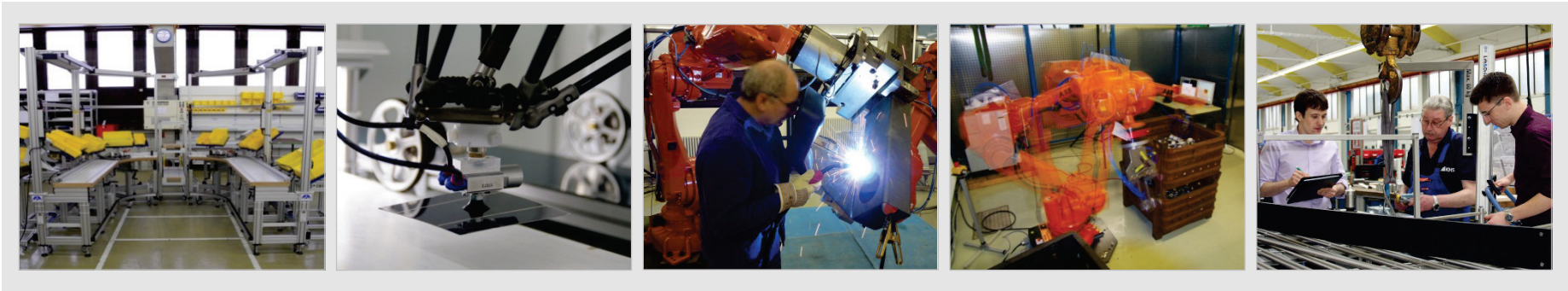
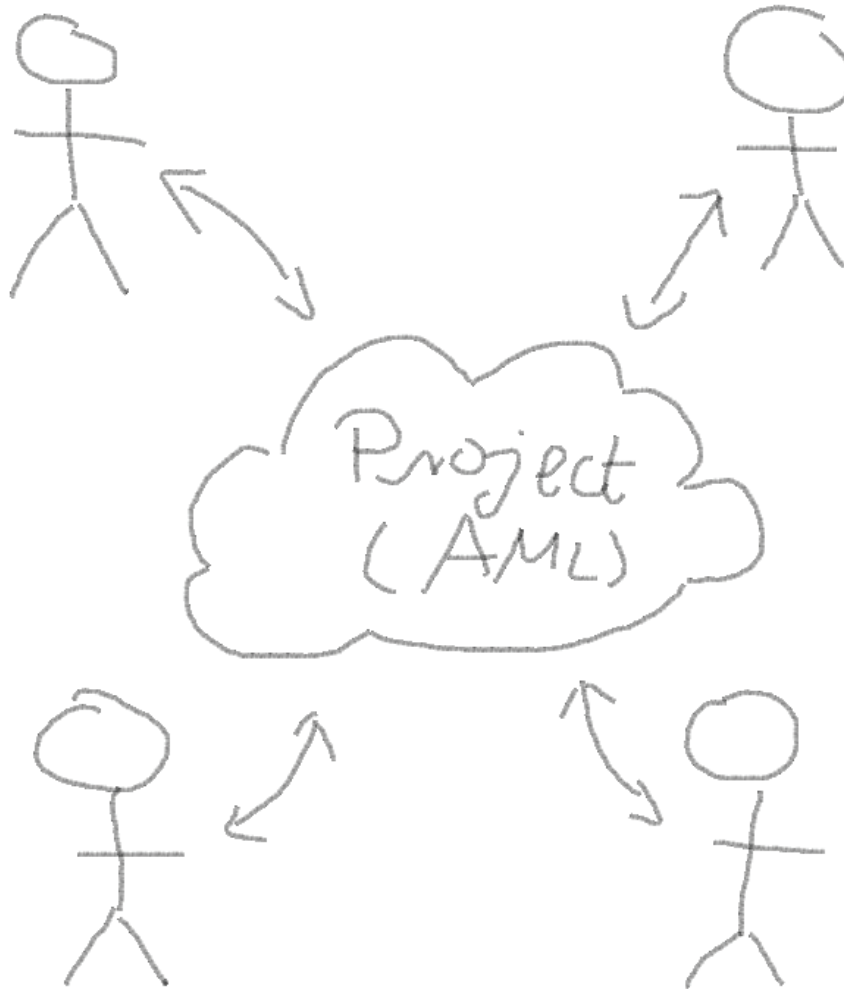


# Virtual Commissioning in a heterogeneous Software Environment

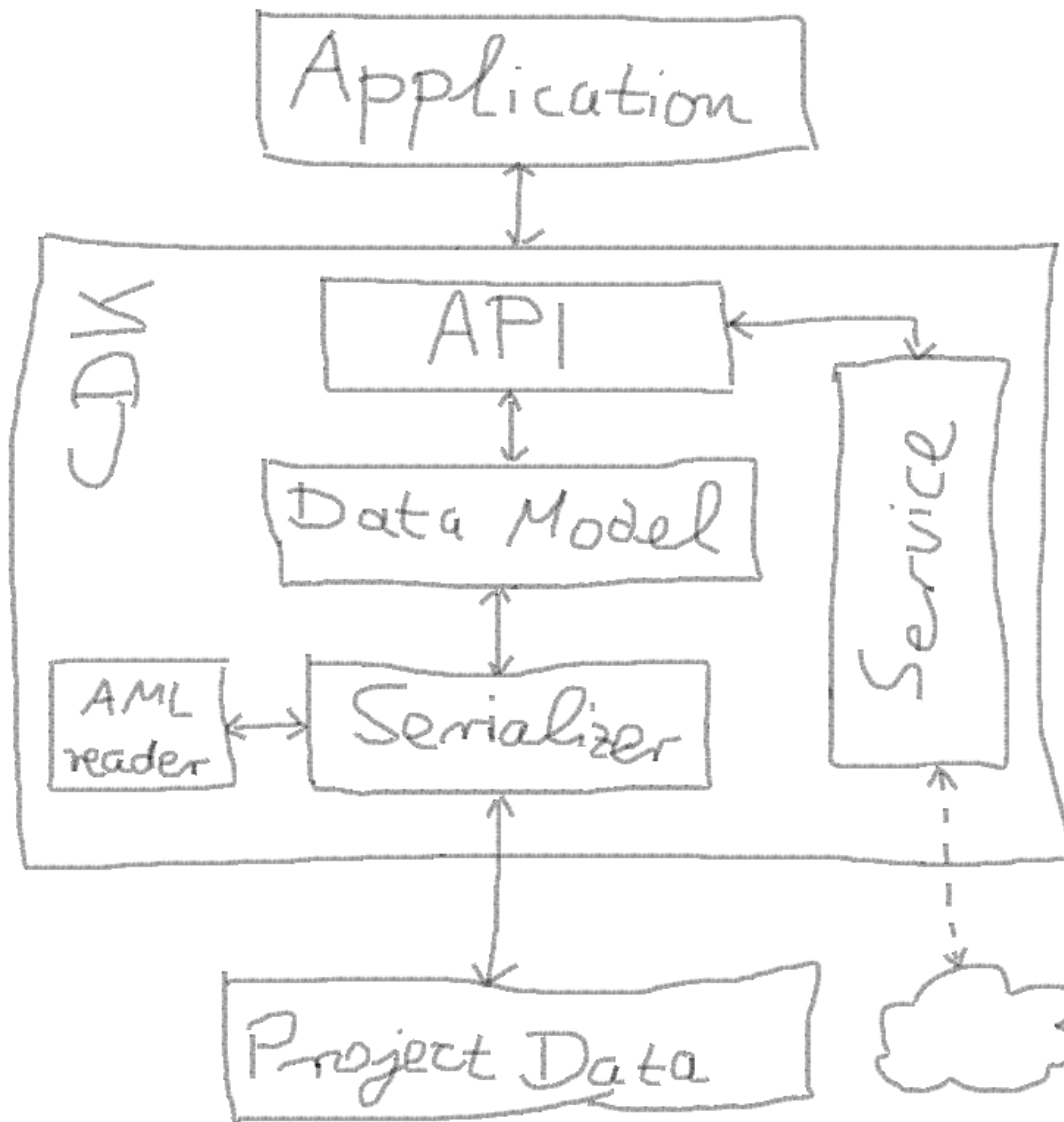
Exchange data, merge and create versions using the conexing development kit



Matthias Bartelt  
7. Oktober 2014



- Experts want (must) work together
- Each expert uses their own software and wants to continue using this software (since he knows how to use it)
- Using a standard data format for exchange will allow them to do so, but any involved software must be extended in order to import/export the data format



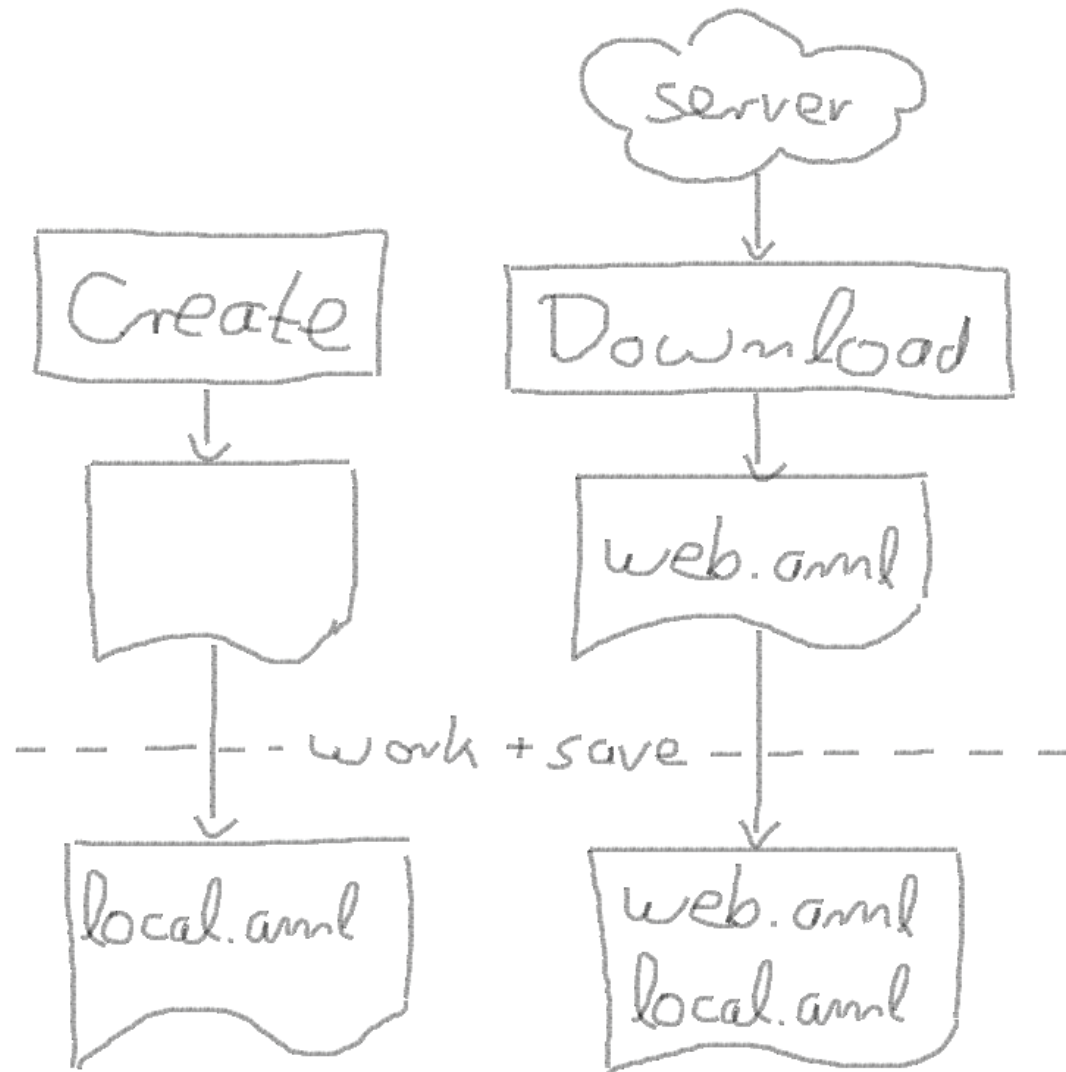
- Using the conexing development kit (CDK) reduces programming effort
- Automatically prevent loss of data within the CDK (especially of data which cannot be handled by the particular software)
- No knowledge of the exchange format is required
- Management of different versions

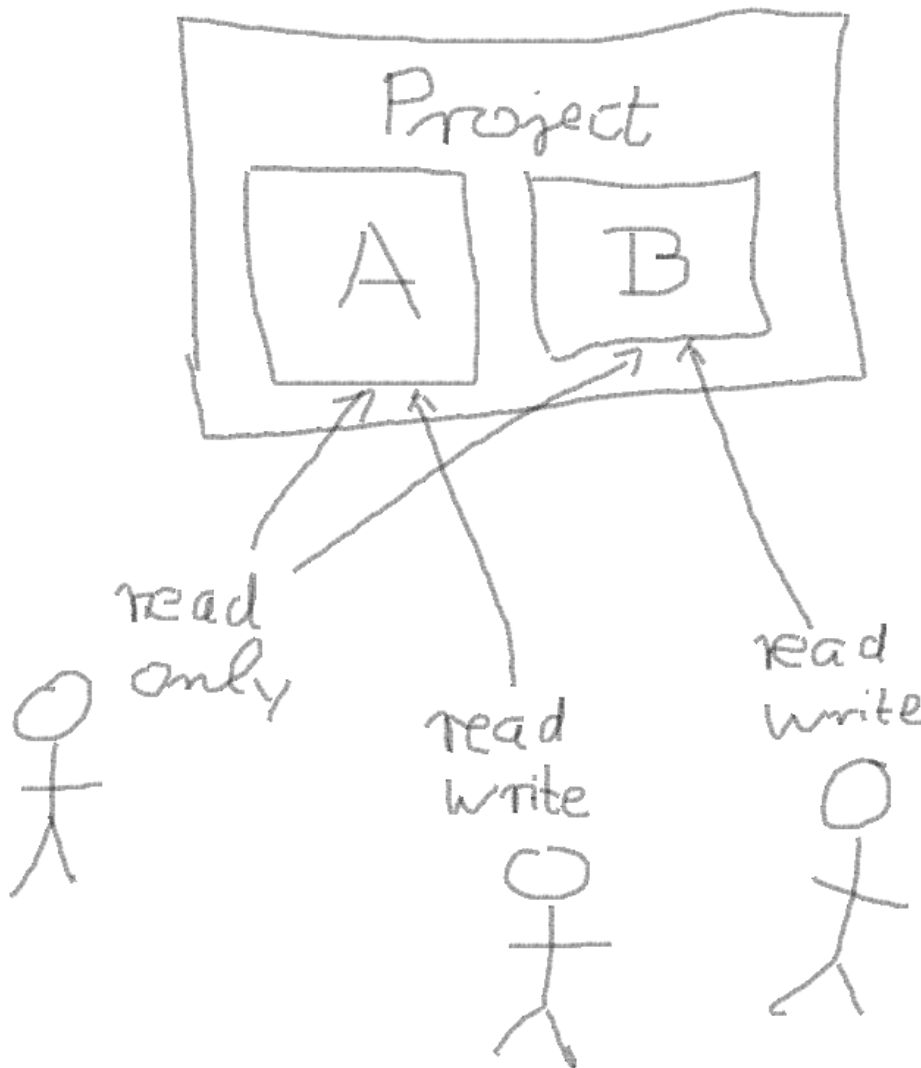


- Handle (partial) exchange of data, taking into account that user may have different access rights
- Data can be stored in a standard format, but there might be different versions of the project
- What if more than one person is working on the same part?

## Get and work with a project (or components)

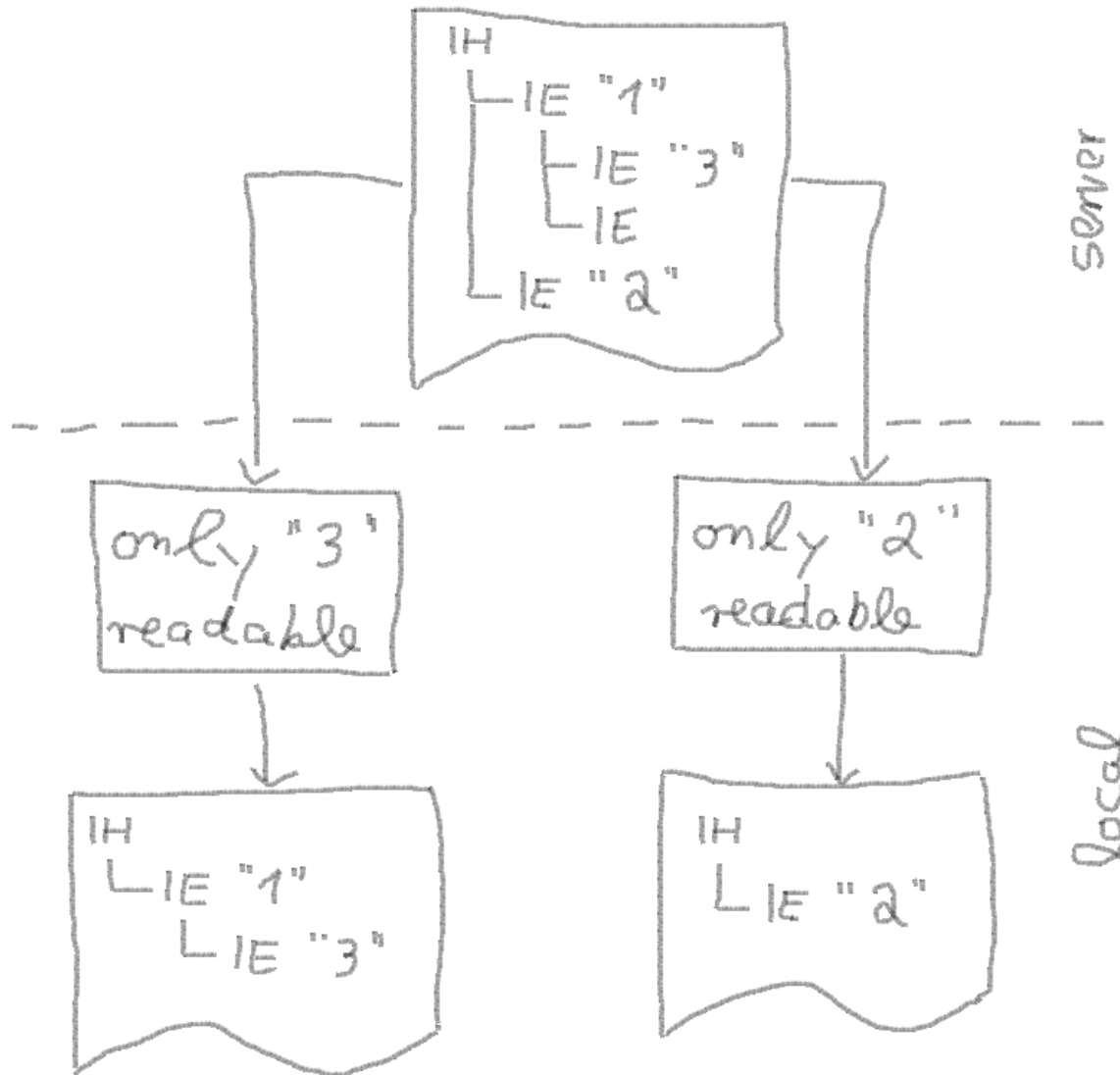
- Webservice provides project data (as well as available components)
- Browse project or search components using SOAP or Web-UI
- Download project; alternatively create a new one
- Saving the edited project results in a (new) local version



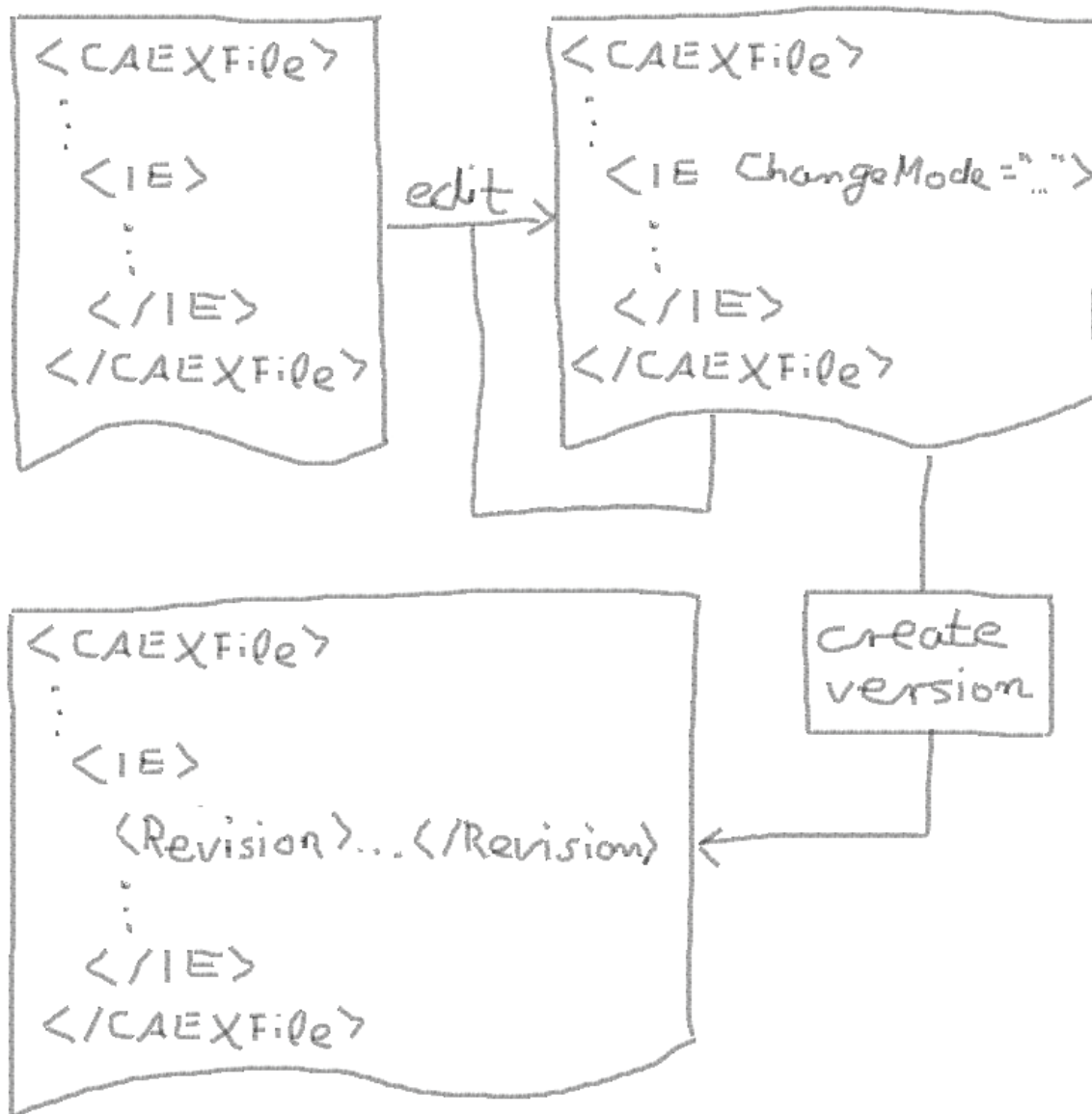


- Right management (on the web-platform)
- Only parts with appropriate access rights can be received
- User has complete access to his local copy (so he can modify parts he is not allowed to commit)
- Webserver will handle re-integration of the particular part of the complete project

## Same project with different access rights

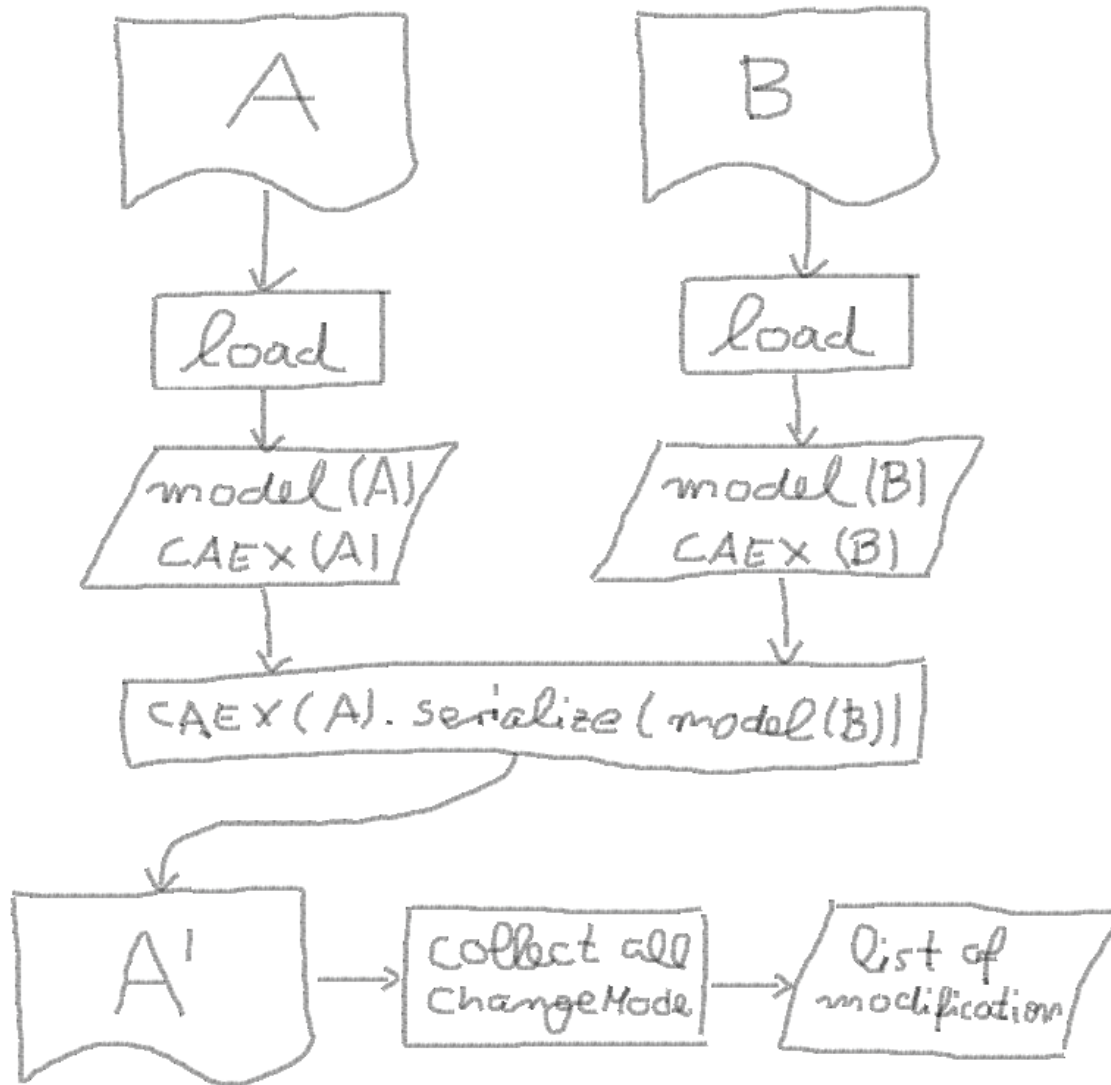


- Server creates individual copy of project according to the user's access rights
- Object without read access are not included
- If an accessible object is located below an inaccessible object, the inaccessible object is commit as plain structural information



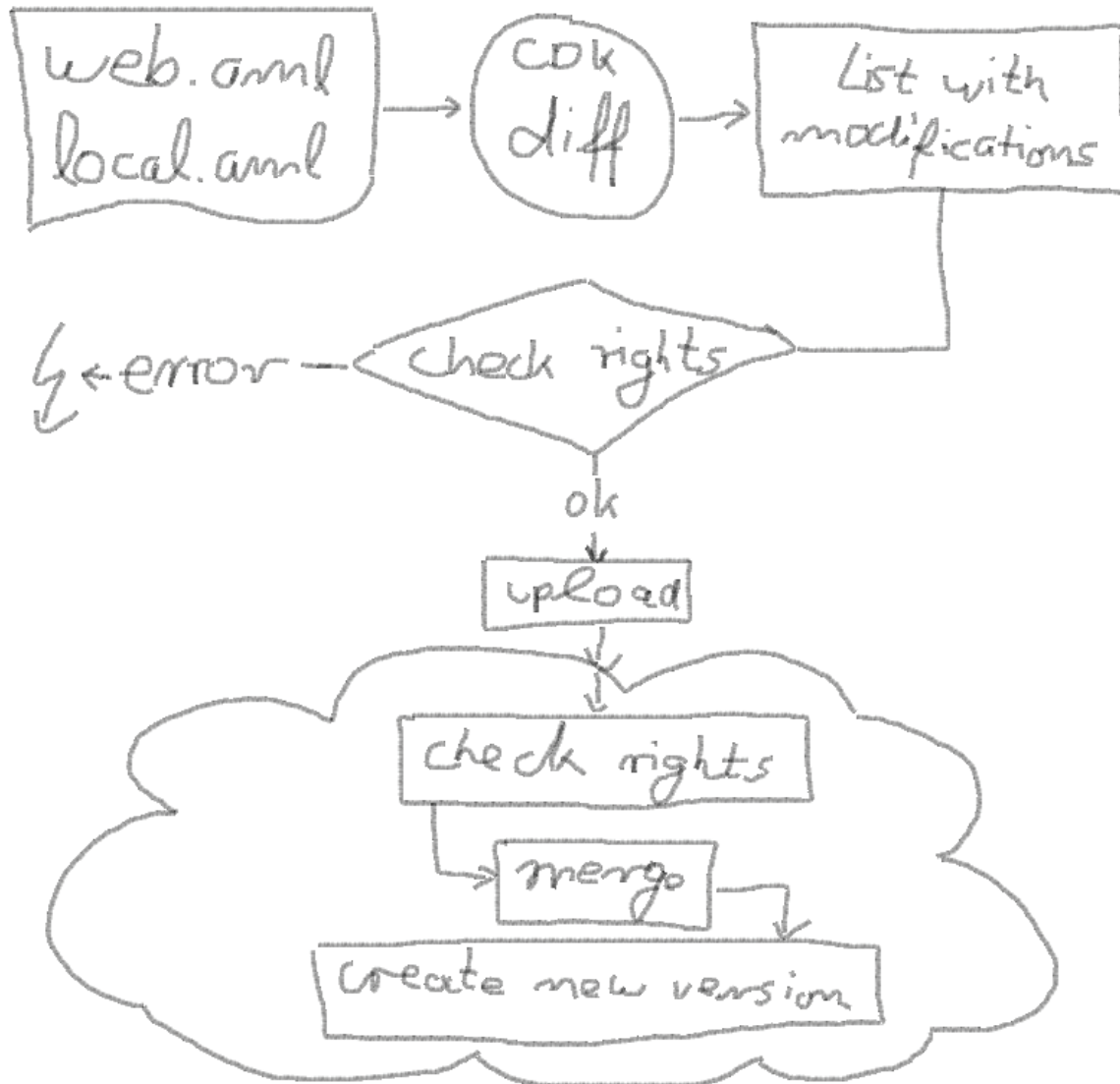
- Using the CDK to load, modify and save a project, any change will become marked automatically by the serializer within the corresponding AML document
- Create a new version by a single method call; appropriate references to the former version will be added automatically





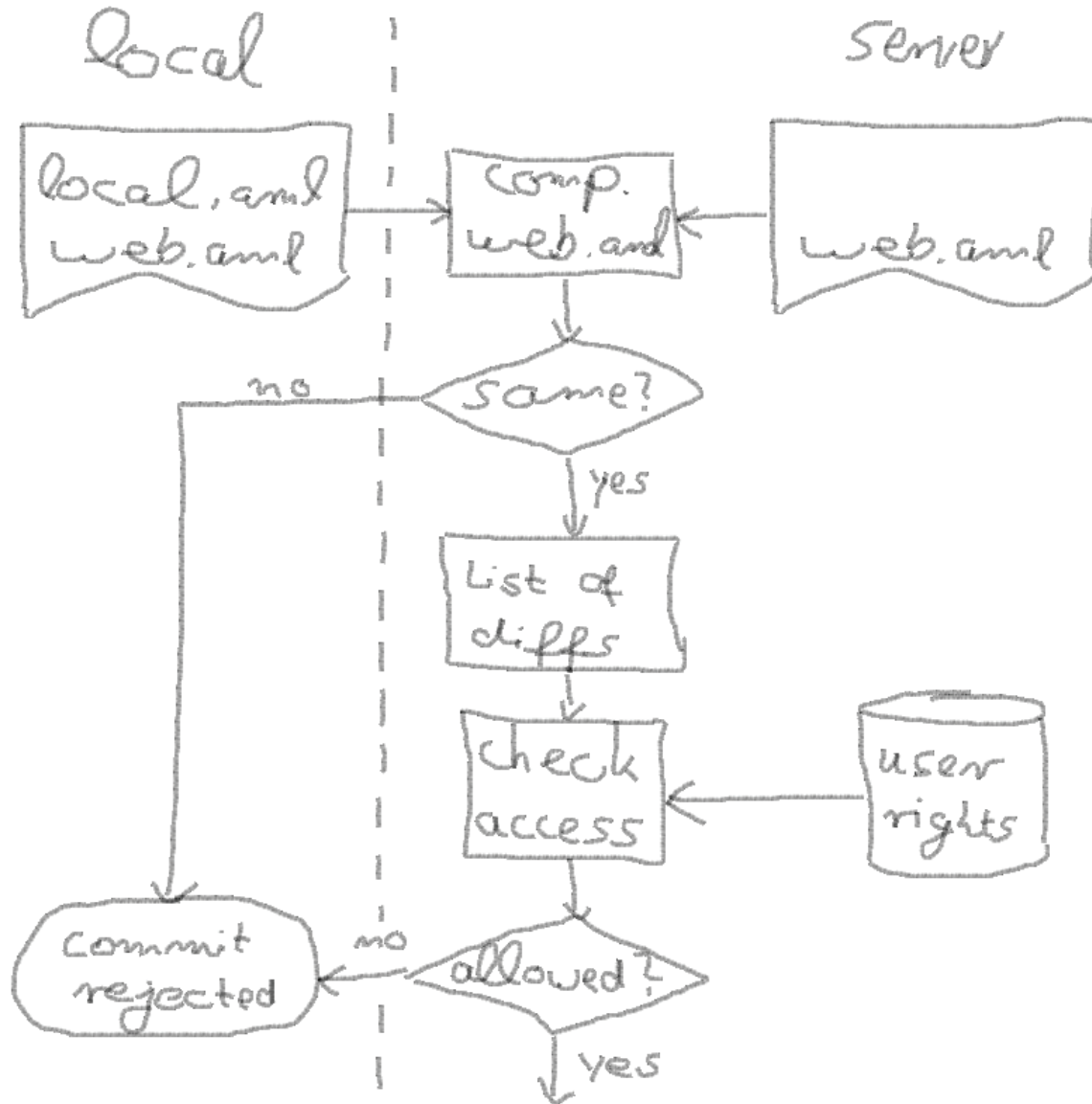
- List of modifications of one file can be obtained just by searching for ChangeMode attributes
- Modifications of a file compared to another can be by loading both files, ignoring all changes and saving the newer file the original file;

## Commit a (changed) project



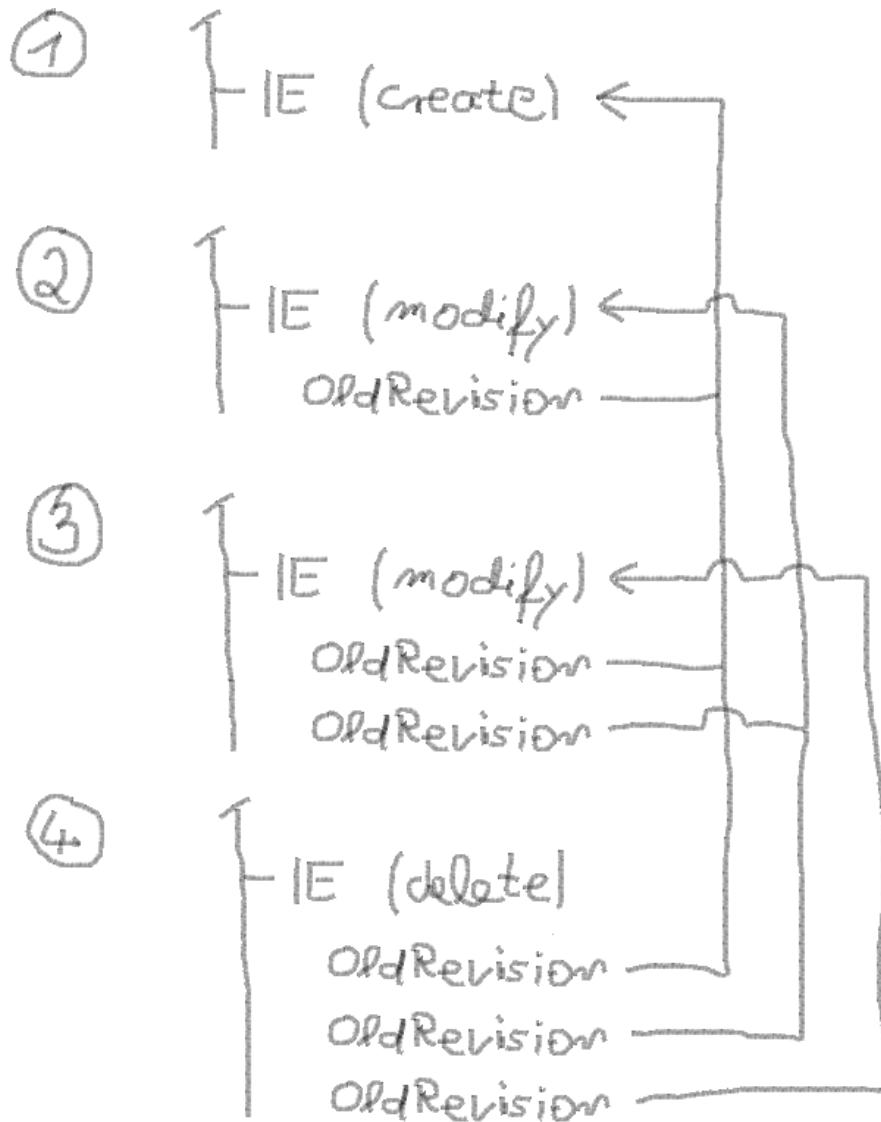
- Create list with (local) changes
- Check if the user has all rights to commit all changes
- Check if some other user has committed changes the user doesn't have (update is required before commit)
- Upload to server (all checks are made again to avoid security issues)
- Merge changes to new version

## Checking rights to commit a version

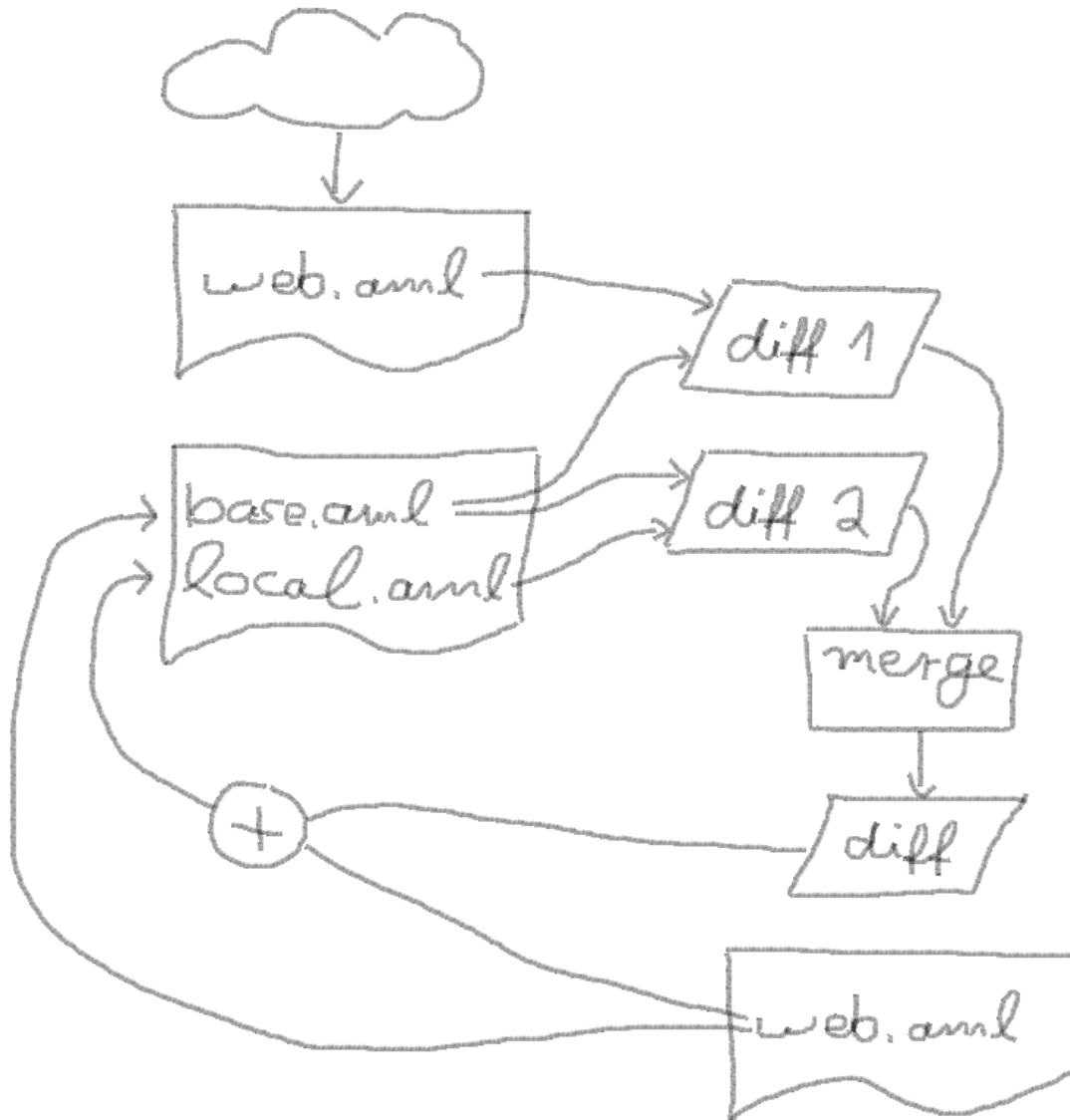


- Is the local version based on the same version as the (current) version on the server?
- Does the user have necessary access rights (i.e. writing) on all modified parts?
- Read and write access can be granted on
  - Components
  - Properties
  - Type of components
  - Role of components

## Merging (on the server)

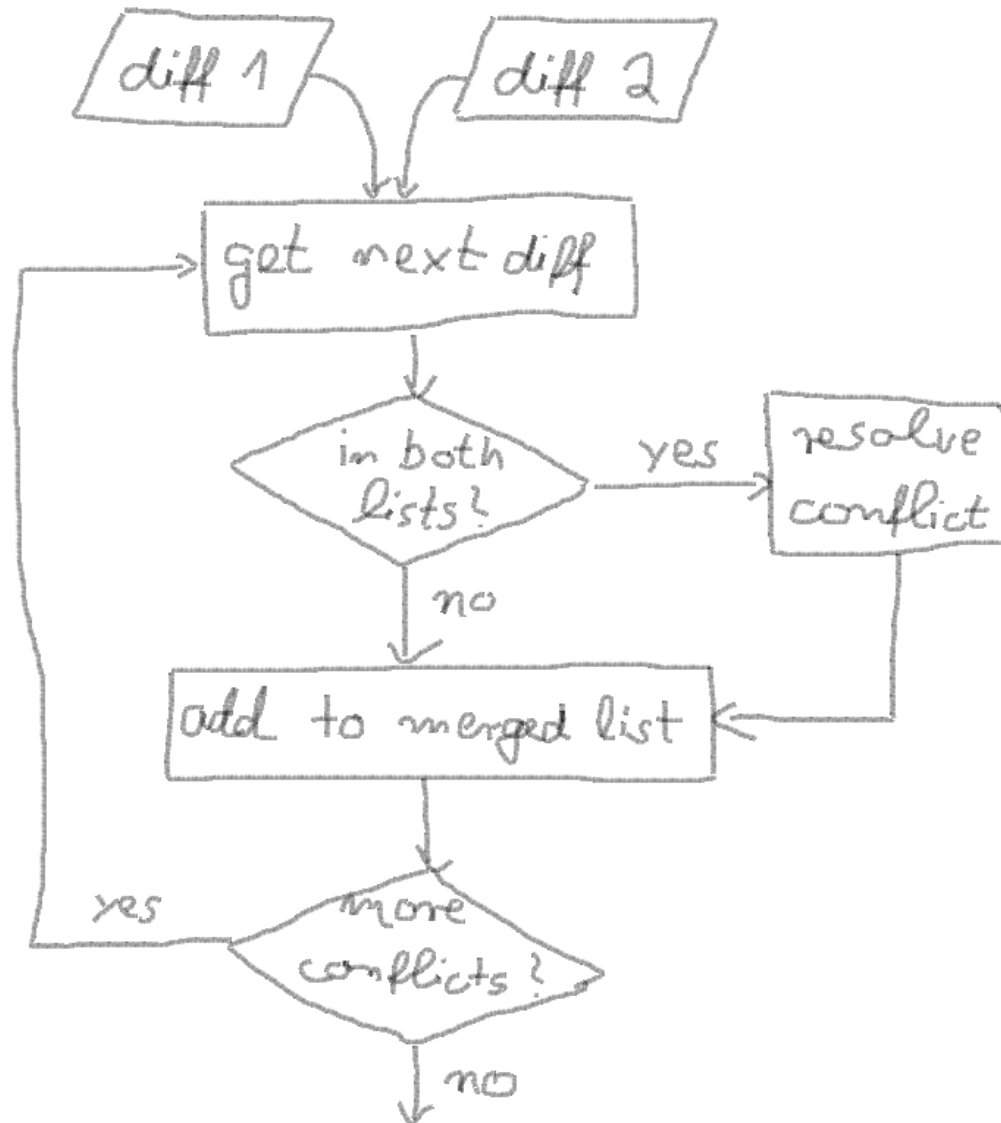


- Server gets a list with modifications
- Server rejects modification if commit check fails
- Server loads current version and integrates any modification while creating a corresponding revision tag
- Current version includes references to any version
- Server may keep all versions



- Update may be required before commit
- Create two lists with modifications: local and current web version compared to base
- Merge both lists to one list of modifications
- Merge this single list with web version to new local version
- Replace local base with web version

- Entries occurring in both lists yield to conflicts (since those objects are changed by (at least) two persons)
- Conflicts cannot be solved automatically; someone has to decide which modification should be used
- Resolve conflicts by deciding to use one of the two versions (of the particular object)





The development kit may be used to

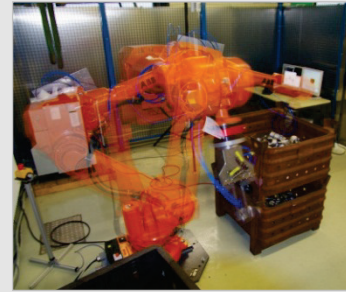
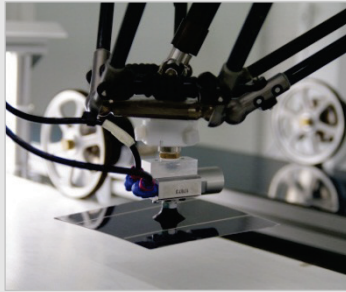
- read/write data
- exchange data (together with the web-platform)
- merge data

Future work

- Modifications with denied permission must be handled somehow



Thank you for your attention!



TU Dortmund University  
Institute of Production Systems  
Professorship in Industrial Robotics  
and Production Automation

[www.IPS.DO](http://www.IPS.DO)

Matthias Bartelt  
<[matthias.bartelt@ips.tu-dortmund.de](mailto:matthias.bartelt@ips.tu-dortmund.de)>

The research and development project  
“conexing” is funded by the German Federal  
Ministry of Education and Research (BMBF)  
within the Framework Concept “Research for  
Tomorrow’s Production” and managed by the  
Project Management Agency Karlsruhe  
(PTKA). The author is responsible for the  
contents of this publication.

[www.conexing.de](http://www.conexing.de)