

# HOUSCAPER

Participatory design tool  
for easy access of architectural polygonization  
using Boolean Marching Cube Algorithm

Delft University of Technology  
Faculty of Architecture and the Built Environment  
Master track Building Technology

**Educational supervisors**

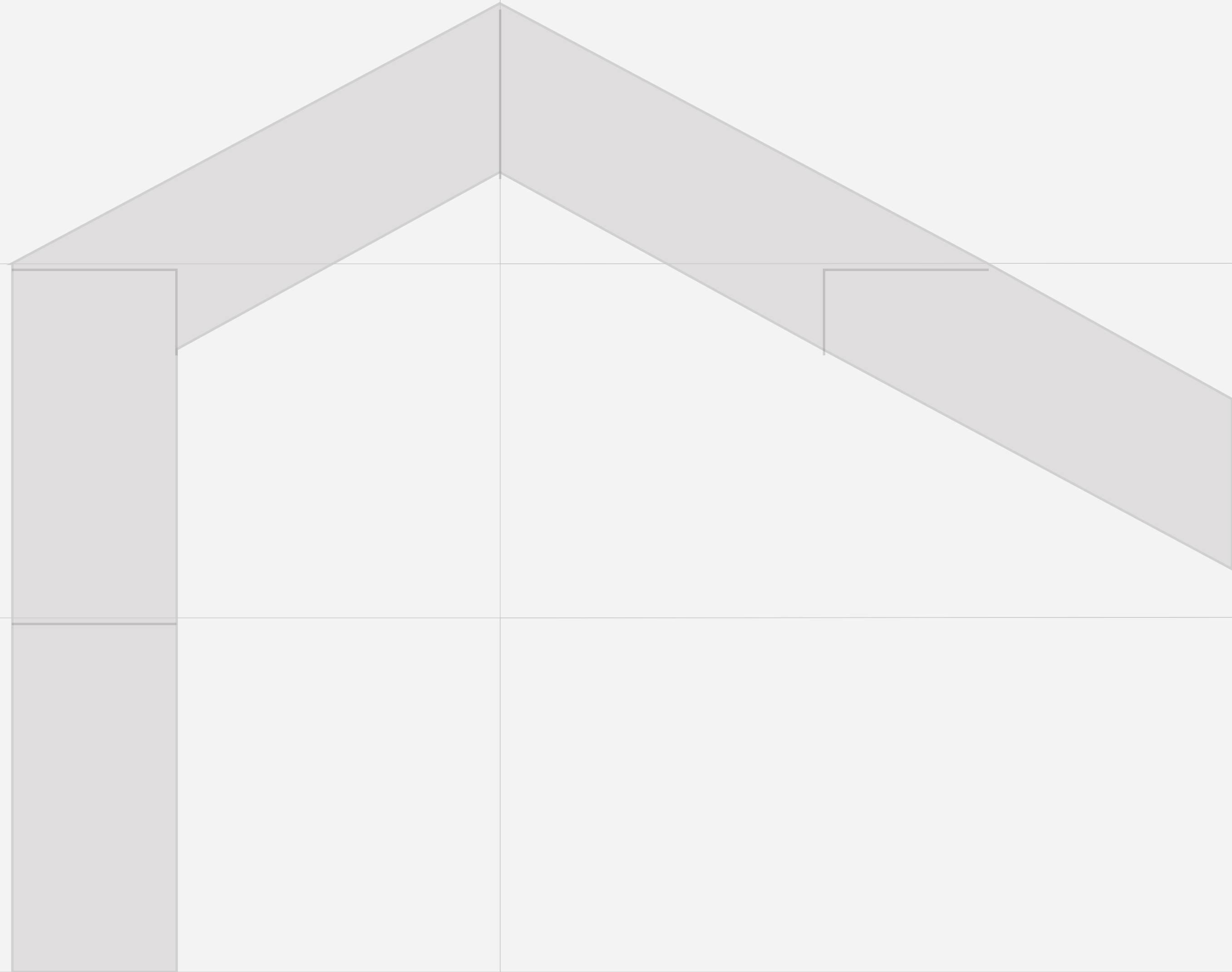
Dr. Ir. Pirouz Nourian  
Dr. Ir. Fred Veer  
Ir. Shervin Azadi

**Board of Examiners**

Delegate examiner: Ir. Geert Coumans

**Student**

Solkyu Park | 5304857



# CONTENT

## Research



Why  
How  
What

1

## Literature Review



Element Types of  
Modular construction  
Computational  
Algorithms

2

## Design Developments



Presets  
Surface tilesets  
Architectural tilesets

3

## Algorithm Developments



Cube ID  
Cube Options  
Cube Placing  
Design rules

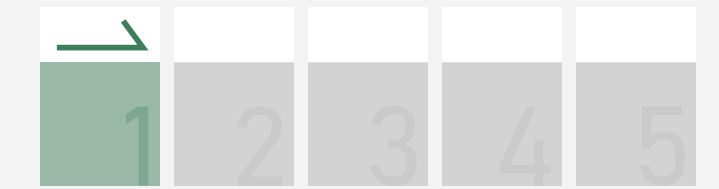
4

## Results

Results  
Structural analysis  
Conclusion  
Future works

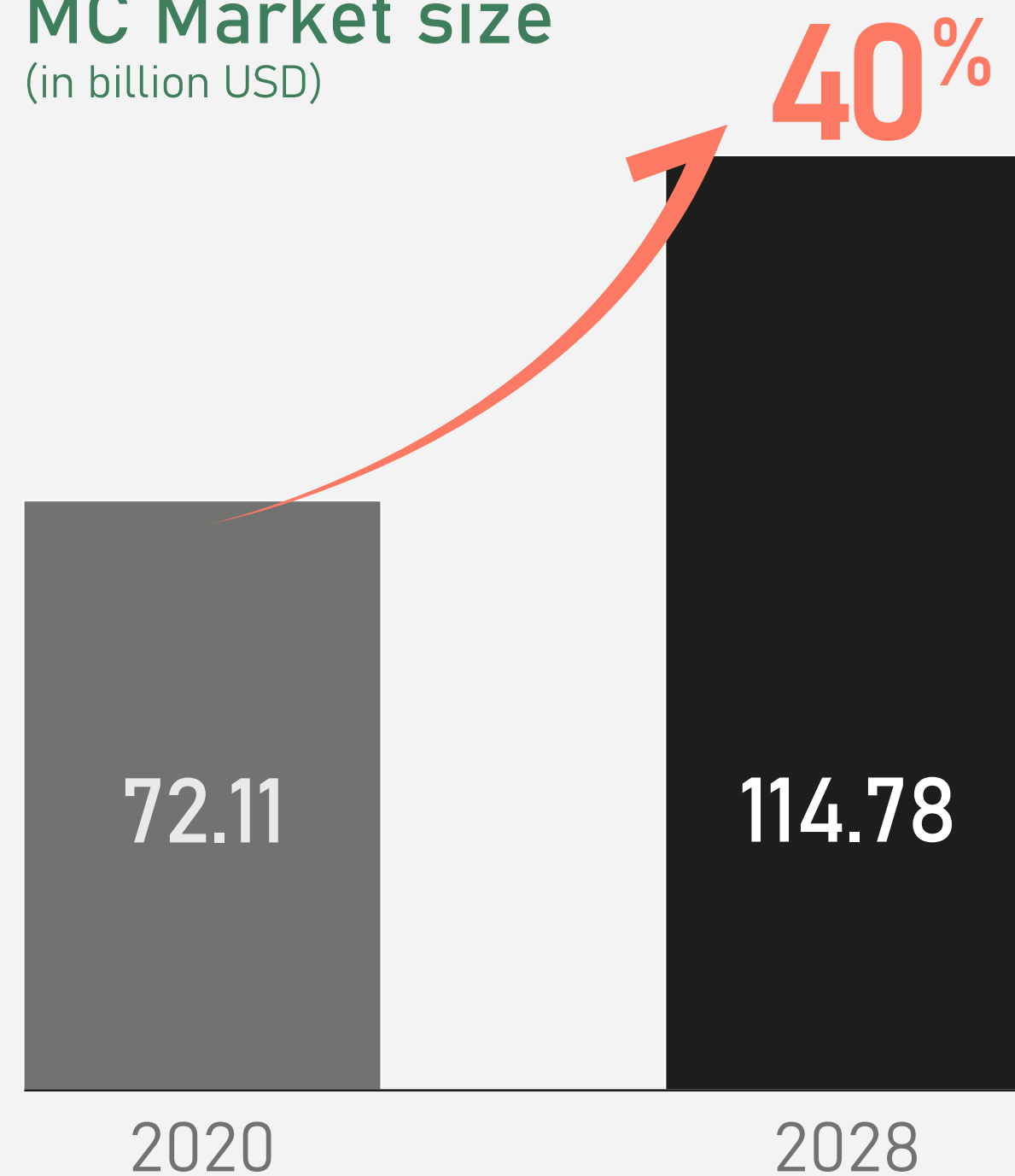
5

# BACKGROUND



The global market for modular construction is expected to grow rapidly the growth puts tremendous pressure on the construction industry to shift into modular construction in a short amount of time.

MC Market size  
(in billion USD)



## Driving factors of increase in MC



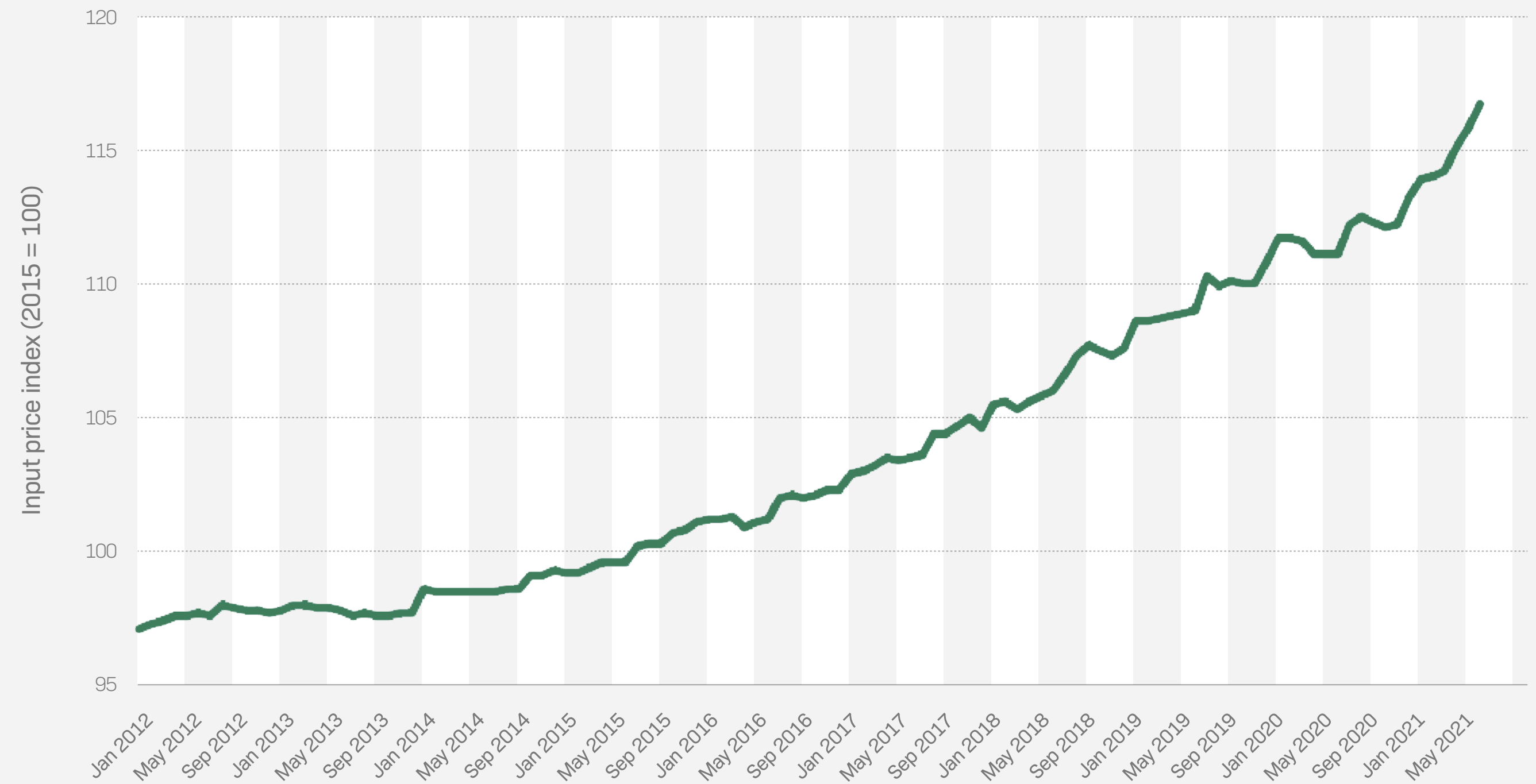
# WHY?

*Problem statement*

Conventional architectural design and construction **bespoke solution** from a **bespoke process**.

**Making the architecture the most expensive purchase in life.**

Input price index of building costs of new dwellings in the Netherlands (January 2012 - June 2021)



\*Source: Monthly price index for new home construction in the Netherlands 2012-2021  
Published by Statista Research Department, Jan 19, 2022

# WHY?

*Problem statement*

Modular construction optimize for  
**mass production** - cost of construction

The demand for lower cost  
builds the misperception

**Modular construction is repetitive**



# HOW?

*Research question*

How to devise a **participatory design tool**  
for constructing valid **modular building frames/cladding**  
using **computational methods**?

# WHAT?

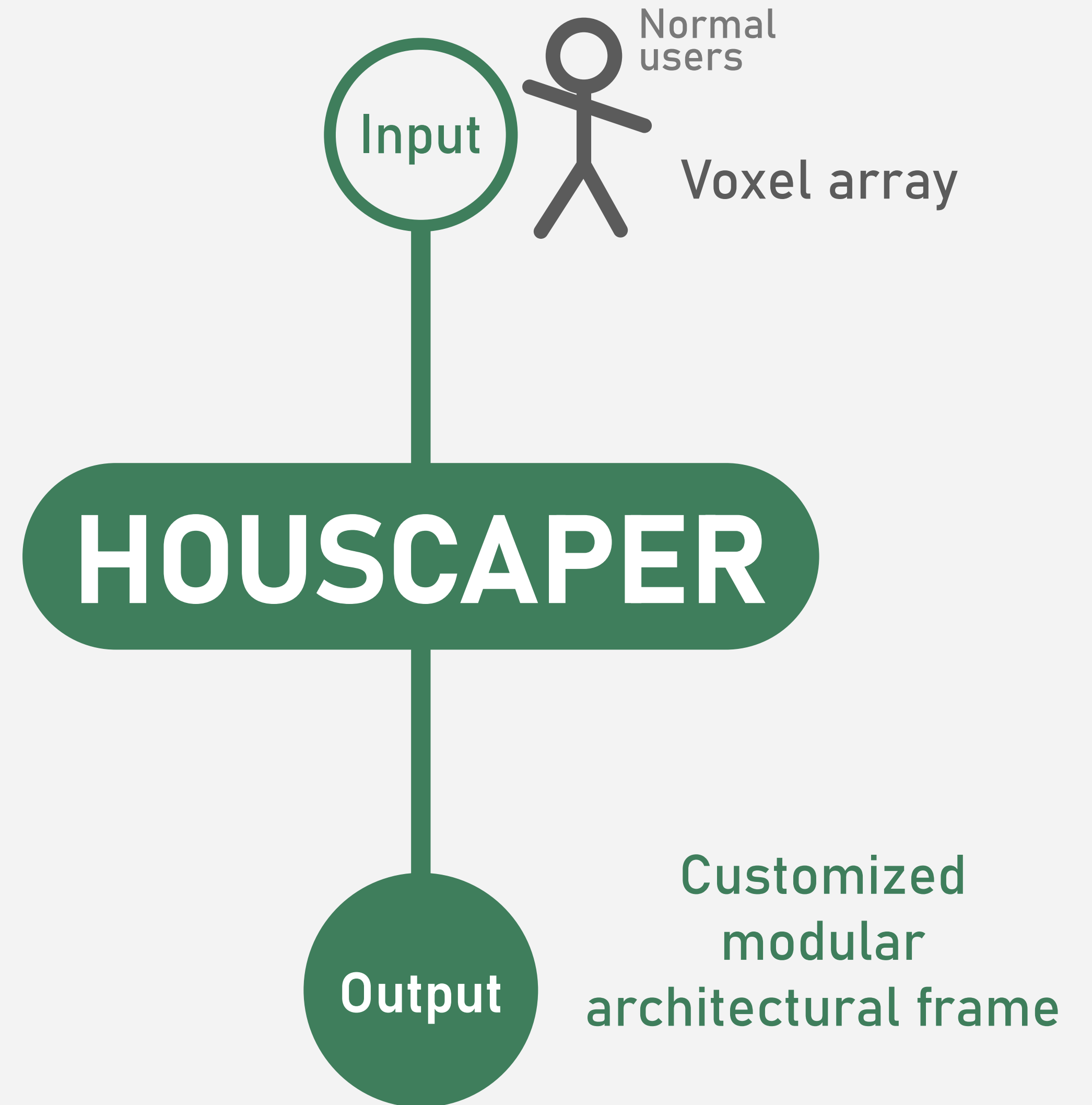
*Research objectives*

1

Develop an Interactive tool for **normal users** to **customize** their modular architecture from the input voxel array.

Offer a set of architectural tiles that incorporates mass customization and enables **expert users** to develop the tileset.

Offer a **structural validity check** of the frame generated.



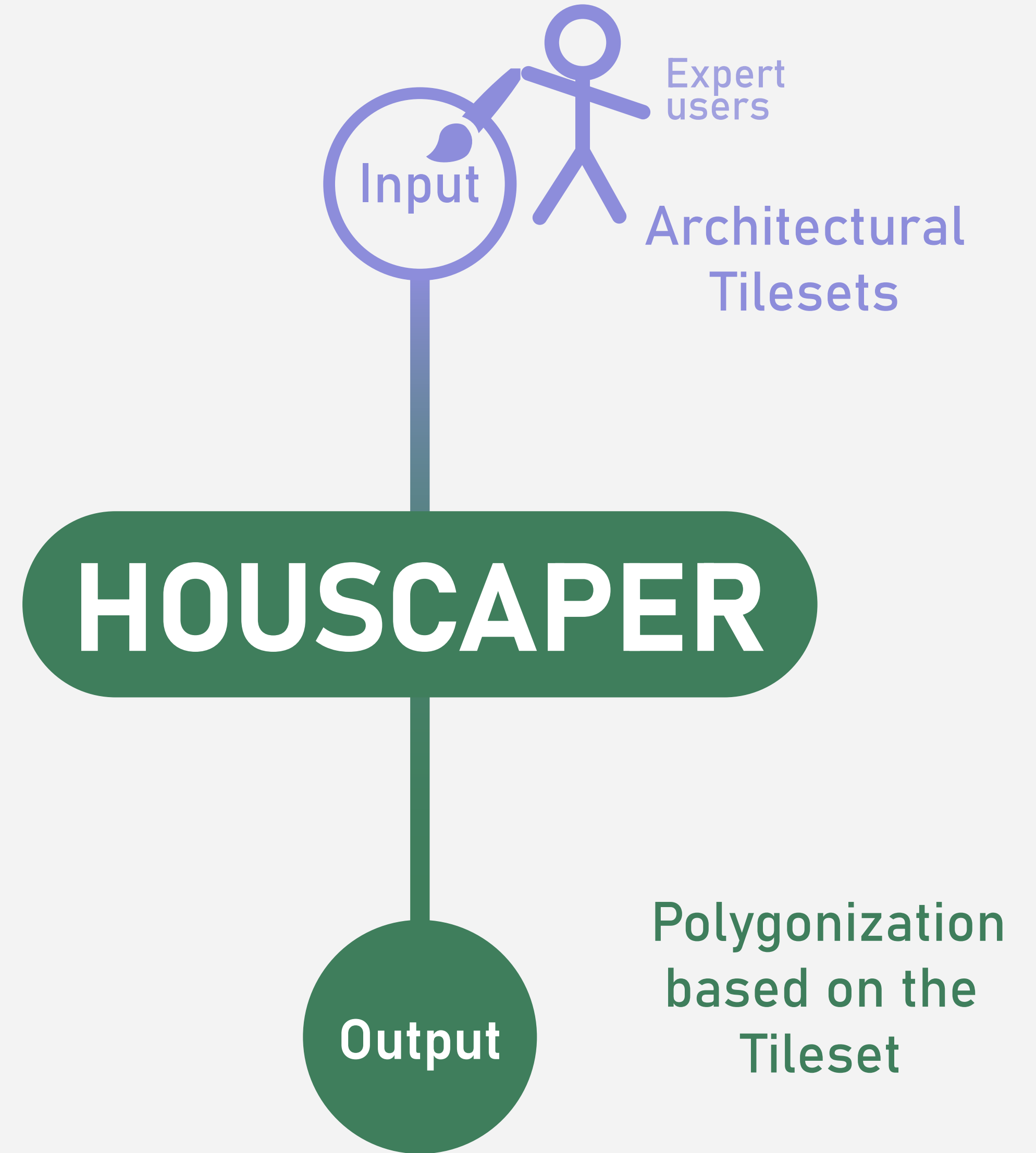
# WHAT?

*Research objectives*

Develop an Interactive tool for **normal users** to **customize** their modular architecture from the input voxel array.

**2** Offer a set of architectural tiles that incorporates mass customization and enables **expert users** to develop the tileset.

Offer a **structural validity check** of the frame generated.





# WHAT?

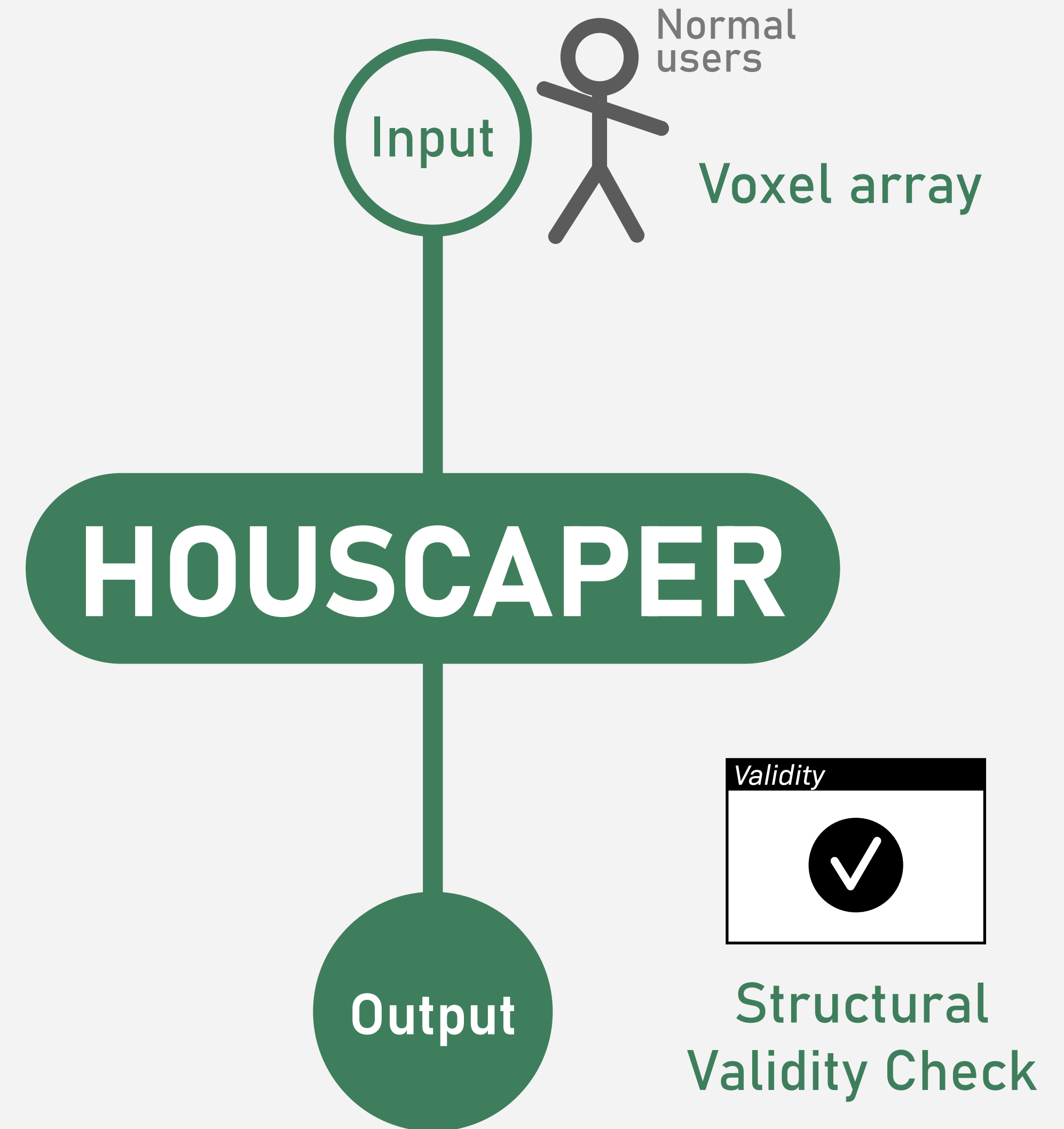
## Research objectives

Develop an Interactive tool for **normal users** to **customize** their modular architecture from the input voxel array.

Offer a set of architectural tiles that incorporates mass customization and enables **expert users** to develop the tileset.

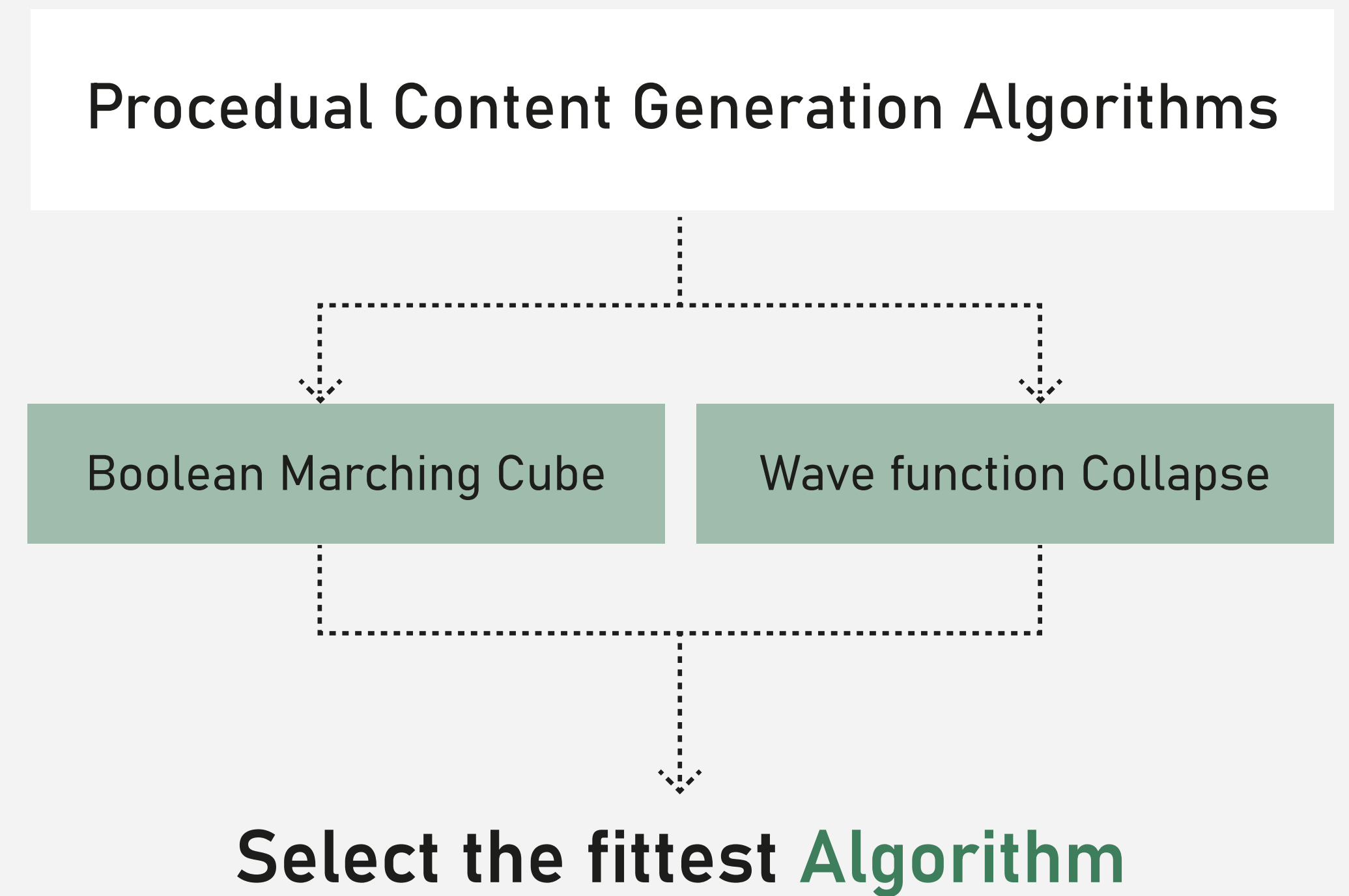
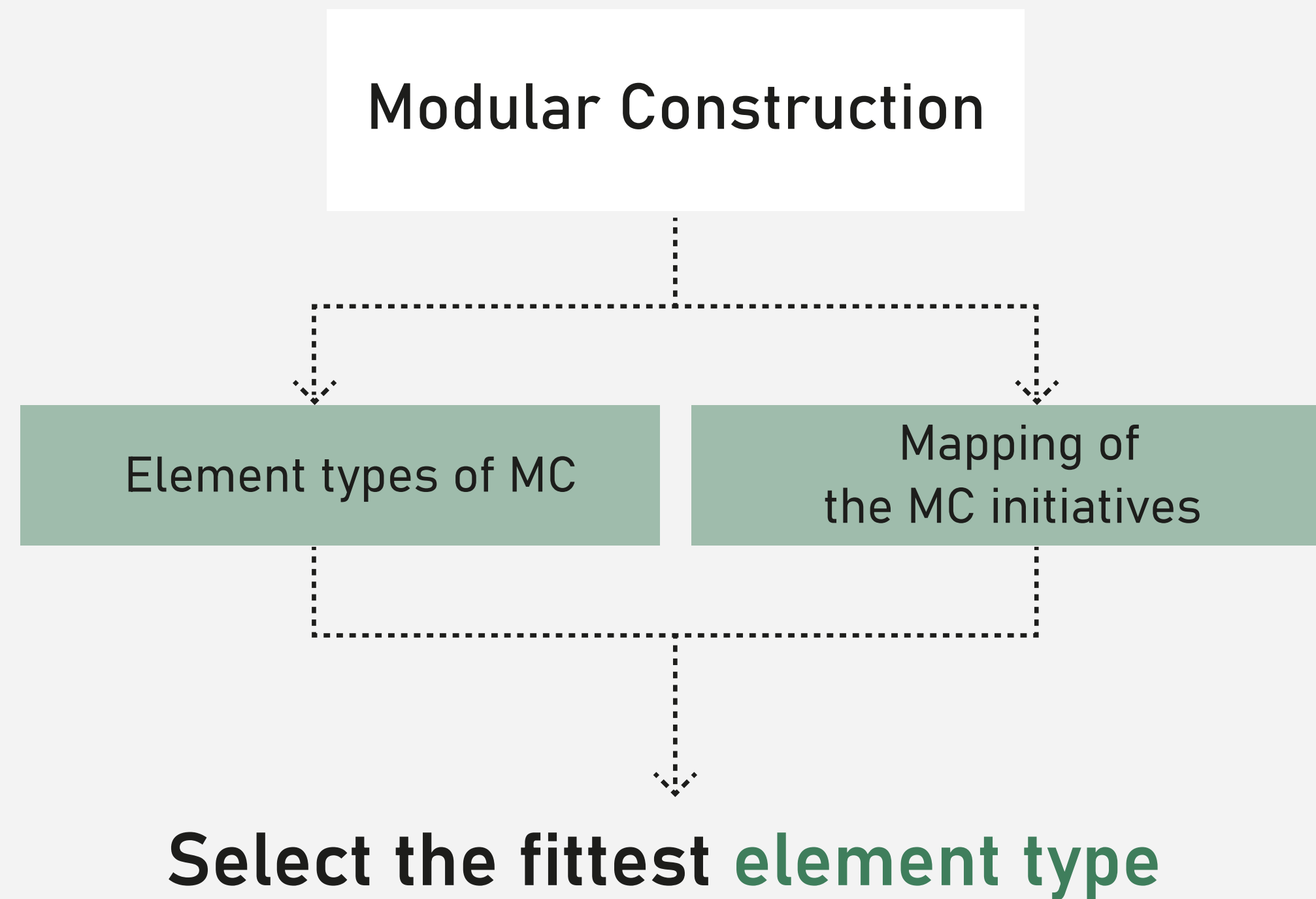
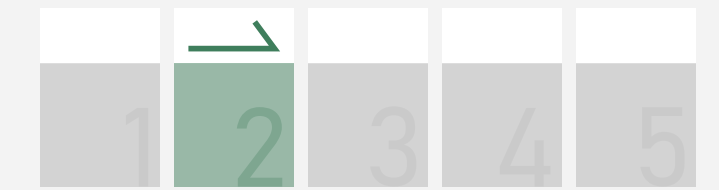
3

Offer a **structural validity check** of the frame generated.



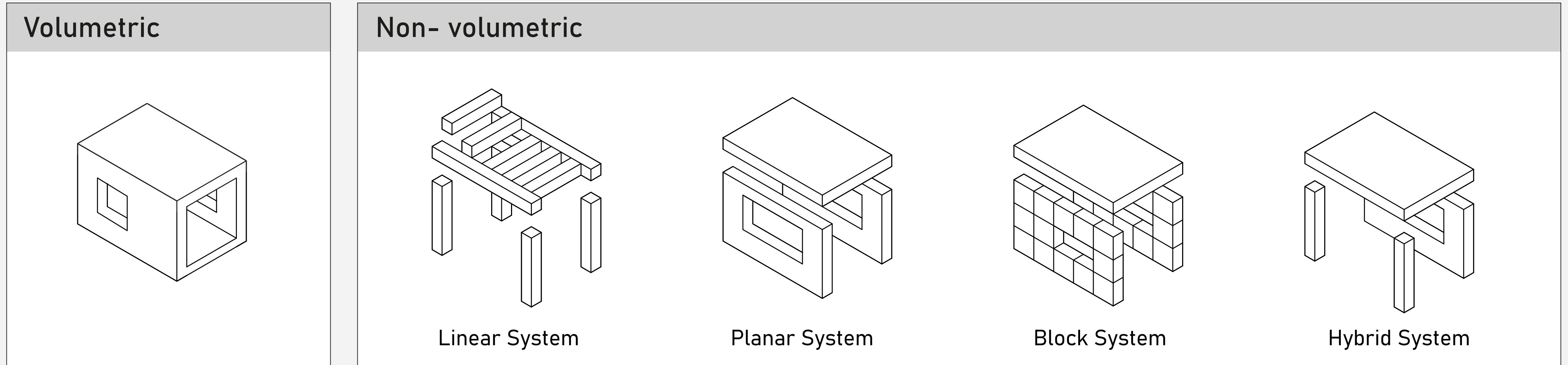
**Video**

# LITERATURE REVIEW



# LITERATURE REVIEW

## Element Types of Modular construction



- Finished in the factory
- Minimal on-site installation
- No need for scaffolding, and lifts



- The size are limited by the load and transportation
- Manufacturing facility significant investment

- Easy to connect
- Widely adapted

- Additional parts, process required

- Less space required volumetric system
- Finished in the factory
- Less space required than volumetric

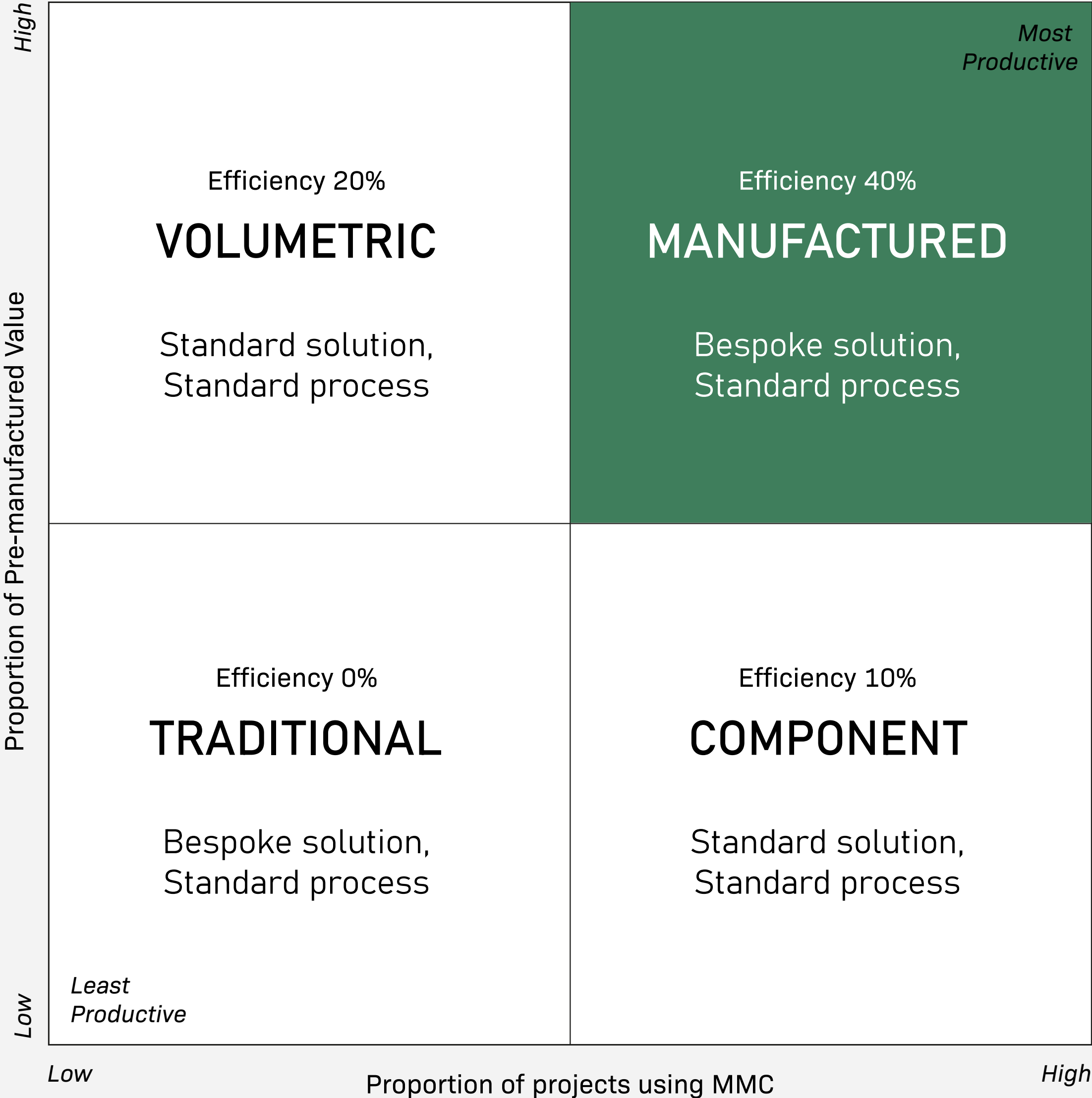
- Prefabricated with insulation
- Lifiable by one person

- More work on site

- Mixing of systems can be cheaper to 3%

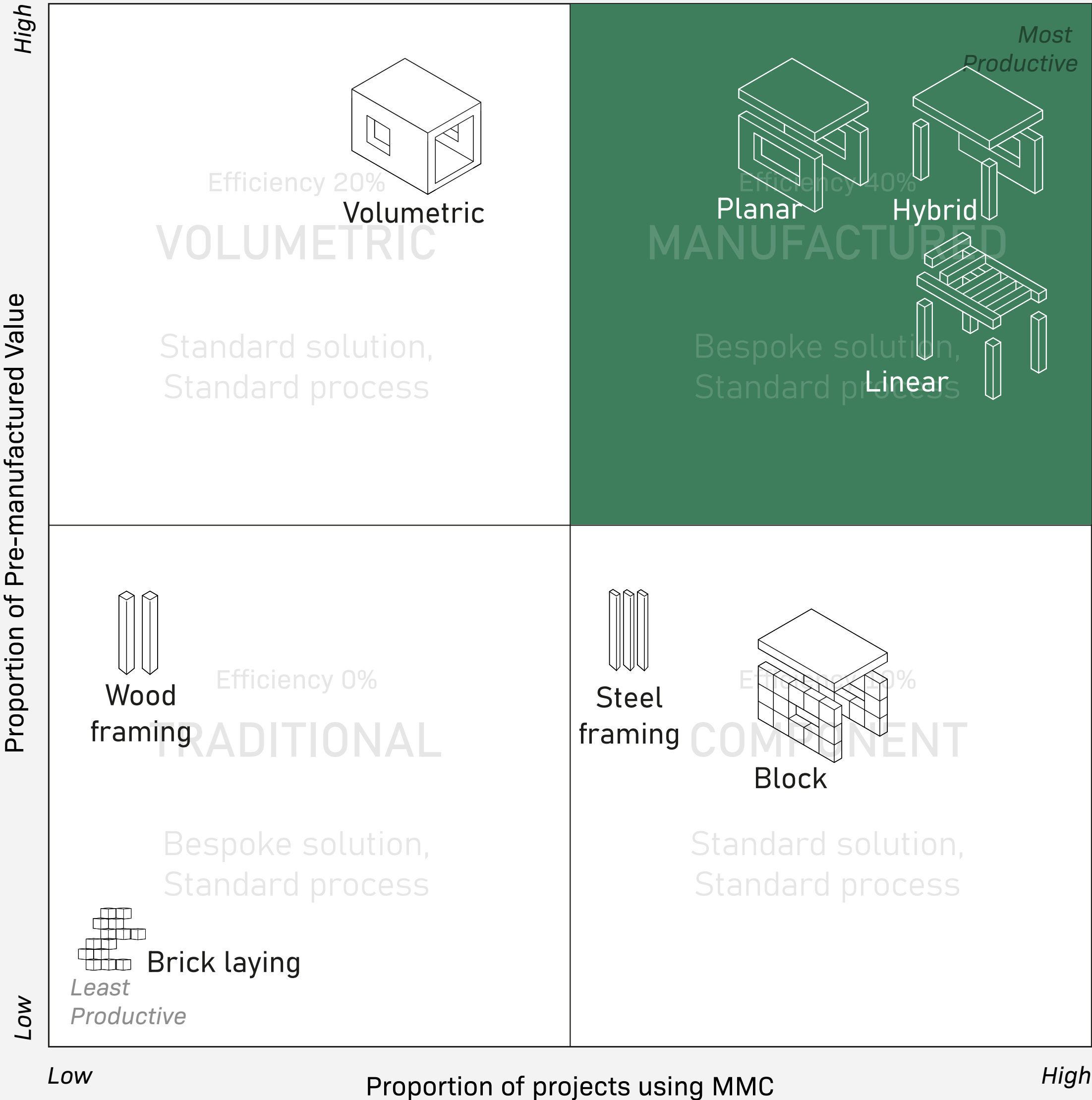
# LITERATURE REVIEW

*Mapping modern methods of construction (MMC) initiative*



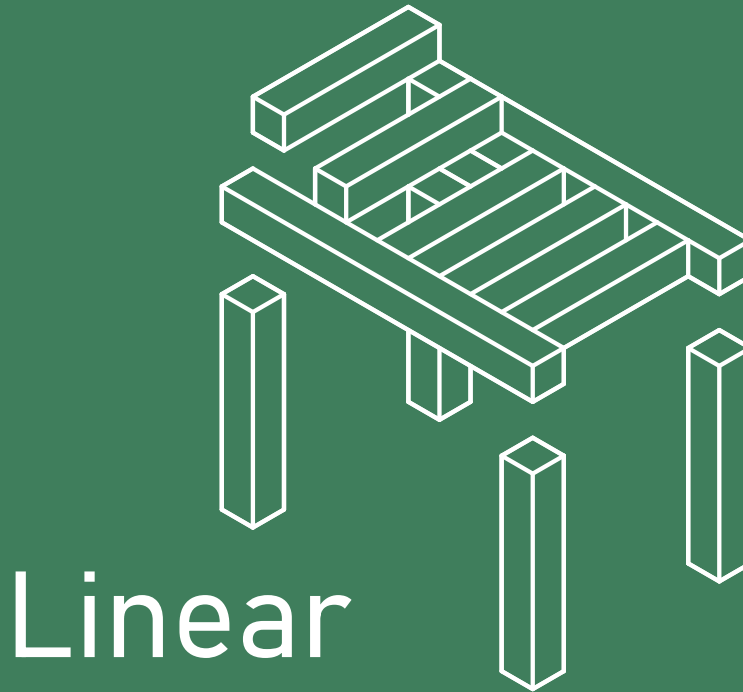
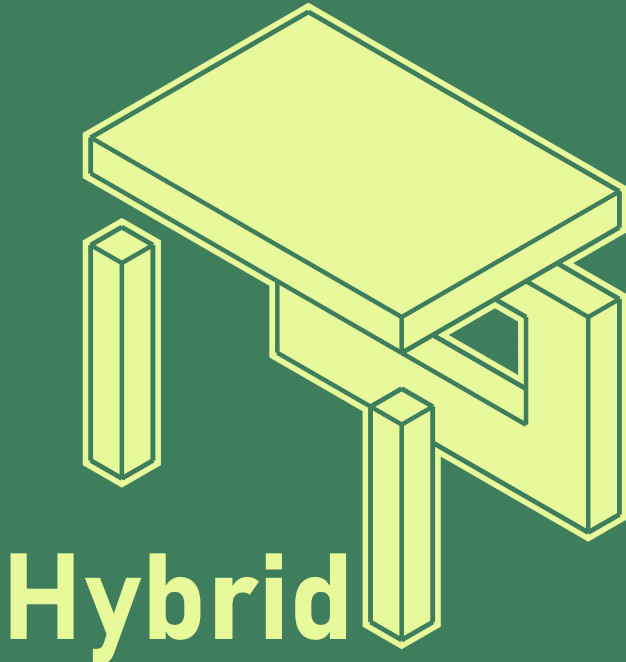
# LITERATURE REVIEW

Mapping modern methods of construction (MMC) initiative



# LITERATURE REVIEW CONCLUSION

*Mapping modern methods of construction (MMC) initiative*



Efficiency 40%

**MANUFACTURED**

Bespoke solution,  
Standard process

# LITERATURE REVIEW

*PCG Algorithms*

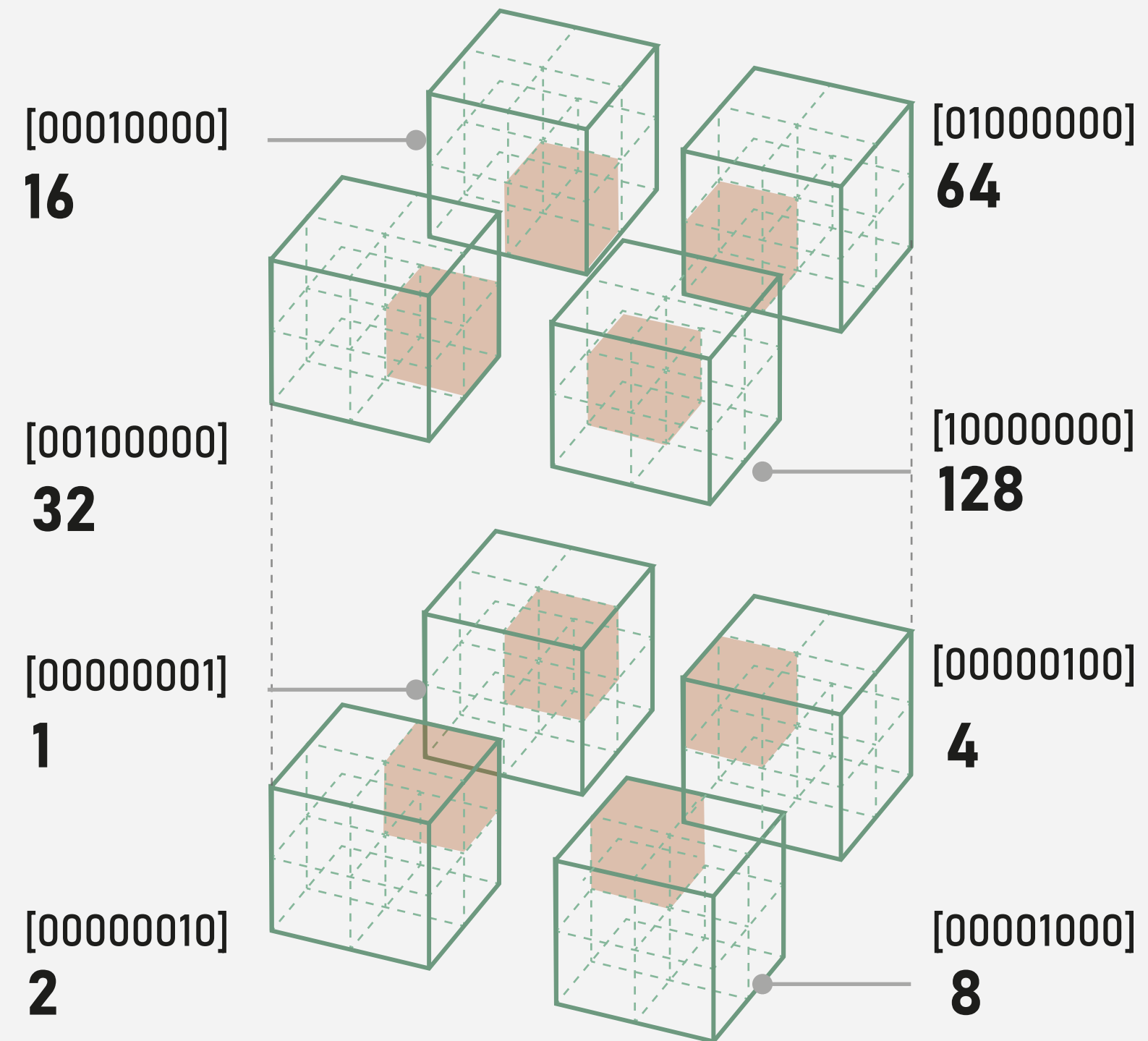
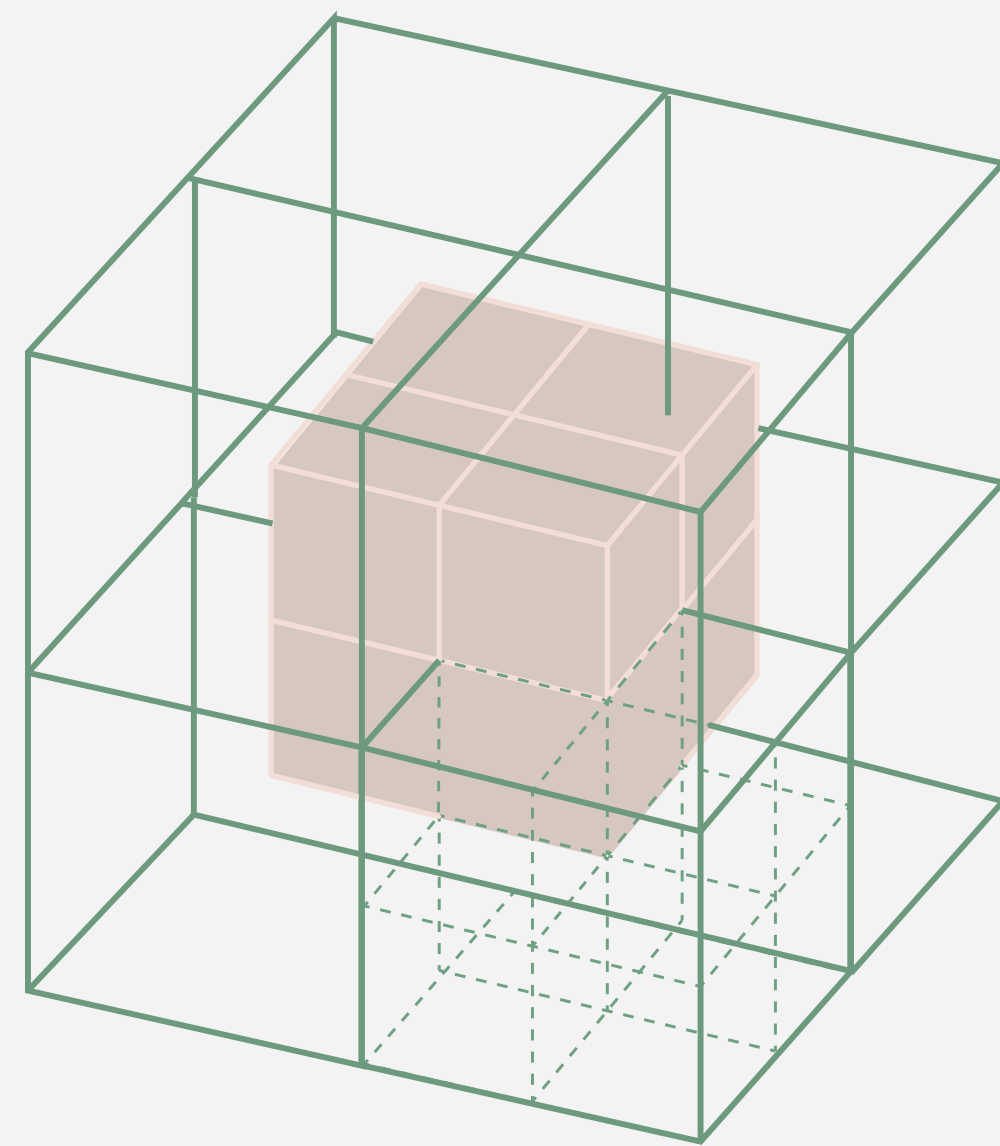
## Boolean Marching Cube

The imaginary cubes surrounding the voxel,

A unique ID based on the location of overlapping parts.

When the unique ID is called, the assigned cube tiles are loaded

256 possible configurations ( $2^8 = 256$ )





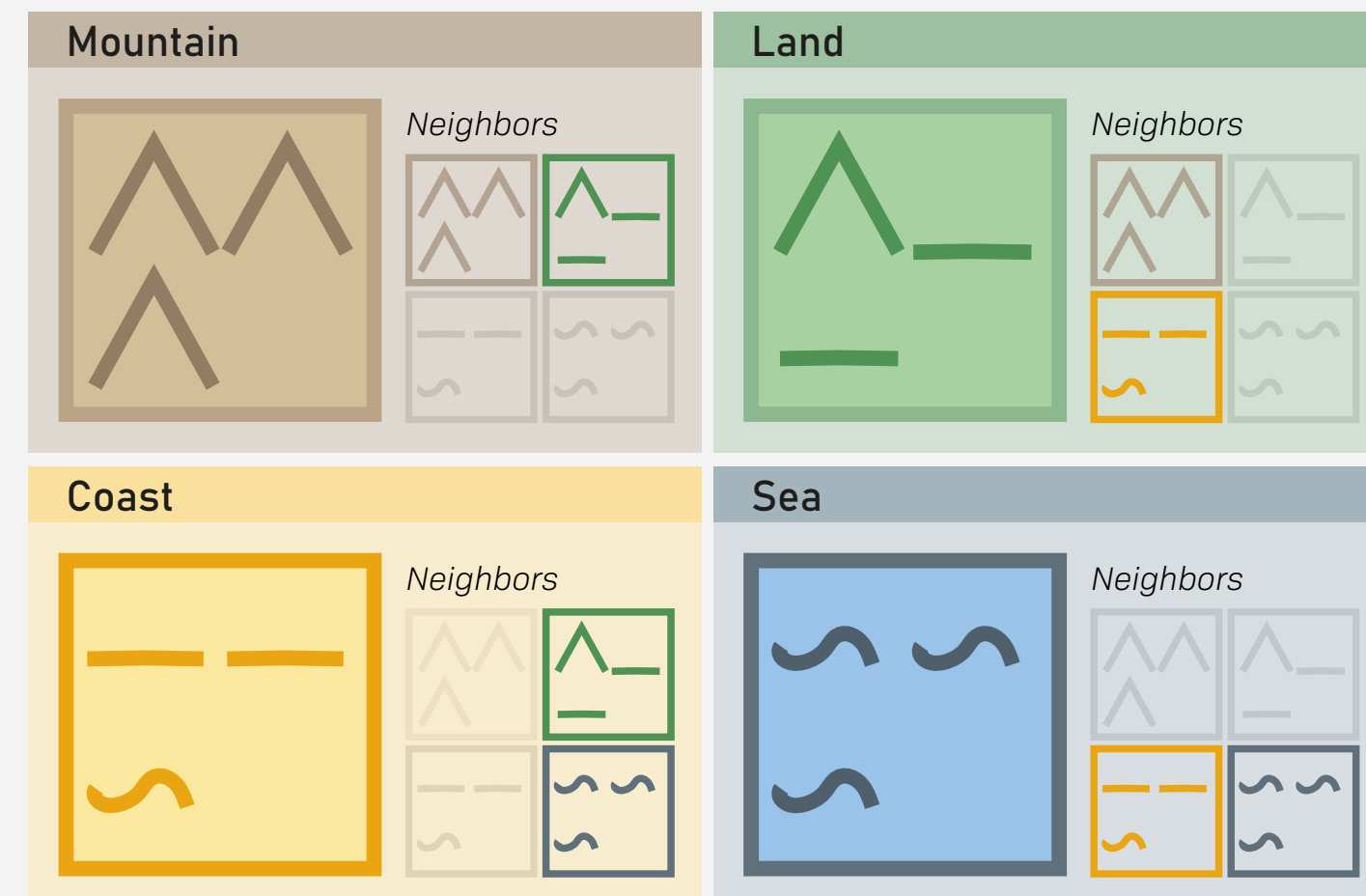
# LITERATURE REVIEW

PCG Algorithms

## Wave function Collapse

The Wave Function collapse algorithm takes the modules containing information of the neighbor constraints to either allow them to be next to themselves or not.

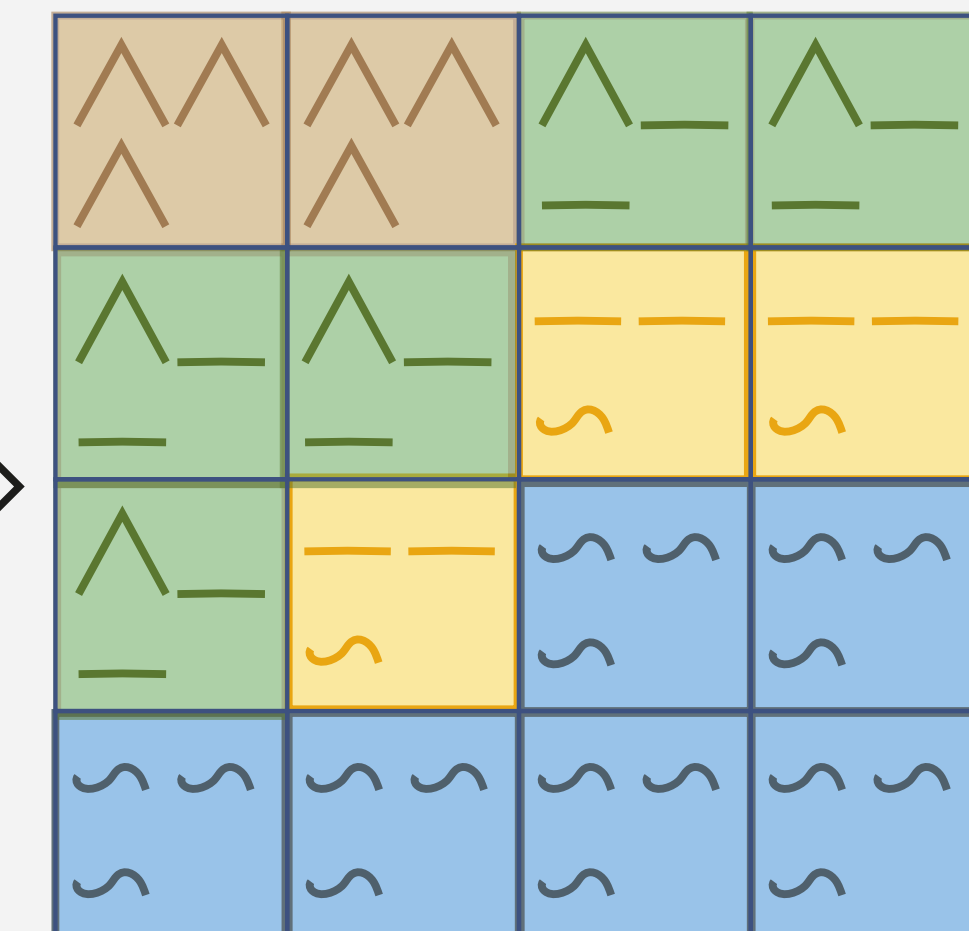
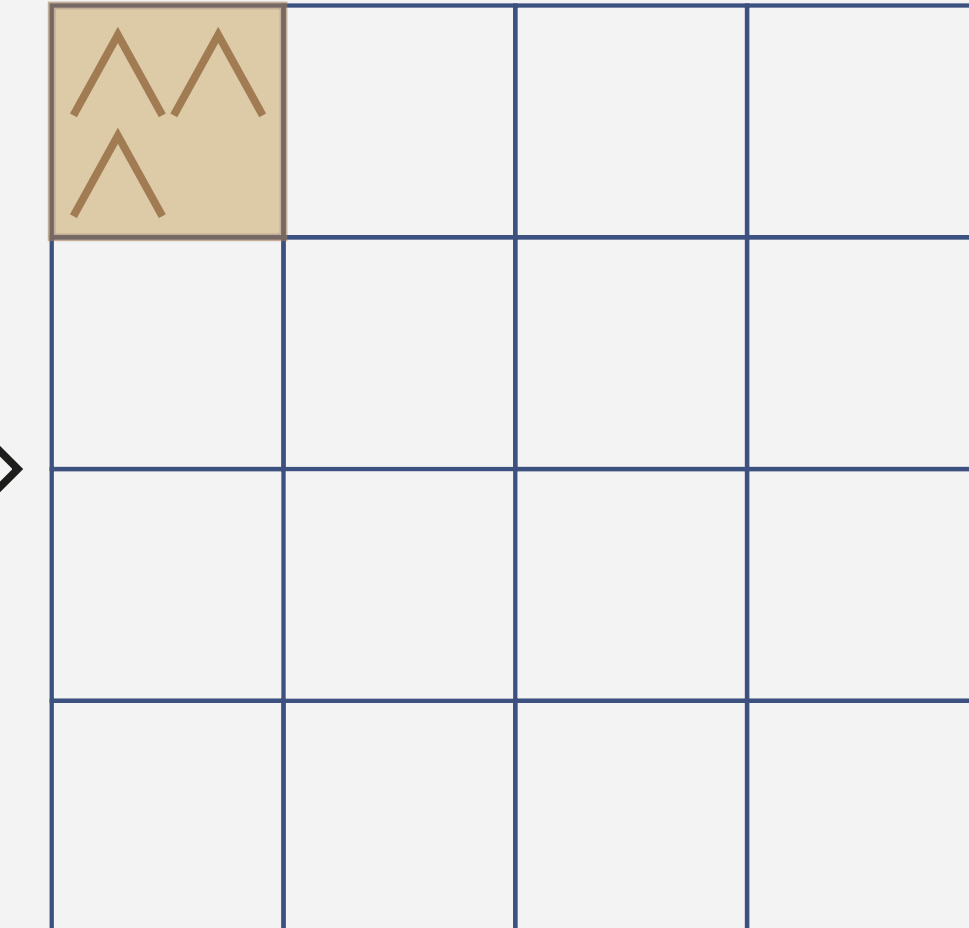
### Modules with neighbor constraints



### Uncollapsed grid



### Collapsed grid



# LITERATURE REVIEW

*PCG Algorithms*

Procedural Content Generation Algorithms

```
graph TD; A[Procedural Content Generation Algorithms] --> B[Boolean Marching Cube]; A --> C[Wave function Collapse]; B --> D[Select the fittest Algorithm]; C --> D;
```

Boolean Marching Cube

Wave function Collapse

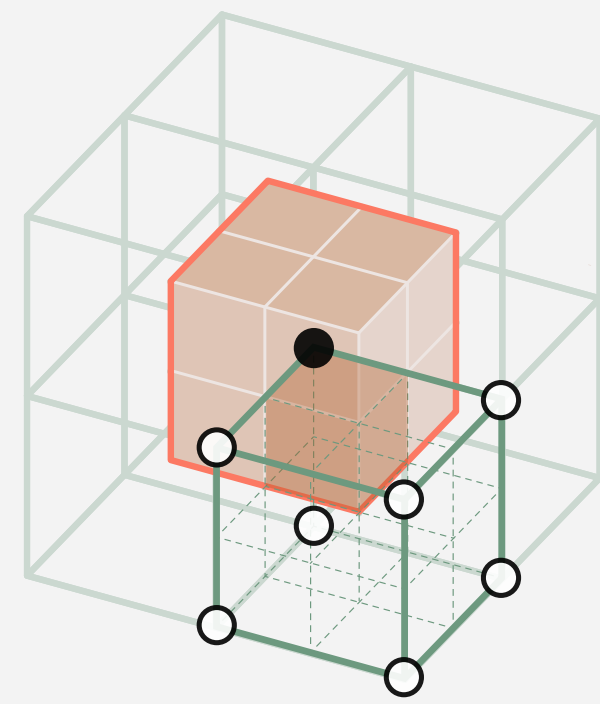
Select the fittest **Algorithm**

LITERATURE REVIEW  
**CONCLUSION**

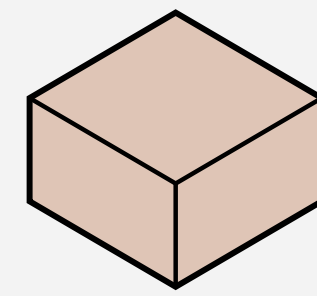
*Algorithms*

	Wave Function Collapse	Boolean Marching Cube
Degree of which experts can encode design ideas	Medium	High
Challenges experienced by users	High	Low
Do not let the user start from scratch	NO	YES
Actions are traceable and impact on result	NO	YES

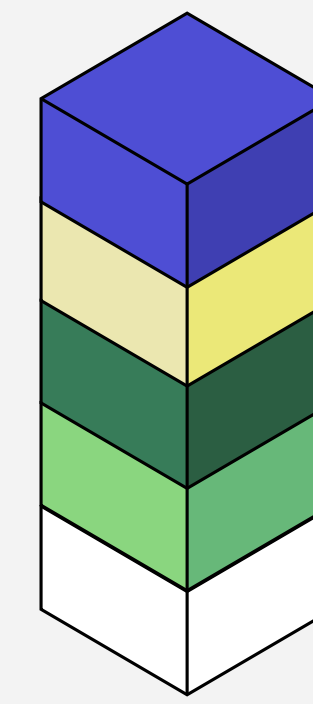
# DESIGN DEVELOPMENTS



**BMC  
Terminology**



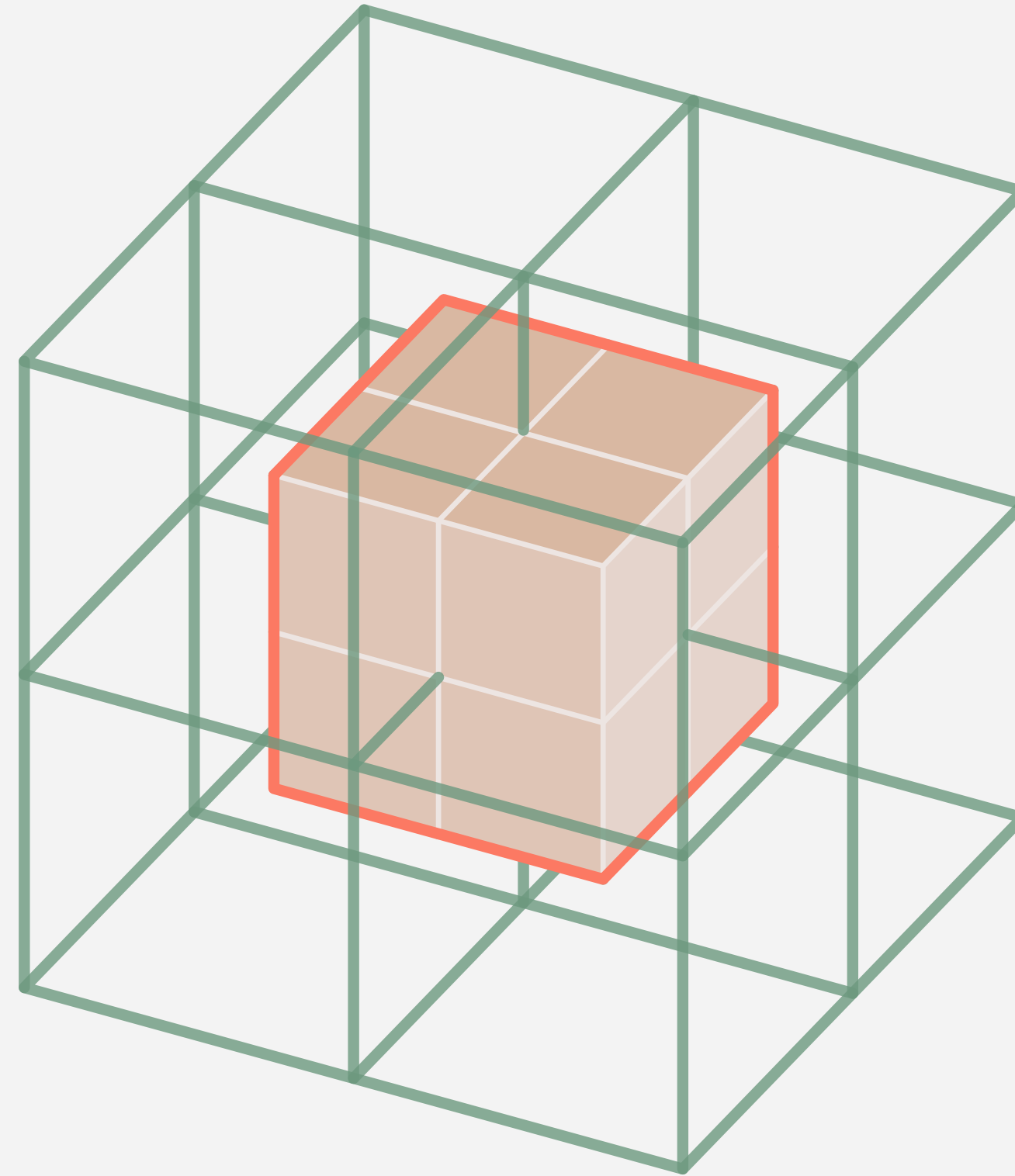
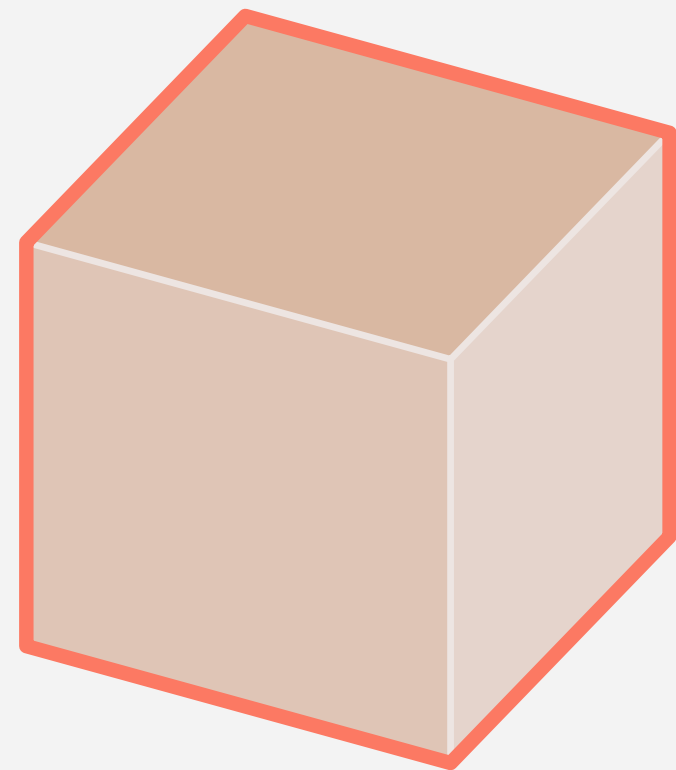
**Surface  
Tilesets**



**Architectural  
Tilesets**

# PRESETTING \_BMC TERMINOLOGY

*Voxel*

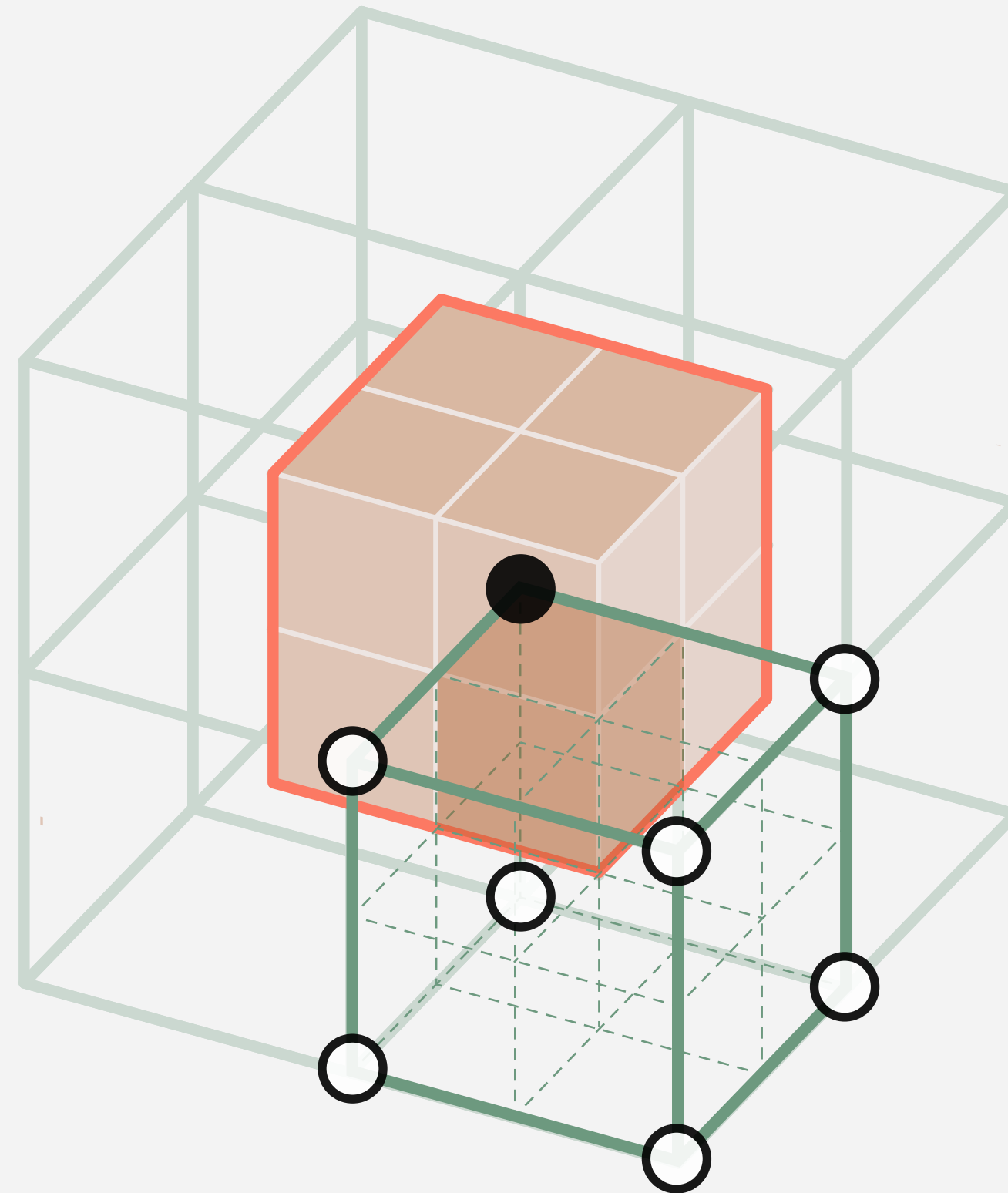
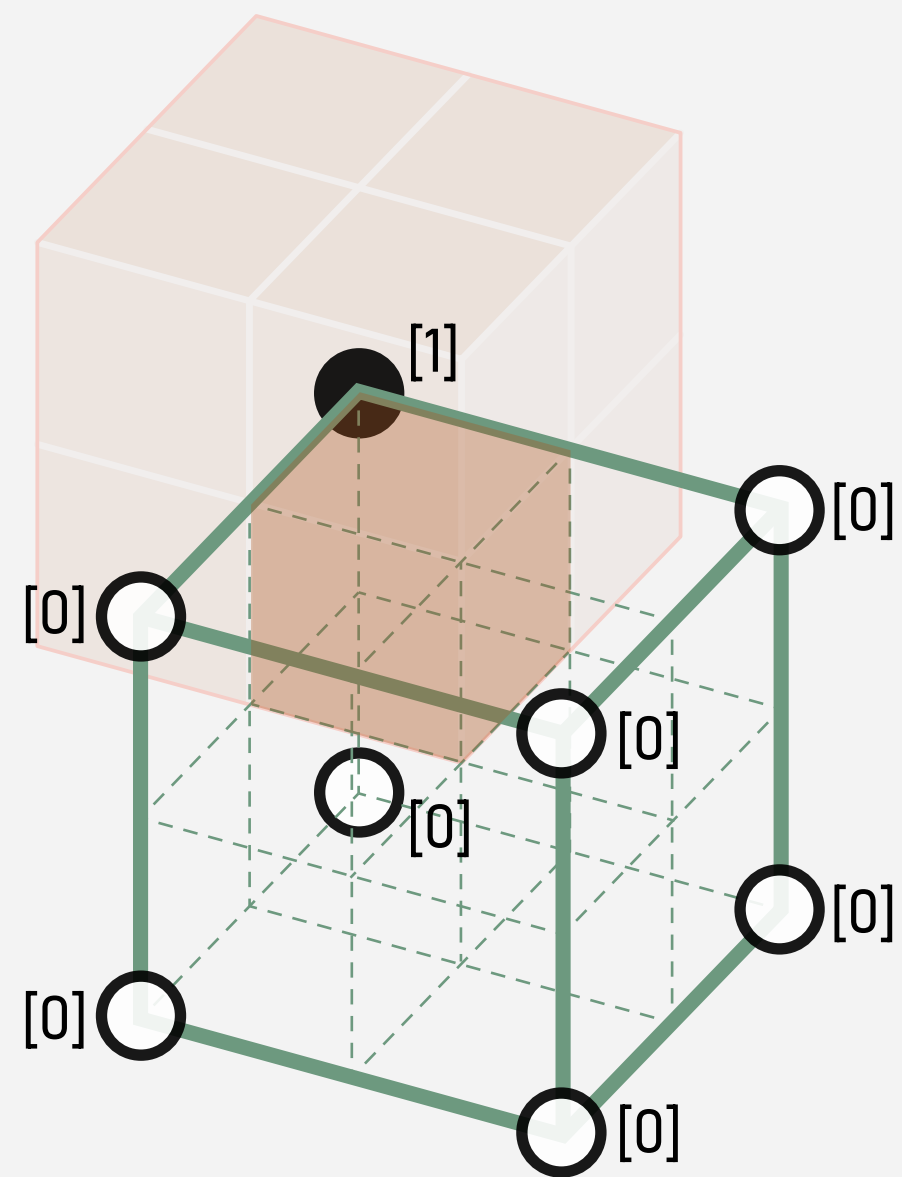


## **Voxel**

A **fundamental component** in making a modular architecture and a **single unit** representation of the initial voxel array from the user input.

# PRESETTING \_BMC TERMINOLOGY

*Cube*

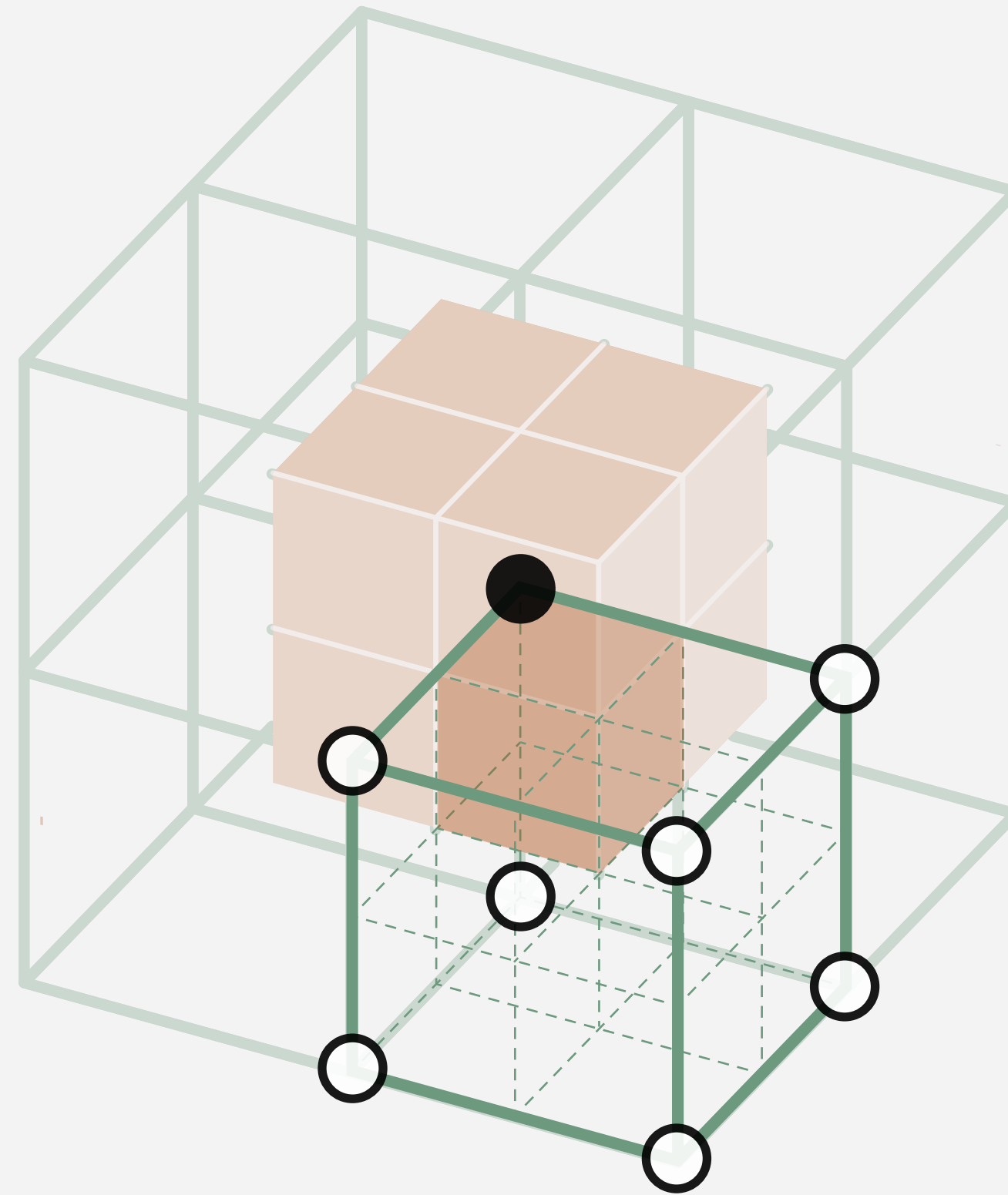
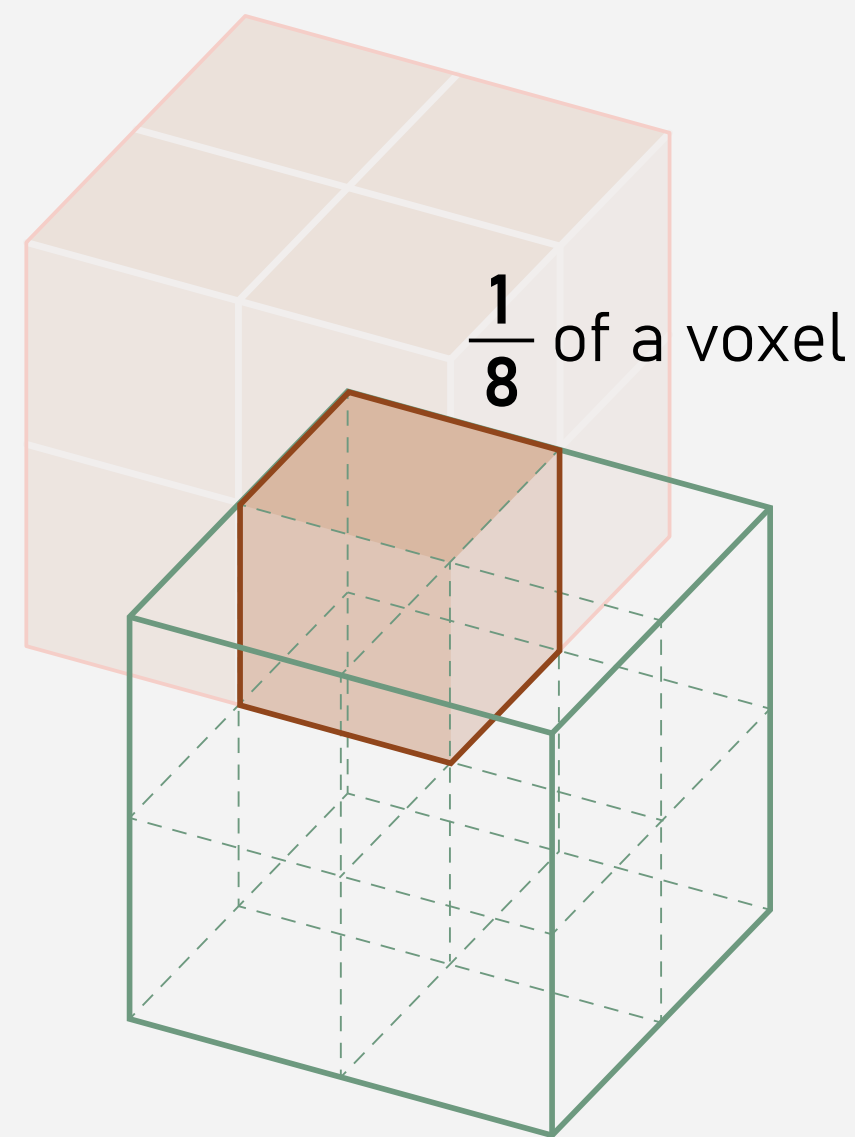


## Cube

A cube reads the **center points of voxels** and checks whether the center point is inside the voxel as a boolean value of 0 and 1.

# PRESETTING \_BMC TERMINOLOGY

*Octant of a voxel*



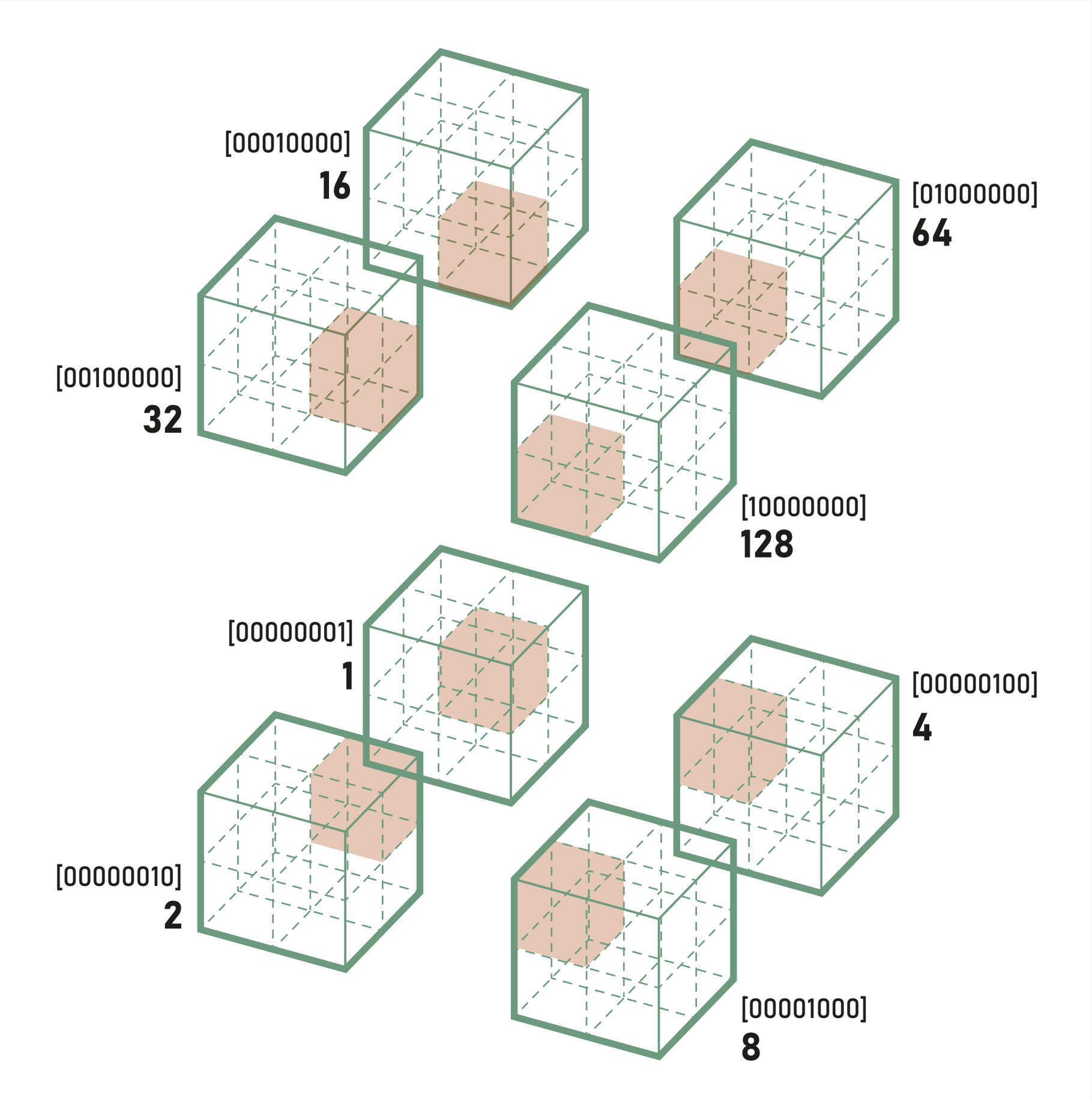
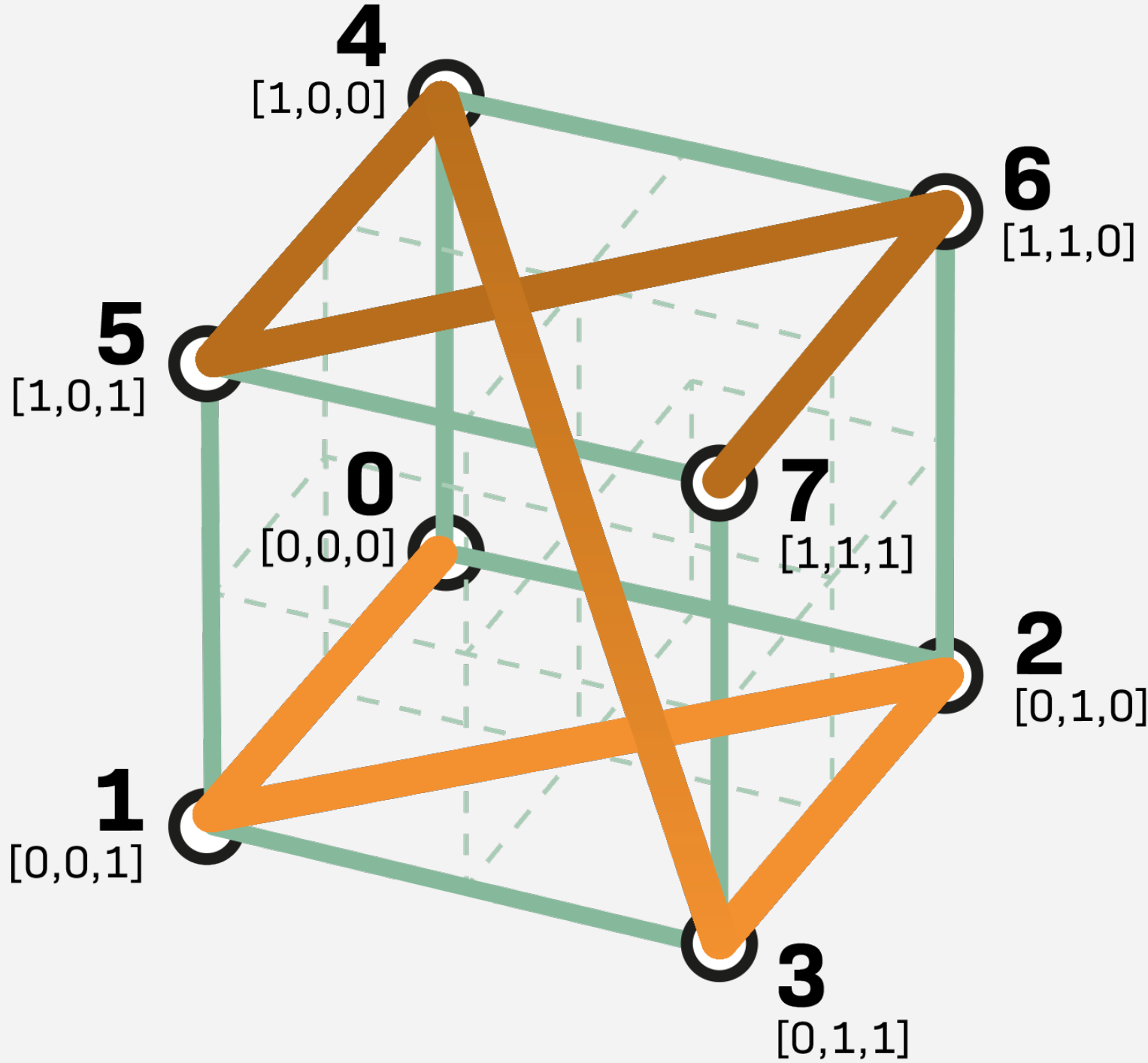
**Octant of a voxel  
(Sub\_voxel)**

**Represents one digit** from the cube configuration.  
It is the region representing the overlap of a voxel and a cube.

# PRESETTING \_BMC SEQUENCE

*BMC reading sequence*

- Inside [1]
- Outside [0]



The **counting sequence** of Sub-voxels is essential in the BMC algorithm  
The sequence determines the reading order the **unique cube IDs**.



































































































































































































# PRESETTING \_CUBE OPTIONS

*Cutting down tilesets*

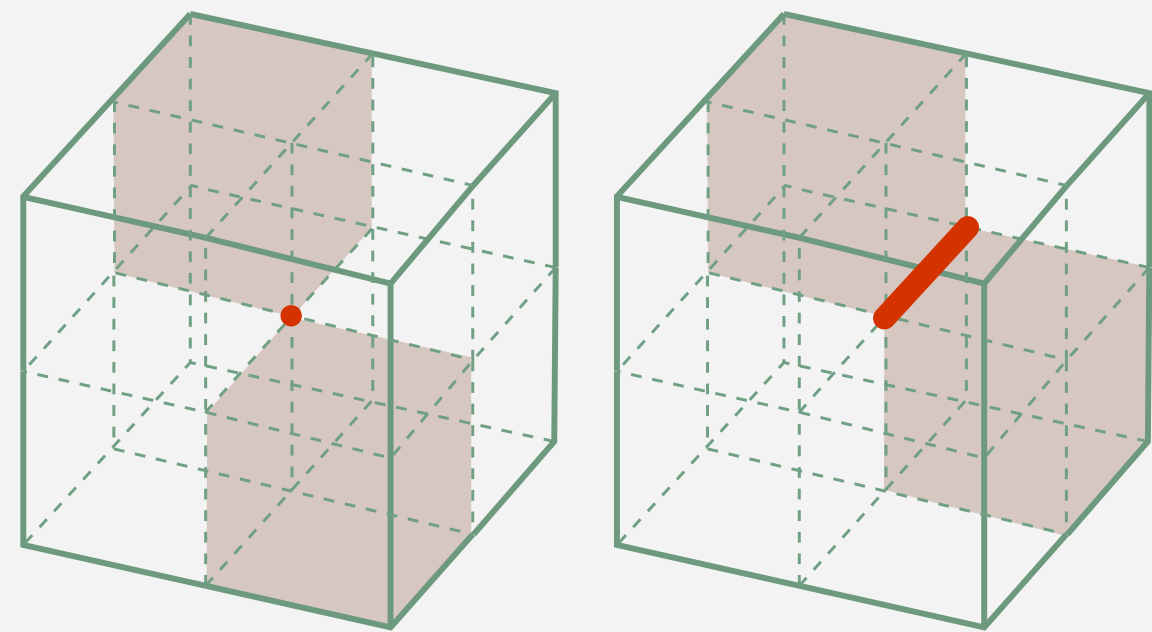
The configuration option of  
one cube is,

$2^8=256$  options.

 0 [0, 0, 0, 0], [0, 0, 0, 0]	 32 [0, 0, 1, 0], [0, 0, 0, 0]	 64 [0, 1, 0, 0], [0, 0, 0, 0]	 96 [0, 1, 1, 0], [0, 0, 0, 0]	 128 [1, 0, 0, 0], [0, 0, 0, 0]	 160 [1, 0, 1, 0], [0, 0, 0, 0]	 192 [1, 1, 0, 0], [0, 0, 0, 0]	 224 [1, 1, 1, 0], [0, 0, 0, 0]
 1 [0, 0, 0, 0], [0, 0, 0, 1]	 33 [0, 0, 1, 0], [0, 0, 0, 1]	 65 [0, 1, 0, 0], [0, 0, 0, 1]	 97 [0, 1, 1, 0], [0, 0, 0, 1]	 129 [1, 0, 0, 0], [0, 0, 0, 1]	 161 [1, 0, 1, 0], [0, 0, 0, 1]	 193 [1, 1, 0, 0], [0, 0, 0, 1]	 225 [1, 1, 1, 0], [0, 0, 0, 1]
 2 [0, 0, 0, 0], [0, 0, 1, 0]	 34 [0, 0, 1, 0], [0, 0, 1, 0]	 66 [0, 1, 0, 0], [0, 0, 1, 0]	 98 [0, 1, 1, 0], [0, 0, 1, 0]	 130 [1, 0, 0, 0], [0, 0, 1, 0]	 162 [1, 0, 1, 0], [0, 0, 1, 0]	 194 [1, 1, 0, 0], [0, 0, 1, 0]	 226 [1, 1, 1, 0], [0, 0, 1, 0]
 3 [0, 0, 0, 0], [0, 0, 1, 1]	 35 [0, 0, 1, 0], [0, 0, 1, 1]	 67 [0, 1, 0, 0], [0, 0, 1, 1]	 99 [0, 1, 1, 0], [0, 0, 1, 1]	 131 [1, 0, 0, 0], [0, 0, 1, 1]	 163 [1, 0, 1, 0], [0, 0, 1, 1]	 195 [1, 1, 0, 0], [0, 0, 1, 1]	 227 [1, 1, 1, 0], [0, 0, 1, 1]
 4 [0, 0, 0, 0], [0, 1, 0, 0]	 36 [0, 0, 1, 0], [0, 1, 0, 0]	 68 [0, 1, 0, 0], [0, 1, 0, 0]	 100 [0, 1, 1, 0], [0, 1, 0, 0]	 132 [1, 0, 0, 0], [0, 1, 0, 0]	 164 [1, 0, 1, 0], [0, 1, 0, 0]	 196 [1, 1, 0, 0], [0, 1, 0, 0]	 228 [1, 1, 1, 0], [0, 1, 0, 0]
 5 [0, 0, 0, 0], [0, 1, 0, 1]	 37 [0, 0, 1, 0], [0, 1, 0, 1]	 69 [0, 1, 0, 0], [0, 1, 0, 1]	 101 [0, 1, 1, 0], [0, 1, 0, 1]	 133 [1, 0, 0, 0], [0, 1, 0, 1]	 165 [1, 0, 1, 0], [0, 1, 0, 1]	 197 [1, 1, 0, 0], [0, 1, 0, 1]	 229 [1, 1, 1, 0], [0, 1, 0, 1]
 6 [0, 0, 0, 0], [0, 1, 1, 0]	 38 [0, 0, 1, 0], [0, 1, 1, 0]	 70 [0, 1, 0, 0], [0, 1, 1, 0]	 102 [0, 1, 1, 0], [0, 1, 1, 0]	 134 [1, 0, 0, 0], [0, 1, 1, 0]	 166 [1, 0, 1, 0], [0, 1, 1, 0]	 198 [1, 1, 0, 0], [0, 1, 1, 0]	 230 [1, 1, 1, 0], [0, 1, 1, 0]
 7 [0, 0, 0, 0], [0, 1, 1, 1]	 39 [0, 0, 1, 0], [0, 1, 1, 1]	 71 [0, 1, 0, 0], [0, 1, 1, 1]	 103 [0, 1, 1, 0], [0, 1, 1, 1]	 135 [1, 0, 0, 0], [0, 1, 1, 1]	 167 [1, 0, 1, 0], [0, 1, 1, 1]	 199 [1, 1, 0, 0], [0, 1, 1, 1]	 231 [1, 1, 1, 0], [0, 1, 1, 1]
 8 [0, 0, 0, 0], [1, 0, 0, 0]	 40 [0, 0, 1, 0], [1, 0, 0, 0]	 72 [0, 1, 0, 0], [1, 0, 0, 0]	 104 [0, 1, 1, 0], [1, 0, 0, 0]	 136 [1, 0, 0, 0], [1, 0, 0, 0]	 168 [1, 0, 1, 0], [1, 0, 0, 0]	 200 [1, 1, 0, 0], [1, 0, 0, 0]	 232 [1, 1, 1, 0], [1, 0, 0, 0]
 9 [0, 0, 0, 0], [1, 0, 0, 1]	 41 [0, 0, 1, 0], [1, 0, 0, 1]	 73 [0, 1, 0, 0], [1, 0, 0, 1]	 105 [0, 1, 1, 0], [1, 0, 0, 1]	 137 [1, 0, 0, 0], [1, 0, 0, 1]	 169 [1, 0, 1, 0], [1, 0, 0, 1]	 201 [1, 1, 0, 0], [1, 0, 0, 1]	 233 [1, 1, 1, 0], [1, 0, 0, 1]
 10 [0, 0, 0, 0], [1, 0, 1, 0]	 42 [0, 0, 1, 0], [1, 0, 1, 0]	 74 [0, 1, 0, 0], [1, 0, 1, 0]	 106 [0, 1, 1, 0], [1, 0, 1, 0]	 138 [1, 0, 0, 0], [1, 0, 1, 0]	 170 [1, 0, 1, 0], [1, 0, 1, 0]	 202 [1, 1, 0, 0], [1, 0, 1, 0]	 234 [1, 1, 1, 0], [1, 0, 1, 0]
 11 [0, 0, 0, 0], [1, 0, 1, 1]	 43 [0, 0, 1, 0], [1, 0, 1, 1]	 75 [0, 1, 0, 0], [1, 0, 1, 1]	 107 [0, 1, 1, 0], [1, 0, 1, 1]	 139 [1, 0, 0, 0], [1, 0, 1, 1]	 171 [1, 0, 1, 0], [1, 0, 1, 1]	 203 [1, 1, 0, 0], [1, 0, 1, 1]	 235 [1, 1, 1, 0], [1, 0, 1, 1]
 12 [0, 0, 0, 0], [1, 1, 0, 0]	 44 [0, 0, 1, 0], [1, 1, 0, 0]	 76 [0, 1, 0, 0], [1, 1, 0, 0]	 108 [0, 1, 1, 0], [1, 1, 0, 0]	 140 [1, 0, 0, 0], [1, 1, 0, 0]	 172 [1, 0, 1, 0], [1, 1, 0, 0]	 204 [1, 1, 0, 0], [1, 1, 0, 0]	 236 [1, 1, 1, 0], [1, 1, 0, 0]
 13 [0, 0, 0, 0], [1, 1, 0, 1]	 45 [0, 0, 1, 0], [1, 1, 0, 1]	 77 [0, 1, 0, 0], [1, 1, 0, 1]	 109 [0, 1, 1, 0], [1, 1, 0, 1]	 141 [1, 0, 0, 0], [1, 1, 0, 1]	 173 [1, 0, 1, 0], [1, 1, 0, 1]	 205 [1, 1, 0, 0], [1, 1, 0, 1]	 237 [1, 1, 1, 0], [1, 1, 0, 1]
 14 [0, 0, 0, 0], [1, 1, 1, 0]	 46 [0, 0, 1, 0], [1, 1, 1, 0]	 78 [0, 1, 0, 0], [1, 1, 1, 0]	 110 [0, 1, 1, 0], [1, 1, 1, 0]	 142 [1, 0, 0, 0], [1, 1, 1, 0]	 174 [1, 0, 1, 0], [1, 1, 1, 0]	 206 [1, 1, 0, 0], [1, 1, 1, 0]	 238 [1, 1, 1, 0], [1, 1, 1, 0]
 15 [0, 0, 0, 0], [1, 1, 1, 1]	 47 [0, 0, 1, 0], [1, 1, 1, 1]	 79 [0, 1, 0, 0], [1, 1, 1, 1]	 111 [0, 1, 1, 0], [1, 1, 1, 1]	 143 [1, 0, 0, 0], [1, 1, 1, 1]	 175 [1, 0, 1, 0], [1, 1, 1, 1]	 207 [1, 1, 0, 0], [1, 1, 1, 1]	 239 [1, 1, 1, 0], [1, 1, 1, 1]
 16 [0, 0, 0, 1], [0, 0, 0, 0]	 48 [0, 0, 1, 1], [0, 0, 0, 0]	 80 [0, 1, 0, 1], [0, 0, 0, 0]	 112 [0, 1, 1, 1], [0, 0, 0, 0]	 144 [1, 0, 0, 1], [0, 0, 0, 0]	 176 [1, 0, 1, 1], [0, 0, 0, 0]	 208 [1, 1, 0, 1], [0, 0, 0, 0]	 240 [1, 1, 1, 1], [0, 0, 0, 0]
 17 [0, 0, 0, 1], [0, 0, 0, 1]	 49 [0, 0, 1, 1], [0, 0, 0, 1]	 81 [0, 1, 0, 1], [0, 0, 0, 1]	 113 [0, 1, 1, 1], [0, 0, 0, 1]	 145 [1, 0, 0, 1], [0, 0, 0, 1]	 177 [1, 0, 1, 1], [0, 0, 0, 1]	 209 [1, 1, 0, 1], [0, 0, 0, 1]	 241 [1, 1, 1, 1], [0, 0, 0, 1]
 18 [0, 0, 0, 1], [0, 0, 1, 0]	 50 [0, 0, 1, 1], [0, 0, 1, 0]	 82 [0, 1, 0, 1], [0, 0, 1, 0]	 114 [0, 1, 1, 1], [0, 0, 1, 0]	 146 [1, 0, 0, 1], [0, 0, 1, 0]	 178 [1, 0, 1, 1], [0, 0, 1, 0]	 210 [1, 1, 0, 1], [0, 0, 1, 0]	 242 [1, 1, 1, 1], [0, 0, 1, 0]
 19 [0, 0, 0, 1], [0, 0, 1, 1]	 51 [0, 0, 1, 1], [0, 0, 1, 1]	 83 [0, 1, 0, 1], [0, 0, 1, 1]	 115 [0, 1, 1, 1], [0, 0, 1, 1]	 147 [1, 0, 0, 1], [0, 0, 1, 1]	 179 [1, 0, 1, 1], [0, 0, 1, 1]	 211 [1, 1, 0, 1], [0, 0, 1, 1]	 243 [1, 1, 1, 1], [0, 0, 1, 1]
 20 [0, 0, 0, 1], [0, 1, 0, 0]	 52 [0, 0, 1, 1], [0, 1, 0, 0]	 84 [0, 1, 0, 1], [0, 1, 0, 0]	 116 [0, 1, 1, 1], [0, 1, 0, 0]	 148 [1, 0, 0, 1], [0, 1, 0, 0]	 180 [1, 0, 1, 1], [0, 1, 0, 0]	 212 [1, 1, 0, 1], [0, 1, 0, 0]	 244 [1, 1, 1, 1], [0, 1, 0, 0]
 21 [0, 0, 0, 1], [0, 1, 0, 1]	 53 [0, 0, 1, 1], [0, 1, 0, 1]	 85 [0, 1, 0, 1], [0, 1, 0, 1]	 117 [0, 1, 1, 1], [0, 1, 0, 1]	 149 [1, 0, 0, 1], [0, 1, 0, 1]	 181 [1, 0, 1, 1], [0, 1, 0, 1]	 213 [1, 1, 0, 1], [0, 1, 0, 1]	 245 [1, 1, 1, 1], [0, 1, 0, 1]
 22 [0, 0, 0, 1], [0, 1, 1, 0]	 54 [0, 0, 1, 1], [0, 1, 1, 0]	 86 [0, 1, 0, 1], [0, 1, 1, 0]	 118 [0, 1, 1, 1], [0, 1, 1, 0]	 150 [1, 0, 0, 1], [0, 1, 1, 0]	 182 [1, 0, 1, 1], [0, 1, 1, 0]	 214 [1, 1, 0, 1], [0, 1, 1, 0]	 246 [1, 1, 1, 1], [0, 1, 1, 0]
 23 [0, 0, 0, 1], [0, 1, 1, 1]	 55 [0, 0, 1, 1], [0, 1, 1, 1]	 87 [0, 1, 0, 1], [0, 1, 1, 1]	 119 [0, 1, 1, 1], [0, 1, 1, 1]	 151 [1, 0, 0, 1], [0, 1, 1, 1]	 183 [1, 0, 1, 1], [0, 1, 1, 1]	 215 [1, 1, 0, 1], [0, 1, 1, 1]	 247 [1, 1, 1, 1], [0, 1, 1, 1]

# PRESETTING \_CUBE OPTIONS

*Cutting down tilesets*



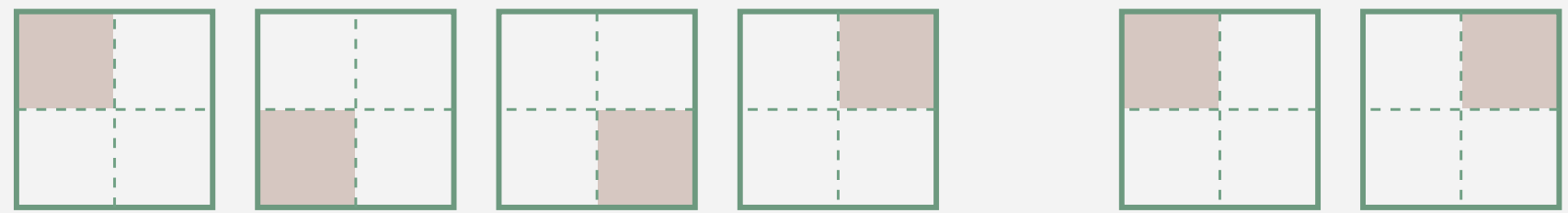
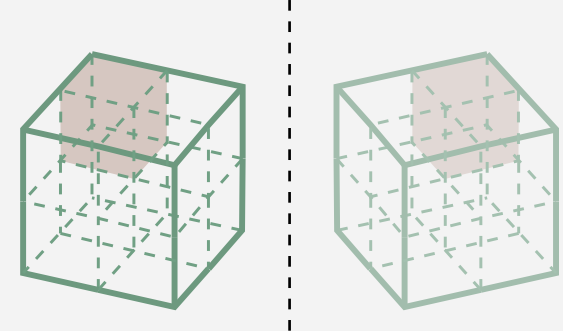
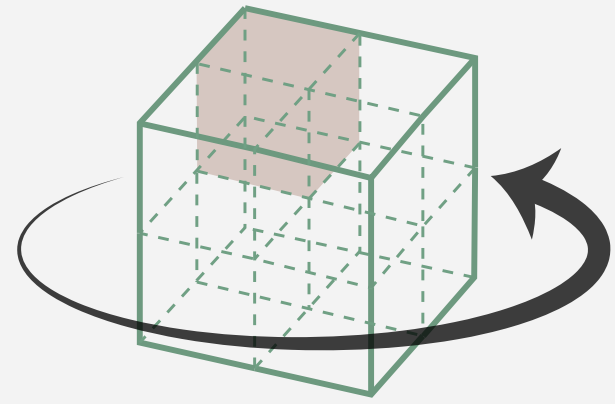
Eliminate  
Point-to-point, Edge-to-edge  
connections of the modules  
do not create  
feasible compositions.

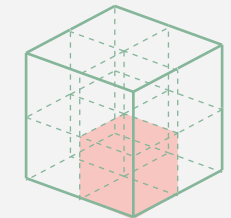
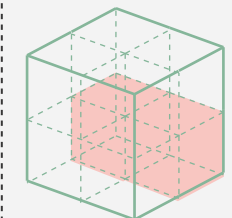
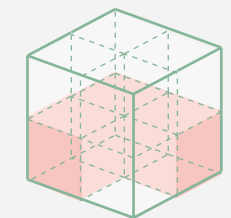
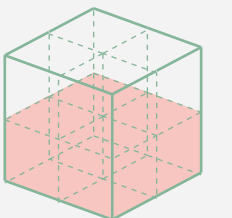
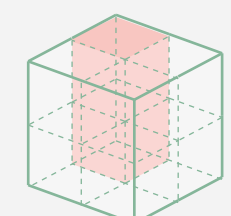
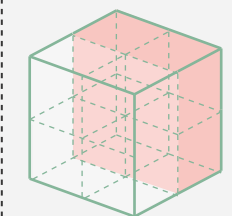
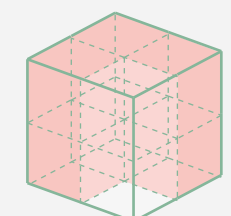
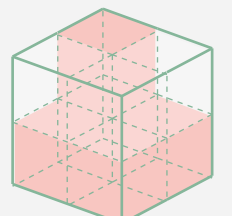
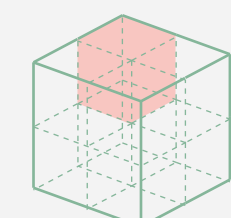
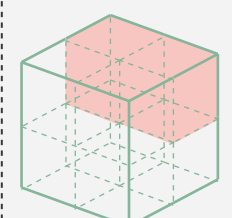
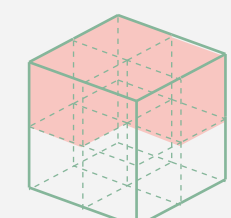
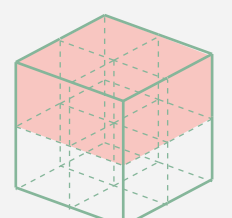
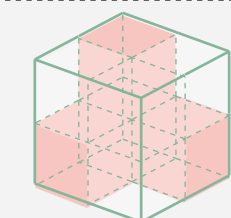
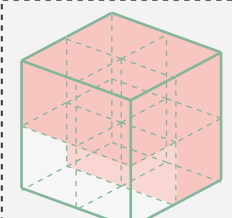
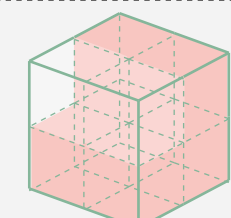
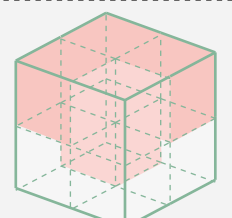
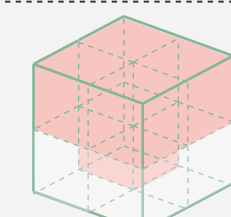
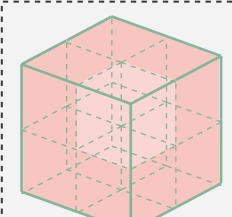
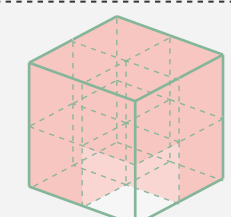
> Cut down to 126 configurations

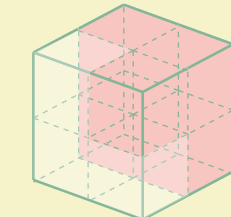
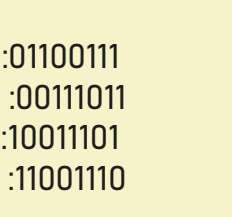
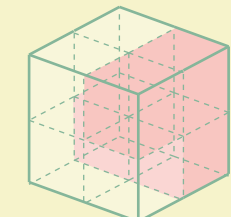
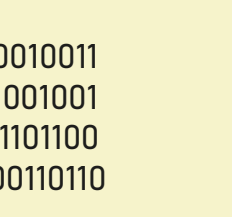
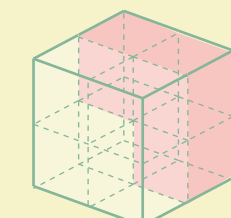
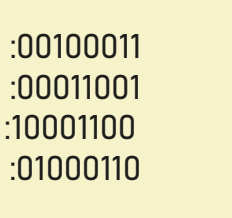
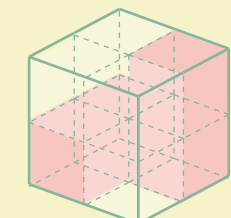
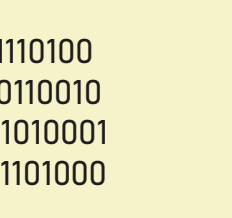
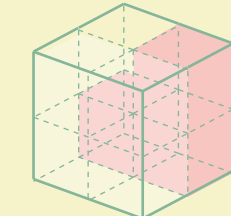
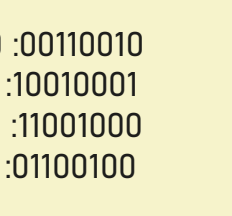
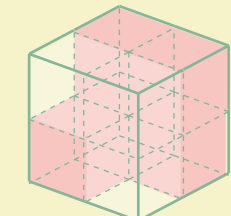
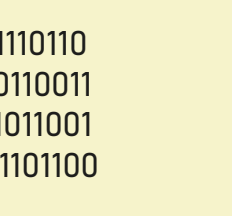
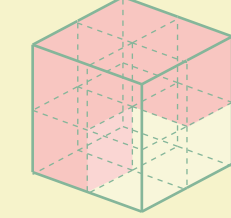
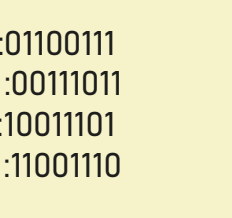
0 [0, 0, 0, 0], [0, 0, 0, 0]	32 [0, 0, 1, 0], [0, 0, 0, 0]	64 [0, 1, 0, 0], [0, 0, 0, 0]	96 [0, 1, 1, 0], [0, 0, 0, 0]	128 [1, 0, 0, 0], [0, 0, 0, 0]	160 [1, 0, 1, 0], [0, 0, 0, 0]	192 [1, 1, 0, 0], [0, 0, 0, 0]	224 [1, 1, 1, 0], [0, 0, 0, 0]
1 [0, 0, 0, 0], [0, 0, 0, 1]	<del> 33 [0, 0, 1, 0], [0, 0, 0, 1]</del>	<del> 65 [0, 1, 0, 0], [0, 0, 0, 1]</del>	<del> 97 [0, 1, 1, 0], [0, 0, 0, 1]</del>	<del> 129 [1, 0, 0, 0], [0, 0, 0, 1]</del>	<del> 161 [1, 0, 1, 0], [0, 0, 0, 1]</del>	<del> 193 [1, 1, 0, 0], [0, 0, 0, 1]</del>	<del> 225 [1, 1, 1, 0], [0, 0, 0, 1]</del>
2 [0, 0, 0, 0], [0, 0, 1, 0]	34 [0, 0, 1, 0], [0, 0, 1, 0]	<del> 66 [0, 1, 0, 0], [0, 0, 1, 0]</del>	<del> 98 [0, 1, 1, 0], [0, 0, 1, 0]</del>	<del> 130 [1, 0, 0, 0], [0, 0, 1, 0]</del>	162 [1, 0, 1, 0], [0, 0, 1, 0]	<del> 194 [1, 1, 0, 0], [0, 0, 1, 0]</del>	<del> 226 [1, 1, 1, 0], [0, 0, 1, 0]</del>
3 [0, 0, 0, 0], [0, 0, 1, 1]	35 [0, 0, 1, 0], [0, 0, 1, 1]	<del> 67 [0, 1, 0, 0], [0, 0, 1, 1]</del>	<del> 99 [0, 1, 1, 0], [0, 0, 1, 1]</del>	<del> 131 [1, 0, 0, 0], [0, 0, 1, 1]</del>	<del> 163 [1, 0, 1, 0], [0, 0, 1, 1]</del>	<del> 195 [1, 1, 0, 0], [0, 0, 1, 1]</del>	<del> 227 [1, 1, 1, 0], [0, 0, 1, 1]</del>
4 [0, 0, 0, 0], [0, 1, 0, 0]	<del> 36 [0, 0, 1, 0], [0, 1, 0, 0]</del>	68 [0, 1, 0, 0], [0, 1, 0, 0]	<del> 100 [0, 1, 1, 0], [0, 1, 0, 0]</del>	<del> 132 [1, 0, 0, 0], [0, 1, 0, 0]</del>	164 [1, 0, 1, 0], [0, 1, 0, 0]	<del> 196 [1, 1, 0, 0], [0, 1, 0, 0]</del>	<del> 228 [1, 1, 1, 0], [0, 1, 0, 0]</del>
5 [0, 0, 0, 0], [0, 1, 0, 1]	<del> 37 [0, 0, 1, 0], [0, 1, 0, 1]</del>	69 [0, 1, 0, 0], [0, 1, 0, 1]	<del> 101 [0, 1, 1, 0], [0, 1, 0, 1]</del>	<del> 133 [1, 0, 0, 0], [0, 1, 0, 1]</del>	165 [1, 0, 1, 0], [0, 1, 0, 1]	<del> 197 [1, 1, 0, 0], [0, 1, 0, 1]</del>	<del> 229 [1, 1, 1, 0], [0, 1, 0, 1]</del>
<del> 6 [0, 0, 0, 0], [0, 1, 1, 0]</del>	<del> 38 [0, 0, 1, 0], [0, 1, 1, 0]</del>	<del> 70 [0, 1, 0, 0], [0, 1, 1, 0]</del>	<del> 102 [0, 1, 1, 0], [0, 1, 1, 0]</del>	<del> 134 [1, 0, 0, 0], [0, 1, 1, 0]</del>	166 [1, 0, 1, 0], [0, 1, 1, 0]	<del> 198 [1, 1, 0, 0], [0, 1, 1, 0]</del>	<del> 230 [1, 1, 1, 0], [0, 1, 1, 0]</del>
7 [0, 0, 0, 0], [0, 1, 1, 1]	39 [0, 0, 1, 0], [0, 1, 1, 1]	<del> 71 [0, 1, 0, 0], [0, 1, 1, 1]</del>	<del> 103 [0, 1, 1, 0], [0, 1, 1, 1]</del>	<del> 135 [1, 0, 0, 0], [0, 1, 1, 1]</del>	167 [1, 0, 1, 0], [0, 1, 1, 1]	<del> 199 [1, 1, 0, 0], [0, 1, 1, 1]</del>	<del> 231 [1, 1, 1, 0], [0, 1, 1, 1]</del>
8 [0, 0, 0, 0], [1, 0, 0, 0]	<del> 40 [0, 0, 1, 0], [1, 0, 0, 0]</del>	<del> 72 [0, 1, 0, 0], [1, 0, 0, 0]</del>	<del> 104 [0, 1, 1, 0], [1, 0, 0, 0]</del>	136 [1, 0, 0, 0], [1, 0, 0, 0]	168 [1, 0, 1, 0], [1, 0, 0, 0]	<del> 200 [1, 1, 0, 0], [1, 0, 0, 0]</del>	<del> 232 [1, 1, 1, 0], [1, 0, 0, 0]</del>
9 [0, 0, 0, 0], [1, 0, 0, 1]	<del> 41 [0, 0, 1, 0], [1, 0, 0, 1]</del>	<del> 73 [0, 1, 0, 0], [1, 0, 0, 1]</del>	<del> 105 [0, 1, 1, 0], [1, 0, 0, 1]</del>	<del> 137 [1, 0, 0, 0], [1, 0, 0, 1]</del>	169 [1, 0, 1, 0], [1, 0, 0, 1]	<del> 201 [1, 1, 0, 0], [1, 0, 0, 1]</del>	<del> 233 [1, 1, 1, 0], [1, 0, 0, 1]</del>
10 [0, 0, 0, 0], [1, 0, 1, 0]	42 [0, 0, 1, 0], [1, 0, 1, 0]	<del> 74 [0, 1, 0, 0], [1, 0, 1, 0]</del>	<del> 106 [0, 1, 1, 0], [1, 0, 1, 0]</del>	138 [1, 0, 0, 0], [1, 0, 1, 0]	170 [1, 0, 1, 0], [1, 0, 1, 0]	<del> 202 [1, 1, 0, 0], [1, 0, 1, 0]</del>	<del> 234 [1, 1, 1, 0], [1, 0, 1, 0]</del>
11 [0, 0, 0, 0], [1, 0, 1, 1]	43 [0, 0, 1, 0], [1, 0, 1, 1]	<del> 75 [0, 1, 0, 0], [1, 0, 1, 1]</del>	<del> 107 [0, 1, 1, 0], [1, 0, 1, 1]</del>	139 [1, 0, 0, 0], [1, 0, 1, 1]	171 [1, 0, 1, 0], [1, 0, 1, 1]	<del> 203 [1, 1, 0, 0], [1, 0, 1, 1]</del>	<del> 235 [1, 1, 1, 0], [1, 0, 1, 1]</del>
12 [0, 0, 0, 0], [1, 1, 0, 0]	<del> 44 [0, 0, 1, 0], [1, 1, 0, 0]</del>	76 [0, 1, 0, 0], [1, 1, 0, 0]	<del> 108 [0, 1, 1, 0], [1, 1, 0, 0]</del>	140 [1, 0, 0, 0], [1, 1, 0, 0]	172 [1, 0, 1, 0], [1, 1, 0, 0]	<del> 204 [1, 1, 0, 0], [1, 1, 0, 0]</del>	<del> 236 [1, 1, 1, 0], [1, 1, 0, 0]</del>
13 [0, 0, 0, 0], [1, 1, 0, 1]	<del> 45 [0, 0, 1, 0], [1, 1, 0, 1]</del>	77 [0, 1, 0, 0], [1, 1, 0, 1]	<del> 109 [0, 1, 1, 0], [1, 1, 0, 1]</del>	141 [1, 0, 0, 0], [1, 1, 0, 1]	<del> 173 [1, 0, 1, 0], [1, 1, 0, 1]</del>	<del> 205 [1, 1, 0, 0], [1, 1, 0, 1]</del>	<del> 237 [1, 1, 1, 0], [1, 1, 0, 1]</del>
14 [0, 0, 0, 0], [1, 1, 1, 0]	46 [0, 0, 1, 0], [1, 1, 1, 0]	78 [0, 1, 0, 0], [1, 1, 1, 0]	<del> 110 [0, 1, 1, 0], [1, 1, 1, 0]</del>	142 [1, 0, 0, 0], [1, 1, 1, 0]	174 [1, 0, 1, 0], [1, 1, 1, 0]	<del> 206 [1, 1, 0, 0], [1, 1, 1, 0]</del>	<del> 238 [1, 1, 1, 0], [1, 1, 1, 0]</del>
15 [0, 0, 0, 0], [1, 1, 1, 1]	47 [0, 0, 1, 0], [1, 1, 1, 1]	79 [0, 1, 0, 0], [1, 1, 1, 1]	<del> 111 [0, 1, 1, 0], [1, 1, 1, 1]</del>	143 [1, 0, 0, 0], [1, 1, 1, 1]	175 [1, 0, 1, 0], [1, 1, 1, 1]	<del> 207 [1, 1, 0, 0], [1, 1, 1, 1]</del>	<del> 239 [1, 1, 1, 0], [1, 1, 1, 1]</del>
16 [0, 0, 0, 1], [0, 0, 0, 0]	48 [0, 0, 1, 1], [0, 0, 0, 0]	80 [0, 1, 0, 1], [0, 0, 0, 0]	112 [0, 1, 1, 1], [0, 0, 0, 0]	<del> 144 [1, 0, 0, 1], [0, 0, 0, 0]</del>	176 [1, 0, 1, 1], [0, 0, 0, 0]	<del> 208 [1, 1, 0, 1], [0, 0, 0, 0]</del>	<del> 240 [1, 1, 1, 1], [0, 0, 0, 0]</del>
<del> 17 [0, 0, 0, 1], [0, 0, 0, 1]</del>	<del> 49 [0, 0, 1, 1], [0, 0, 0, 1]</del>	81 [0, 1, 0, 1], [0, 0, 0, 1]	113 [0, 1, 1, 1], [0, 0, 0, 1]	<del> 145 [1, 0, 0, 1], [0, 0, 0, 1]</del>	177 [1, 0, 1, 1], [0, 0, 0, 1]	<del> 209 [1, 1, 0, 1], [0, 0, 0, 1]</del>	<del> 241 [1, 1, 1, 1], [0, 0, 0, 1]</del>
<del> 18 [0, 0, 0, 1], [0, 0, 1, 0]</del>	50 [0, 0, 1, 1], [0, 0, 1, 0]	<del> 82 [0, 1, 0, 1], [0, 0, 1, 0]</del>	114 [0, 1, 1, 1], [0, 0, 1, 0]	<del> 146 [1, 0, 0, 1], [0, 0, 1, 0]</del>	178 [1, 0, 1, 1], [0, 0, 1, 0]	<del> 210 [1, 1, 0, 1], [0, 0, 1, 0]</del>	<del> 242 [1, 1, 1, 1], [0, 0, 1, 0]</del>
19 [0, 0, 0, 1], [0, 0, 1, 1]	<del> 51 [0, 0, 1, 1], [0, 0, 1, 1]</del>	<del> 83 [0, 1, 0, 1], [0, 0, 1, 1]</del>	115 [0, 1, 1, 1], [0, 0, 1, 1]	<del> 147 [1, 0, 0, 1], [0, 0, 1, 1]</del>	179 [1, 0, 1, 1], [0, 0, 1, 1]	<del> 211 [1, 1, 0, 1], [0, 0, 1, 1]</del>	<del> 243 [1, 1, 1, 1], [0, 0, 1, 1]</del>
<del> 20 [0, 0, 0, 1], [0, 1, 0, 0]</del>	52 [0, 0, 1, 1], [0, 1, 0, 0]	<del> 84 [0, 1, 0, 1], [0, 1, 0, 0]</del>	<del> 116 [0, 1, 1, 1], [0, 1, 0, 0]</del>	<del> 148 [1, 0, 0, 1], [0, 1, 0, 0]</del>	180 [1, 0, 1, 1], [0, 1, 0, 0]	<del> 212 [1, 1, 0, 1], [0, 1, 0, 0]</del>	<del> 244 [1, 1, 1, 1], [0, 1, 0, 0]</del>
<del> 21 [0, 0, 0, 1], [0, 1, 0, 1]</del>	<del> 53 [0, 0, 1, 1], [0, 1, 0, 1]</del>	85 [0, 1, 0, 1], [0, 1, 0, 1]	117 [0, 1, 1, 1], [0, 1, 0, 1]	<del> 149 [1, 0, 0, 1], [0, 1, 0, 1]</del>	181 [1, 0, 1, 1], [0, 1, 0, 1]	<del> 213 [1, 1, 0, 1], [0, 1, 0, 1]</del>	<del> 245 [1, 1, 1, 1], [0, 1, 0, 1]</del>
<del> 22 [0, 0, 0, 1], [0, 1, 1, 0]</del>	<del> 54 [0, 0, 1, 1], [0, 1, 1, 0]</del>	<del> 86 [0, 1, 0, 1], [0, 1, 1, 0]</del>	118 [0, 1, 1, 1], [0, 1, 1, 0]	<del> 150 [1, 0, 0, 1], [0, 1, 1, 0]</del>	182 [1, 0, 1, 1], [0, 1, 1, 0]	<del> 214 [1, 1, 0, 1], [0, 1, 1, 0]</del>	<del> 246 [1, 1, 1, 1], [0, 1, 1, 0]</del>
23 [0, 0, 0, 1], [0, 1, 1, 1]	55 [0, 0, 1, 1], [0, 1, 1, 1]	87 [0, 1, 0, 1], [0, 1, 1, 1]	<del> 119 [0, 1, 1, 1], [0, 1, 1, 1]</del>	<del> 151 [1, 0, 0, 1], [0, 1, 1, 1]</del>	183 [1, 0, 1, 1], [0, 1, 1, 1]	<del> 215 [1, 1, 0, 1], [0, 1, 1, 1]</del>	<del> 247 [1, 1, 1, 1], [0, 1, 1, 1]</del>
<del> 24 [0, 0, 0, 1], [1, 0, 0, 0]</del>	<del> 56 [0, 0, 1, 1], [1, 0, 0, 0]</del>	88 [0, 1, 0, 1], [1, 0, 0, 0]	<del> 120 [0, 1, 1, 1], [1, 0, 0, 0]</del>	<del> 152 [1, 0, 0, 1], [1, 0, 0, 0]</del>	184 [1, 0, 1, 1], [1, 0, 0, 0]	<del> 216 [1, 1, 0, 1], [1, 0, 0, 0]</del>	<del> 248 [1, 1, 1, 1], [1, 0, 0, 0]</del>
<del> 25 [0, 0, 0, 1], [1, 0, 0, 1]</del>	<del> 57 [0, 0, 1, 1], [1, 0, 0, 1]</del>	89 [0, 1, 0, 1], [1, 0, 0, 1]	<del> 121 [0, 1, 1, 1], [1, 0, 0, 1]</del>	<del> 153 [1, 0, 0, 1], [1, 0, 0, 1]</del>	185 [1, 0, 1, 1], [1, 0, 0, 1]	<del> 217 [1, 1, 0, 1], [1, 0, 0, 1]</del>	<del> 249 [1, 1, 1, 1], [1, 0, 0, 1]</del>
<del> 26 [0, 0, 0, 1], [1, 0, 1, 0]</del>	58 [0, 0, 1, 1], [1, 0, 1, 0]	<del> 90 [0, 1, 0, 1], [1, 0, 1, 0]</del>	<del> 122 [0, 1, 1, 1], [1, 0, 1, 0]</del>	<del> 154 [1, 0, 0, 1], [1, 0, 1, 0]</del>	186 [1, 0, 1, 1], [1, 0, 1, 0]	<del> 218 [1, 1, 0, 1], [1, 0, 1, 0]</del>	<del> 250 [1, 1, 1, 1], [1, 0, 1, 0]</del>
27 [0, 0, 0, 1], [1, 0, 1, 1]	59 [0, 0, 1, 1], [1, 0, 1, 1]	<del> 91 [0, 1, 0, 1], [1, 0, 1, 1]</del>	<del> 123 [0, 1, 1, 1], [1, 0, 1, 1]</del>	155 [1, 0, 0, 1], [1, 0, 1, 1]	187 [1, 0, 1, 1], [1, 0, 1, 1]	<del> 219 [1, 1, 0, 1], [1, 0, 1, 1]</del>	<del> 251 [1, 1, 1, 1], [1, 0, 1, 1]</del>
<del> 28 [0, 0, 0, 1], [1, 1, 0, 0]</del>	<del> 60 [0, 0, 1, 1], [1, 1, 0, 0]</del>	92 [0, 1, 0, 1], [1, 1, 0, 0]	<del> 124 [0, 1, 1, 1], [1, 1, 0, 0]</del>	<del> 156 [1, 0, 0, 1], [1, 1, 0, 0]</del>	188 [1, 0, 1, 1], [1, 1, 0, 0]	<del> 220 [1, 1, 0, 1], [1, 1, 0, 0]</del>	<del> 252 [1, 1, 1, 1], [1, 1, 0, 0]</del>
<del> 29 [0, 0, 0, 1], [1, 1, 0, 1]</del>	<del> 61 [0, 0, 1, 1], [1, 1, 0, 1]</del>	93 [0, 1, 0, 1], [1, 1, 0, 1]	<del> 125 [0, 1, 1, 1], [1, 1, 0, 1]</del>	<del> 157 [1, 0, 0, 1], [1, 1, 0, 1]</del>	189 [1, 0, 1, 1], [1, 1, 0, 1]	<del> 221 [1, 1, 0, 1], [1, 1, 0, 1]</del>	<del> 253 [1, 1, 1, 1], [1, 1, 0, 1]</del>
<del> 30 [0, 0, 0, 1], [1, 1, 1, 0]</del>	<del> 62 [0, 0, 1, 1], [1, 1, 1, 0]</del>	94 [0, 1, 0, 1], [1, 1, 1, 0]	<del> 126 [0, 1, 1, 1], [1, 1, 1, 0]</del>	<del> 158 [1, 0, 0, 1], [1, 1, 1, 0]</del>	190 [1, 0, 1, 1], [1, 1, 1, 0]	<del> 222 [1, 1, 0, 1], [1, 1, 1, 0]</del>	<del> 254 [1, 1, 1, 1], [1, 1, 1, 0]</del>
31 [0, 0, 0, 1], [1, 1, 1, 1]	63 [0, 0, 1, 1], [1, 1, 1, 1]	95 [0, 1, 0, 1], [1, 1, 1, 1]	<del> 127 [0, 1, 1, 1], [1, 1, 1, 1]</del>	<del> 159 [1, 0, 0, 1], [1, 1, 1, 1]</del>	191 [1, 0, 1, 1], [1, 1, 1, 1]	<del> 223 [1, 1, 0, 1], [1, 1, 1, 1]</del>	<del> 255 [1, 1, 1, 1], [1, 1, 1, 1]</del>

# PRESETTING \_GROUPED CUBES

*Cutting down tilesets*



 032 :00100000 016 :00010000 128 :10000000 064 :01000000	 048 :00110000 144 :10010000 192 :11000000 096 :01100000	 112 :01110000 176 :10110000 208 :11010000 224 :11100000	 240 :11110000
 034 :00100010 017 :00010001 136 :10001000 068 :01000100	 051 :00110011 153 :10011001 204 :11001100 102 :01100110	 119 :01110111 187 :10111011 221 :11011101 238 :11101110	 114 :01110010 177 :10110001 216 :11011000 228 :11101000
 002 :00000010 001 :00000001 008 :00001000 004 :00000100	 003 :00000011 009 :00001001 012 :00001100 006 :00000110	 039 :00100111 027 :00011011 141 :10001101 078 :01001110	 015 :00001111
 114 :01110010 177 :10110001 216 :11011000 228 :11100100	 063 :00111111 159 :10011111 207 :11001111 111 :01101111	 243 :11110011 249 :11111001 252 :11111100 246 :11110110	 039 :00100111 027 :00011011 141 :10001101 078 :01001110
 047 :00101111 031 :00011111 143 :10001111 079 :01001111	 247 :11110111 251 :11111011 253 :11111101 254 :11111110	 251 :11110111 253 :11111011 254 :11111110 247 :11110111	

 103 :01100111 059 :00111011 157 :10011101 206 :11001110	 155 :10011011 205 :11001101 110 :01101110 055 :00110111	 147 :10010011 201 :11001001 108 :01101100 054 :00110110	 099 :01100011 057 :00111001 156 :10011100 198 :11000110
 035 :00100011 025 :00011001 140 :10001100 070 :01000110	 019 :00010011 137 :10001001 076 :01001100 038 :00100110	 116 :01110100 178 :10110010 209 :11010001 232 :11101000	 184 :10111000 212 :11010100 226 :11100010 113 :01110001
 050 :00110010 145 :10010001 200 :11001000 100 :01100100	 049 :00110001 152 :10011000 196 :11000100 098 :01100010	 118 :01110110 179 :10110011 217 :11011001 236 :11101100	 185 :10111001 220 :11011100 230 :11100110 115 :01110011
 103 :01100111 059 :00111011 157 :10011101 206 :11001110	 155 :10011011 205 :11001101 110 :01101110 055 :00110111		

*\*Each tileset owns unique configuring ID of the same geometry.*

Horizontal symmetry and orthogonal rotation in four directions "equivalent upto tiles"

> Leave 26 unique tilesets.

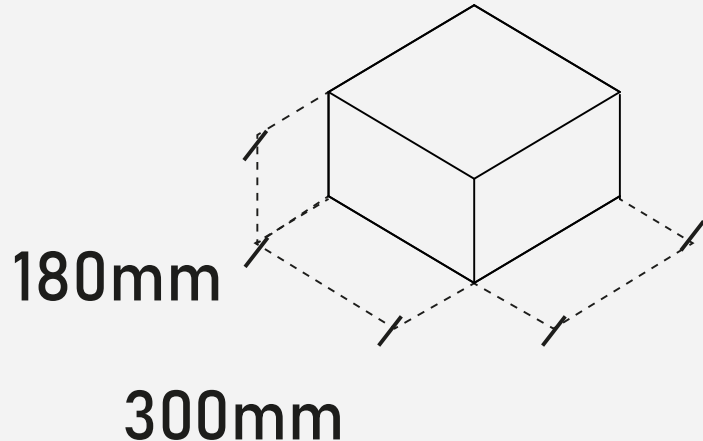
# PRESETTING \_DIMENSION

*Module dimension defining*

General rule for stairs stride

$$2 \text{ Risers} + 1 \text{ Tread} = 660 \text{ mm}$$

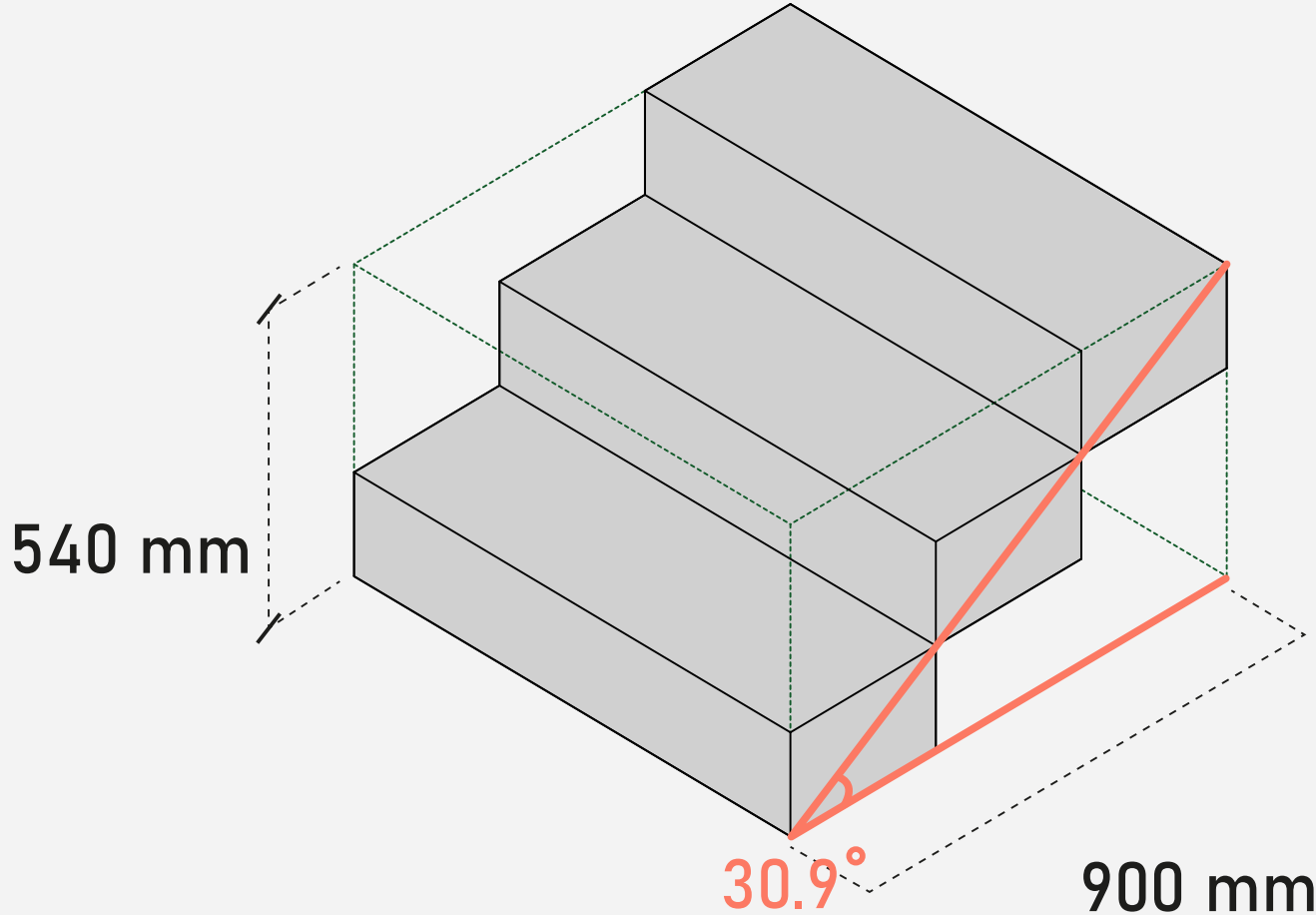
$$2 * 180 + 300 = 660 \text{ mm}$$



The width and height as multiplication of stair stride

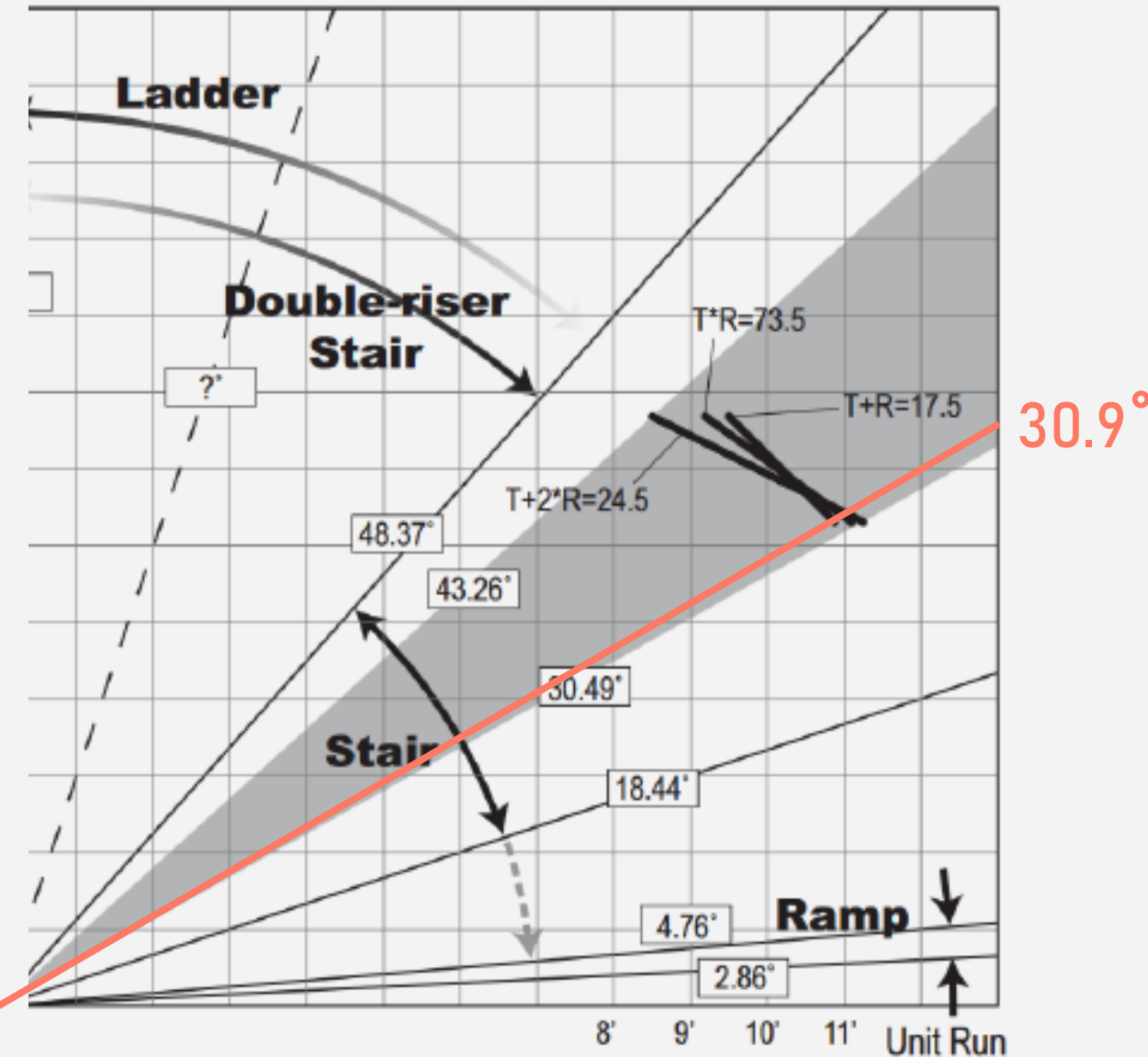
$$180 * 3 = 540$$

$$300 * 3 = 900$$



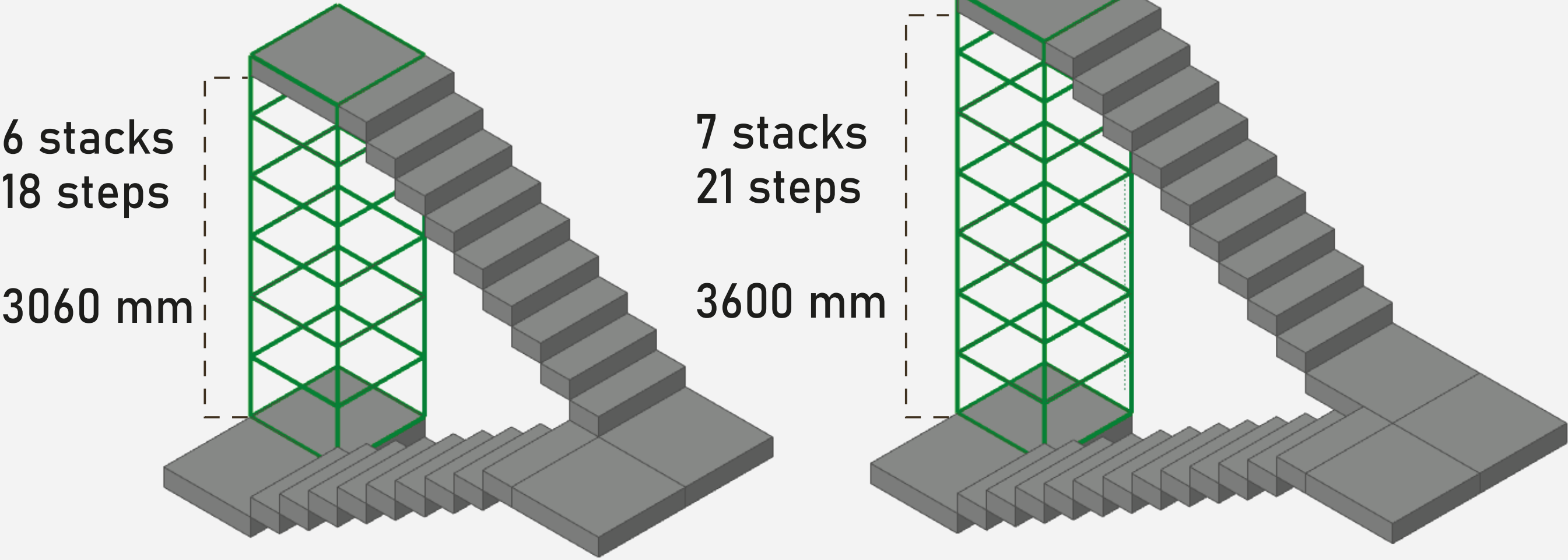
$$\text{Riser } 180 / \text{tread } 300 = 0.6$$

$$= \tan^{-1}(0.6) = 30.9^\circ$$

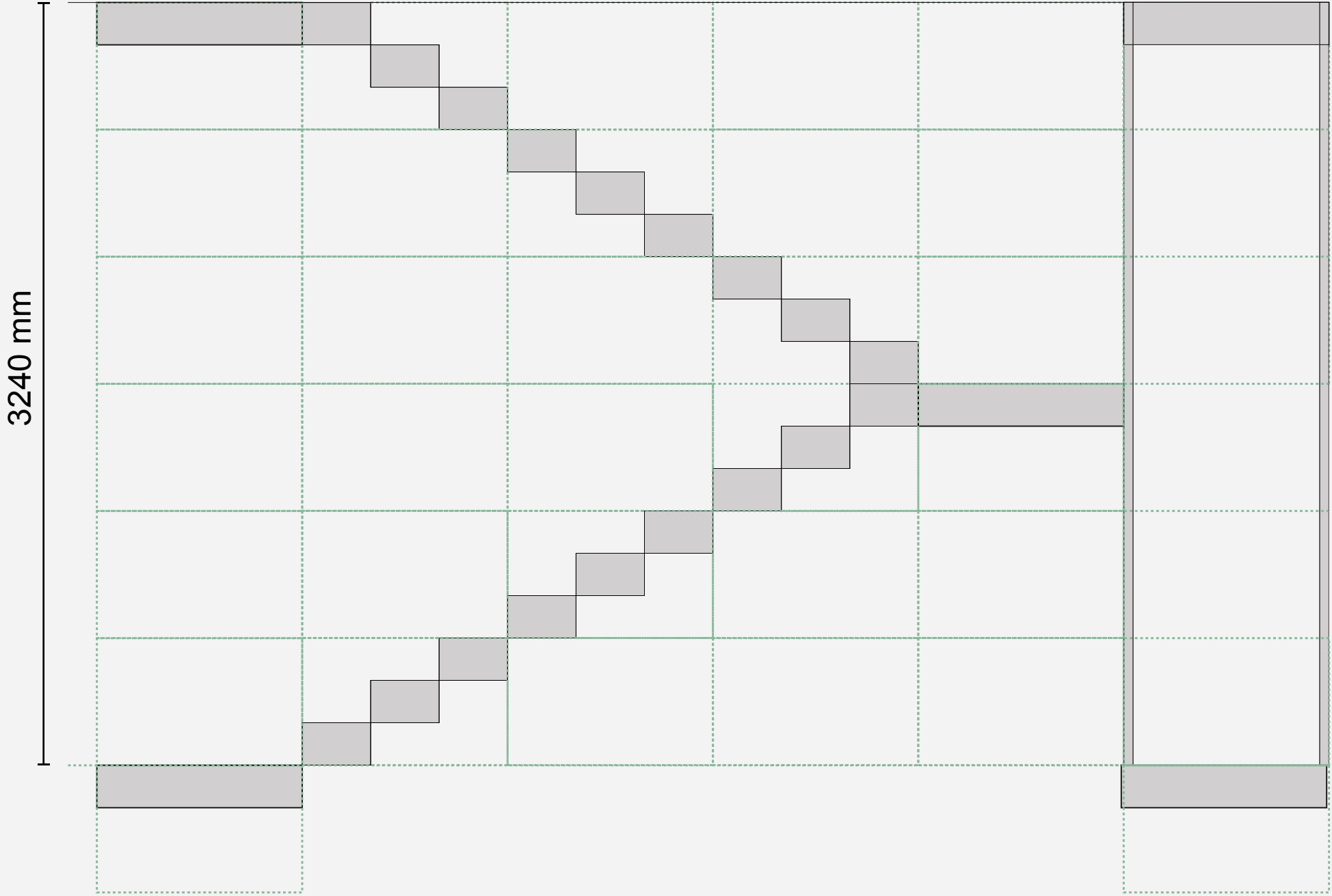


# PRESETTING \_DIMENSION

*Module dimension defining*



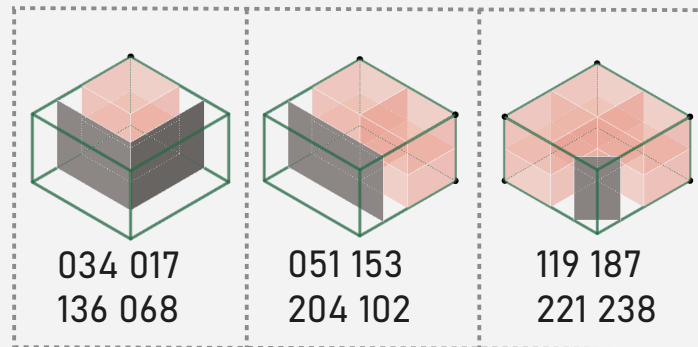
6 of the modules create a  
valid floor-to-floor height of 3240 mm  
(floor to ceiling height of 3060 mm)



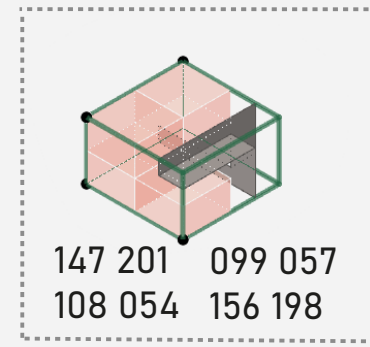
# SURFACE TILESETS

Standard  
Surface tilesets

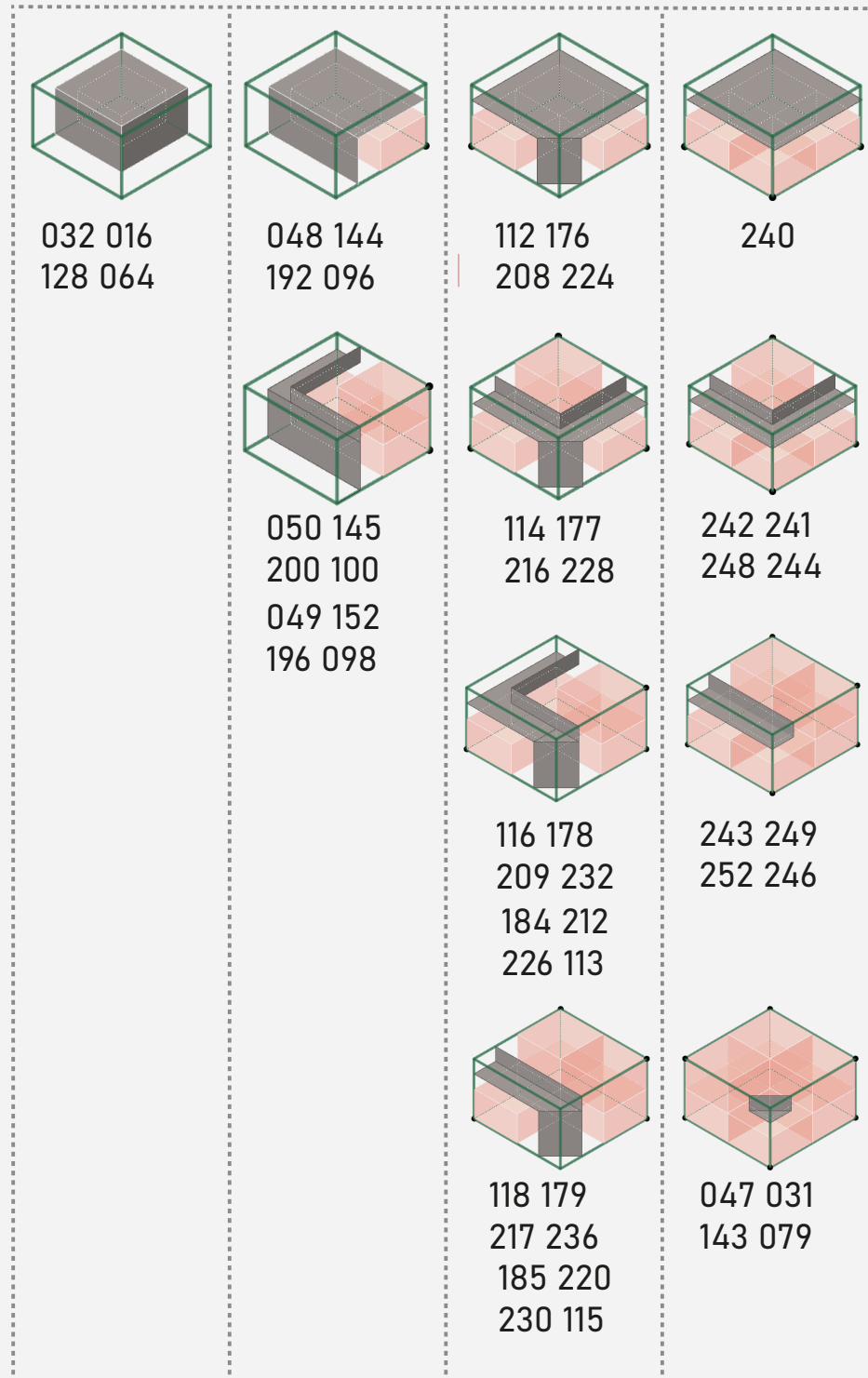
## Wall tiles



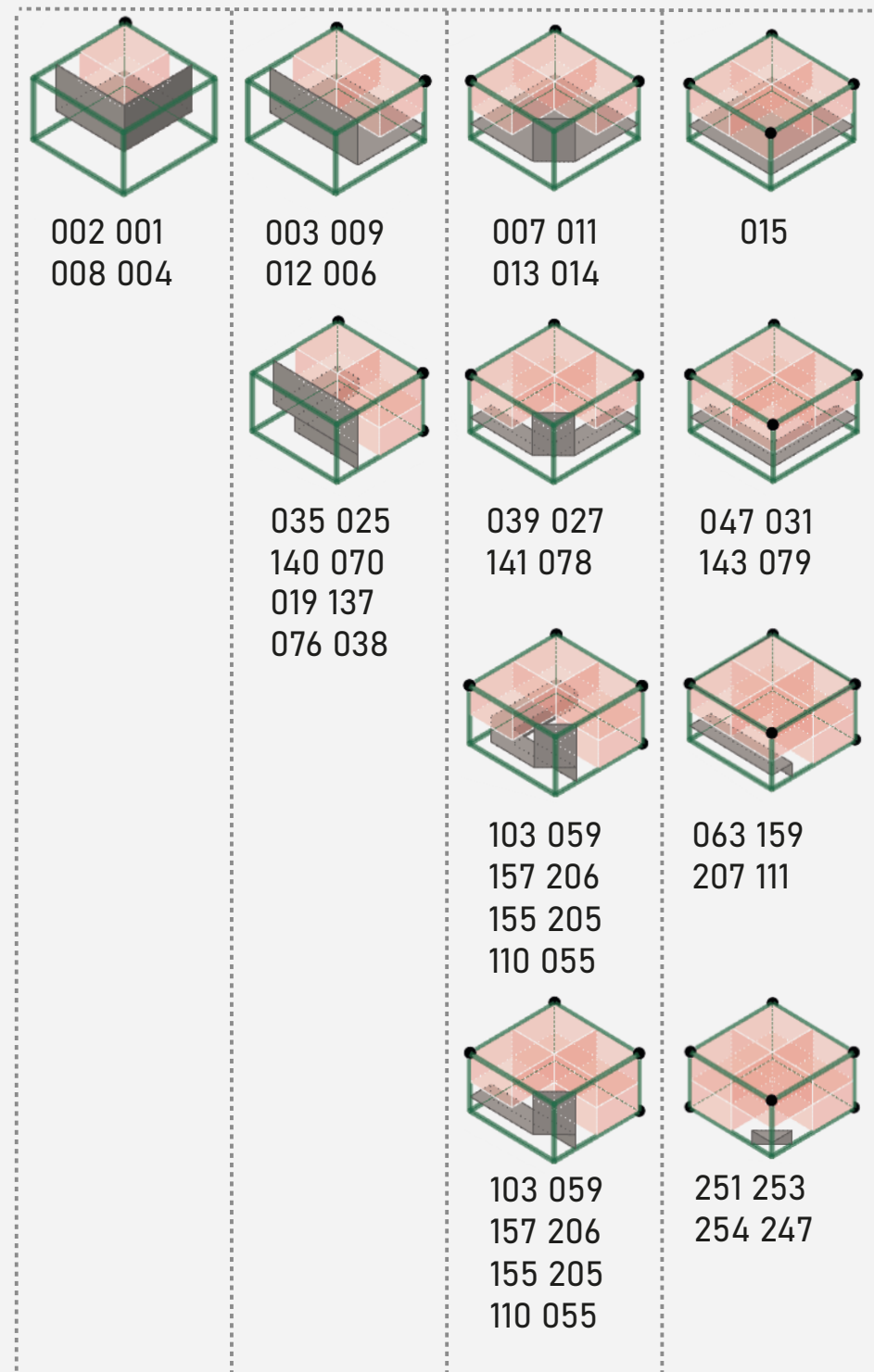
## Extra tile



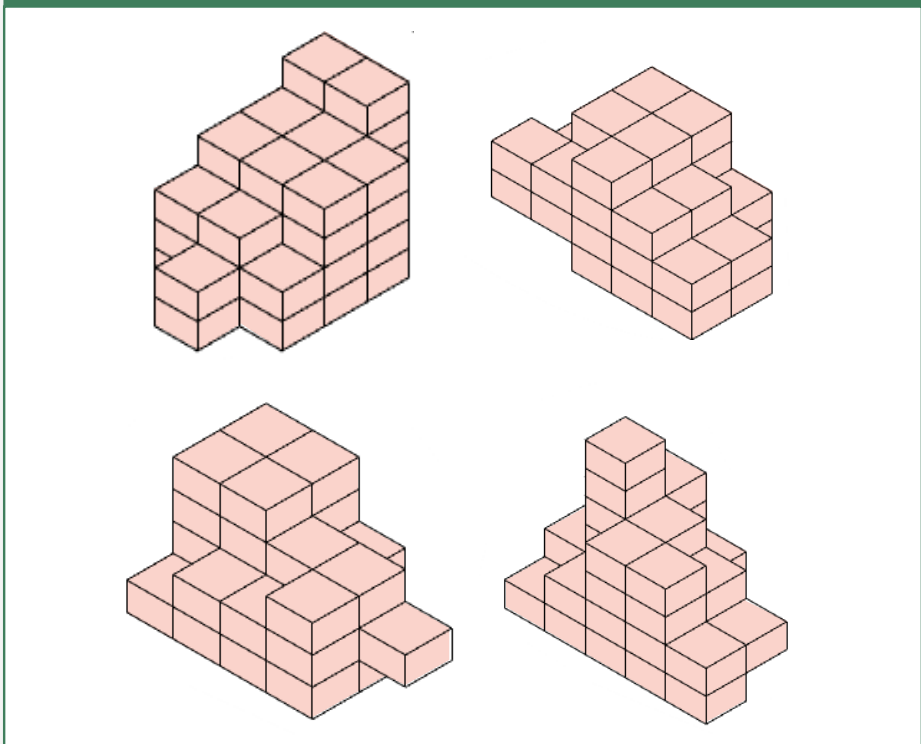
## Roof tiles



## Floor tiles

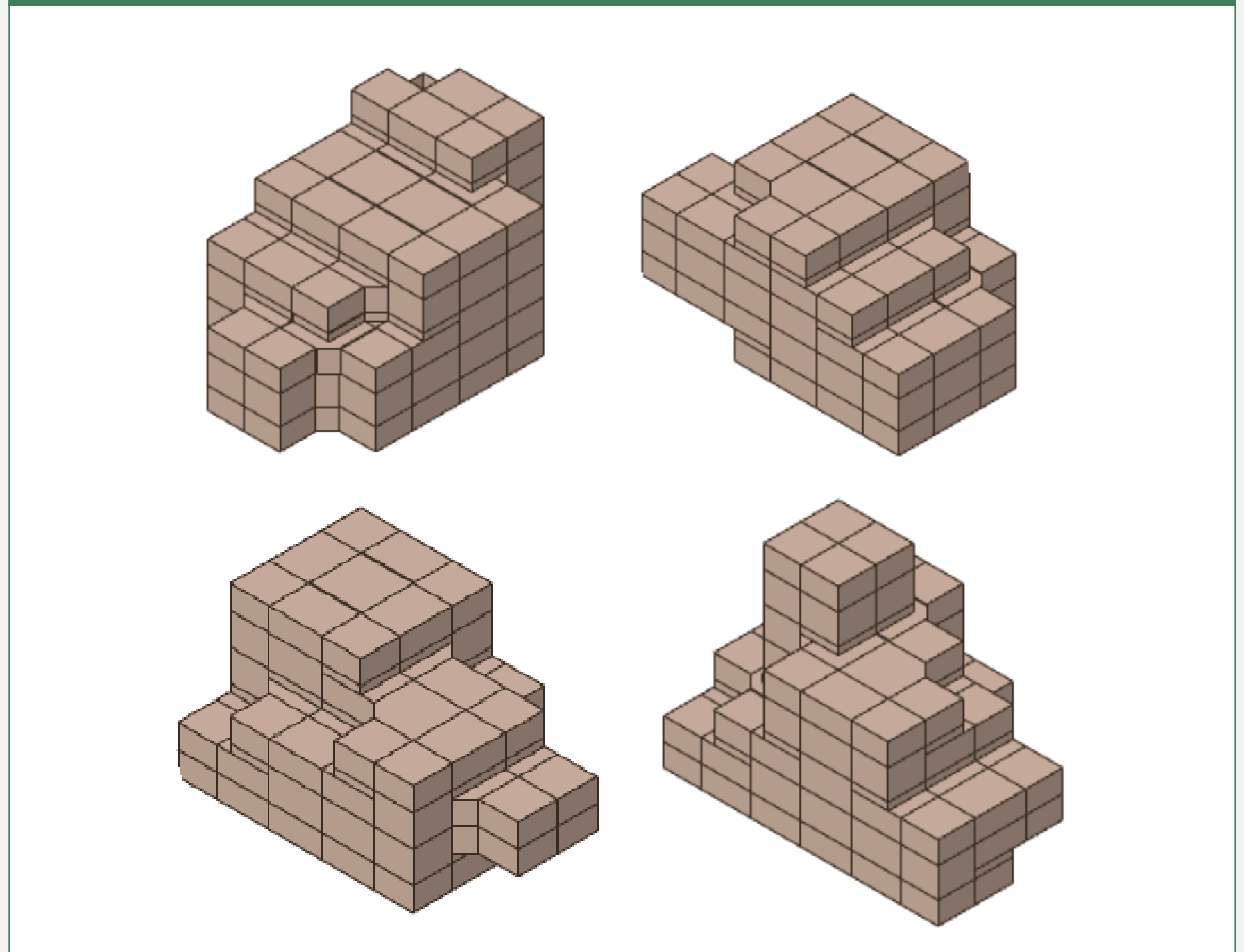


## Input



**BMC**

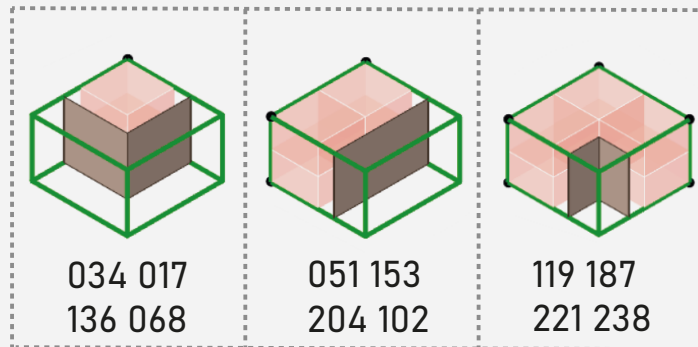
## Output



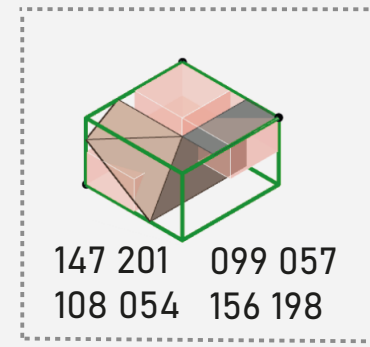
# SURFACE TILESETS

*Differentiated  
floor and roof  
tilesets*

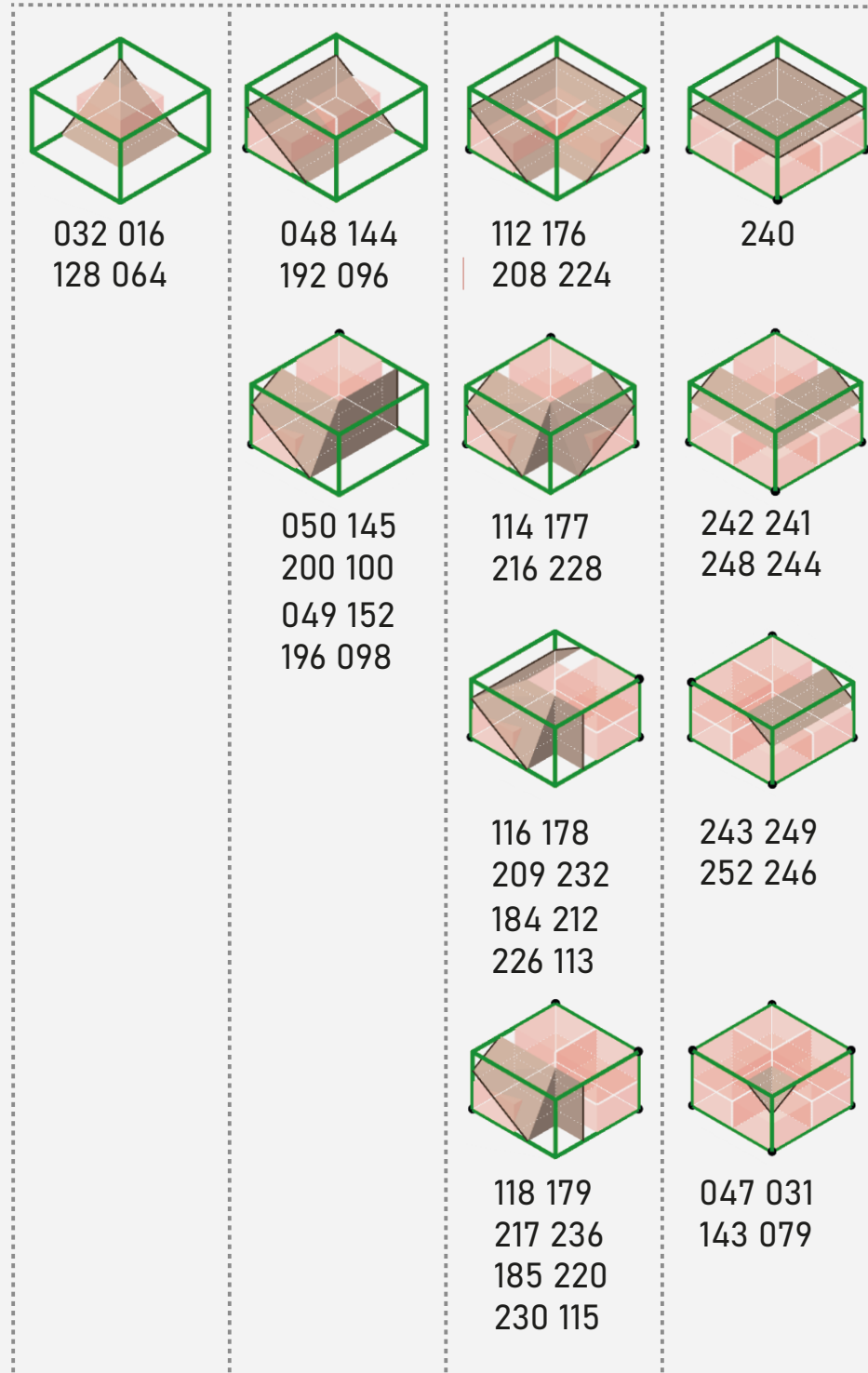
## Wall tiles



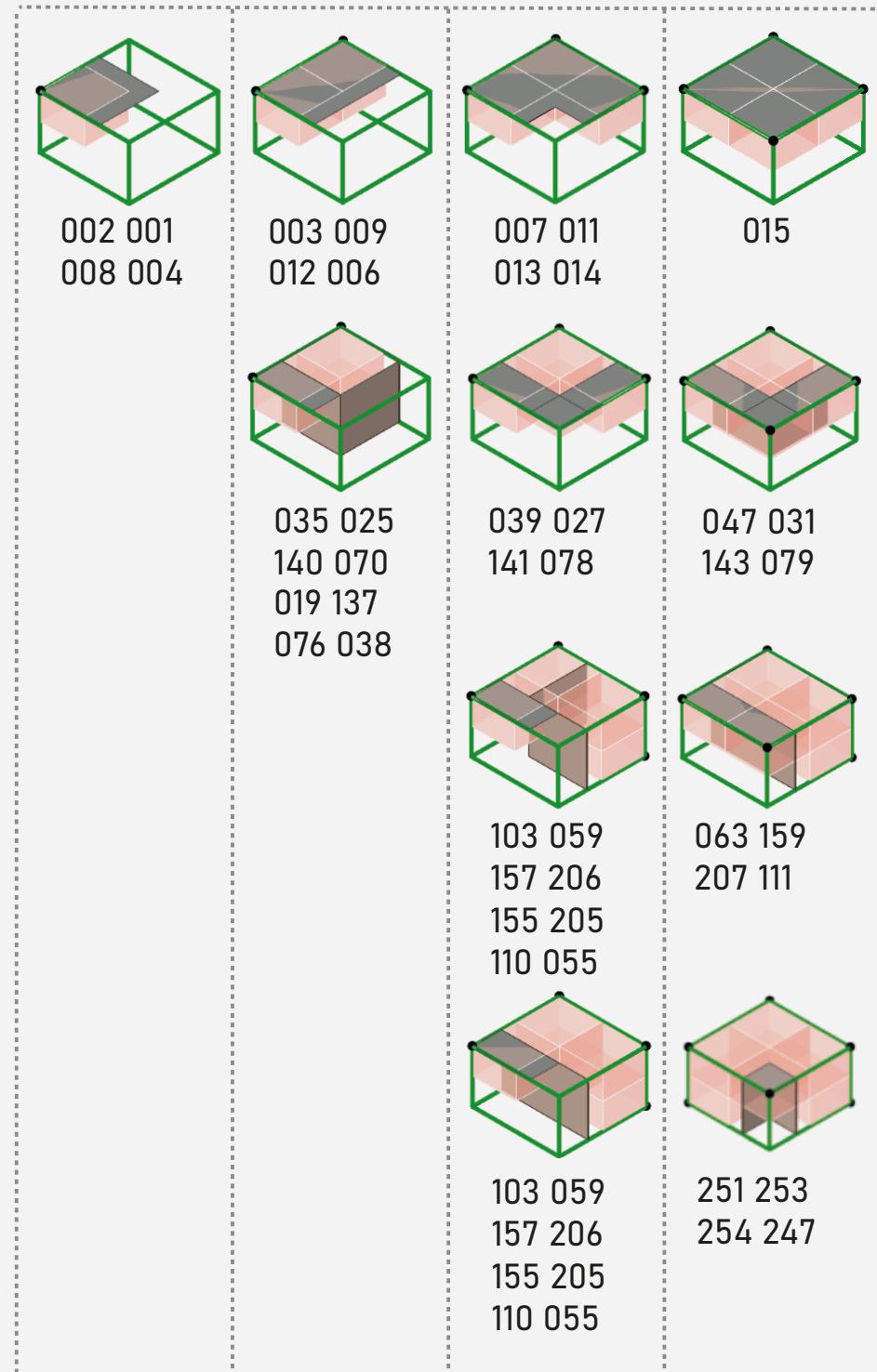
## Extra tile



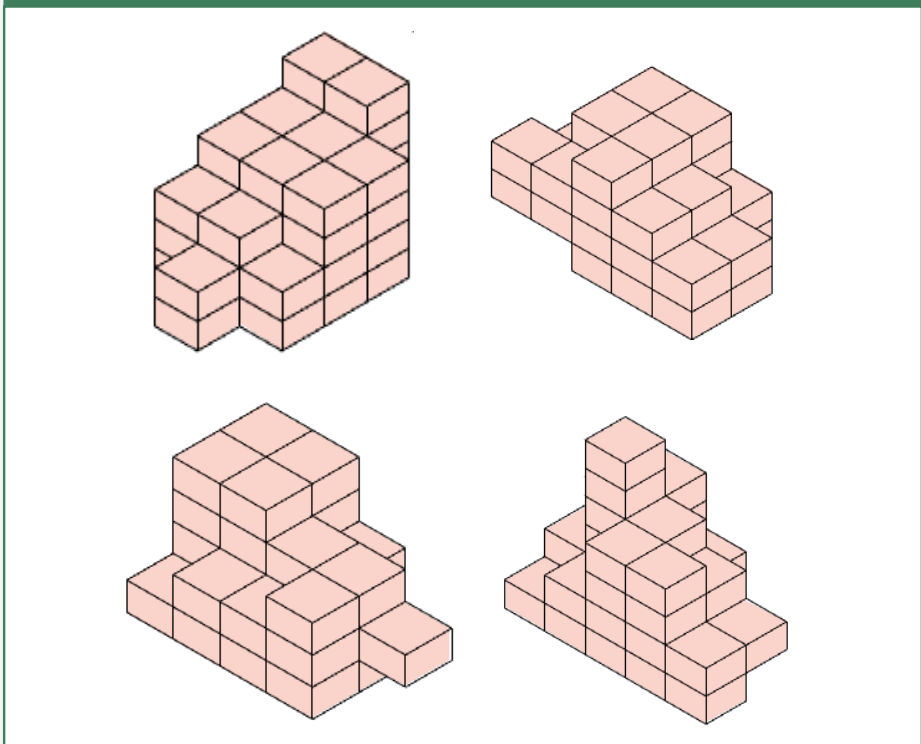
## Roof tiles



## Floor tiles

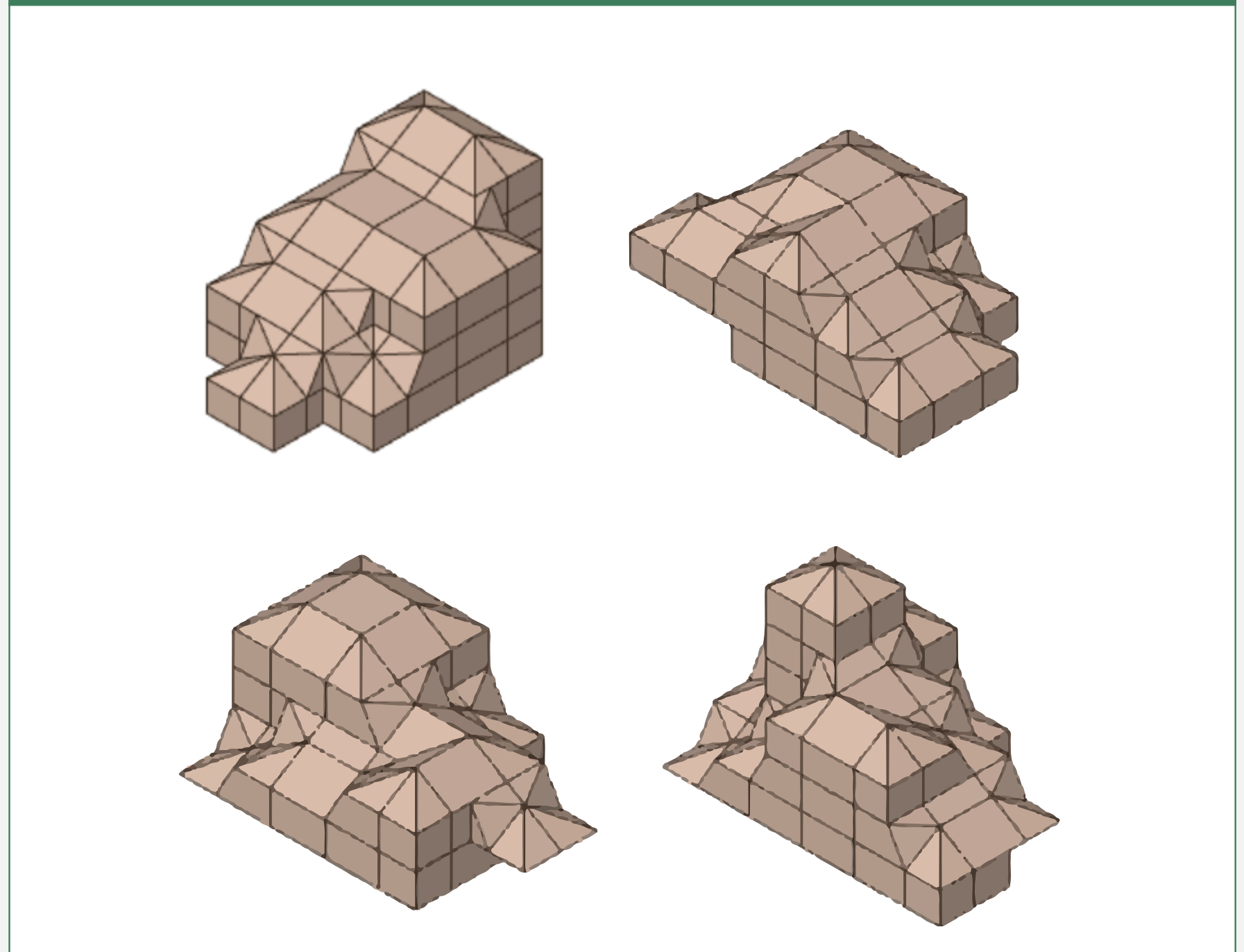


## Input



**BMC**

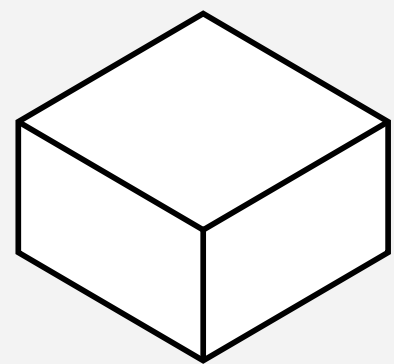
## Output



# ARCHITECTURAL TILESETS

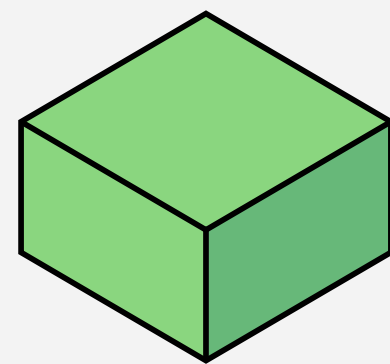
*Voxel function*

To achieve the architecture variations,  
multiple function of voxels need to be placed.



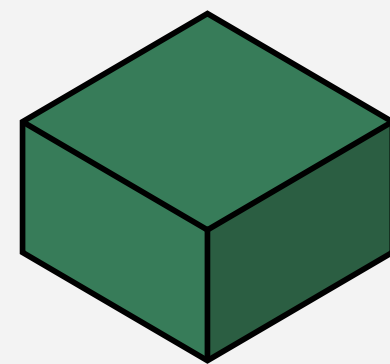
**Base Voxel**

Octant voxel ID : 1



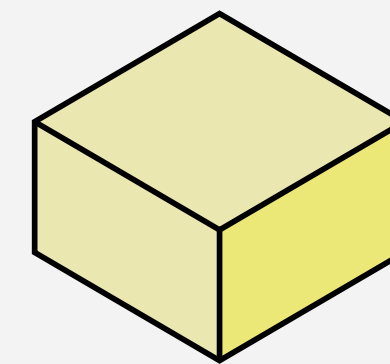
**Floor Voxel**

Octant voxel ID : 2



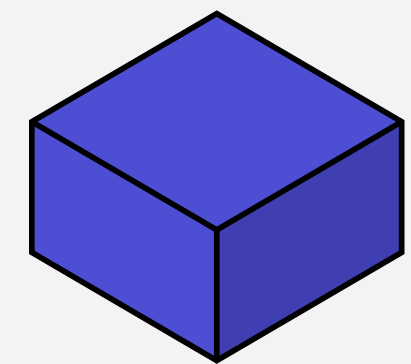
**Foundation Voxel**

Octant voxel ID : 3



**Opening Voxel**

Octant voxel ID : 4



**Window Voxel**

Octant voxel ID : 5



# ARCHITECTURAL TILESETS

*Binary to Heximal*

The possible options for sub-voxels are six (0,1,2,3,4 and 5).

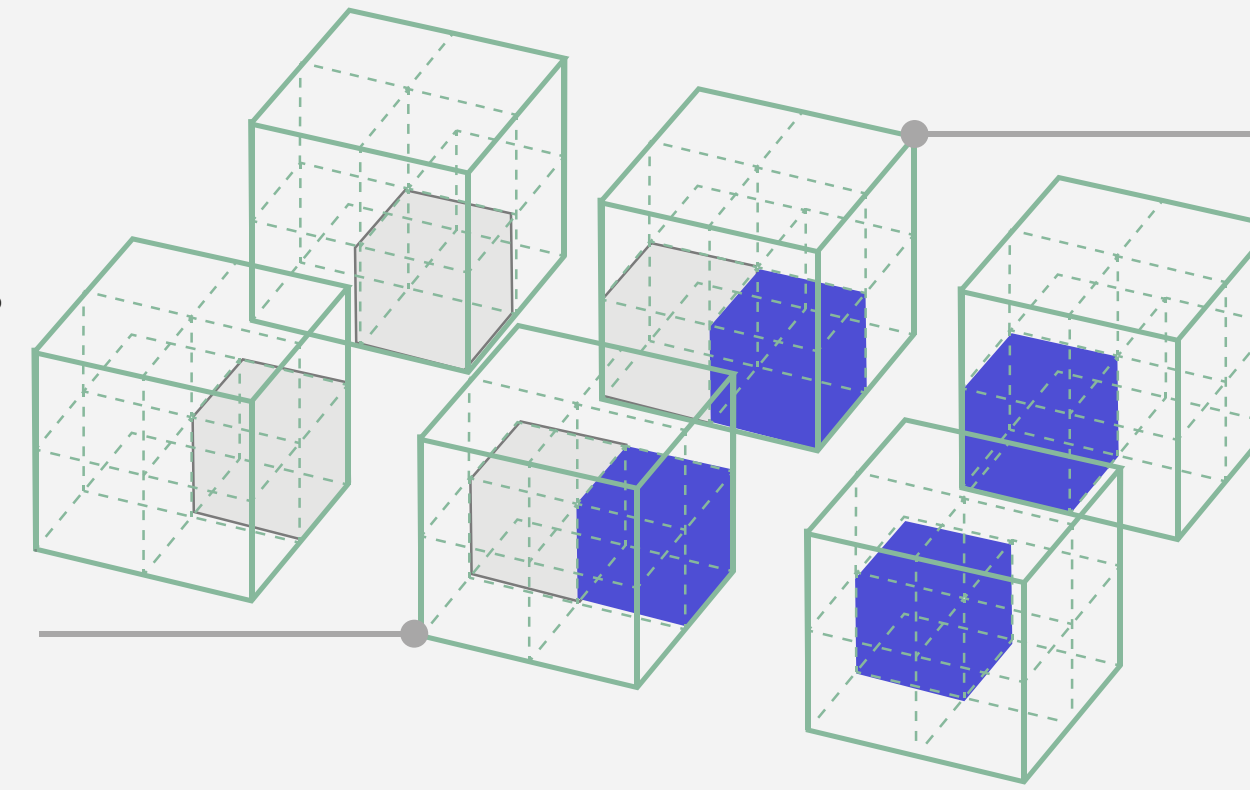
Binary to heximal (6 elements)

In 8-digit the possible cube configuration increases 256 to 1679615 ( $6^8$ ).

$$[00010000]_6 = 1296_{10}$$

$$[00100000]_6 = 7776_{10}$$

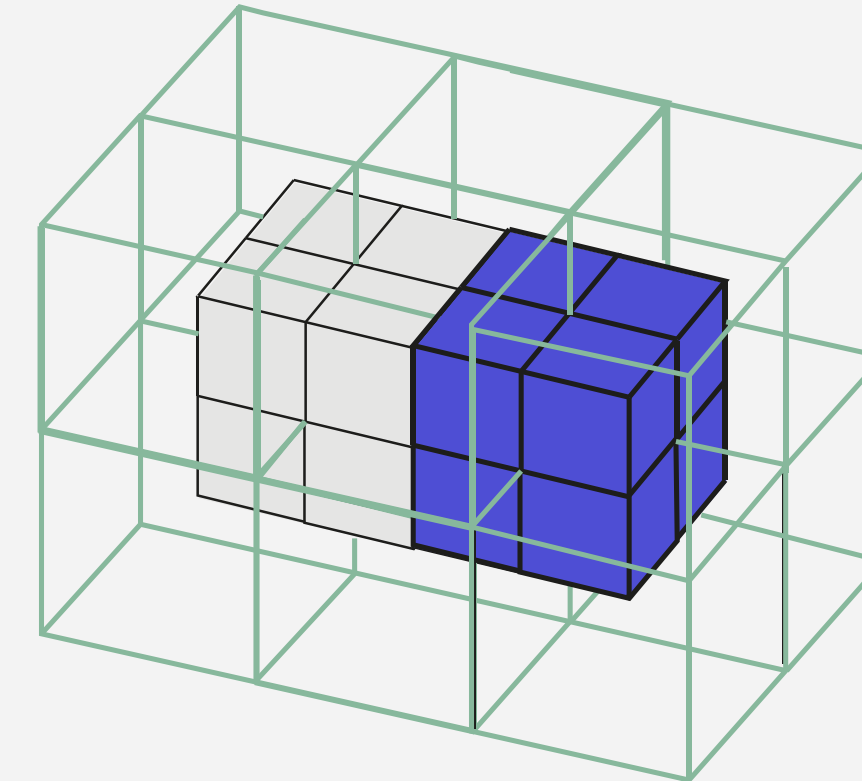
$$[10500000]_6 = 295488_{10}$$



$$[01050000]_6 = 53136_{10}$$

$$[05000000]_6 = 233280_{10}$$

$$[50000000]_6 = 1399680_{10}$$



**Possible configuration**

Binary (256)

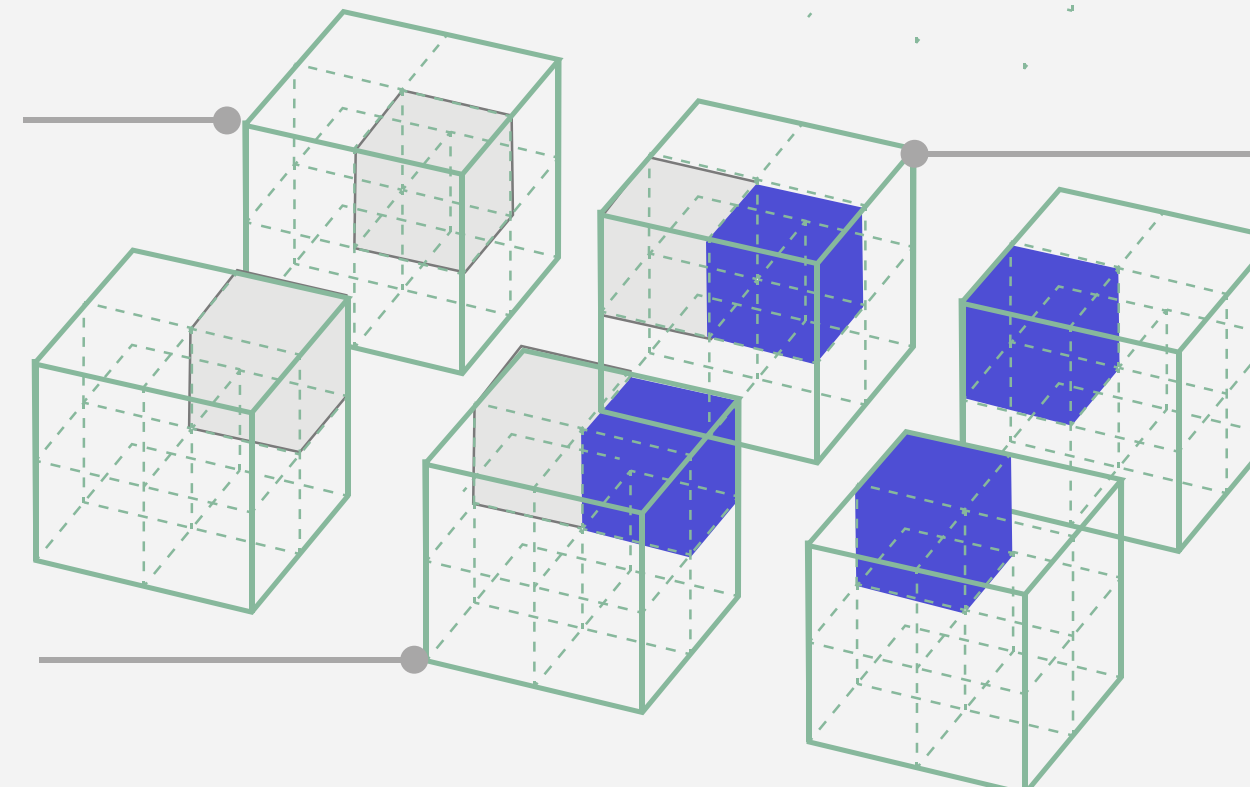


Heximal (1679616)

$$[00000001]_6 = 1_{10}$$

$$[00000010]_6 = 6_{10}$$

$$[00001050]_6 = 246_{10}$$



$$[00000105]_6 = 41_{10}$$

$$[00000500]_6 = 180_{10}$$

$$[00005000]_6 = 1080_{10}$$

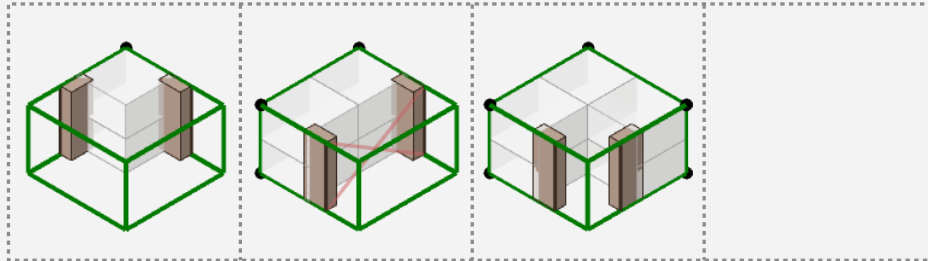
# ARCHITECTURAL TILESETS

## Architectural tiles (B)

The 26 base tiles classified  
in different functions

Voxel ID\_1

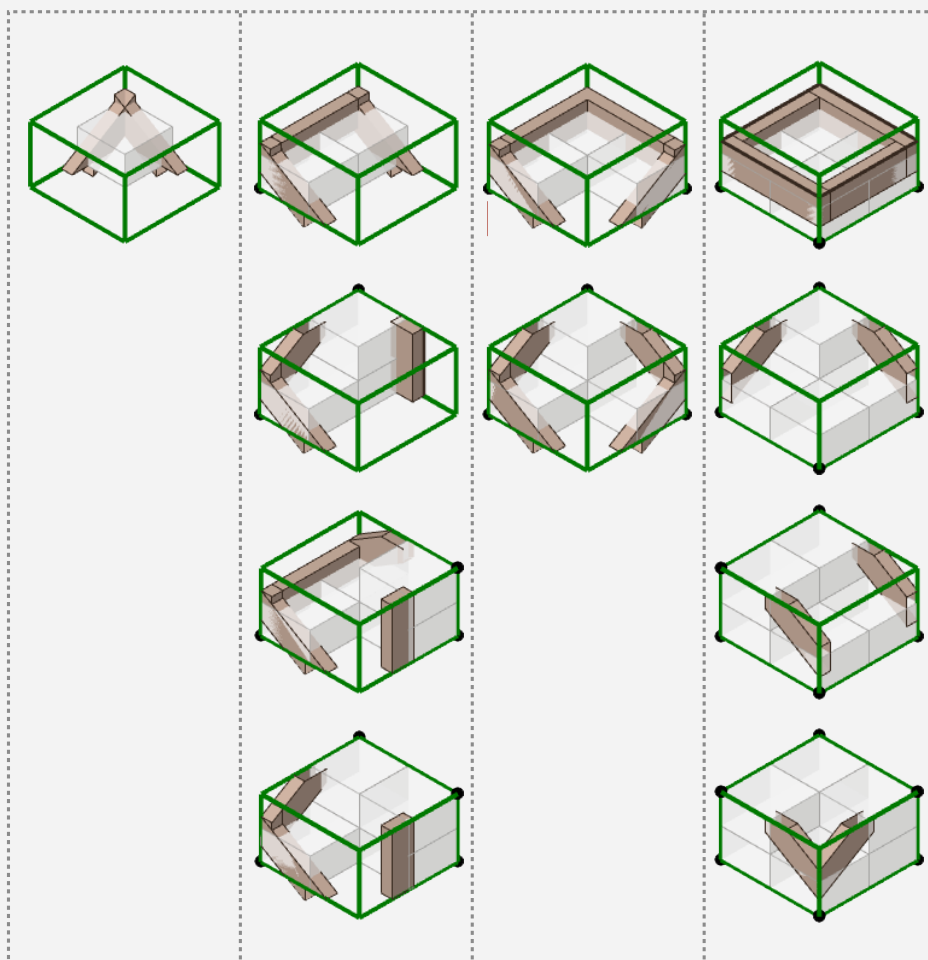
### Wall frame tiles



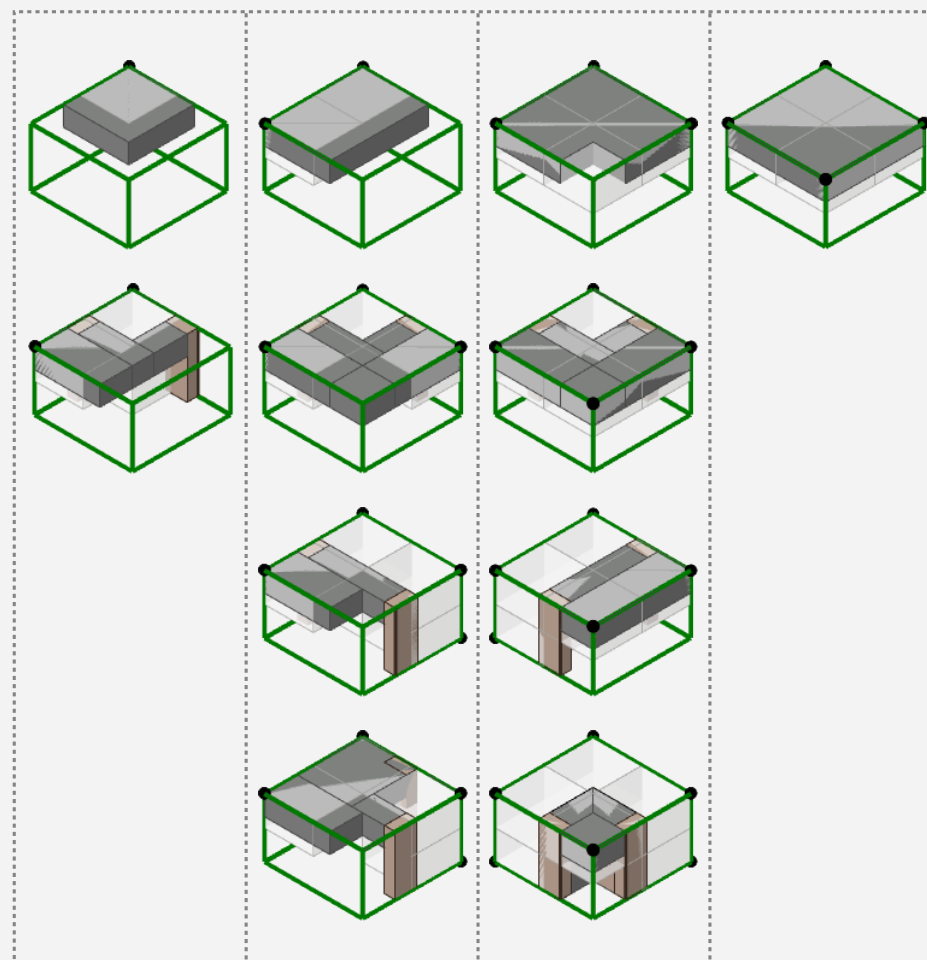
### Extra tile



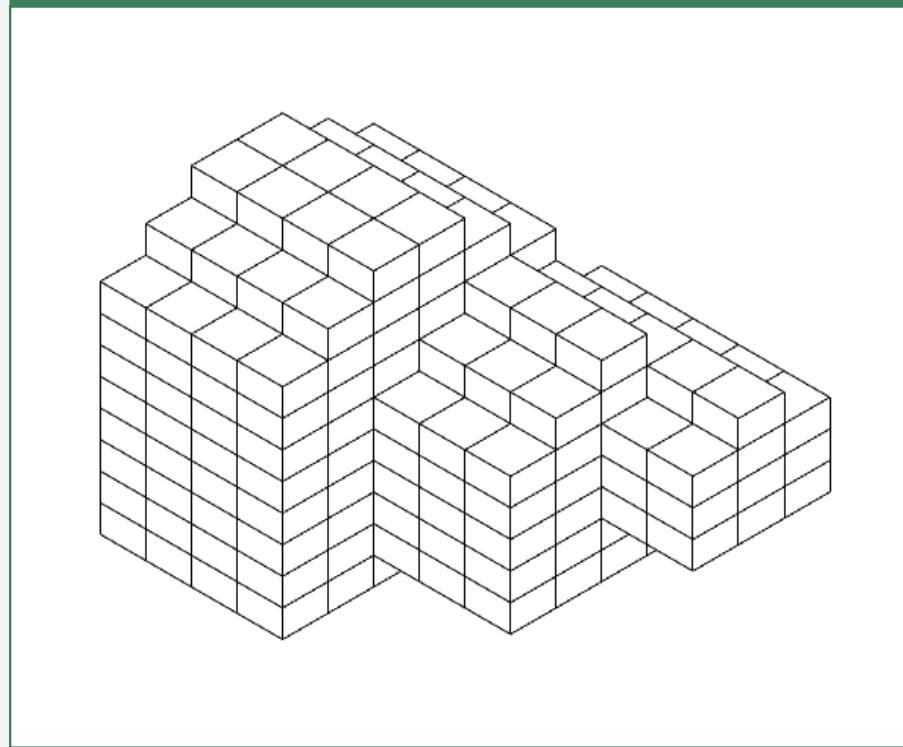
### Top frame tiles



### Bottom frame tiles

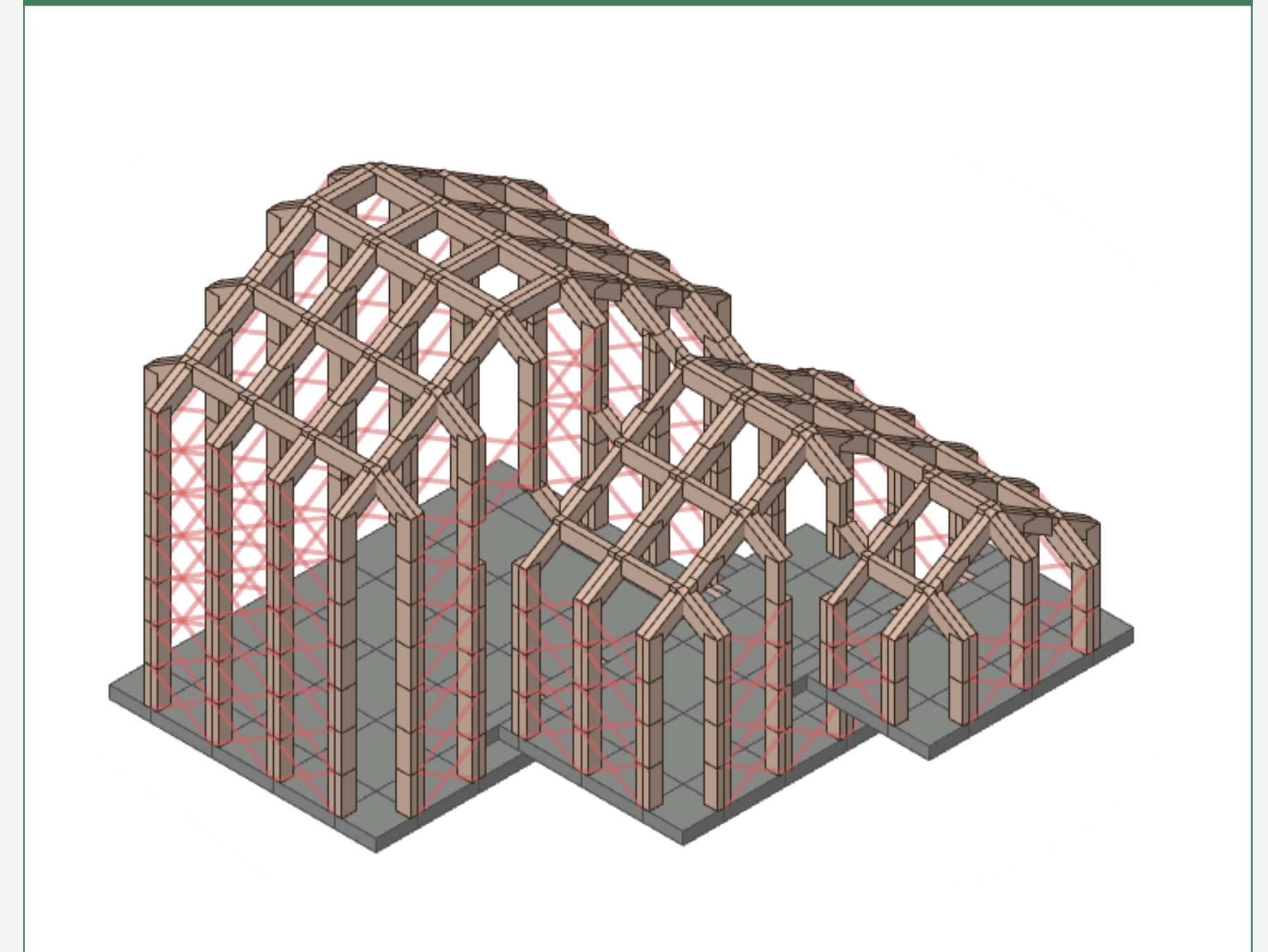


Input



HOUSCAPER

Output



# ARCHITECTURAL TILESETS

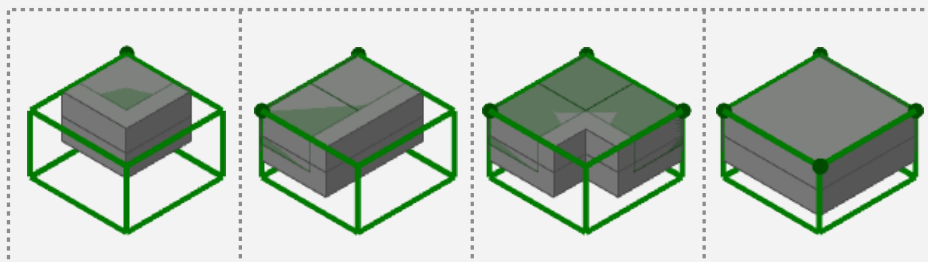
Foundation tiles

Floor tiles

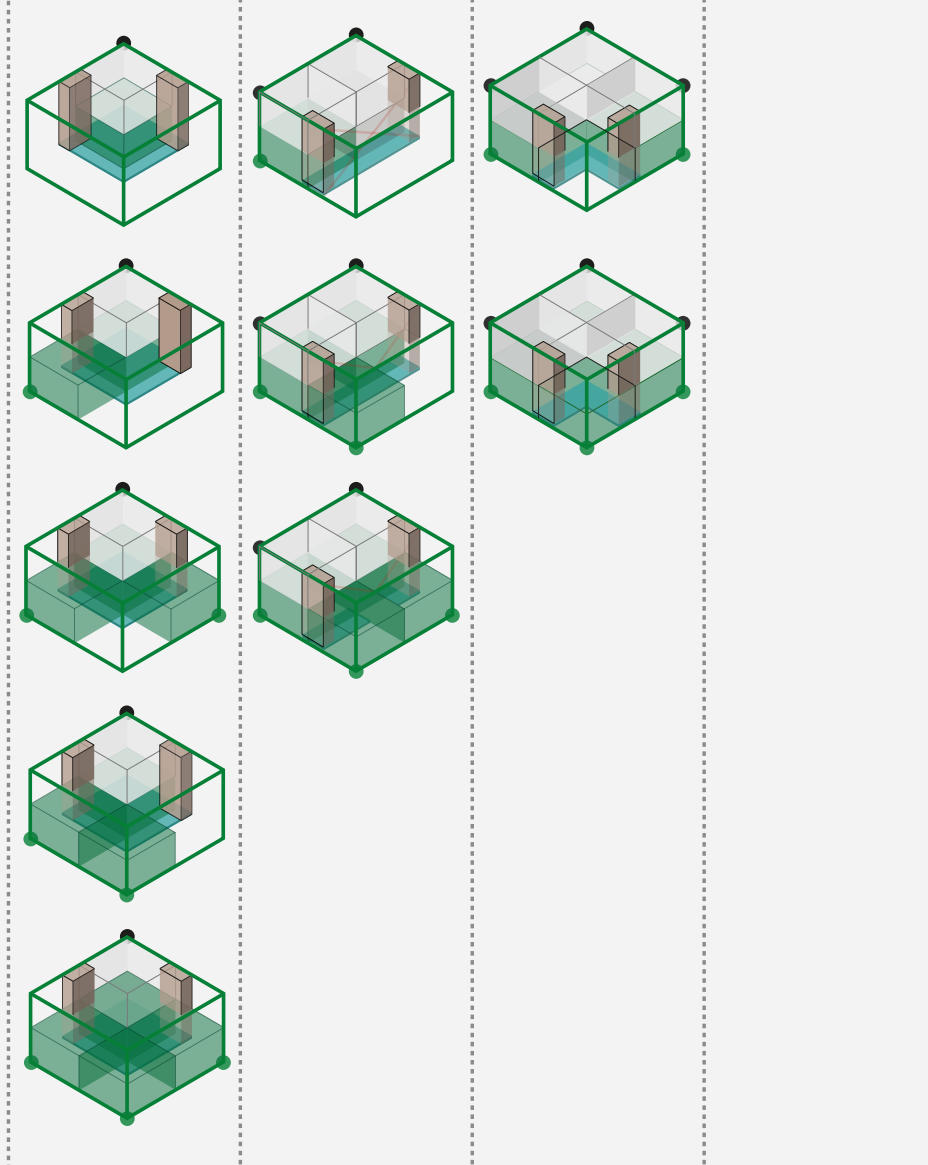
Voxel ID\_3

Voxel ID\_2

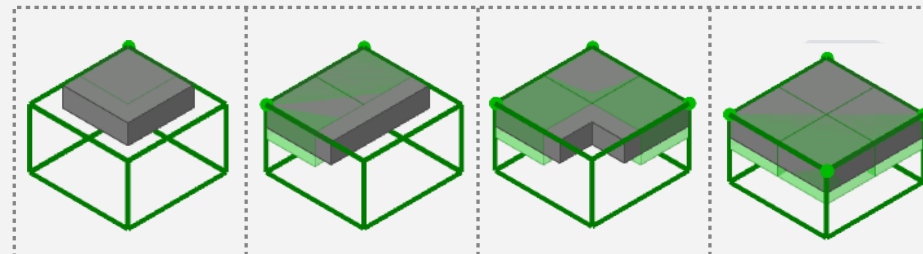
Foundation tiles



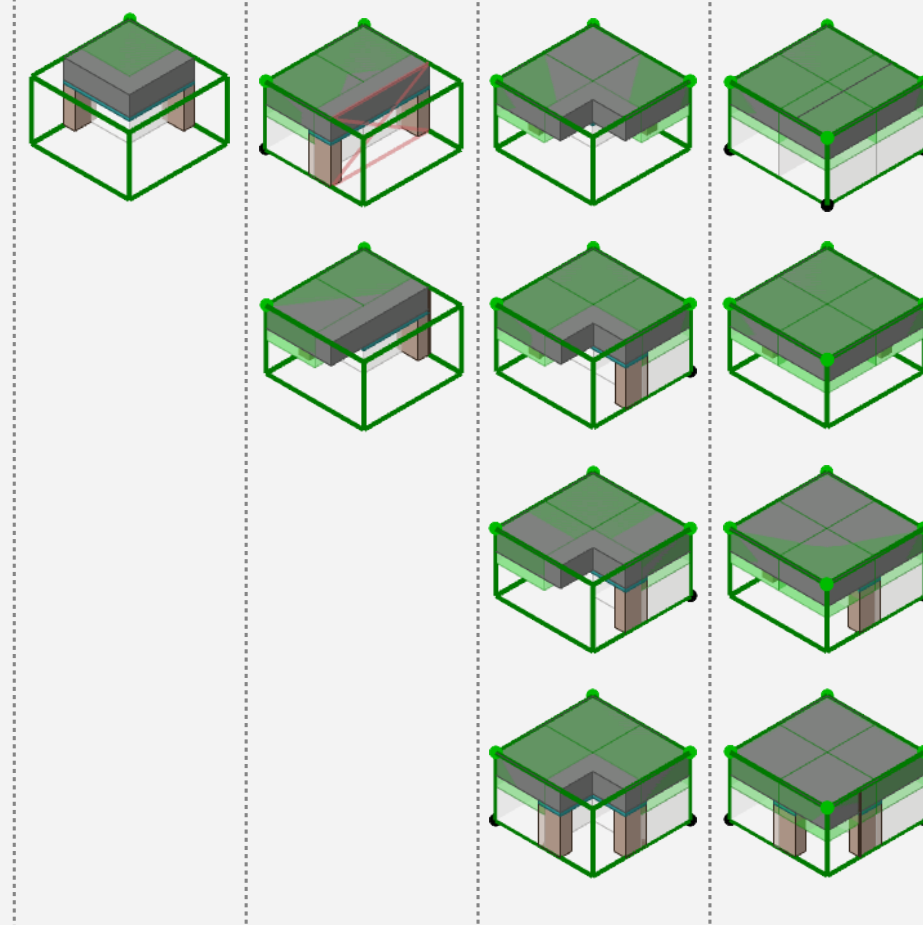
Foundation to frame (Up)



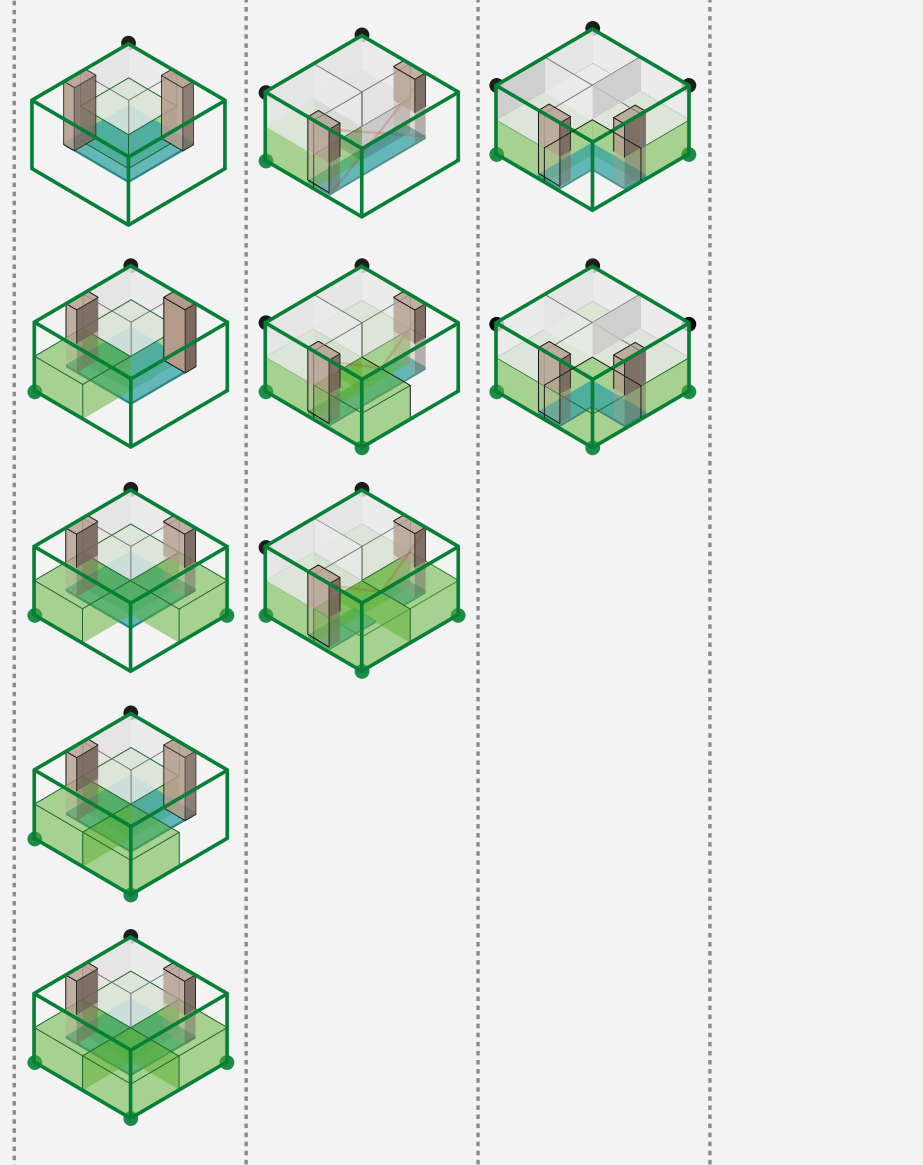
Floor tiles



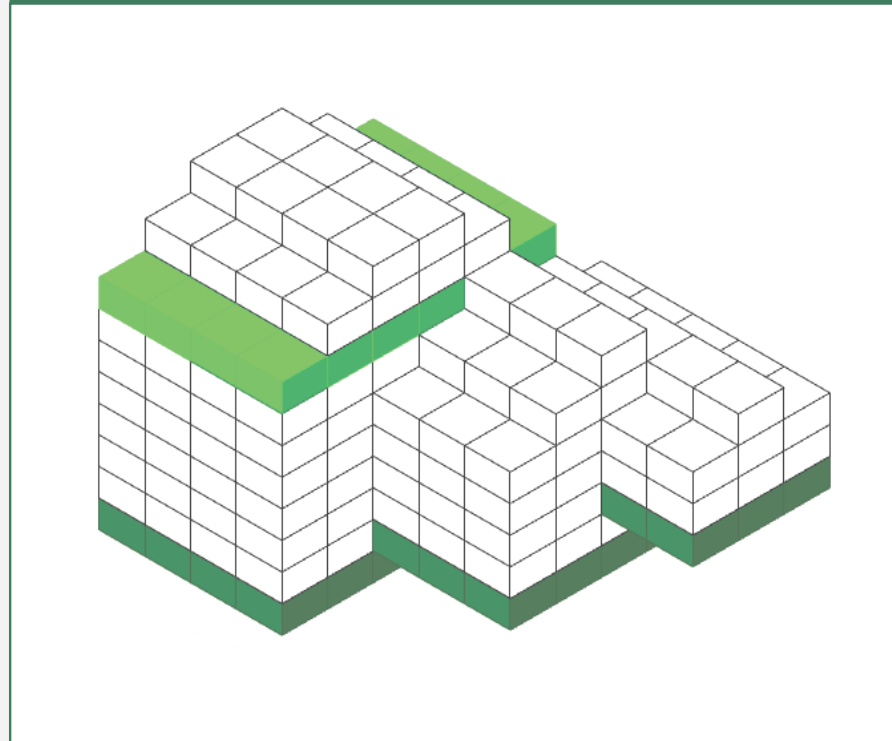
Frame connection (Down)



Floor to frame (Up)

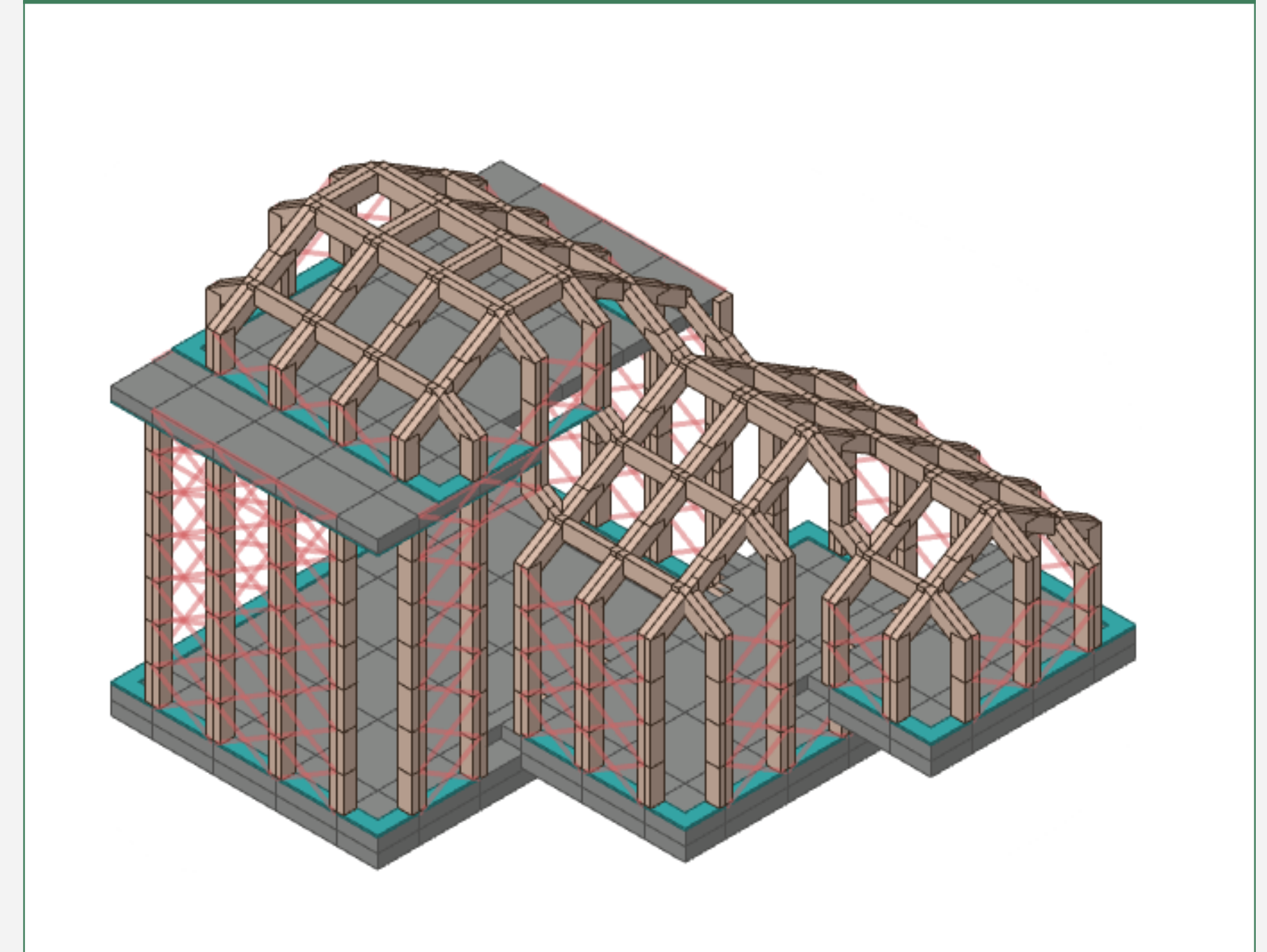


Input



HOUSCAPER

Output



# ARCHITECTURAL TILESETS

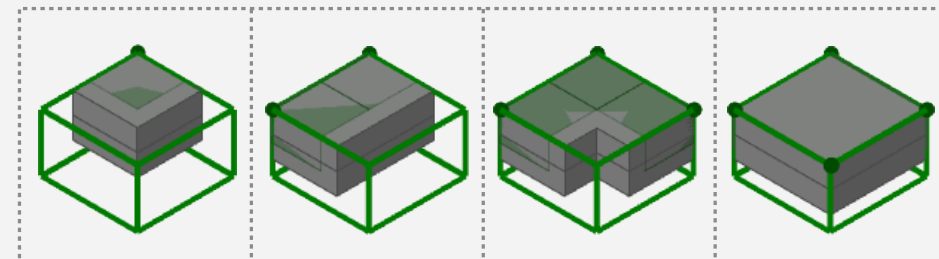
*Foundation tiles*

*Floor tiles*

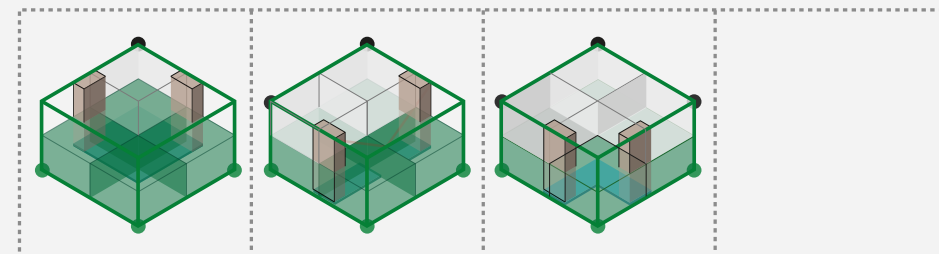
Voxel ID\_3

Voxel ID\_2

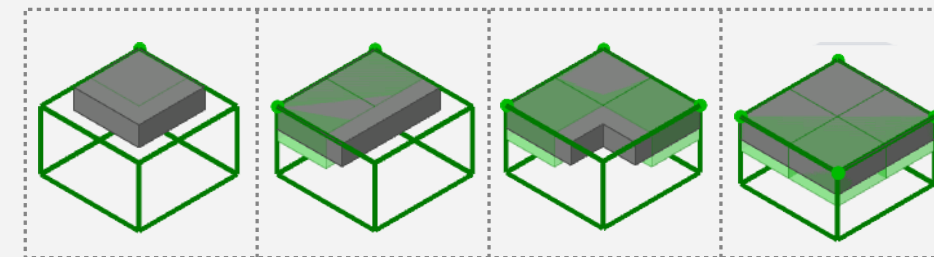
Foundation tiles



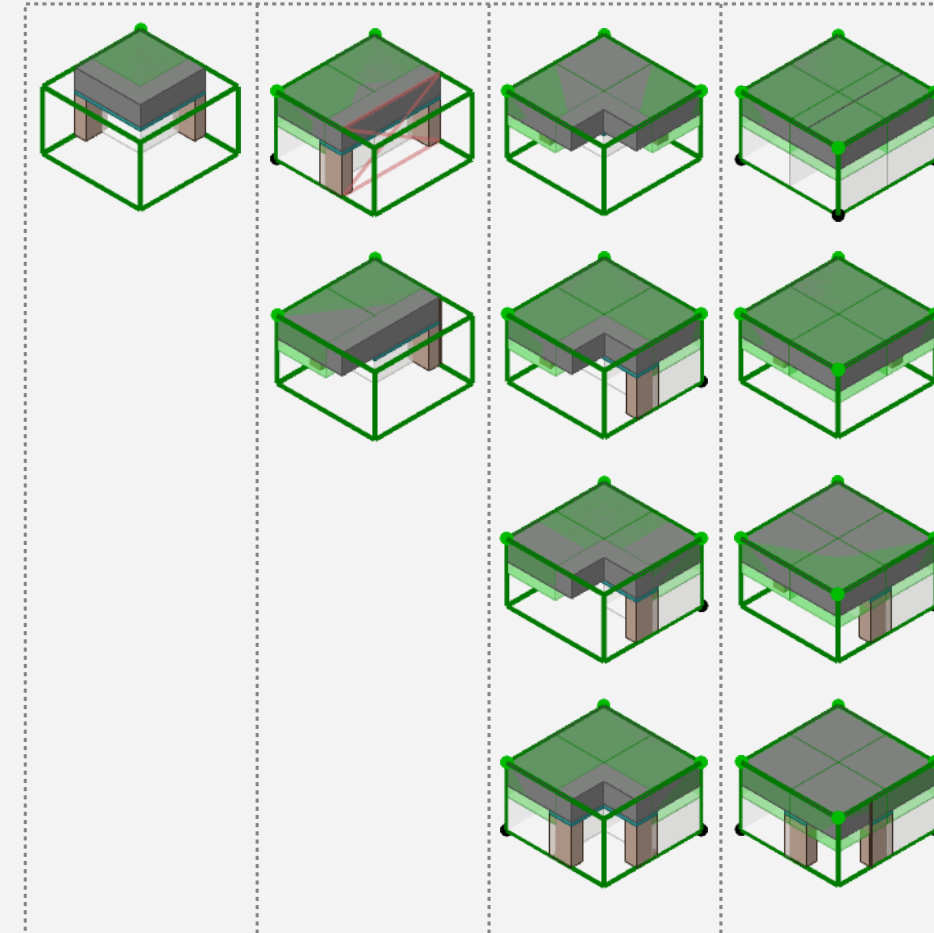
Foundation to frame connection (Up)



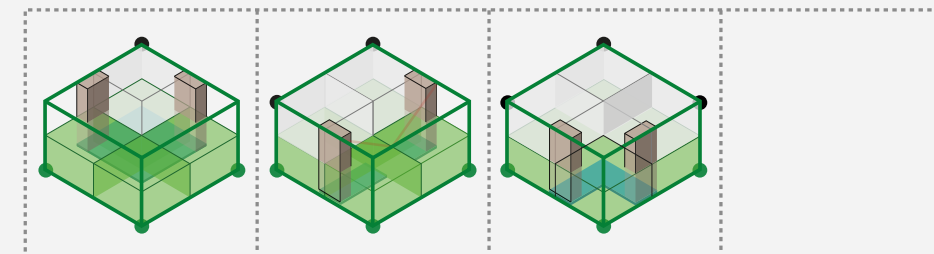
Floor tiles



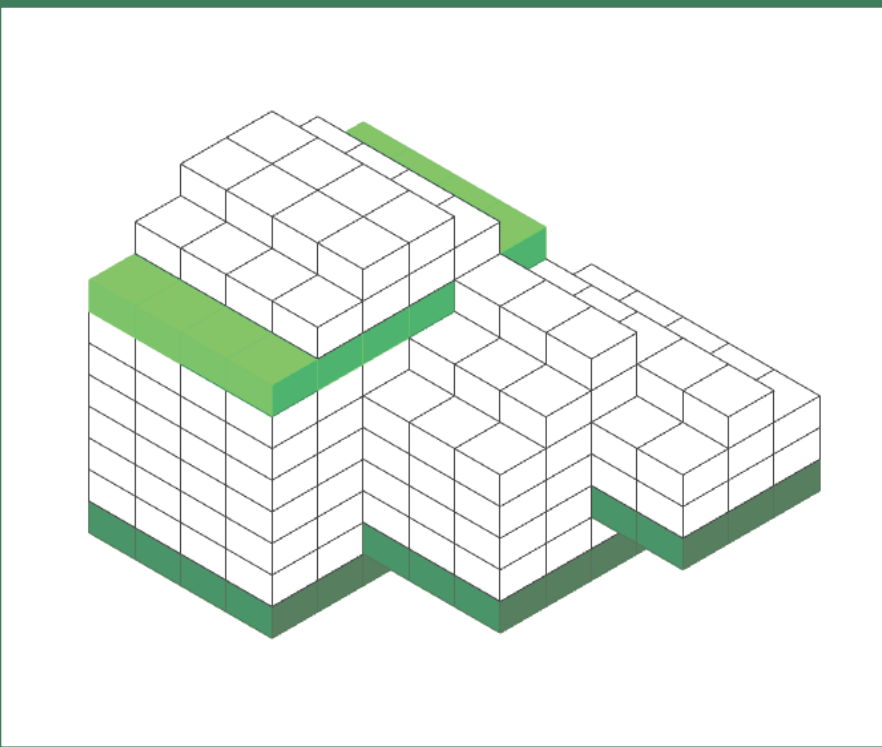
Frame connection (Down)



Floor to frame connection (Up)

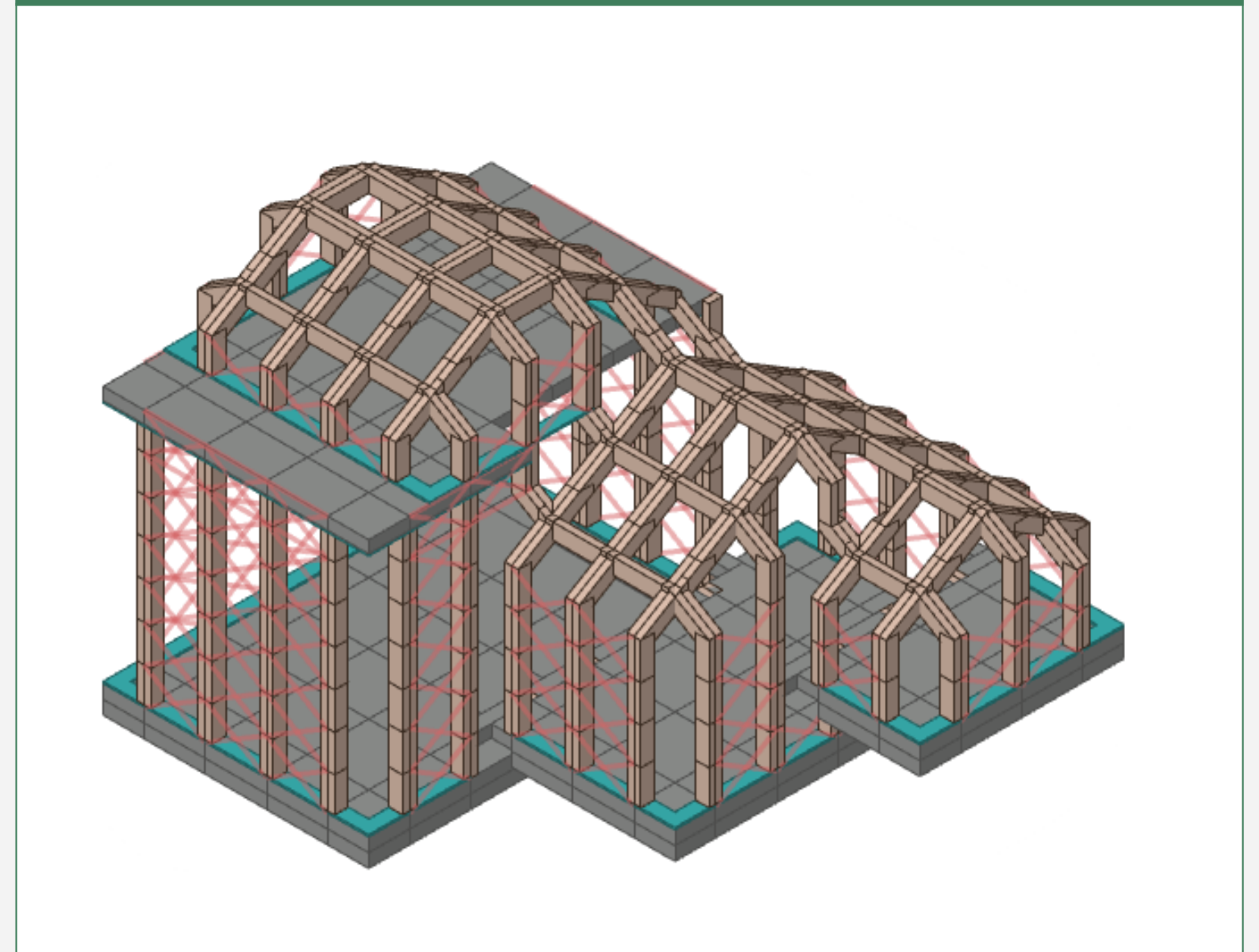


Input



HOUSCAPER

Output



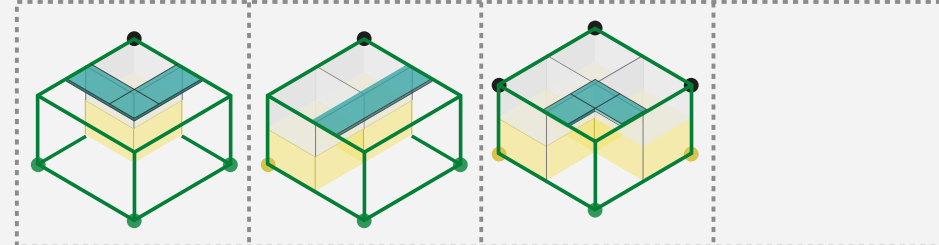
# ARCHITECTURAL TILESETS

## Opening tiles

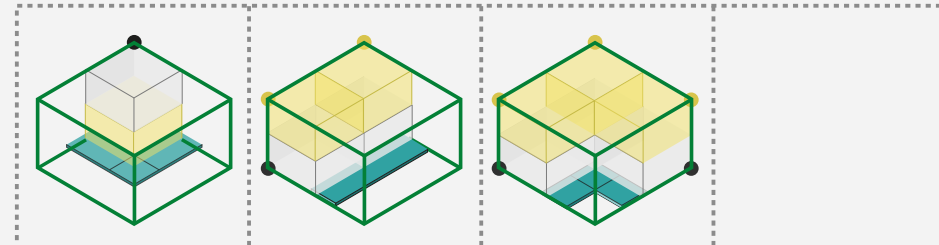
Voxel ID\_4

### Opening tiles

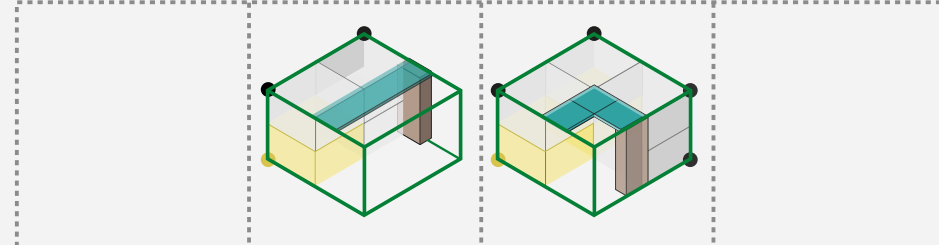
#### Top-middle tiles



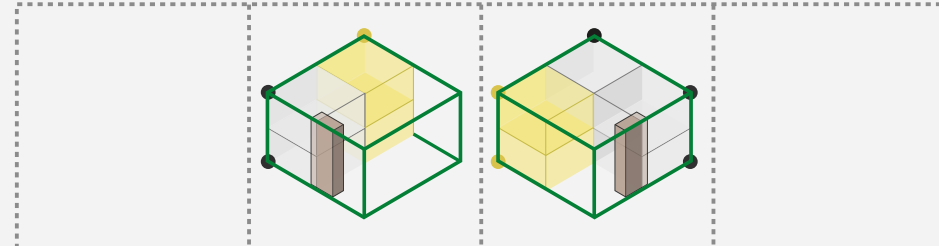
#### Bottom-middle tiles



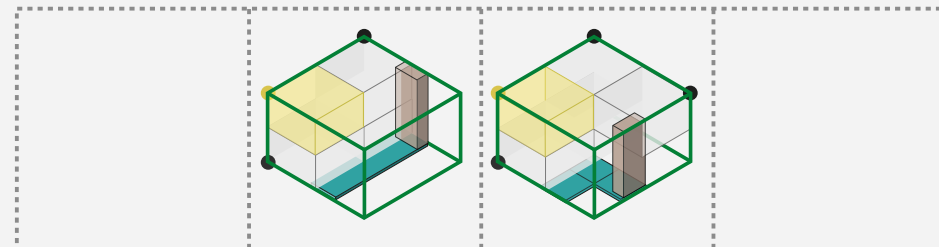
#### Top-side tiles



#### Middle-side tiles

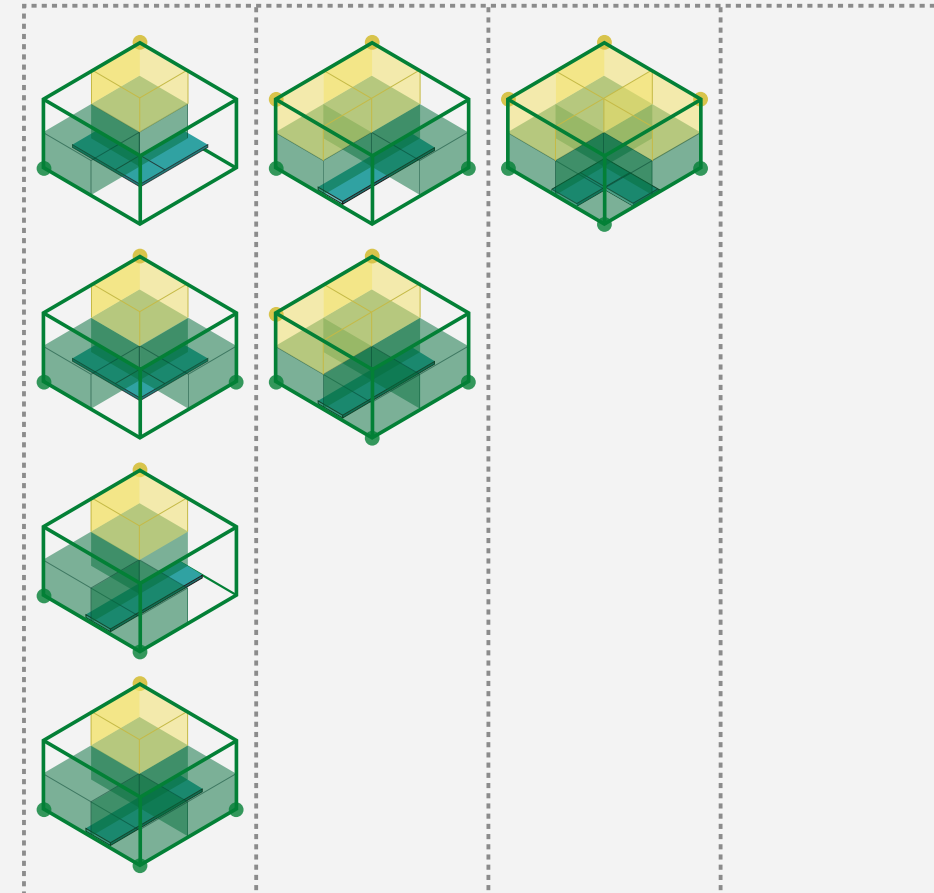


#### Bottom-side tiles

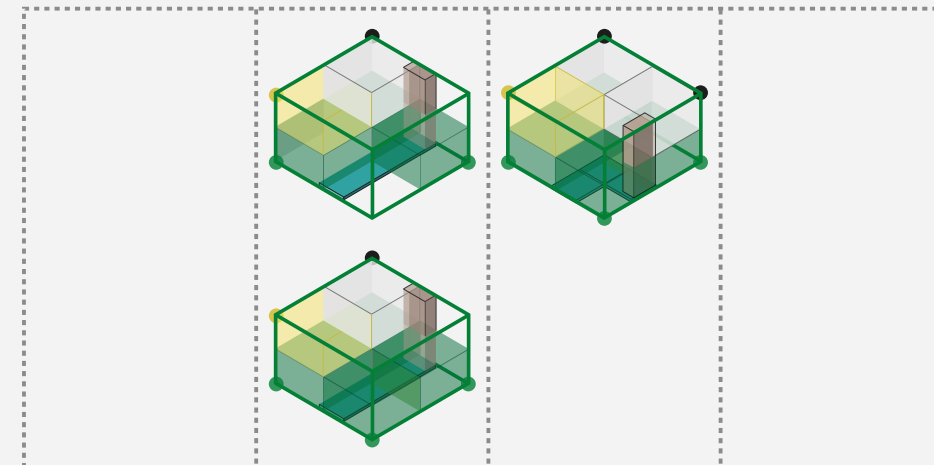


### Foundation to opening tiles

#### Up

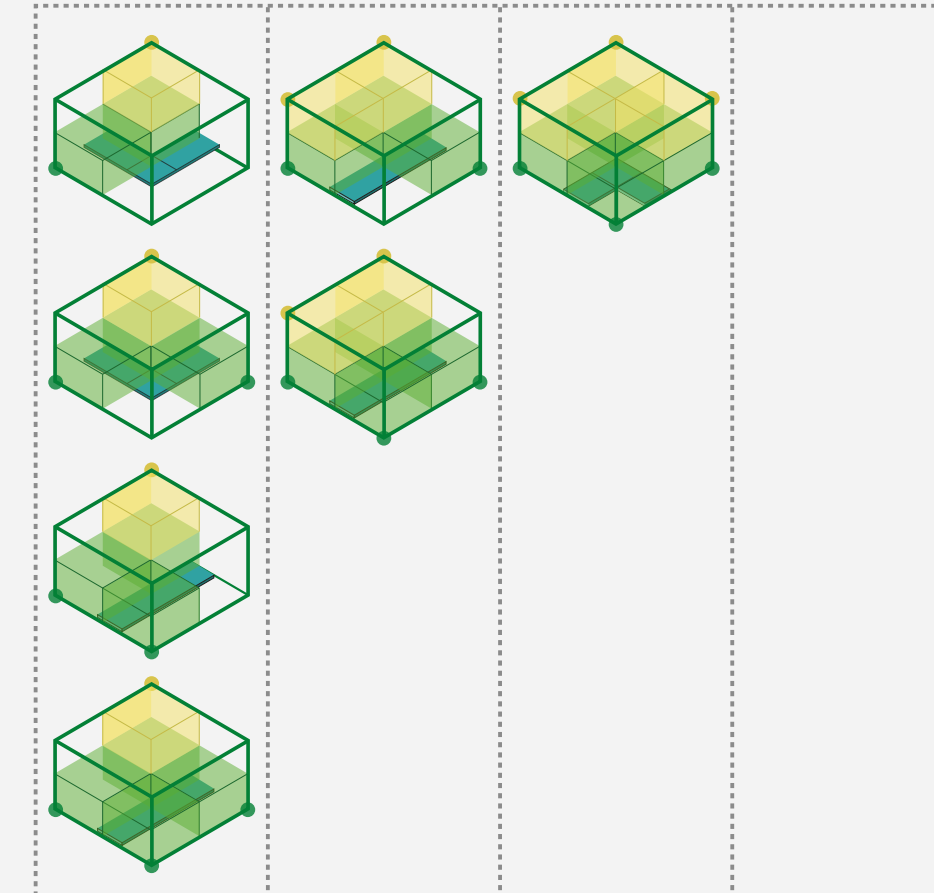


#### Side

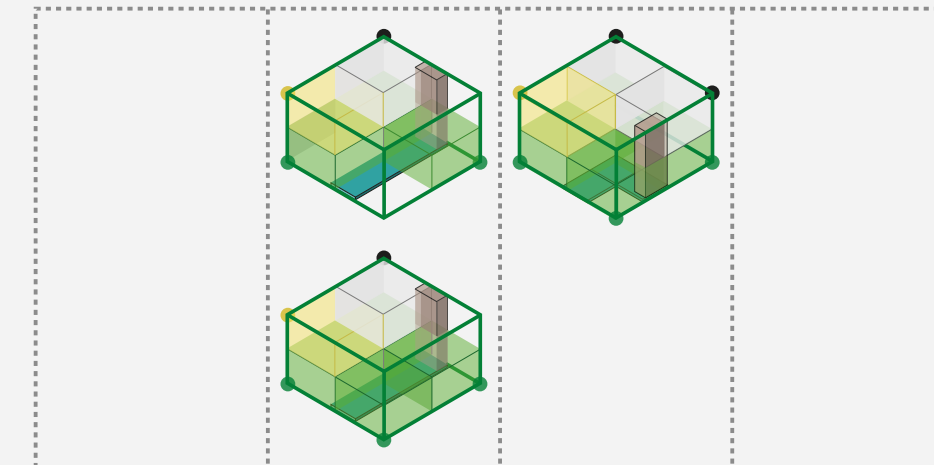


### Floor to opening tiles

#### Up



#### Side



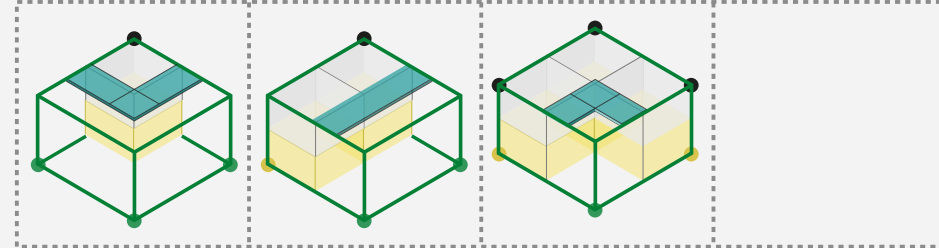
# ARCHITECTURAL TILESETS

## Opening tiles

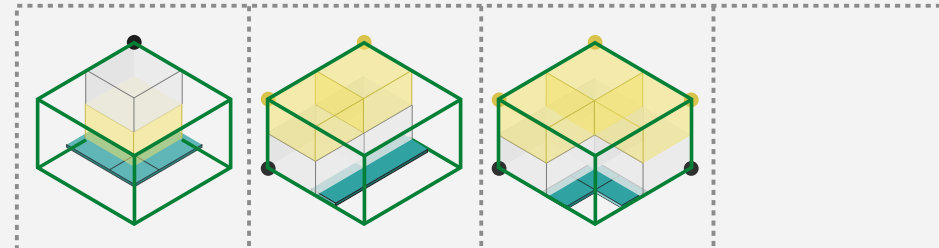
Voxel ID\_4

### Opening tiles

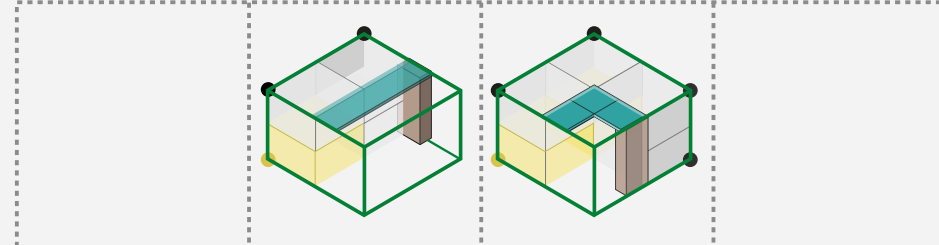
#### Top-middle tiles



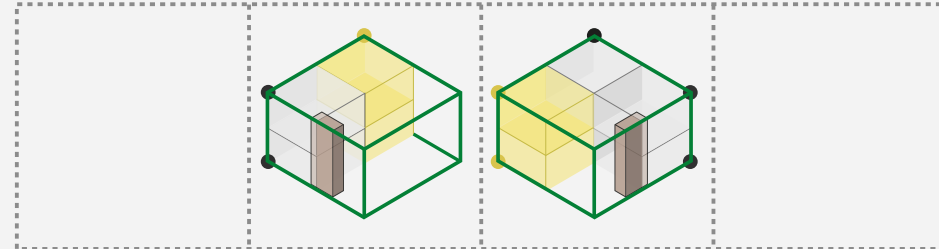
#### Bottom-middle tiles



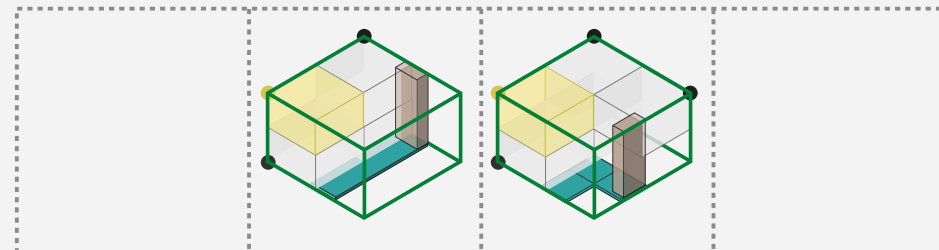
#### Top-side tiles



#### Middle-side tiles

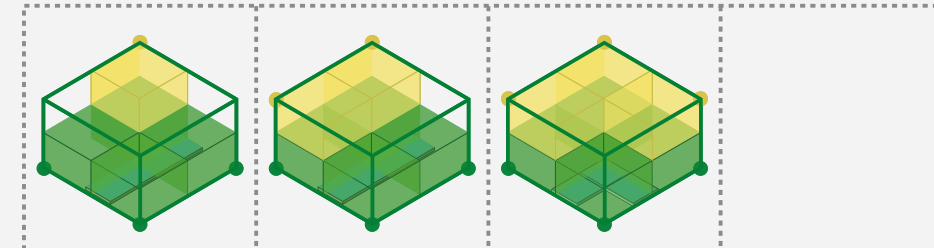


#### Bottom-side tiles

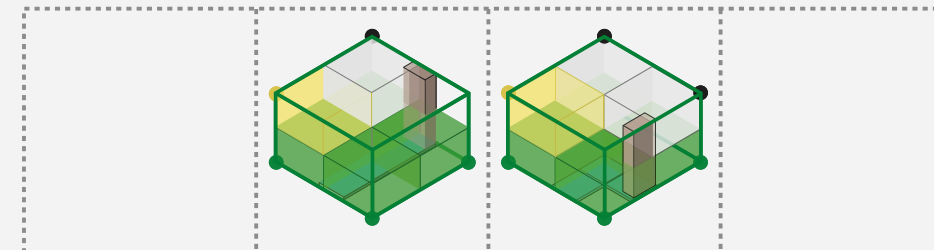


### Foundation & Floor to opening tiles

#### Up



#### Side



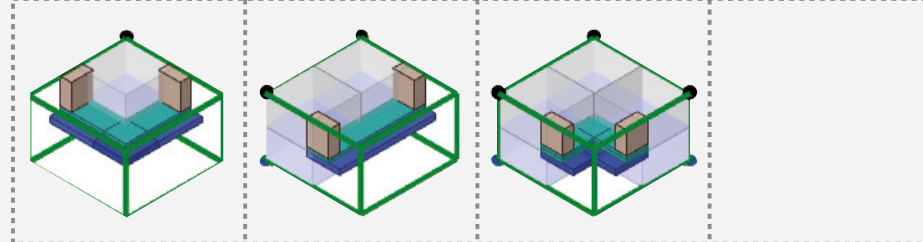
# ARCHITECTURAL TILESETS

*Window tiles*

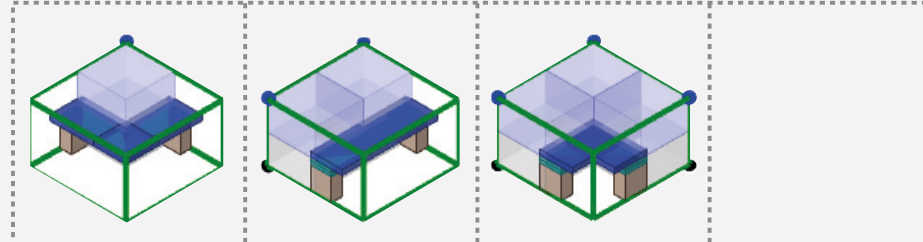
Voxel ID\_5

## Window tiles

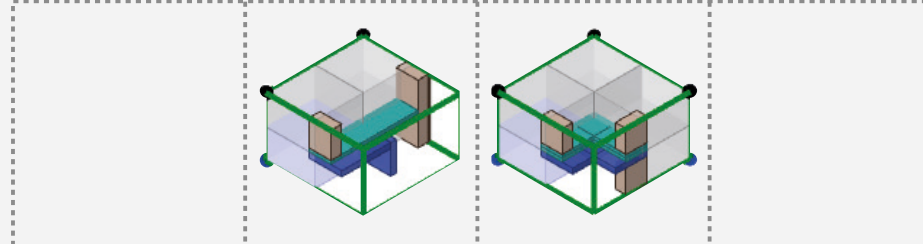
### Top-middle tiles



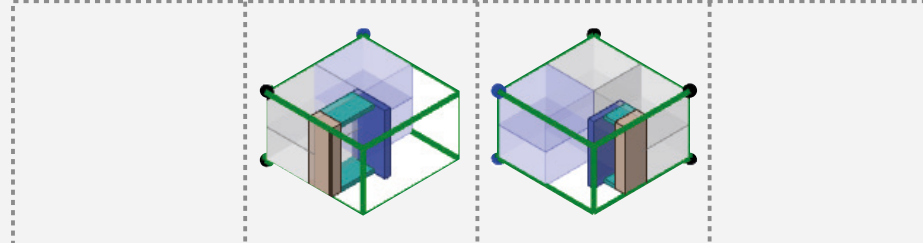
### Bottom-middle tiles



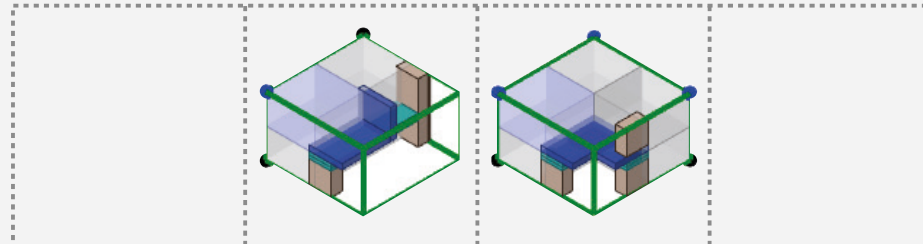
### Top-side tiles



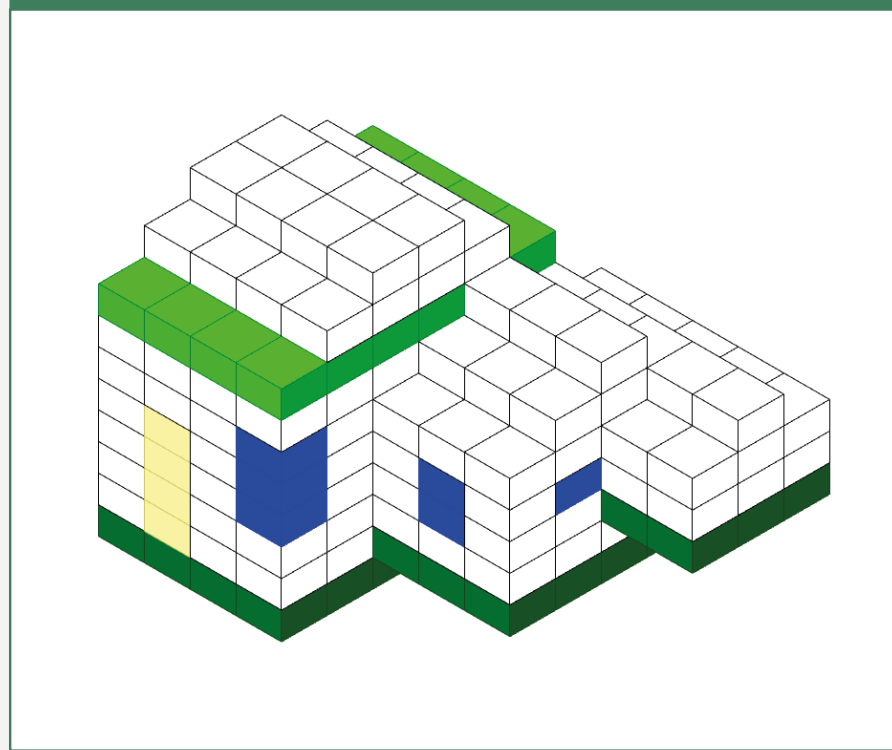
### Middle-side tiles



### Bottom-side tiles

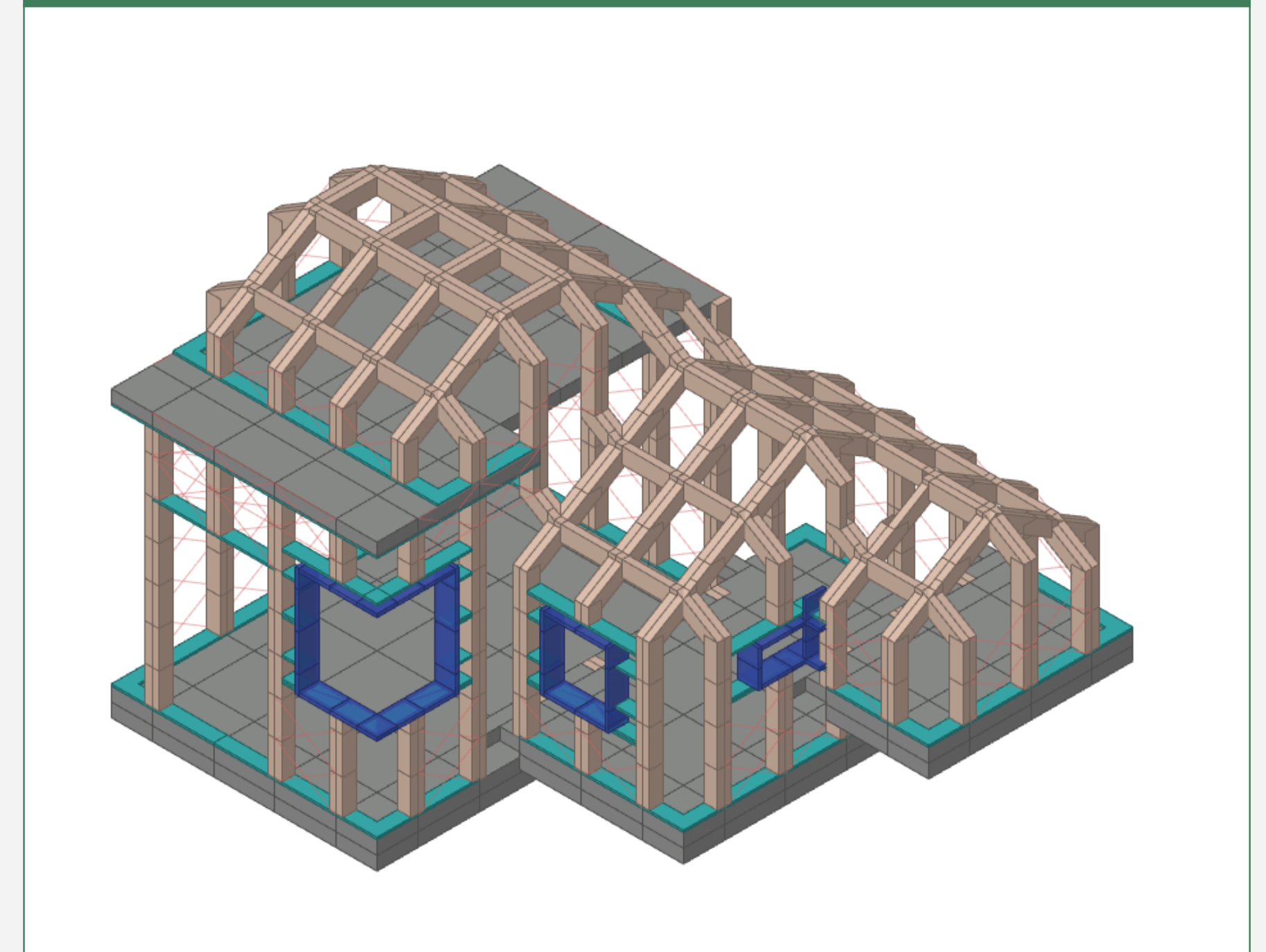


*Input*



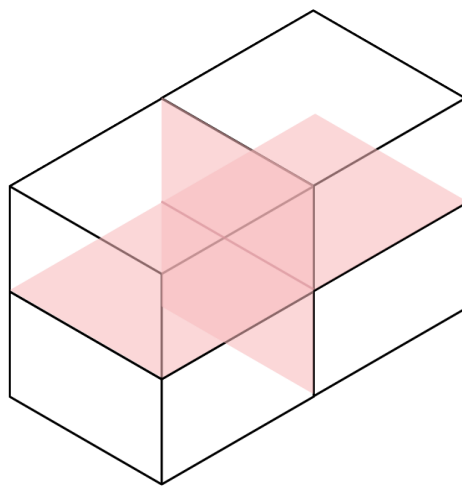
**HOUSCAPER**

*Output*

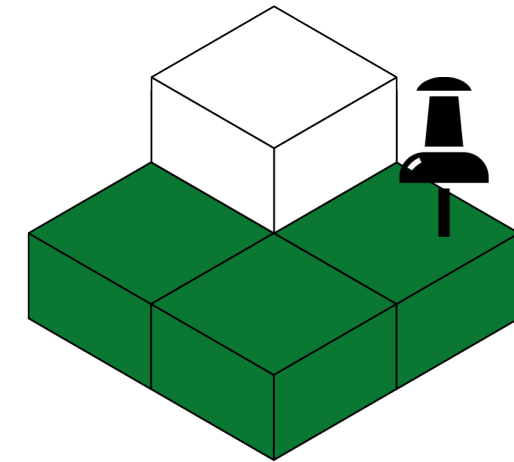


# DESIGN RULES

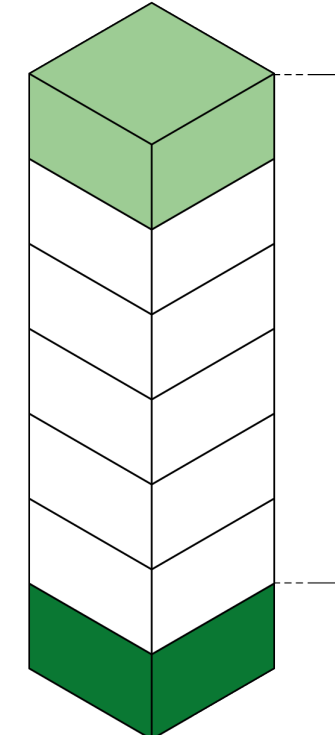
*For Normal users*

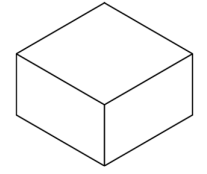


All voxels =  
**Surface connect**

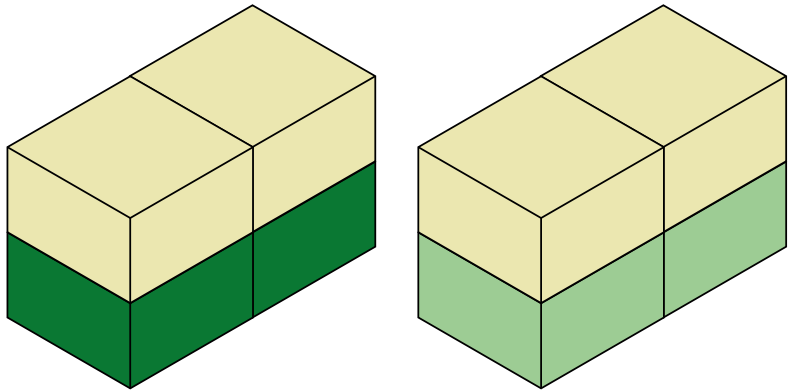


Foundation voxels =  
**at the bottom**

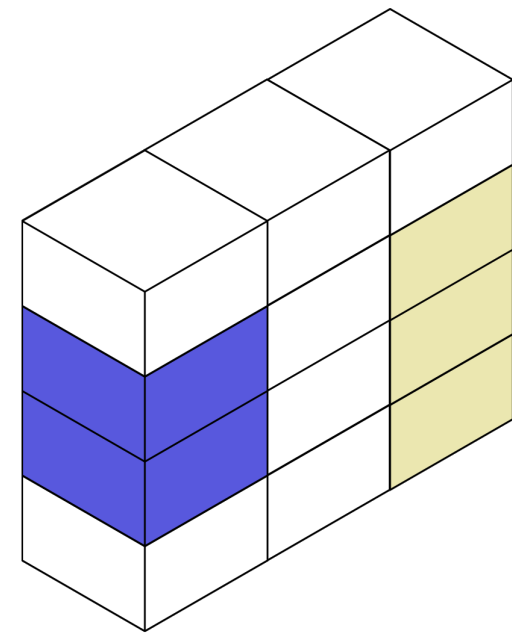


3240 mm  
=  x 6  
= Floor to floor

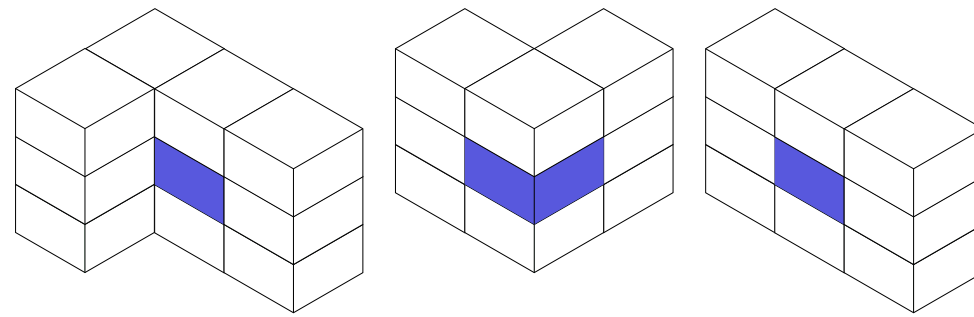
**6 stack** of tiles  
= 3240mm



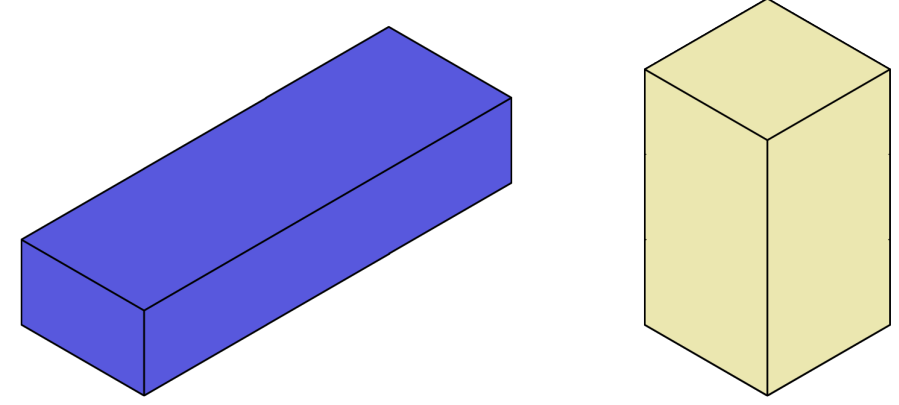
The yellow voxel **starts from green voxels.**



Color voxels  
**should not be placed**  
next to each other



**The minimum units** of  
window tilesets

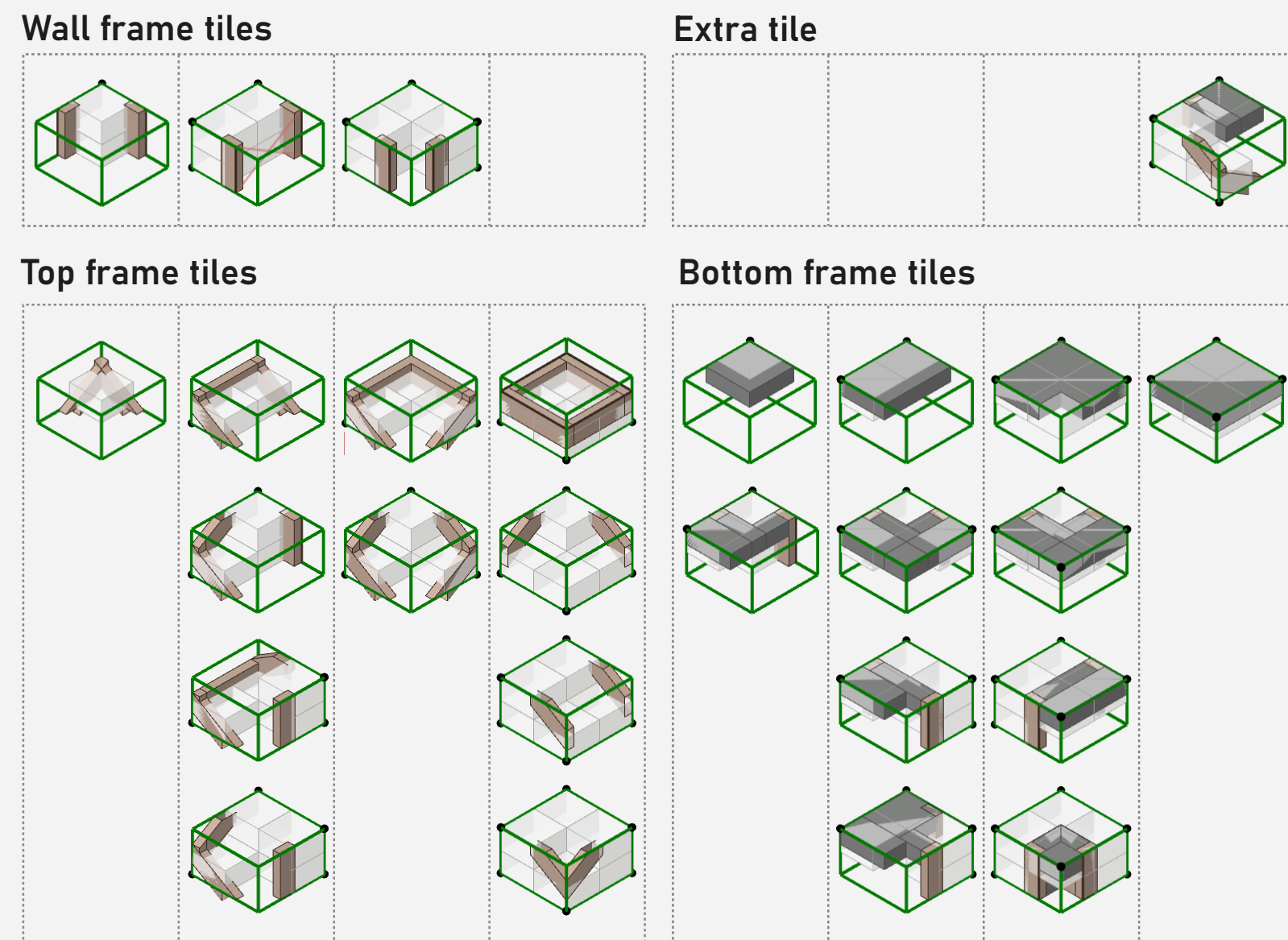


continuous placing in  
rectangular shape  
= **bigger openings**

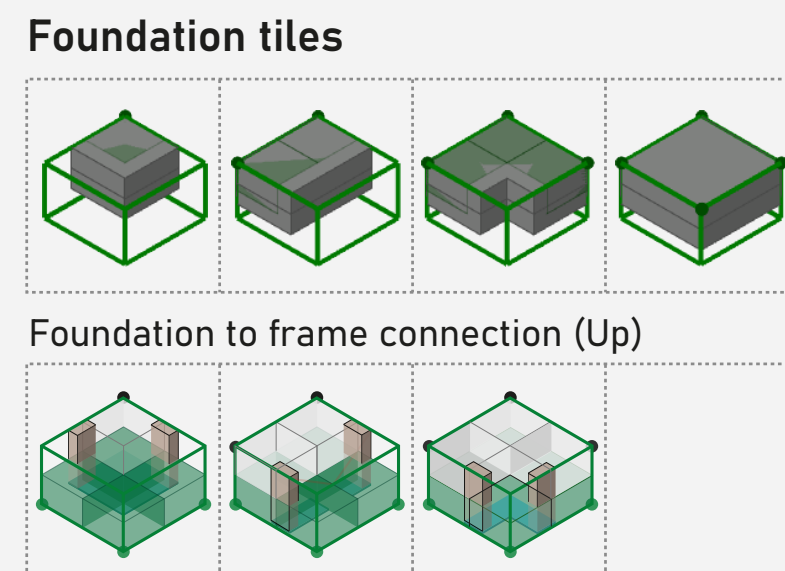


# ARCHITECTURAL TILESETS OVERVIEW

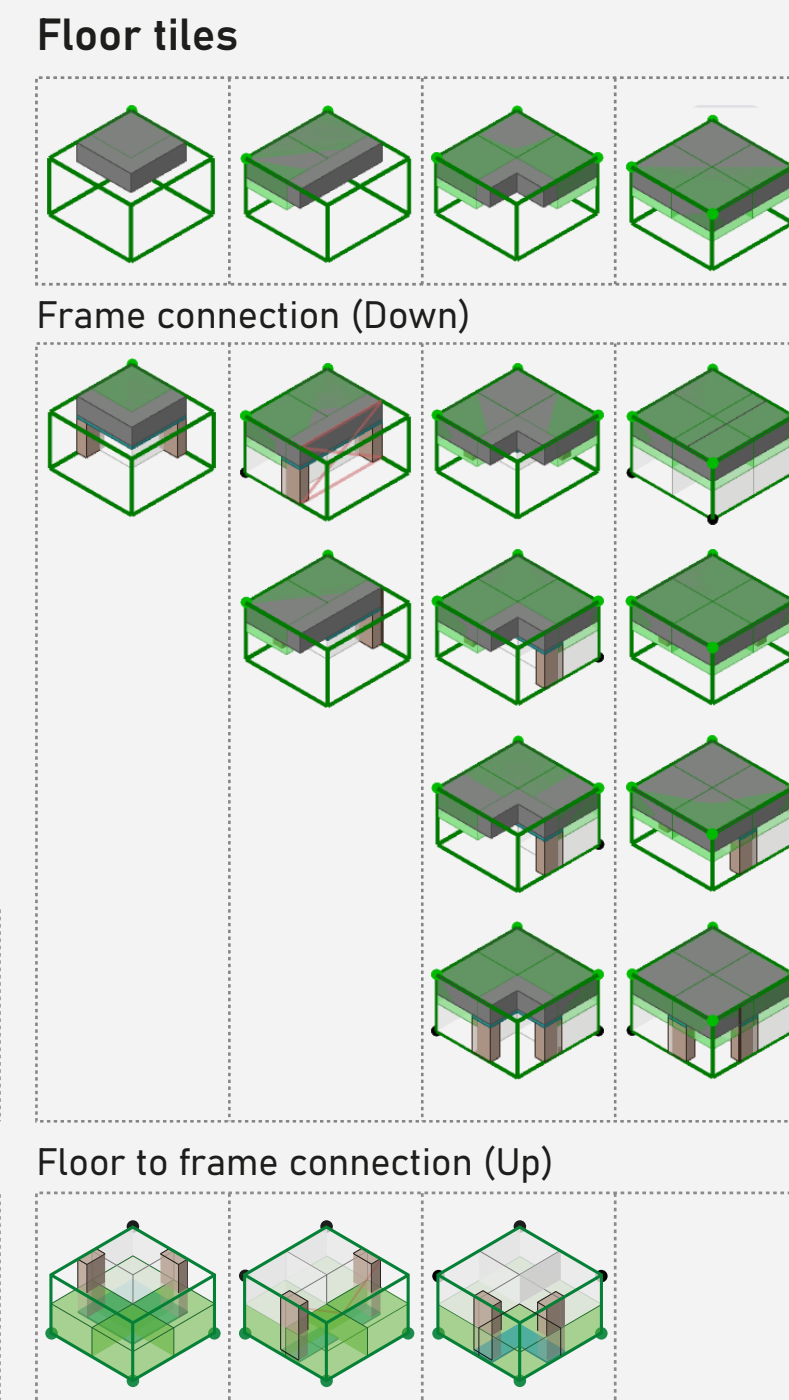
Voxel ID\_1



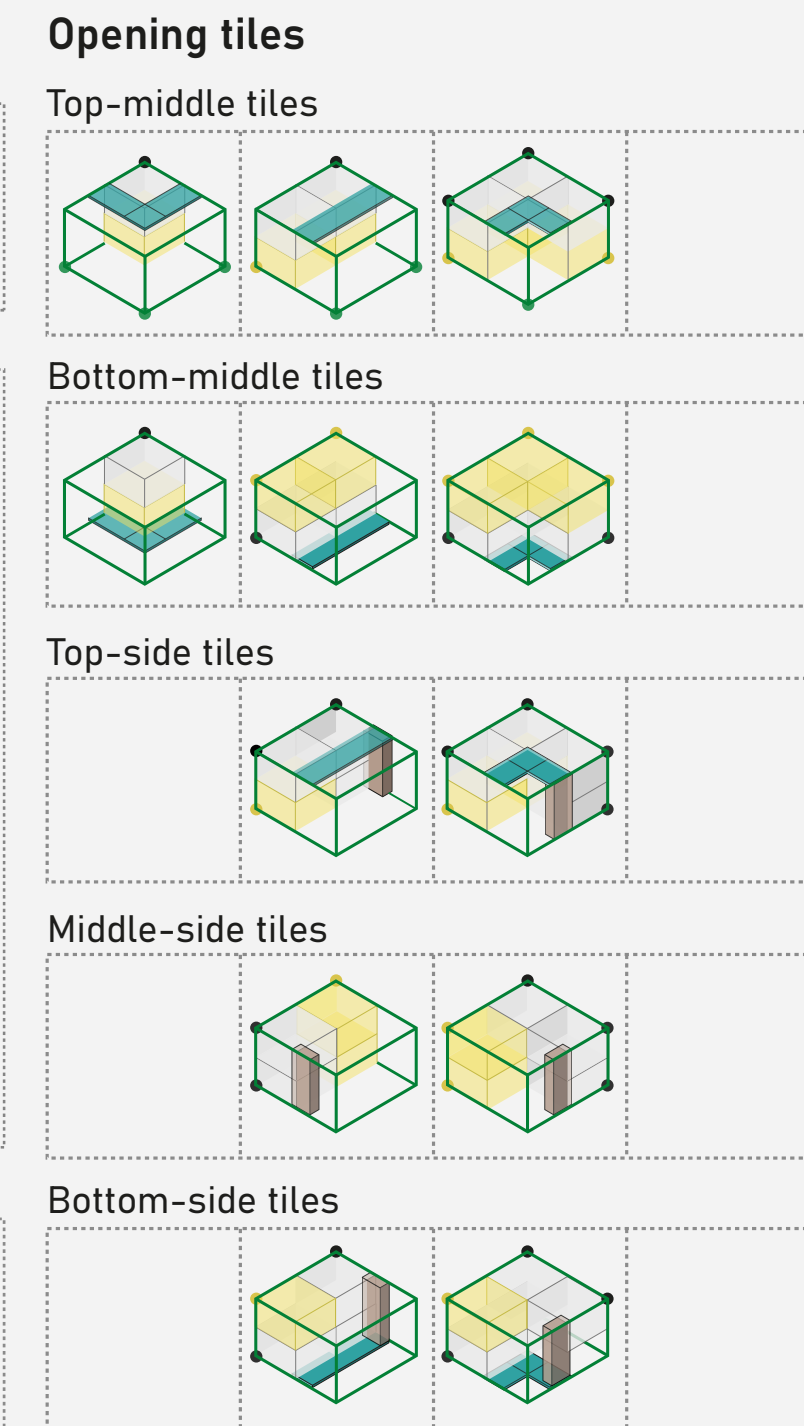
Voxel ID\_3



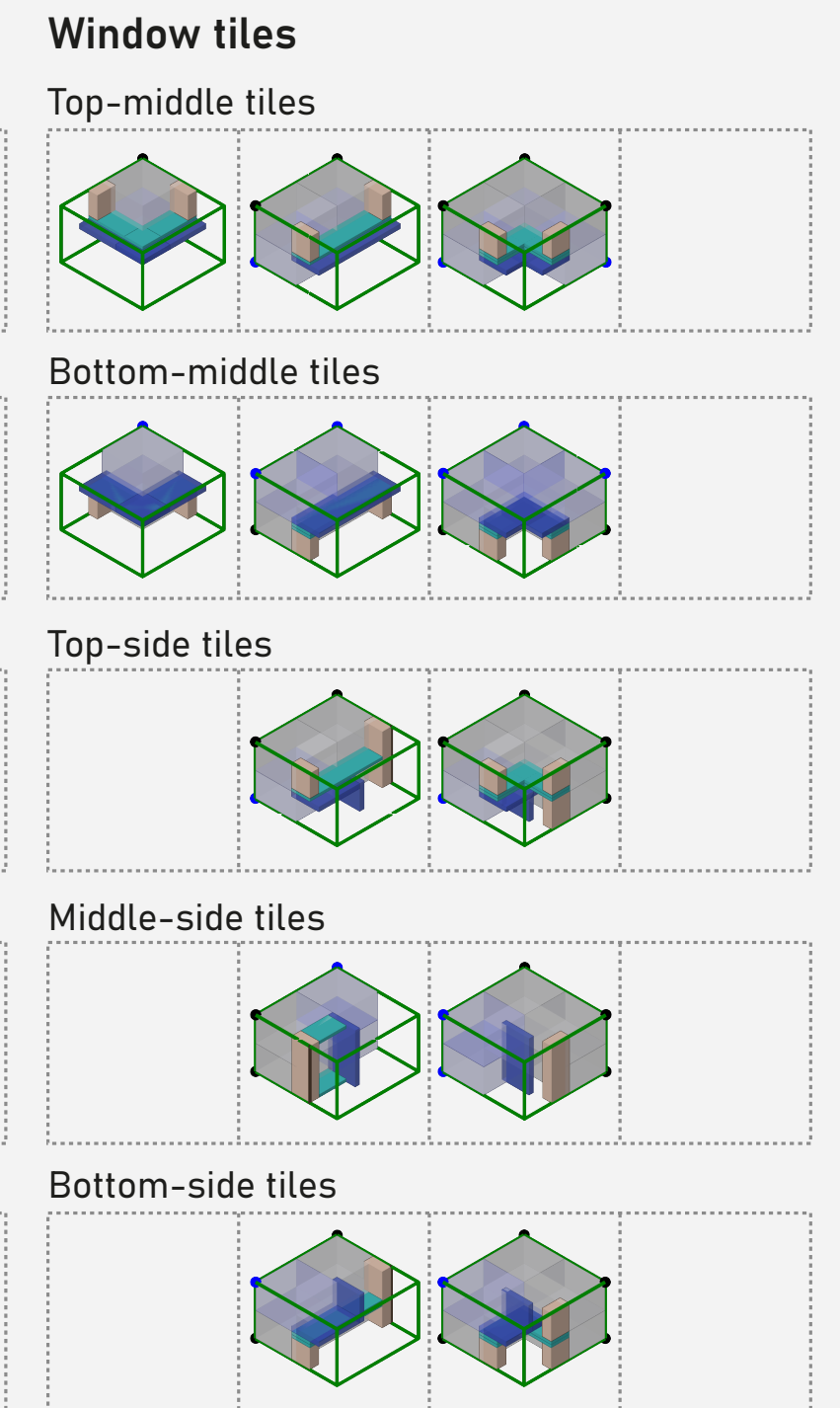
Voxel ID\_2



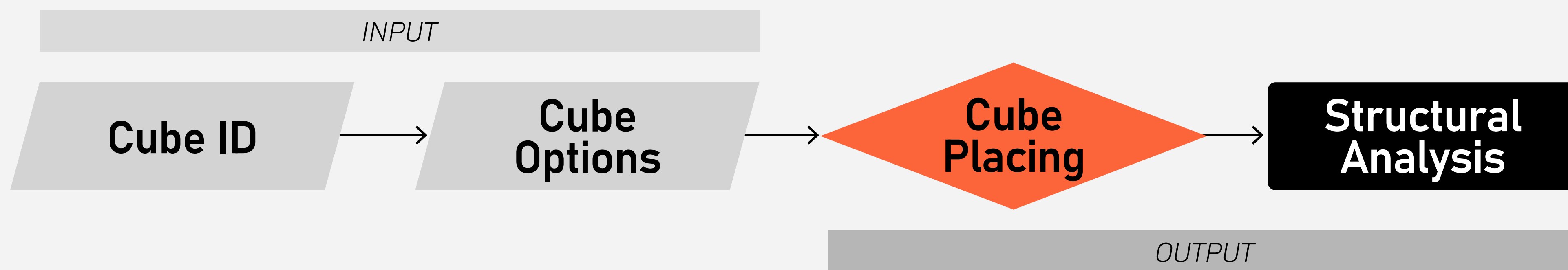
Voxel ID\_4

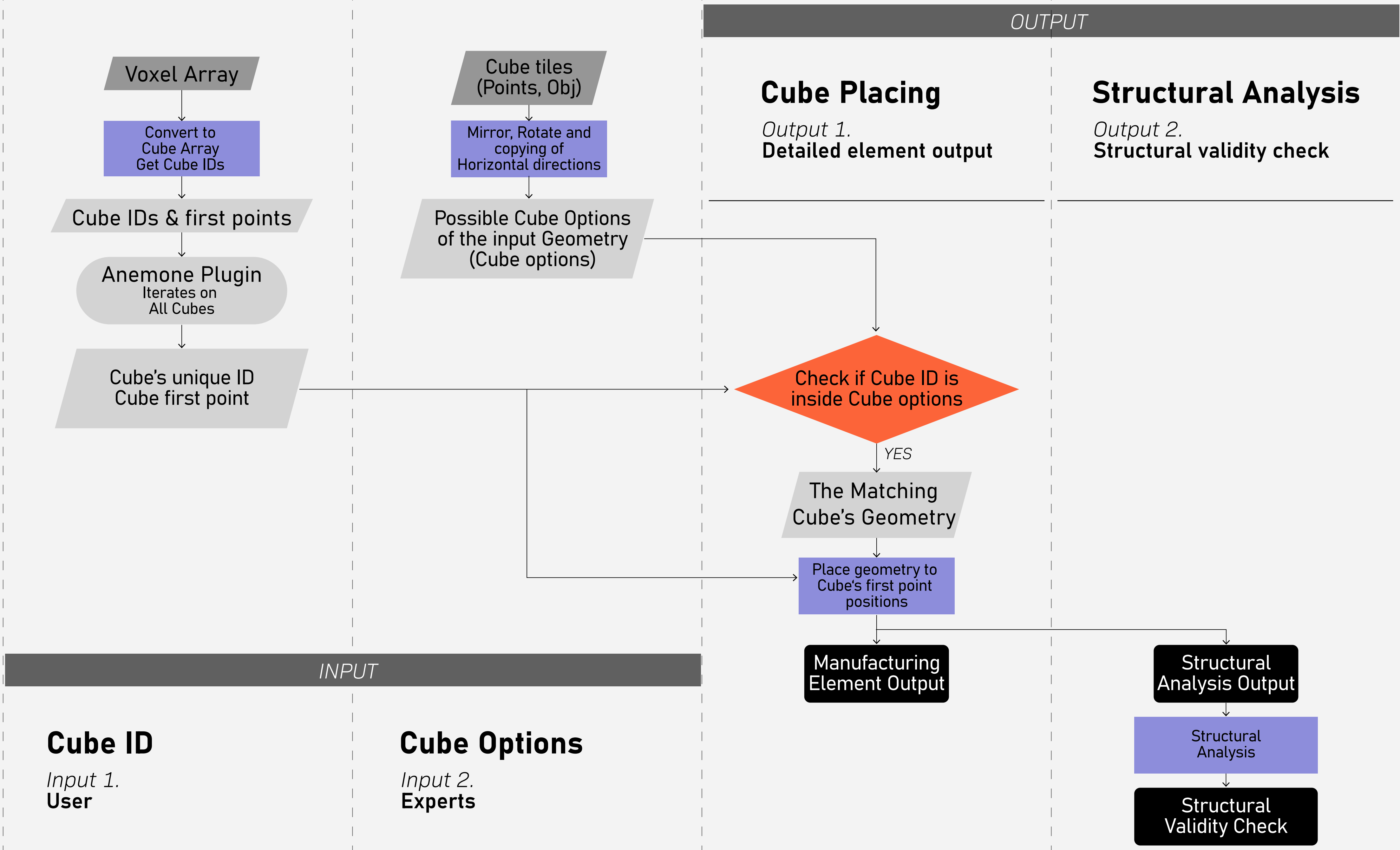


Voxel ID\_5



# ALGORITHM DEVELOPMENTS





Voxel Array

Cube tiles (Points, Obj)

Cube Placing

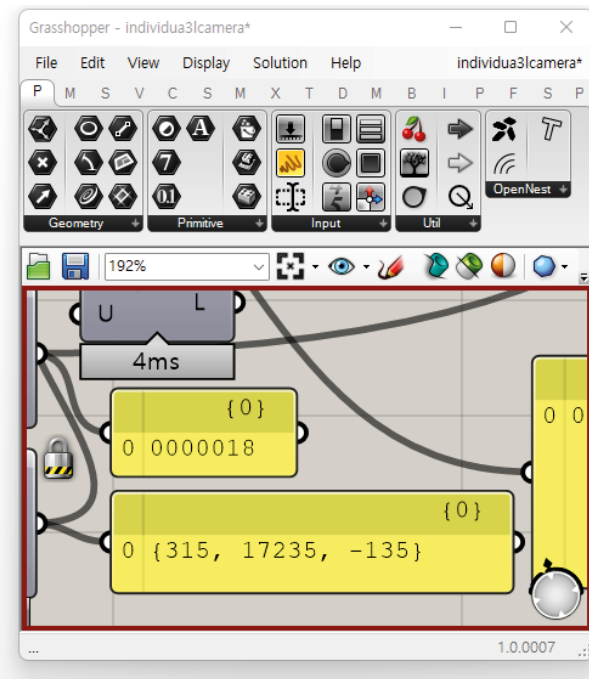
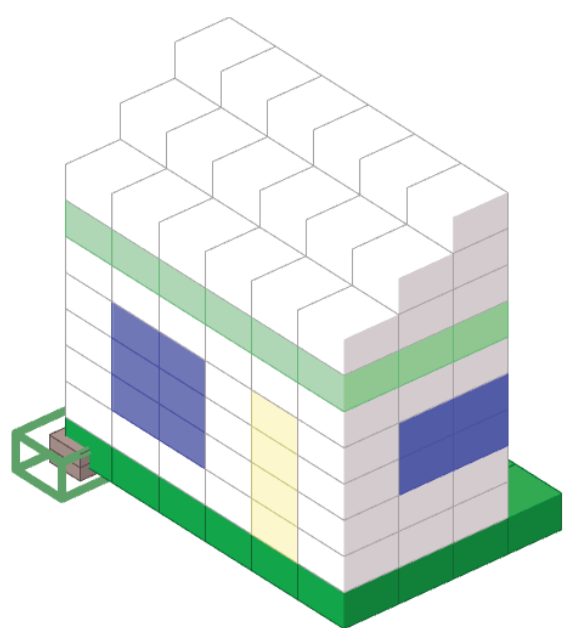
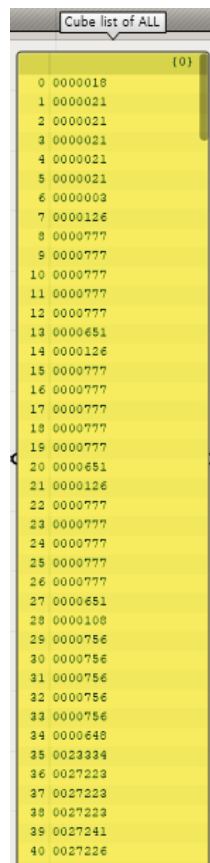
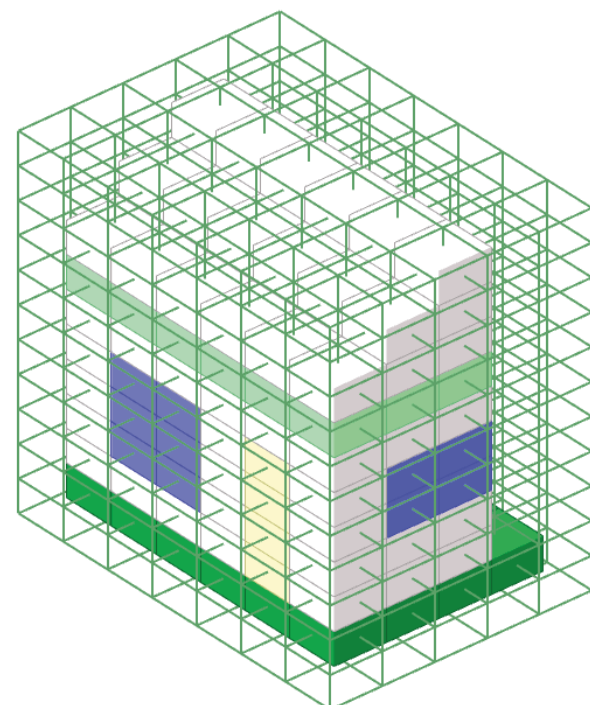
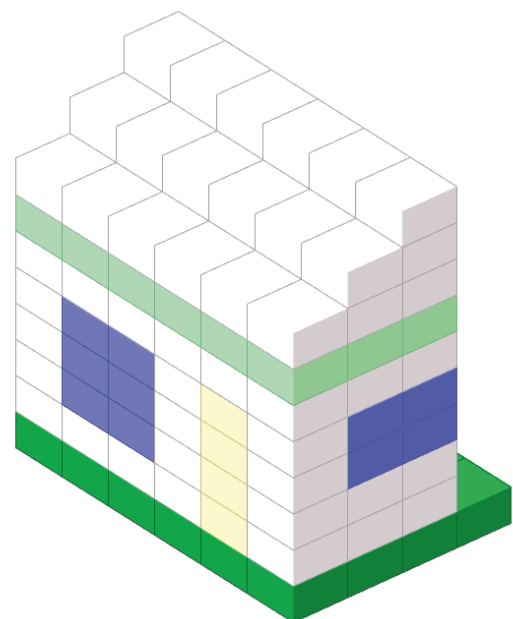
Structural Analysis

Output 2. Structural validity check

User input voxel array

Convert to cube array

Iterate on all cube / cube id and cube first point



Cube ID

Input 1. User

Cube Options

Input 2. Experts

Structural Analysis Output

Structural Analysis

Structural Validity Check

Element Output

Voxel Array

Convert to  
Cube Array  
Get Cube IDs

Cube IDs & first points

Anemone Plugin  
Iterates on  
All Cubes

Cube's unique ID  
Cube first point

Cube tiles  
(Points, Obj)

Cube Placing

Structural Analysis

*Input*

Cube Frame

Octant of voxels (in vertices)

Tileset Geometries

HOUSCAPER  
TILE  
GENERATOR

*Output*

0009083    0282313

0326988    0054498

0009103    0281453

0327708    0054618

INP

**Cube ID**

Input 1.  
User

**Cube Options**

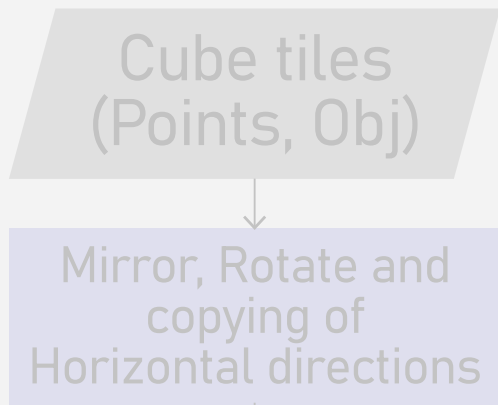
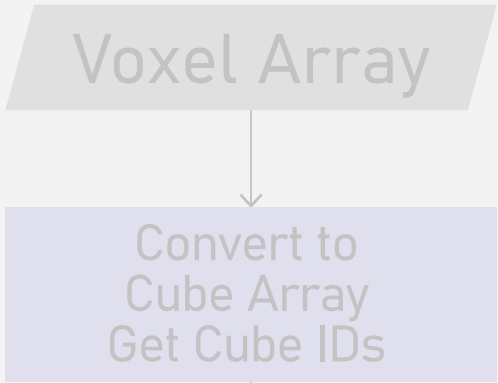
Input 2.  
Experts

Element Output

Analysis Output

Structural  
Analysis

Structural  
Validity Check



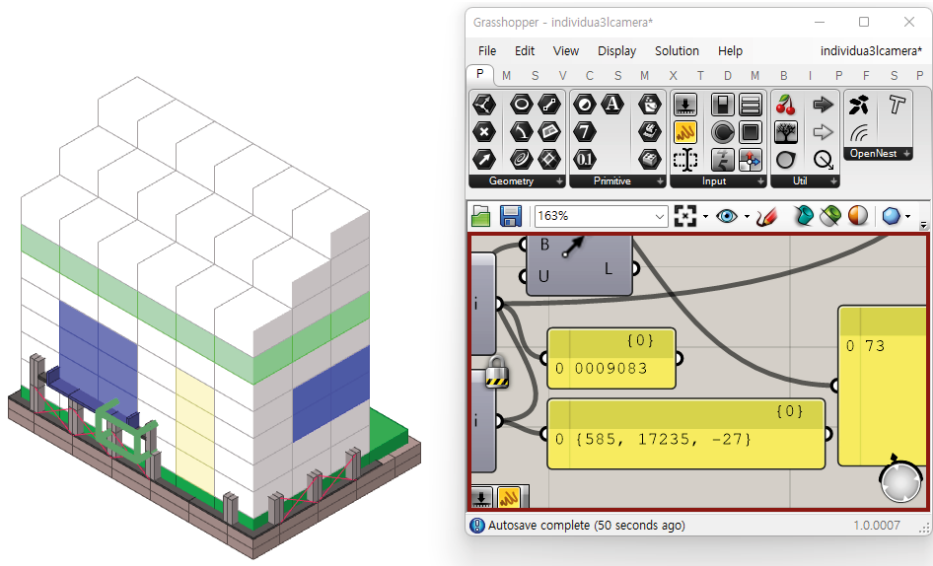
# Cube Placing

Output 1.  
Detailed element output

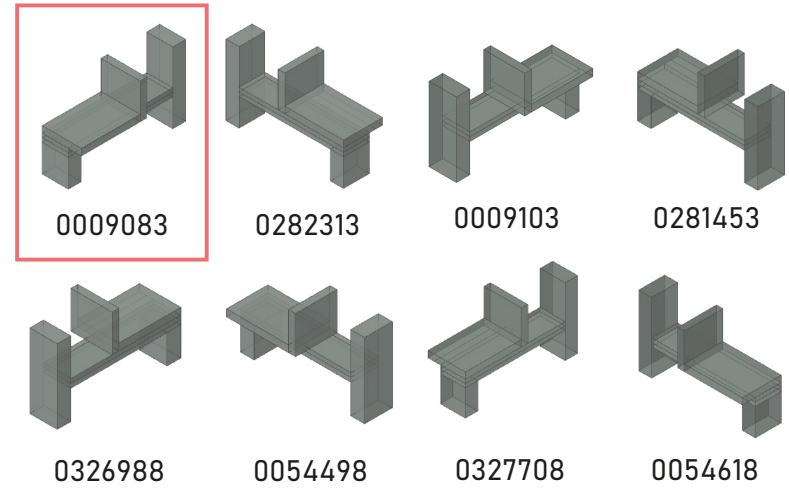
# Structural Analysis

Output 2.  
Structural validity check

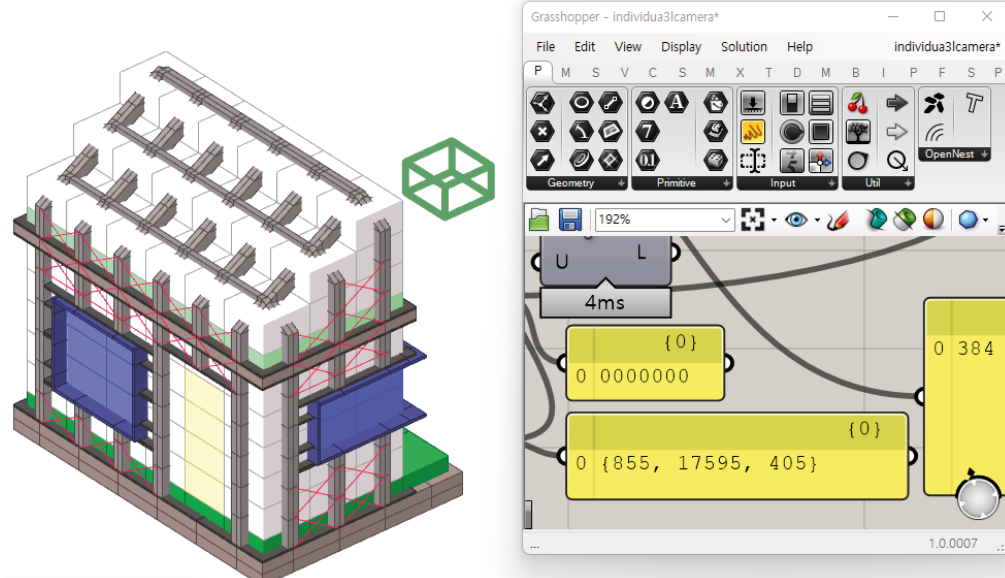
Cube ID



Search for cube id in cube options

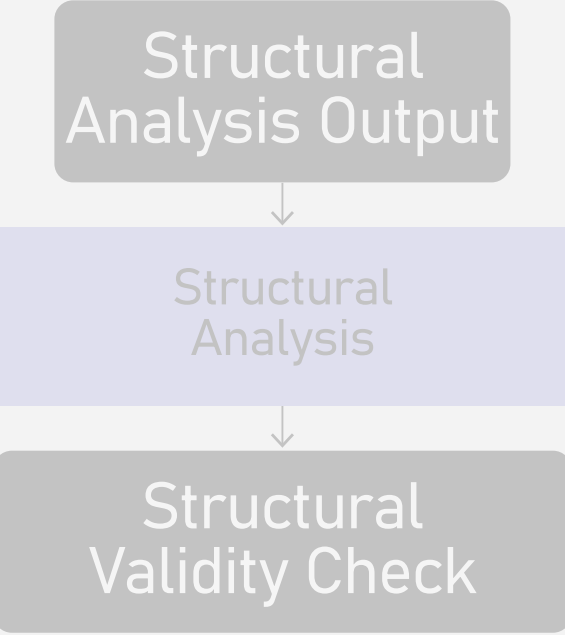


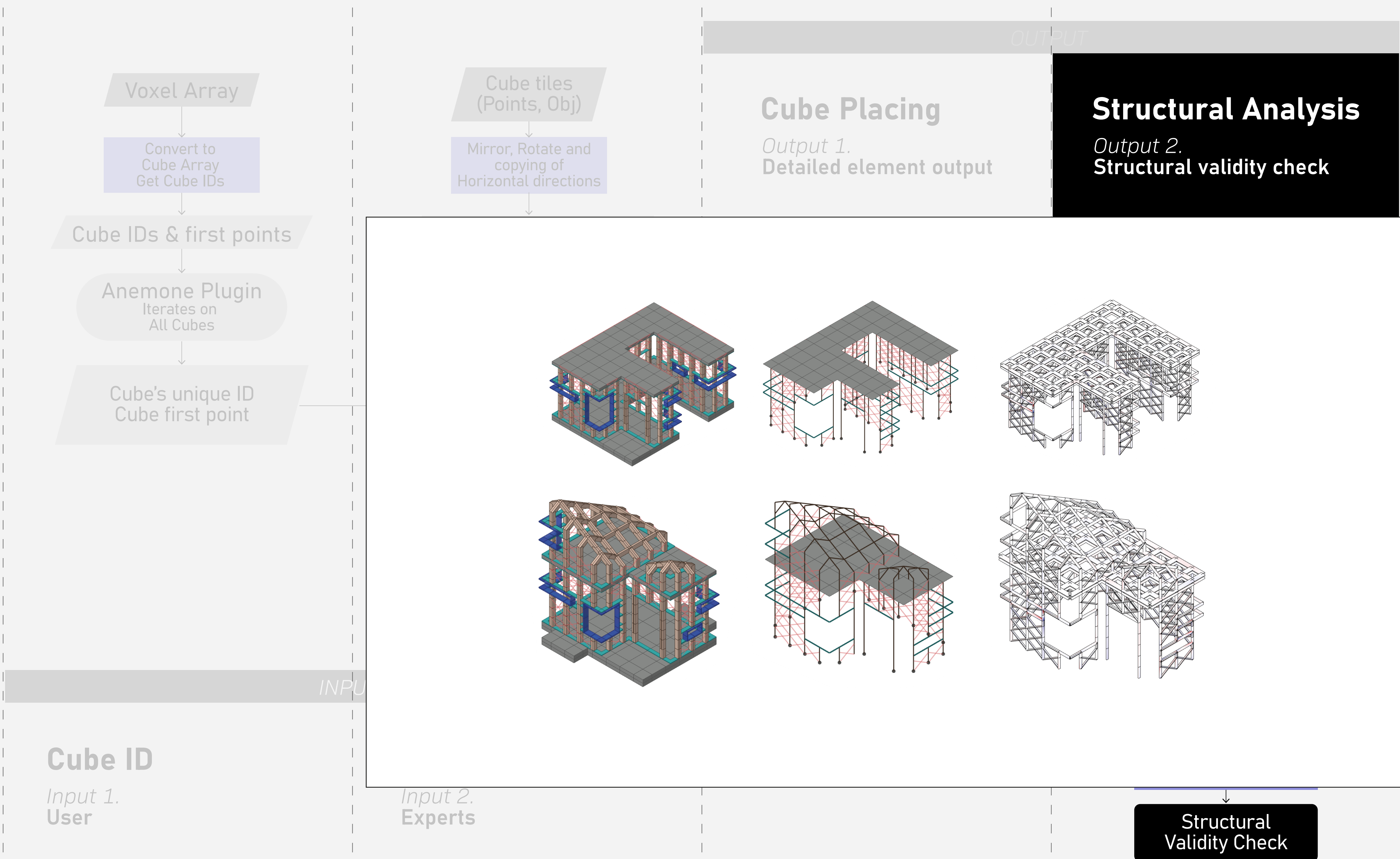
Cube placing/ running the algorithm



Input 1.  
User

Input 2.  
Experts





Voxel Array

Convert to  
Cube Array  
Get Cube IDs

Cube IDs & first points

Anemone Plugin  
Iterates on  
All Cubes

Cube's unique ID  
Cube first point

Cube tiles  
(Points, Obj)

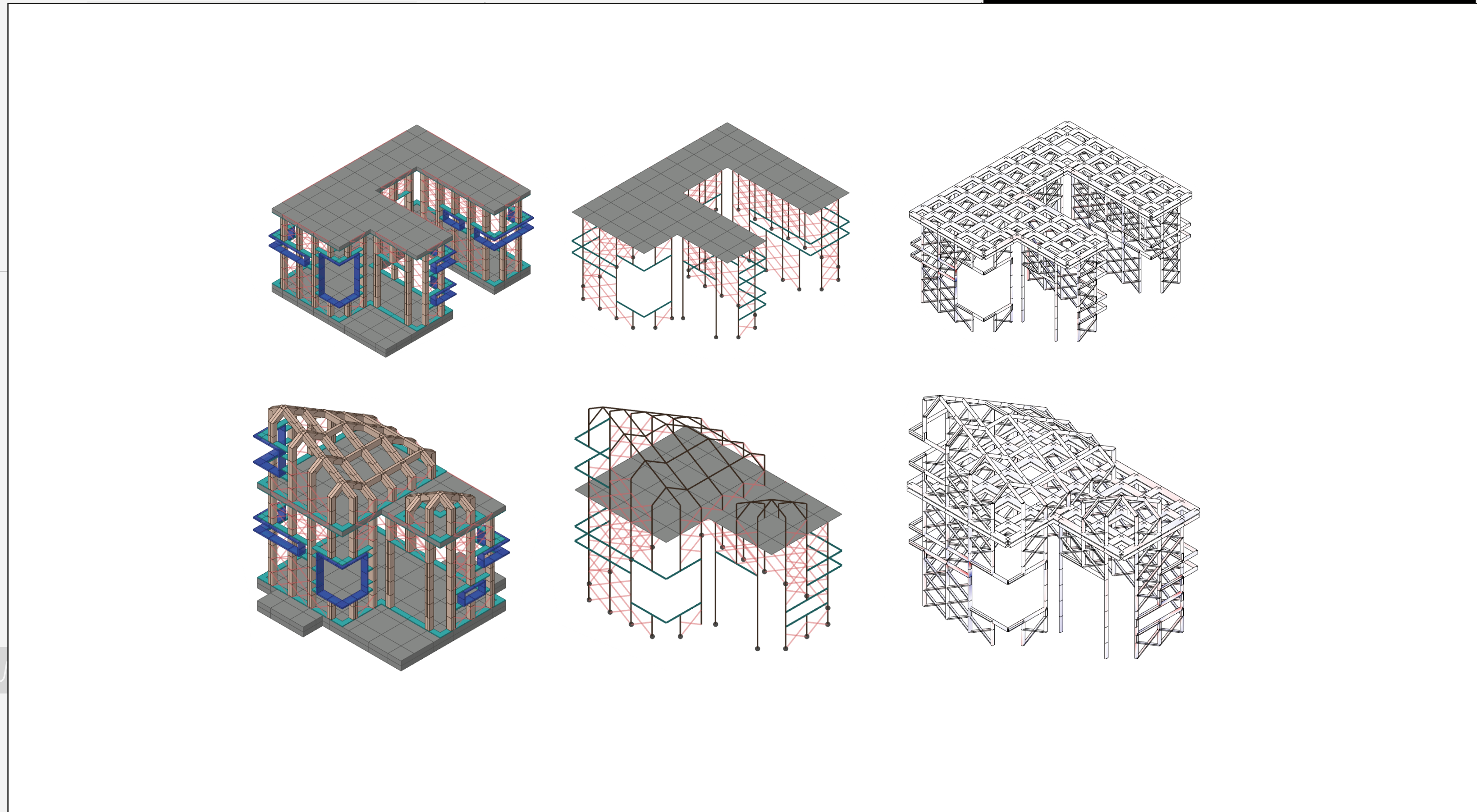
Mirror, Rotate and  
copying of  
Horizontal directions

Cube Placing

Output 1.  
Detailed element output

Structural Analysis

Output 2.  
Structural validity check



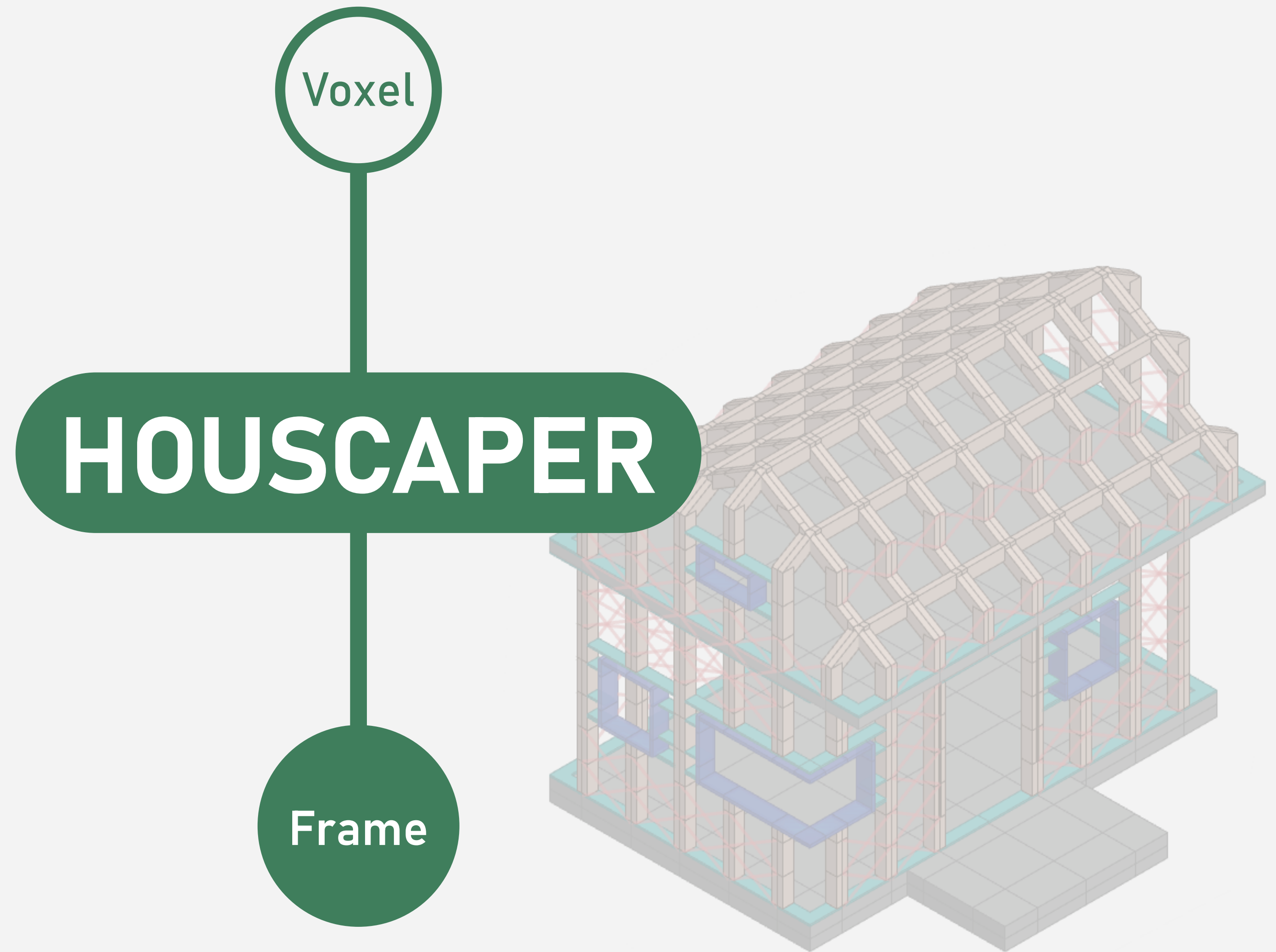
INPUT

Cube ID  
Input 1.  
User

Input 2.  
Experts

Structural  
Validity Check

# RESULTS

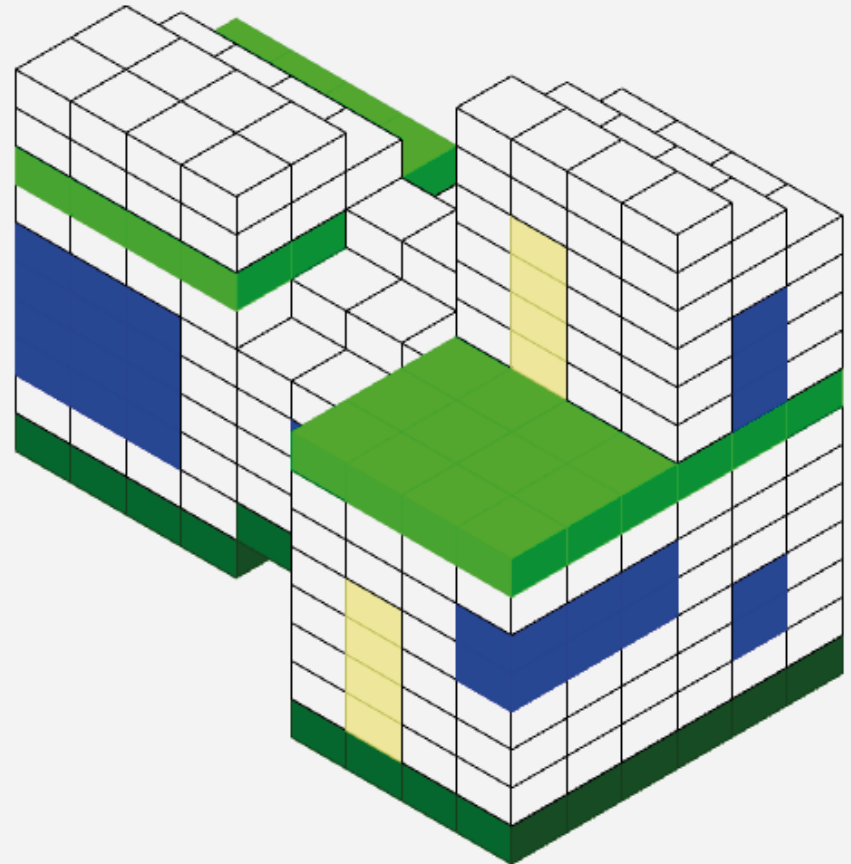
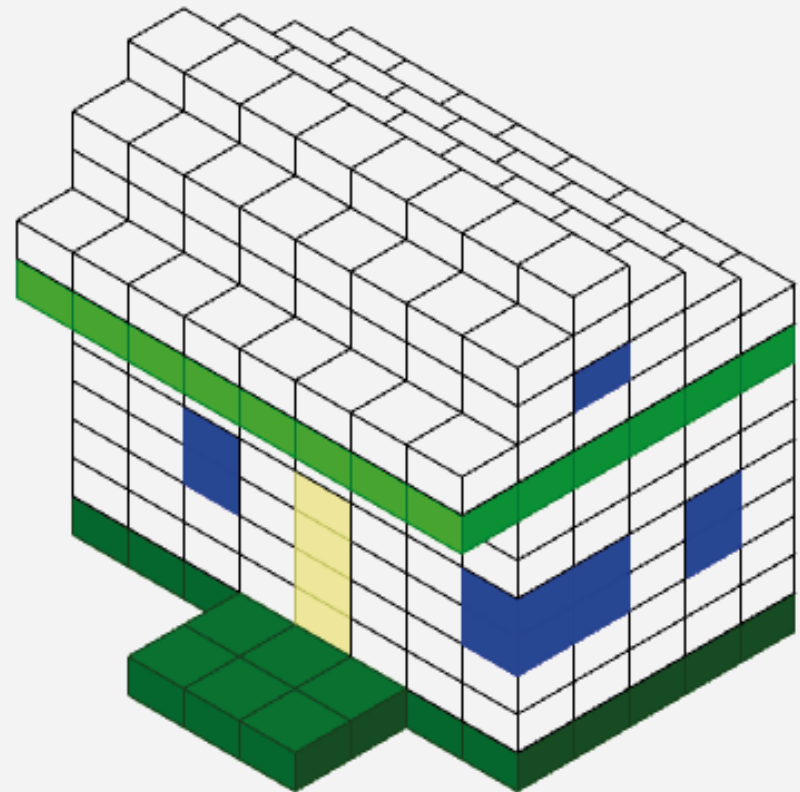
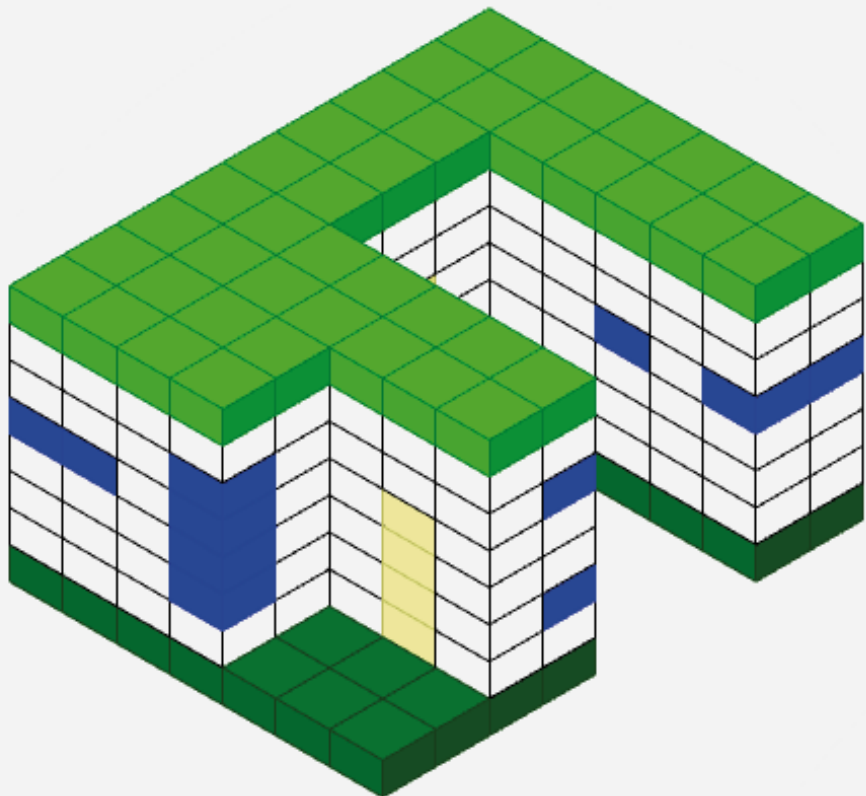
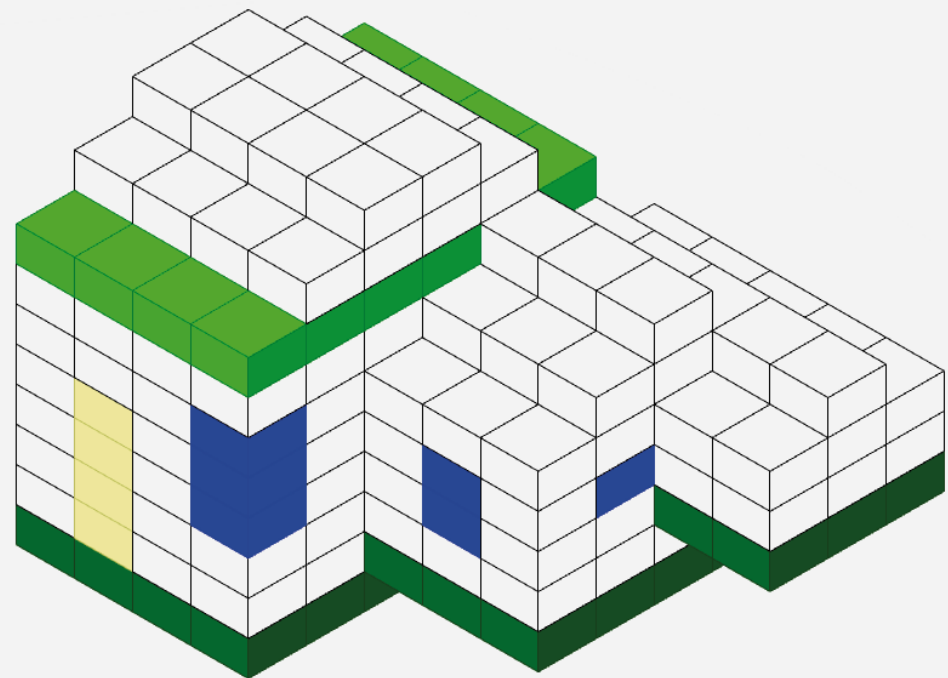
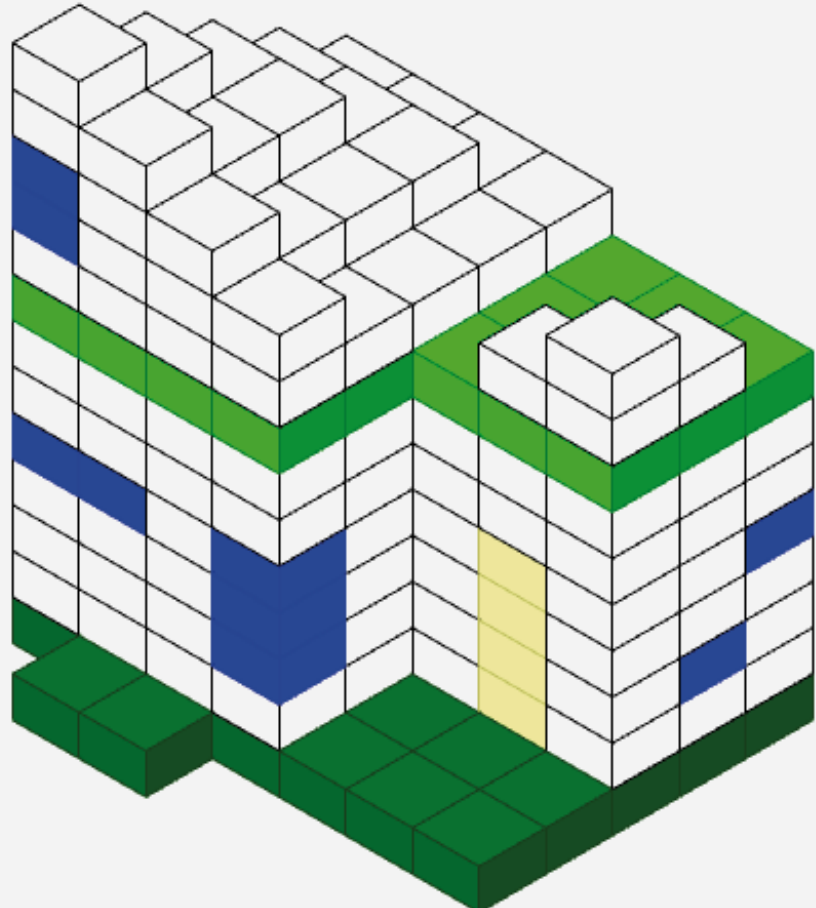
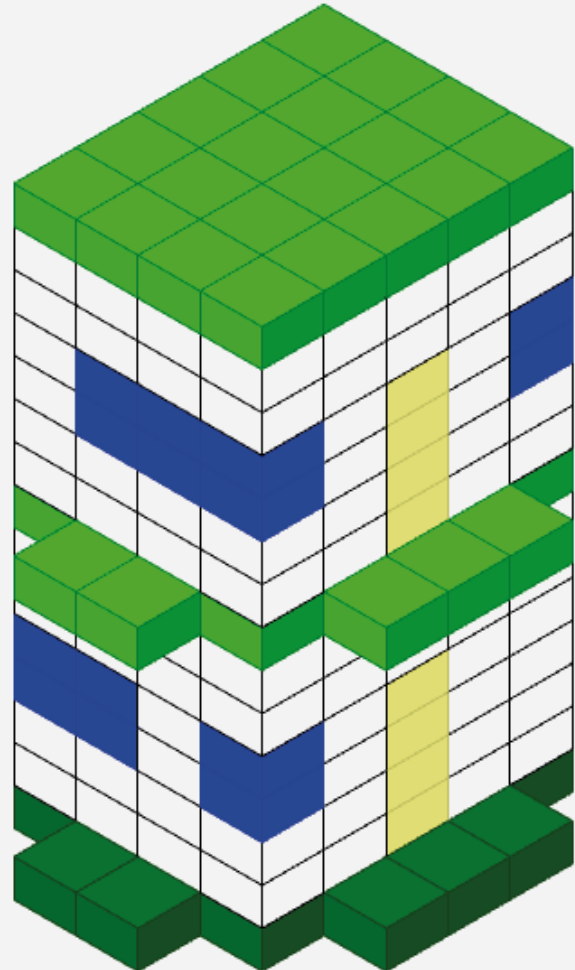




# BUILD WITH COLORED VOXELS

## 01 Easy stacking of voxels

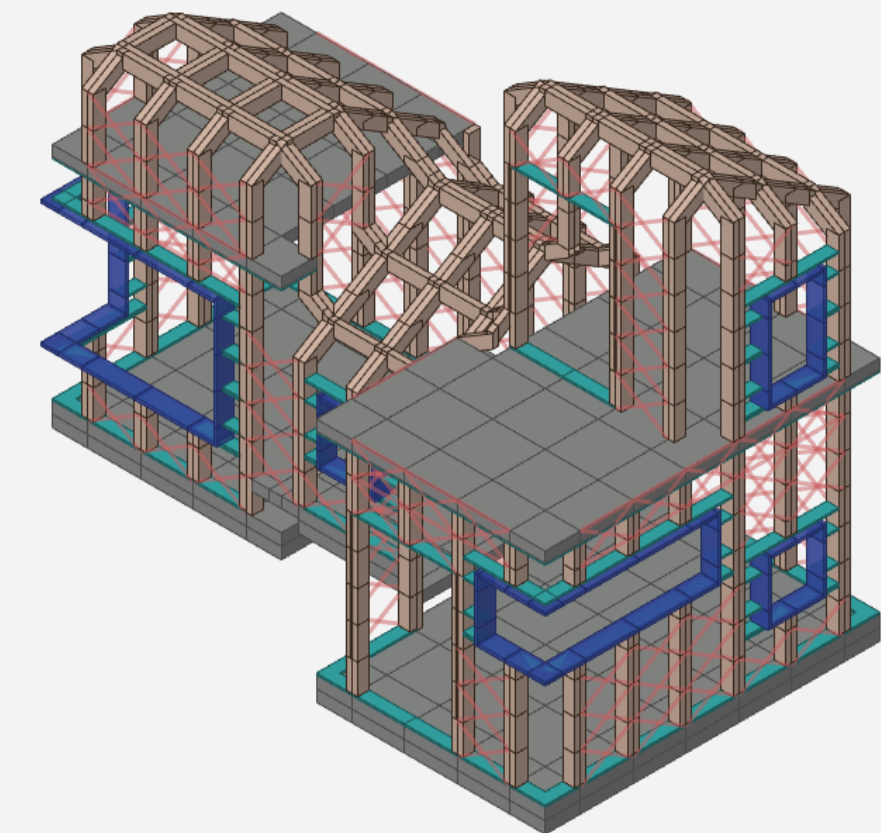
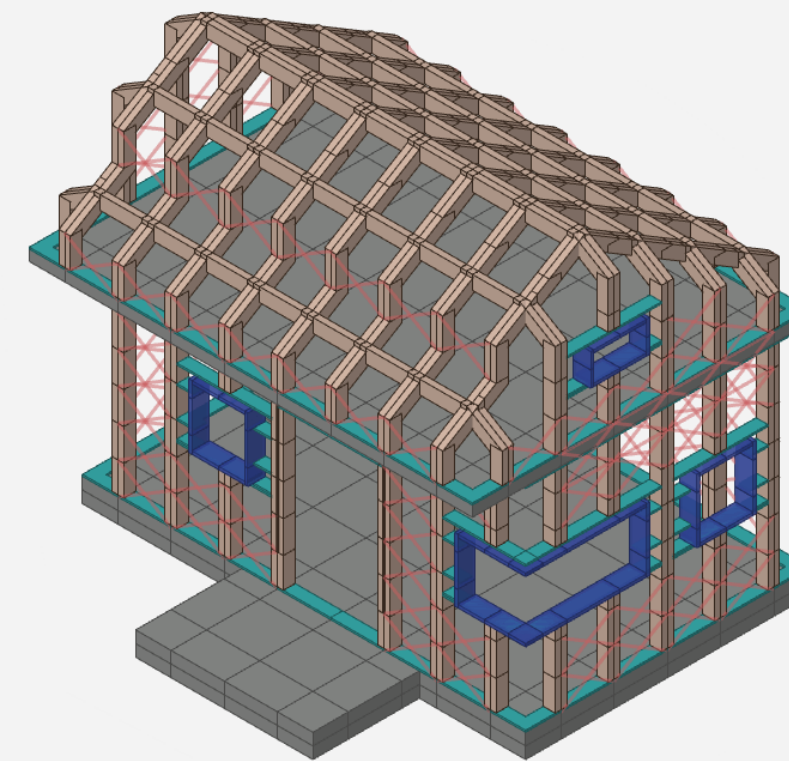
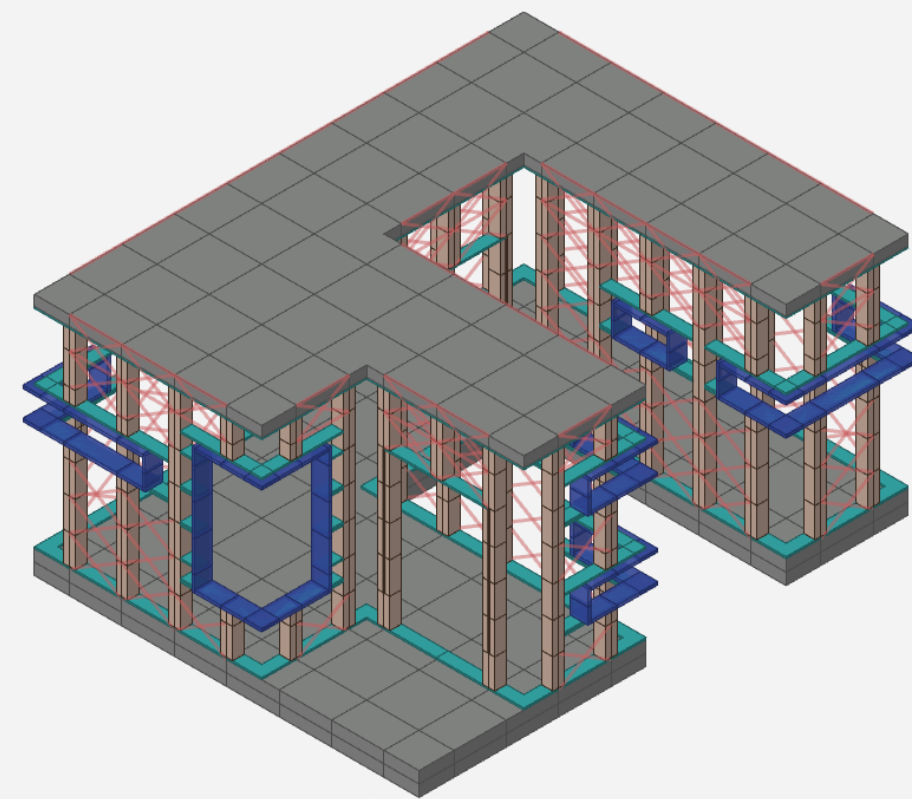
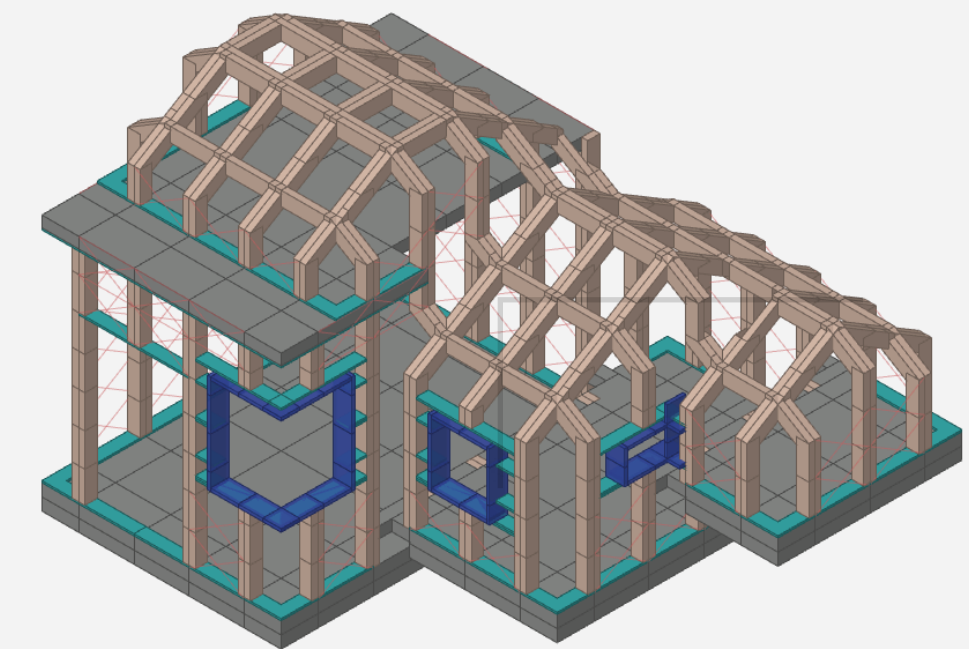
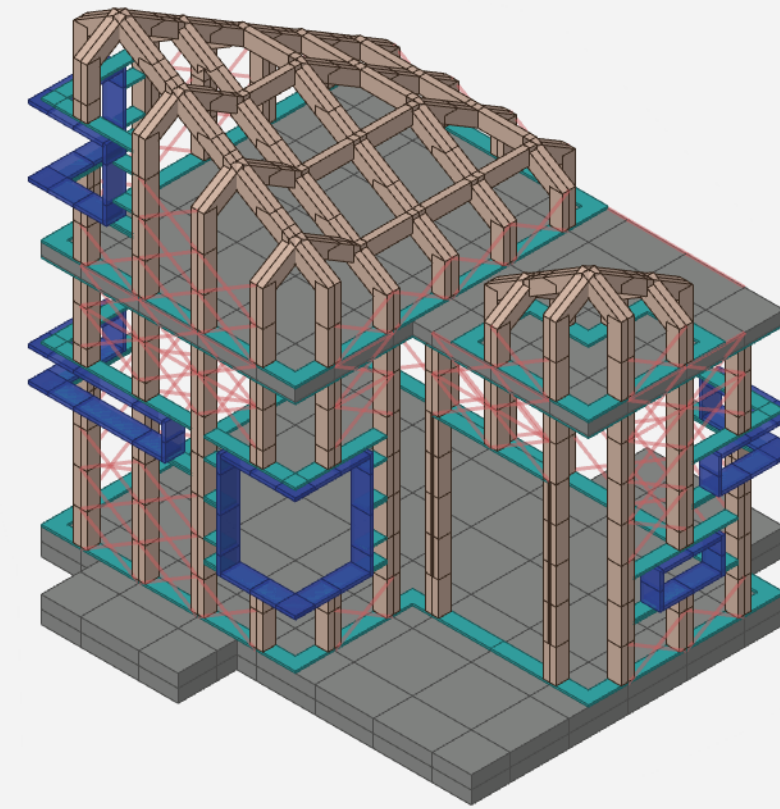
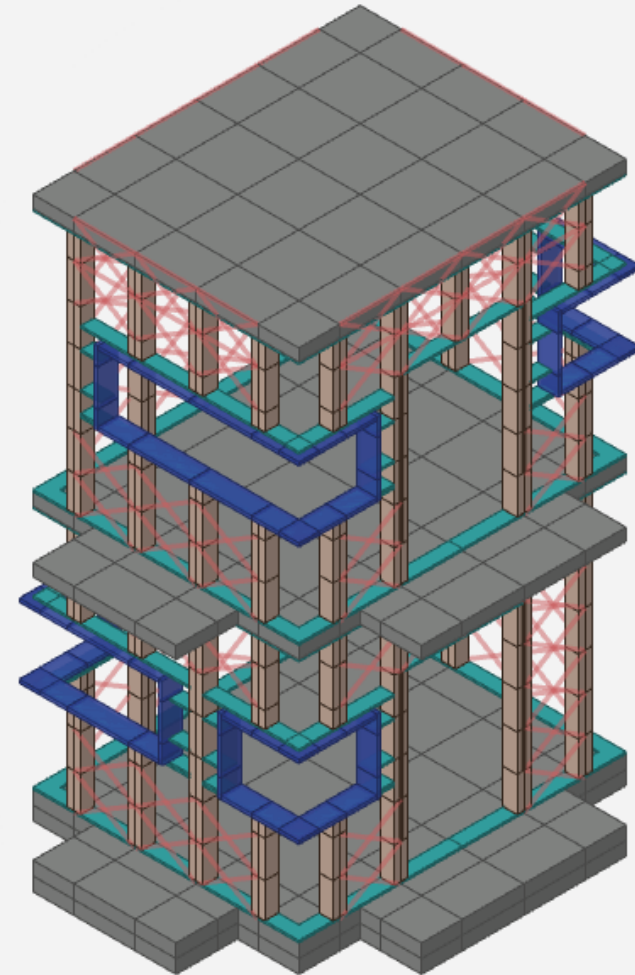
Combinatorial creativeness  
in participatorial design.



# RESULTING ARCHITECTURAL FRAMES

## 02 Polygonization of modular cube tilesets

Differentiate materials in  
different layer savings

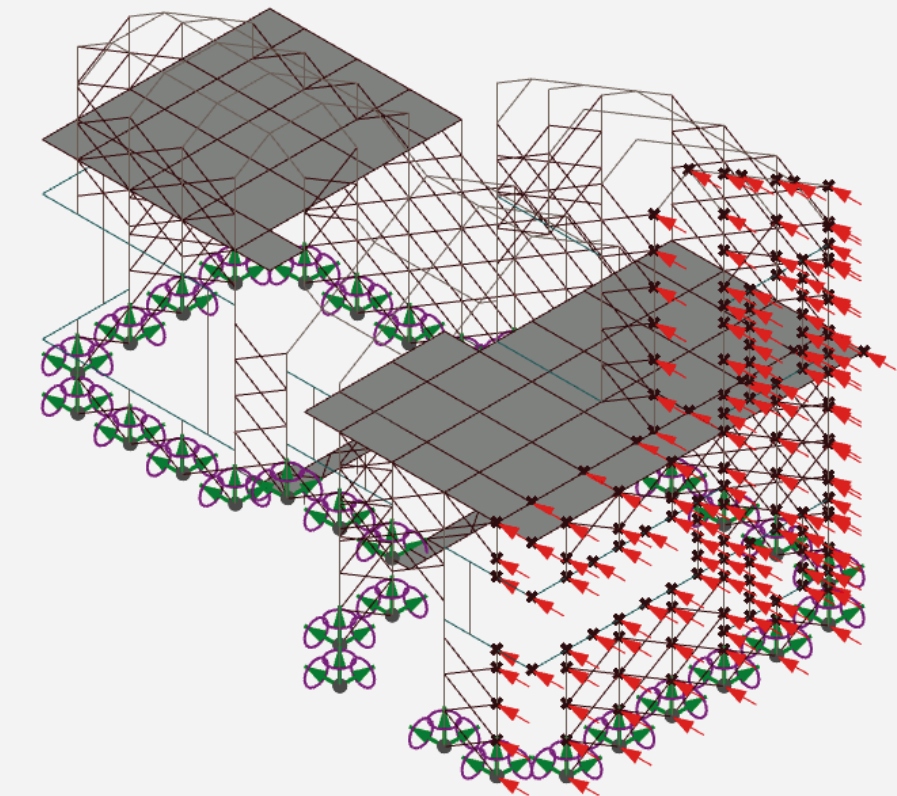
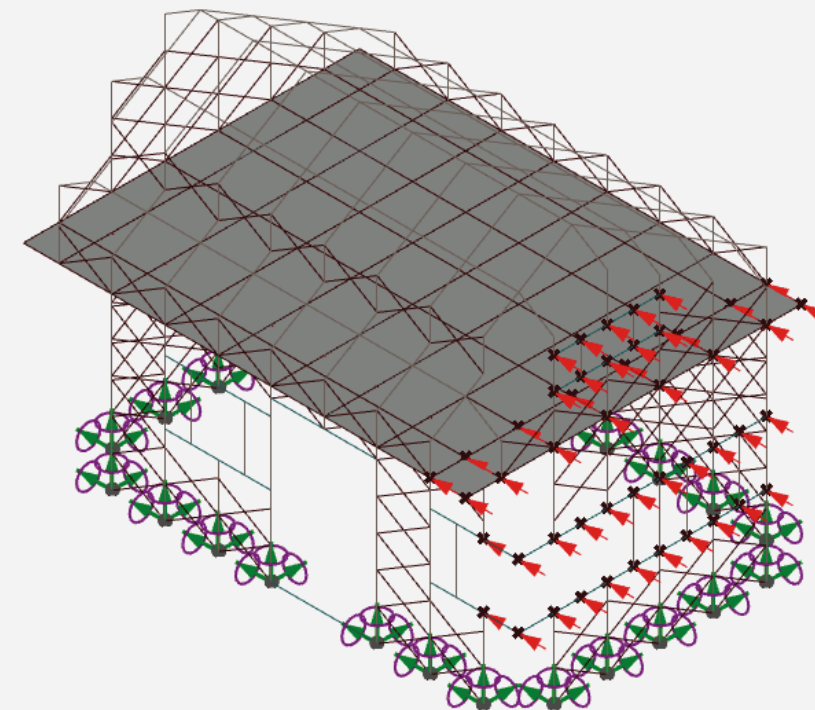
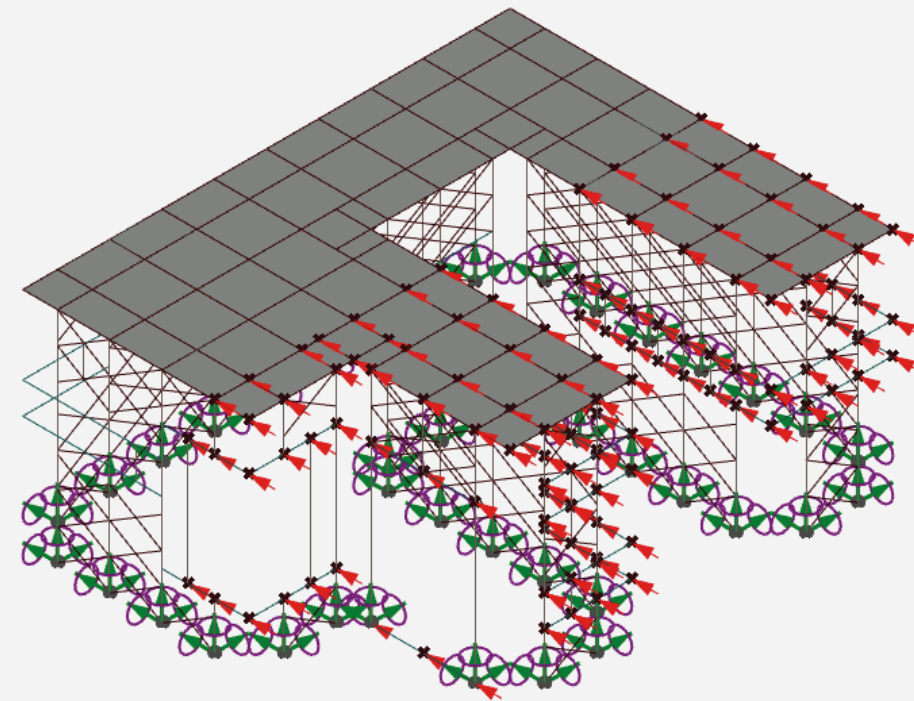
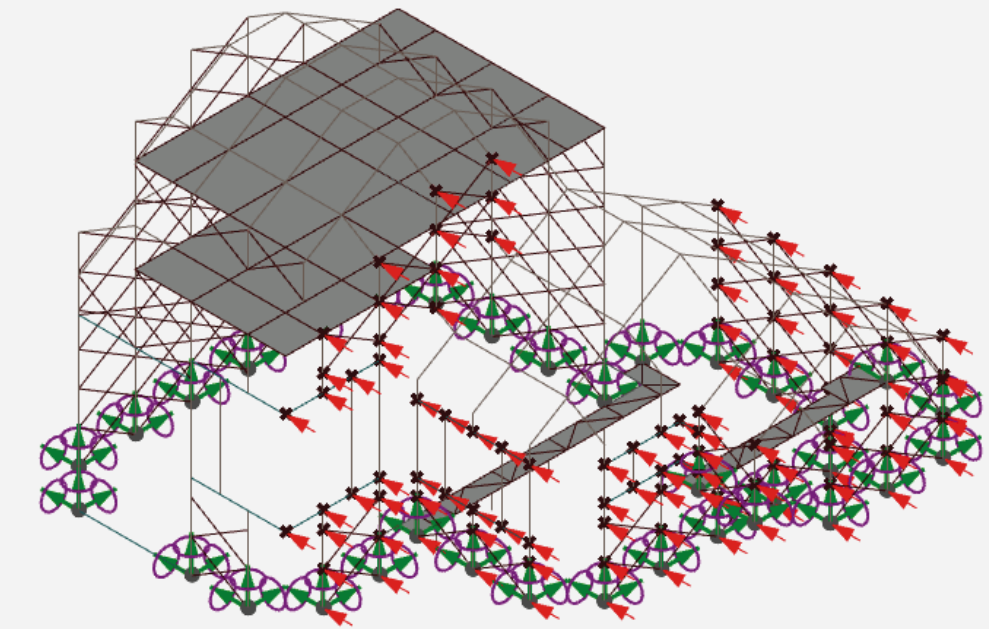
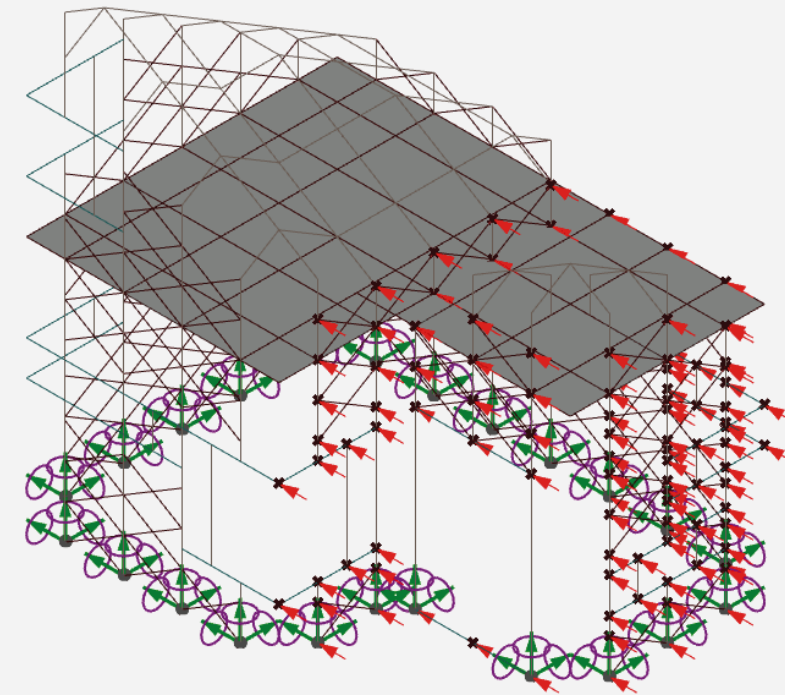
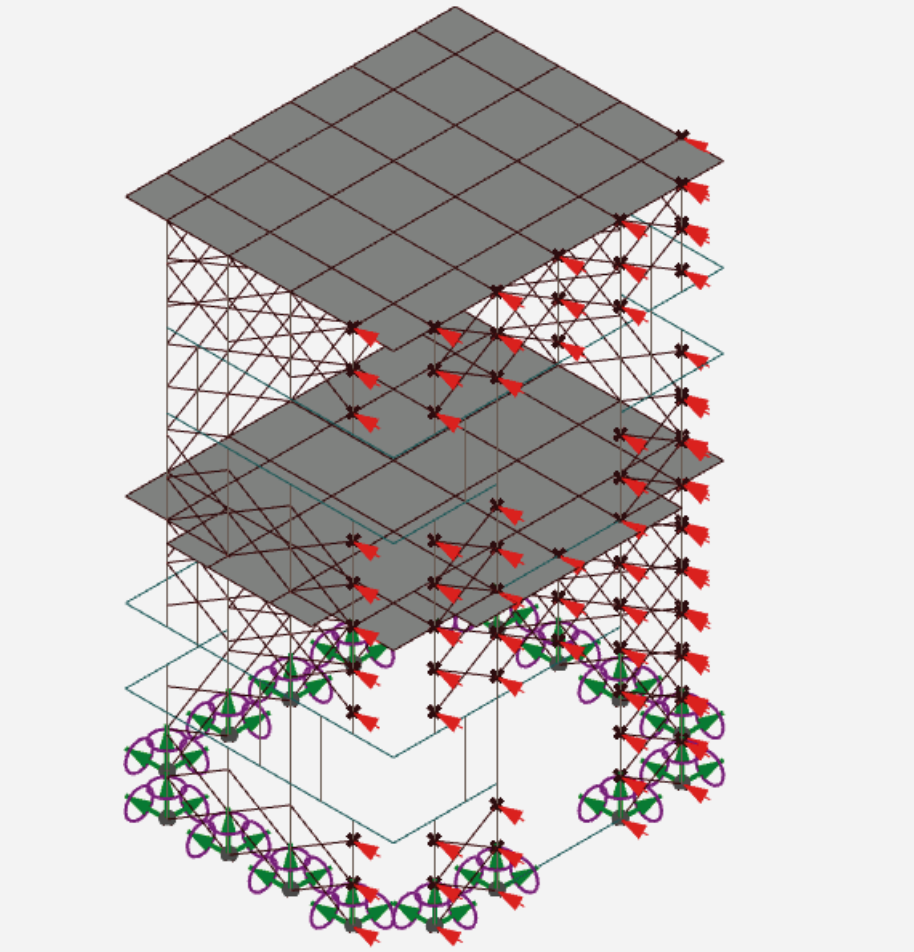


# STRUCTURAL VALIDITY CHECK

## FEM(Karamba)

### 03 Load apply

Apply of the structure  
load and materiality.



### Load Applied

1.5 G Gravity load  
1000N/m Windload on one side

# STRUCTURAL VALIDITY CHECK

## FEM(Karamba)

### 03 Structural envelope

Under assumption of  
small and medium sized  
architecture stands with  
structural envelope

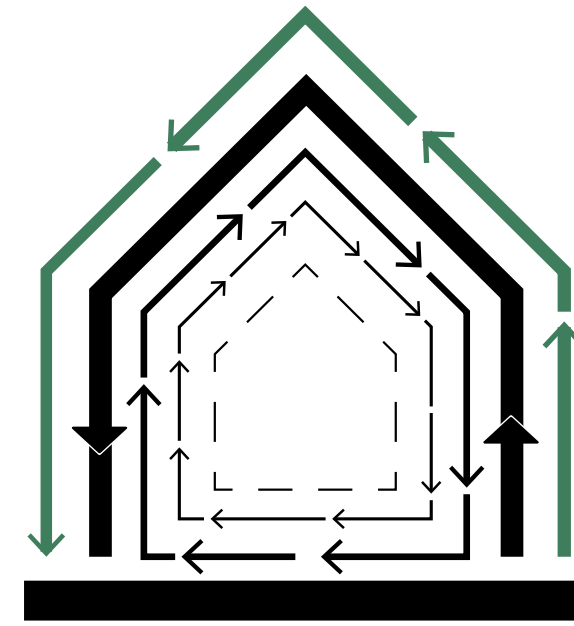
structural analysis can be  
made

## GoDesign Framework

### HOUSCAPER

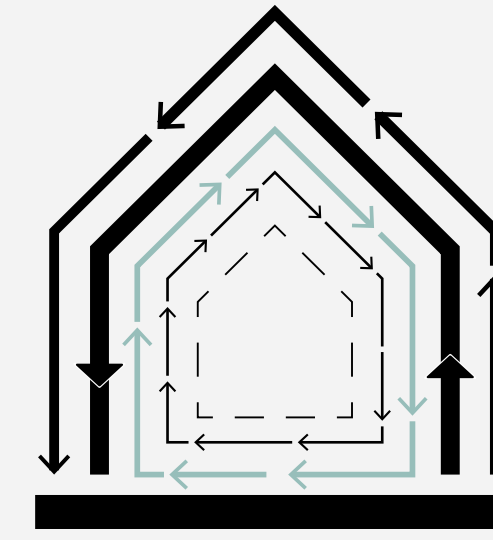


Structure > 200 years

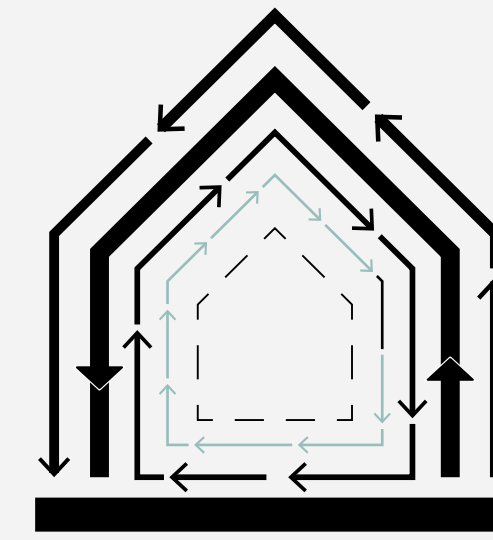


Skin  $\approx$  50 years

### Generative zoning configurations



Systems  $\approx$  25 years



Space plan  $\approx$  15 years

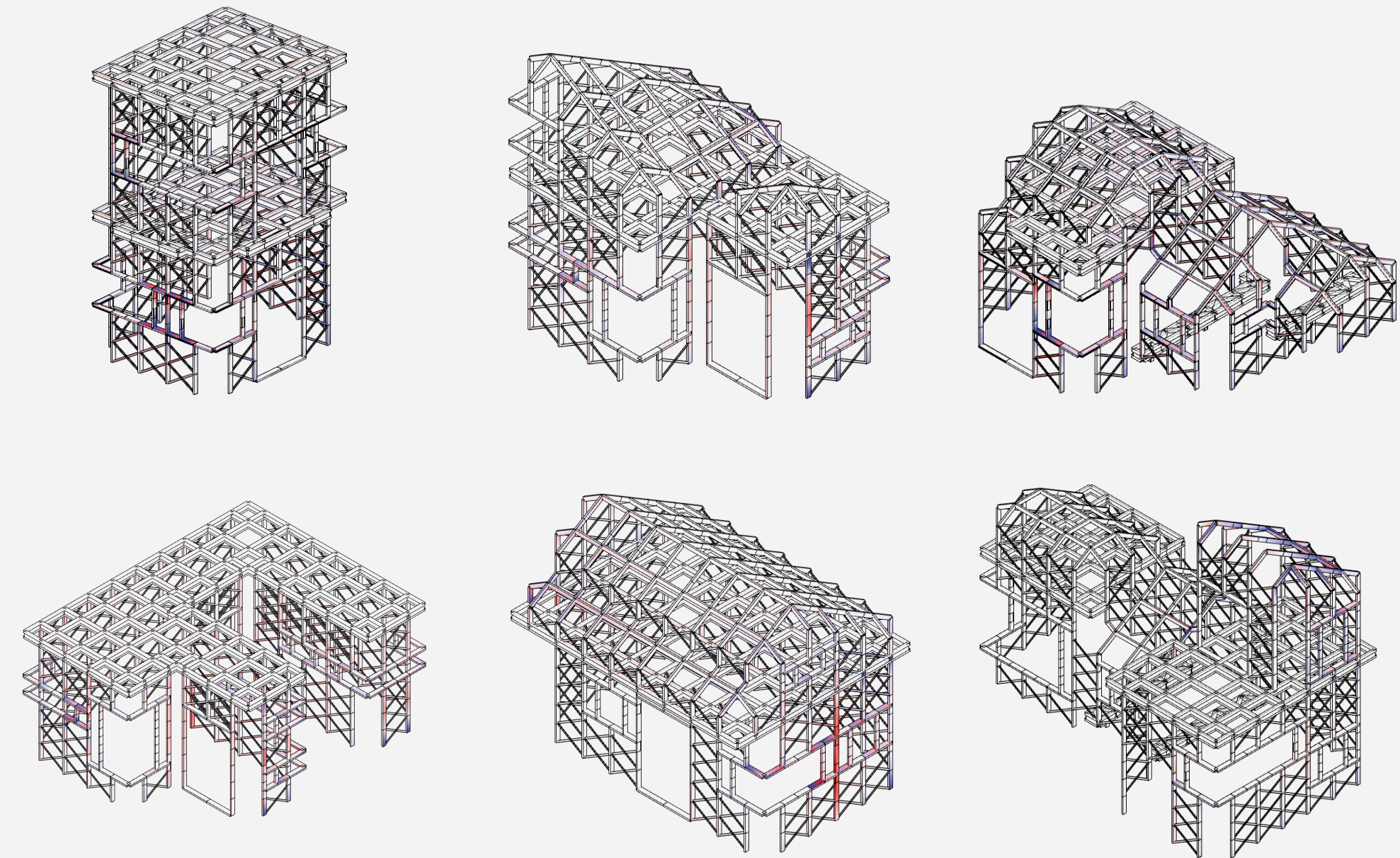
- + Freedom of interior configuration
- + less cost
- + less embodied energy

# STRUCTURAL VALIDITY CHECK

## FEM(Karamba)

### 03 material apply

Apply crosssection and materiality.



### Materiality

\* S355 Steel

Young's Module	$E$	205315 MPa
Poisson ratio	$\nu$	0.29
Yield Stress	$R_{p02}$	425 MPa
Work hardening exponent	$n$	0.165
Maximum technical stress	$R_{m\ tech}$	557 MPa
Maximum true stress	$R_{m\ true}$	639 MPa
Strain at maximum load	$\epsilon_u$	14.64%
Stress at rupture	$R_f$	514 MPa
Strain at rupture	$\epsilon_r$	15.55%

### Cross-section

Vertical Frame	Rectangular section	Height 150 cm Width 50 cm Thickness 1.6 mm
Horizontal Frame	Rectangular section	Height 50mm Width 150 mm Thickness 1.6 mm
Support Frame	Rectangular section	Height 40 mm Width 16 mm Thickness 1.6 mm
Floor Frame	I section	Height 200mm Upper width 100 mm Lower width 100 mm Thickness 1.6 mm

# STRUCTURAL VALIDITY CHECK

## FEM(Karamba)

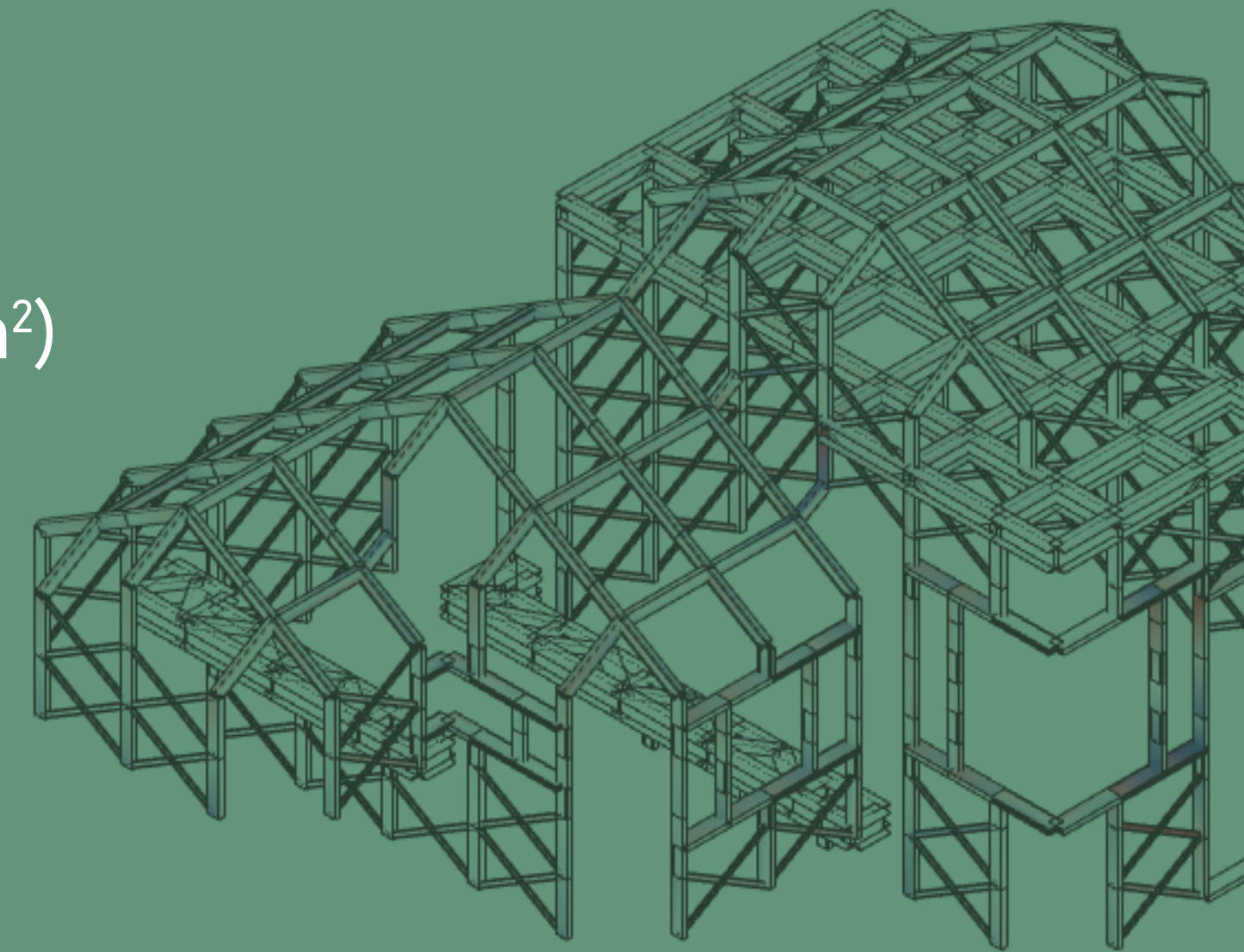
**If** structure's axial stress is less than

half of the material's technical axial stress ( $278 \text{ n/mm}^2$ )

The structure is **valid**.

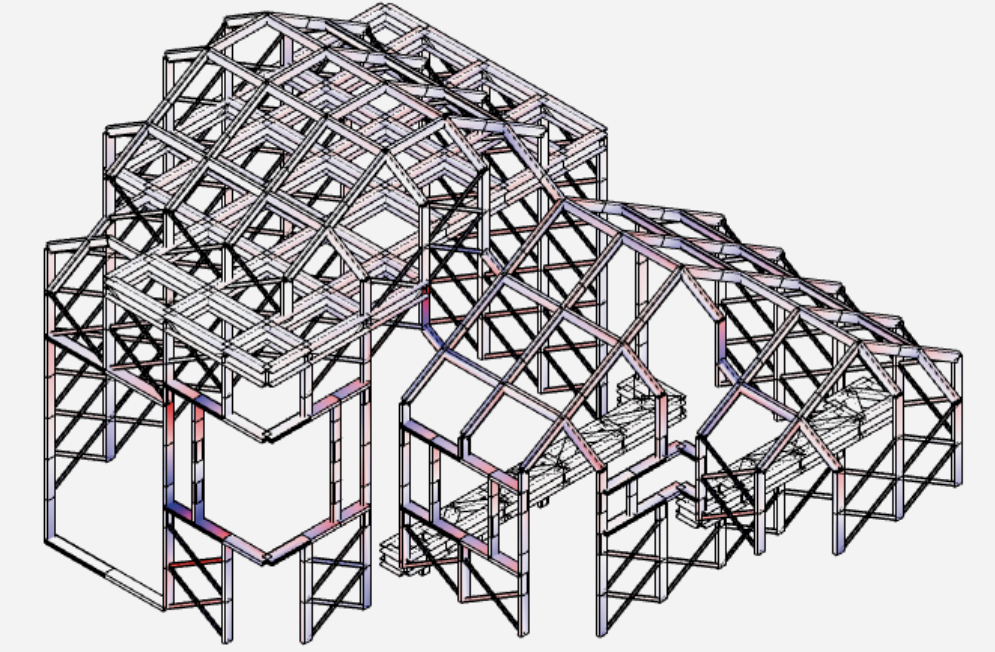
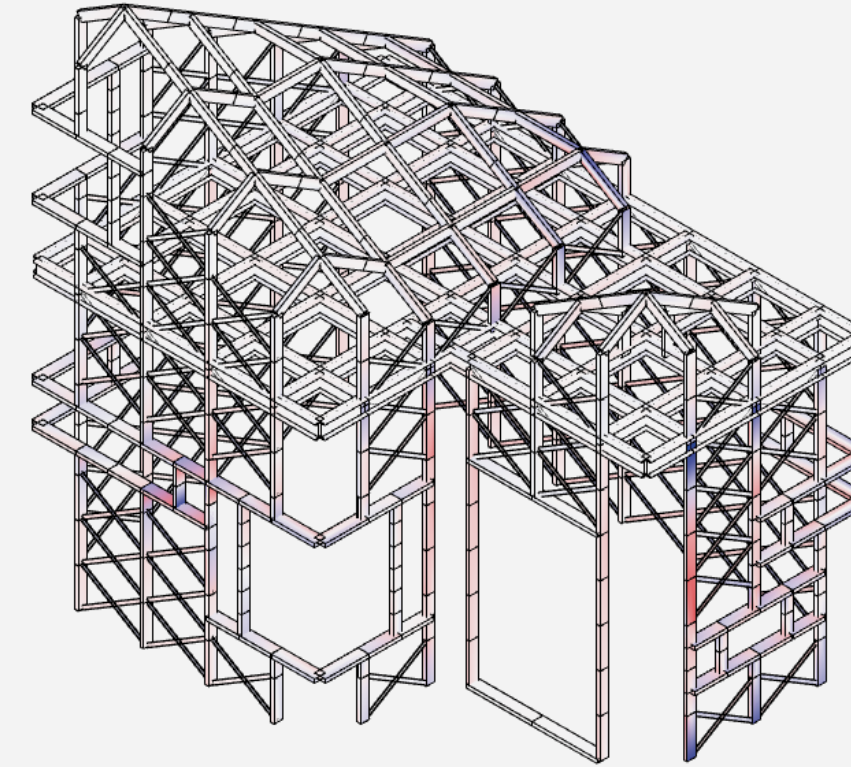
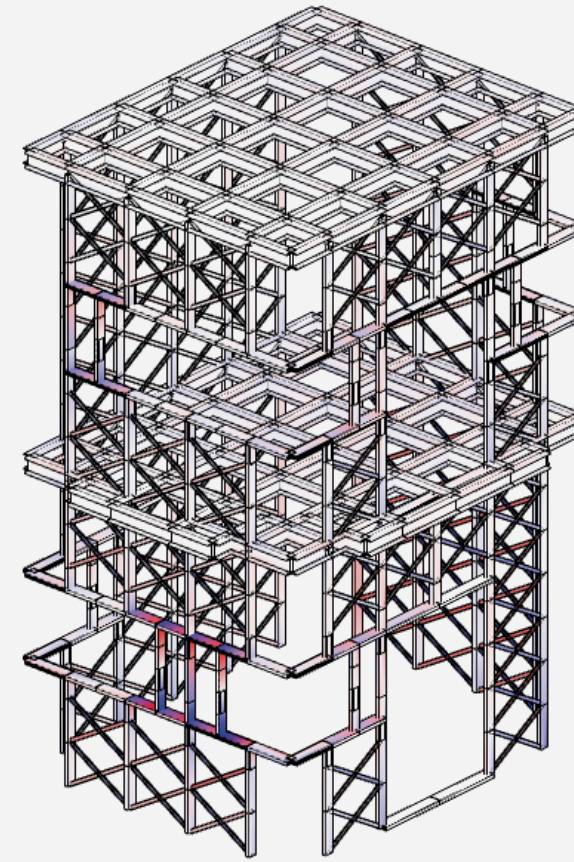
**Else,**

The structure is **not valid**.



# STRUCTURAL VALIDITY CHECK

## FEM(Karamba)



Max compression (n/mm<sup>2</sup>)  
Max tension (n/mm<sup>2</sup>)  
Max deformation (mm)

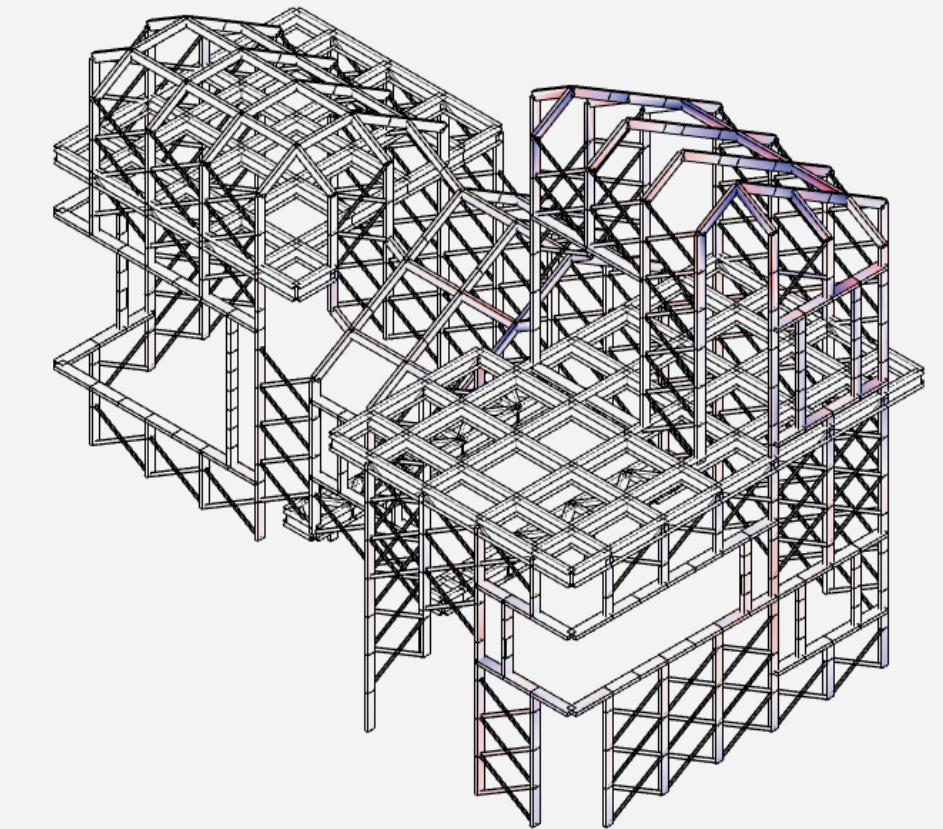
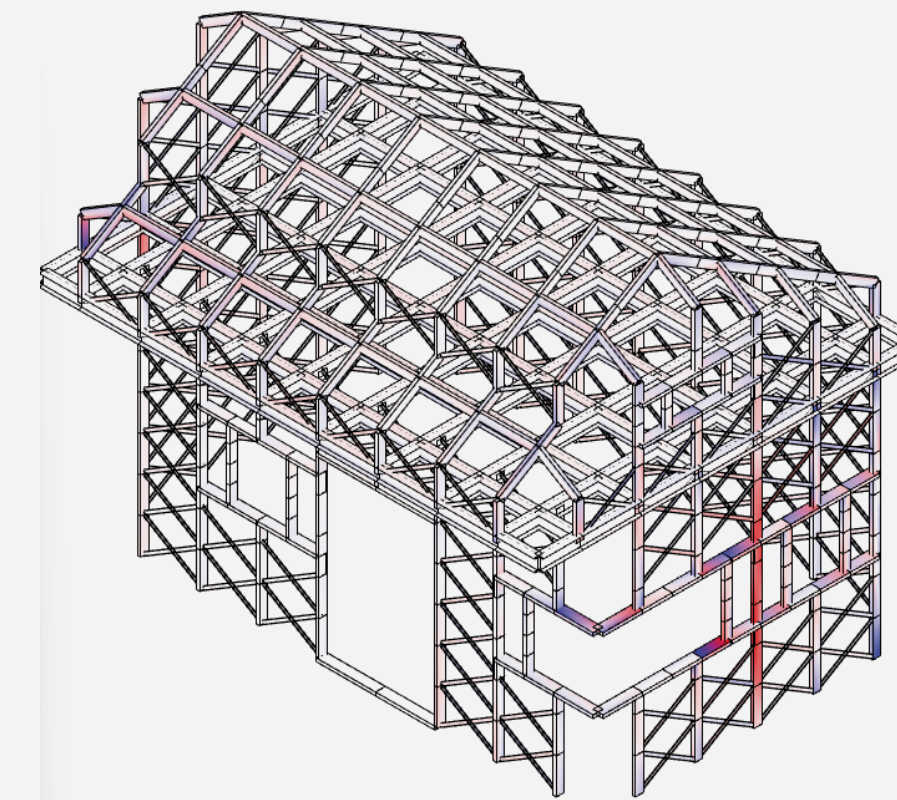
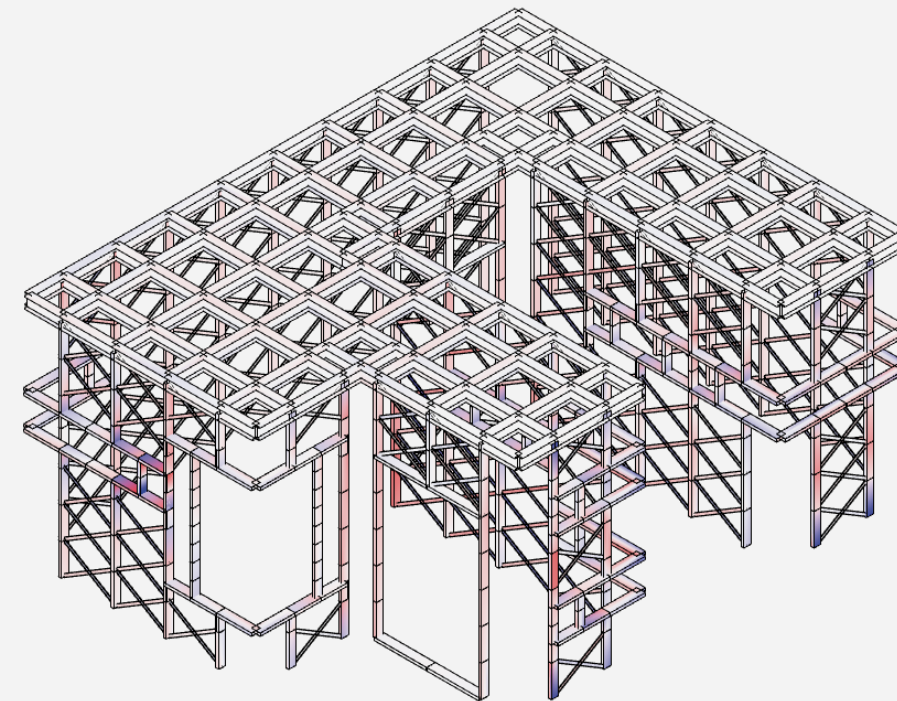
290  
286  
20



162  
145  
12



106  
68  
6



Max compression (n/mm<sup>2</sup>)  
Max tension (n/mm<sup>2</sup>)  
Max deformation (mm)

95.8  
95.2  
6.4



247  
205  
11



293  
291  
110



# CONCLUSION

HOUSCAPER

The tool **generates polygonization of the voxel array** using the tileset. As all parts are completely modular, the finished geometry can be **precisely calculated with the CPQ** (configure, price, quotation) of the material used for the frame generated.

By using a set of design rules and grouping the same geometries, **users can work with a reasonable number of tilesets**. A total of suggested 493 configurations, 109 architectural tilesets are grouped into 72 tilesets. **Expert/designer users can add more designs** by simply assigning a tileset to the system.

**The tool can visualize the finished architectural frame and run a structural analysis** of the structural envelope, providing an instant validity check of the structural frame. The structural analysis script allows alternation of material and crosssections of linear building elements providing expert users with freedom of choice in materiality.



## Limitations

The research has mainly focused on the polygonization aspect of modular housing using a voxel array. The modeling of the parts remains basic to give users a simple visual impression of the tool.

The structural analysis also works with linear modular components by assigning crosssections and material properties in FEM analysis.

However, site property variations and specifications are ignored to bring more generality to the project.

## Future works

1. A web version, a lighter version of the tool designed for the mass as a publishable SaaS (software as a Service), enables public access to the device for both the mass and the experts.
2. Development of a generative design method for the voxel array design. Already optimization methods introduced in GoDesign show possibilities to generative design the zone configuration.
3. Develop a script enabling the parts outside of Module sizes, such as the truss inside th wall and roof. The trusses connecting edge-to-edge point of frame sizes often exceed the modules' size.

THANK

YOU.

