Software Tools and Data Formats for Data Exchange in Airplane Predesign

Markus Litz, Holger Cornelsen, Hans-Peter Kersken

Simulation and Software Technology

German Aerospace Center (DLR)

PDE 2008

Noordwijk, The Netherlands

March 27th, 2008

Overview

- Motivation and Background
- Common Data Format
- Software Tools
- Integration Framework

Motivation and Background

- The predesign of new airplane configurations involves many different technical disciplines
- Goal: Find an optimal design
- Strong dependencies exist between the disciplines
  - A combination of discipline-local optima does not lead to a global optimum
  - Necessary: global optimization process
    - ➔ Look at the overall system
- Therefore: Cooperation between the individual technical disciplines is essential

Situation at DLR

- Many aerospace institutes, each one specialized on its own technical discipline
- Simulation software is institute-specific, proprietary I/O formats
- Interdisciplinary cooperation in some cross-institutional projects

But:

- Ad-hoc definition of interfaces and data formats in each project
- No common data format for all application codes
- No automated process chains
  - Applications used manually and separately from each other

Linking of Discipline-Specific Design Tools

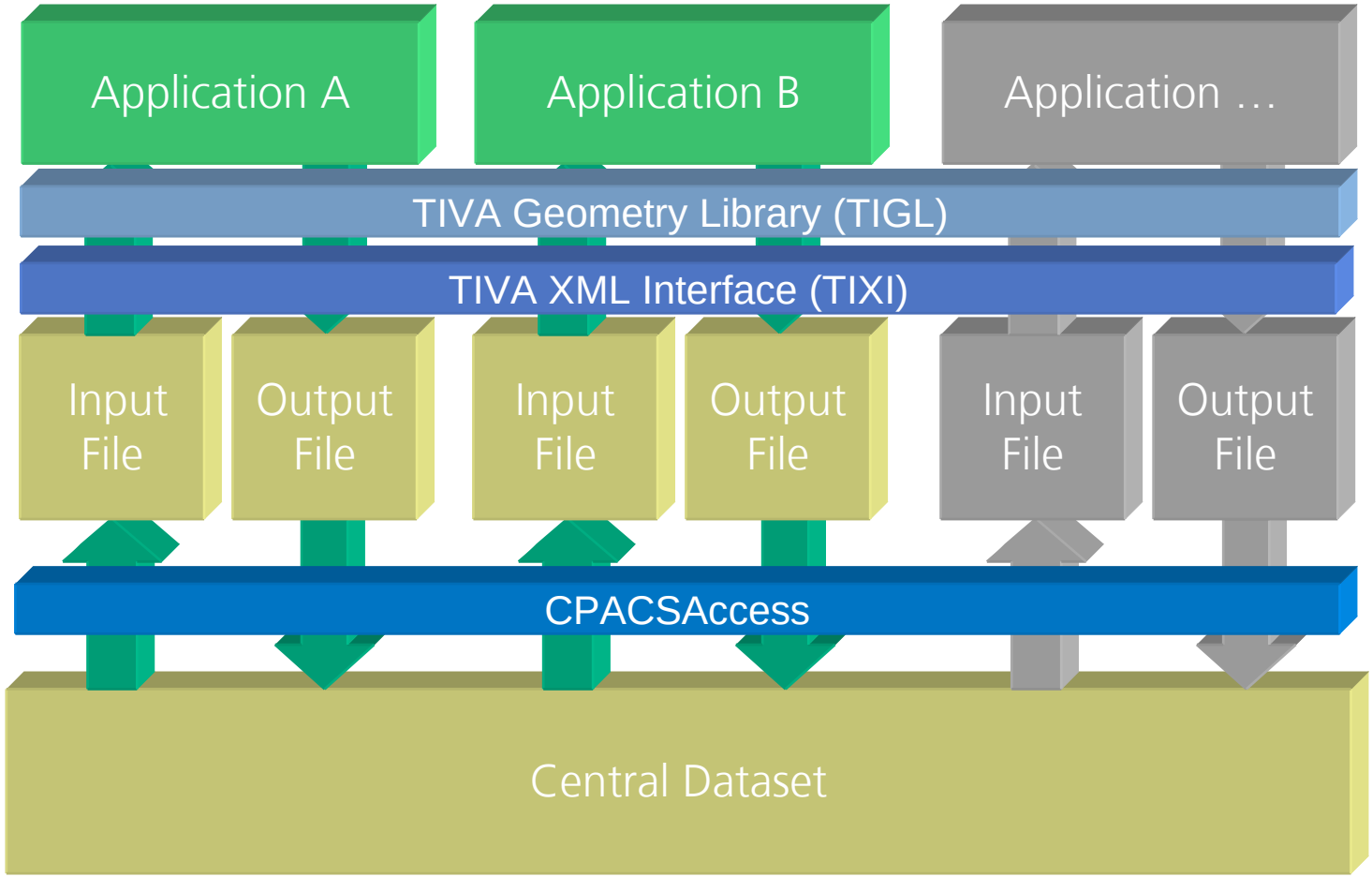
- Goal

  A DLR-wide system to enable the multi-disciplinary design and analysis of airplane configurations in the predesign phase.

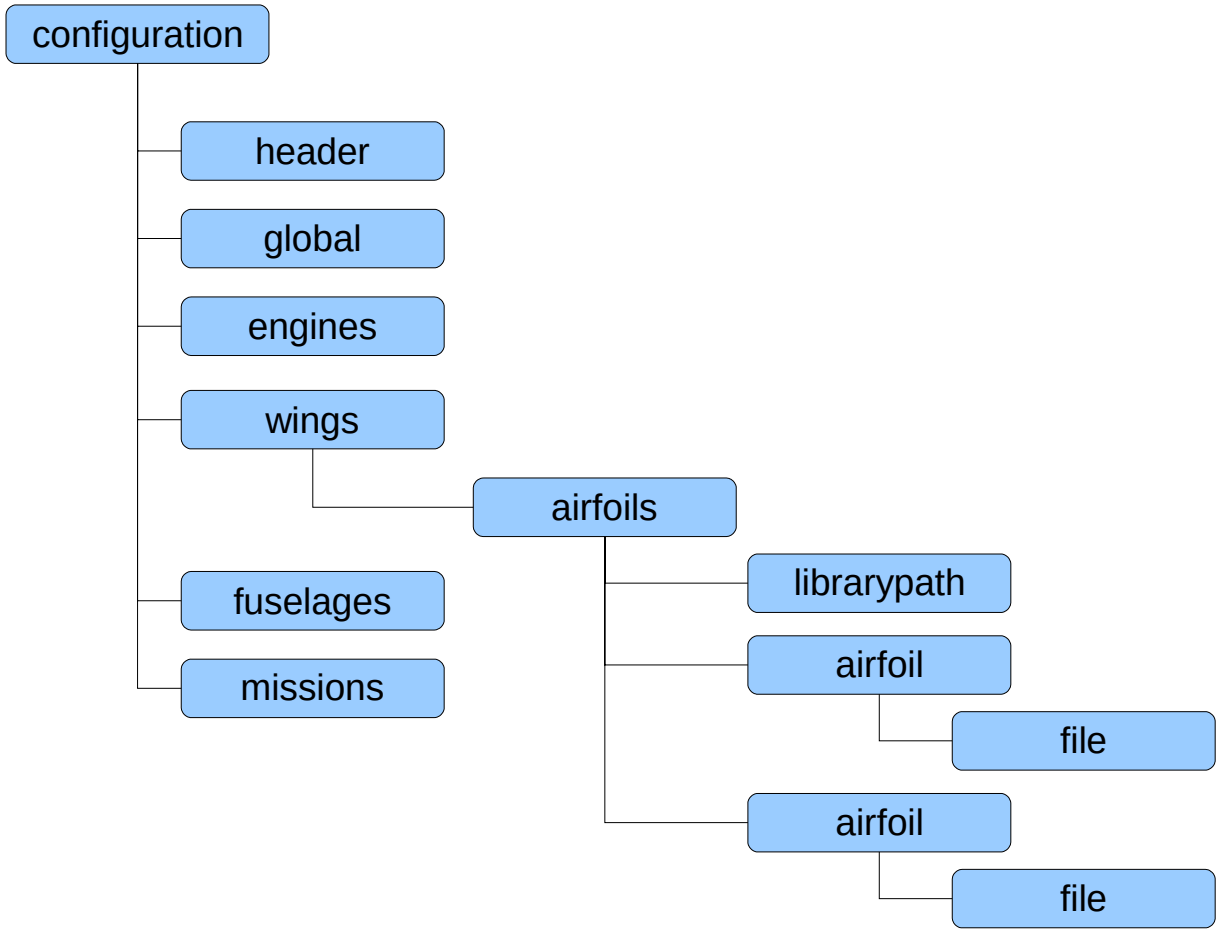
- Under development at DLR in the following aeronautics projects:
  - TIVA I/II – Technology integration for the virtual aircraft
  - UCAV 2010 – Unmanned combat air vehicle
  - EVITA – Evaluation of innovative turbine engines

- Similar requirements in DLR space projects

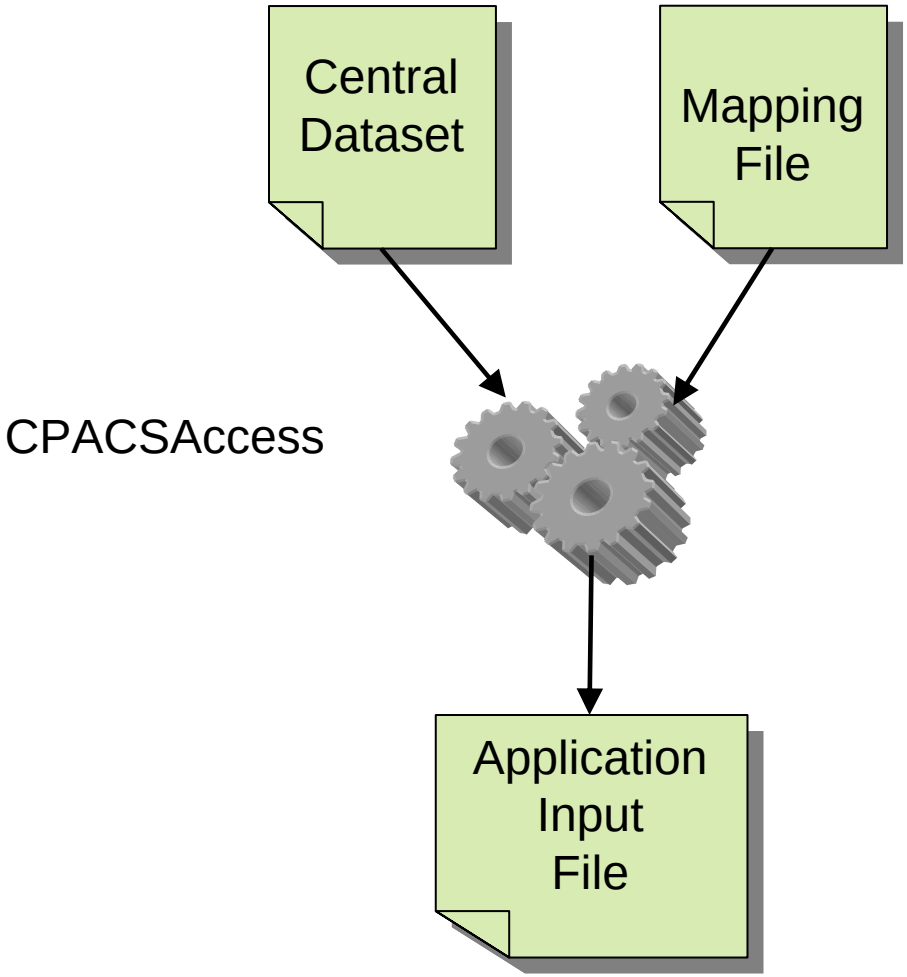
System Overview



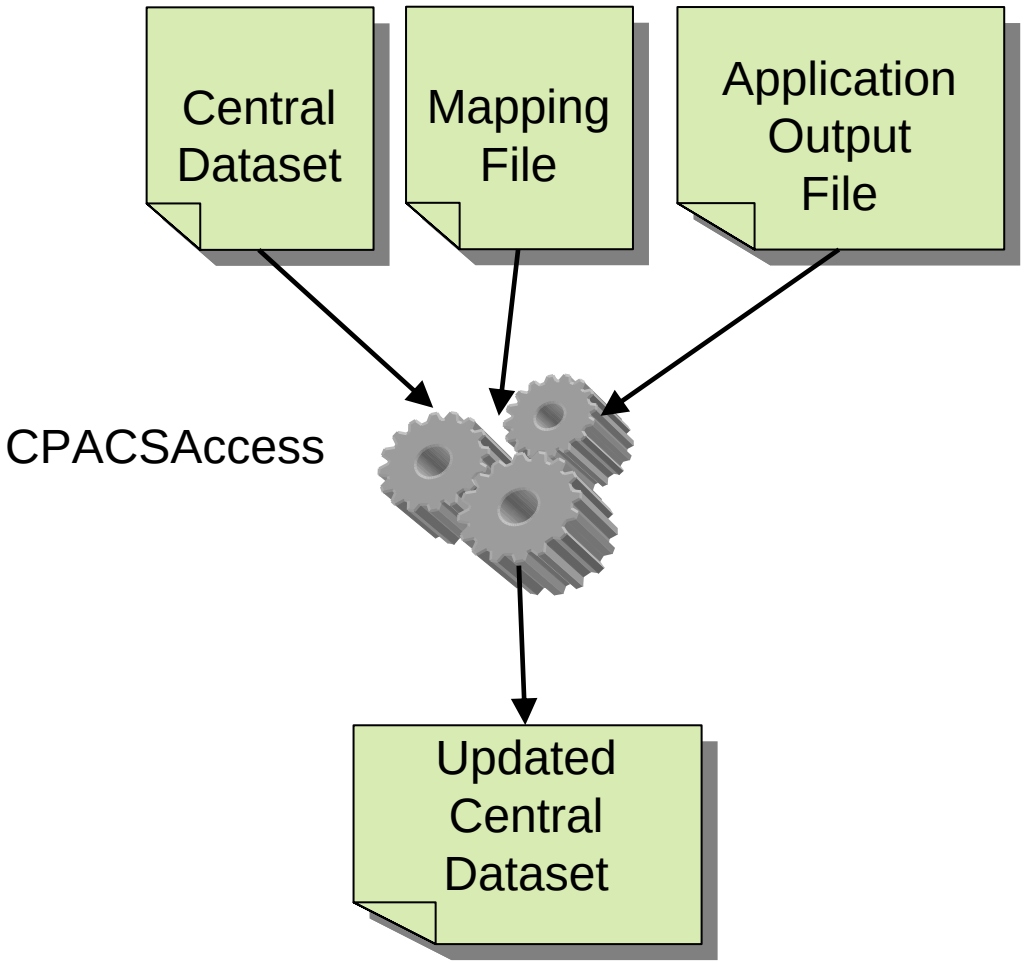
Structure of the Central Dataset



Data Export from the Central Dataset



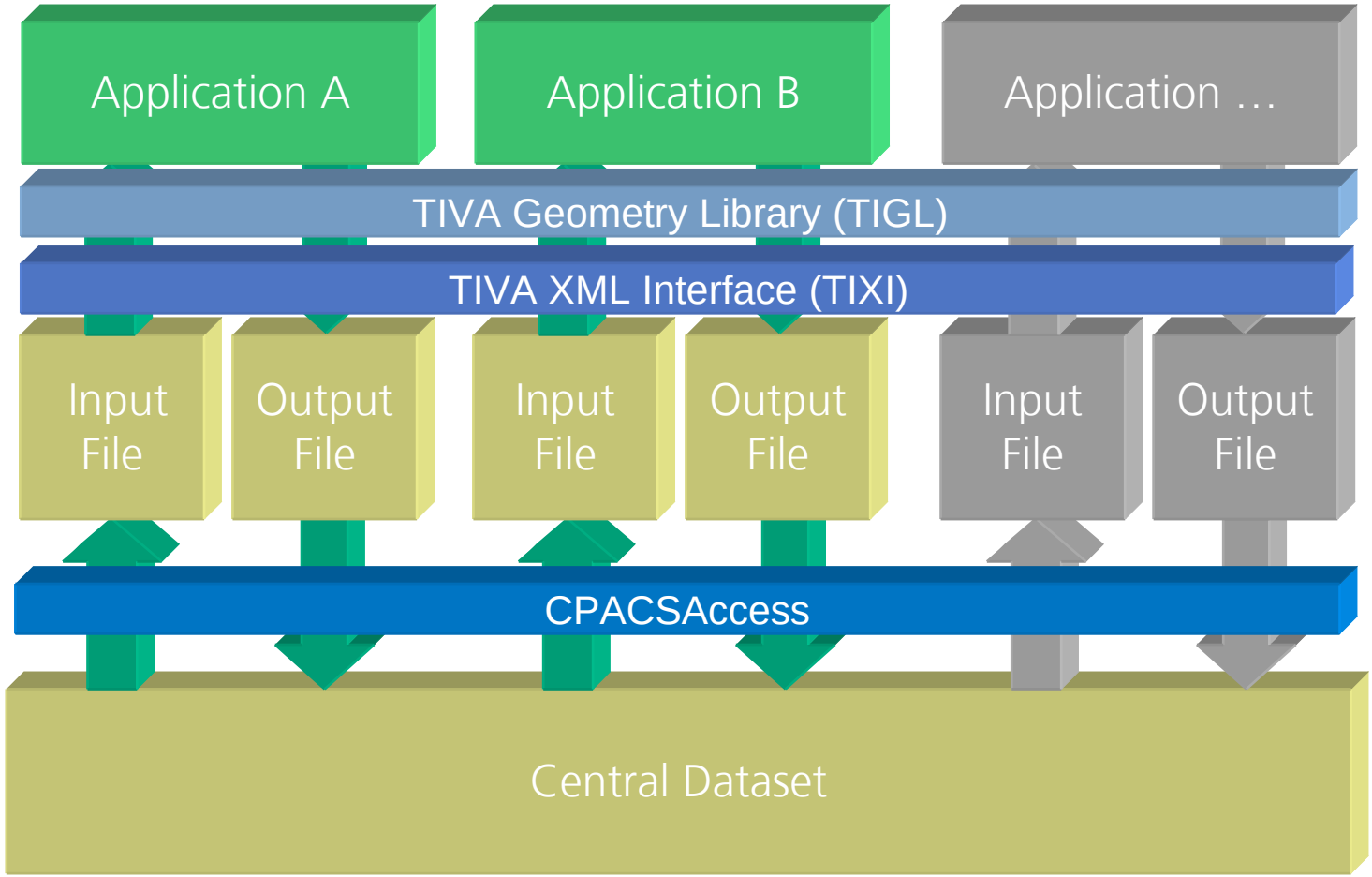
Data Import into the Central Dataset



Example of a Mapping File

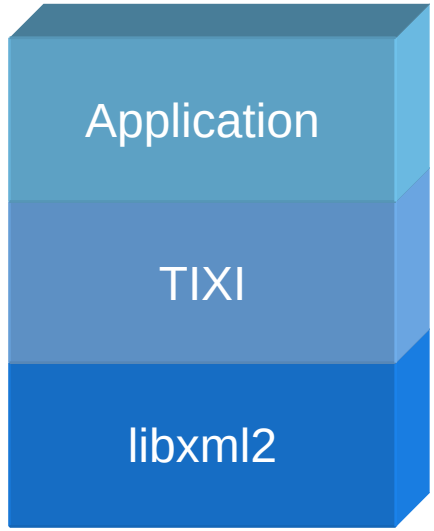
```
<?xml version="1.0" encoding="UTF-8"?>
<map:mappings xmlns:map="http://www.dlr.de/sistec/tool/mapping">

  <map:mapping>
    <map:source>/result</map:source>
    <map:target>/configuration/application[@name="IBUCK"]/result</map:target>
  </map:mapping>

  <map:mapping>
    <map:source>/result/values</map:source>
    <map:target>/configuration/common/values</map:target>
  </map:mapping>

 <map:mapping>
    <map:source>/result/old_name</map:source>
    <map:target>/configuration/common/new_name</map:target>
  </map:mapping>
     .
     .
     .
</map:mappings>
```

TIXI – TIVA XML Interface (I)

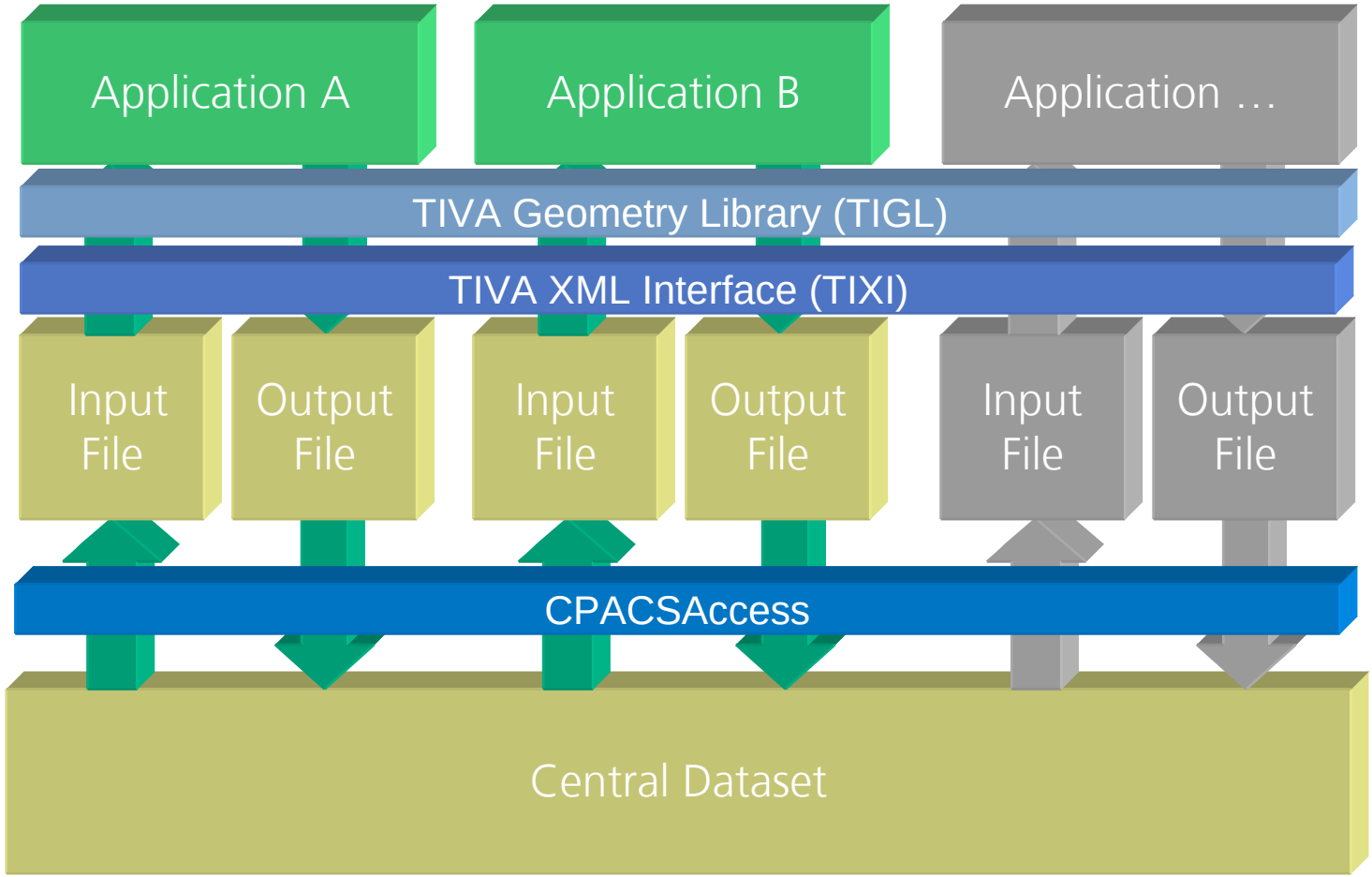


TIXI – TIVA XML Interface (II)

- Library for XML I/O
- Uses libxml2 of the Gnome project
- Provides simple access to XML content through XPath expressions
  - Functions for reading and writing of
    - Strings
    - Floating point and integer numbers
    - Matrixes
    - 3D-Points
  - Checks for existence of elements
- C, Fortran, and Python interface

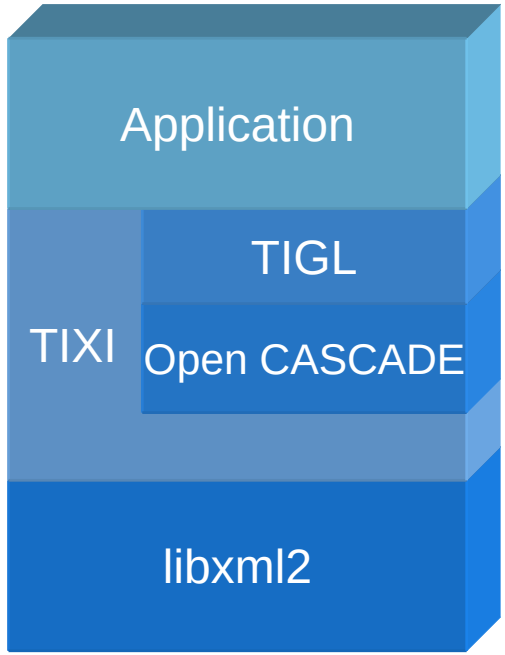


TIGL – TIVA Geometry Library (I)

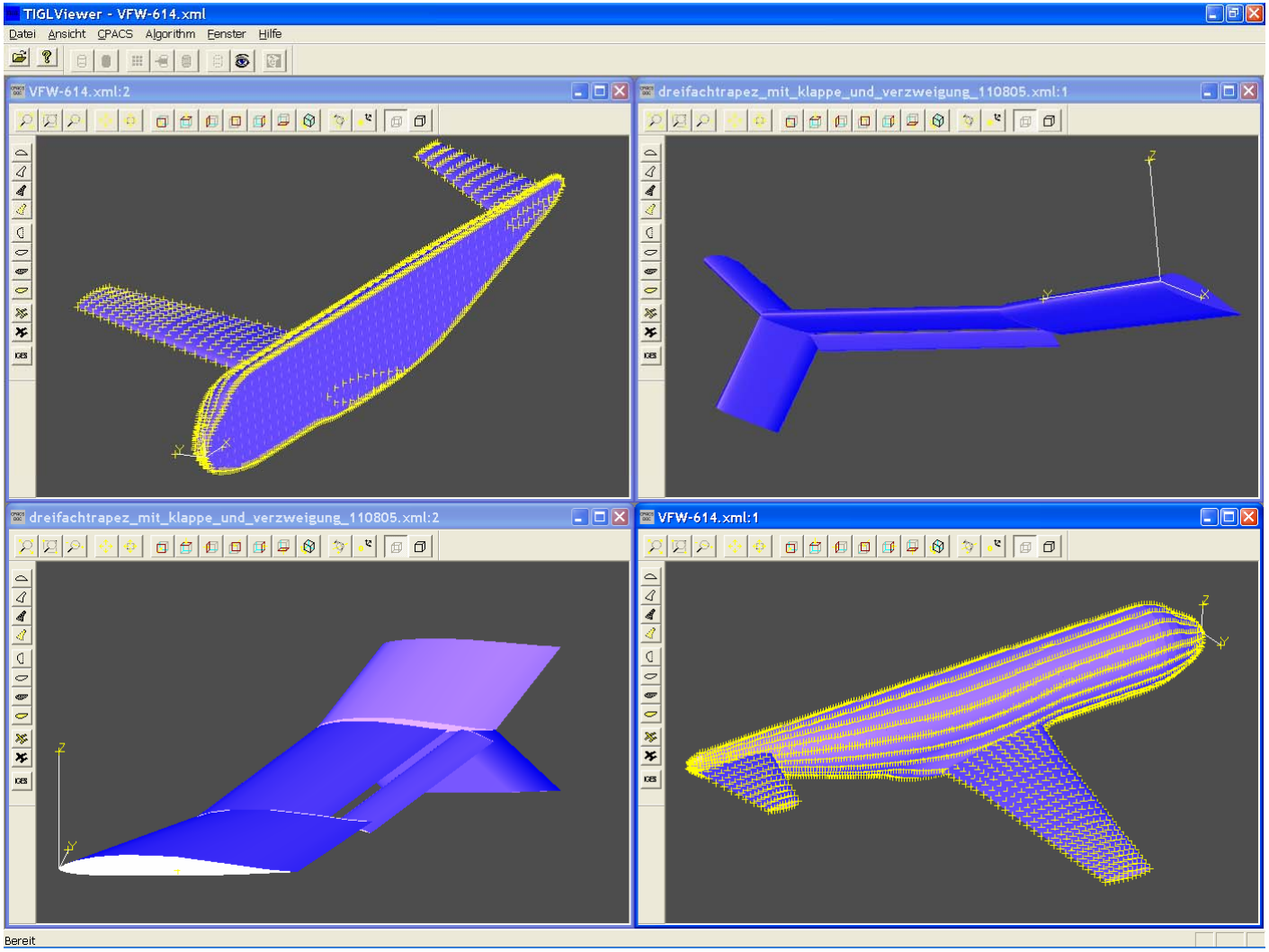


TIGL – TIVA Geometry Library (II)

- Reading and processing the geometry information stored in the central dataset
  - Currently only for fuselages and wings
  - Uses Open CASCADE
- Construction of the 3D geometry
  - Creation of surfaces from cross sections
  - Used e.g. for calculation of surface points in absolute Cartesian coordinates
- Export of the airplane geometry in IGES or STL format
- C, Fortran, and Python interface



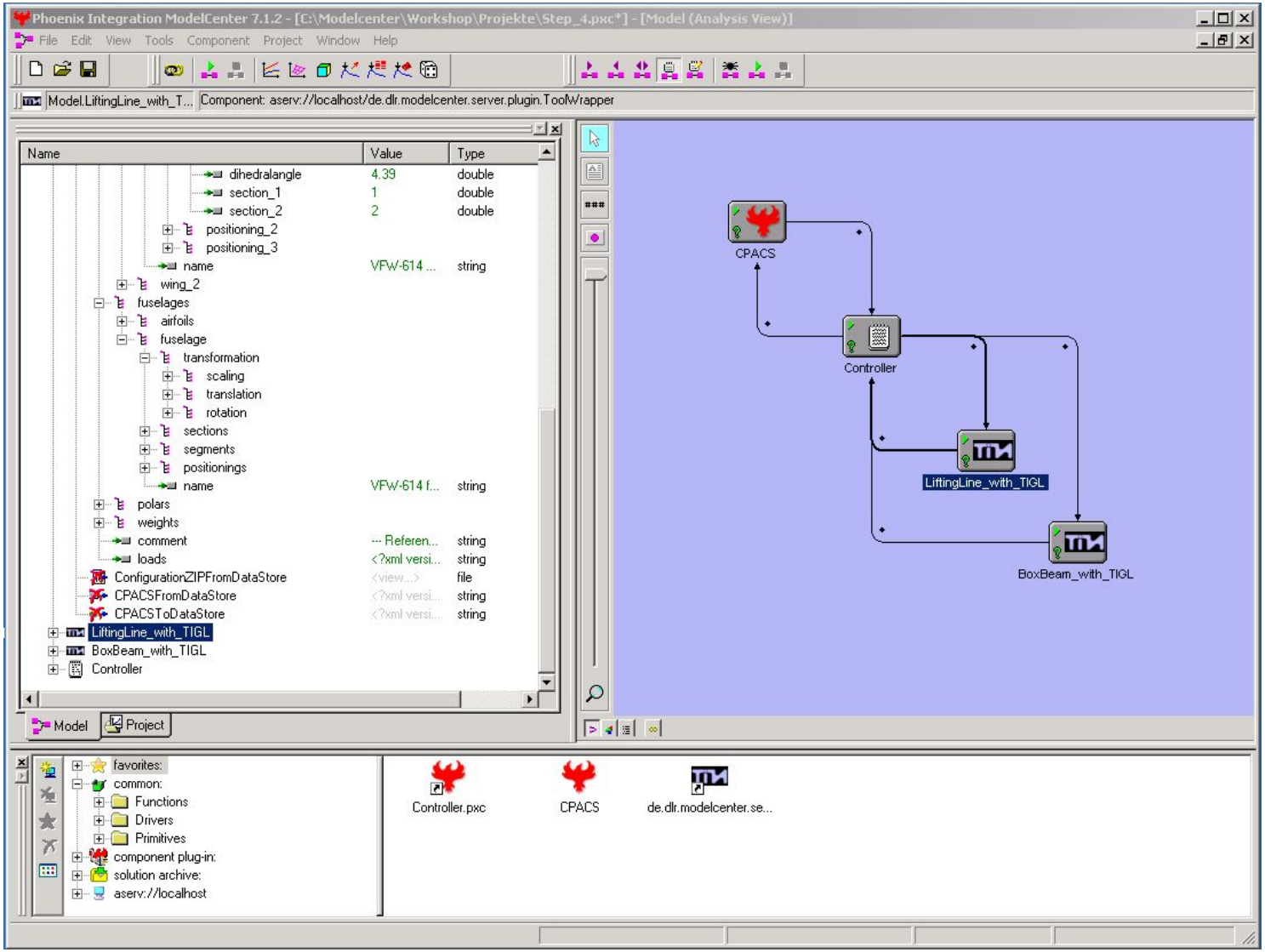
TIGLViewer – Visualization Tool for TIGL



Framework Integration (I)

- Integration framework used: ModelCenter
- Central dataset and applications components realized as ModelCenter plugins
  - Central dataset component
    - Interface between ModelCenter and central dataset
    - Update of the central dataset
    - Export and import from and into ModelCenter
  - Application wrapper component
    - Generic component that wraps an individual (standalone) application for use in ModelCenter
    - Generation of input files, application startup and mapping of results into central dataset
  - Controller component
    - Coordinates the components of the process chain

Framework Integration (II)



Summary and Outlook

- Steps to set-up a framework for collaborative engineering:
  - Define a common data format
  - Enable applications to use it
  - Integrate separate tools into a workflow system

- Future work
  - Implementation of an interface to STEP
  - Extension of the central dataset:
    - Geometry modeling for other construction units
    - Mission control data
  - Application of tools and common data format in other projects