# Documentation IDM Media

Integriertes
Datenmodell

**IDM Media Doc 4.1.0**

This documentation is valid in connection with
the **IDML version 4.1.0**

**Status: 01.11.2025**

## Content

# 1. Introduction

## 1.1. Image formats

The different IDM processing systems support different image formats.

In order to provide data that can be processed by everyone, it is recommended to work only with the following image and video formats in the IDM data:
- PNG
- TIF/TIFF
- JPG/JPEG
- SVG
- MP4

## 1.2. Image sizes

An image size of 2048x2048 px is recommended for millieu images, colour advice, article images (sketches) and variant images. The specification is not obligatory, but serves as orientation for the data creators.

## 1.3. IDM version types

The setting vectors stored in the magnetic planner drawings (SVG) must correspond to the setting variant types of the respective IDML documentation. The corresponding information can be found in the section Version Types below the introduction:

| Feature | Feature text | Type | Remark/possible variations |
|---|---|---|---|
| 1160 | connecting type top 1 | O | ID of the connection vector in the SVG: AVO1 |
| 1161 | connecting type top 2 | O | ID of the connection vector in the SVG: AVO2 |
| 1162 | connecting type top 3 | O | ID of the connection vector in the SVG: AVO3 |
| 1163 | connecting type top 4 | O | ID of the connection vector in the SVG: AVO4 |
| 1164 | connecting type top 5 | O | ID of the connection vector in the SVG: AVO5 |
| 1165 | Connection type bottom 1 | O | ID of the+D127:D161 connection vector in the |
| 1166 | Connection type bottom 2 | O | ID of the connection vector in the SVG: AVU2 |
| 1167 | Connection type bottom 3 | O | ID of the connection vector in the SVG: AVU3 |
| 1168 | Connection type bottom 4 | O | ID of the connection vector in the SVG: AVU4 |
| 1169 | Connection type bottom 5 | O | ID of the connection vector in the SVG: AVU5 |
| 1170 | Connection type top 6 | O | ID of the connection vector in the SVG: AVO6 |
| 1171 | Connection type top 7 | O | ID of the connection vector in the SVG: AVO7 |
| 1172 | Connection type top 8 | O | ID of the connection vector in the SVG: AVO8 |
| 1173 | Connection type top 9 | O | ID of the connection vector in the SVG: AVO9 |
| 1174 | Connection type top 10 | O | ID of the connection vector in the SVG: AVO10 |
| 1175 | Connection type bottom 6 | O | ID of the connection vector in the SVG: AVU6 |
| 1176 | Connection type bottom 7 | O | ID of the connection vector in the SVG: AVU7 |
| 1177 | Connection type bottom 8 | O | ID of the connection vector in the SVG: AVU8 |
| 1178 | Connection type bottom 9 | O | ID of the connection vector in the SVG: AVU9 |
| 1179 | Connection type bottom 10 | O | ID of the connection vector in the SVG: AVU10 |
| 1180 | Connection type north 1 | O | ID of the connection vector in the SVG: AVN1 |
| 1181 | Connection type north 2 | O | ID of the connection vector in the SVG: AVN2 |
| 1182 | Connection type north 3 | O | ID of the connection vector in the SVG: AVN3 |
| 1183 | Connection type north 4 | O | ID of the connection vector in the SVG: AVN4 |
| 1184 | Connection type north 5 | O | ID of the connection vector in the SVG: AVN5 |
| 1185 | Connection type north 6 | O | ID of the connection vector in the SVG: AVN6 |
| 1186 | Connection type north 7 | O | ID of the connection vector in the SVG: AVN7 |
| 1187 | Connection type north 8 | O | ID of the connection vector in the SVG: AVN8 |
| 1188 | Connection type north 9 | O | ID of the connection vector in the SVG: AVN9 |
| 1189 | Connection type north 10 | O | ID of the connection vector in the SVG: AVN10 |
| 1190 | Connection type south 1 | O | ID of the connection vector in the SVG: AVS1 |
| 1191 | Connection type south 2 | O | ID of the connection vector in the SVG: AVS2 |
| 1192 | Connection type south 3 | O | ID of the connection vector in the SVG: AVS3 |
| 1193 | Connection type south 4 | O | ID of the connection vector in the SVG: AVS4 |
| 1194 | Connection type south 5 | O | ID of the connection vector in the SVG: AVS5 |
| 1195 | Connection type south 6 | O | ID of the connection vector in the SVG: AVS6 |
| 1196 | Connection type south 7 | O | ID of the connection vector in the SVG: AVS7 |
| 1197 | Connection type south 8 | O | ID of the connection vector in the SVG: AVS8 |
| 1198 | Connection type south 9 | O | ID of the connection vector in the SVG: AVS9 |
| 1199 | Connection type south 10 | O | ID of the connection vector in the SVG: AVS10 |

In addition, the variant types planning width and depth were defined:

| | | | |
|---|---|---|---|
| 1095 | planning width | M | width of the article |
| 1096 | planning depth | M | depth of the article |
| 1097 | planning height | M | height of the article |

The valid setting variant types can be defined on the article via connection rules, if required also depending on the variant.

# 2. Description of the SVG files

## 2.1. General

Basically, the XML-based SVG format has been defined as the standard format for the creation of the graphical data for the 2D Magnetic Planner in IDM Living format. UTF-8 is to be used as the standard for the XML information.

In this way, you can scale up or down as you wish and display a connection behaviour.
In the format, connection vectors can be drawn and named. In addition, outlines must be drawn, which are used for collision checks.

SVG data is stored in XML structures and can be generated with various graphics programmes, e.g. Adobe-Illustrater, CorelDRAW or SketchUp.

In the SVG format, it is possible on the one hand to store the technical information for the functionalities of a magnet planner in a generally valid way, and on the other hand no data creator is regulated in the graphic representation of his assortments.

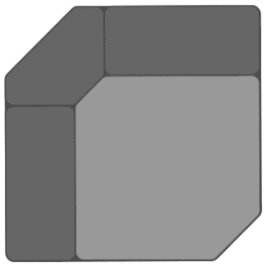
Once an SVG file has been created, it should not be edited. If editing is necessary, the original should always be used and then saved as an SVG.

## 2.2. Basic information on the SVG

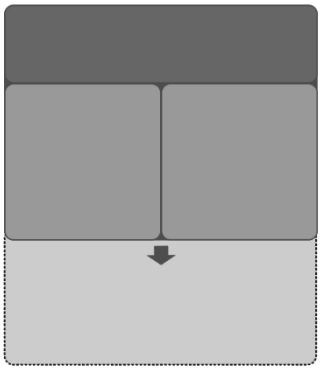
- Total object is displayed starting from x in interval 0-1 and y in interval 0-1
- If the line width is to be mapped completely, values cannot be based on x: 0 and y: 0 but must be increased by half the line width.
- Each type of connection variant, e.g. AVL or AVR, may only be present in an SVG a maximum of once.
- Width and height are integrated into the header of the SVG, no specifications in % are allowed.
- As of IDML format 4.0.0, the DPI value is integrated as an element below DETAIL_INFO behind the LAYER element:
    - Adobe Illustrator 72 dpi
    - Coral Draw 96 dpi
- All objects are shown on a scale of 1:10.
- The number of nodes must be reduced to a minimum when drawing the vectors.
- The orientation of the drawing must match the scale.
- Orientation on the screen when creating the graphics:
    - Back part on top
    - Front part of seat below
    - Corners on top and left side back part
- The following must be observed for synchronisation from 2D to 3D:
    - A line thickness of 0 is assumed for calculation and scaling
    - The 0 point of the 3D graphic is derived from the upper left corner of the OLT of an SVG
    - The dimensions of the item must be identical to the OLT on a scale of 1:10

### 2.3. Examples of orientation

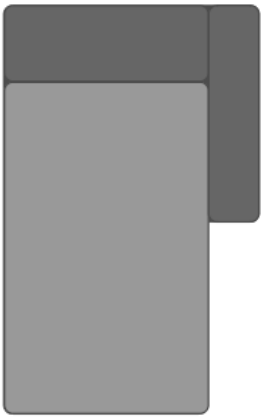
Corner



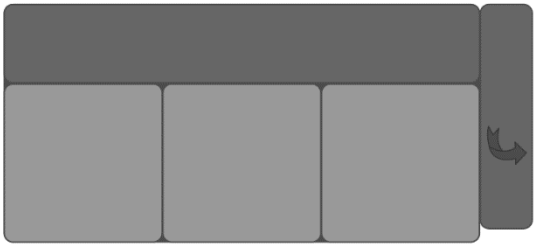
2-seater with function



Longchair with armrest



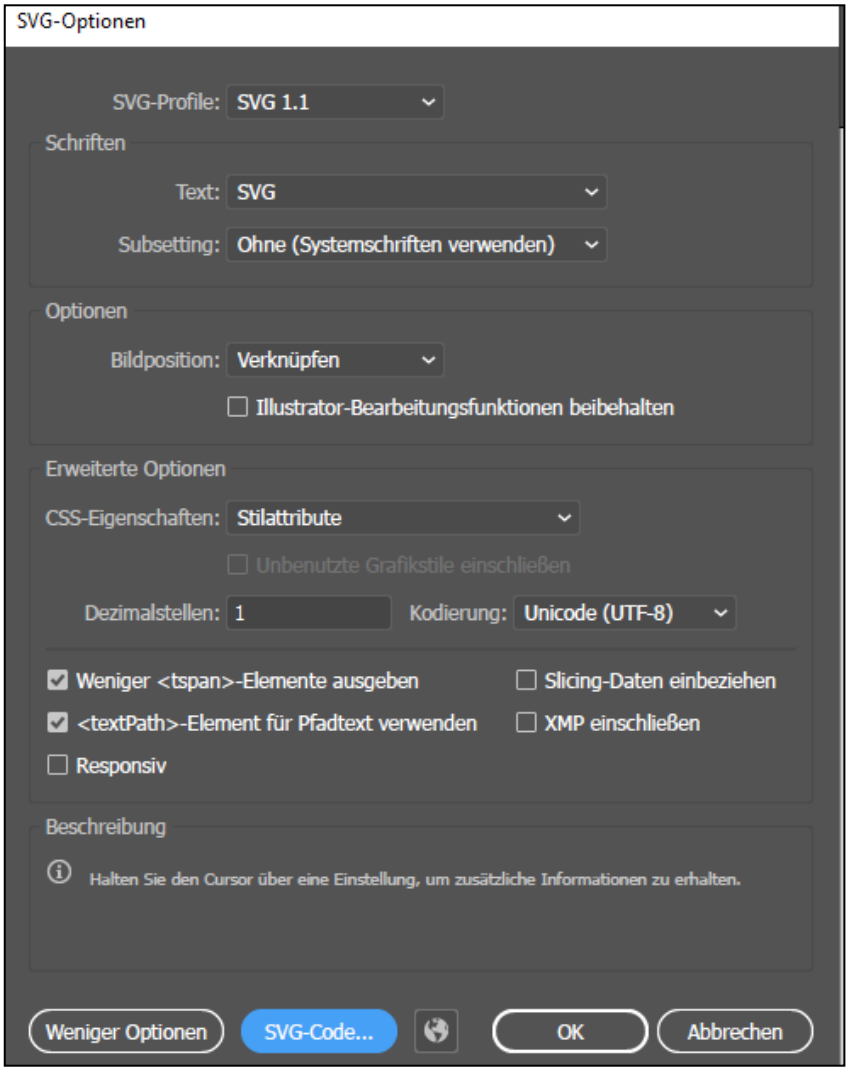
3-seater with armrest



## 2.4. Recommendations for saving an SVG file

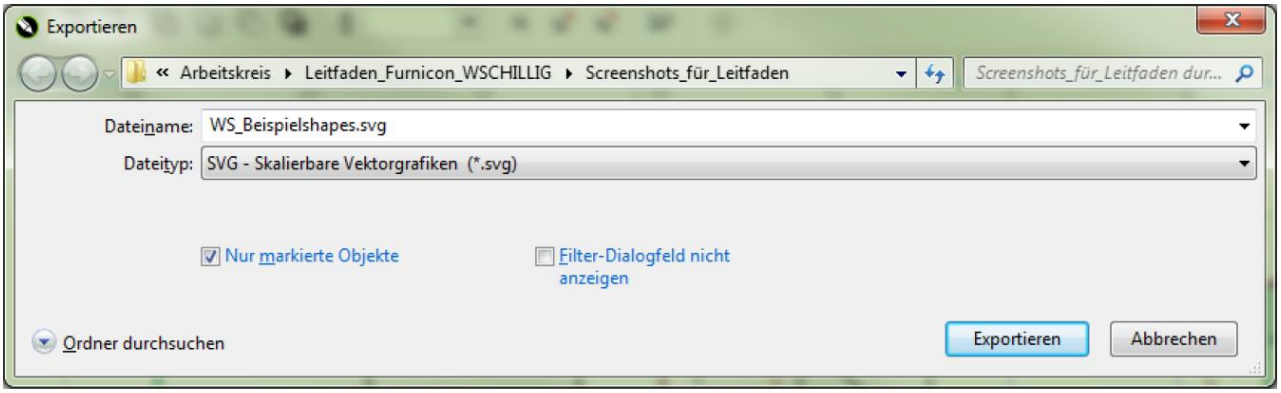
### 2.4.1. Illustrator

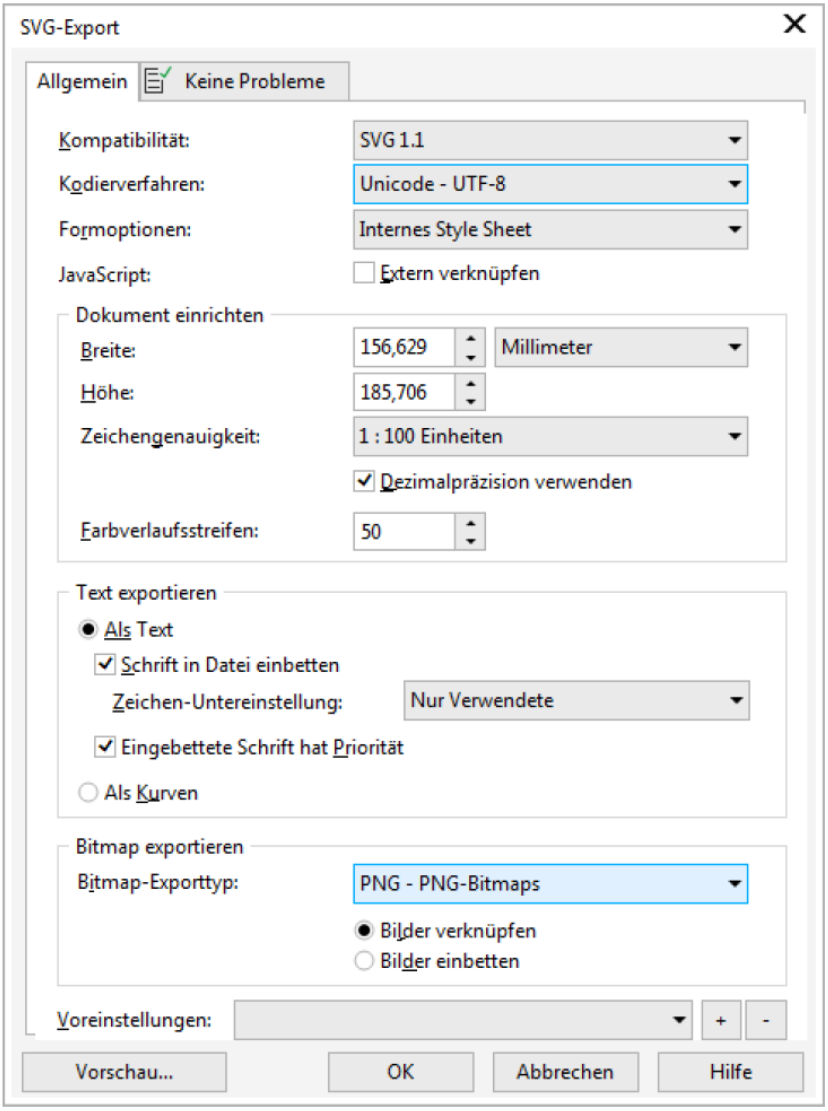
SVG saving options in Adobe Illustrator (to include the dimensions of the drawing area, the "Responsive" checkbox must be unchecked)



### 2.4.2. Corel Draw

Select SVG with all layers and check "Only selected objects".



## 2.5. Info structures in the SVG file

In order to be able to read out the technical and graphic information for a magnet planner unambiguously from the SVG files, it is necessary to clearly identify the individual pieces of information within the SVG.

The following structure objects have been defined, which are further described in detail using examples below:

- VIT (Visual Image Type)
- VIF (Visual Image Function)
- TE (Technical Level )
    - SAH1 (StretchAreaHorizontal 1)
    - SAH2 (StretchAreaHorizontal 2)
    - SAH...(StretchAreaHorizontal... )
    - SAV1 (StretchAreaVertical 1)
    - SAV2 (StretchAreaVertical 2)
    - SAV... (StretchAreaVertical...)
    - AVL (Connection Vector Left)
    - AVR (Connection Vector Right)
    - AVN1(Connection vector north1)
    - AVN2(Connection vector north2)
    - AVN...(Connection Vector North...)
    - AVS1(Connection vector south1)
    - AVS2(Connection vector south2)
    - AVS...(Connection Vector South...)
    - AVU1(Connection Vector Bottom1)
    - AVU2(Connection Vector Sub2)
    - AVU...(Connection Vector Down...)
    - AVO1(Connection Vector Top1)
    - AVO2(Connection vector Top2)
    - AVO...(Connection Vector Top...)
    - OLF (Outline Function )
    - OLT (Outline Type )

## 2.6. General specifications for the structure objects

It is to be regulated that the elements are in different levels. There are levels for VIT and VIF and the technical level "TE" for all other elements. The elements themselves can also be in the group defined under the ID.

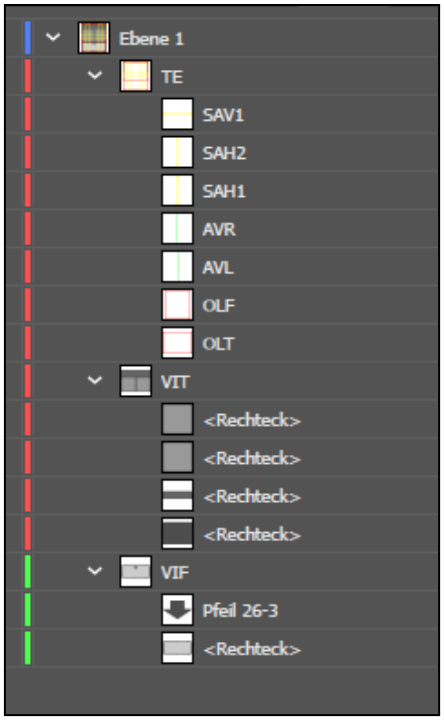
The technical level (TE) is in principle hidden by the configurators.

Everything that is not in the TE level is not hidden, is visible, unless it is switched to invisible.

Structure objects of type OLT and AVX should be placed on a technical level and thus be invisible

The software interpreting the SVG must be able to make the technical layer invisible, even if it is defined as visible in the SVG data.

As a matter of principle, transparency is to be dispensed with.

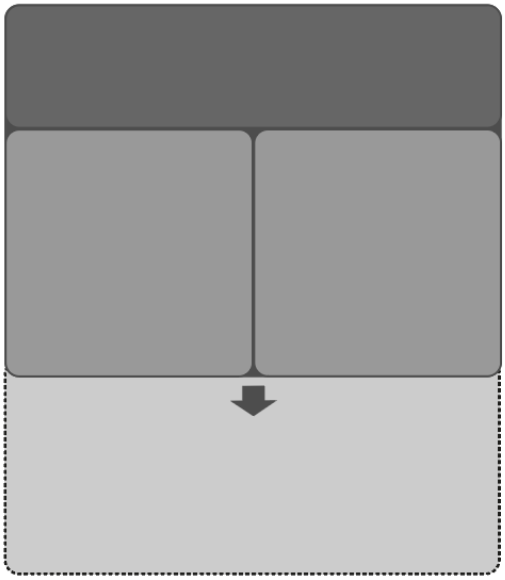


## 2.7. Description of the structural objects

The description of the structural objects is based on the following example article:

**2-seater with pull-out bed function**

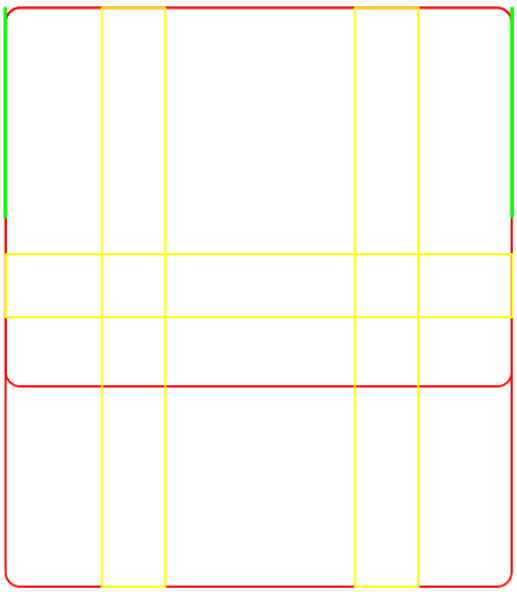
Levels VIT / VIF



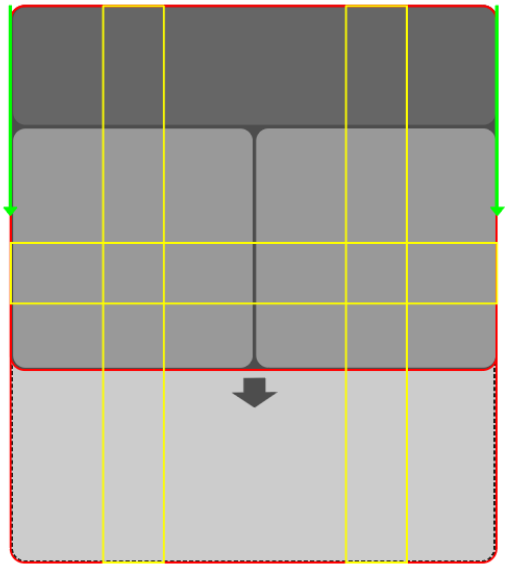
Visualisation levels VIT

VIF

Level TE



Levels VIT / VIF / TE

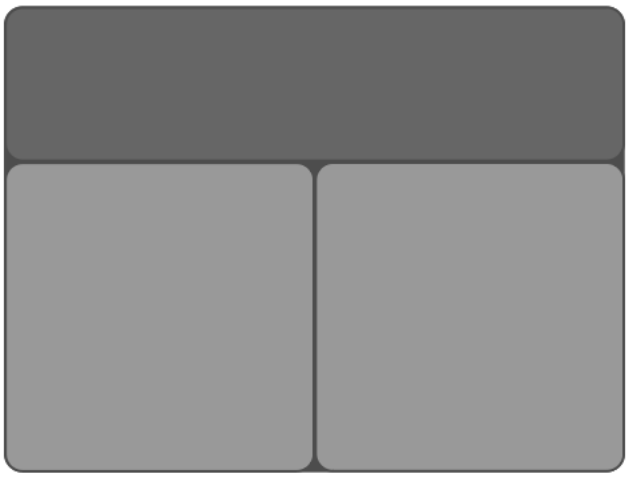


## 2.8. Visualisation levels (TYPE and FUNCTION)

- VIT (Visual Image Type)
- VIF (Visual Image Function)

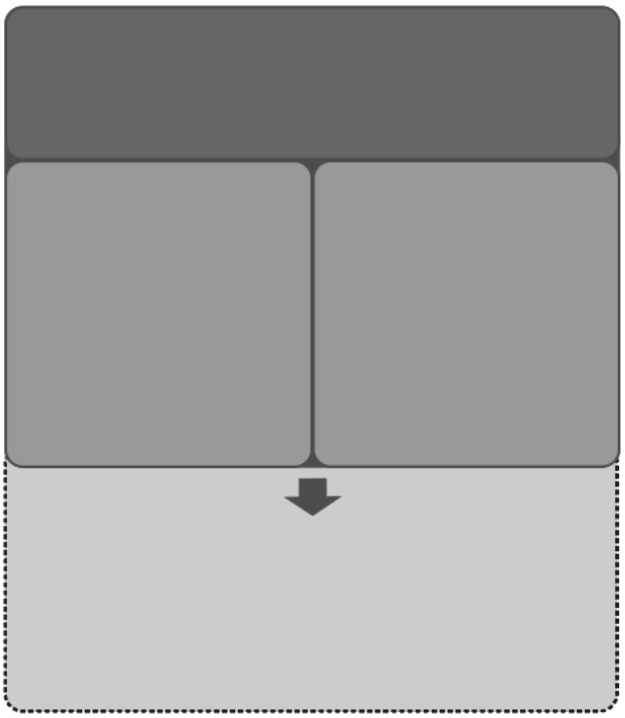
- The visualisation levels VIT and VIF are represented in the structure as independent levels
- Below the layers, any graphic objects can be used for individual presentation.
- id="VIT" and id="VIF" are identifiable in the SVG.
- Transparency should be avoided in the functional areas.
- It is recommended to place the VIF level above the VIT level.
- The VIT level should not be hidden.

VIT



VIF

VIT and VIF

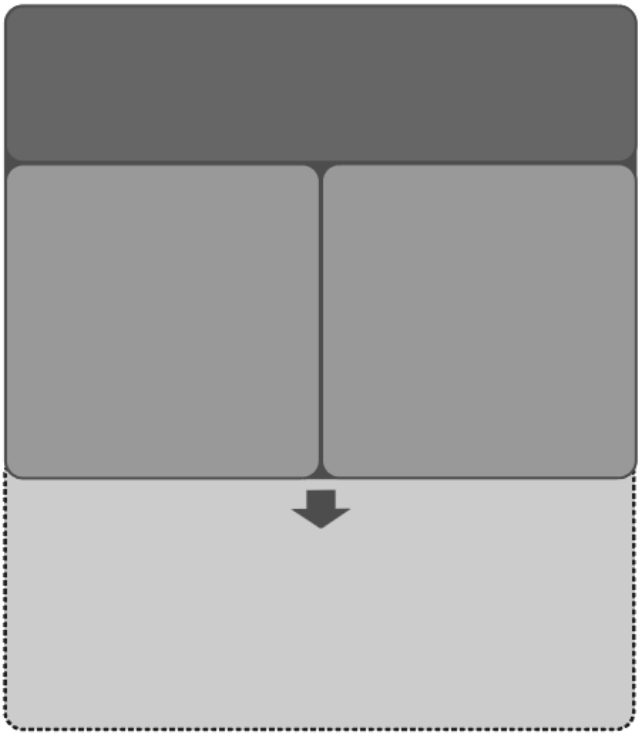


```
020     </foreignObject>
021     <g i:extraneous="self">
022         <g id="VIT">
023             <g>
024                 <path fill="#737373" d="M0,14.173C0,6.378,6.378,0,14.173,0h357.165c7.795,0,14.173,6.378,14.173,14.173v240.945
025                     c0,7.795-6.378,14.173-14.173,14.173H14.173C6.378,269.292,0,262.914,0,255.118V14.173z"/>
026             </g>
027             <g>
028                 <g>
029                     <path fill="#BFBFBF" d="M0,14.173C0,6.378,6.378,0,14.173,0h357.165c7.795,0,14.173,6.378,14.173,14.173v34.016
030                         c0,7.795-6.378,14.173-14.173,14.173H14.173C6.378,62.362,0,55.984,0,48.189V14.173z"/>
031                 </g>
032                 <defs>
033                     <filter id="Adobe_OpacityMaskFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="385.511" height="62.362">
034
035                         <feColorMatrix  type="matrix" values="-1 0 0 0 1  0 -1 0 0 1  0 0 -1 0 1  0 0 0 1 0" color-interpolation-filters="sRGB" result="source"/>
036                     </filter>
037                 </defs>
038                 <mask maskUnits="userSpaceOnUse" x="0" y="0" width="385.511" height="62.362" id="SVGID_1_">
039                     <g filter="url(#Adobe_OpacityMaskFilter)">
040
041                         <image overflow="visible" width="391" height="68" xlink:href="
042 EAMCAwYAAAALEAAAFVgAABoP/2wCEABALCwsMCxAMDBAXDw0PFxsUEBAUGx8XFxcXFx8eFxoaGhoX
```

```
186         </g>
187         <g id="VIF">
188
189             <polygon fill="none" stroke="#737373" stroke-width="1.0001" stroke-miterlimit="2.6131" stroke-dasharray="5.0003,3.0002" points="
190             385.592,473.426 408.191,496.064 408.22,473.418          "/>
191
192             <polygon fill="none" stroke="#737373" stroke-width="1.0001" stroke-miterlimit="2.6131" stroke-dasharray="5.0003,3.0002" points="
193             0,269.292 0,473.387 408.189,473.387 408.189,291.967 385.513,269.337             "/>
194             <g>
195                 <path fill="none" stroke="#737373" stroke-width="0.7087" stroke-miterlimit="2.6131" d="M187.749,277.973
196                     c-3.118,0-4.187,2.076-2.374,4.61317.583,10.616c1.813,2.537,4.778,2.537,6.59,017.583-10.616
197                     c1.812-2.537,0.744-4.613-2.375-4.613H187.749z"/>
198             </g>
199         </g>
```

2-seater intermediate unit without functional surface (VIT)

2-seater intermediate unit with functional surface (VIT and VIF)



## 2.9. OLT (OutlineType)

- OLT is a continuous closed graph. It can be a poligon train,which represents the footprint of the item at a scale of 1:10.
- The outline type is identifiable in the SVG as polygon id="OLT".
- In addition to the polygon element, path, circle and rect are also permitted.
- The size references the IDM variant types
  - 95 - Planning width
  - 96 - Planning depth
- If no planning width and planning depth are available, the dimensions from the SVG apply.
- Structure objects of type OLT should be placed on a technical level and thus be invisible.
- The software interpreting the SVG must be able to switch the technical layer invisible, even if it is defined as visible in the SVG data.

**OLT (on VIT)**

**OLT (alone)**

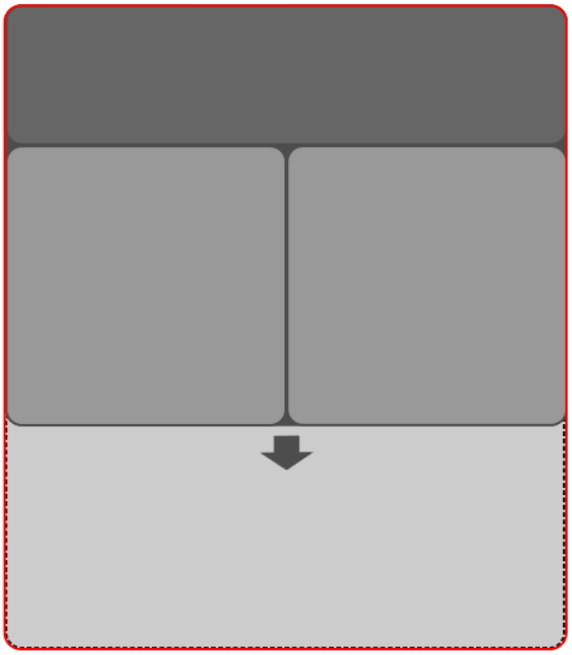
```
<g id="TE">
    <polygon id="SAV1" fill="none" stroke="#640000" stroke-miterlimit="2.6131" points="0,151.654 385.512,151.654 385.512,180
        0,180 "/>
    <polygon id="OLF" fill="none" stroke="#FA0064" stroke-miterlimit="2.6131" points="408.189,291.968 385.513,269.292 385.511,0
        0,0 0,260.789 0,473.386 385.513,473.386 408.191,496.064 "/>
    <polygon id="OLT" fill="none" stroke="#006400" stroke-miterlimit="2.6131" points="385.513,269.292 385.513,269.245
        385.511,269.245 385.511,0 0,0 0,269.245 "/>

        <line id="AVL" fill="none" stroke="#0064C8" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="0" y1="0" x2="0" y2="260.788"/>

        <line id="AVR" fill="none" stroke="#00C800" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="385.511" y1="0.001" x2="385.511" y2="260.789"/>
</g>
```

- .

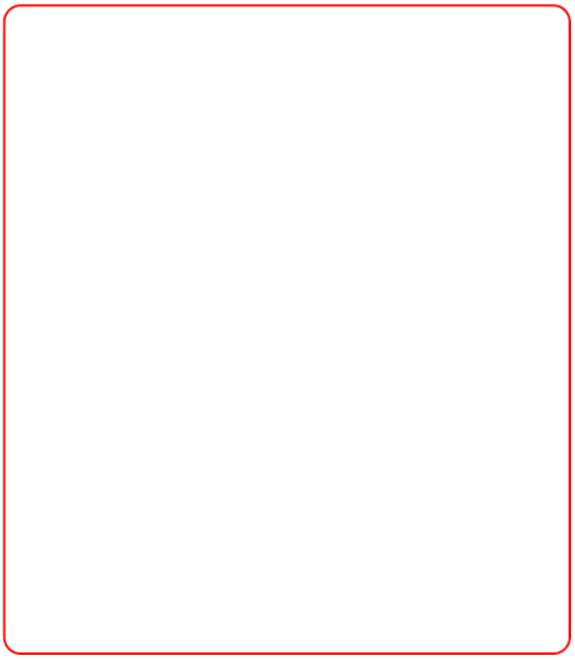
## 2.10. OLF (outline function)

- OLF is an element that is closed in one go and represents the footprint of the item including the extended functions.
- The outline function is identifiable in the SVG as polygon id="OLF".
- In addition to the polygon element, path, circle and rect are also permitted.
- For the OLF the dimensions of the SVG apply
- Structure objects of type OLF should be placed on a technical level and thus be invisible.
- The software interpreting the SVG must be able to switch the technical layer invisible, even if it is defined as visible in the SVG data.

**OLF (on VIT/VIF)**



**OLF (alone)**



```
200         <g id="TE">
201             <polygon id="SAV1" fill="none" stroke="#640000" stroke-miterlimit="2.6131" points="0,151.654 385.512,151.654 385.512,180
202                 0,180           "/>
203             <polygon id="OLF" fill="none" stroke="#FA0064" stroke-miterlimit="2.6131" points="408.189,291.968 385.513,269.292 385.511,0
204                 0,0 0,260.789 0,473.386 385.513,473.386 408.191,496.064             "/>
205             <polygon id="OLT" fill="none" stroke="#006400" stroke-miterlimit="2.6131" points="385.513,269.292 385.513,269.245
206                 385.511,269.245 385.511,0 0,0 0,269.245             "/>
207
208                 <line id="AVL" fill="none" stroke="#0064C8" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="0" y1="0" x2="0" y2="260.788"/>
209
210                 <line id="AVR" fill="none" stroke="#00C800" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="385.511" y1="0.001" x2="385.511" y2="260.789"/>
211         </g>
```

## 2.11. Connection vectors

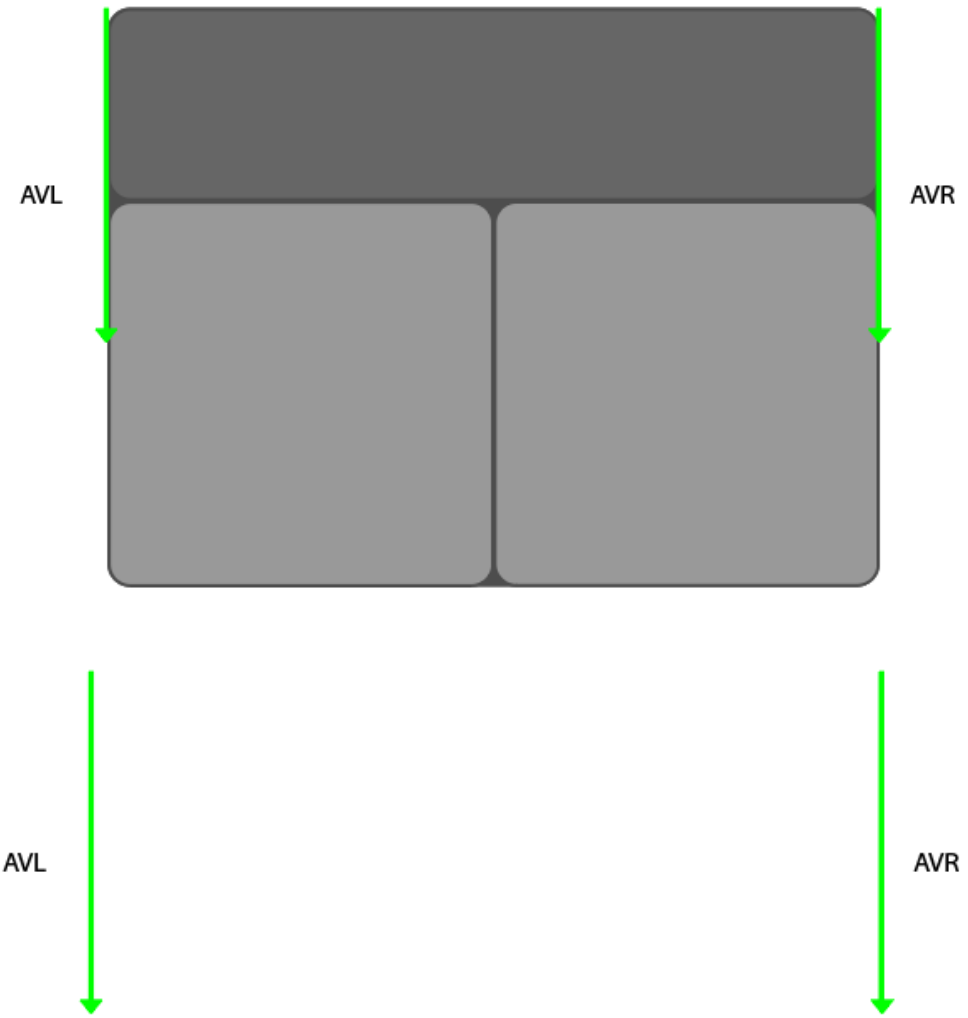
- A connection vector is a line that defining the position to connect the neighbour element.
- The connection vector is identifiable as line id="AV...".
- Connection vectors AVL and AVR should start at the back side (if possible) and should be set towards the seat being directed to one side
- There must be no different directions for connection pairs.
- When snapping, the starting point is placed on the starting point according to the direction of the vectors to be connected.
- Connection vectors can be of any length, they may be partially or completely outside the OLT, but they must be within the drawing area.

There are the following connection vectors, with values indicated in numbers from 1-10:

- AVL (Approach Vector Left)
- AVR (Approach Vector Right)
- AVN1(Approach vector north1)
- AVN2(Approach vector north2)
- AVN... (Approach Vector North...)
- AVS1(Approach vector south1)
- AVS2(Approach vector south2)
- AVS... (Approach Vector South...)
- AVU1(Approach Vector Bottom1)
- AVU2(Approach Vector Sub2)
- AVU... (Approach Vector Down...)
- AVO1(Approach Vector Top1)
- AVO2(Approach vector Top2)
- AVO... (Approach Vector Top...)

This example has only one AVL and AVR

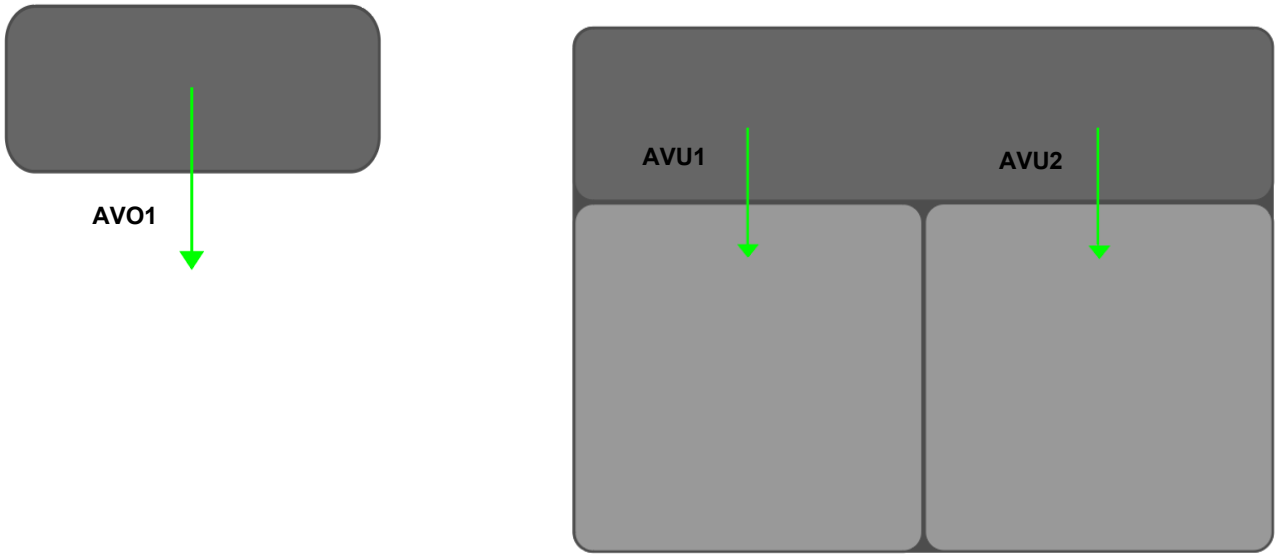
**AVL* / AVR ***



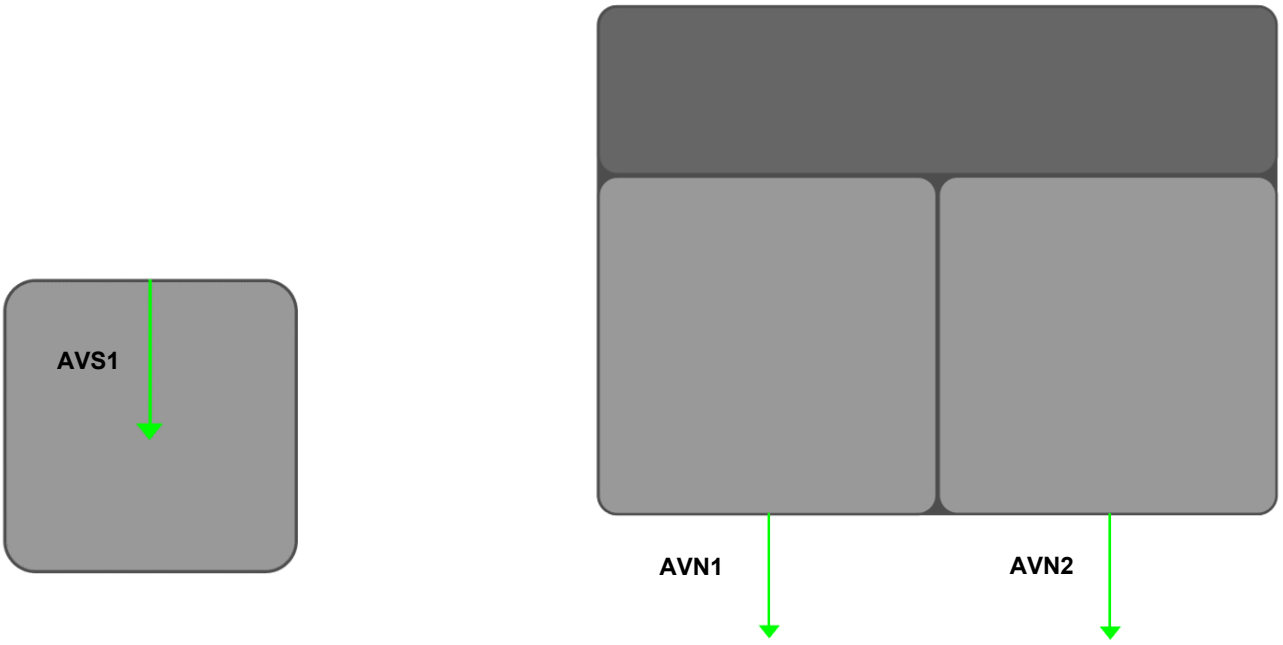

```
207
208             <line id="AVL" fill="none" stroke="#0064C8" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="0" y1="0" x2="0" y2="260.788"/>
209
210             <line id="AVR" fill="none" stroke="#00C800" stroke-width="0.9841" stroke-miterlimit="2.6131" x1="385.511" y1="0.001" x2="385.511" y2="260.789"/>
211         </g>
```

### 2.11.1. Examples for the use of the connection vectors

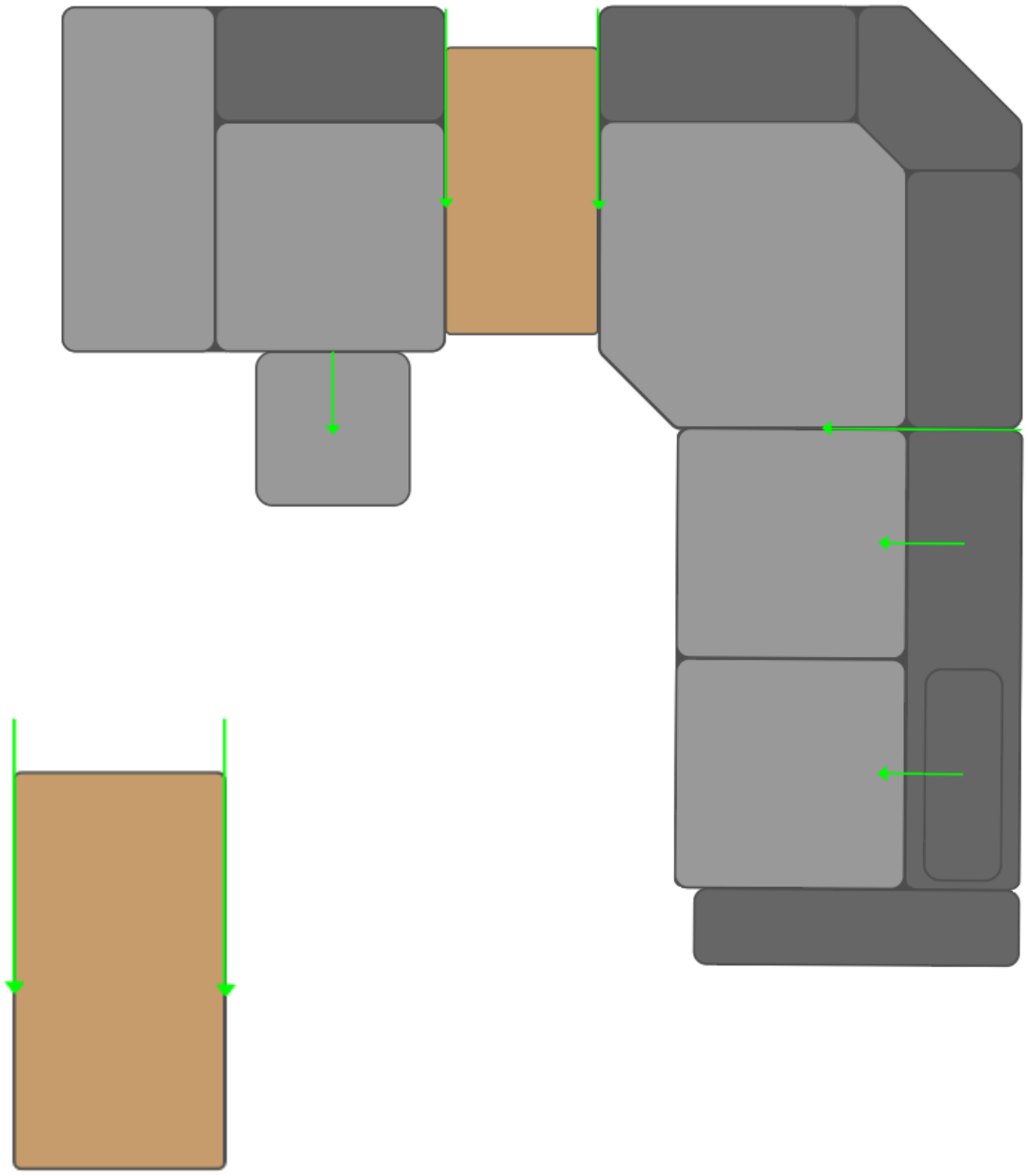
**Example for AVU /AVO** (e.g. for headrests)**




**Example for AVN /AVS** (e.g. for footrests)**



**Example of connection vectors* outside the OLT**



* The arrow symbol is only used to indicate the direction of the vector

## 2.12. StretchArea

The stretch area describes the area of the entire article that is proportionally variable when the planning width (variant type 95) or planning depth (variant type 96) is changed.

The width and depth of the SVG is determined by the planning width and planning depth. These have an effect on the OLT/OLF. If StretchAreas are present, stretching is done proportionally within these StretchAreas.

The stretch areas must always bear the ID designation, e.g. SAH1, not SAH.

The stretch areas must always be continuous horizontally and vertically across OLT and OLF.

Only expansion is possible, no compression.

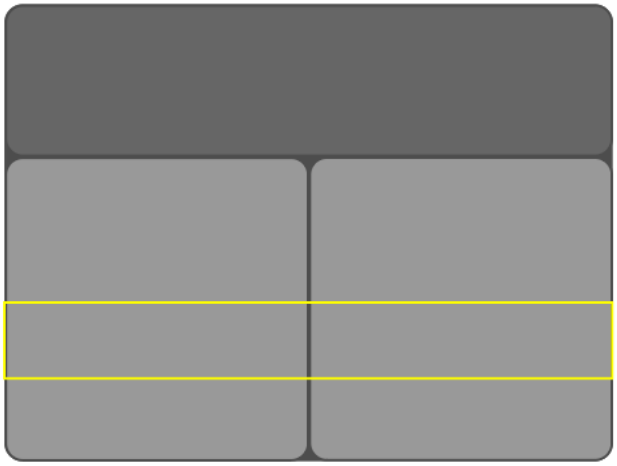
StrechAreas are formed as rectangles.

The expansion is proportional to its size.

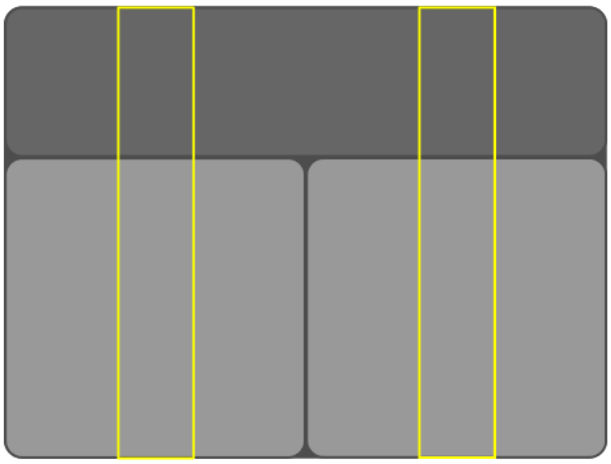
The stretch areas are divided into horizontal and vertical.

- SAH1 ( StretchAreaHorizontal 1 )
- SAH2 ( StretchAreaHorizontal 2 )
- .....
- SAV1 ( StretchAreaVertical 1 )
- SAV2 ( StretchAreaVertical 2 )
- .....

**Mapping/application of a vertical depth change SAV1 (vertical depth change over the seat surface, the depth of the back surface remains).**



**Illustration/application of a horizontal width change SAH1 / SAH2 (uniform, horizontal width change over the seat surfaces and backrest)**



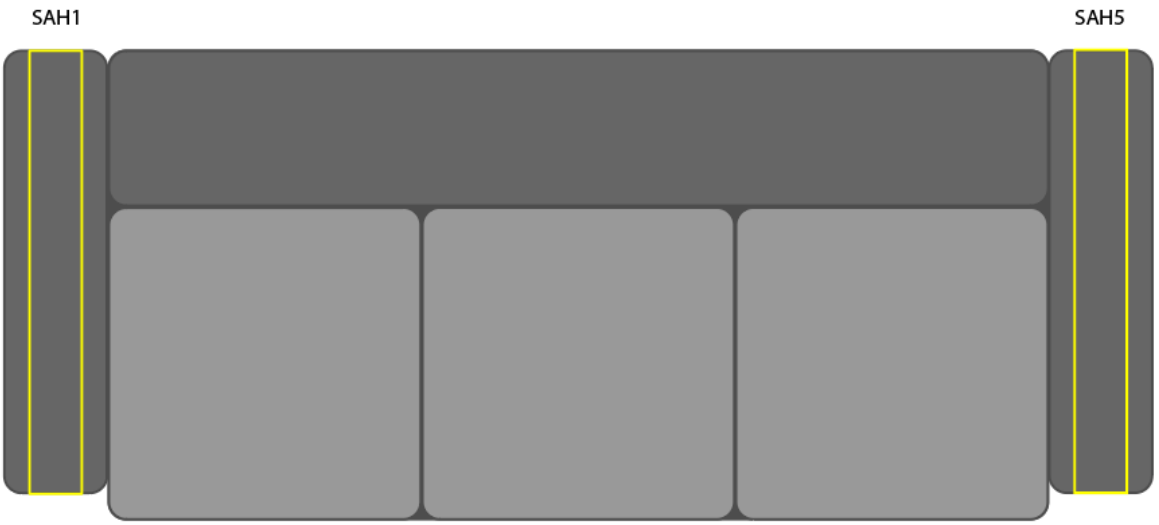
### 2.12.1. Examples for the use of StrechAreas

Illustration using the example of a 3-seater with armrests

1. **Example**

   Superimposed area: SAH1 and SAH5 on the armrests

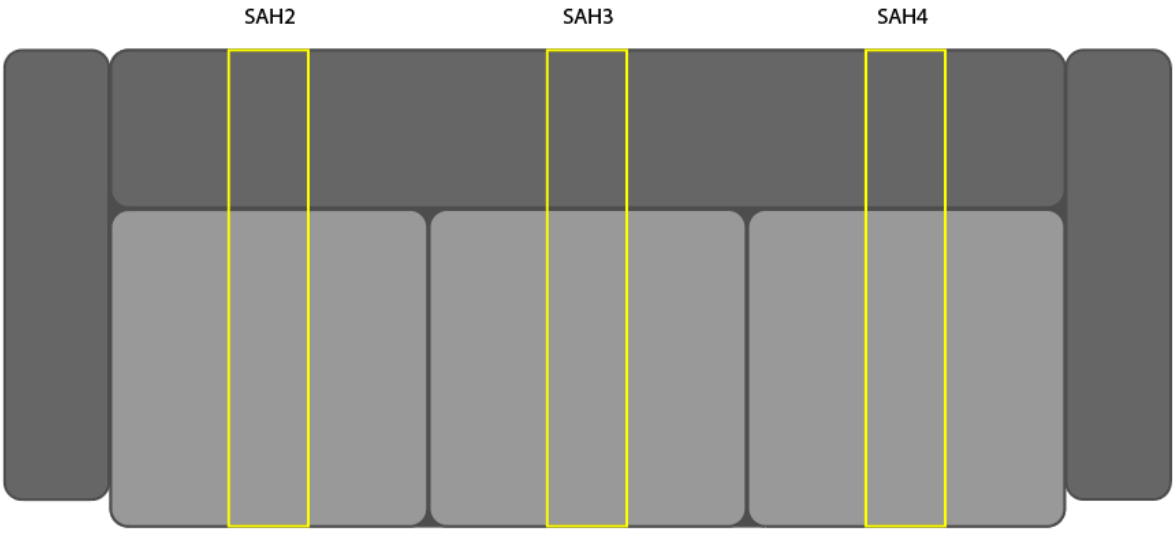
   In these areas the armrests are distorted evenly to the desired width



## 2. Example

Area shown: SAH2, SAH3 and SAH4

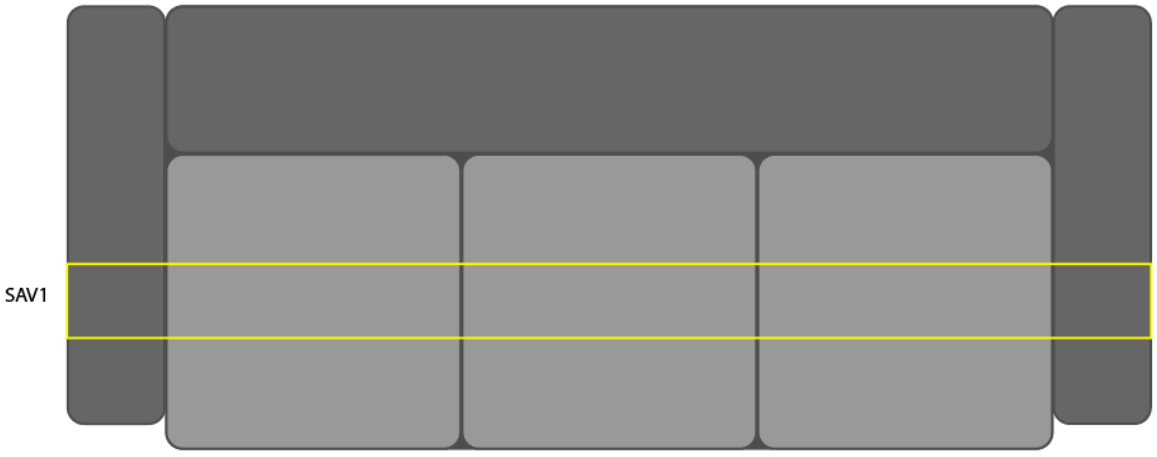
In these areas, the seat surfaces and backs are evenly distorted to the desired width.




## 3. Example

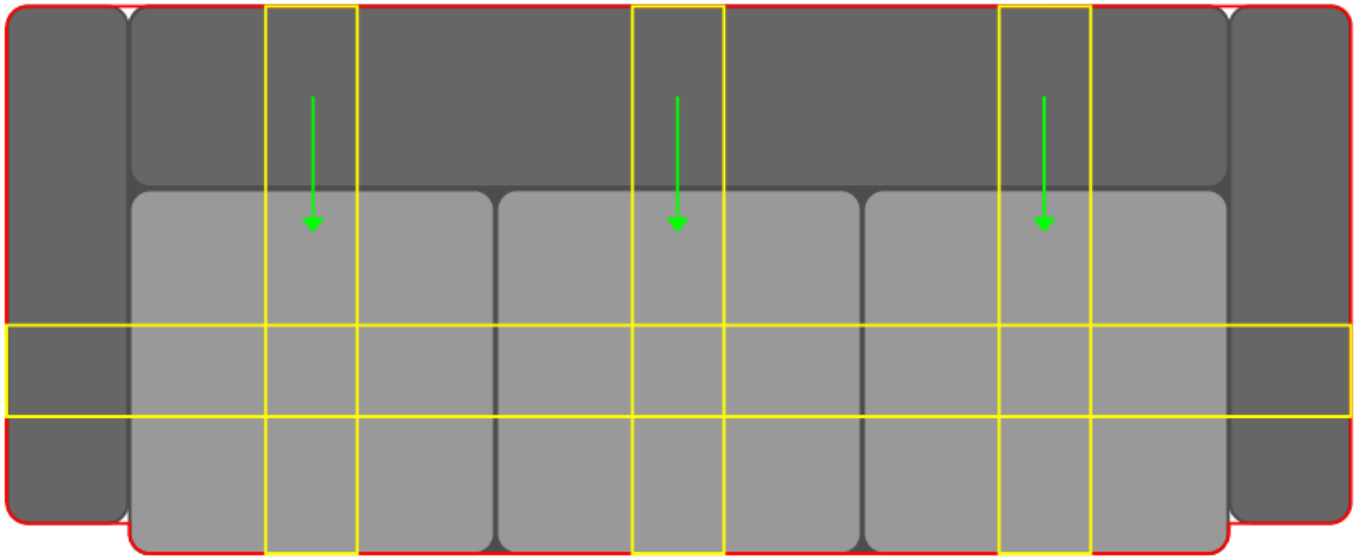
Superimposed area: SAV1
In this area, the seat surfaces and armrests are evenly distorted to the desired depth.

SAV1 on the seats



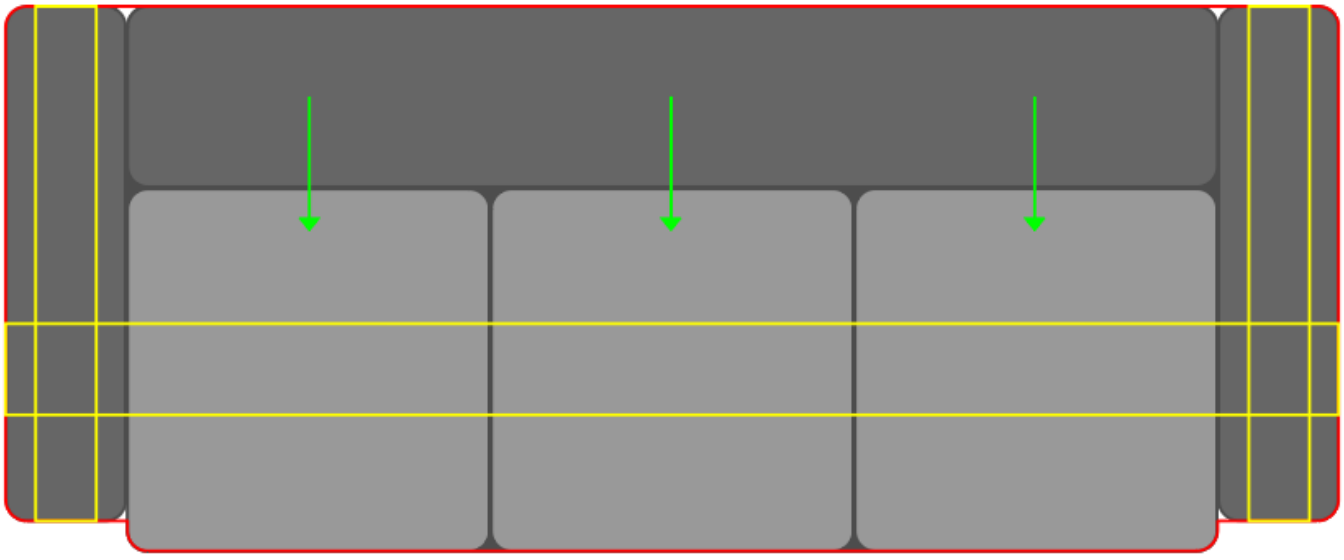
## 4. Example of stretch areas

3-seater sofa with stretch areas on the seats and headrest adjustment points*



## 5. Example of stretch areas

3-seater sofa with stretch areas on the armrests and headrest adjustment points*.



* The vectors are of course only drawn as a line, but in order to clarify the direction for the documentary, they are shown here as an arrow.

* The arrow symbol is only used to indicate the direction of the vector

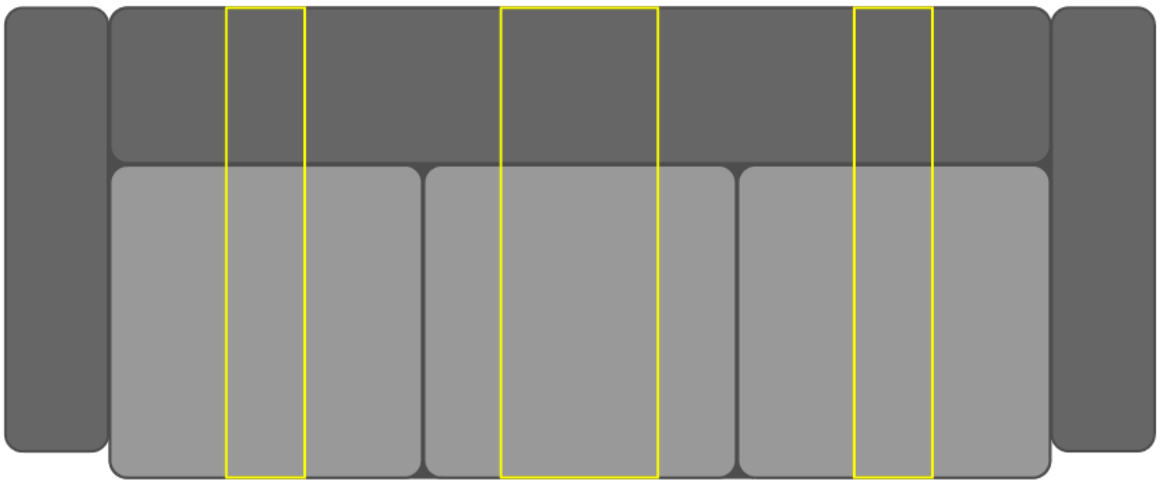
**6. IMPORTANT NOTICE:**

In the case of contiguous stretch surfaces, ensure that they are uniform in size, as they will be distorted in equal percentages.
z. Ex.
10 mm = 100%
11 mm = 110% (here the area is increased by 1 mm)

20 mm = 100%
22 mm = 110% (here the area is increased by 2 mm)

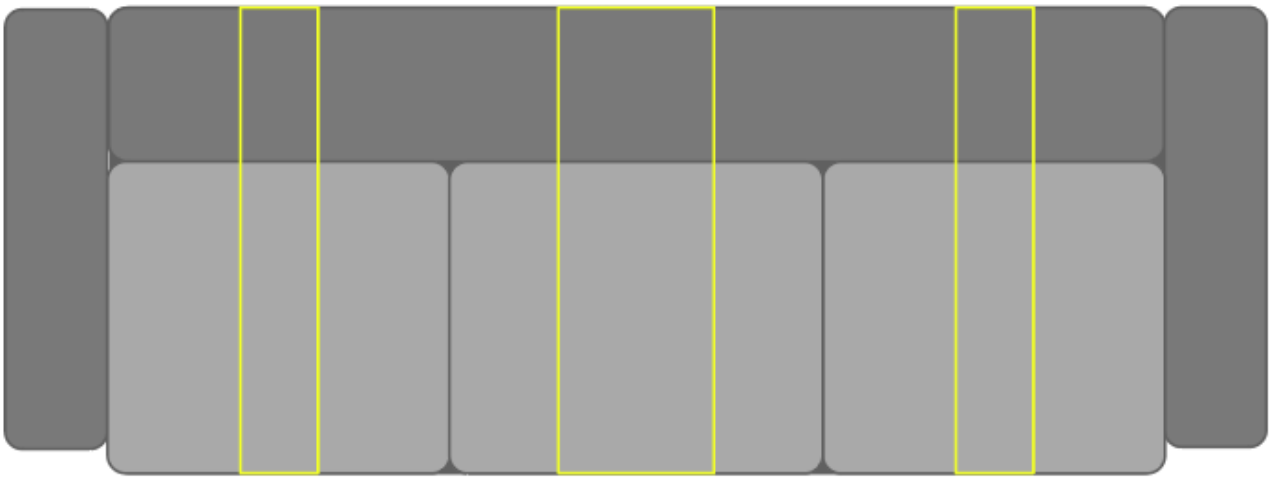
**Example of misrepresentation except explicitly requested:**

SAH2 = 10mm SAH3 = 20 mm SAH4 = 10 mm




If each stretch area is proportionally enlarged to 110%, the results are as follows:

Range SAH2 enlarged by 1 mm
SAH3 enlarged by 2 mm
SAH4 enlarged by 1 mm



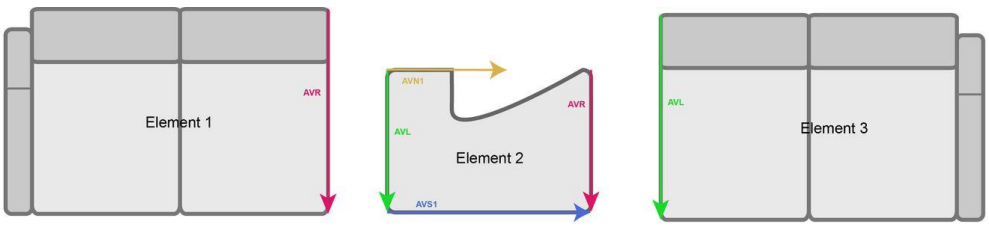
# 4. Connection logics

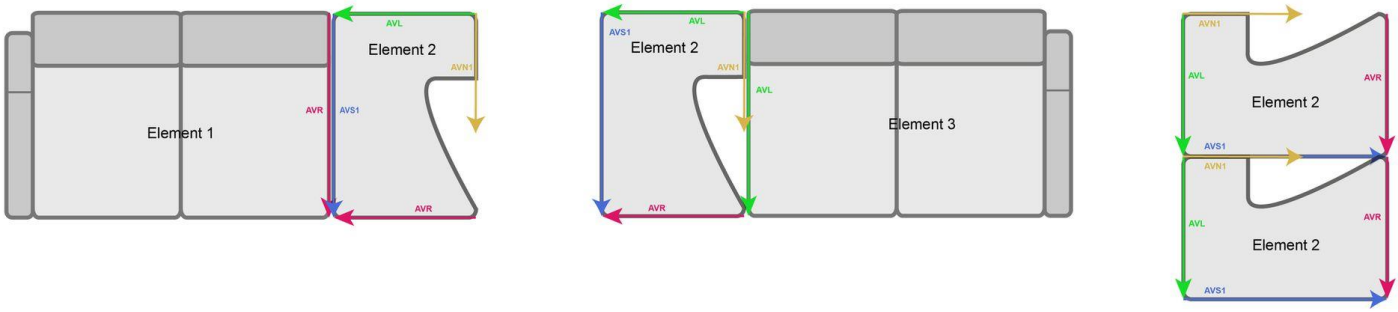
## 4.1. Connection rules

The following rules apply to a 2D magnetic planner as to which items may be planned or connected directly next to each other:

- Basically, the valid variants of the connection types are compared.
- If at least one matching, valid variant (same value) is included in two variant types, connection may be made according to the following rules:
  - Connection variant type L may only be attached to connection variant type R.
  - Connection variant type R may only be attached to connection variant type L.
  - Connection variant type Top may only be attached to connection variant type Bottom
  - Connection variant type Bottom may only be attached to connection variant type Top
  - Connection variant type North may only be attached to connection variant type South
  - Connection variant type South may only be attached to connection variant type North.
- A switch in the IDML data below SERIES enables additional setting options:
  - Default is: North may only connect to South / South may only connect to North.
  - When the switch is activated, North may also connect to L and South to R .

Without a valid set of connection variant types, the connection point should be hidden in the configurator.



## 4.2. Connection behaviour

Which article can “snap” to which is stored in the IDM data on the item using rules (decision tables) that relate to the connection vectors. In doing so, it is determined which possible connection variants the item has. Depending on whether or not the item can have a left or a right connection, a number of possible right or left connection variants is defined.

If an item has a connection variant type, but no rules for its connection options (connection variants), then no connection options are excluded for this connection variant type. This means that all elements with the appropriate connection variant type can be connected.

In principle, commercial data are placed above the SVG data. The exception are SVGs, which offer less than the commercial data. In such a case, the item only offers contact points that are available in the respective SVG.

Each item should know if it matches with another item. For this purpose, information about its connection behaviour is given via the IDM format. Each attachable item should be given information on each connection side about which attachment variants it can accommodate (right or left, top or bottom, north or south).

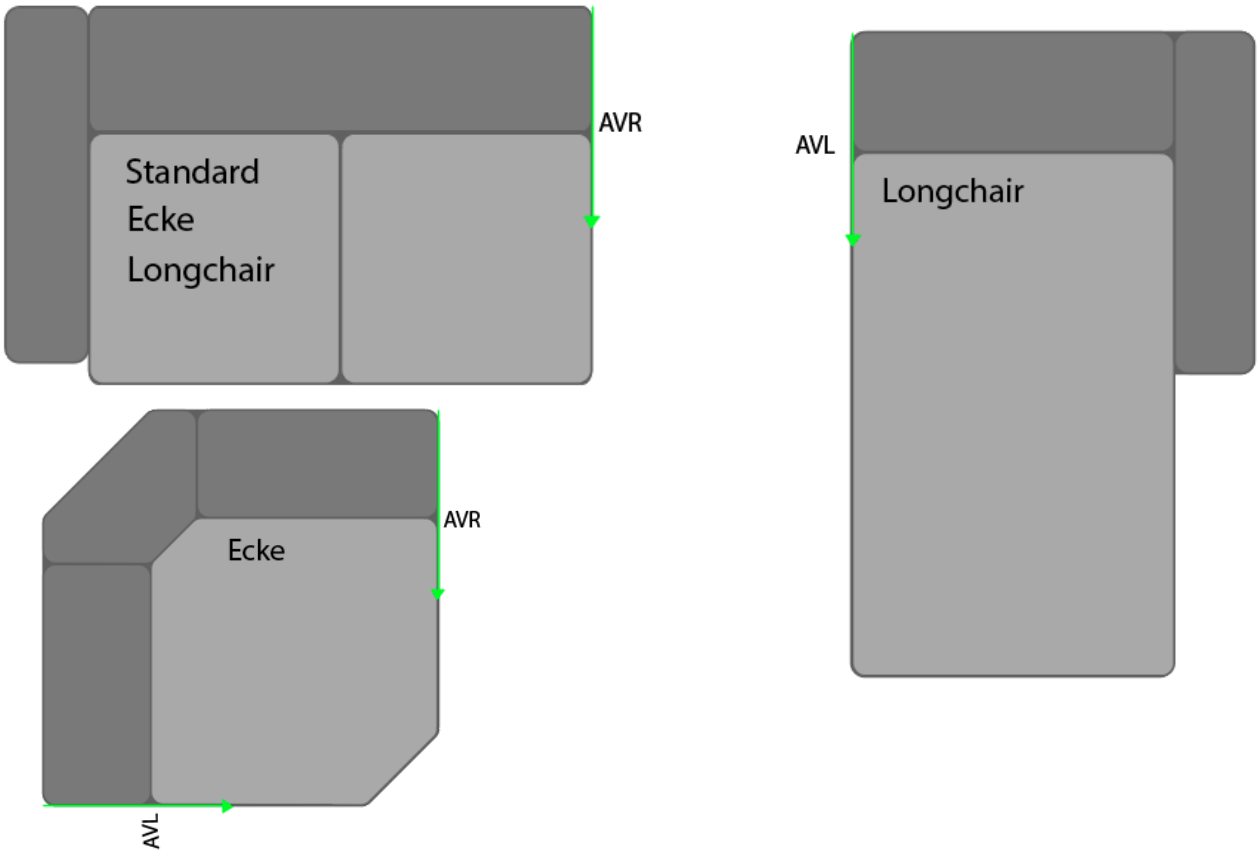
### 4.2.1. Example of the connection behaviour

A connecting sofa with armrest on the left has the connection feature AVR and the connection options "Standard", "Corner" and "Long chair". The long chair with armrest on the right has the connection variant type AVL and the "long chair" connection option. The two elements can be connected because they have the same connection option and the mutual connection variant types AVL and AVR according to the connection rules.

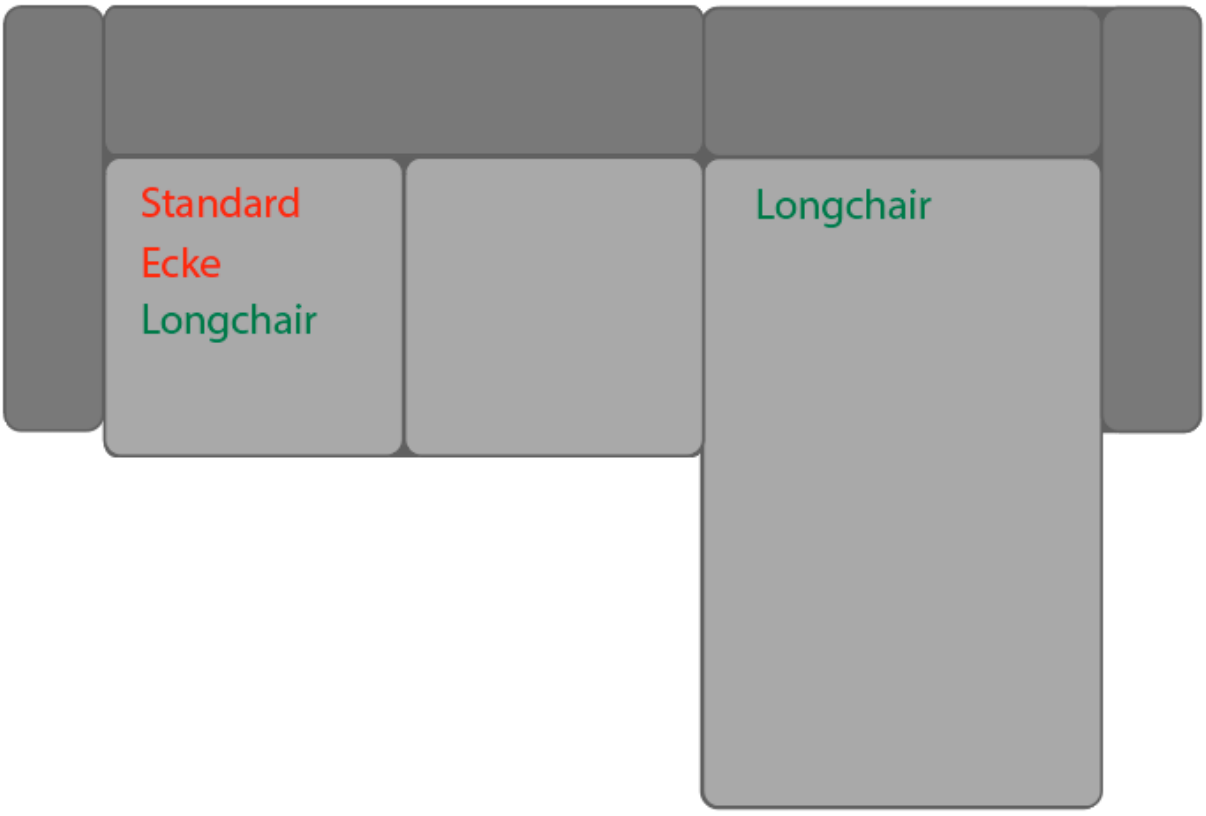
A pointed corner item with connection variants AVL and AVR and the connection option "corner" can therefore be connected to the connecting sofa but not to the long chair.

These rules about connection variant types can also be processed using IDMP decision tables (rules in the IDMP data) if the special connection variant types and the left or right connection variant are defined and these are assigned to the item using rules.

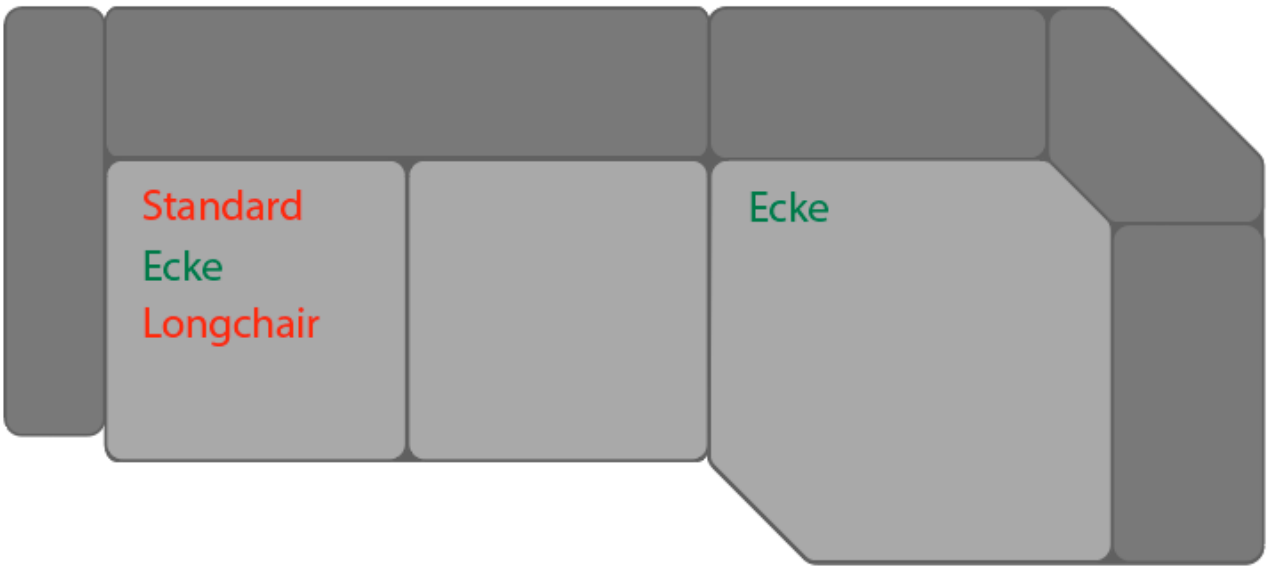
**SVG's with connection options:**



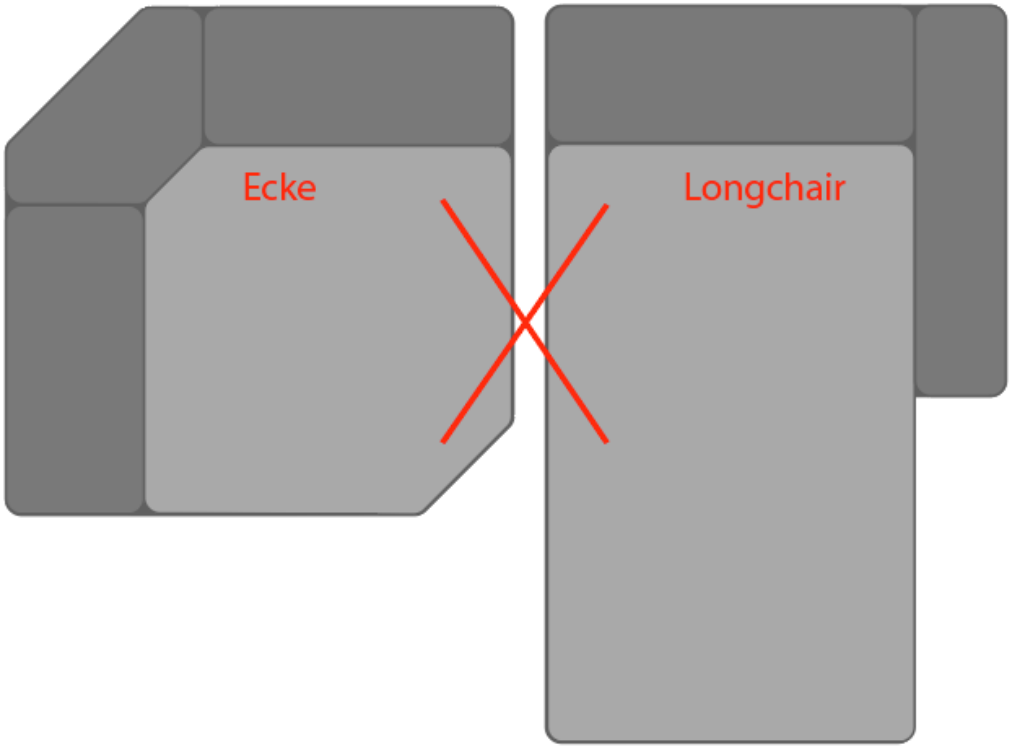

**Permitted connection: Connection sofa to Longchar:**



**Permitted connection: connecting sofa to corner:**




**Unauthorised connection: corner to long chair:**



### 4.2.2. Extended connection logic:

Since seat height and the seat depth are processed using options, rules can also be used to assign a great number of connection features. The number of valid connection features thus results from a result of rules.

### 4.2.3. Example:

Depending on an item with a seat height of 45, the left connection option “Standard 45” is assigned. Only items with a right connection option also named „Standard 45" can then be connected.

In order to be able to connect 2 items, they must have the same connection option and the respective matching connection features.

The feature class specifies the connection vectors. If an SVG has a connection vector that does not appear in the feature class, the connection vector is hidden.
If an item supports connection vectors, these must also be included in the feature class assigned to the item.

The data creator has to ensure that SVG data is compliant with commercial data.
The connection features of the feature class should be checked against all SVG files of an item and each connection must occur at least once in at least one of the assigned SVGs. Additional data in SVGs, that are not included in the commercial data, are ignored.

# 5. Planning rules / collision checks

## 5.1. General

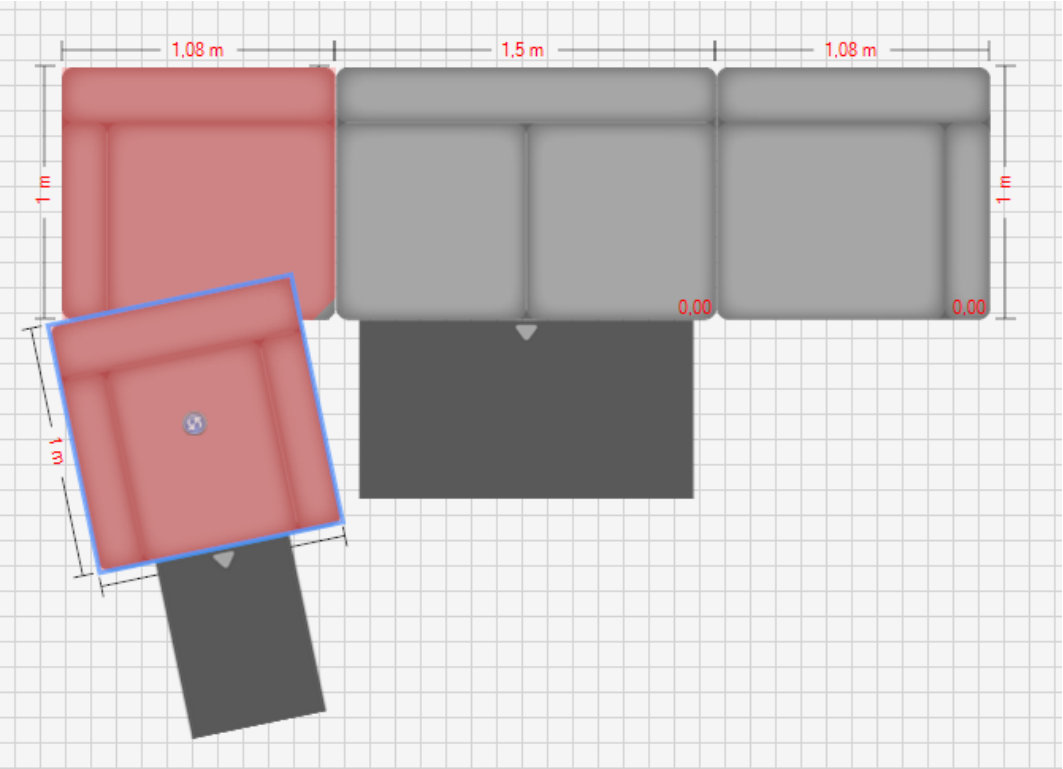
The basic placement rules in the magnet planner are:

- OLT areas must not overlap within a layer.
- OLT and OLF areas must not overlap.
- OLF areas are allowed to overlap with a note for the planner.
- The technical level can be bevelled to prevent levels from colliding. Consequently OLTs are bevelled.
- If there is a stretch area, the data is based on the commercial data.
- The stretch areas must always be continuous horizontally and vertically.
- Only expansion is possible, no compression.

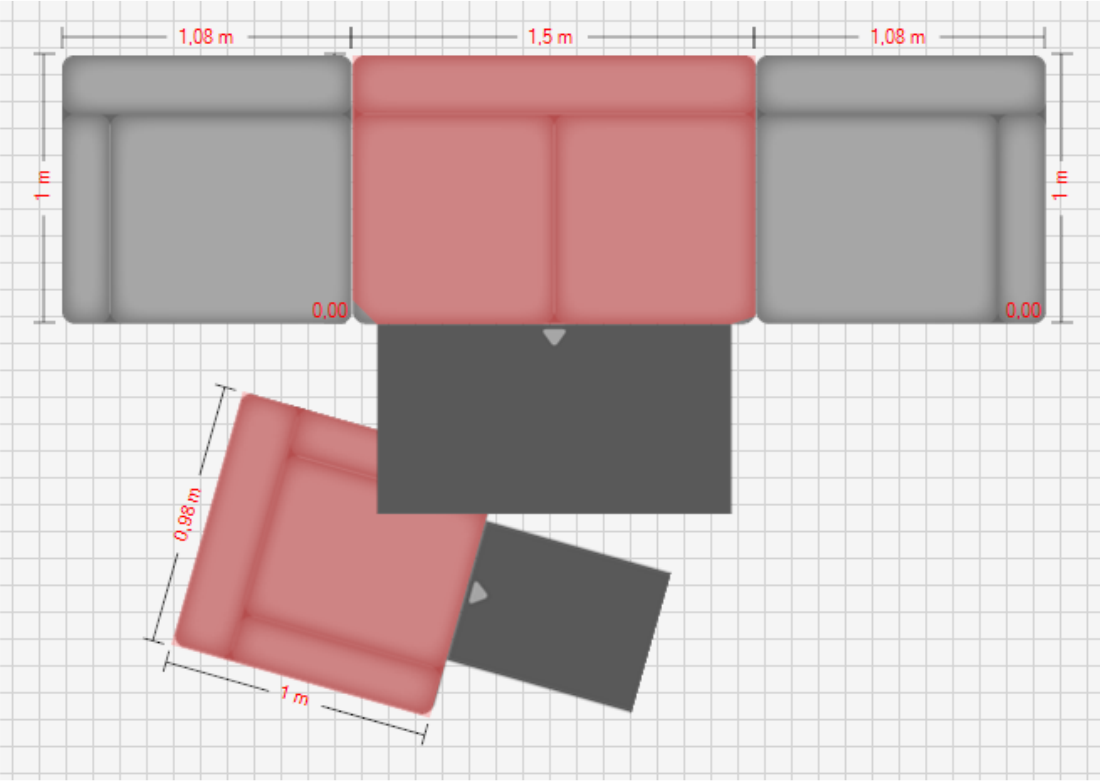
The following are examples:

## 5.2. Overlap

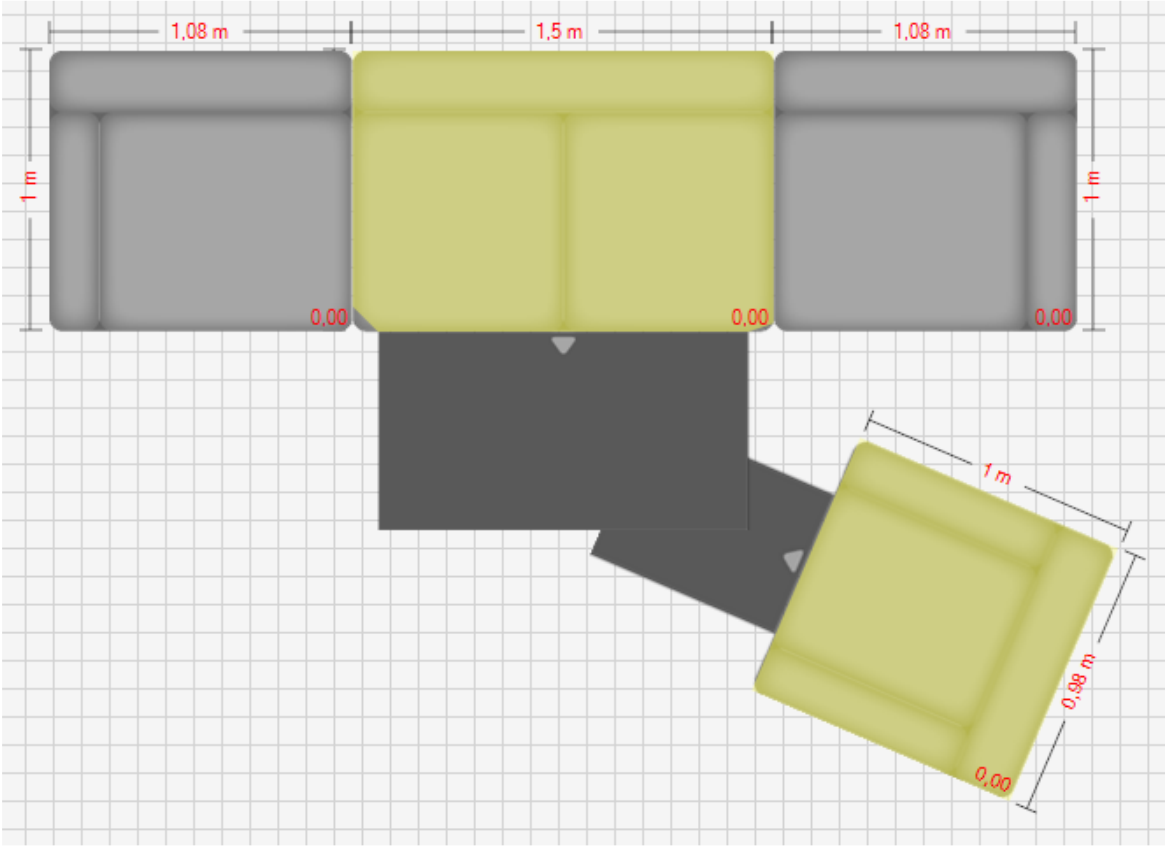
Items that overlap at the same index generate errors.



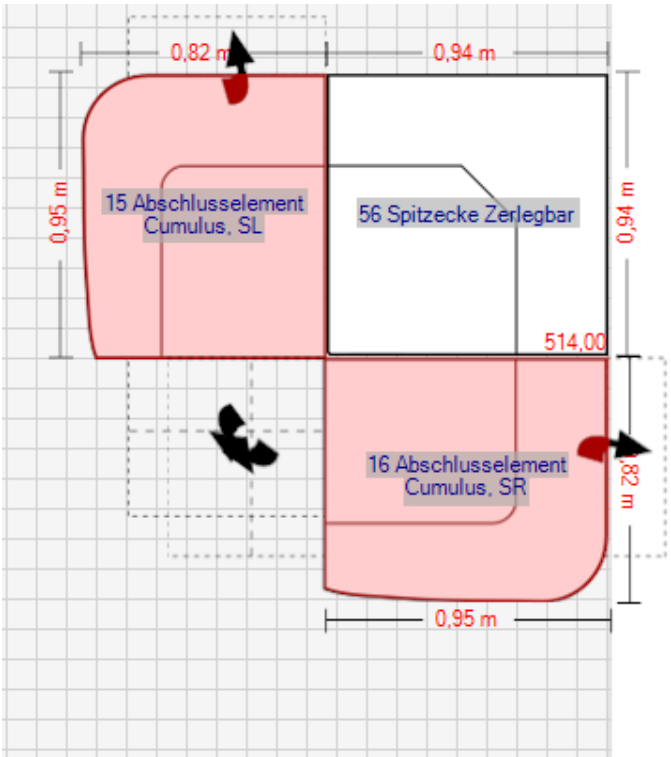

Articles that overlap with function surfaces at the same index generate errors.

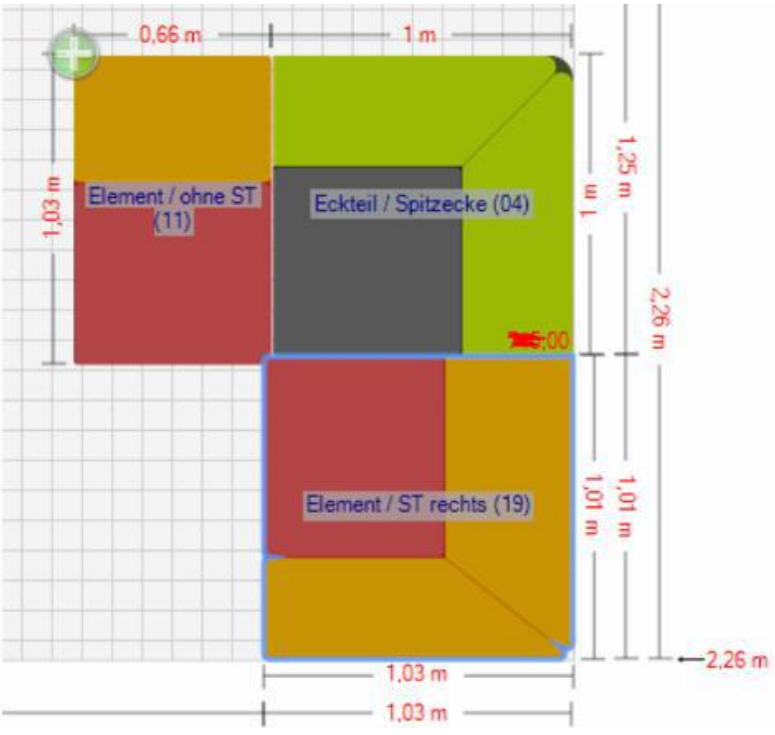


Function surfaces of items that overlap with other function surfaces at the same index generate warnings.




There must be no overlap at OLT:



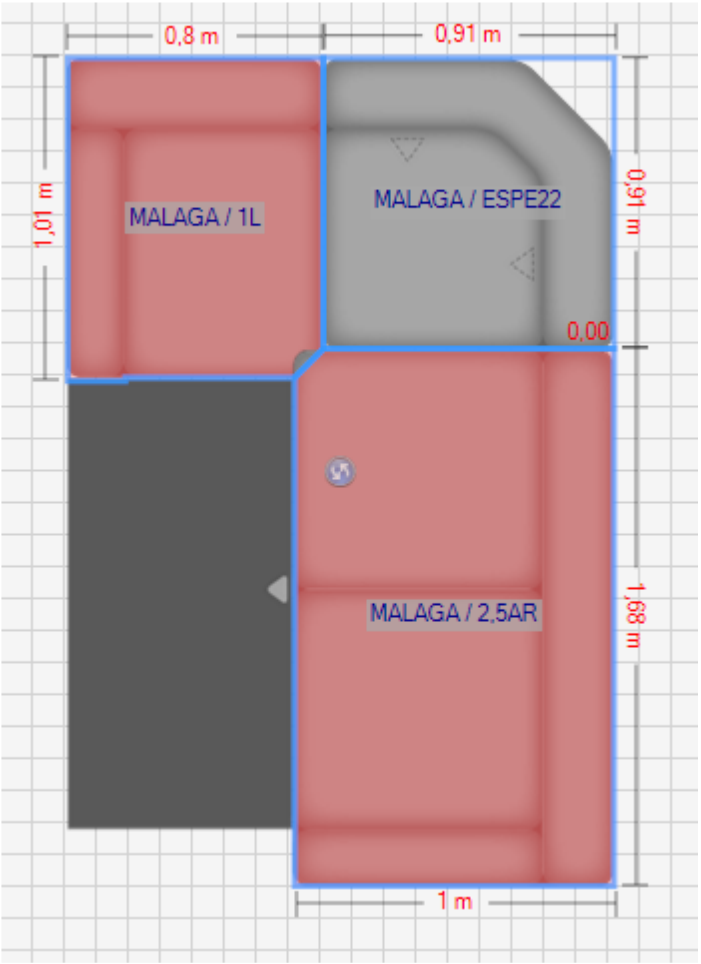



In principle, a composition where two OLT levels overlap should not be valid. To avoid this error, intermediate and final elements have a bevelled OLT.

OLT must not overlap.

The visualisation level is not affected by this.

The same applies to functional areas, here too there is no tolerance for overlaps:

# Contact

**Data Competence Center e. V.**

Goebenstraße 4-10

32052 Herford

Info: www.dcc-moebel.org

Dr.-Ing. Olaf Plümer

E-mail: pluemer@dcc-moebel.org

Tel.: +49 52 21 / 12 65-37

Anika Degenhard

E-mail: degenhard@dcc-moebel.org

Tel.: +49 52 21 / 12 65 - 38