Data Persistency in Vehicles

Confidential, Draft

11 Pages

Text in Red is here to help you. Delete it when you have followed the instructions.
The <Document Title> and <Company Name> can be set from the File=>Properties menu. To update it onscreen, press F9. To update all of the fields in the document Select All (CTRL-A), then hit F9. Set the confidentiality and release level by selecting one from each list; Confidential, Members Only or Open Distribution and Draft, Final Draft, Release. The date is set automagically when the document is saved. ***Note that FINAL DRAFT and RELEASE documents should only be released as PDF, NOT AS WORD DOCUMENTS***

Abstract

10 point Arial Centered.

Put information about the purpose of the document and the information that it contains here. This text should not extend beyond this front page. If it does, revise the abstract.

0 Document Information

0.1 Table of Contents

0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in [1].

```
Source code is shown in this typeface.
```

0.3 Revision History

The last named individual in this history is currently responsible for this document.

All Page Within This Box

Revision	Date	Comments
Initial	*04/23/2007*	*Initial draft.* *Maintz, Solid Information Technology, franz-josef.maintz@solidDB.com*

1 Introduction

Describe the state of the RFP and explain how the RFP came into existence. Discuss any other issues that are not addressed in the following chapters.

All Page Within This Box

2 Application Domain

This is a description of the real world, unrelated to computer solutions. E.g. if you must make a text editor, you describe how the authors input text, do review cycles with peers and publish. This section should define the terminology that is used in later sections and paint a clear picture about issues and facts of the problem domain. Do not discuss solutions or problems that are solved, focus on a description of the context of the problem. This should answer questions like: What actors/entities take part in the problem domain? What is their role?

It should describe the world as it is now, not when the solution is applied. This is a real world description, not of a computer system solving the real world problem!

This section should also list assumptions that the RFC may make about the problem context. E.g. the number of users of a gateway shall not exceed 100.

The difficult part is to not describe the solution, or the problem. This section should strictly limit itself to the context of the problem and the solution.

Also, you should list existing products that exist in the problem context. E.g. for the Communications architecture there are Java Phone API, Java Messaging Service and javax.microedition.io that operate in this scope. It is not required to evaluate them, just list them.

2.1 Terminology + Abbreviations

DB = Database

DBMS = Database Management System

ECU = Electronic Control Unit

SQL = Structured Query Language

DRM = Digital Rights Management

3 Problem Description

Detailed description of what problem is actually being solved. This section should answer questions like: What problem is being addressed? Who will benefit from the proposed solution? Why does the OSGi need this RFC?

Future applications require a special focus on data persistance in Vehicles. Today the in-vehicle data id mainly under OEM control. During vehicle services the garage installes update, pathces and feature enhancements usually via OBD diagnose interfaces. These interfaces are not meant for end-user purposes and can´t be accessed without special OBD equipment. Therefore no need for advanced data management to this point.

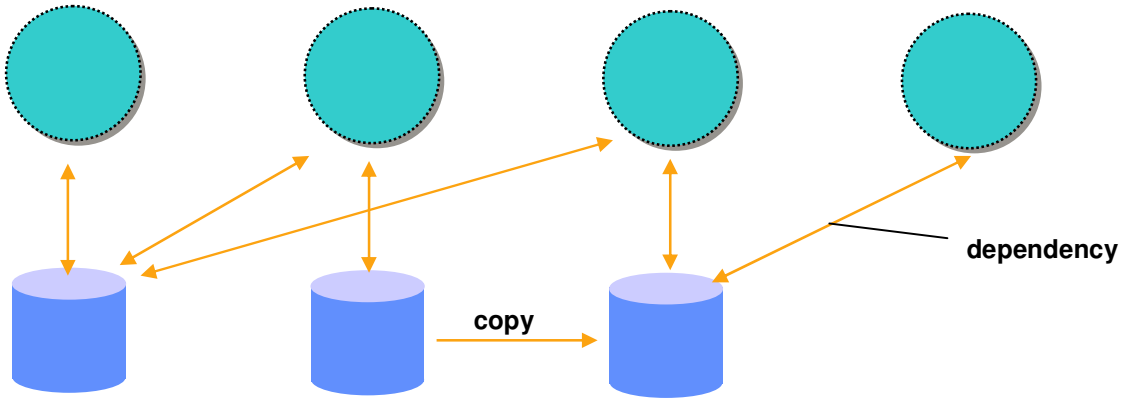
The main facus until today was the vehicle itself and the OEM maintaining the vehicle functionality over it´s lifetime. This will change rapidly. Modern Services around vehicles are end-user centric and the person who buys a service should be able to use this service independant from the vehicle he is using.

One example is a public transportation system where you pay the bus or train per smart card and no longer cash. The in-vehicle payment terminal must identify the smart card and store the user data with timing and route information. At the end of the day the vehicle on-board data must be synchronized with a central server system for exaple using wireless technologies for example WLAN. The central server can accumulate the user data on a monthly base for invoicing.

This sounds easy but has a lot of critical aspects inside. First the data stored in-vehicle is sensitive data for the end-user since his identification is stored along with trip data. This could be useful data for Market studies and shouldn´t be published without end-user´s approval. But also for the vehicle operator this data is important since it´s revenue to him. Secure data storage and 100% safe geographical replication with the Server is a must. In addition the in-vehicle terminal must be always available. If a customer can´t pay due to a technical problem the operator looses money.

That´s why DB replication, DB backup, DB recover from failures, etc are very important to ensure data persistancy in vehicles.

Example of a persistent data structure:



Persistent data structures (files)

Problems:

- *Each application program has its own storage*
- *Programs are dependent on the other program's storage (because they use it)*
- *Programs accessing the same storage face concurency control problems.*
- *Sometimes you have to copy files, inflicting copy dependencies*
- *Maintain overall consistency becomes difficult (especially in the presense of failures).*
- *Use of the storage may be cumbersome.*
- *Maintaining of the storage stucture is work-prone.*

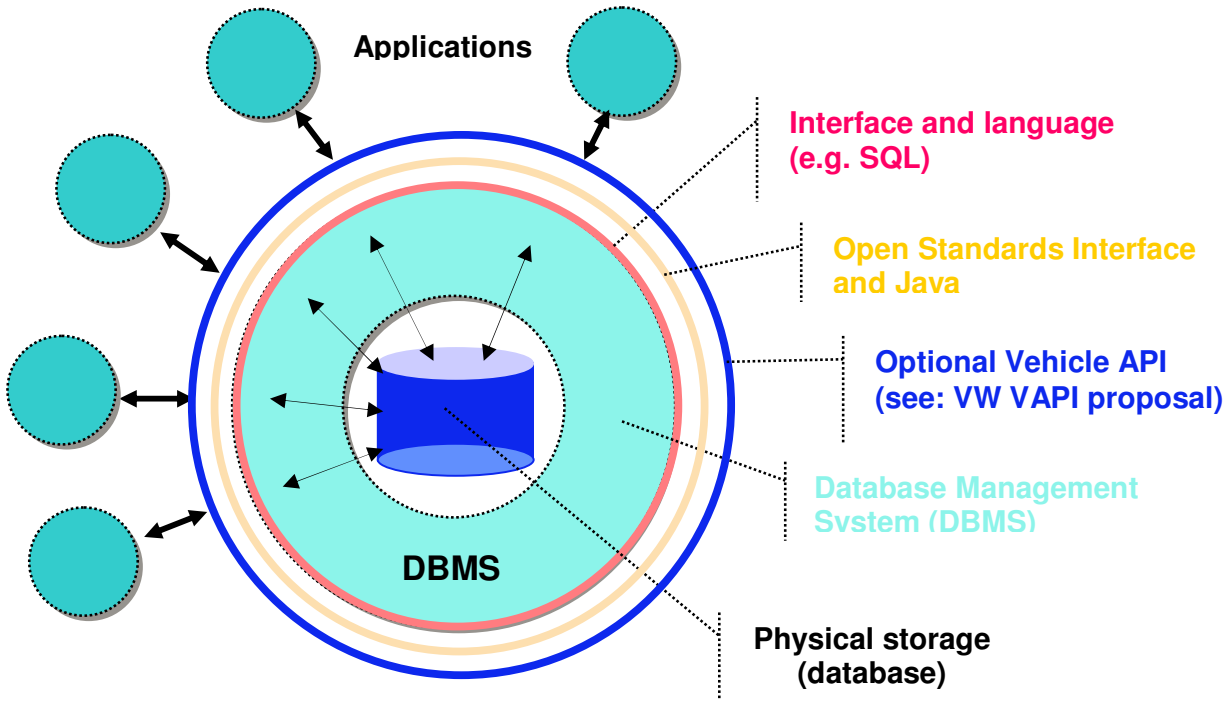
What are the benefits of a DBMS (Database Management System) in vehicles?

The vehicle eco systems is using more and more advanced communication standards originally invented by the Telecom industies. End-user´s like to have similar services provided in vehicles than they are to at home. Like video streaming, MP3 downloads, reading Emails in the vehicle, synchronising addresses and contact data, interfacing portable commodity equipment, etc.

In oder to provide all these services the Telco´s of this world aree using powerful servers with advanced data management software to check the user´s rights to use a special service and to give him immediate access to it. Then the data stream of the music or video must be handled. That´s why databases are heavily used in the Telco industies today and this trend will move into vehicles very fast.

The challenge to be solved are to run all this under the available performance of ECU´s in vehicles, the limited resources in DRAM memory, restrictions in using rotating media´s (Winchesters, DVD´s, etc) and various different Operating Systems used by in-vehicle ECU´s.

For the OSGi VEG the special challenge will be to keep the protocol overhead of the OSGi layer to a minimum in order to allow the data management software to be implemented with highest possible performance.



Challenges of using a Database in a Vehicle

- *Several Interfaces to pass for data to/from in-vehicle bus systems via ECU´s to outside world*
- *Relatively low performance of in-car bus systems (MOST and CAN are no GBit technologies)*
- *Limited RAM memory space in vehicle and limited usability of Winchester disks or DVD´s due to vibrations, temperature, aging, etc*
- *Multi-user access*

Additional Benefits of using a relational Database:

- *Multi-user ACID transactions (Atomic, Consistent, Isolated, Durable)*
- *Machine-readable dynamically defined data definitions*

All Page Within This Box

- *Server-enforced data consistency and integrity*
- *Automatic recovery*
- *Common high level language*
- *Access controls*

Database requirements for Vehicle usage:

- Small Footprint
- Transaction safe operation
- Data replication via different Operating Systems and geographical nodes within and outside the Vehicle
- Standalone In-Memory Database or a combination with File System Storage Engine
- “Always-on” support to ensure uninterrupted database access
- High Availability for secure data management like Toll Collect fees
- Rights Management for Data access and distribution

4 Use Cases

List one or more use case models that describe system functionality.

- *Public Transportation Systems*
  - *Smart Cards to pay the fair. The Vehicle needs to store the passenger and trip details, synchronize the in-vehicle data with a central server system for invoicing.*
  - *Terminals for tickets, Information points, entertainment systems,*
- *Toll Collect*
  - *City, Highway, etc toll systems communication with vehicles*
- *In-Car data management*
  - *Maintain Hardware levels of ECU´s and related Firmware/Software versions for Service and Diagnostics*
  - *Switch over critical applications from one broken ECU to a fully functional unit (drive by wire etc)*
  - *End-user data management (addresses, emails, MP3 lists, voice data, ….)*
- *Car tracking and remote control*
  - *Insurances to charge per distance driven and not via annual fees*
  - *Car tracking and remote control for stolen cars*
  - *Download user´s voice profiles and personal settings to any rental car*
  - *…*
- *Car-to-Car communication*
  - *Vehicles inform each other about traffic jams, ice warnings, hard braking of vehicles in front,…*
  - *Fleet Management*
  - *…*
- *Navigation and Guidance systems*
  - *Download info´s and navigate with temporary events like roads under construction for this time*

- *Off-board Navigation*
- *Temporary buy point of interest data for the region of interest*
- *Dynamically change routes from central servers for transportation companies using lorries more efficiently.*

- *Many more…..*

5 Requirements

List the detailed requirements that should hold for the solution. This should be described as statements that can be tested in some way.

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Requirements & Specifications. Michael Jackson. ISBN 0-201-87712-0

Add references simply by adding new items. You can then cross-refer to them by chosing <Insert><Cross Reference><Numbered Item> and then selecting the paragraph. ***STATIC REFERENCES (I.E. BODGED) ARE NOT ACCEPTABLE, SOMEONE WILL HAVE TO UPDATE THEM LATER, SO DO IT PROPERLY NOW.***

6.2 Author's Address

Name	Franz-Josef Maintz
Company	Solid Information Technology Oy
Address	Stefan-George-Ring 19
Voice	+49 151 19319822
e-mail	Franz-josef.maintz@solidDB.com

6.3 End of Document

All Page Within This Box