

Feature Requirements Report

Feature Description

The aim of this feature is to add a new two-way communication between ADT and BLMS server for several applications. One OTX to be able to get data from BLMS through a key. One OTX to be able to push data in the BLMS through a key.

The user of these OTX libraries is a Renault Diagnostic User.

I believe we should

have two epics for these features:

1. OTX library to write to BLMS server

2. OTX library to read from BLMS

server

I believe we should have some user stories like this:

-> As a Renault Diagnostic User, I want to call an ETAS OTX library so that I can write data to BLMS server.

-> As a Renault Diagnostic User, I want to call an ETAS OTX library so that I can read data from BLMS server.

Some technical explanation about each library:

1. OTX

library to write to BLMS server

-> An OTX calls the OTX library request to write data to BLMS server. It sends the

OTX library data to be written in the form of a key value pair MAP

-> the OTX library calls a host service provider,

which is a service in ADT runtime. The OTX library pass the data to be written. The ADT runtime reads the BIN (Battery Identification Number). If the BIN is missing, the host service provider returns an error Message and the OTX library returns an error boolean and an error message.

-> If the BIN could be read, the ADT runtime calls a

rest endpoint from ADT server which is PUT updateByBin/{bin} and the server performs the write to the BLMS server. This endpoint already exist and there is no new development to be done on server side.

-> the server

sends the status to the ADT runtime and ADT runtime host service provider sends OTX the status and OTX returns error false and a positive message.

-> when the server writes in BLMS, a feedback ADT.SERVICE.BLM is sent to feedback viewer

2. OTX Read BLMS

-> An OTX calls the OTX library with a list of keys. These are values that the OTX wants to read from BLMS

-> The OTX library through a host service provider sends the list of keys to the ADT runtime.

-> The ADT runtime reads the BIN. If BIN could not be read, an error message is returned to the OTX library and the process stops. The OTX library returns an error message to the OTX calling it

-> if BIN could

be read, the ADT runtime sends to the ADT server the keys and BIN through a new rest endpoint that needs to be developped

-> In ADT server side, it needs to perform an http request with a new API. This API is unknown for now. This is new development to be made

-> ADT server then sends a feedback ADT.SERVICE.BLMS of data that has been read from BLMS

-> The server then sends the data and status to ADT runtime

-> ADT runtime then

sends it to OTX library which in turn returns it to the OTX calling it

"

Context

eral OTX scenarios are needed to communicate with the BLMS server. We need to be able to read

and write data through 'keys'. The goal is the same as the BVM Read / Write concept.

The writing of data from

ADT to BLMS through this scenario should be instantly done to let NewDialogys read the data just after the

scenario.

Example for more context:

For electric vehicles, Renault rule states that in the event of an accident with

airbag deployment, the traction battery must be replaced and sent for recycling.

This leads to a significant

increase in the repair cost of damaged vehicles, as in addition to bodywork expenses, the additional cost of

replacing the traction battery must also be covered.

In many cases, this situation leads to:

- High risk of

economically non-repairable vehicle to be scrapped (the repair cost exceeds the car's residual value). From the

perspective of insurers and specialized media, this is a technical and ecological nonsense.

- Insurance

companies compensating for vehicles that are lightly/moderately damaged, even when they are less than 2 years

old in some cases.

With the introduction of EDR (GSR2 regulation) in July 2024, it becomes possible to implement a traction battery diagnostic based on Airbag ECU data.

ADT reads the concerned GSR2 DIDs and

provide a calculated result to state if the battery is reusable or not. These DIDs and the results are sent to BLMS for traceability: unlike the traction battery State of Health (SoH) that is sent automatically to BLMS via ADT, these GSR2 DIDs are GDPR compliant, they shall not be sent automatically to BLMS when we connect ADT to the vehicle as we must have the client agreement to read them and send them to BLMS.

"

Summary

The Two-Way Communication with BLMS Server feature enables Renault Diagnostic Users to seamlessly write and read data to/from the Battery Lifecycle Management System (BLMS) server through ETAS OTX libraries. This feature consists of two primary epics: OTX library for writing data to BLMS server, utilizing an existing endpoint, and OTX library for reading data from BLMS server, requiring development of a new REST endpoint and API integration. The feature provides instant updates for downstream applications, enhances diagnostic capabilities, and offers robust error handling and real-time operational insights through feedback mechanisms. This enhancement is crucial for efficient vehicle diagnostics, particularly for electric vehicles, where it supports the implementation of traction battery diagnostics based on Airbag ECU data, potentially reducing repair costs and environmental impact.

Benefit Hypothesis

We believe that by implementing the Two-Way Communication with BLMS Server feature, Renault Diagnostic Users will experience a significant reduction in diagnostic time and an increase in accuracy, leading to improved overall efficiency and customer satisfaction. Specifically, the instant updates to BLMS will ensure that downstream applications, like NewDialogys, have access to the latest data, enhancing the repair process. Furthermore, the feature is expected to contribute to reduced repair costs for electric vehicles by facilitating informed decisions on

traction battery replacement, ultimately benefiting both Renault and its customers environmentally and economically.

Key indicators of success will include metrics on reduced diagnostic time, increased first-time fix rates, and a decrease in economically non-repairable vehicle cases due to battery replacement costs.

User Stories

US-1: As a Renault Diagnostic User, I want to call an ETAS OTX library to write data to the BLMS server

through a key-value pair MAP, so that I can instantly update BLMS for downstream applications like

NewDialogys to

ensure seamless vehicle diagnostics

Acceptance Criteria:

- The OTX library successfully sends a key-value pair MAP to the ADT runtime.
- The ADT runtime writes the data to BLMS using the existing PUT updateByBin/{bin} endpoint.
- The OTX library receives a success status and a positive message from the ADT runtime.

Estimate: M

Open Questions

1. What is the expected payload size for the key-value pair MAP in write operations, and are there any specific formatting requirements?