openMDM® eclipse working group

Architecture and API workshop

May 6th, 2015

AUDI AG Ingolstadt
<table>
<thead>
<tr>
<th>Time</th>
<th>Item</th>
<th>Duration</th>
<th>Speaker(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:00 Uhr</td>
<td>welcome</td>
<td>5 Min.</td>
<td>Sven Wittig (Audi)</td>
</tr>
<tr>
<td></td>
<td>current status and goals of the meeting</td>
<td>10 Min.</td>
<td>Sven Wittig (Audi)</td>
</tr>
<tr>
<td>10:15 Uhr</td>
<td>contribution offer by HighQSoft</td>
<td>5 Min.</td>
<td>Andreas Hofmann (HighQSoft)</td>
</tr>
<tr>
<td></td>
<td>introduction to HighQSoft HQL</td>
<td>30 Min.</td>
<td>Andreas Hofmann (HighQSoft)</td>
</tr>
<tr>
<td>10:45</td>
<td>current status of the API and business layer design</td>
<td>30 Min.</td>
<td>Stefan Beese (EPOS CAT)</td>
</tr>
<tr>
<td>11:15</td>
<td>break</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11:30</td>
<td>architecture review and positioning of the items presented</td>
<td>60 Min.</td>
<td>Andres Almiray (Canoo)</td>
</tr>
<tr>
<td></td>
<td>coverage of openMDM® API / BL functions by HQL</td>
<td></td>
<td>all</td>
</tr>
<tr>
<td></td>
<td>consequences of HQL integration to the openMDM® API / BL discussion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12:30</td>
<td>break / lunch</td>
<td>45 min</td>
<td></td>
</tr>
<tr>
<td>13:15</td>
<td><strong>decision: inclusion of HighQSoft HQL to the architecture</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>further actions to be taken</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Participants

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

Legend:
- openMDM(R) 5 community managed item - components
- openMDM(R) 5 community managed item - concepts
- openMDM(R) application provider main process
- openMDM(R) service provider main process
- openMDM(R) system owner process
- openMDM(R) system owner (community: user) managed item
- openMDM(R) application provider managed item
Status Architecture / BL

Architecture definition (openMDM® toolkit)

Component definition

Business Model definition (openMDM® brand)

Application model definition (ASAM ODS)

openMDM® API definition
### Status Architecture / BL

#### Milestone

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

#### Milestone Timeline

- **30.03.2015**: Milestone 0
- **06.04.2015**: Milestone 1
- **13.04.2015**: Milestone 2
- **20.04.2015**: Milestone 3
- **27.04.2015**: Milestone 4
- **04.05.2015**: Milestone 5
- **11.05.2015**: Milestone 6
- **18.05.2015**: Milestone 7
- **25.05.2015**: Milestone 8
- **01.06.2015**: Milestone 9
- **08.06.2015**: Milestone 10
- **15.06.2015**: Milestone 11
- **22.06.2015**: Milestone 12
- **29.06.2015**: Milestone 13
- **06.07.2015**: Milestone 14
- **13.07.2015**: Milestone 15
- **20.07.2015**: Milestone 16

**Second planning update, Milestone estimated**

**Second planning update, Milestone confirmed by responsible**
HQL positioning within architecture

openMDM® 5 API

Query Collection → PlainOldJavaObjects → Dynamic Properties Support

Criteria API (JPA) → String Query → DS Function

Persistence

Incl. mass storage handling

Tbd: streaming of data

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

Incl. ODS client

HQL

ASAM ODS DB Adapter
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 defined by EPOS must apply to the ATFXes also or not.