

# Software Lab:

Modeling:	<div style="width: 100%; height: 10px; background-color: #005596;"></div>
Mathematics:	<div style="width: 25%; height: 10px; background-color: #005596;"></div>
Programming:	<div style="width: 75%; height: 10px; background-color: #005596;"></div>
Science:	<div style="width: 50%; height: 10px; background-color: #005596;"></div>

## Smart BIM Customization

### Description

Seamless BIM workflows depend on the alignment between **client-defined exchange information requirements** and the **data produced by engineers using BIM authoring tools**. In practice, this alignment is often lacking. Engineers are therefore required to manually adjust their BIM exports to meet client expectations—a process that is time-consuming, error-prone, and difficult to scale.

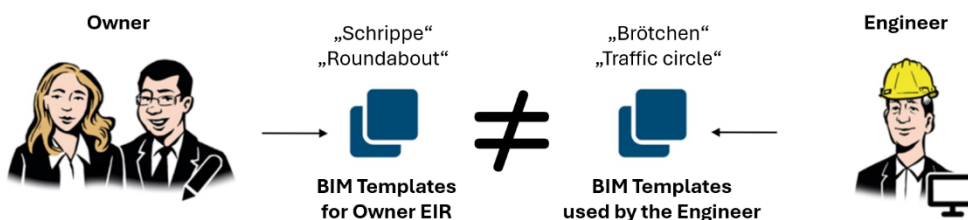
This work is based on the assumption that the majority of required information is **not missing**, but already present in the BIM model. The key challenge lies in **re-coding and mapping existing data** so that it can be correctly interpreted by the client.

The objective of this task is to develop **automated solutions** that match client expectations with data generated in established engineering workflows. Both client requirements and engineering outputs are assumed to be described using the **IDS (Information Delivery Specification) format** defined by buildingSMART.

The main focus is on designing and prototyping a **framework that compares and aligns two IDS definitions**—one representing client expectations and one representing engineering outputs—in order to generate a **patch specification** that can be applied to the engineer's BIM model to achieve compliance.

## Challenge for BIM authoring

The customer's requirements meet the engineers' optimised workflow



### Task: Tool support for conversion from Engineer BIM to AIA BIM

Team Bonsai (Krieger, Weise, Gmeiner) – Hackathon zum BIM-Portal des Bundes 2025

### Tasks

#### Analyze Information Requirements

- Study the **IDS format (buildingSMART)** and its role in expressing exchange information requirements.
- Analyze two IDS definitions:
  - one representing **client expectations**
  - one representing **engineering outputs**
- Identify overlaps, mismatches, and implicit equivalences between both definitions.
- Document typical recoding scenarios (e.g. property renaming, type mapping, value transformation).

### Design a Matching Concept

- Develop a conceptual approach for **matching client and engineer IDS definitions**.
- Define rules or strategies for:
  - identifying equivalent properties and entities
  - resolving naming and structural differences
  - prioritizing reuse of existing data over creating new data
- Describe how the matching results can be represented as a **patch specification**.

### Implement an Automated Prototype

- Implement Python-based scripts to:
  - parse and compare IDS definitions
  - match corresponding requirements
  - generate a patch specification for the IFC model
- Ensure the solution is modular and reusable.

### Apply and Validate the Patch

- Apply the generated patch to the IFC model.
- Validate the updated model against the **client IDS**.
- Compare compliance results before and after patching.

### Evaluate and Reflect

- Evaluate the effectiveness and limitations of the automated matching approach.
- Discuss scalability, robustness, and interoperability issues.
- Reflect on the potential of IDS-based automation in real-world BIM workflows.

### Supervisor

Dr.-Ing. Matthias Weise / AEC3 Deutschland GmbH, [mw@aec3.de](mailto:mw@aec3.de)

### References

*The idea has been developed in a Hackathon and is documented at this GitHub repository (<https://github.com/bimhelden/PyScripts4BPHackathon2025>)*