openMDM® eclipse working group

Architecture and API workshop

May 6th, 2015

AUDI AG Ingolstadt

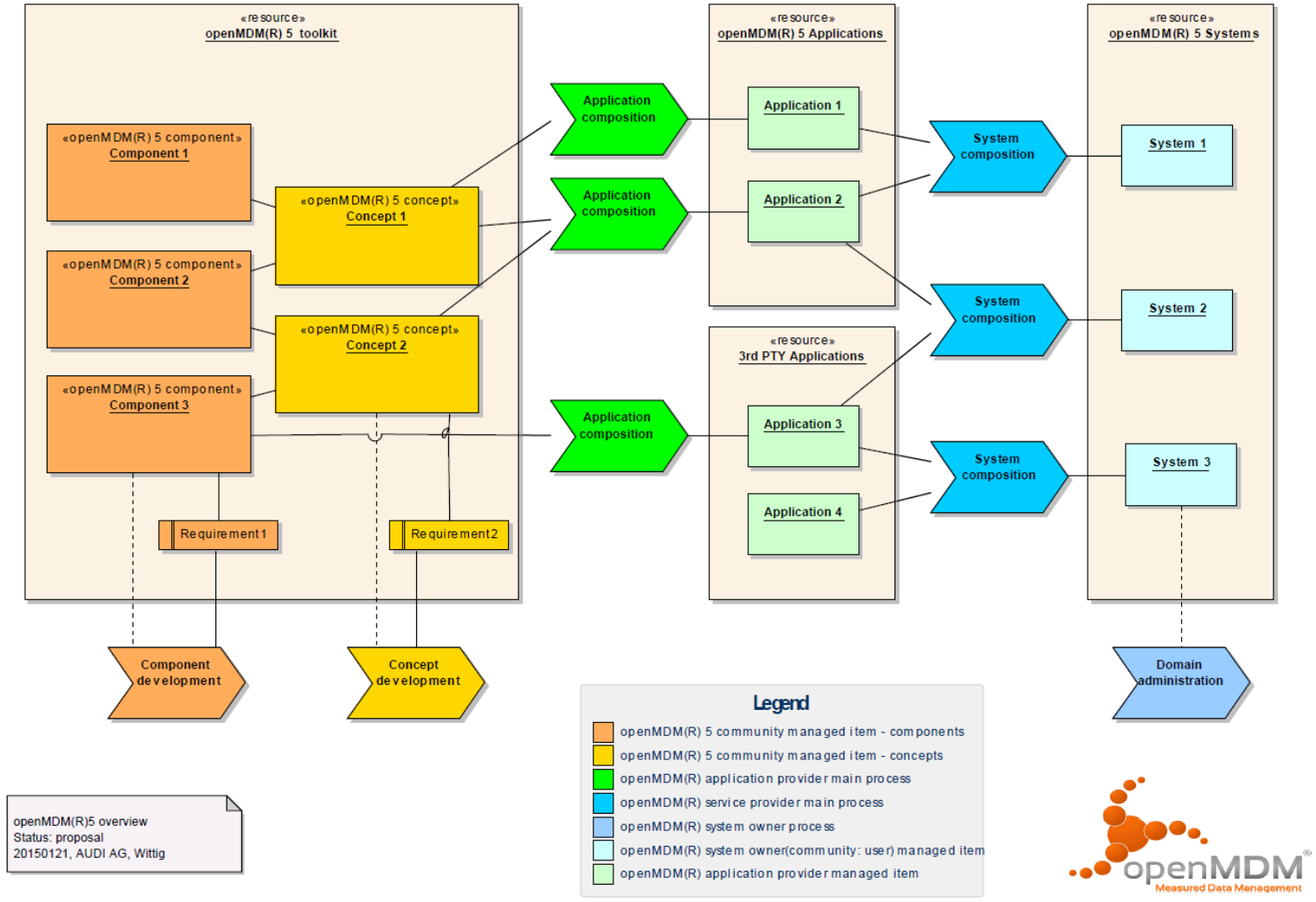
Agenda

10:00 Uhr	welcome	5 Min.	Sven Wittig (Audi)
	current status and goals of the meeting	10 Min.	Sven Wittig (Audi)
10:15 Uhr	contribution offer by HighQSoft	5 Min.	Andreas Hofmann (HighQSoft)
	introduction to HighQSoft HQL	30 Min.	Andreas Hofmann (HighQSoft)
10:45	current status of the API and business layer design	30 Min.	Stefan Beese (EPOS CAT)
11:15	break		
11:30	architecture review and positioning of the items presented	60 Min.	Andres Almiray (Canoo)
	coverage of openMDM® API / BL functions by HQL		all
	consequences of HQL integration to the openMDM® API / BL		
	discussion		
12:30	break / lunch	45 min	
13:15	decision: inclusion of HighQSoft HQL to the architecture		
	further actions to be taken		

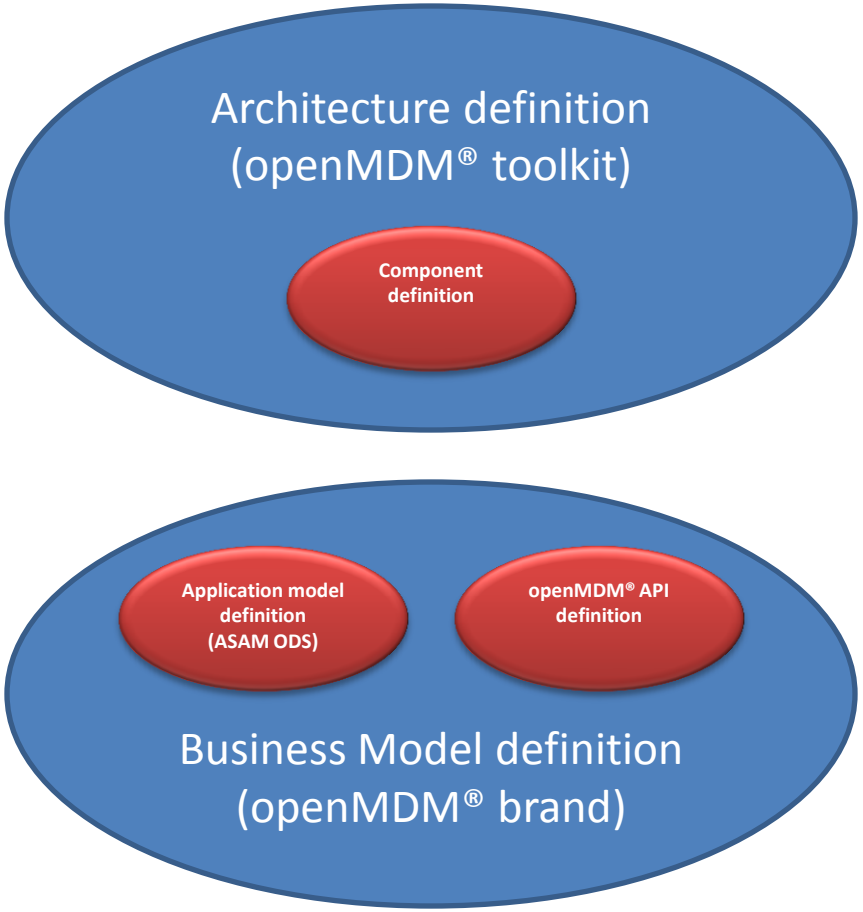
Participants

Andreas.Benzing@ics-ag.de	AC	present
andreas.hofmann@highqsoft.de		present
andres.almiray@canoo.com		present
C.Weyermann@Peak-Solution.de		present
christian.rechner@audi.de	AC	conferencing / temporarily
Gerwin.Mathwig@daimler.com	SC, PL MDM@WEB	present
Hans.Bothe@highqsoft.de	SC	
hans-dirk.walter@canoo.com	QC	present
hj.kremer@peak-solution.de	SC	
m.koller@peak-solution.de		present
reinhard.pirthauer@epos-cat.de		
Sebastian.Dirsch@gigatronik.com		present
sibylle.peter@canoo.com		present
stefan.beese@epos-cat.de	PL MDM|BL	present
Stefan.Ebeling@bmw.de	QC	conferencing / temporarily
stefan.holz@gigatronik.com	SC	present
Sven.wittig@audi.de	SC	present
SWartini@MuellerBBM-vas.de	AC	present
Ulrich.Bleicher@bmw.de	SC	
Viktor.Stoehr@gigatronik.com		conferencing / temporarily

Status Architecture / BL

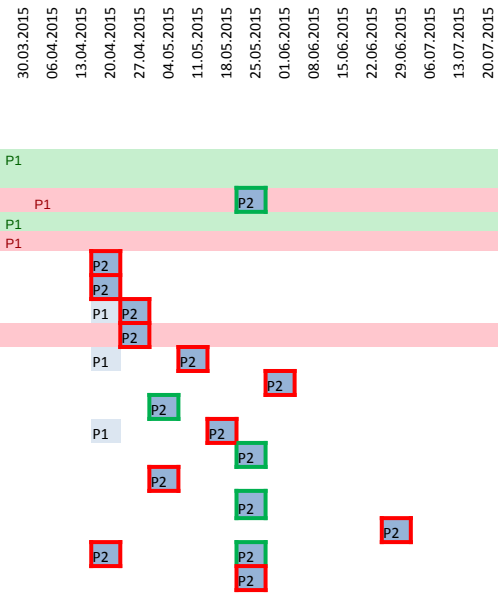


Status Architecture / BL



Status Architecture / BL

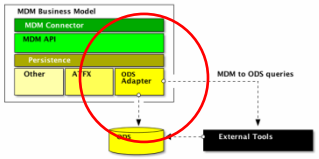
Status	Milestone	Responsible	Item
done	M010000	CANOO	publication of the application framework technology decision proposal for decision
done	M020000	CANOO	publication of the communication technology decision proposal for decision
delay	M030000	CANOO	publication of the first Release of the complete architecture definition for review to the whole community
done	M040000	AC	decision on the application framework technology
delay	M050000	AC,SC	decision on the communication technology
in progress	M060000	AC	review of the architecture
delay	M070000	SC	review of the architecture by openMDM ® community members as community service
delay	M080000	AC	release of architecture definition
delay	M090000	EPOS	publication of the first UML definition of the openMDM® 5 API
delay	M100000	AC	release of the definition of the UML definition openMDM® 5 API
delay	M110000	EPOS	delivery of the MDM API ODS implementation
open	M120000	EPOS	delivery of the modularized application model
delay	M130000	AC	release of the definition of the openMDM® 5 application model
open	M140000	EPOS	delivery of the installable application model version (suitable for setting up an openMDM® 5 ODS server)
waiting	M150000	EPOS	project setup for business layer
open	M160000	EPOS	delivery of the first application frame by April 30th
open	M170000	EPOS	delivery of the openMDM® integration server
open	M180000	CANOO	declaration of which interfaces are openMDM® core
open	M190000	EPOS	delivery of the core business interfaces definition



P2 second planning update, Milestone estimated

P2 second planning update, Milestone confirmed by responsable

HQL positioning within architecture



For static parts of the ASAM ODS Application model incl. Handling of files and their properties

Tbd: For dynamic / generic Parts of the ASAM ODS Application model

Query Collection

PlainOldJavaObjects

Dynamic Properties Support

openMDM® 5 API

Incl. mass storage handling

Criteria API (JPA)

parse

String Query

DS Function

Tbd: streaming of data

Persistence

other

ASAM ODS ATFX

ODS client

HQL

ASAM ODS DB Adapter

Notes - Introduction to HQL

Motivation
- Complex ODS Query Structure
- Implementation Pitfalls
- Multiple Transactions Impossible

Requirements
- Independent of ODS Model
- Aware of ODS Base Model
- Complete ODS API Functionality
- Similar to SQL

Architecture
- ANTLR Parser provides Text Interface to Object-Oriented Layer
- OO Layer Implements ODS Calls

Implementation
- Application Examples

Visions & Ideas
- Support for Aggregation Functions
- No Support for ValueMatrix
- Implement JPA Interface

Discussion
- Lazy Loading
  - Results still contains ODS information
  - Result freed from CORBA, no background loading
- Session Handling
  - Transparent creation of additional CoSessions
  - Initial session created manually and passed to HQL engine

Notes – API Status

Model Composition

- Overview
- Minimal Model
- ASAM Model Extensions
- openMDM Model Extension
  - Security
  - Workflow

API Structure

- Performance vs usability on different levels of abstraction
  - openMDM4 approaches
  - Possibilities for openMDM5
- Use cases for business objects
- Use cases for test data

Discussion

- Motivation
  - Modules are present because ODS is available, not because ODS is required
  - Assuming a generic data storage rather than ODS, what is the result?
  - One benefit of sticking to ODS for all data is simplified provenance
- Application
  - How are Model and Extensions used? Available as ATFX-Files which can be used to set up a new database with ODSCompare
- Adaptation
  - How can data model be changed when the application changes?
  - Static vs dynamic attributes and entities → available, presentation refers to static model only
  - openMDM model has base types for all types of entities, i.e. entity Vehicle is an instance of UnitUnderTest
  - Java Classes representing entities are required to leverage Java features, solution similar to JPA but for ODS is required

Next Steps

- Finish UML specification of API (was proposed for April)
- Approval by AC required

Notes – Architecture Overview

Walkthrough of Architecture Specification
HQL fits in MDM Business Model – Persistence Layer
Discussion

- External applications
  - More detailed description in specification
  - Example should be added for a Matlab Component
- Agreement on position of HQL

Architecture Discussion
Abstraction from ODS required

- HQL vs openMDM API?

References
https://bugs.eclipse.org/bugs/show_bug.cgi?id=453767
https://bugs.eclipse.org/bugs/show_bug.cgi?id=457424

Committments

After a detailed discussion of the HQL features presented by Andreas Hofmann and the positioning of HQL within the openMDM® architecture the participants of the meeting agree on the following statements:

- HQL fits to the openMDM® 5 architecture
- HQL simplifies the implementation of the openMDM® 5 API significantly
- HighQSoft HQL provides ASAM ODS 5.3 compatibility, that is, any ASAM ODS server can be deployed in the lower layers

Therefore the contribution of HQL by HighQSoft is highly welcome. The participants recommend the AC and SC to act respectively.

ToDo ´s

- The terms used within the architecture documents have to be reviewed, if necessary cleared and included to the openMDM® EWG‘s glossary. Conflicts between the wordings of „old MDM style“ and the new architecture have to be resolved (CANOO).
- Description of what the modules mentioned in the architecture picture (Criteria API, DS function, String query..) do (CANOO).

- Identifying dynamic and static parts of the application model, map them into API functionality (EPOS, within existing milestone API UML Design)
- Structuring of the API with respect to the functionality presented to the layers above (EPOS, within existing milestone API UML Design)
- Structuring of the API with respect of the functionality accessed from lower layers (EPOS, within existing milestone API UML Design)

Open Issues

This is a part containing questions which arose writing the minutes. I attach them, because they are related to the API but not part of the project.

- What happens, if possibly the application models of different data sources differ (at least in their dynamic parts)? Which way conflicts are resolved?
- Who will provide the openMDM® connector or at least a first implementation to get an initial system to work?
- Within the architecture pictures it should be reflected, that ATFX also is standardized by ASAM ODS. Therefore, a wording like „ASAM ODS ATFX“ and „ASAM ODS DB“ seems better – even because an ASAM ODS application model applies to both of them. It should be cleared, if the application model structure definied by EPOS must apply th the ATFXes also or not.