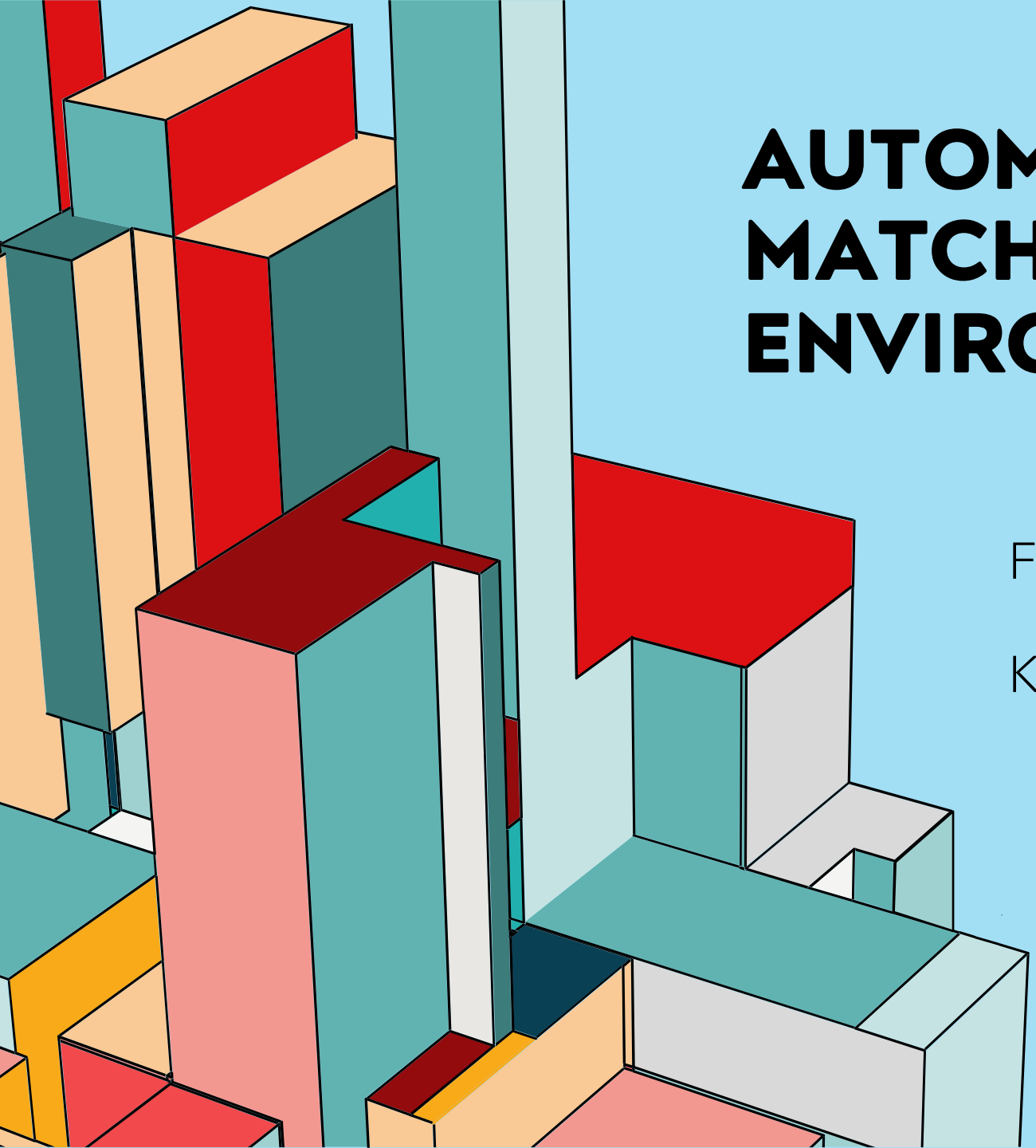


AUTOMATIZATION OF MATCHING IFC DATA AND ENVIRONMENTAL DATA

Frede Søndergaard Møllegaard - s203729

Kasper Holst – s233432

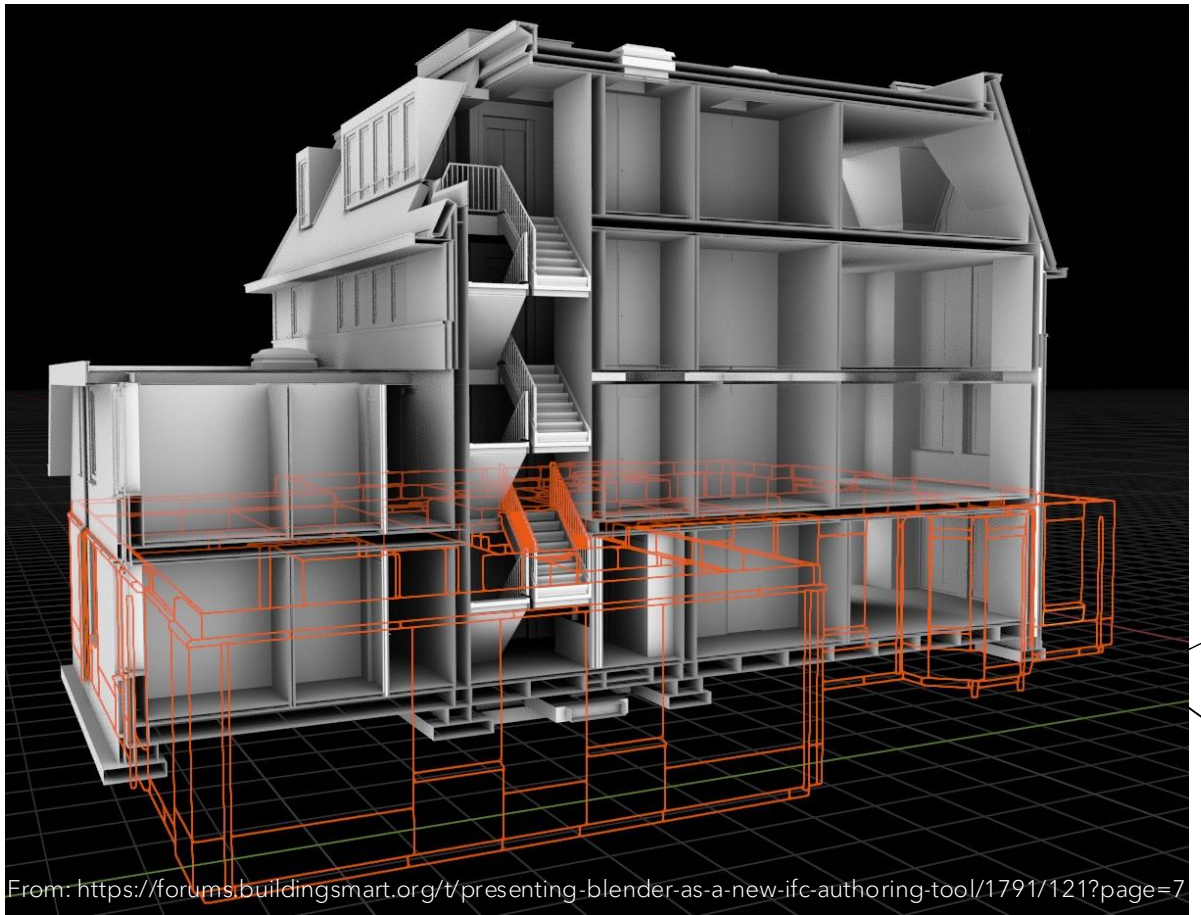
- Aim of the tool

- User of the tool

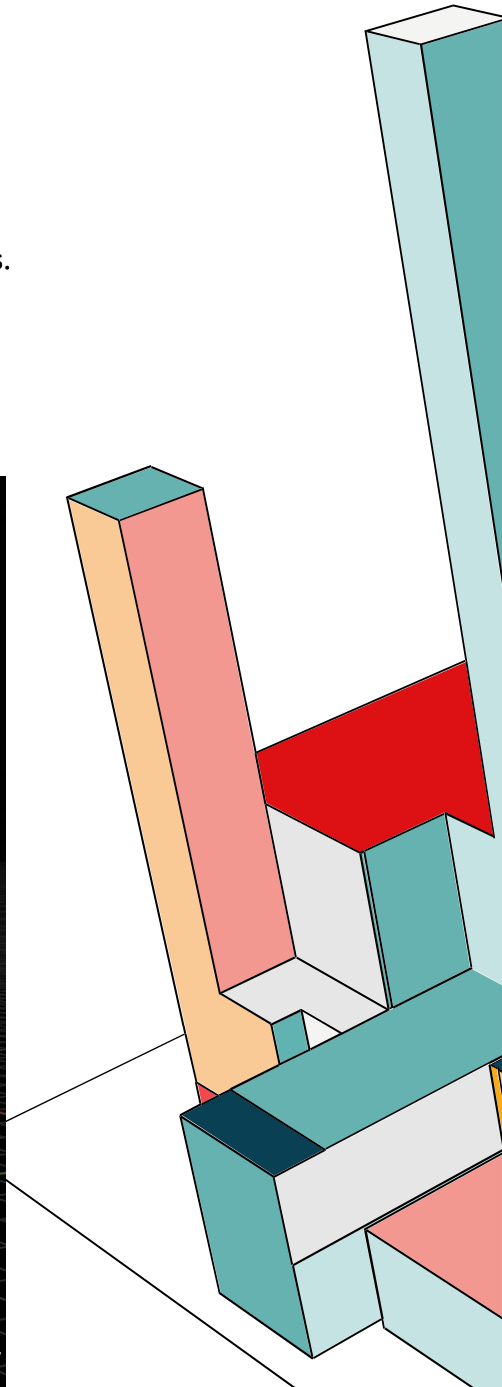
- Use case diagram

- Script of the tool

- Import Ifc data and environmental data.
- Match the materials used in the building model with the corresponding environmental data.
- Help in automatization of generating LCI list of the building model's materials.
- Reduce time used by sustainability analysts.



From: https://forums.buildingsmart.org/t/presenting-blender-as-a-new-ifc-authoring-tool/1791/121?page=7



- Aim of the tool

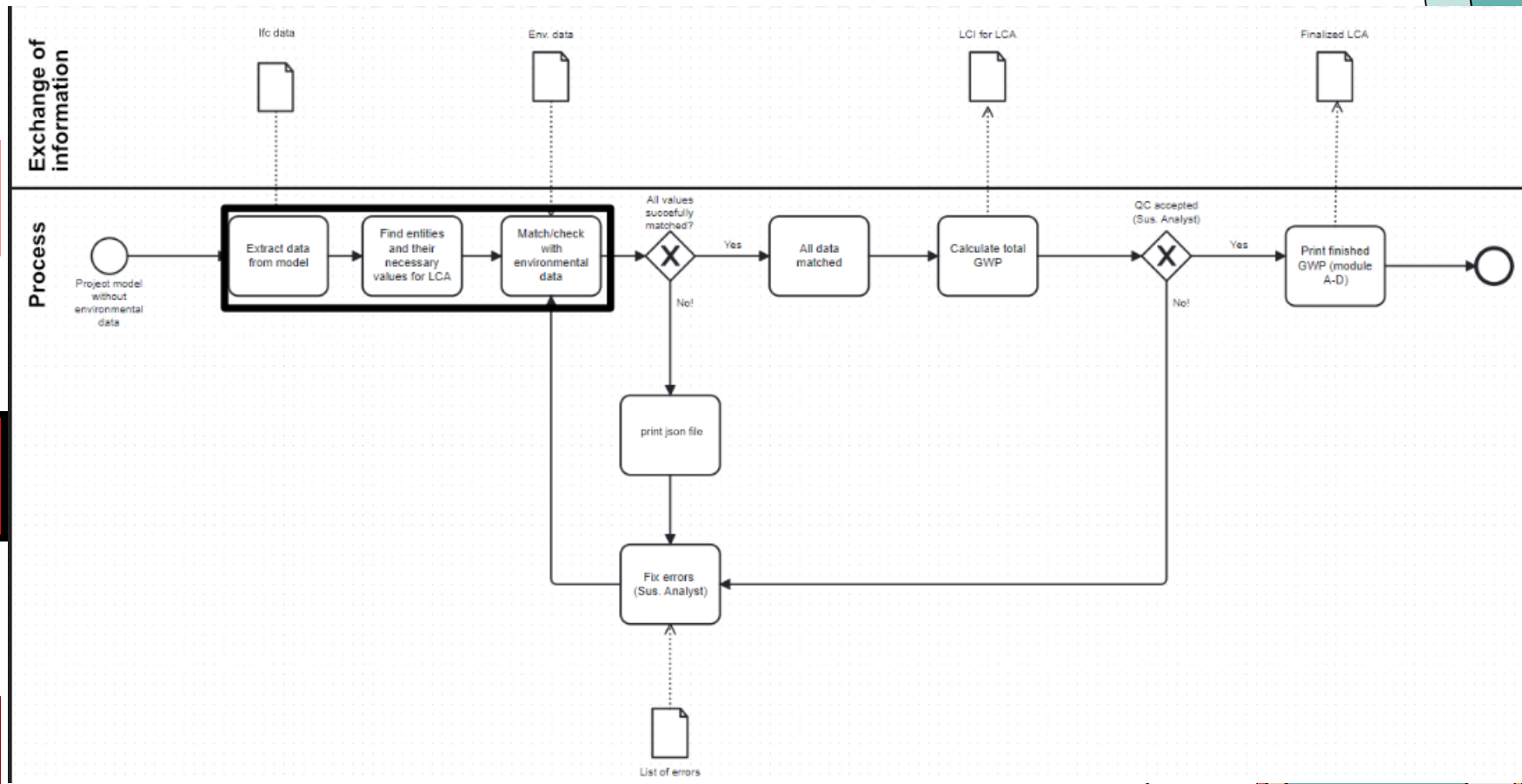
- User of the tool

- Use case diagram

- Script of the tool

- Sustaianability analysts
- Early design stage: Architects, Structure and Project managers
- Late design stage: Project managers and Architects

- Aim of the tool
- User of the tool
- Use case diagram
- Script of the tool



- Aim of the tool
- User of the tool
- Use case diagram
- Script of the tool

Import of environmental data and setting up the document matrix → making it searchable

```
# Load the Excel file
file_path_xl = 'File_path/Excel_EPD_Data.xlsx'

# Read the Excel file into a DataFrame
df = pd.read_excel(file_path_xl)
df.columns = df.columns.str.lower()
df = df.iloc[1:]
df_selected = df.iloc[:, [1, 2, 4]]
```

```
Document matrix of env. data:
    20  300mm  37  4mm  af  aluminiumsprofil  aluminum  ...  tørrede  valsede  vægelementer  wall  walls  windstopper
ood
1    0      0   0    0   0                 0         0  ...        0        0             0     0      1            0
  0
2    0      0   0    0   0                 0         0  ...        0        0             0     0      1            0
  0
3    0      1   1    0   0                 0         0  ...        0        0             0     0      1            0
  0
4    0      0   0    0   1                 0         0  ...        1        0             0     0      1            0
  1
5    0      0   0    0   0                 0         0  ...        0        0             0     0      1            1
  0
6    0      0   0    0   0                 0         0  ...        0        1             0     0      1            0
  0
7    0      0   0    0   0                 0         0  ...        0        0             0     0      1            0
  0
8    1      0   0    0   0                 0         0  ...        0        0             1     0      1            0
  0
9    1      0   0    0   0                 0         0  ...        0        0             1     0      1            0
  0
10   0      0   0    0   1                 0         0  ...        1        0             0     1      0            0
  2
11   0      1   1    0   0                 0         0  ...        0        0             0     1      0            0
  1
12   0      0   0    0   0                 0         0  ...        0        0             0     1      0            0
```

- Aim of the tool
- User of the tool
- Use case diagram
- **Script of the tool**

Retrieve Ifc model, loop through walls types and layers, make Query Vectors

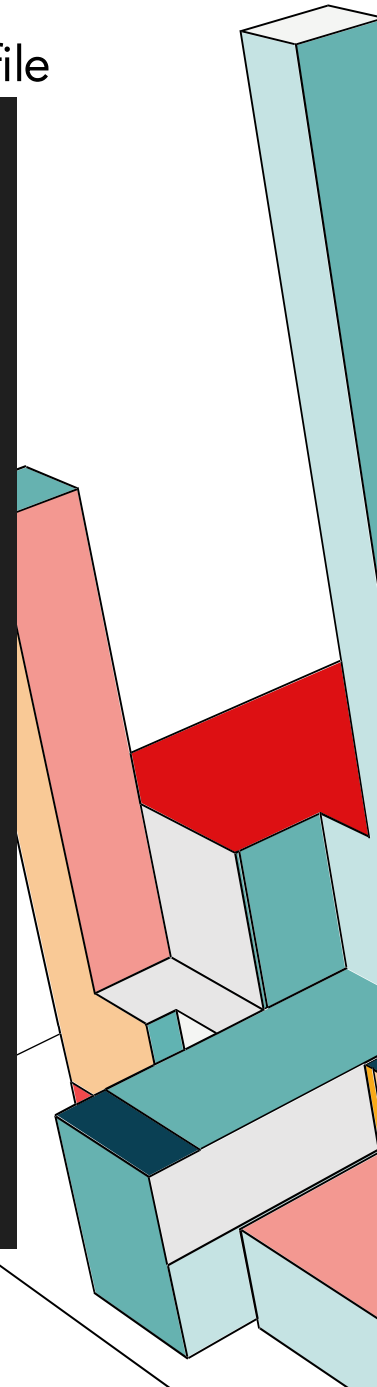
```python
# IFC file import
name = 'File_path/Adv.BIM/CES_BLD_24_06_ARC'
model_url = name + ".ifc"

if os.path.exists(model_url):
    model = ifcopenshell.open(model_url)
else:
    raise FileNotFoundError(f"Model file {model_url} does not exist")

# Initialize wall type areas and material layers
wall_types = model.by_type("IfcWallType")
wall_type_areas = {}
```

```
Query Vectors per wall:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
[0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1]
Query Vectors per wall:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
Query Vectors per wall:
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
```

- Aim of the tool
- User of the tool
- Use case diagram
- Script of the tool

Export the data that has been found by the tool into a JSON file

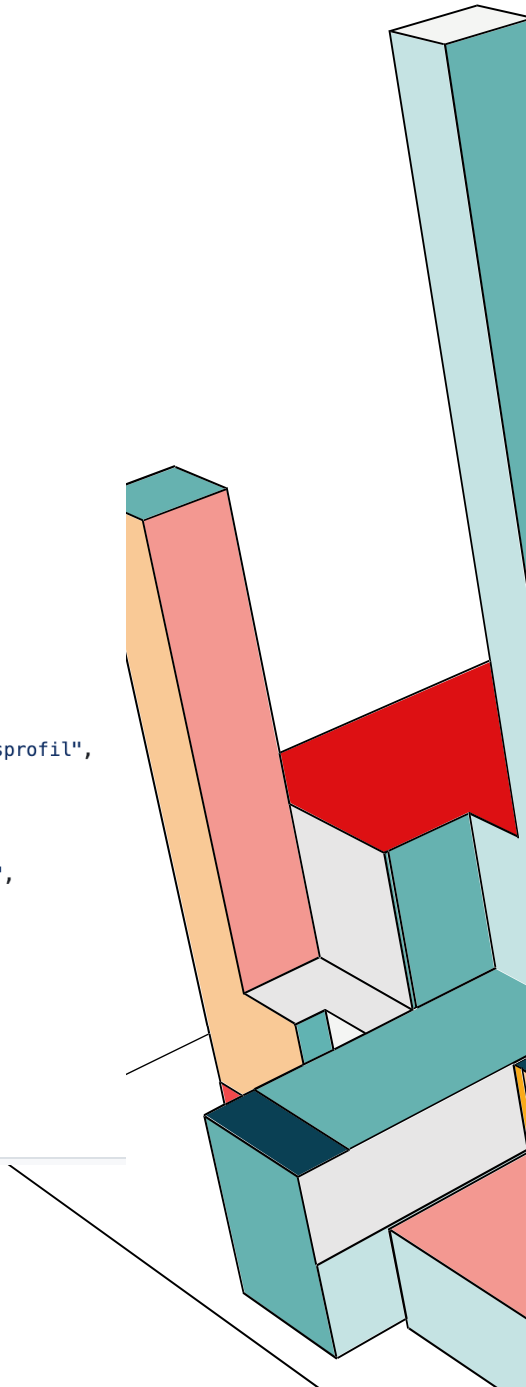
```python
# Step 2: Generate matches and export to JSON
bow_df, vectorizer = create_bow_matrix(df_selected)

def create_query_vector_for_layer(material_layer, vectorizer):
    material, thickness = material_layer
    query_string = f"{material} {thickness}"
    query_vector = vectorizer.transform([query_string]).toarray()[0]
    return query_vector

def compute_similarity_for_layer(bow_df, query_vector):
    similarities = cosine_similarity(bow_df, query_vector.reshape(1, -1))
    return sorted(list(enumerate(similarities.flatten())), key=lambda x: x[1], reverse=True)

# Structure to hold the final results
json_data = []

# Step 3: Write JSON to file
json_file_name = 'wall_material_matches.json'

with open(json_file_name, 'w') as json_file:
    json.dump(json_data, json_file, indent=4)

# Step 4: Generate the hash for the JSON file
def hash_json_file(file_path, algorithm='sha256'):
    hasher = hashlib.new(algorithm)
    with open(file_path, 'rb') as file:
        hasher.update(file.read())
    return hasher.hexdigest()

# Compute the hash of the exported JSON file
json_hash = hash_json_file(json_file_name)
print(f"The SHA-256 hash of the JSON file is: {json_hash}")
```



- Aim of the tool
- User of the tool
- Use case diagram
- Script of the tool

OUTPUT: JSON file with the five best matches


```
{
    "wall_type": "Basic Wall:Interior wall - glass",
    "wall_id": 579827,
    "area_m2": 145.9463639672935,
    "material_layers": [
        {
            "material_layer_chosen": "Glass, Clear Glazing",
            "thickness_mm": 70.0,
            "matches": {
                "match_1": {
                    "material_name": "Internal glass walls glass Glas 4mm",
                    "similarity_score": 0.7071
                },
                "match_2": {
                    "material_name": "Internal glass walls aluminum profile aluminiumsprofil",
                    "similarity_score": 0.4082
                },
                "match_3": {
                    "material_name": "Internal glass walls EPDM rubber EPDM-t\u00e6tning til aluminiumsprofil",
                    "similarity_score": 0.3015
                },
                "match_4": {
                    "material_name": "External walls Primer Overflade, facademaling, grundere, silikat",
                    "similarity_score": 0.0
                },
                "match_5": {
                    "material_name": "External walls Gypsum board Gyproc Climate",
                    "similarity_score": 0.0
                }
            }
        }
    ]
```

THANK YOU

Frede Søndergaard Møllegaard s203729

Kasper Holst – s233432