POTSDAM INSTITUTE FOR
CLIMATE IMPACT RESEARCH

Data Version Management

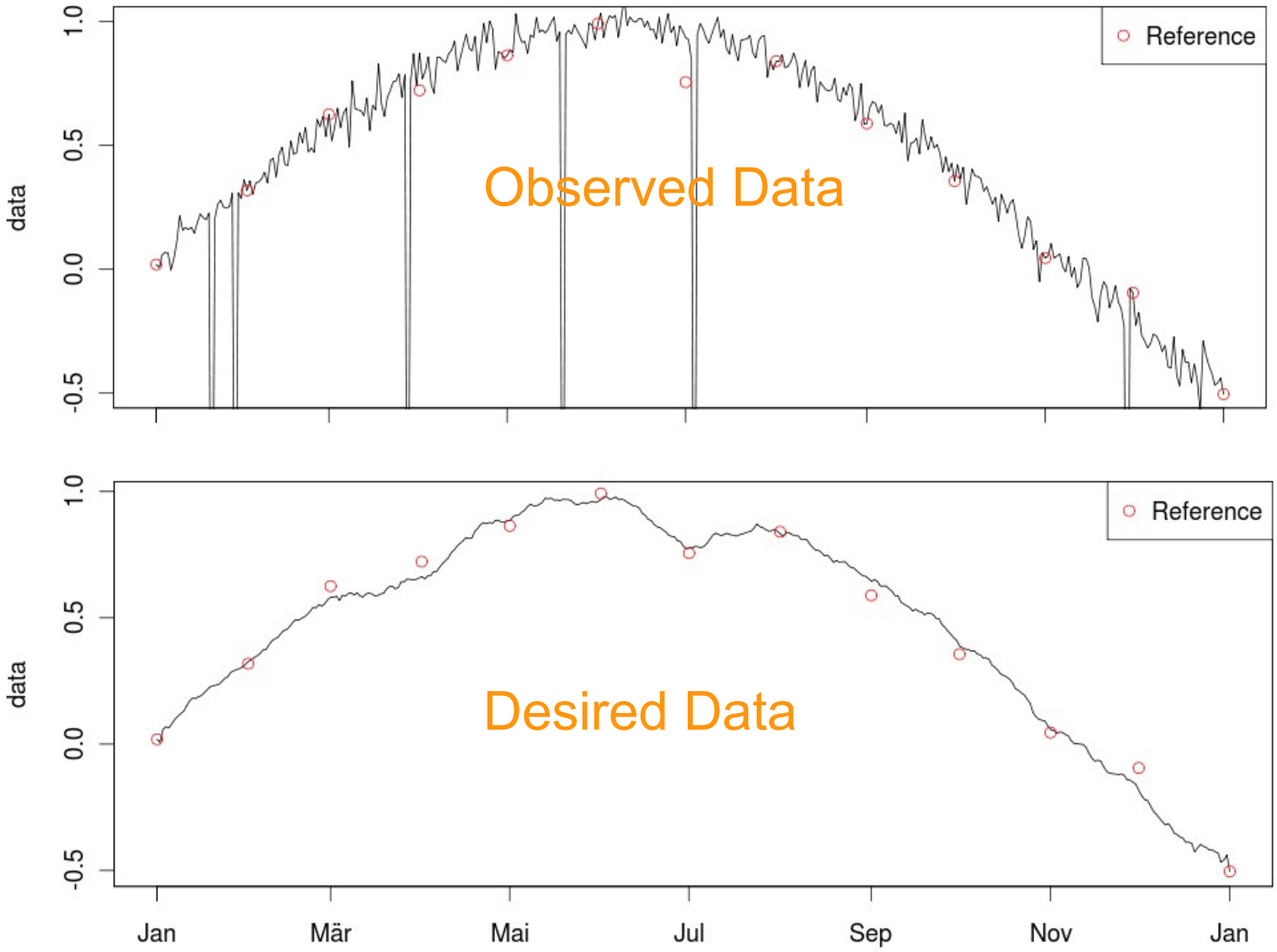
Workshop on data repositories in environmental sciences

Dominik Reusser and Markus Wrobel

March 2011

Leibniz
Gemeinschaft

Reproducibility of data harmonization



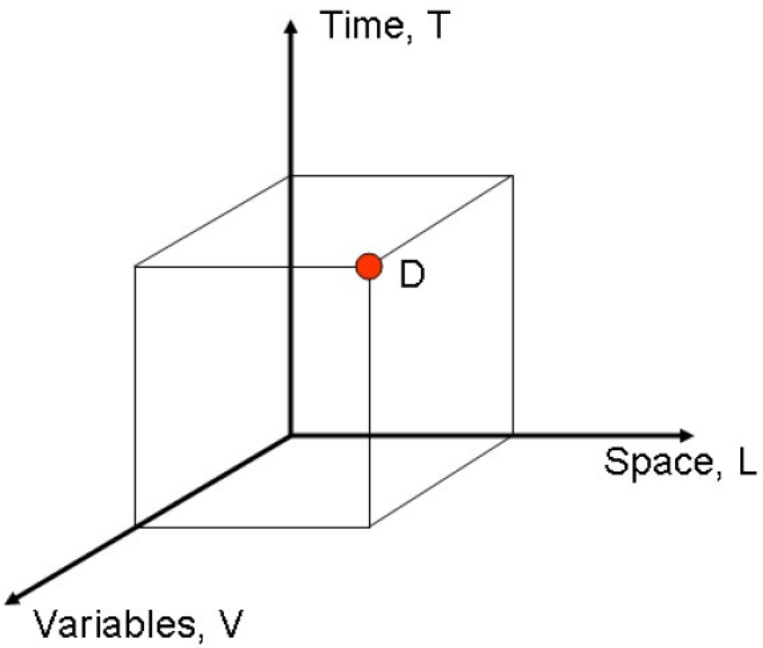
Outline

- **Existing approach (in hydrology)**
- **Idea to version management**
- **Example session**
- **Implementation details**

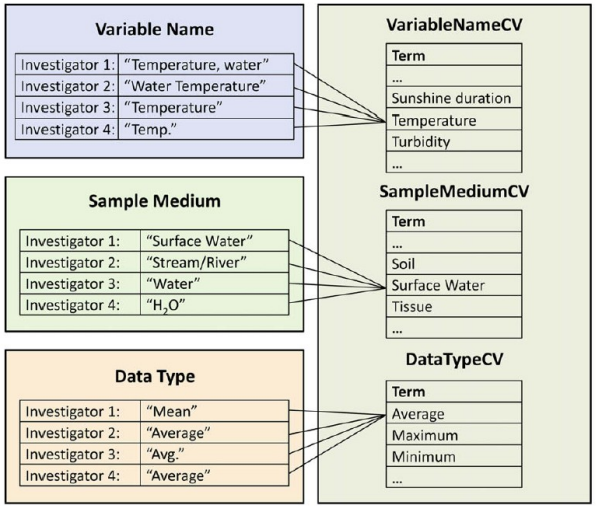
Existing approach

Observations Data Model Hydrology

Fundamental triple for data characterization



Controlled Vocabulary

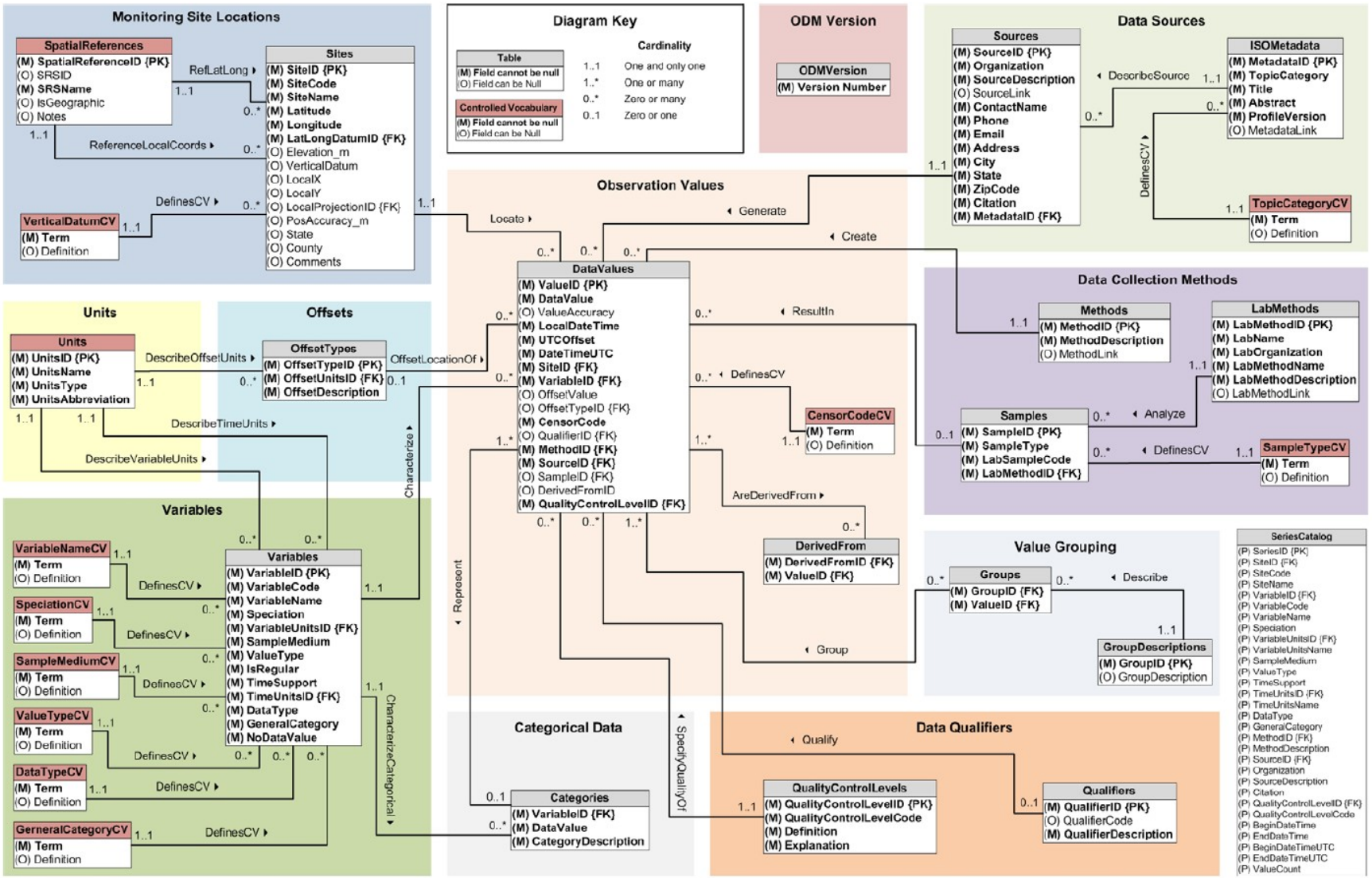


WaterOneFlow Web Service

Horsburgh, et al., 2009 Environmental Modelling & Software

Horsburgh et al., 2008 Water Resources Reserach

Observations Data Model



Horsburgh, et al., 2009 Environmental Modelling & Software
Horsburgh et al., 2008 Water Resources Reserach

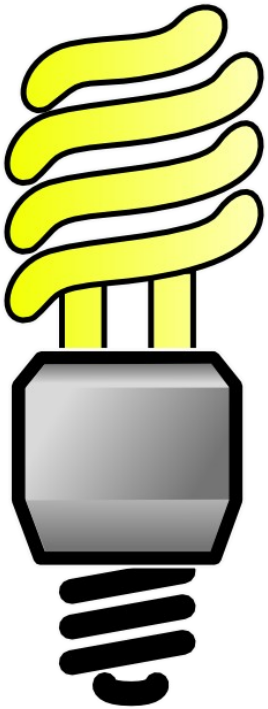
Quality control levels

- **Kind of versioning system**
  - → **Not well integrated into data model**
  - → **Copying entire data series**
  - → **Changes not trackable**

Extension

Adding Version Management: The Idea

- **User Interface:**
  add, update and remove data
  query data
- **Handling of versions similar to version management systems (e.g. svn):**
  - **Make edits to local working copy**
  - **Incremental storage of changes upon commit**
  - **Retrieve old version by identification number or date**
- **Assumption: No versions for meta data**



PIK

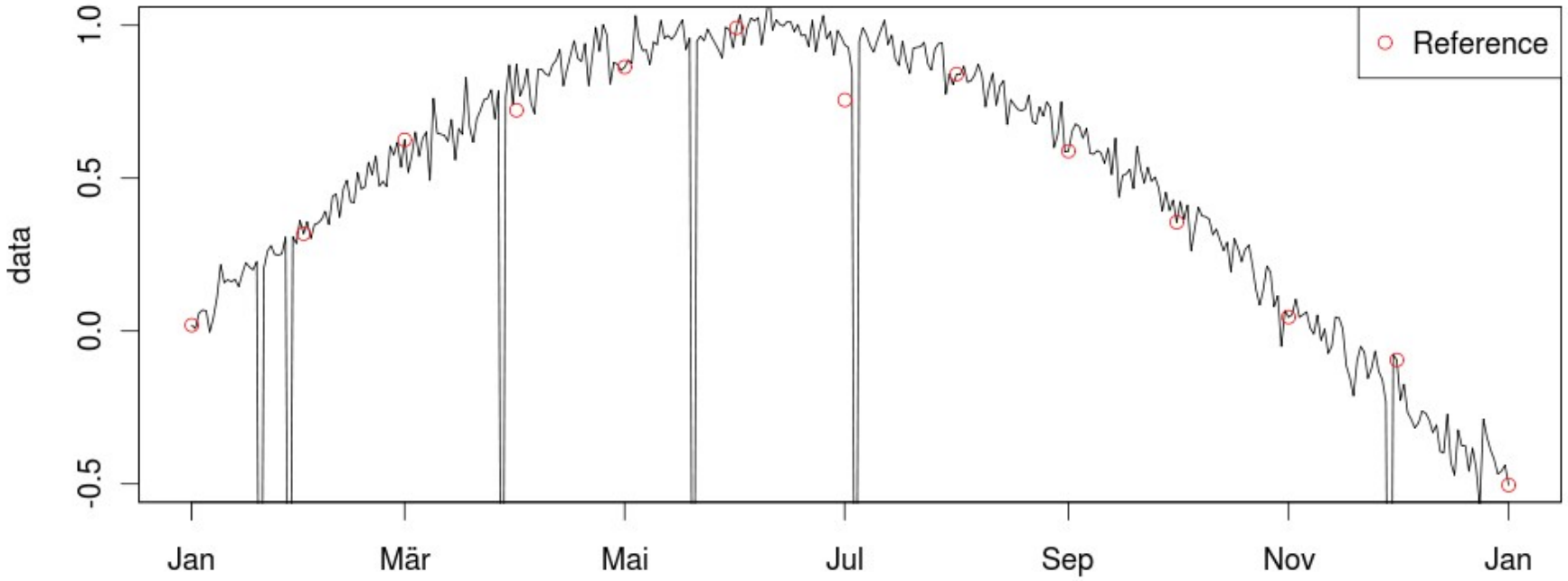
Changes to the data base

- **Copy table structure of** DataValues**:**
  - DataValuesArchive **containing outdated values only**
  - **additional** VersionID **field**
- **Note:** DataValues **remains unaltered compared to missing data version management**
- **New** Versions **table:**
  storing editing time, comment and responsible person

Example Session

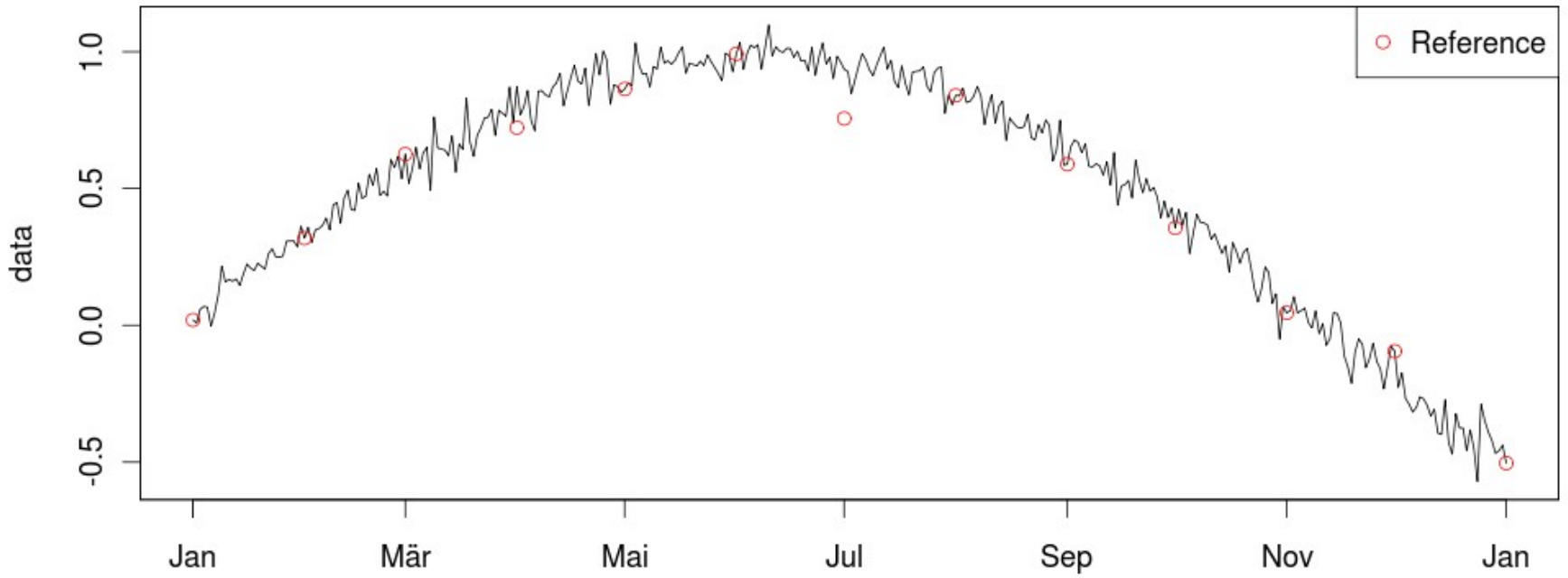
Example session (pseudo code)

addDataValues(sensorData)

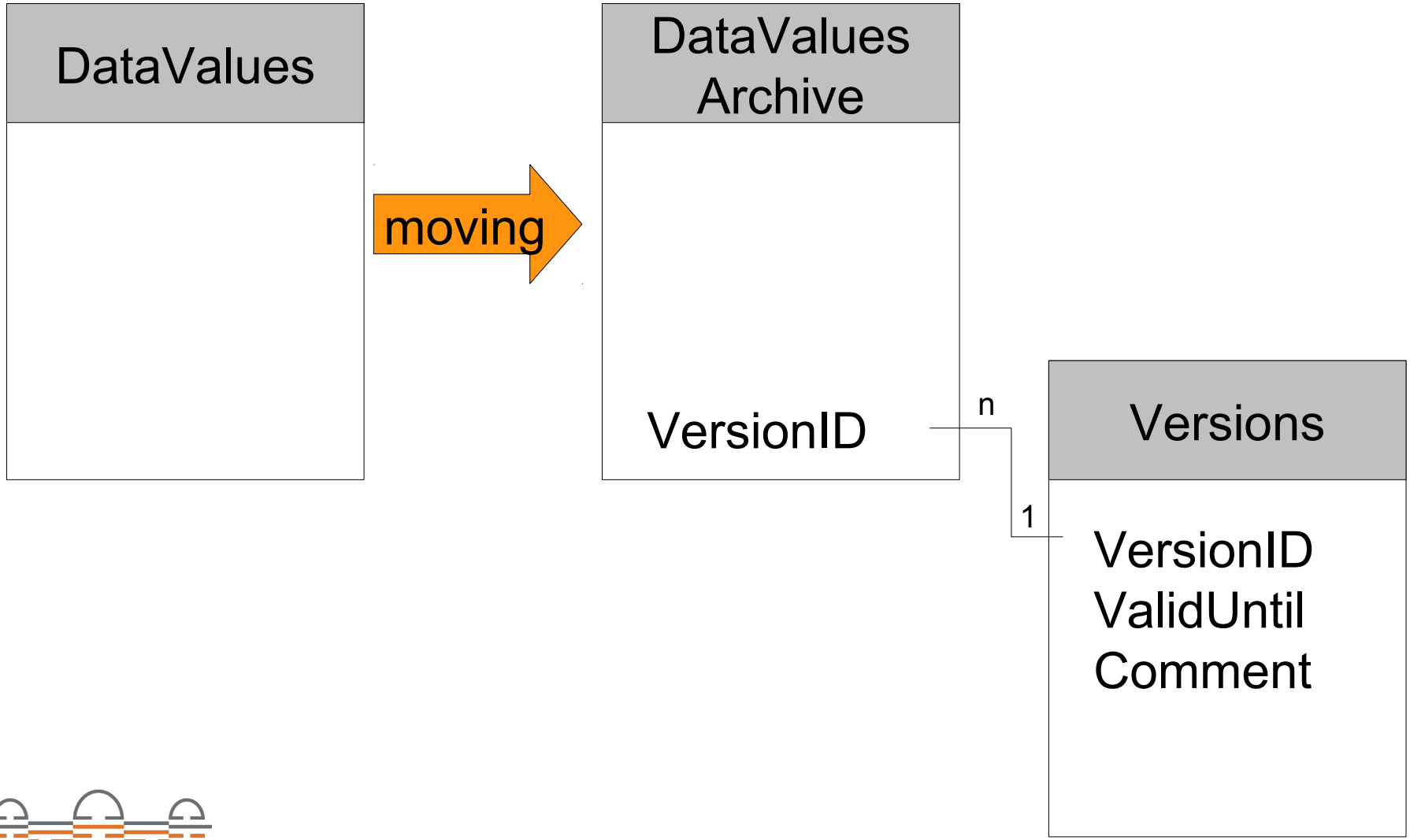


Example session (pseudo code)

deleteDataValues(DataValue=-2,
reason="Removing NA-values (Value of -2)")

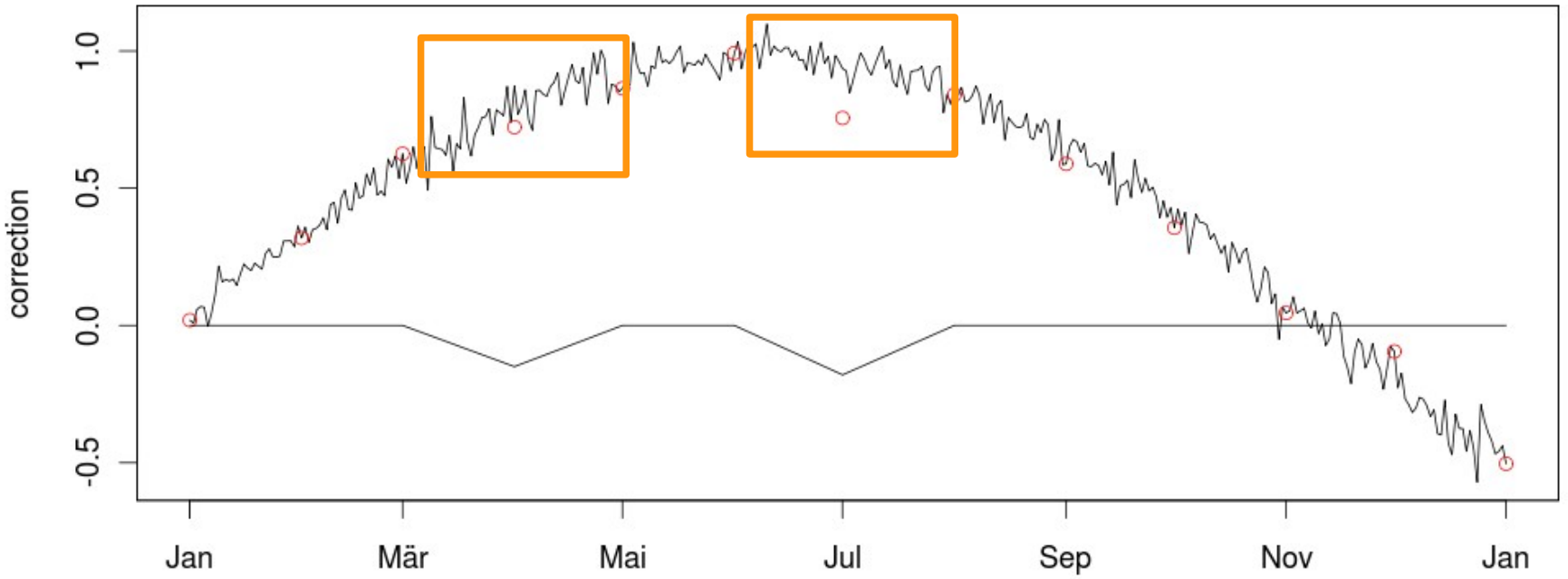


Deleted records



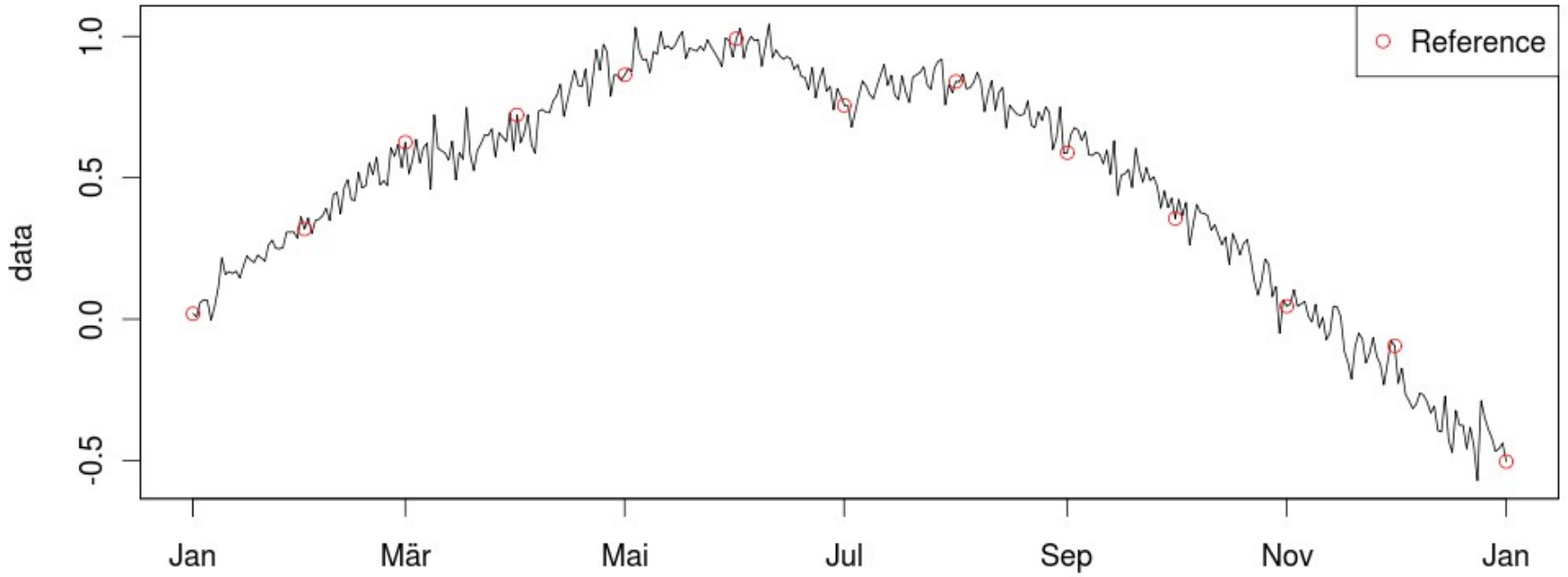
Example session (pseudo code)

Calculating the correction



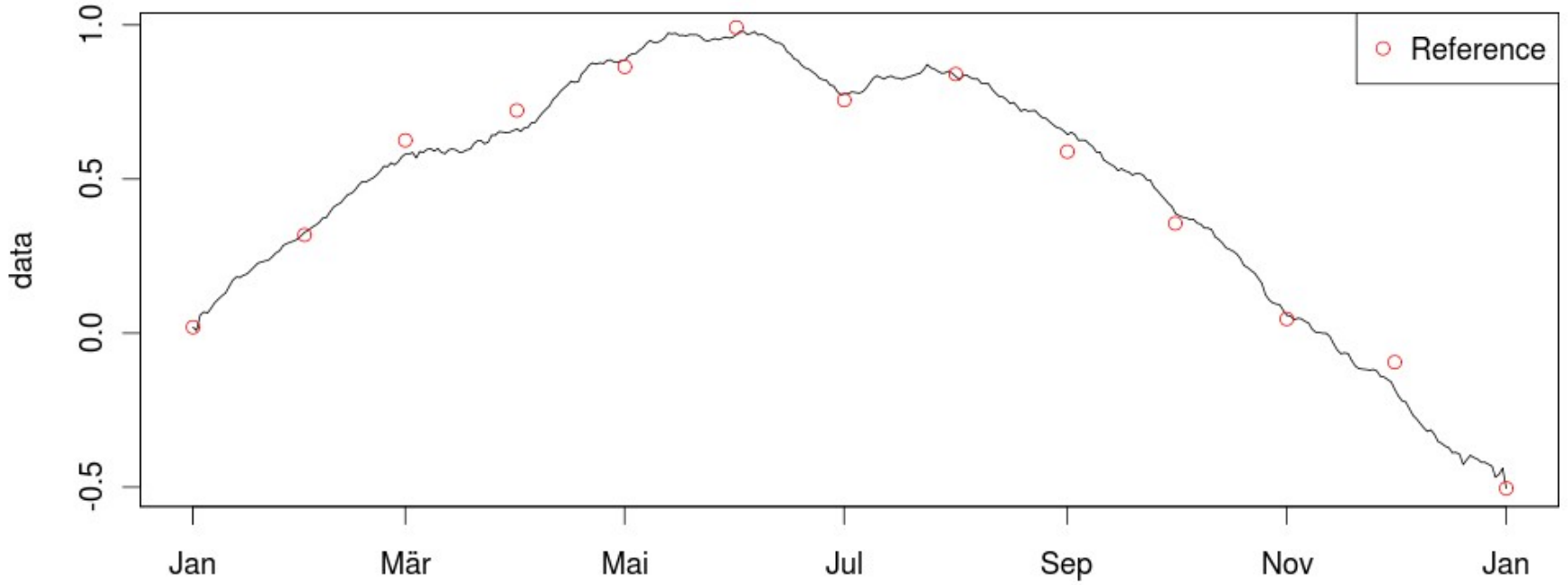
Example session (pseudo code)

updateDataValues(DataValue+Correction,
reason="Correcting sensor drift")



Example session (pseudo code)

updateDataValues(rollmean(DataValue),
reason="Applying rolling mean")



Example session (pseudo code)

GetDataVersions()

VersionID	ValidUntil	VersionComment
1	2011-02-15 11:50:21	
2	2011-02-25 11:59:50	Removing NA-values (Value of -2)
3	2011-02-25 12:05:35	Correcting sensor drift
4	NA	Applying rolling mean

Implementation Details

Three-layer architecture

User interface + Program Logic
e.g. addDataValues, getDataValues,
updateDataValues, deleteDataVaules

Database access interface
e.g. IaddDataValues, IgetDataValues,
IupdateDataValues, IdeleteDataValues

Database implementation
e.g. Observation Data Model 1.1 on MySQL

Changes to the user interface

- **Add „change comment“ to** updateDataValues **and** deleteDataValues
- **Add version identification (ID or Date) to** getDataValues
- **Add** getDataVersions
- **Changes to the internal logic:**
  - **Archive data before making changes**
  - **Additional logic to obtain previous versions**

PIK

Changes to the data base access interface

- **Add version number management:** IgetCurrentDataVersions, IaddDataVersion, IgetDataVersions
- **Add handling of versioned data:** IarchiveDataValues, IgetOldDataValues
- **Simple standard implementation (do nothing) for data bases without version management**

PIK

Conclusions

- **Proposal for Data Version Management**
- **Based on Observation Data Model (and similar)**
- **Following ideas from text version management such as svn**
- **Classical 3-layer architecture**
- **Small modifications to each layer sufficient**

Prototype is being developed in R and is available upon request

Thank you for your attention

PIK

Work of Dominik Reusser financed by BMBF (OPAQUE and PROGRESS)