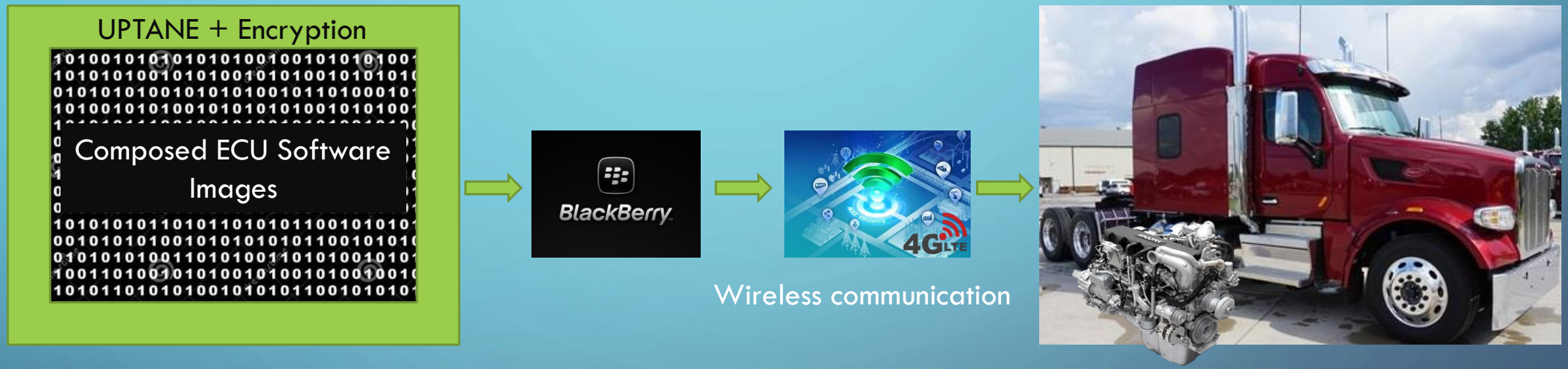
BY DAVID KRUGER

DAVID.KRUGER@PACCAR.COM

AGENDA

- Intro to PACCAR OTA:
- Path to Uptane
- Implementation of Uptane
- Challenges

PACCAR OVER THE AIR UPDATES



ROAD TO UPTANE

- OTA program developed with a security first mindset
  - 2 choices – Uptane or Custom built
- 2017 – NCC group introduced PACCAR to Uptane
- 2018 – Worked with our Supplier for implementing Uptane validation on the Primary (TCU) 2 months of development effort
- 2019 – Penetration tests completed and found zero high or medium vulnerabilities with the packaging and Uptane flow.

UPTANE IMPLEMENTATION

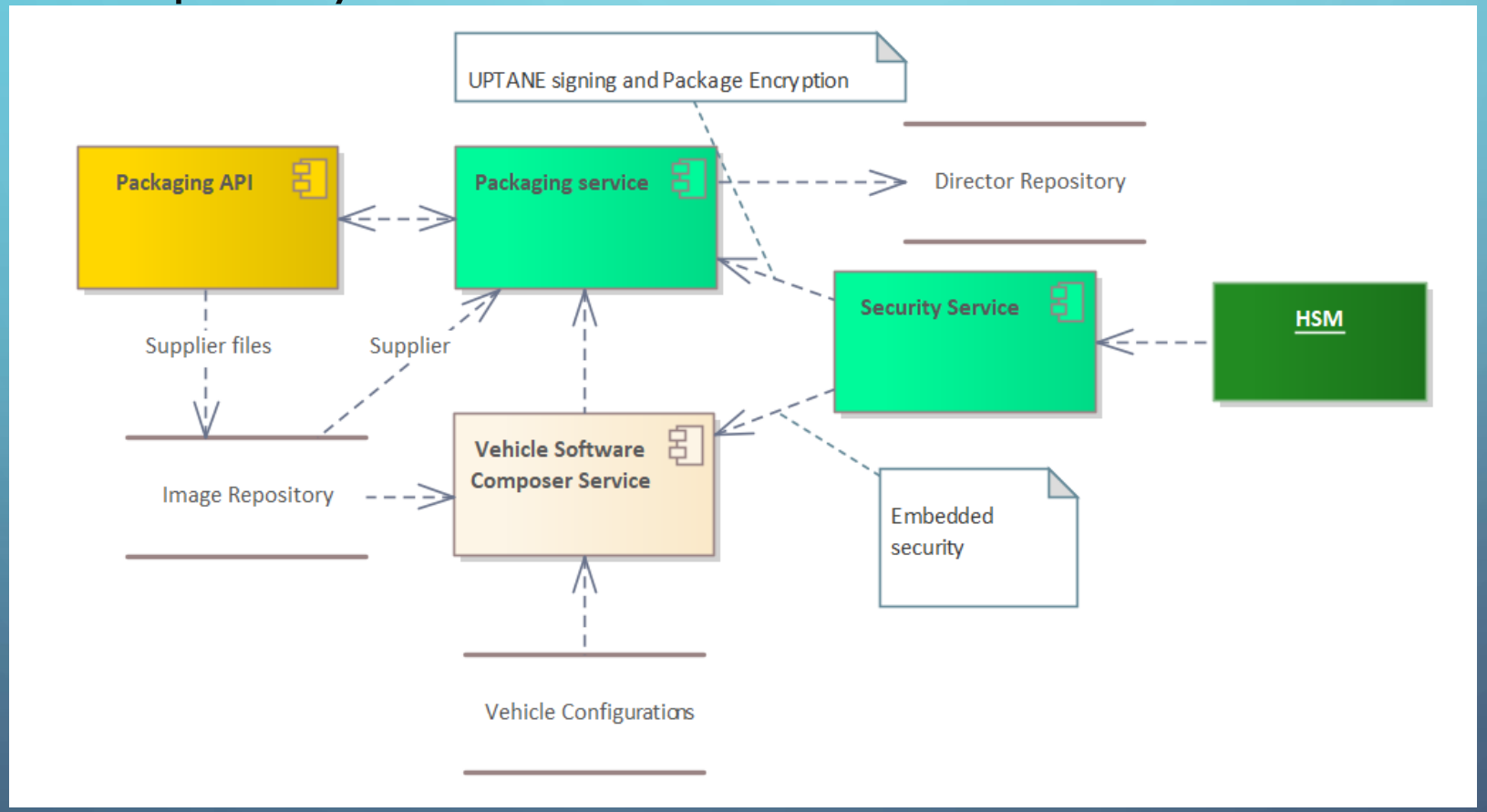
- At the time of this project few public implementations existed and no implementations in .NET
  - PACCAR implemented the Uptane framework with .NET (C#)
  - Example JSON files were found and used to generate the object models.
  - Several adjustments were made while the Uptane standard was being formed.
- Signing algorithms used are mathematically different and supported both by the Primary and the HSM.
- The Image repository is on-premise
- The ‘Director’ role is shared between PACCAR and OTA Supplier.
  - Director repository is a PACCAR AWS S3 bucket

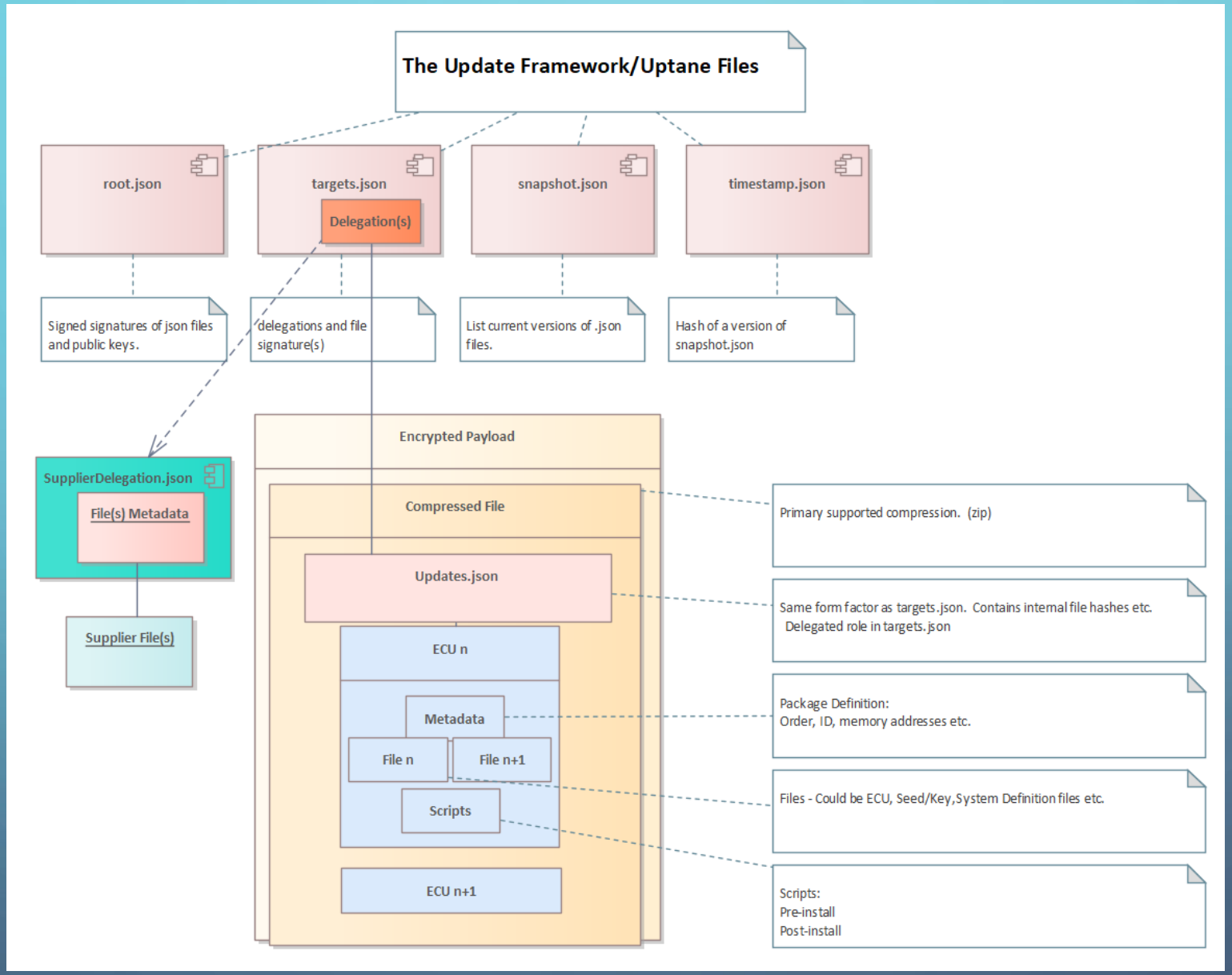
UPTANE IMPLEMENTATION - DIRECTOR ROLE

- PACCAR Backoffice receives inventories and requests to build and secure packages
  - Encrypted packages and Uptane files stored in the Director repository.
- OTA supplier provides signed time, inventories etc. in their Backoffice and on the TCU (Primary)
- The encrypted package and Uptane files are pulled from the Director repository by the primary (TCU).
- Secondaries (ECUs) use existing proprietary security.

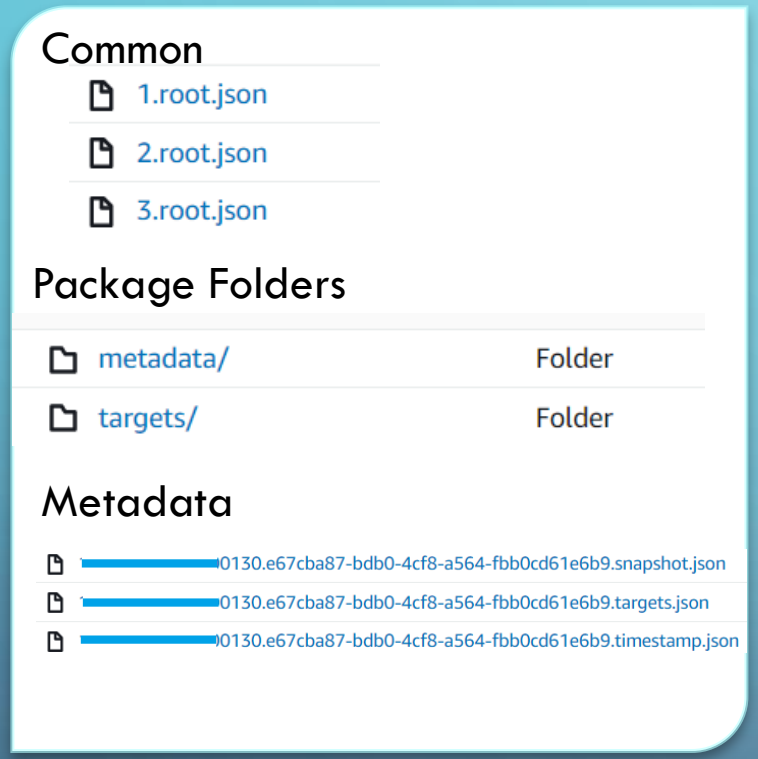
UPTANE IMPLEMENTATION

- Image repository: On prem PACCAR secured server
- Director repository: AWS S3




The Update Framework/Uptane Files
root.json
targets.json
Delegation(s)
snapshot.json
timestamp.json
Signed signatures of json files and public keys.
delegations and file signature(s)
List current versions of .json files.
Hash of a version of snapshot.json
Encrypted Payload
SupplierDelegation.json
File(s) Metadata
Supplier File(s)
Compressed File
Updates.json
ECU n
Metadata
File n
File n+1
Scripts
ECU n+1
Primary supported compression. (zip)
Same form factor as targets.json. Contains internal file hashes etc. Delegated role in targets.json
Package Definition:
Order, ID, memory addresses etc.
Files - Could be ECU, Seed/Key, System Definition files etc.
Scripts:
Pre-install
Post-install

UPTANE IMPLEMENTATION- DIRECTOR REPO



- Director repository: AWS S3
  - Root*.json in a common folder
  - Packages and Uptane files are separated into subfolders for a given update

UPTANE PACKAGE VALIDATION

- Uptane validation of encrypted file begins
  - OTA supplier (Director) and Primary validate time.
  - Root*.json updated if required
  - Timestamp->snapshot->Targets
- Package authenticated and then decrypted
- Package Files extracted
- Files inside of package use a delegated role
  - Inner files use the delegation stored in the targets.json file.

INVENTORY

- Inventory of the vehicle software is maintained frequently to update the Director(s).
  - Service tools may update ECUs outside of OTA
  - Packages expire based on Uptane metadata.
  - Customers can decline updates

UPTANE CHALLENGES

- At the time, Uptane documents were changing quickly
- Using an OTA provider as part of the director role made adhering to Uptane standards challenging.
- Finding supported signing algorithms between the TCU and Backoffice
- No supporting diagrams available for technical implementation of Uptane framework

QUESTIONS