Deployment Plan

Revision

Version 5
9/8/23 1:53 PM

SME

Charles Wilson

Abstract

This document describes the process to deploy software onto the vehicle.

Group / Owner

devops / Information Systems Security Developer

Motivation

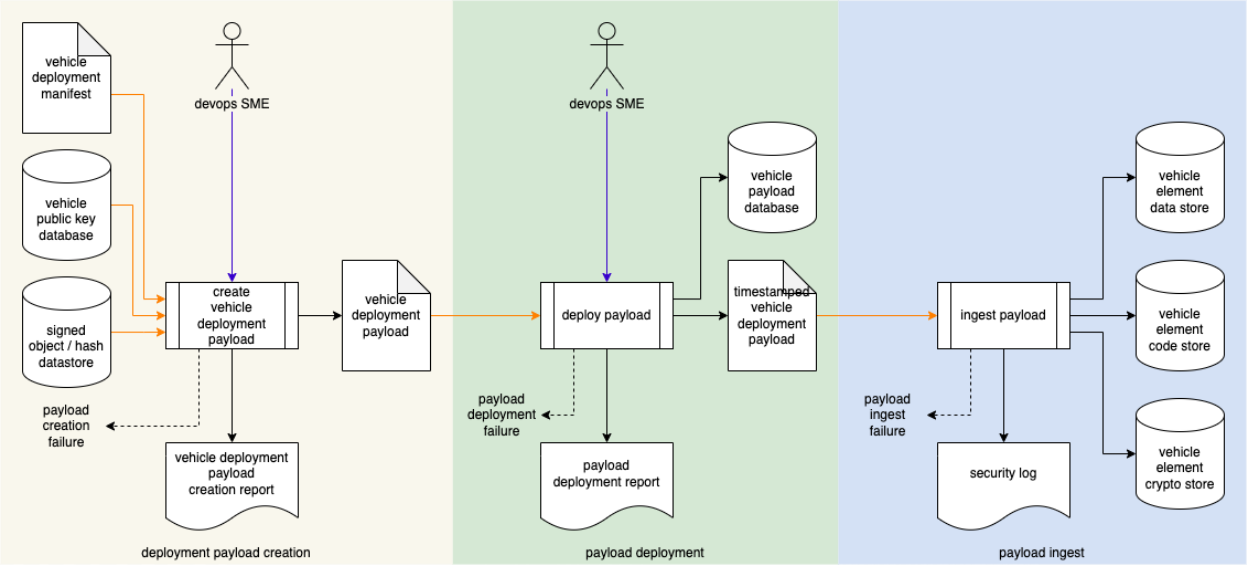
This document is motivated by the need to have formal processes to securely deploy software into safety-critical, cyber-physical systems for certification of compliance to standards such as **ISO/SAE 21434** and **ISO 26262**.

License

Overview

The deployment plan details how software, configuration data, and cryptographic materials are deployed to the vehicle.

The follow diagram shows the workflow to be used:



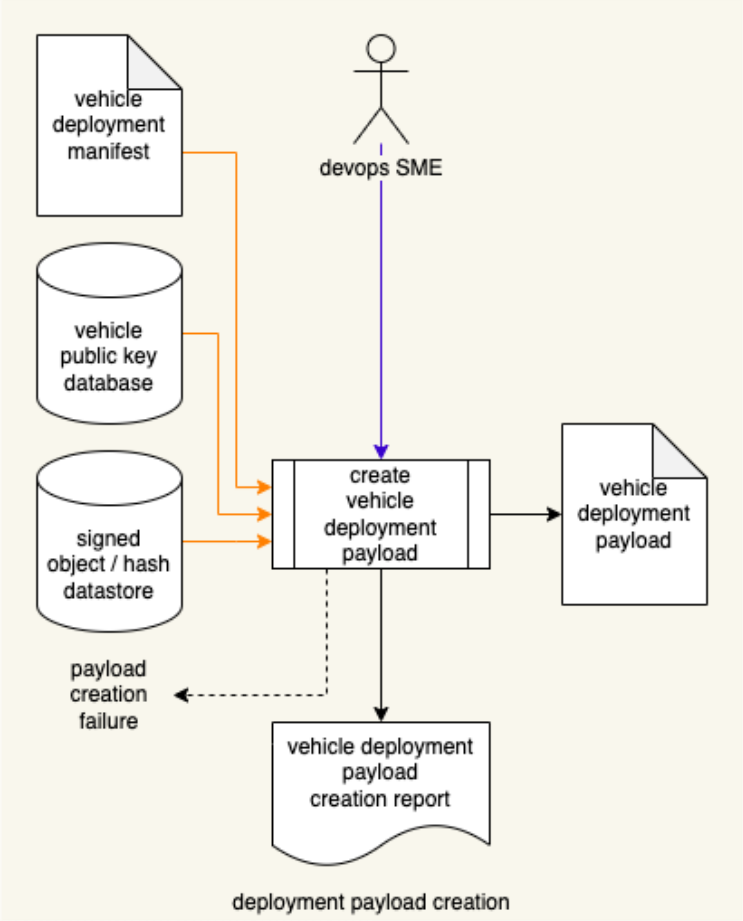
Note: This document does not address initial system provisioning as that process is highly implementation specific.

Note: This process presumes execution of the **Release Integrity Plan** [2].

Process

Deployment Payload Creation

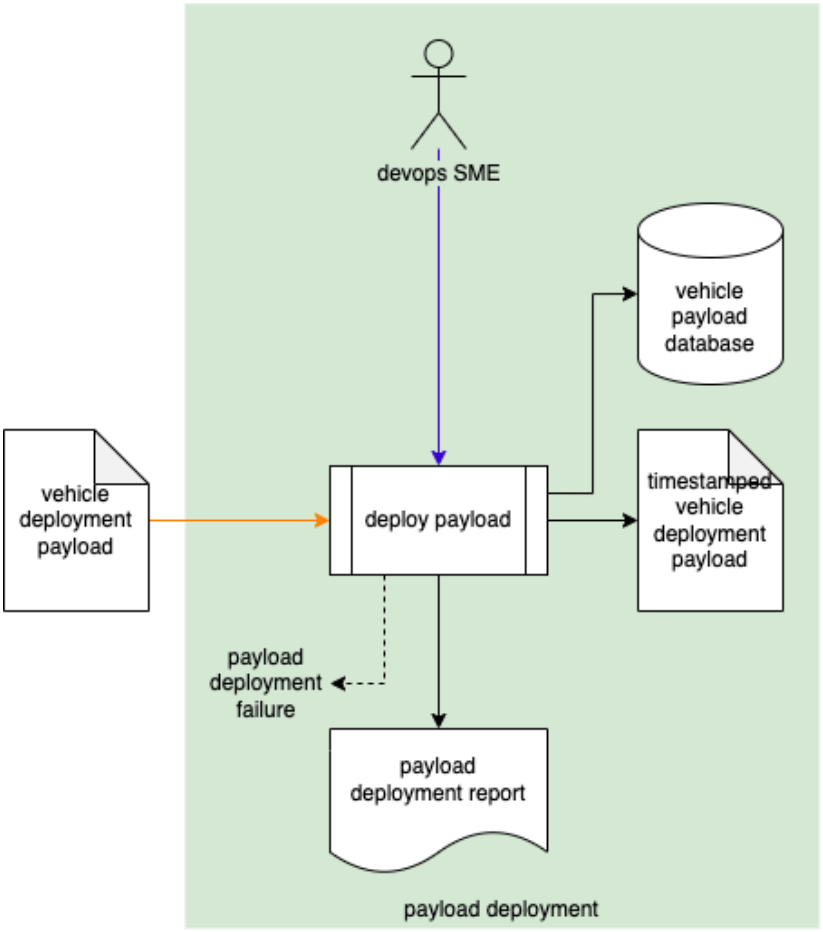
Inputs	signed object / hash database vehicle deployment manifest vehicle public key database
Outputs	vehicle deployment payload
Participants	Devops SME



The Devops SME initiates the **create vehicle deployment payload** process. Using the **vehicle deployment manifest**, the applicable items from the **signed object / hash datastore** are retrieved. These are combined into a package along with the manifest and encrypted using the target vehicle's public key (retrieved from the **vehicle public key database**). This package is referred to as the **vehicle deployment payload**. A **vehicle deployment payload creation report** is created. If the payload is unable to be created, a **payload creation failure** notification is generated.

Payload Deployment

Inputs	Vehicle deployment payload
Outputs	Timestamped vehicle deployment payload Vehicle payload database
Participants	Devops SME

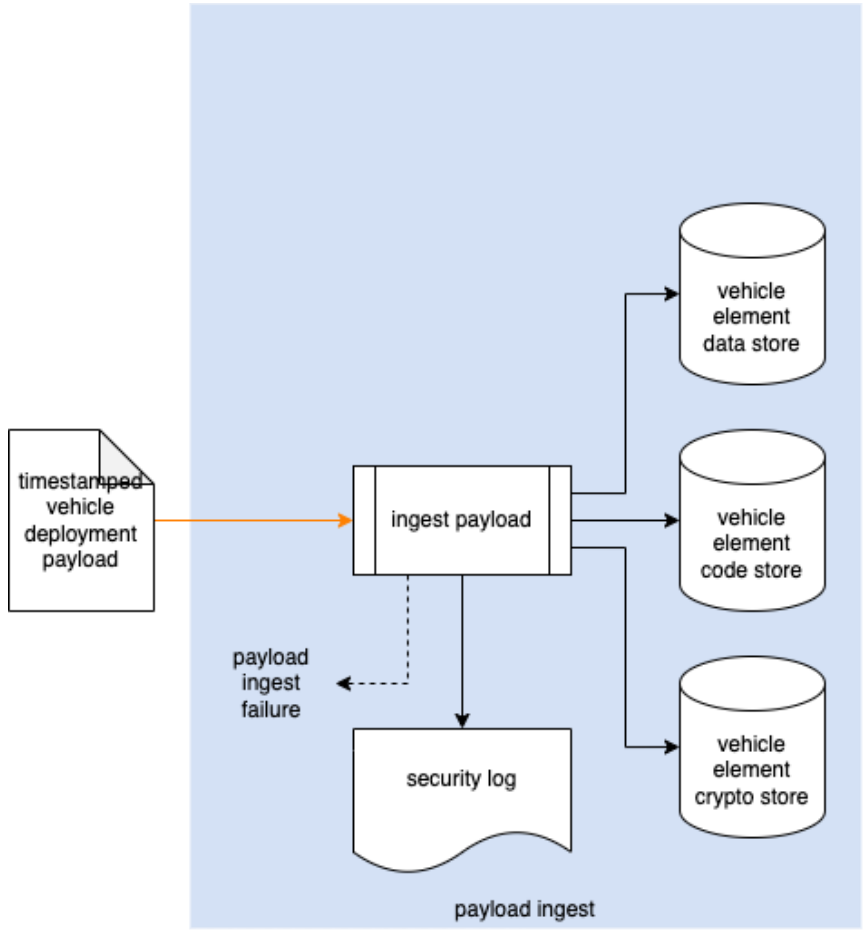


The Devops SME interfaces their deployment device to the vehicle. The **vehicle deployment payload** for the vehicle is taken as input and a deployment timestamp is applied. The **timestamped vehicle deployment payload** is then downloaded to the vehicle and a copy stored in the **vehicle payload database**. A **payload deployment report** will be generated documenting the payload deployment activity. If downloading fails, a **payload deployment failure** notification will be generated.

Note: The specifics of payload downloading are not addressed in this process document as they are both highly implementation specific and the addressed by existing best practices documents [1,3,4].

Payload Ingest

Inputs	Timestamped vehicle deployment payload
Outputs	vehicle element data store vehicle element code store payload deployment report
Participants	**none**



The downloaded **timestamped vehicle deployment payload** is then systematically verified and unfolded. The amount of unfolding depends on the complexity of the system and the number of individual elements being addressed by the payload, as the payload may be either a full (initial) or delta (update). The vehicle state is then checkpointed. Materials extracted from the payload are stored in the **vehicle element data store**, **vehicle element code store**, or **vehicle element crypto store** based on the type of material. Ingest activities will be recorded in the vehicle's **security log**. If ingest fails, a **payload ingest failure** notification will be generated.

Note: Ingest failures require that the vehicle state be rolled back to the checkpointed state.

Implementation Commentary

The following section provides commentary intended to assist the creators of procedures (tertiary documents) implementing this process.

Initial and Update Scenarios

Initial and update scenarios require distinct implementations. The implications that must be considered include storage (multiple executables) and security posture (root-of-trust, identity). It is considered more secure to remove initial provisioning software once provisioning has been completed.

Sensitive Information Handling

It is critical to ensure the proper handling for any sensitive data embodied in the system. This includes:

- Credentials
- Certificates
- PII
- Logs

Secure processes must be emplaced prior to handling any such data. Sensitive data should only be in an unsecure form for as long as it is needed. Unsecured forms of sensitive data must be erased using appropriate mechanisms.

Timeliness

It is critically important to ensure that updates mandated by cybersecurity-related issues are delivered to the customer (be they a higher tier supplier, the OEM, or vehicle owner) in a timely manner. It is understood that there are many factors that contribute to the specific time line for any given update and that these are managed by the organization.

References

1. **Draft Recommendation on Software Updates of the Task Force on Cyber Security and Over-the-air issues of UNECE WP.29 GRVA**
   https://www.unece.org/fileadmin/DAM/trans/doc/2018/wp29grva/GRVA-01-18.pdf
2. **Release Integrity Plan** (AVCDL secondary document)
3. **UN Regulation No. 156 - Software update and software update management system**
   https://unece.org/sites/default/files/2021-03/R156e.pdf
4. **Uptane**
   https://uptane.github.io