Sprint 1 – WK49-WK50

- Prerequisites:
  - MDM4 API documentation
  - MDM4 business object model
  - Eclipse tooling definitions (maven as build)
- Common Goals:
  - Define MDM API as a valid component and its position
    - MDM API is the Location of the business logic
    - Review and evaluate the MDMAPI features and decide on their integration into MDM 5
    - Define new features necessary in the API
    - Define compatibility of MDM4 and MDM5 application model
    - Define locking of openMDM® 5 application model version and API version
- EPOS
  - Obtain features of the openMDM® 4 API
- CANOO
  - Define Main Goals for the Architecture and publish them for further maintenance in the MDM@WEB Project
  - Define interface spec language
- Results:
  - Initial Feature set of MDM API 5
  - Initial Version of the architecture picture, showing API and its position with dependent components
  - Resolved and clearified and documented, how MDM and ODS relate to each other
  - Proposal if MDM API should be handled as a separate project or not

Sprint 1 -> 2 Bridge (20150108)

Place: Audi Ingolstadt
Date: 2015-01-08
Participants: Hans-Dirk Walther, Andres Almiray, Reingard Pirthauer, Stefan Beese, Christian Rechner, Sven Wittig

- Requirements will be documented until further decision on community level in Jira. The Jira instance will be provided by CANOO AG without additional costs. CANOO provides access for EPOS and Audi until 20150116. When a community solution is available, the requirements will be migrated.
- The catalogue of baseline documents will be stored in a cload storage provided by CANOO AG. The catalogue will be maintained by CANOO AG and must be referenced within the new generated architecture documentation.
- All requirements have to be documented with the corresponding stakeholders. Architecture decisions have to be taken and documented with reference to the corresponding requirements.
- Text documentation will be provided in DITA until further decision. If necessary the doc sources will be migrated afterwards. The DITA tooling can be obtained from Audi on special request.
- All documents generated by this project must be published under the EPL on the eclipse sites from now on. Until further decision the generated documents will be published. The document sources will be migrated if necessary and published when documentatioon methodology and tooling for the community are defined.
- A listing of openMDM® components published with the last integration client will be provided by Audi
- The API / business layer project will be placed within a new eclipse project „openMDM® 5 business layer“ driven by Audi.
- For the openMDM® API interface definition UML will be used (EPOS will work with enterprise architect)

Sprint 2 – WK02-WK04

- Prerequisites:
  - Agreement on the features for the MDM API 5 from Sprint 1
  - Initial architecture picture
  - Decision of MDM API will be handles as a separate project
  - Decision on interface spec language
- Common Goals:
  - Agreement on the positioning of the MDM API within the project landscape
  - If decided as a separate project: infrastructure available
- EPOS
  - first draft of the MDM API definition
  - First draft of the API test cases
  - Drive decision on API protocol with the architecture committee

- CANOO
  - Definition of a component as a part of the openMDM toolkit which can be subject to
    - Testing
    - Commercial order placement
    - System configuration by a systemn configurator
    - Possibly Deployment units
  - Picture of the runtime communication of components as decision criteria for the architecture committee on the protocol for the openMDM® API

Sprint 3 – WK05-WK07

- Prerequisites:
  - (from now all published in openMDM® web site)
  - API definition draft
  - API Test Cases draft
  - Component definition draft
  - repository of prerequisite documents considered
- Common Goals:
  - Definition of the several aspects of a component (processes, use cases and possibly ressources)
  - complete architecture overview and documentation structure
- EPOS
  - detailed API definition
  - detailed API test cases
- CANOO
  - feature list (high level requirements) / architecture goals consolidated (ready for decision by the architecture committee)
  - component definition refined
    - delivery conditions
    - testing issues
    - development / acceptance criteria
  - architecture documentation draft