The Long Game

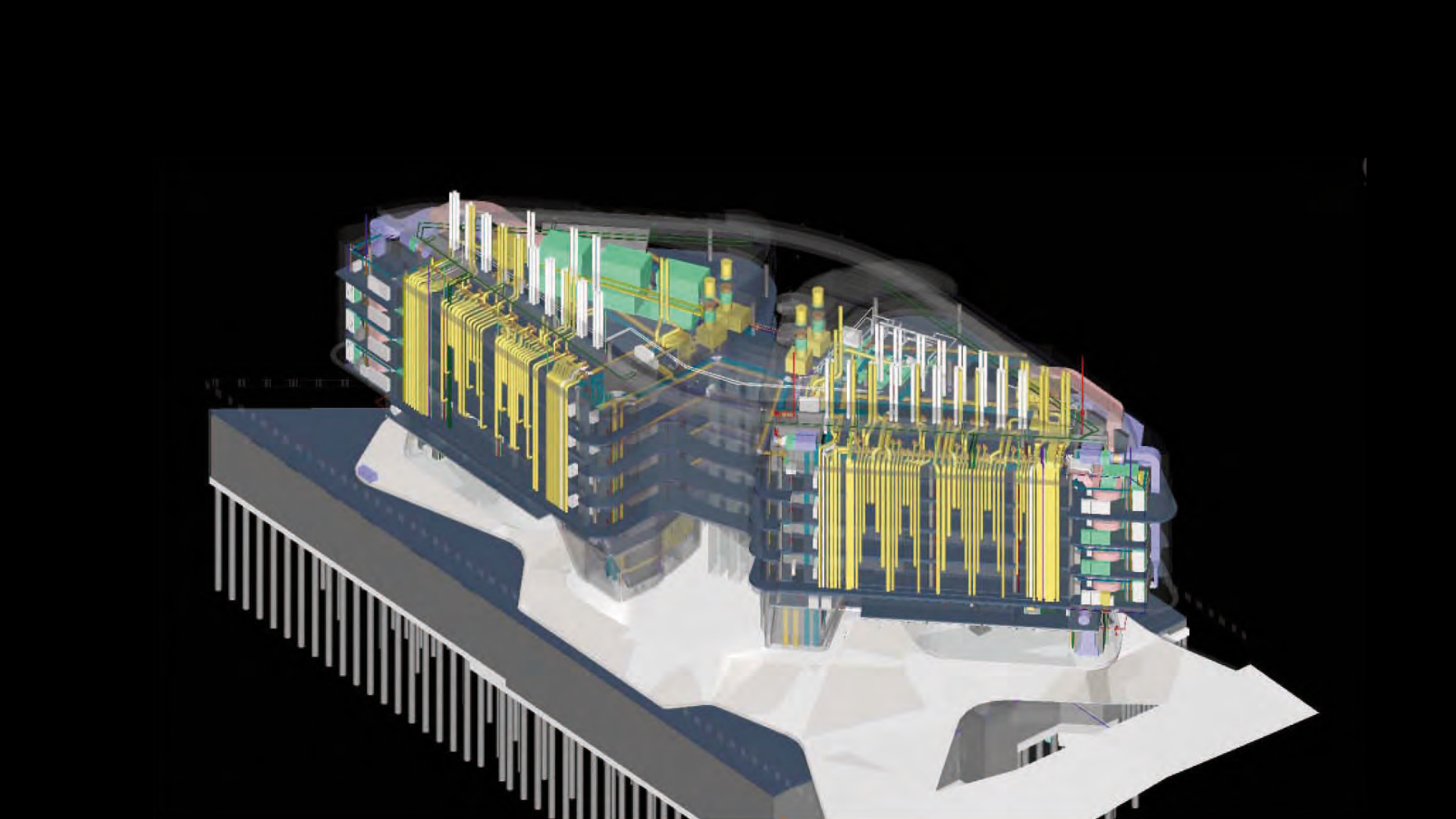
Maintaining Complex Interoperable Workflows Through Multiple Phases and Teams

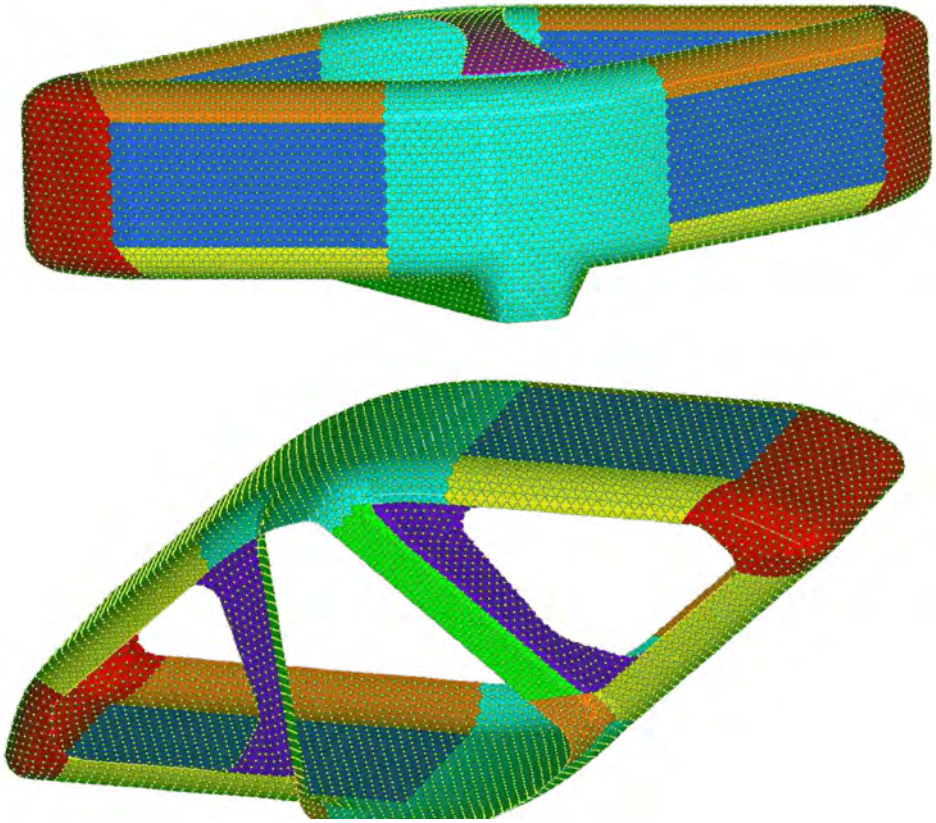
Shane Burger
Principal, Global Director of Technical Innovation, WOODS BAGOT

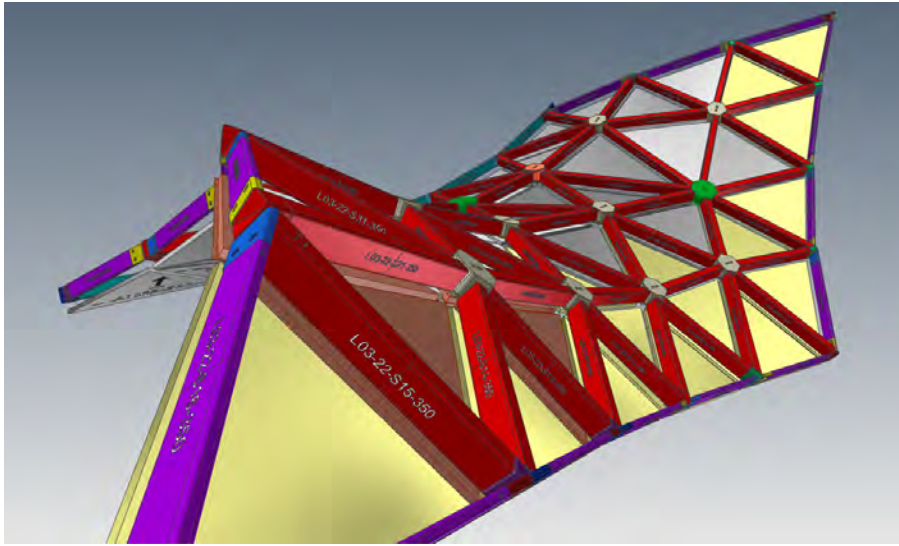
Join the conversation #AU2017 @WB_DigitalCraft @shaneburger @woodsbagot

AUTODESK. UNIVERSITY

WOODS BAGOT

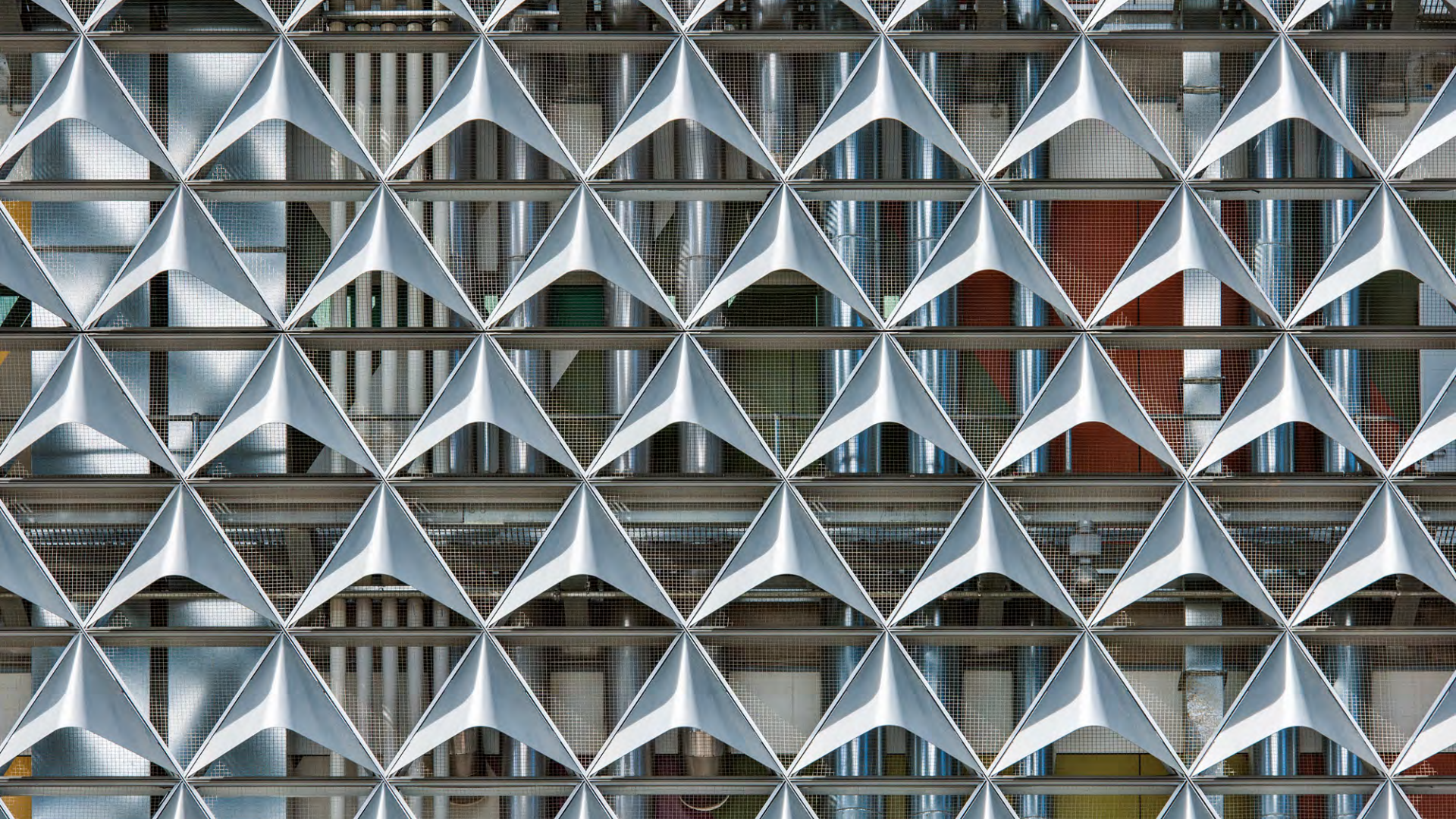


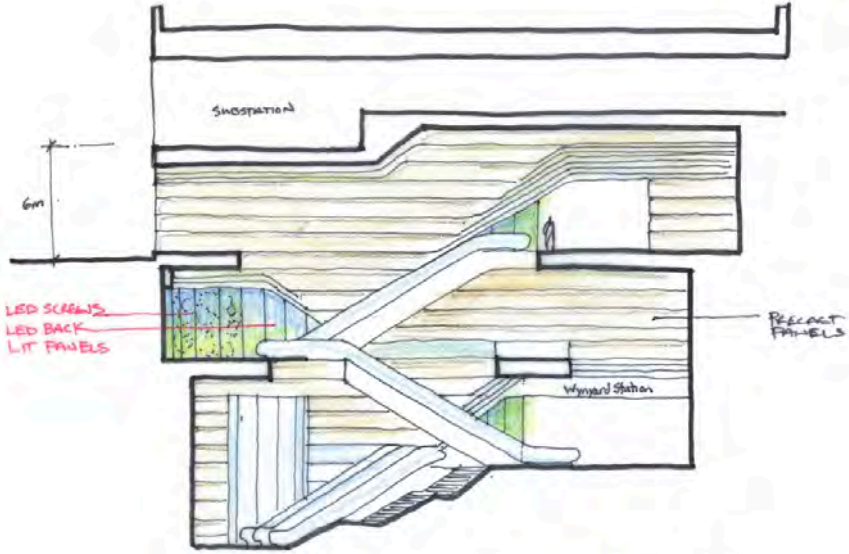



L03-22-S15-350

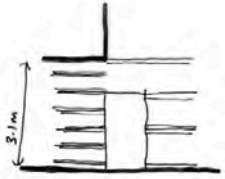

规
范

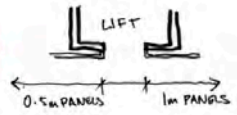

L8 02
L7 16



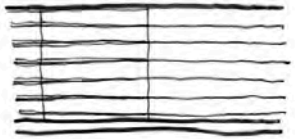


SECTION OP 1
19.04.12

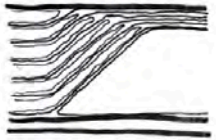


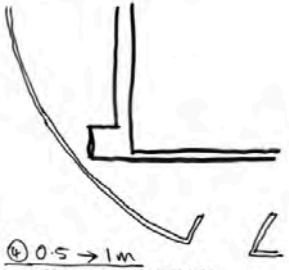


① CONCOURSE LEVEL ② LIFT



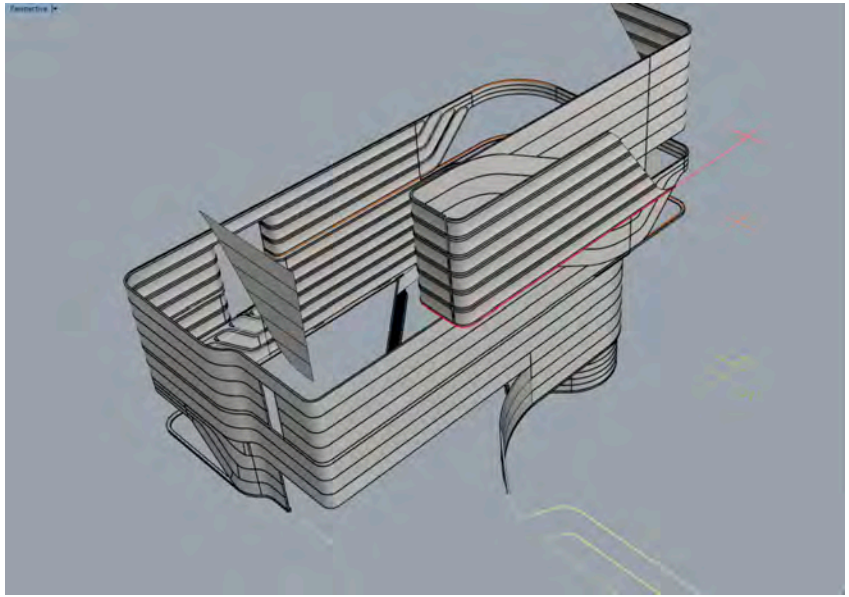
② PRECAST WALLS THAT FLATTEN
③ WYNYARD STN ENTRY

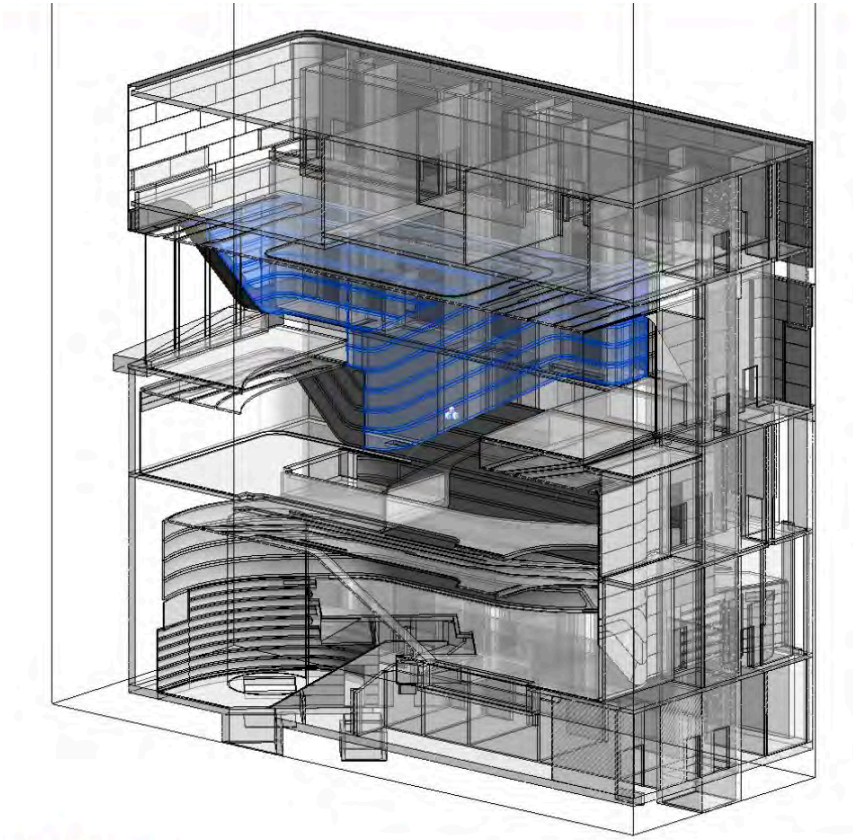


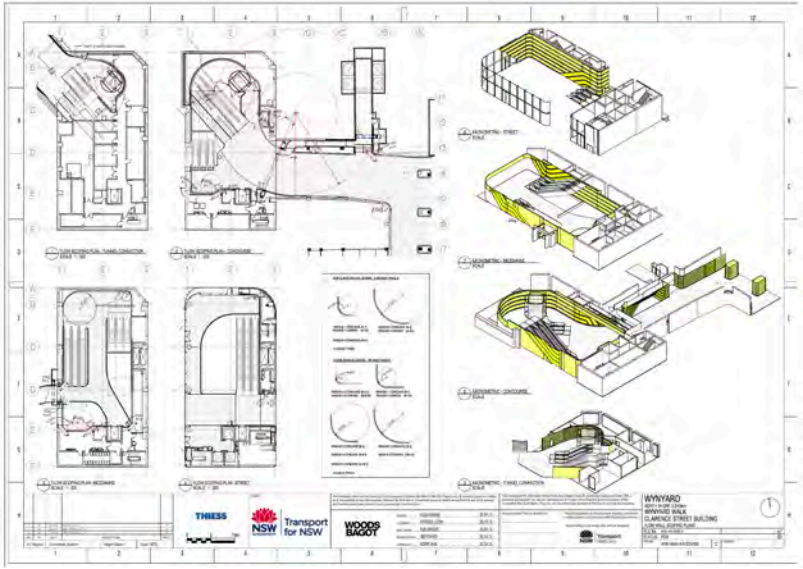


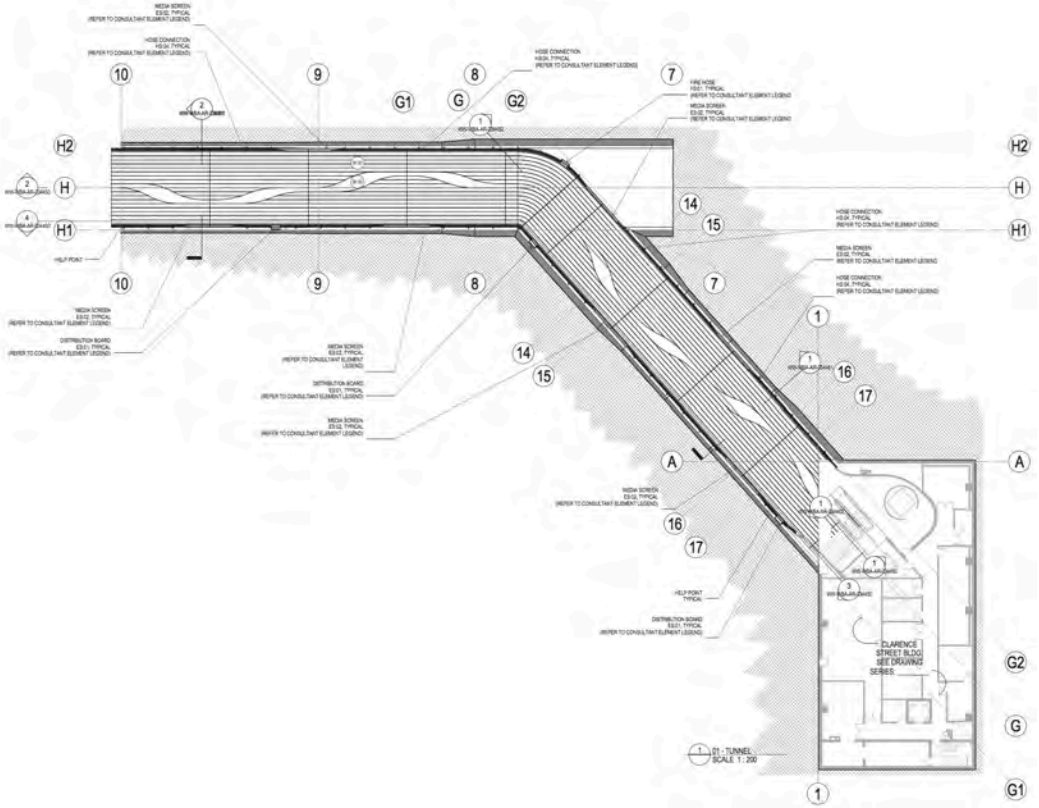
③ MEDIA SCREEN ②
RAILWAY HOUSE

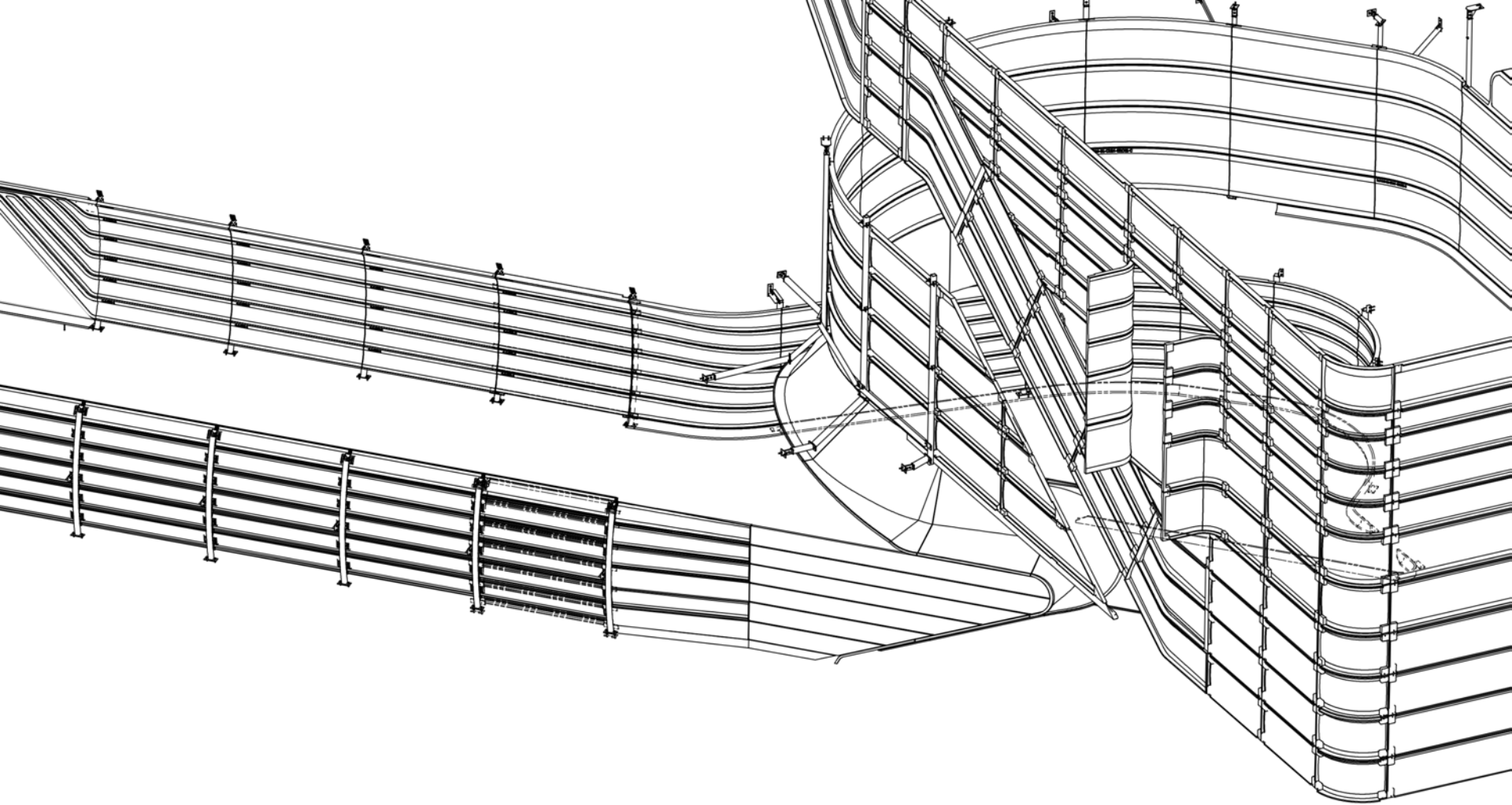
④ 0.5 → 1m
TRANSITION PANEL



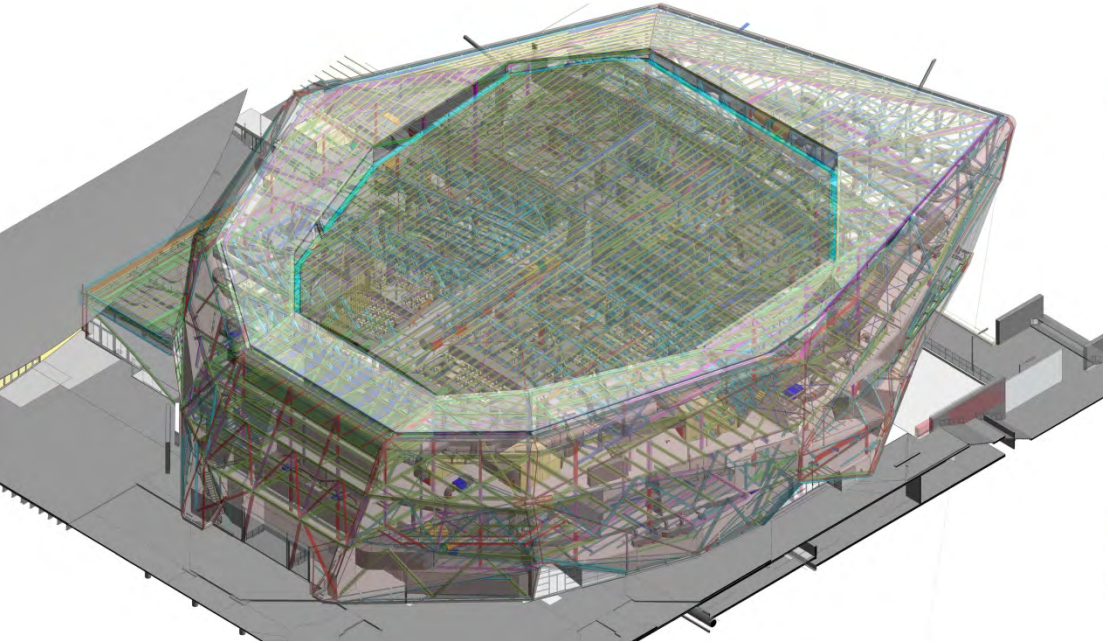


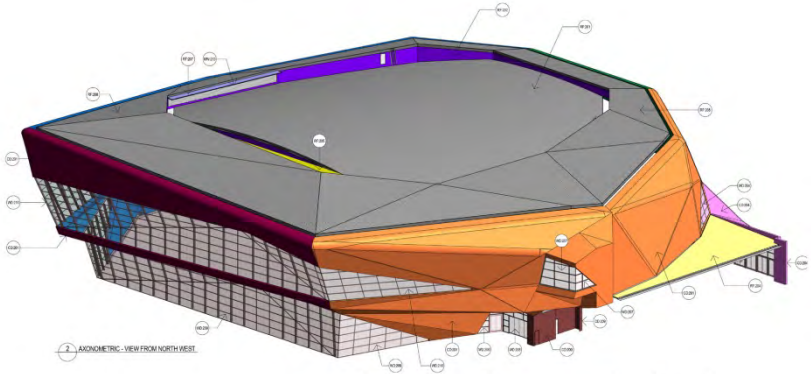

THIESS
NSW Transport for NSW
WOODS BAGOT
WYNYARD
WYNYARD WALK
CLARENCE STREET BUILDING


MEDIA SCREEN
E3.02, TYPICAL
(REFER TO CONSULTANT ELEMENT LEGEND)
HOSE CONNECTION
H3.04, TYPICAL
(REFER TO CONSULTANT ELEMENT LEGEND)
FIRE HOSE
H3.01 - TYPICAL
(REFER TO CONSULTANT ELEMENT LEGEND)
HELP POINT
DISTRIBUTION BOARD
E3.01, TYPICAL
(REFER TO CONSULTANT ELEMENT LEGEND)
CLARENCE STREET BLDG SEE DRAWING SERIES
01 - TUNNEL
SCALE 1 : 200

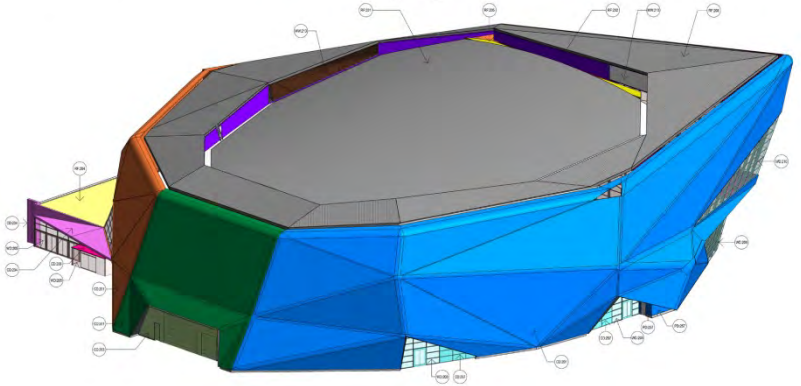



phasis of a transport exchange
cy of travel to the quality of the experience.

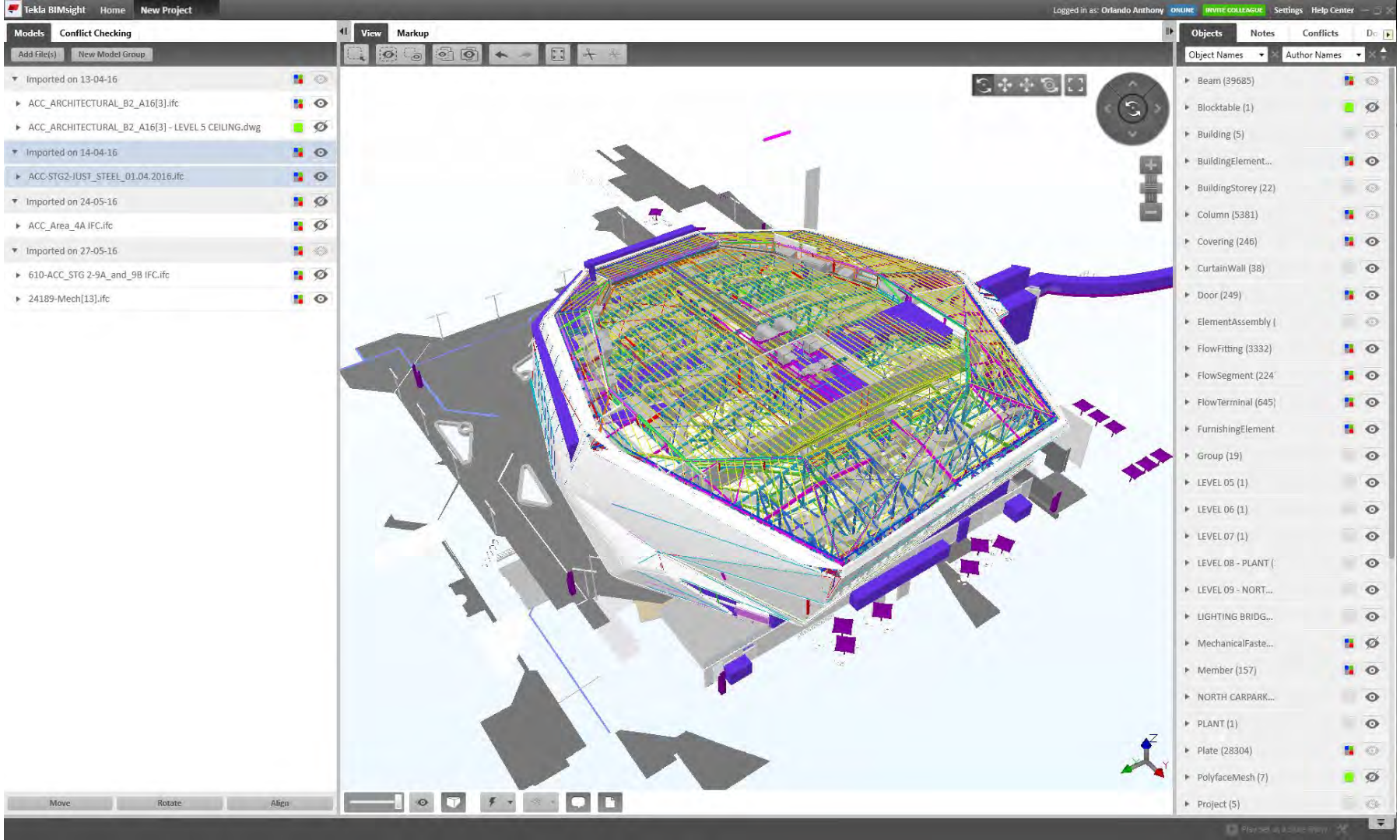


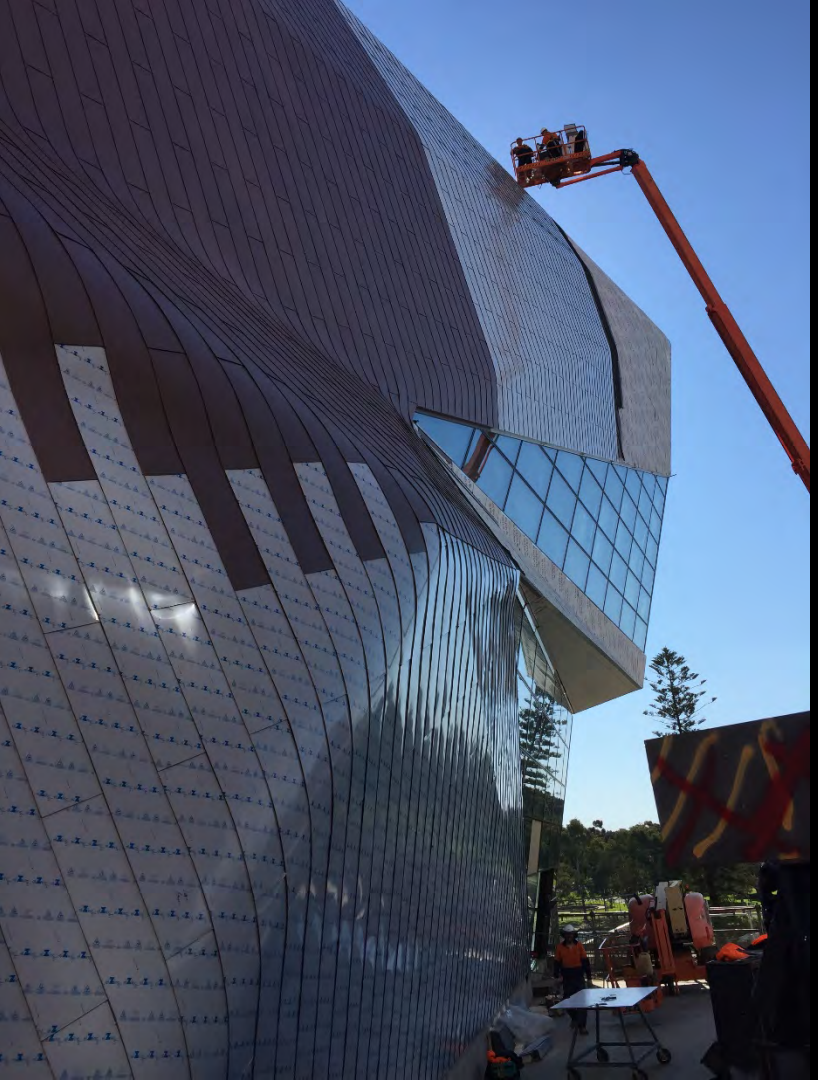


2 AXONOMETRIC - VIEW FROM NORTH WEST



3 AXONOMETRIC - VIEW FROM SOUTH EAST


Tekla BIMsight
Home
New Project
Logged in as: Orlando Anthony
ONLINE
INVITE COLLEAGUE
Settings
Help Center
Models
Conflict Checking
Add File(s)
New Model Group
Imported on 13-04-16
ACC_ARCHITECTURAL_B2_A16[3].ifc
ACC_ARCHITECTURAL_B2_A16[3] - LEVEL 5 CEILING.dwg
Imported on 14-04-16
ACC-STG2-JUST_STEEL_01.04.2016.ifc
Imported on 24-05-16
ACC_Area_4A IFC.ifc
Imported on 27-05-16
610-ACC_STG 2-9A_and_9B IFC.ifc
24189-Mech[13].ifc
Move
Rotate
Align
View
Markup
Objects
Notes
Conflicts
Object Names
Author Names
Beam (39685)
Blocktable (1)
Building (5)
BuildingElement...
BuildingStorey (22)
Column (5381)
Covering (246)
CurtainWall (38)
Door (249)
ElementAssembly (
FlowFitting (3332)
FlowSegment (224
FlowTerminal (645)
FurnishingElement
Group (19)
LEVEL 05 (1)
LEVEL 06 (1)
LEVEL 07 (1)
LEVEL 08 - PLANT (
LEVEL 09 - NORT...
LIGHTING BRIDG...
MechanicalFaste...
Member (157)
NORTH CARPARK...
PLANT (1)
Plate (28304)
PolyfaceMesh (7)
Project (5)




NTRE


ADELAIDE CONVENTION CENTRE

WOODS BAGOT
DESIGN TECHNOLOGY

is excited to announce

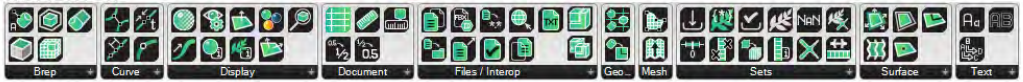
WOMBAT

WomBIM for Revit

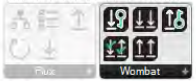
Wombat Dynamo

Wombat Grasshopper

WOMBATGH



WOMBATGH FLUX



WOMBATGH HUMAN UI

WOMBAT DYNAMO

BoundingBox
Dimensions
Evaluate

Column
CoordinateSystem
IsSlanted

CurtainGrid
AddGridLine
ByWall
GetPanels

Curve
SmartOffset

Element
ById
FilterByParameterValue
FilterByType
GetPropertyNames
GetType
TryGetPropertyValue
Unpin

FamilyInstance
ByPointHostTypeAndLevel
IsInPlace
Orientation
SetType

IO
ExportAsDWG
ExportAsFBX

Level
ByName

List
FilterByName
NullItem
Unwrap
Wrap
NullToEmptyList

Opening
ByPathTypeAndLevel

Plane
Deconstruct

Point
Deconstruct

Polyline
ByPoints
ByPolyCurve
ClosestPoint
Reduce
ToPolyCurve
Count
Vertices

String
Format

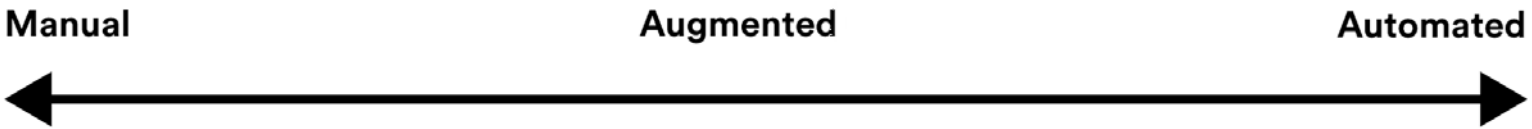
Vector
Deconstruct

Wall
AllowDisallowJoins
ByCurveAndLocation
ByProfile
InsertDoorOrWindow

BuildingPad
ByOutlinesTypeAndLevel
ByOutlineTypeAndLevel
ByName
Name

Railing
ByPathTypeAndLevel
ByName
Name

Roof
ByProfile
ByEdgesAndSlopes
ByFootprint

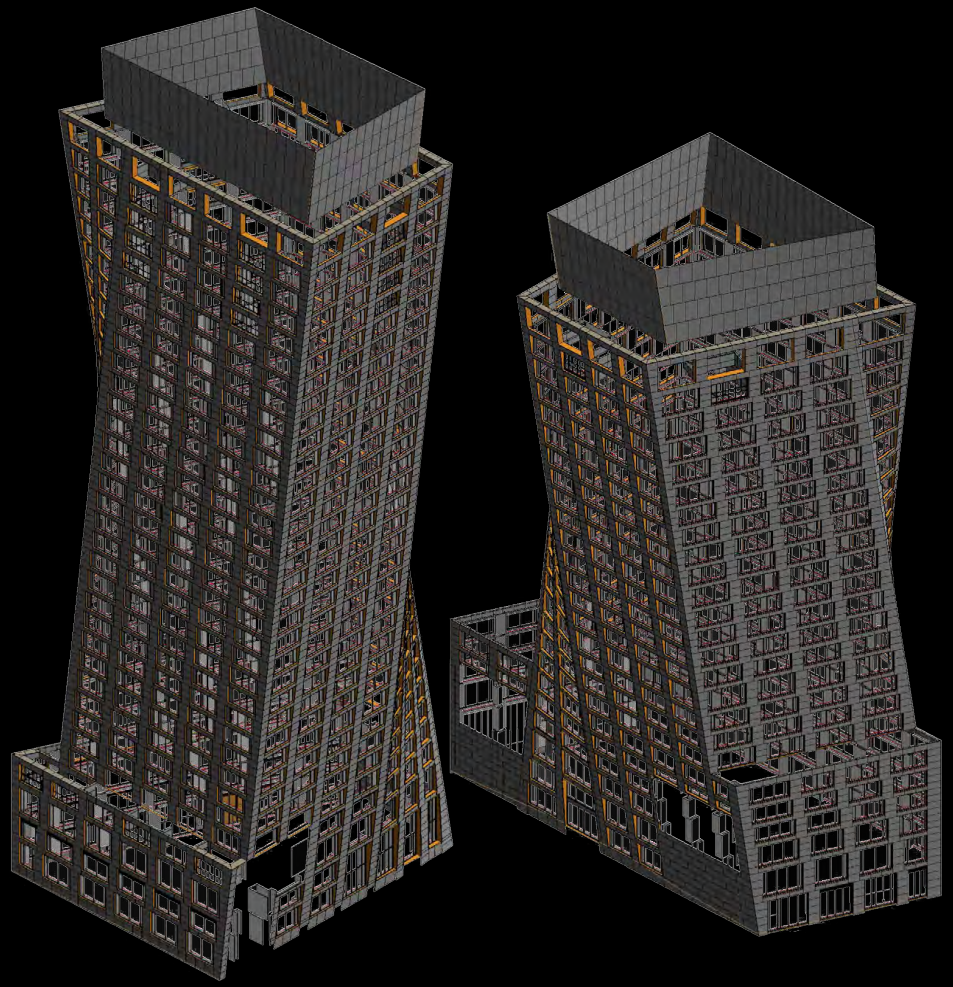


WOODS BAGOT
DESIGN TECHNOLOGY

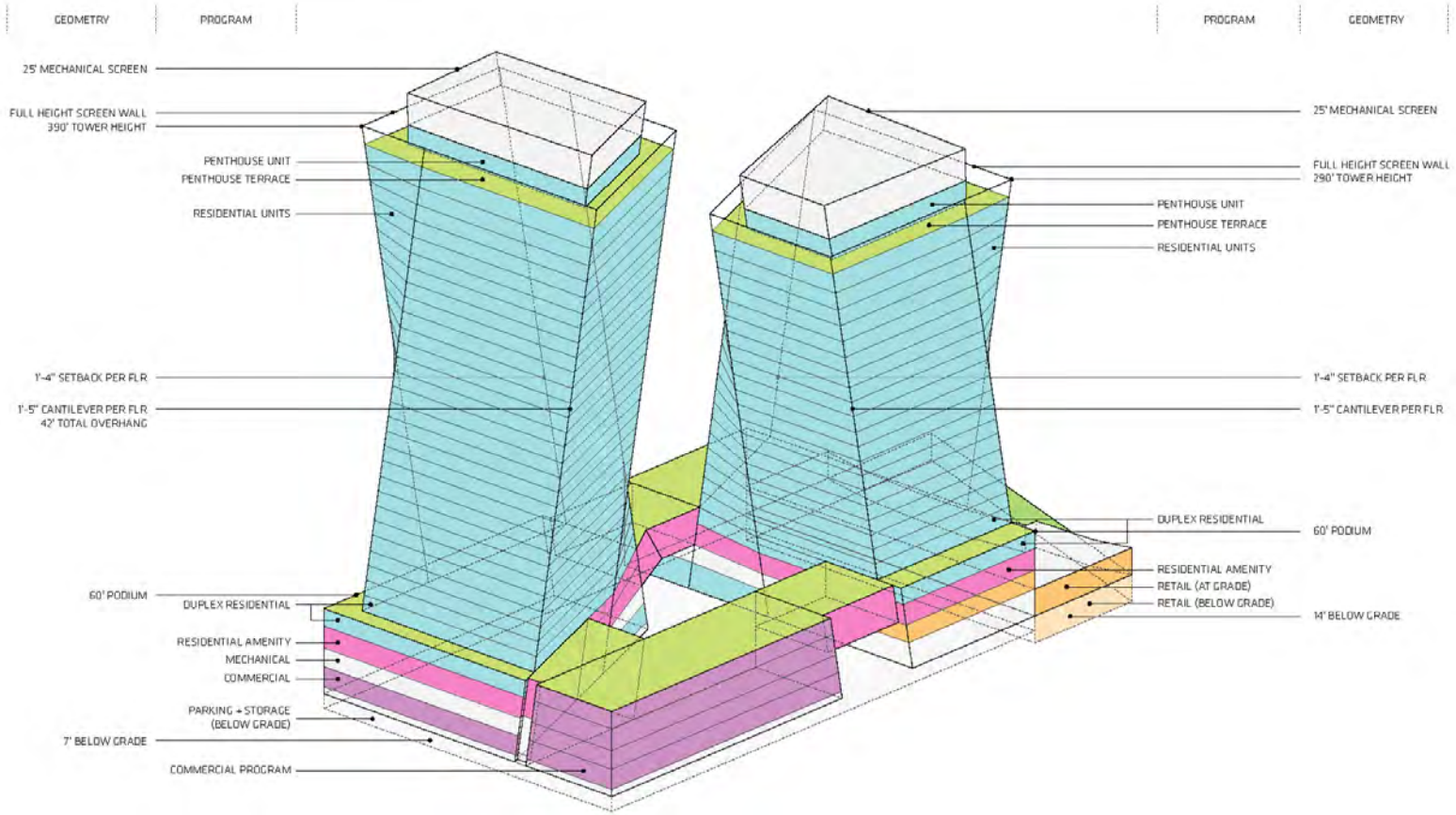
76 Eleventh Avenue

BIG
Woods Bagot
WSP

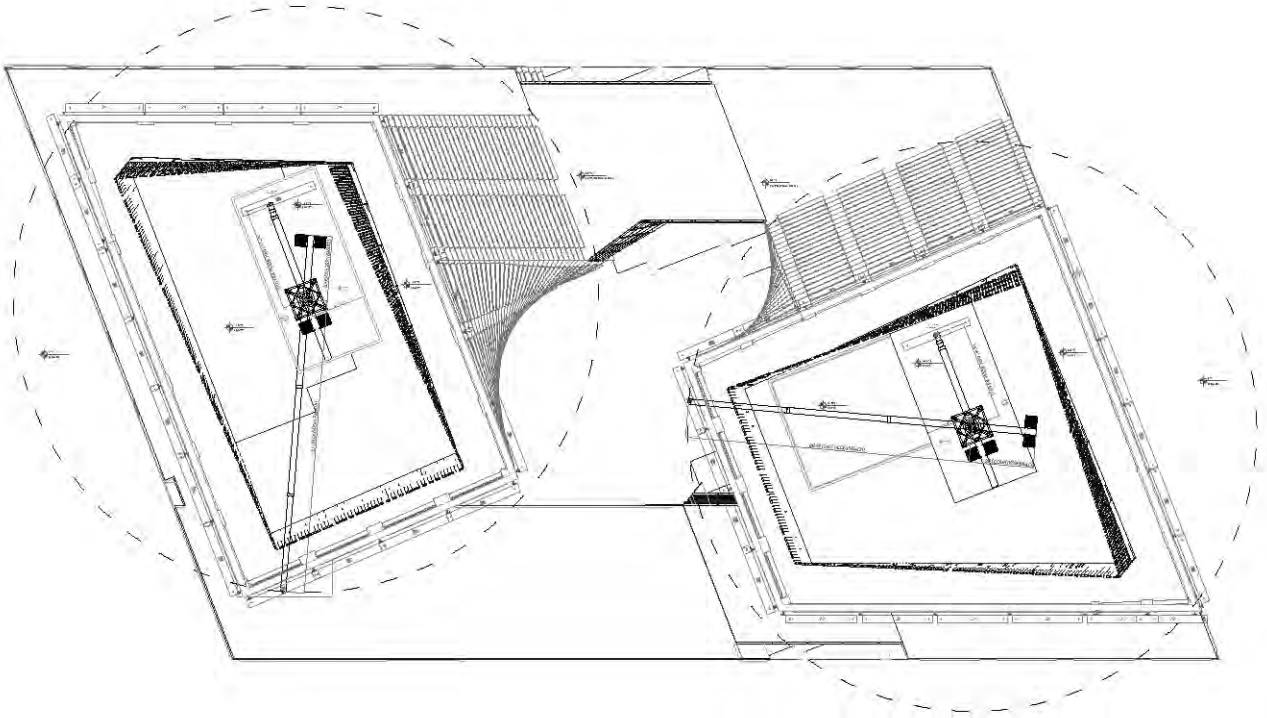
Rhino | Grasshopper | Flux | Dynamo | Revit



Design
Geometry + Program



Design
Geometry / Plan



Design
Facade

Tools Required

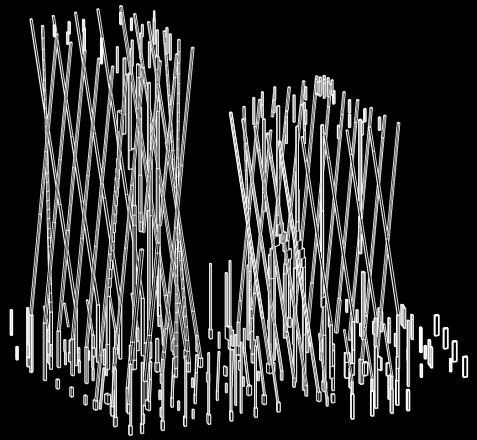
RHINO I GRASSHOPPER I FLUX I DYNAMO I REVIT

Flux.Grasshopper
Telepathy
MetaHopper
treesloth
Human
Human UI
HDTTreeUtils

Flux.Revit

Inputs

Three Base Inputs

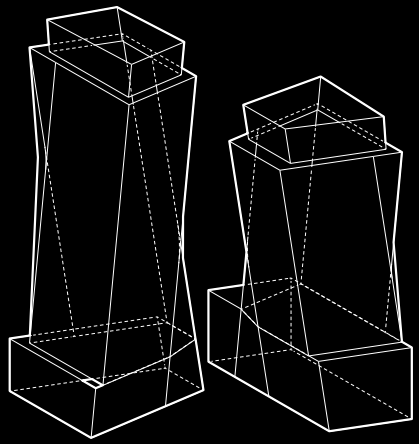


Structural.rvt

Structural Revit model provided by WSP

Architecture.rvt

Levels provided by Woods Bagot in base architectural Revit model



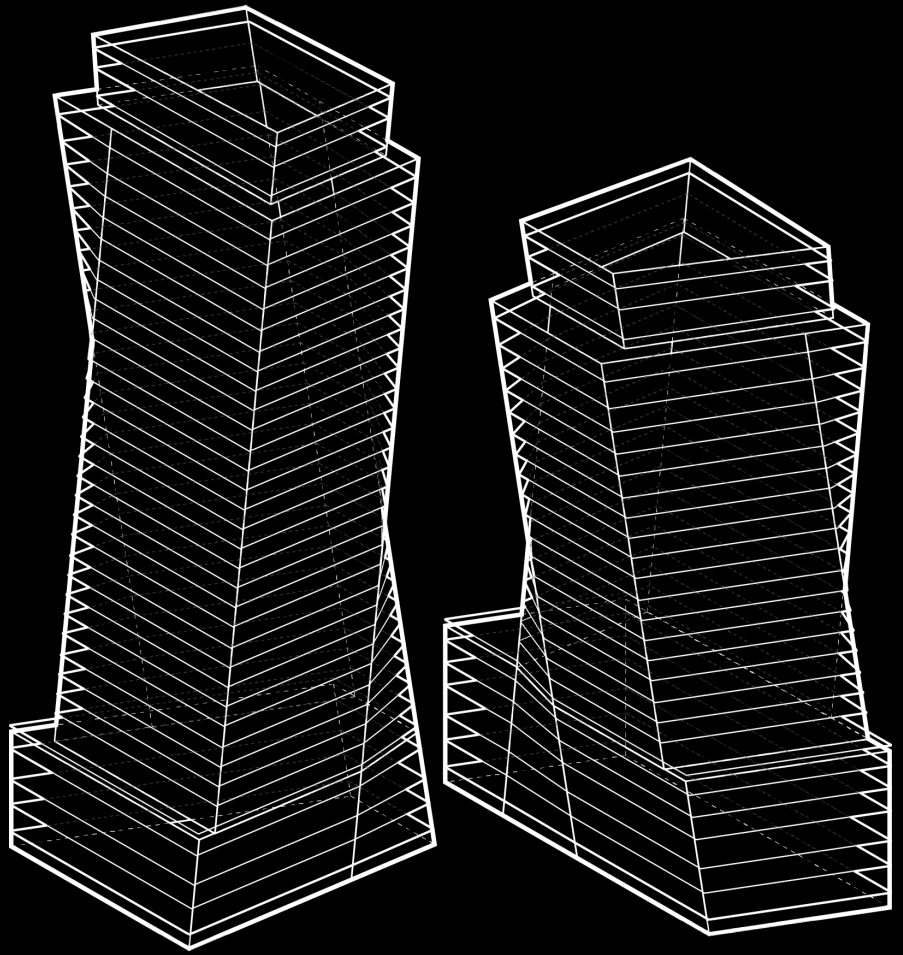
Massing.3dm

Massing Rhino model provided by BIG

Rhino | Grasshopper | Flux | Dynamo | Revit

Massing + Levels

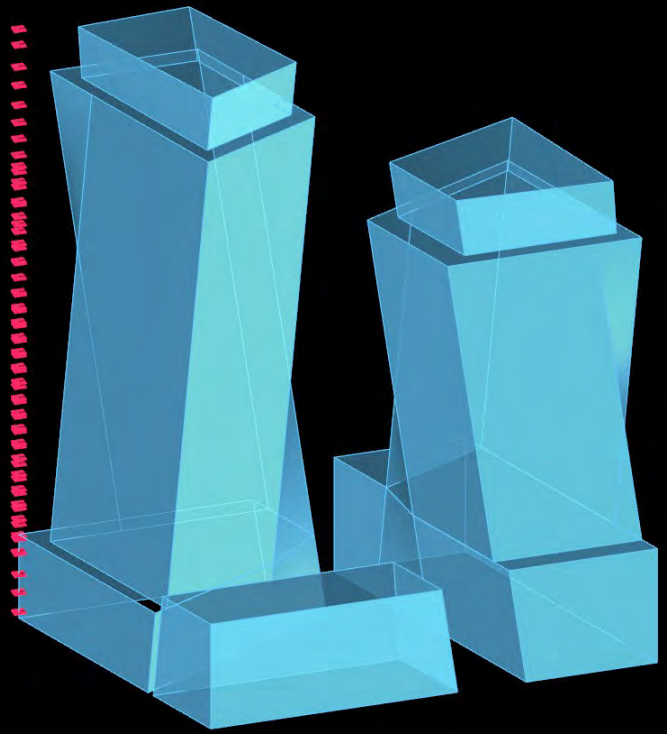
Resilient systems for iteration



Rhino | Grasshopper | Flux | Dynamo | Revit

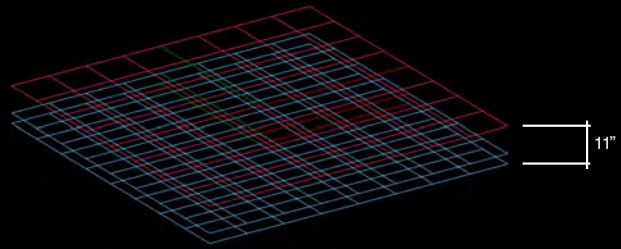
Levels

Level Planes (TOFF + TOS + ZFA Planes)



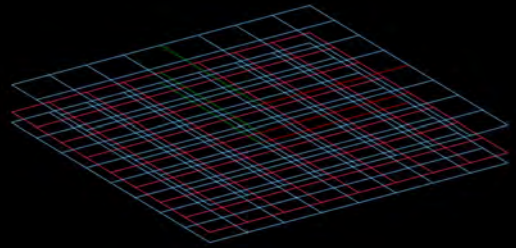
ZFA plane

ZFA level created by an 11” vertical offset from the TOS plane



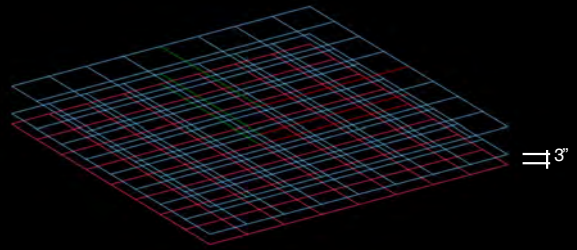
TOFF plane

As defined in architectural Revit model

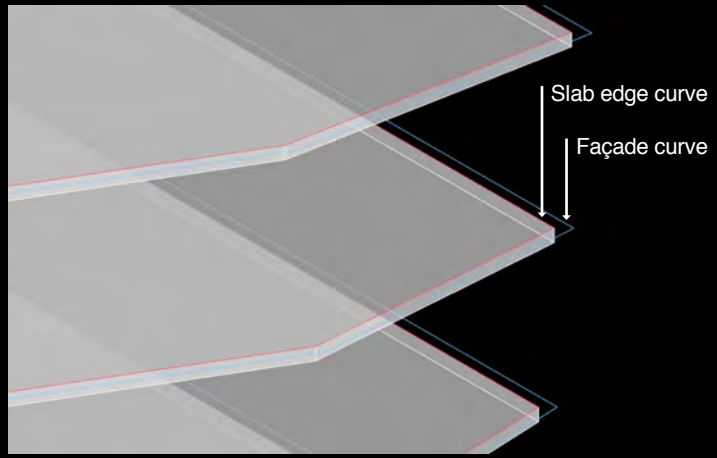


TOS plane

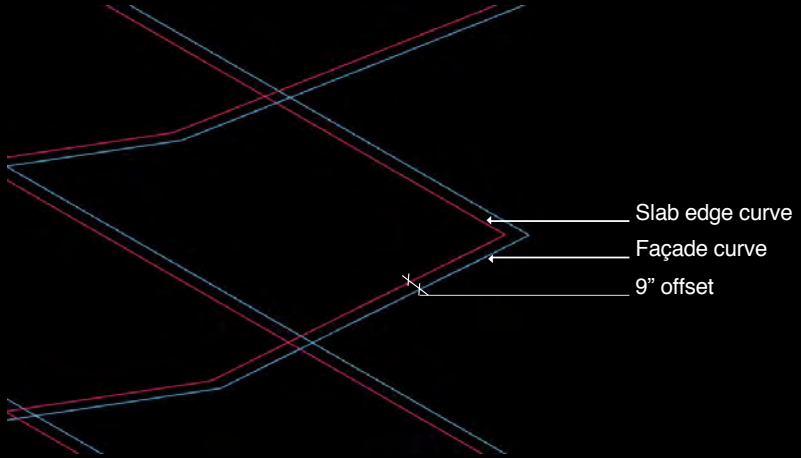
Top of Slab level created by offsetting the TOFF by the finish thickness (3”) in the negative direction



Levels
Slab Edge Curves



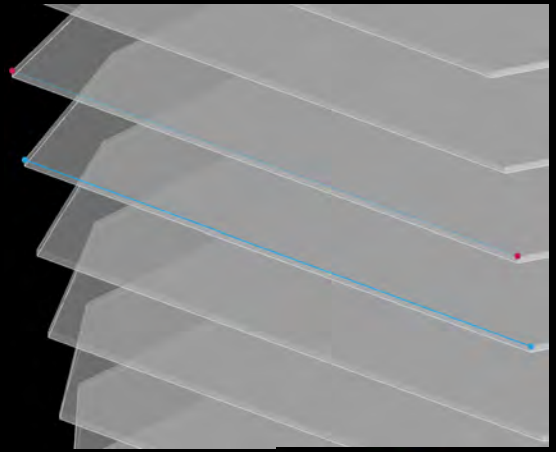
Slab outline curves generated by offsetting the façade curve by the ‘slab offset from exterior wall’



Slab edge curve inset by ‘slab offset from exterior wall’ which = 9”

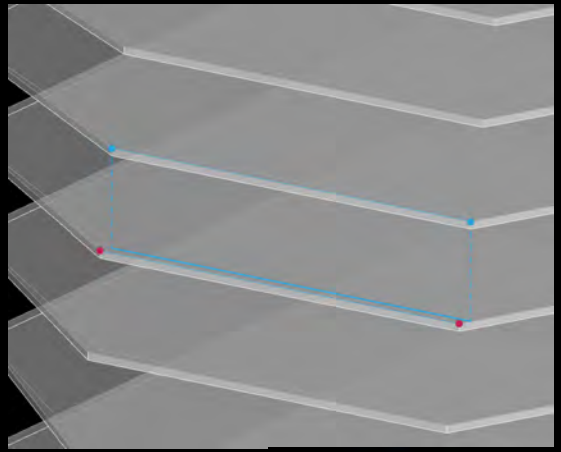
Levels

Window Wall Curves



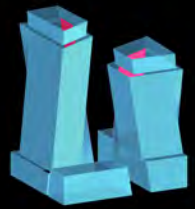
Vertical & Cantilevered facades

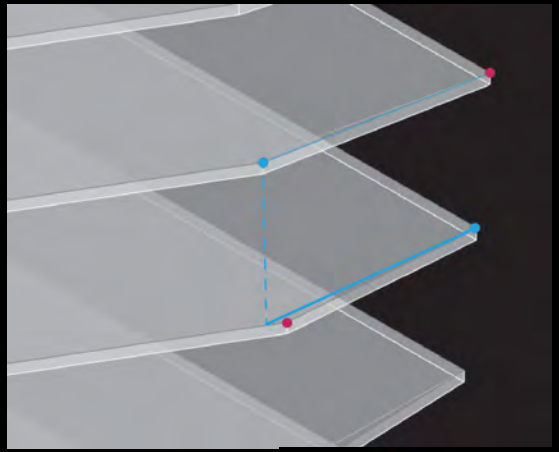
The window wall curve is derived by using the slab edge points at the level of the Window Wall curve



Sloped facades

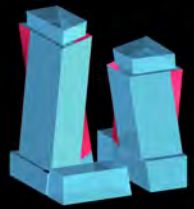
The window wall curve is derived from the slab edge curve from the floor above and projected down to the appropriate level





Ruled facades

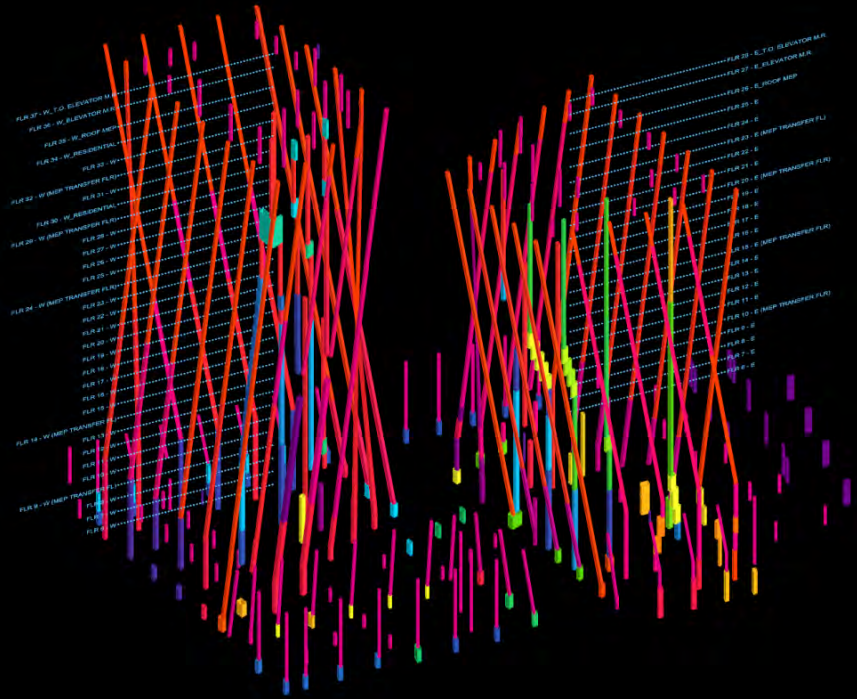
Window wall curve created by evaluating 2 slab edge curves (current floor SE curb and the above floor SE curve) - the four points of these

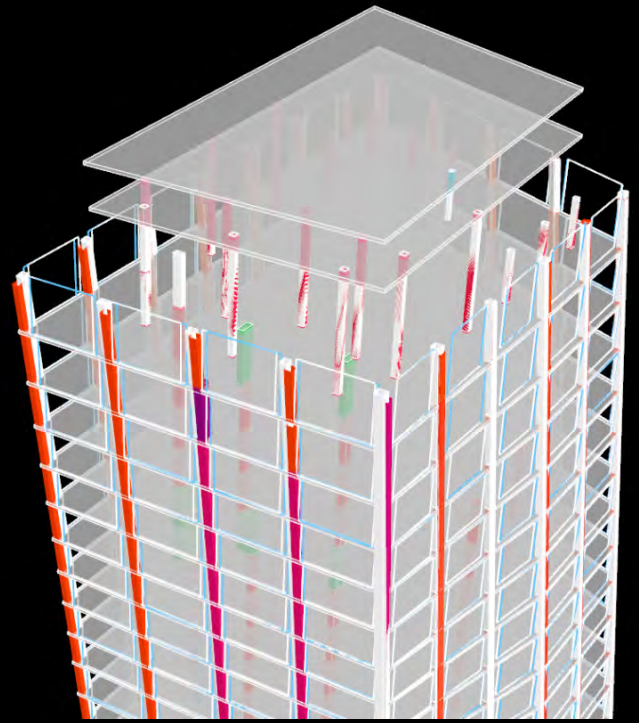


Rhino | Grasshopper | Flux | Dynamo | Revit

Structure & Facades

The Unstructured Data Problem

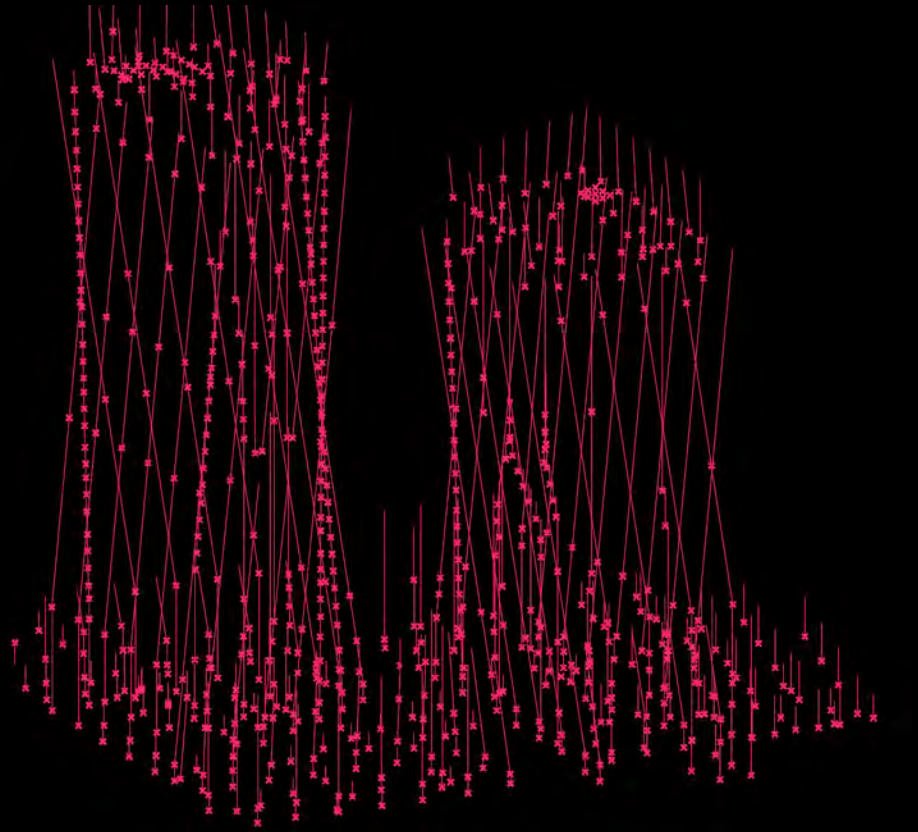




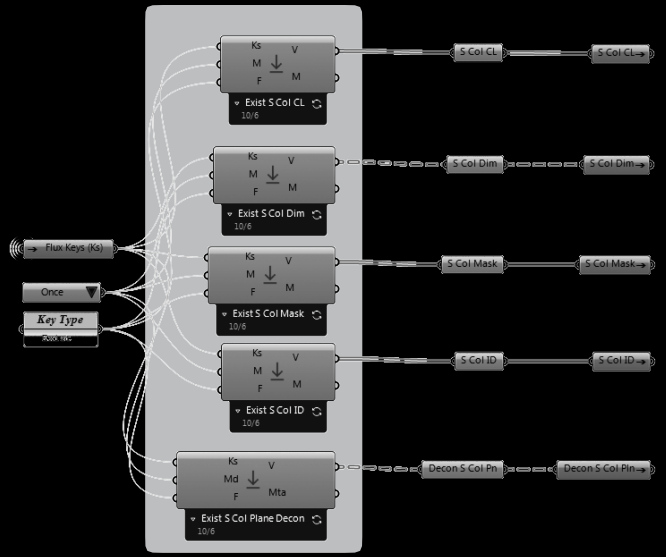
Rhino | Grasshopper | Flux | Dynamo | Revit

Structure

Receive Structural Data from Revit to Grasshopper via Dynamo



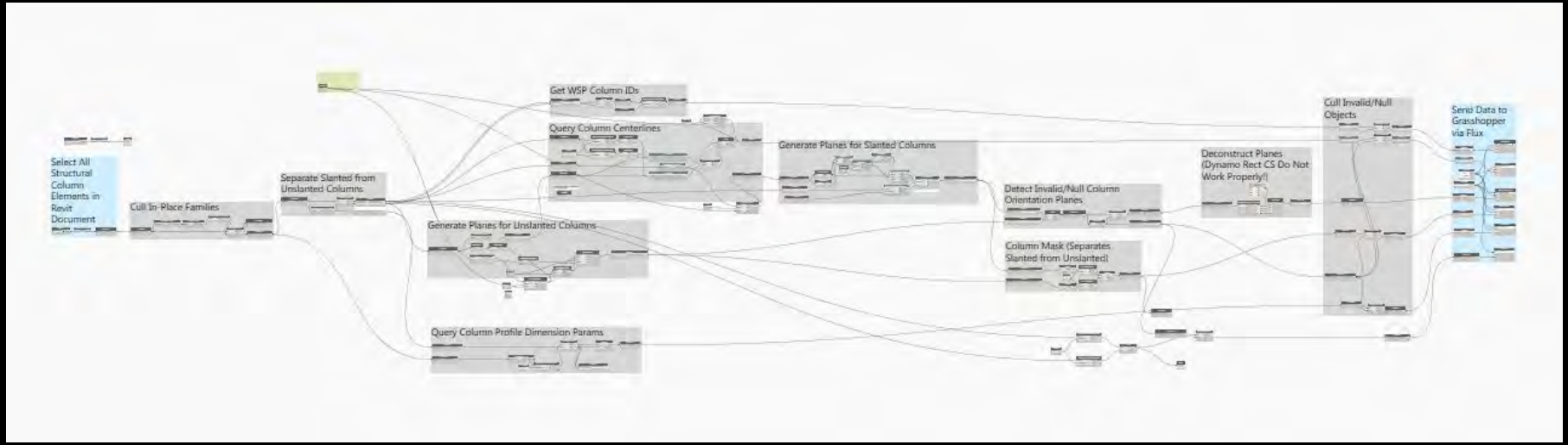
Data received from Flux



Rhino | Grasshopper | Flux | Dynamo | Revit

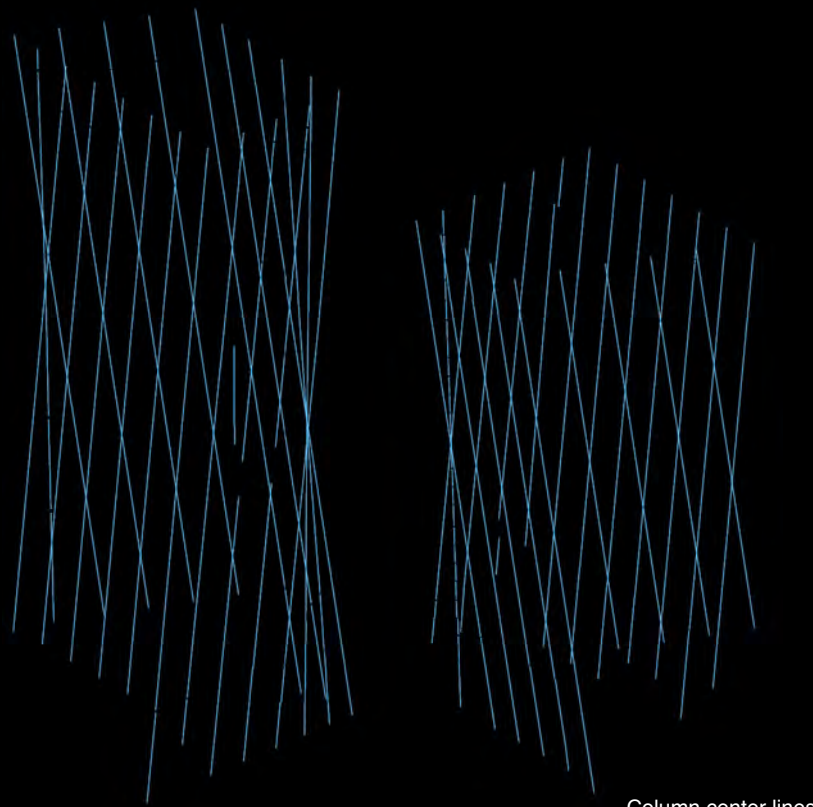
Structure

Grasshopper definition to receive Flux structural data

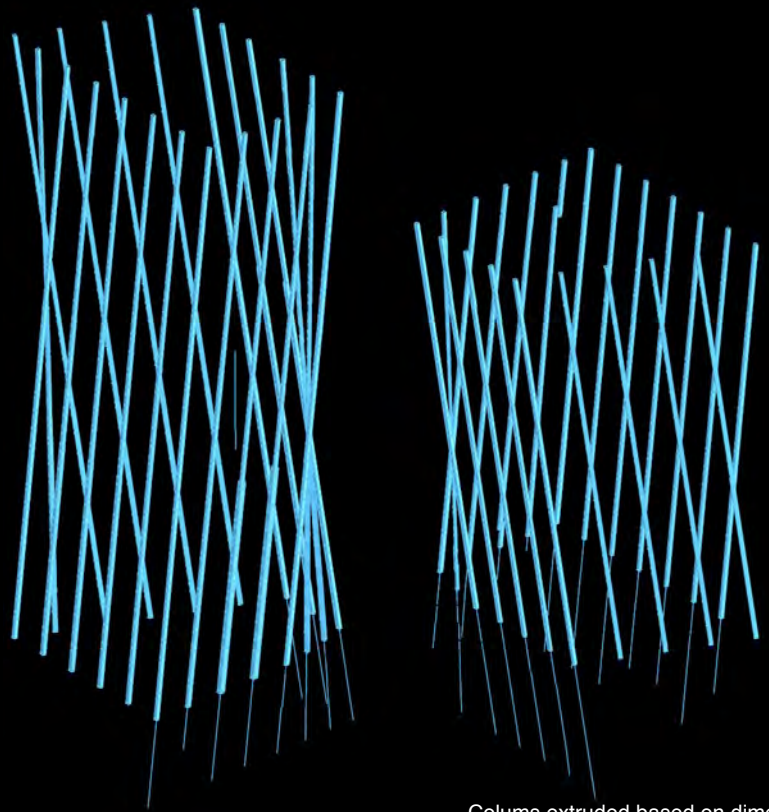


Structure

organize structure data relative to Grasshopper data



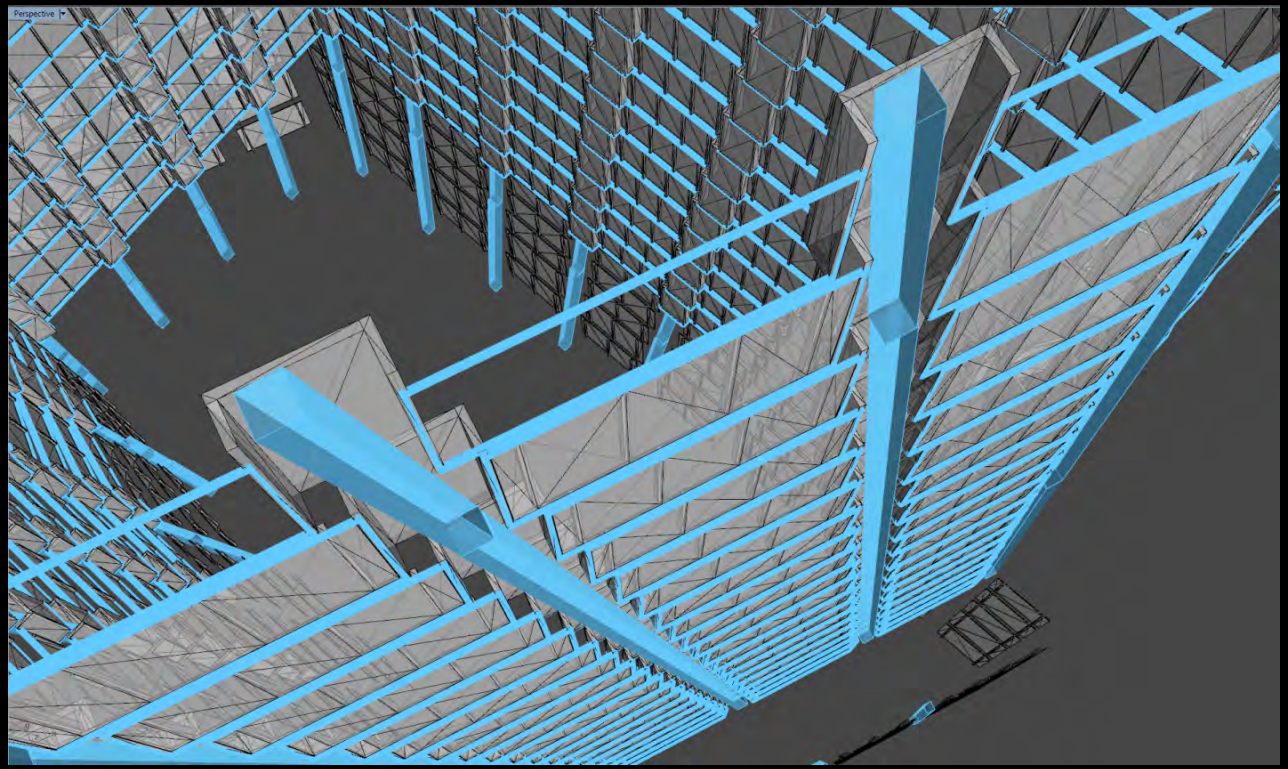
Column center lines



Colums extruded based on dimensions

Rhino | Grasshopper | Flux | Dynamo | Revit

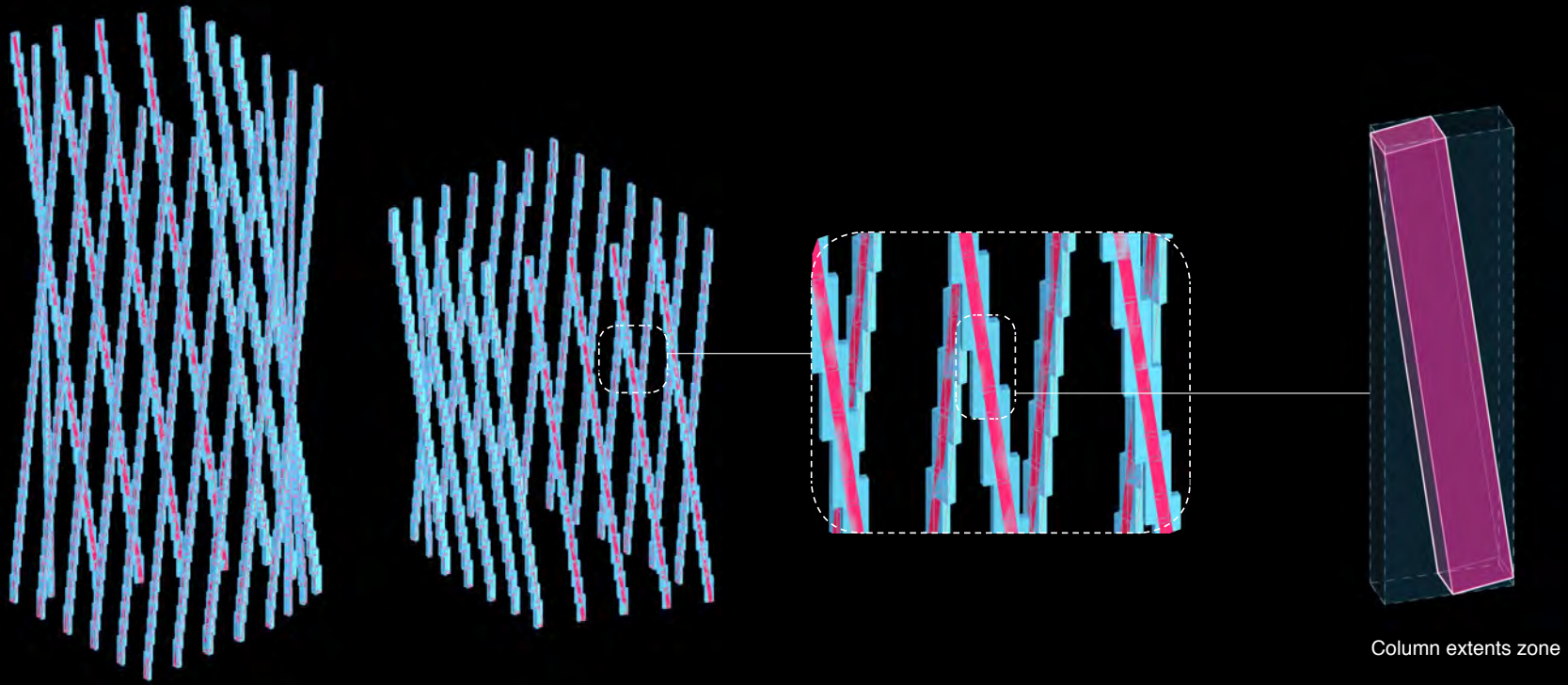
Structure
column clear zones



Rhino | Grasshopper | Flux | Dynamo | Revit

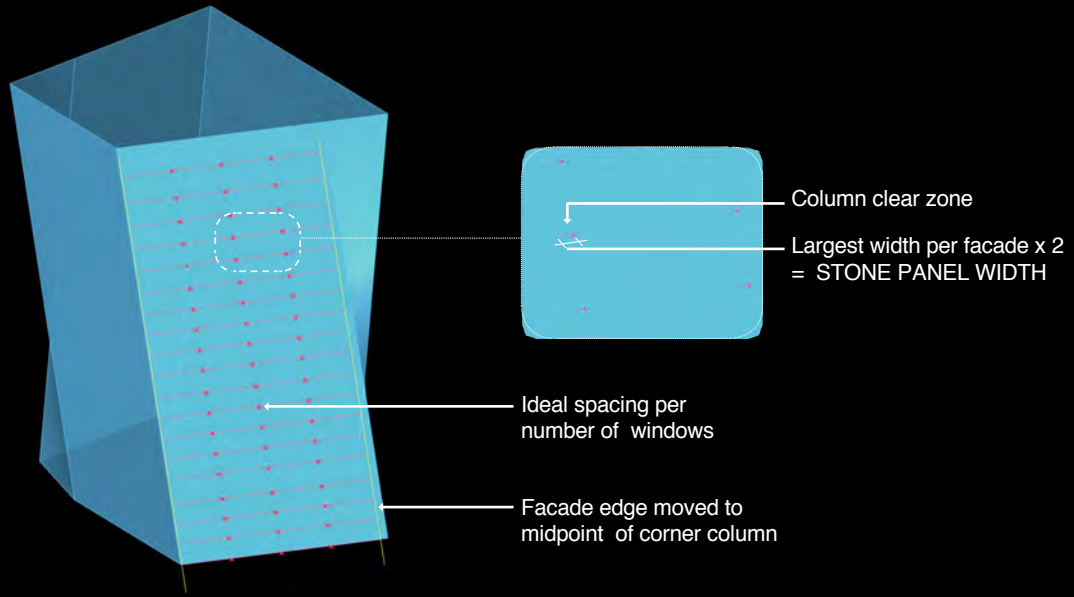
Structure

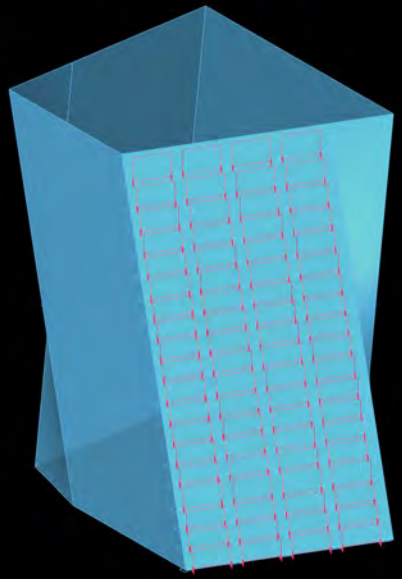
column clear zones + extents profile



Windows/Column Gyp

regular facade windows

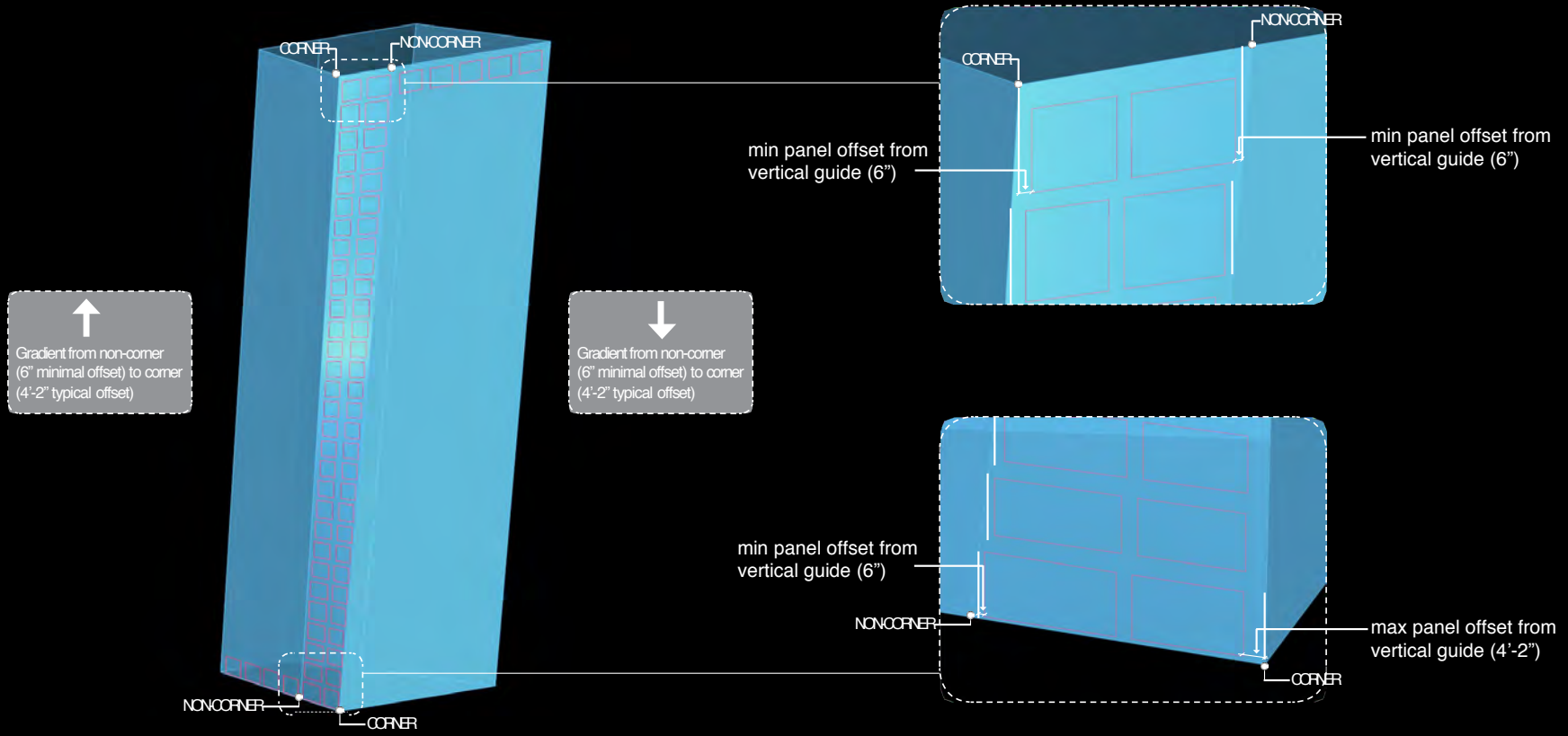




Rhino | Grasshopper | Flux | Dynamo | Revit

Windows/Column Gyp

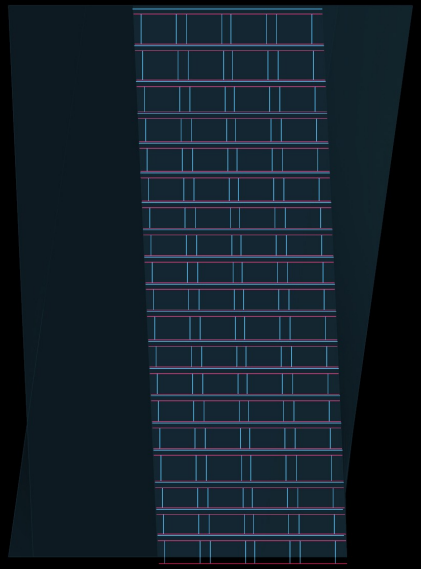
ruled facade windows

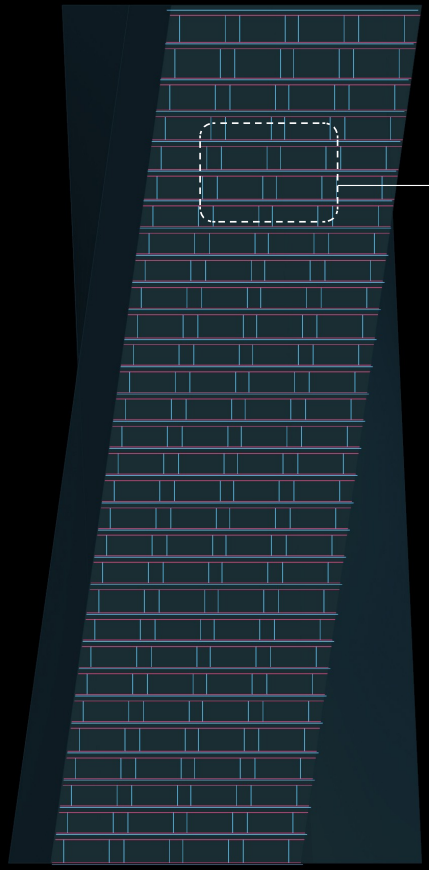


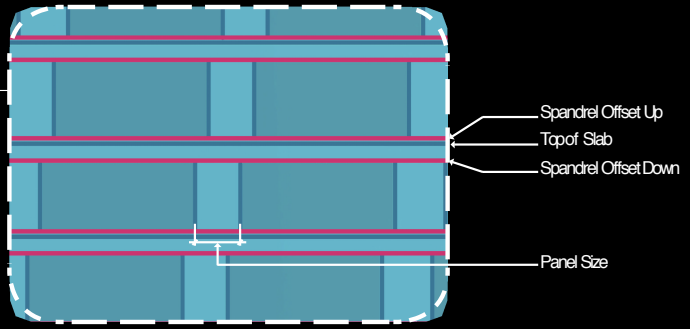
Rhino | Grasshopper | Flux | Dynamo | Revit

Windows/Column Gyp

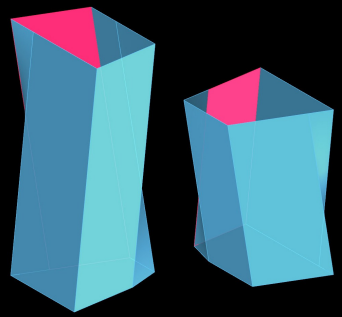
window placement







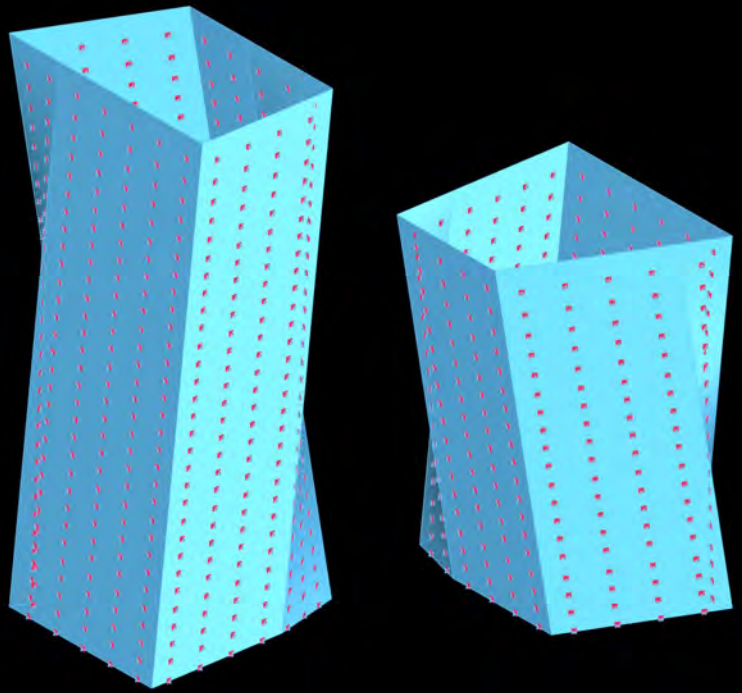
Window outline is derived by the horizontal spandrels curves and vertical edge of stone panels.



Rhino | Grasshopper | Flux | Dynamo | Revit

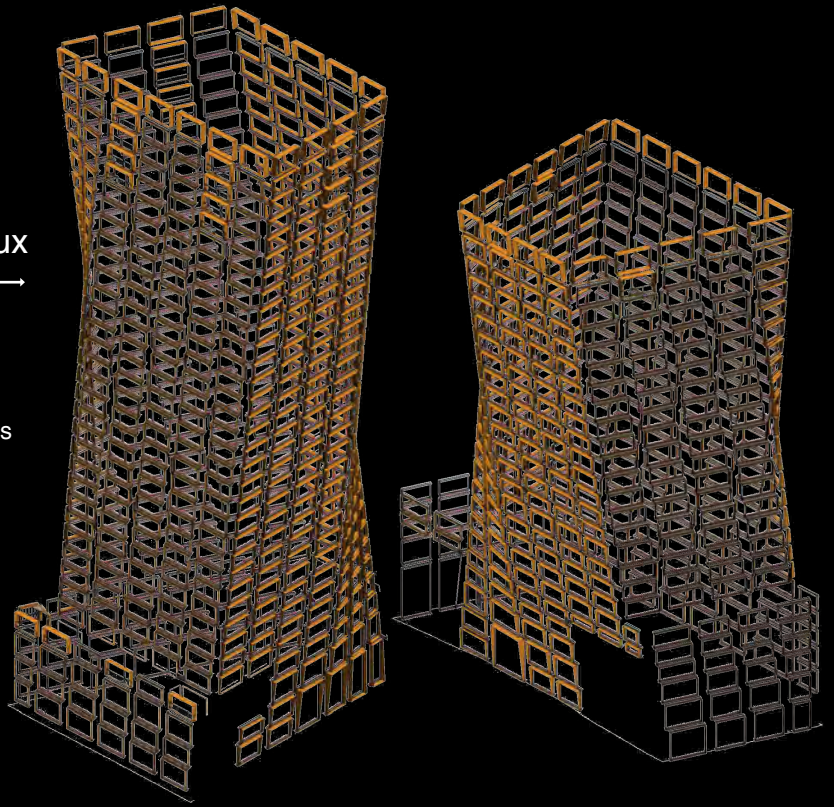
Windows/Column Gyp

overall window placement



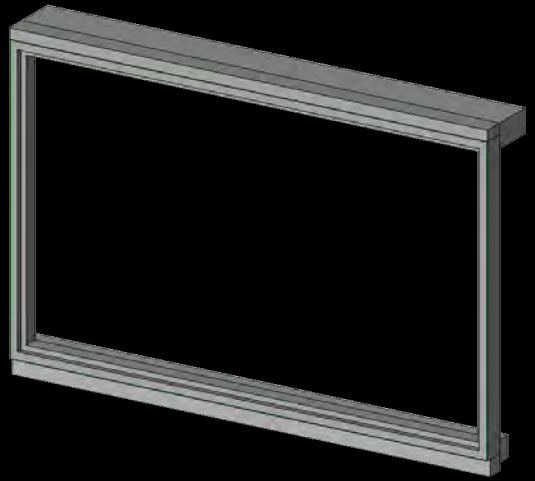
Data sent to Flux

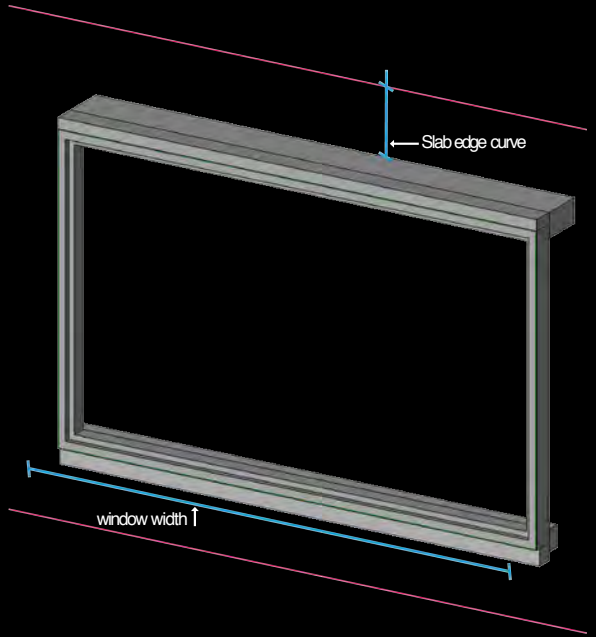
- window positions
- window rotation
- floor to floor heights
- window width
- window corners



Windows/Column Gyp

overall window placement





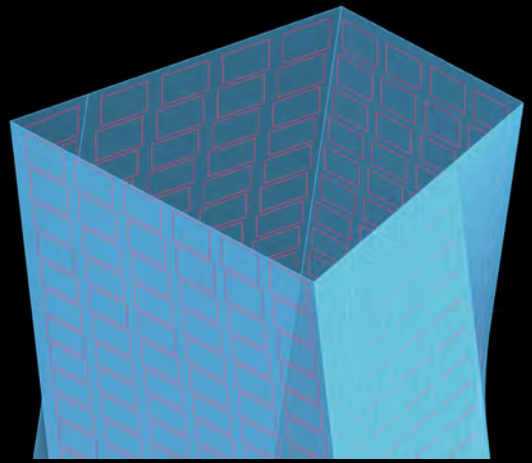
Using data pulled from Flux, Revit window families are placed on the correct point + level, and in the correct orientation (via Dynamo).

Parameters are added to Revit families for remaining data pulled in from Flux (ie. window width and floor to floor height)

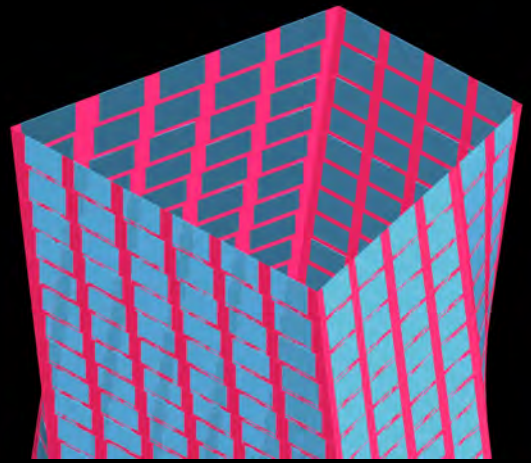
Rhino | Grasshopper | Flux | Dynamo | Revit

Cladding/Panels

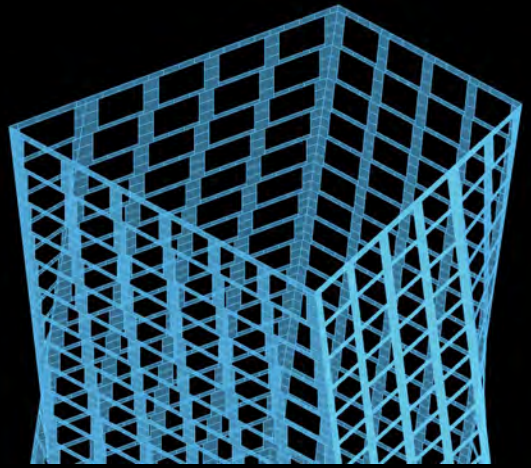
Stone Panel Logic



Crown surfaces are split horizontally.
Vertical edge curves are arrayed horizontally based on number of crown panels.



Stepping of panels follows angle of corner vertical edge curve.



Panel corners extracted to push into Revit

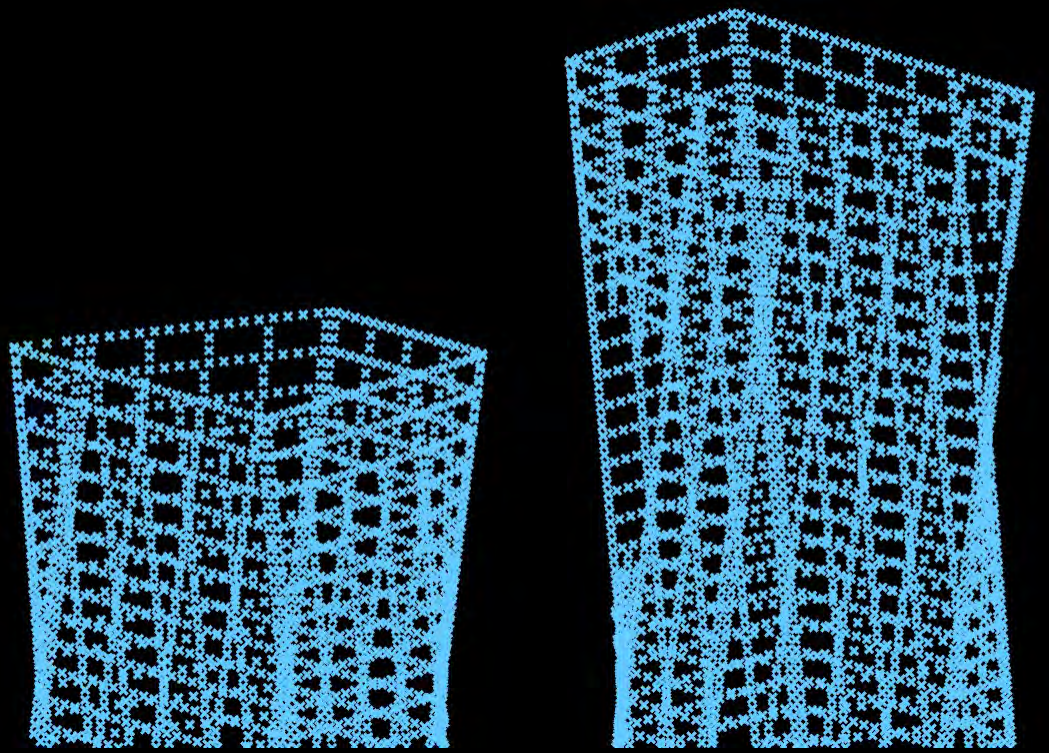
Rhino | Grasshopper | Flux | Dynamo | Revit

Cladding/Panels

Stone Panel Placement into Revit/Revit Families via Flux

Data sent to Flux

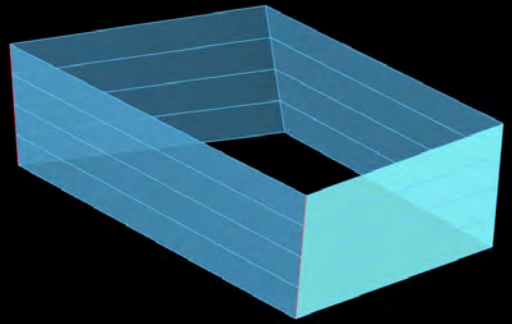
- placement point
- panel rotation
- panel z height
- panel y depth
- bottom left/right dim
- top left/right dim



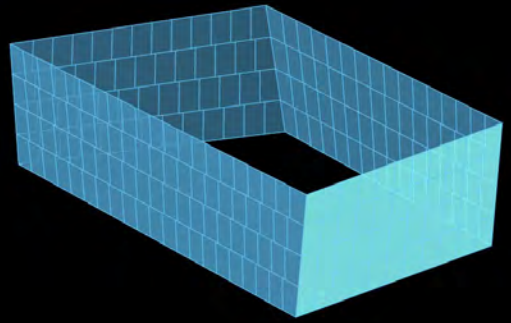
Rhino | Grasshopper | Flux | Dynamo | Revit

Cladding/Panels

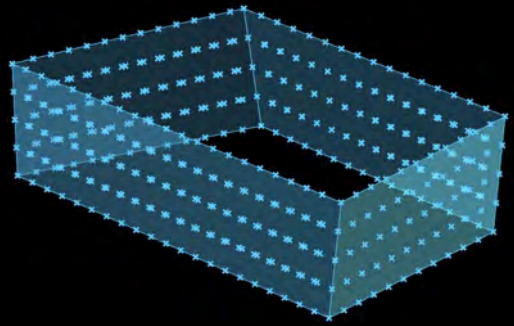
Crown Panel Logic



Crown surfaces are split horizontally.
Vertical edge curves are arrayed horizontally based on number of crown panels.



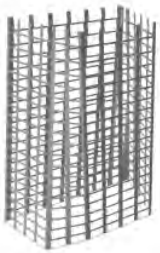
Stepping of panels follows angle of corner vertical edge curve.



Panel corners extracted to push into Revit

Cladding/Panels

Revit and the Need for Speed



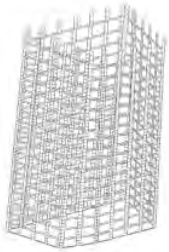
Define minimum viable geometry in Grasshopper

Create Revit adaptive component (<1 minute)

Send Grasshopper points to Flux

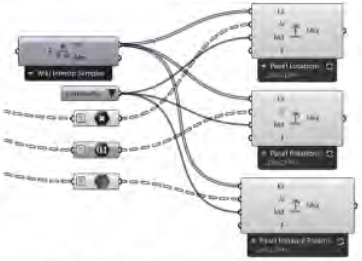
Receive points from Flux in Dynamo

Place adaptive components in Revit (2700 elements in 4 minutes)

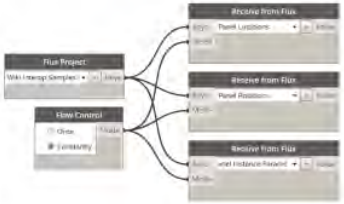


Placed elements in Revit

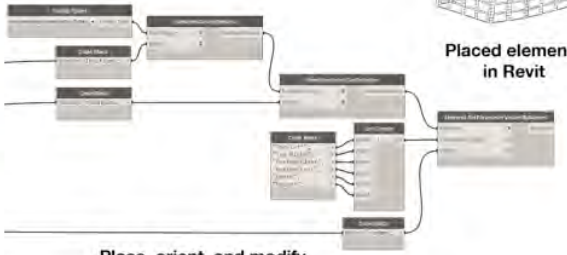
Create Revit generic family with instance parameters (2-4 hours)



Send Grasshopper locations, orientations, and param data to Flux



Receive data from Flux in Dynamo

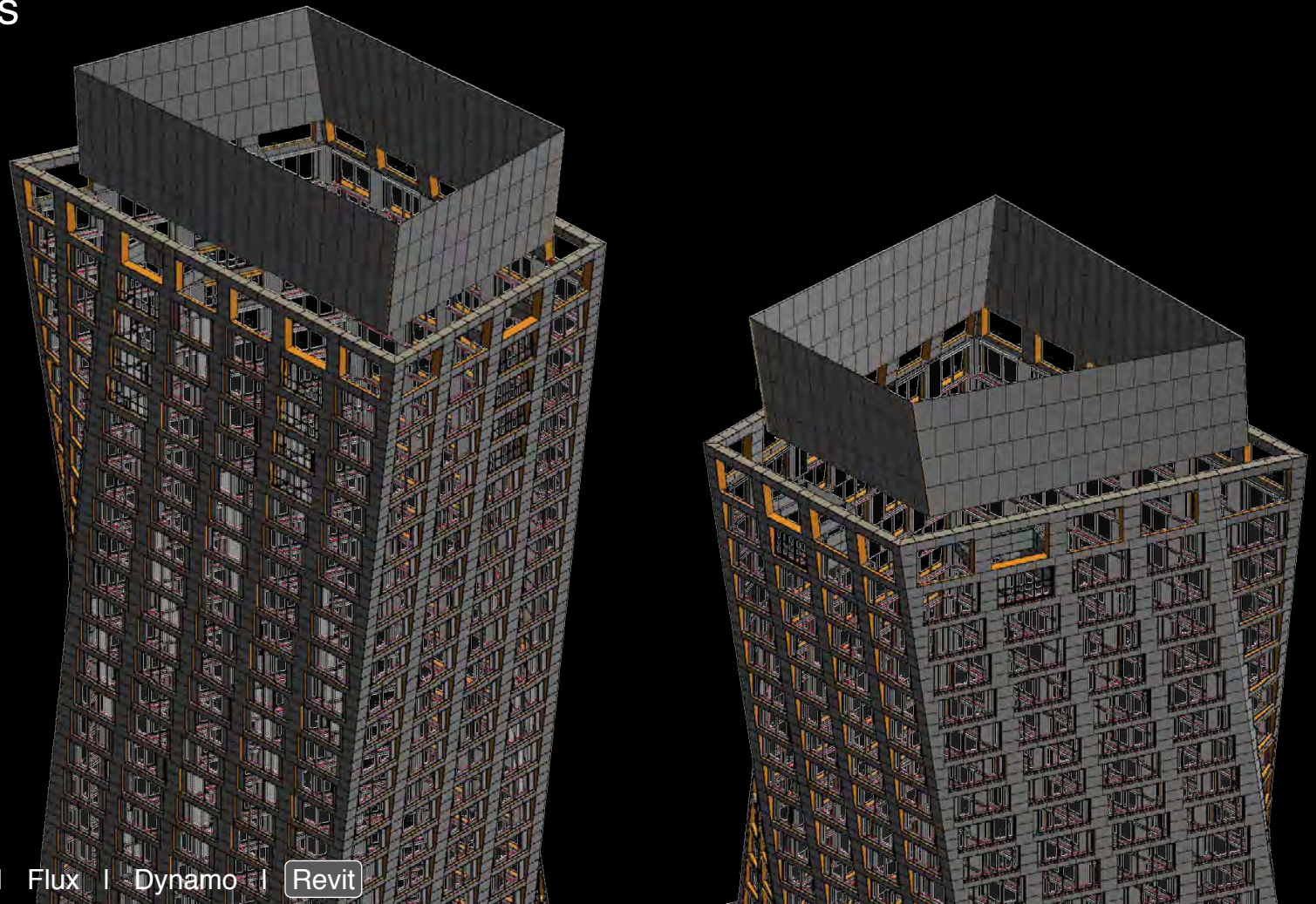


Place, orient, and modify generic family instances in Revit (2700 elements in < 15 seconds)

Rhino | Grasshopper | Flux | Dynamo | Revit

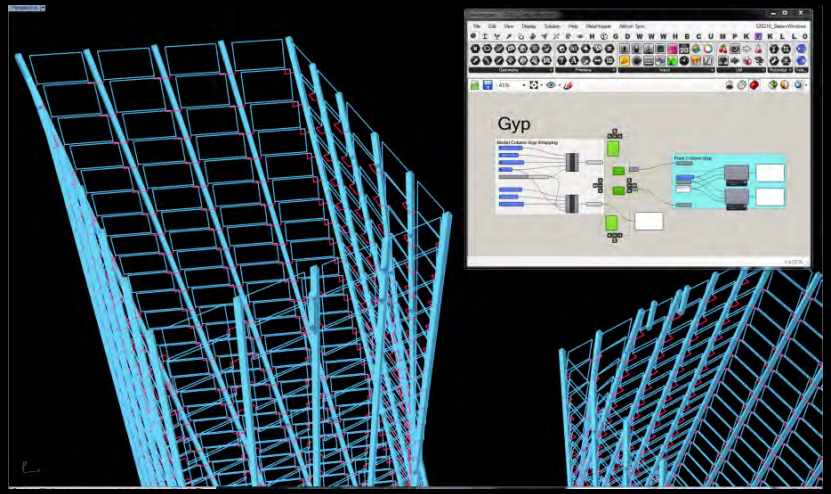
Cladding/Panels

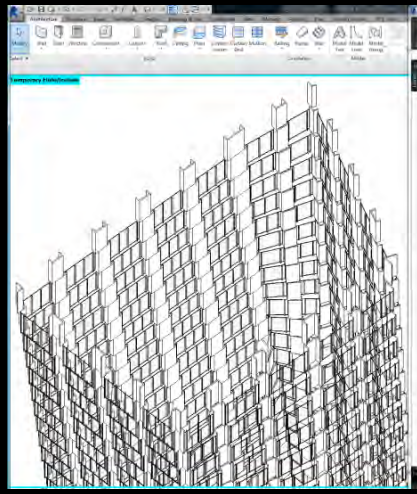
Global vs Local

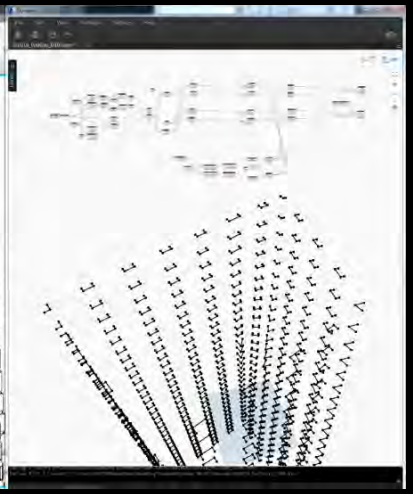


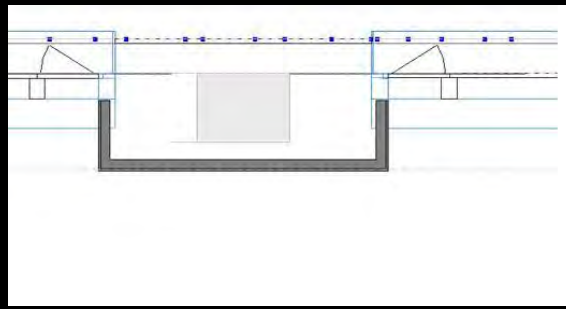
Rhino | Grasshopper | Flux | Dynamo | Revit

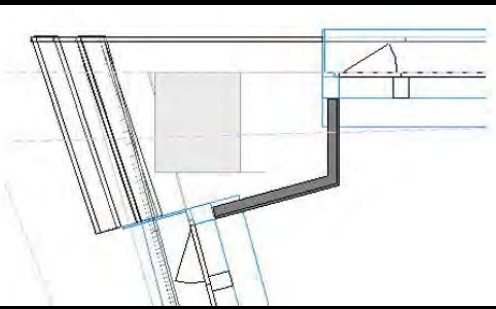
Parametric Drywall

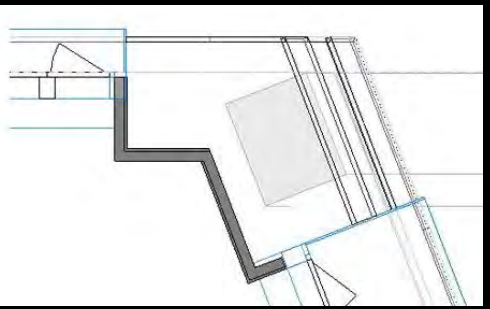










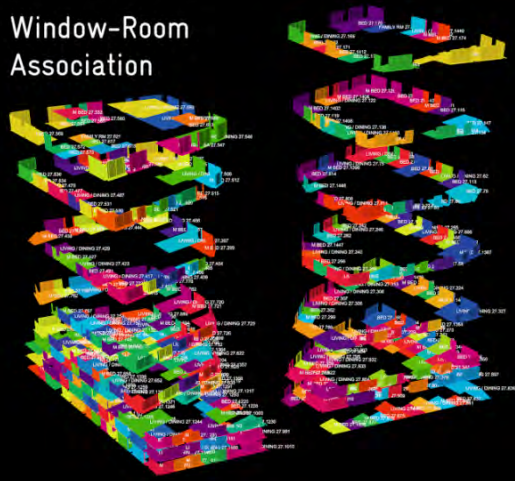


Rhino | Grasshopper | Flux | Dynamo | Revit

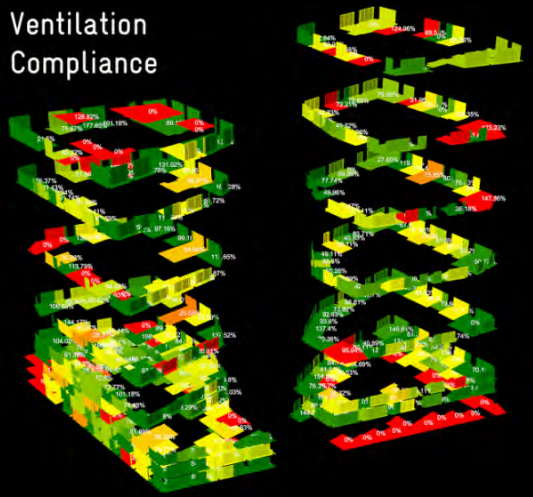
Analysis

Other examples of interop approach

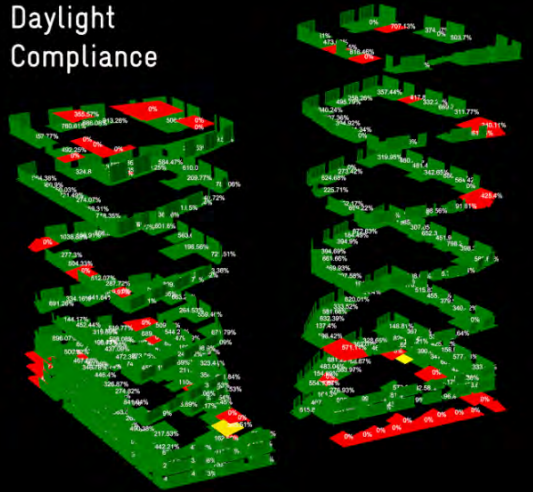
Window-Room Association



Ventilation Compliance



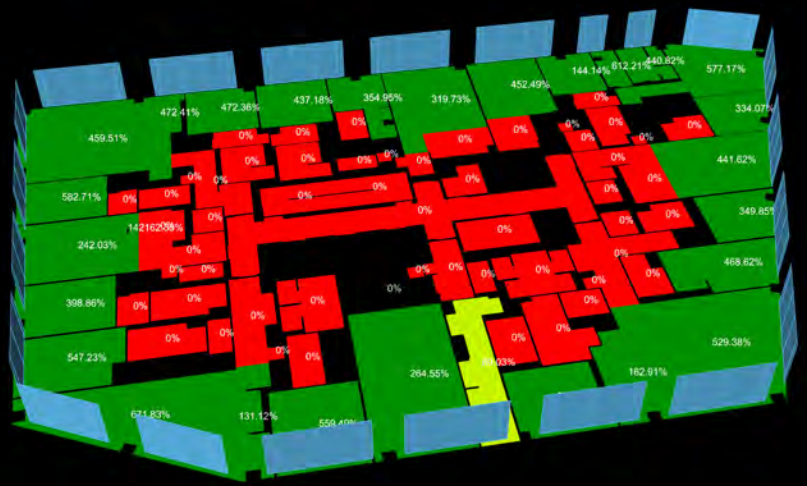
Daylight Compliance

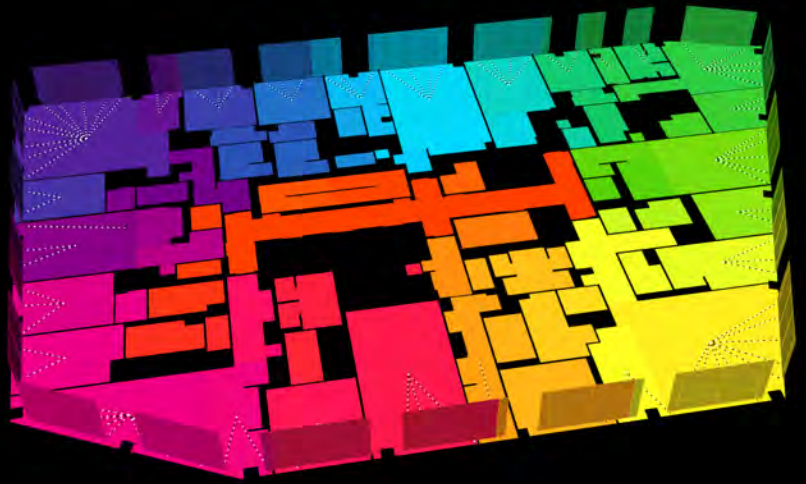


Rhino | Grasshopper | Flux | Dynamo | Revit

Analysis

Other examples of interop approach

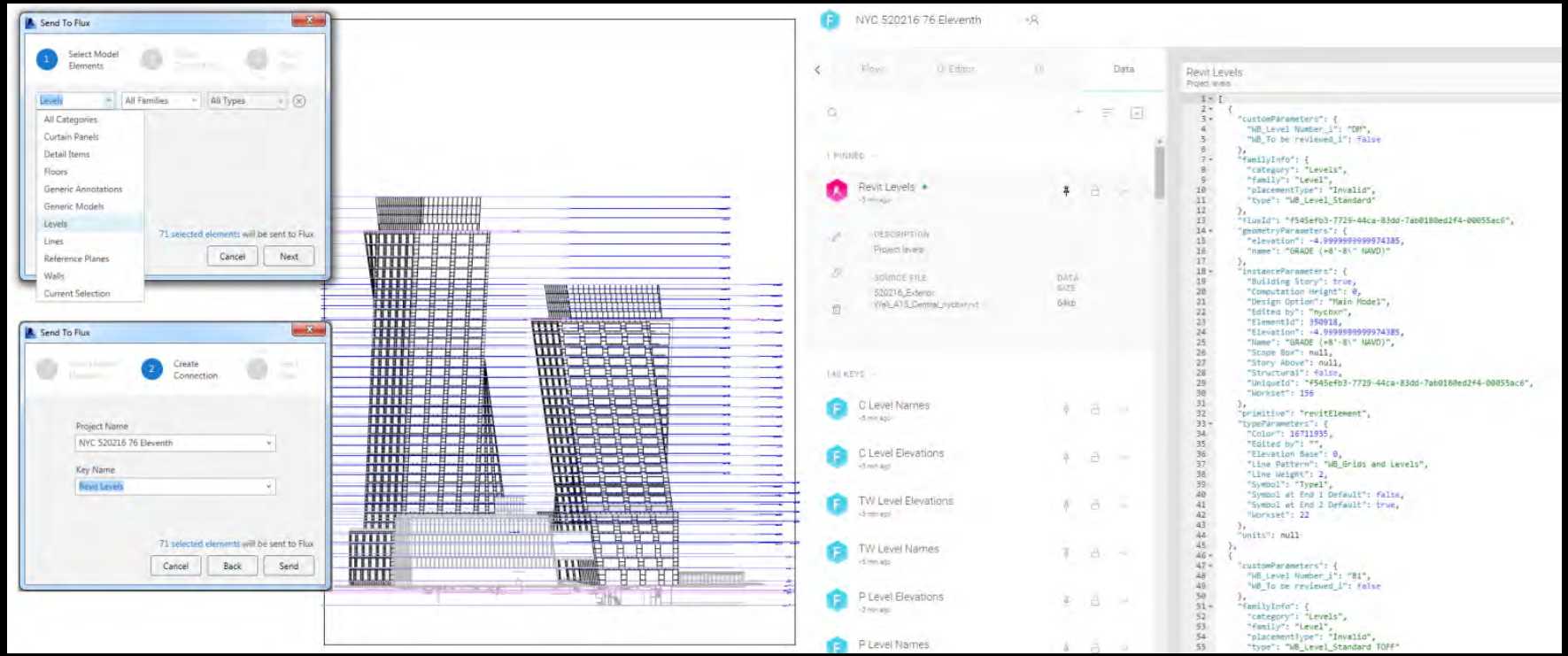




Rhino | Grasshopper | Flux | Dynamo | Revit

Data Formats

JSON



Rhino | Grasshopper | Flux | Dynamo | Revit

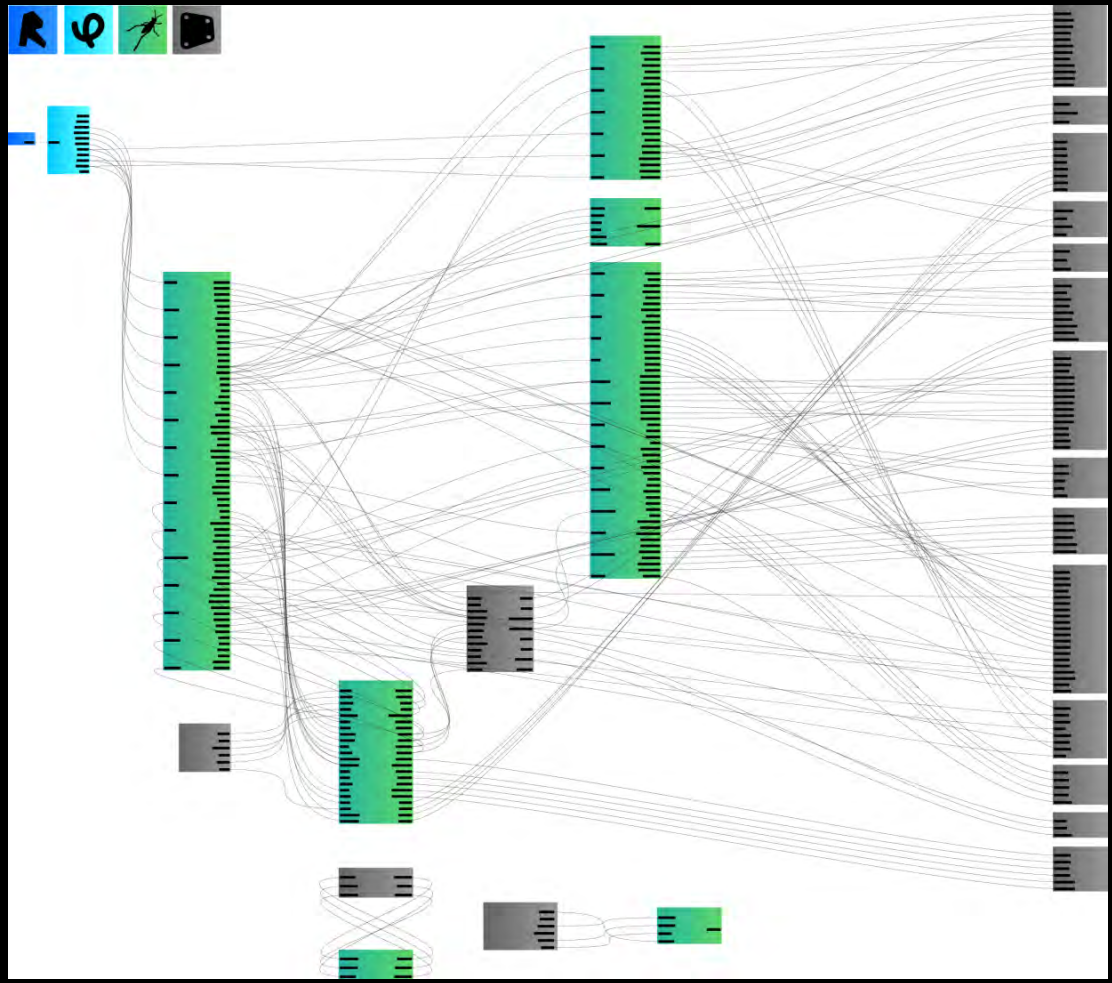
Data Formats

JSON


```
{
  "familyInfo": {
    "Category": "Levels",
    "Family": "Level",
    "Type": "WB_Level_Standard"
  },
  "geometryParameters": {
    "Elevation": 54.99999999999699,
    "Name": "FLR 5 - PODIUM EAST"
  },
  "instanceParameters": {
    "Building Story": true,
    "Computation Height": 0,
    "Edited by": "",
    "ElementId": 1013505,
    "Elevation": 54.99999999999699,
    "Name": "FLR 5 - PODIUM EAST",
    "Scope Box": null,
    "Story Above": null,
    "Structural": false,
    "UniqueId": "608634be-46b3-4622-b1e0-1ba0
    "Workset": 0
  },
  "primitive": "revitElement",
  "typeParameters": {
    "Color": 16711935,
    "Edited by": "",
    "Elevation Base": 0,
    "Line Pattern": "WB_Grids and Levels",
    "Line Weight": 2,
    "Symbol": "Type1",
    "Symbol at End 1 Default": false,
    "Symbol at End 2 Default": true,
    "Workset": 22
  }
```

The Metagraph

Interop Overview



Rhino | Grasshopper | Flux | Dynamo | Revit

Design to Fabrication

The Potential

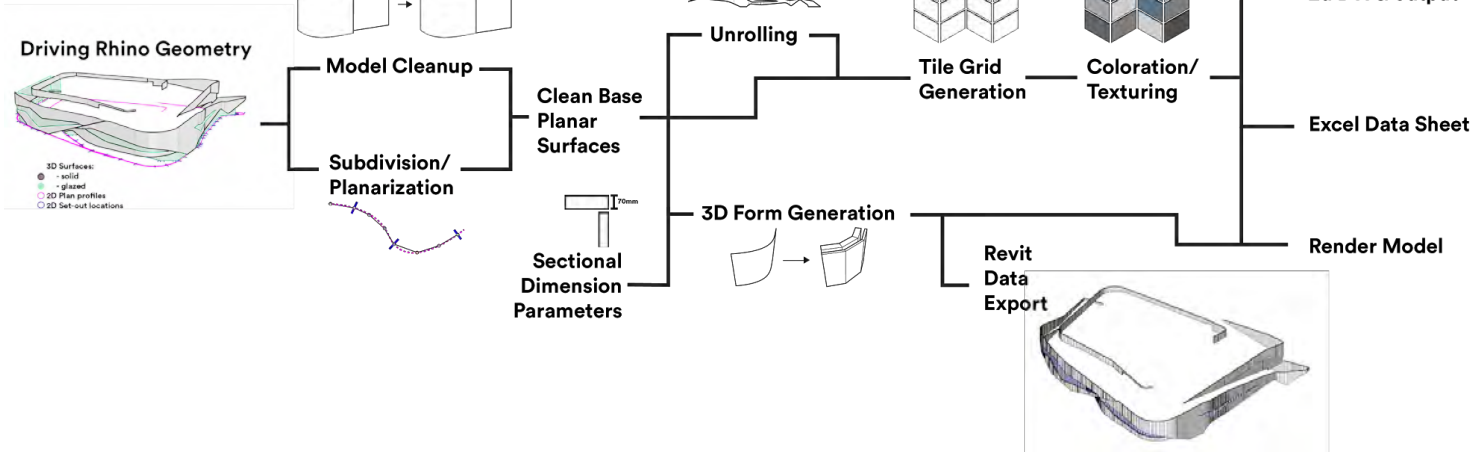
Rhino | Grasshopper | Flux | Dynamo | Revit

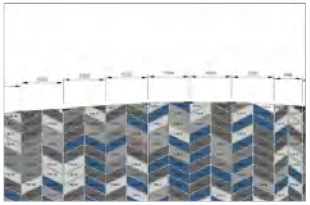
Current Status

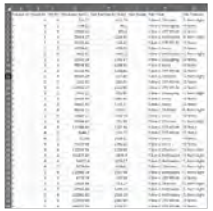
On site

Designing the Design Workflow

Incorporating volatility and ”obsolete” deliverables







Designing the Design Workflow

One size does not fit all

Workflow Considerations

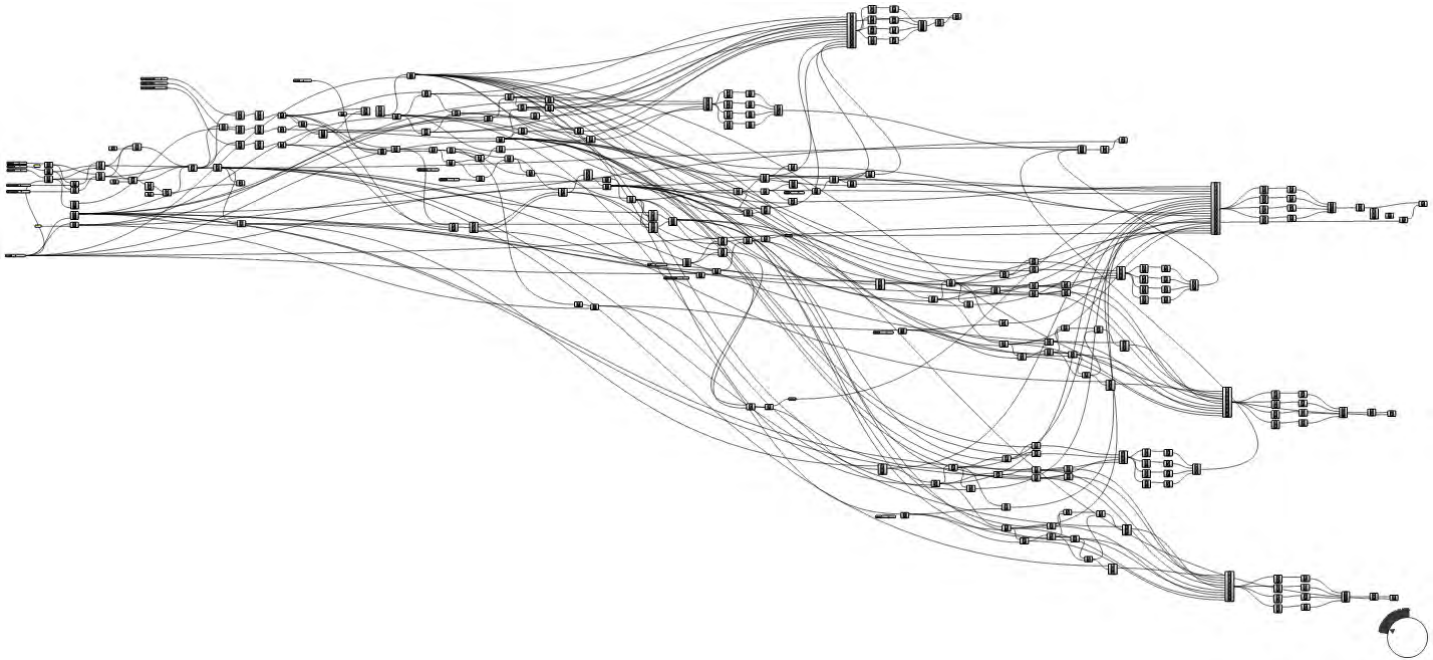
1. Contractual and deliverable requirements
2. Regional variation
3. Project team skill/preference
4. Phase(s)/workflow lifecycle
5. Stakeholder interest
6. Software utilized
7. Geometric considerations
8. How the team wants to communicate (Slack!)
9. …

Workflow Determinations

1. Use Grasshopper as much as possible in order to quickly solve complex geometry and leverage the interrelationships encapsulated within a single graph.
2. Minimize the use of Dynamo for complex geometry – it is slower and less agile compared to Grasshopper. Restrict it's use to where it shines: Revit model interaction for automated element placement and parameter value population.
3. Encapsulate all data interoperability logic within a single Flux project.
4. Define each aspect of the model once and only once.
5. Identify the minimal non-programmatic inputs required to drive the computational model.

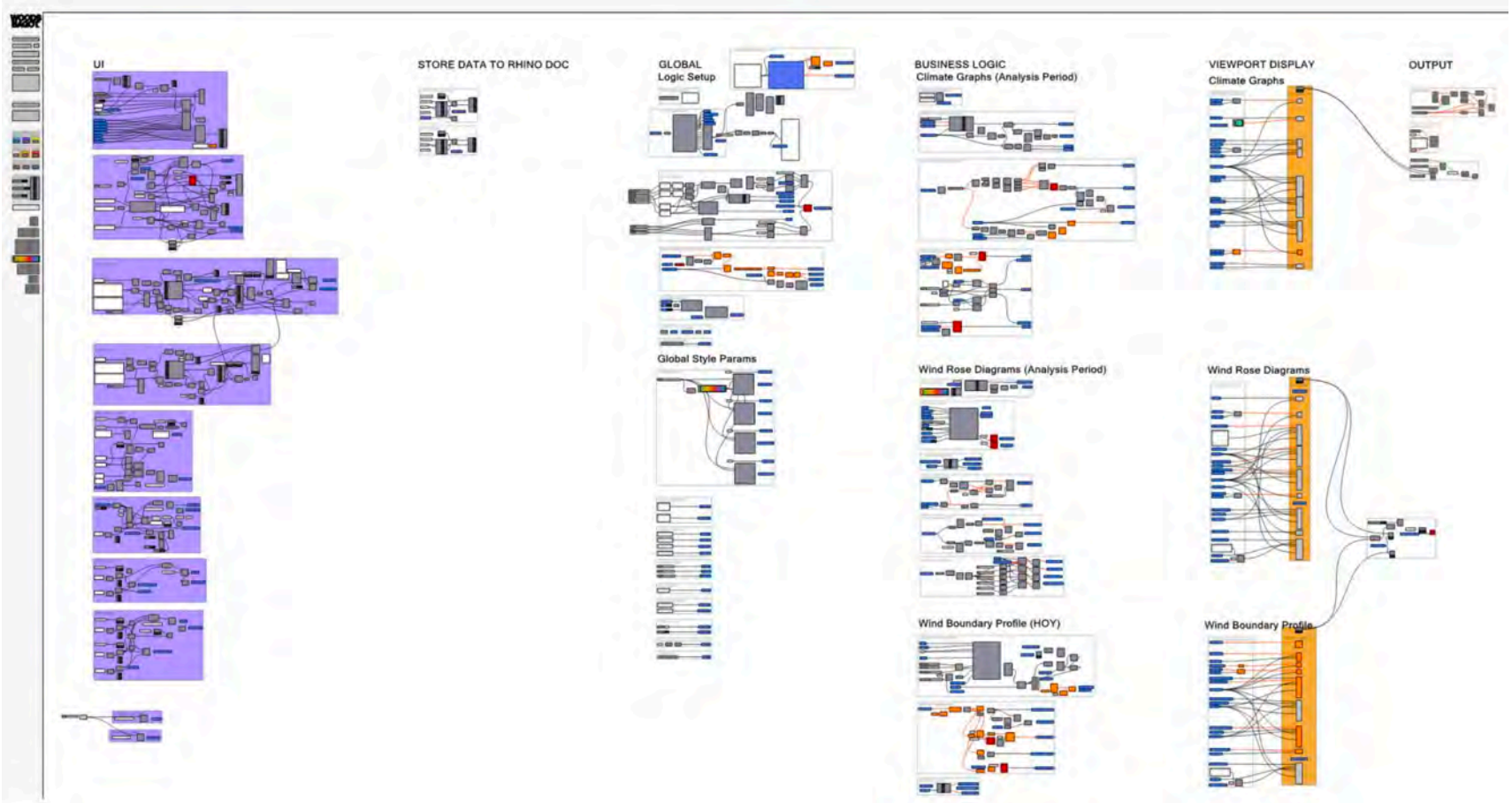
Working in Teams

Inheriting someone's logic



Working in Teams

Structuring for collaboration

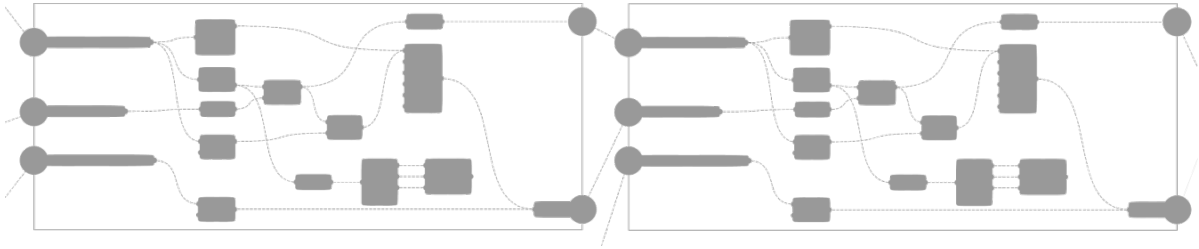


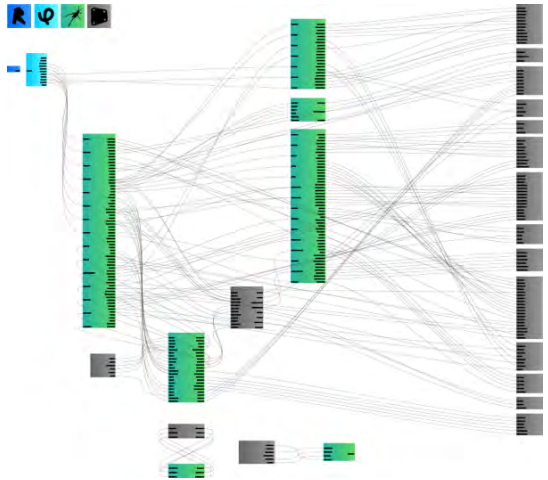
Working in Teams
Linked Graphs

Scalability

For reasons largely stemming from performance and intelligibility, it's unlikely (and certainly not recommended) to tackle all programmatic functionality with a single script (a "hydra" or "spaghetti monster").

Scalability can be solved not only by making visual programming environments more powerful, but also by lending situational intelligence to scripts so that they can be modular and interdependent.





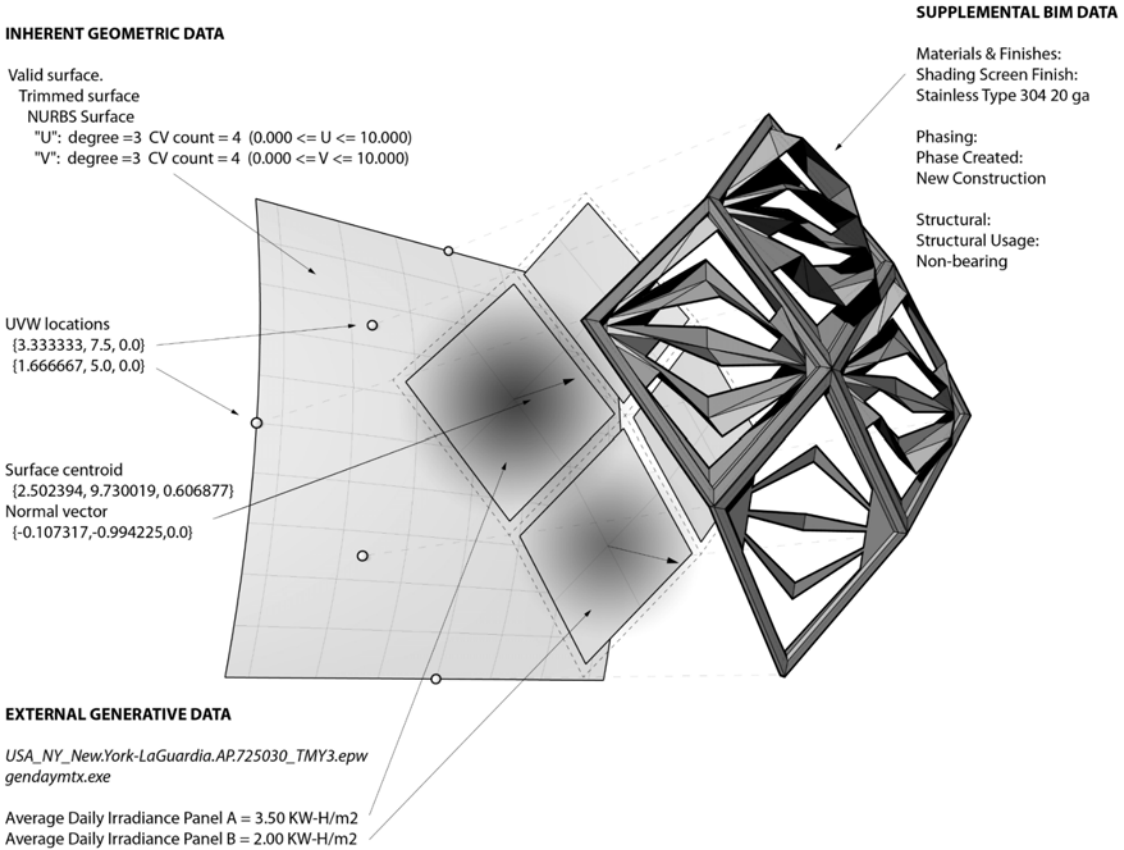
Where to place model intelligence

"Several users who have responded to this thread have honed in on one of the things Dynamo offers which separates it from applications like Grasshopper and GC. Through its integration with Revit, we like to say that Dynamo enables you to **choose where you want to put your intelligence**. For example, you might have an Adaptive Component family in **Revit that has incredibly complex internal relationships** that you've constructed and refined over many months or years. **This family has a large amount of embedded intelligence. But, it has limited situational intelligence**. That is, you place it next to another version of itself in a project and the two instances can't talk to each other, and they can't respond in any variable way to other drivers in the project. This is where you can add an additional layer of intelligence with Dynamo, using Dynamo to get parameters from one to set parameters on the other, or to set parameters on the instances based on some other value in the project. By comparison to GC or Grasshopper, you'd have to build all of this functionality in the graph, which is totally possible, albeit a bit unwieldy. On a more prosaic level, Dynamo solves the problem that making your geometry in an application that is not Revit, when Revit is where you are building the final deliverable, is a pain in the ass."

Ian Keough from dynamobim.org/forums/topic/we-dont-draw-spaghetti-buildings

Where to place model intelligence

(Most) Everything is Quantifiable



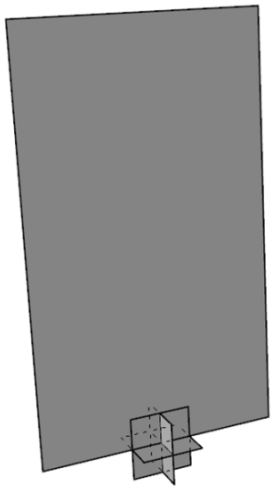
Where to place model intelligence

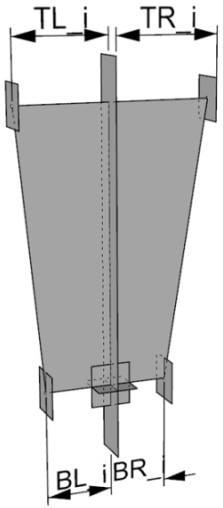
Interrelated Hierarchies of Intelligence

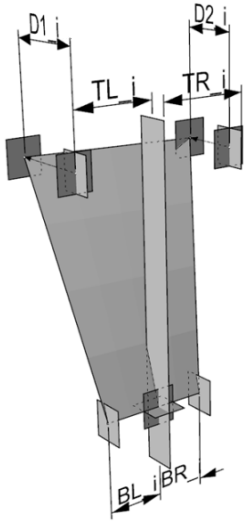
Object Intelligence

Object intelligence can be found in a parametric object capable of adapting holistically in response to changes in one or more of its parameter values.

A good example of object intelligence is a parametric family in Revit. The object itself is quite smart, but it has no awareness of its context, such as its relationship with neighboring elements.







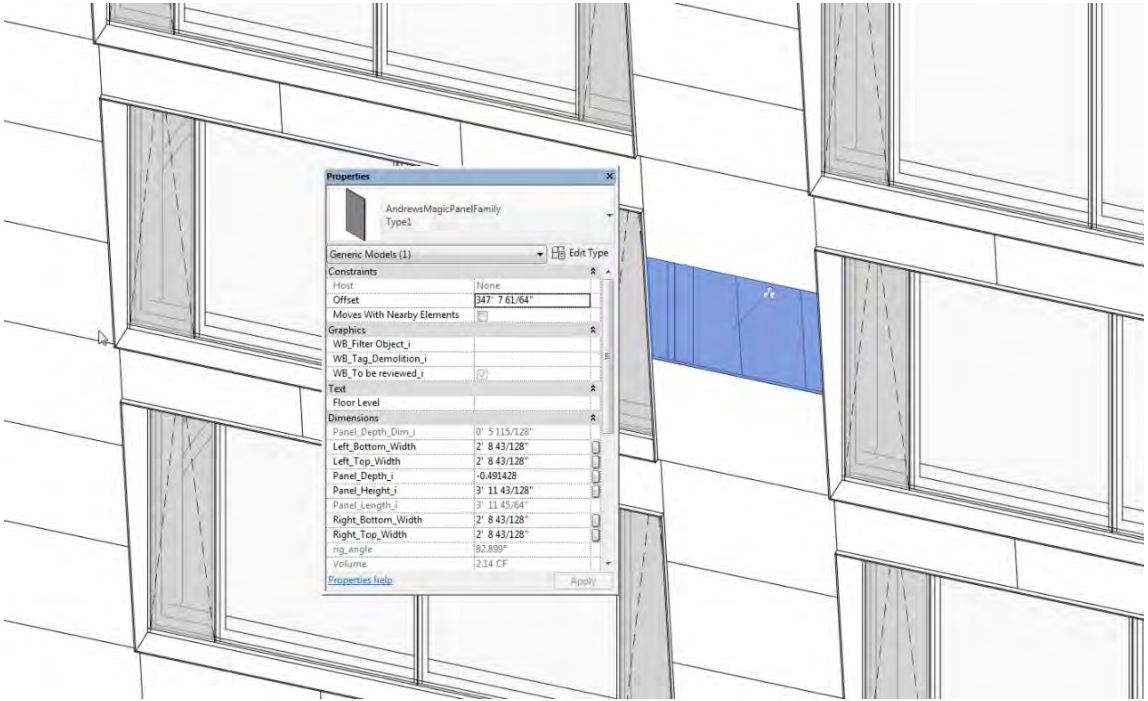
Where to place model intelligence

Interrelated Hierarchies of Intelligence

Situational Intelligence

Situational intelligence arises through the construction of inter-element relationships. Objects become aware of both themselves and other elements in the model.

Scripts are often used to induce situational intelligence on model elements by constructing relationships between their properties, such as positioning exterior wall panels based on the positions of windows.



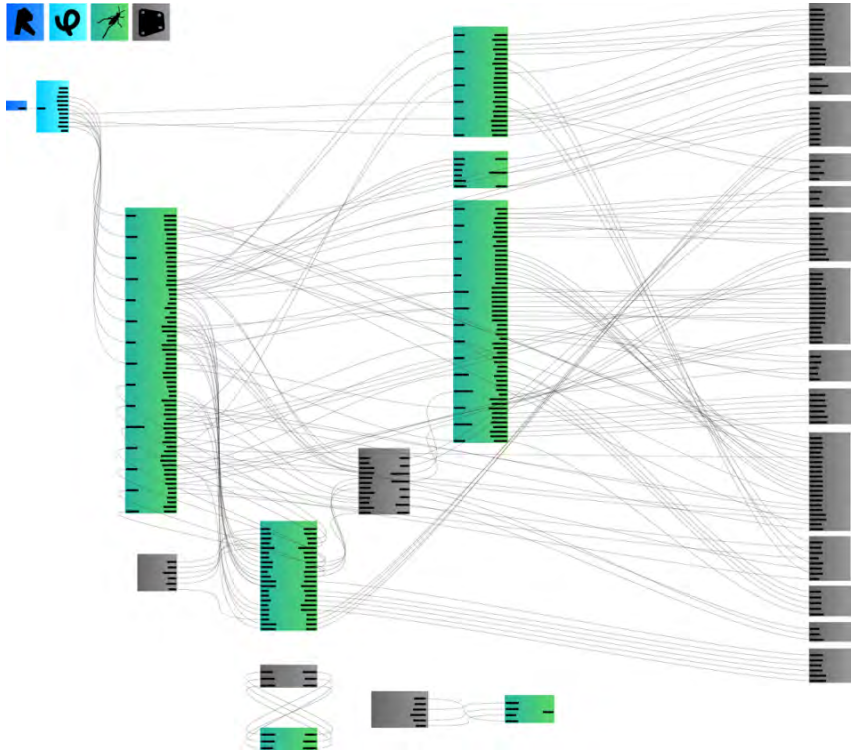
Where to place model intelligence

Interrelated Hierarchies of Intelligence

Systemic Intelligence

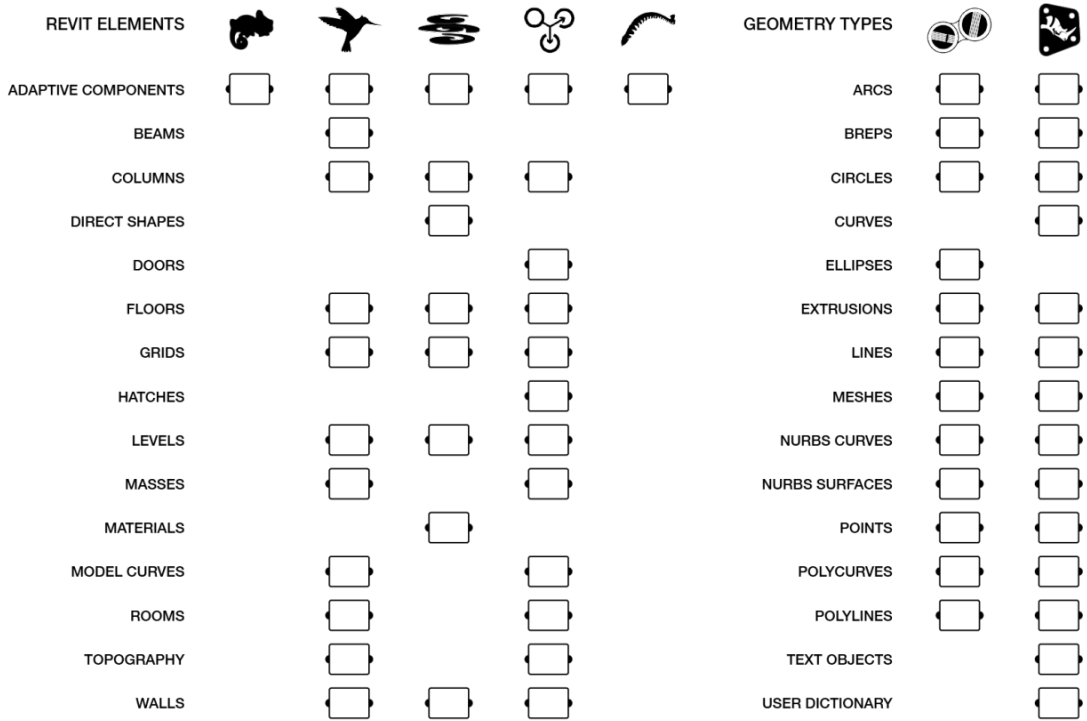
Systemic intelligence goes beyond the interrelatedness of elements in a singular model context to the relationships of interdependent models themselves, and any scripts that operate within the model contexts.

There is currently no solution for driving system-level model relationships, especially not multi-platform relationships



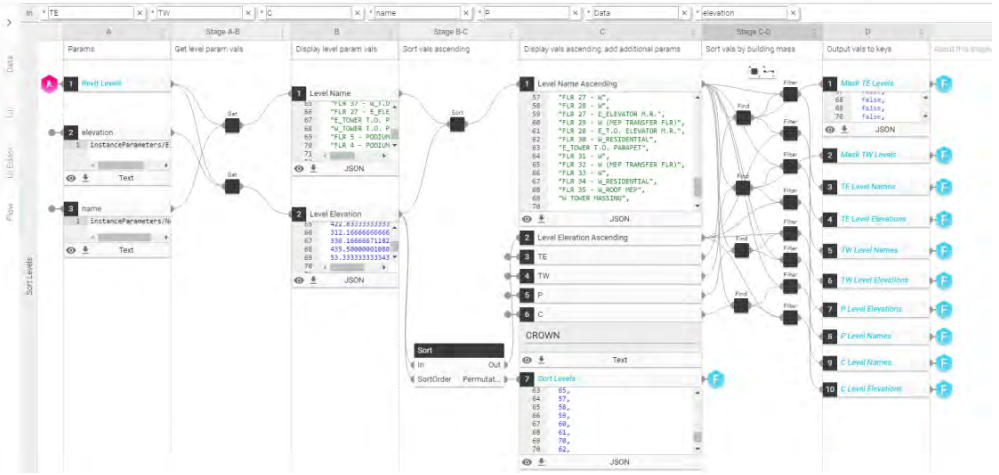
Interop Data Paradigms

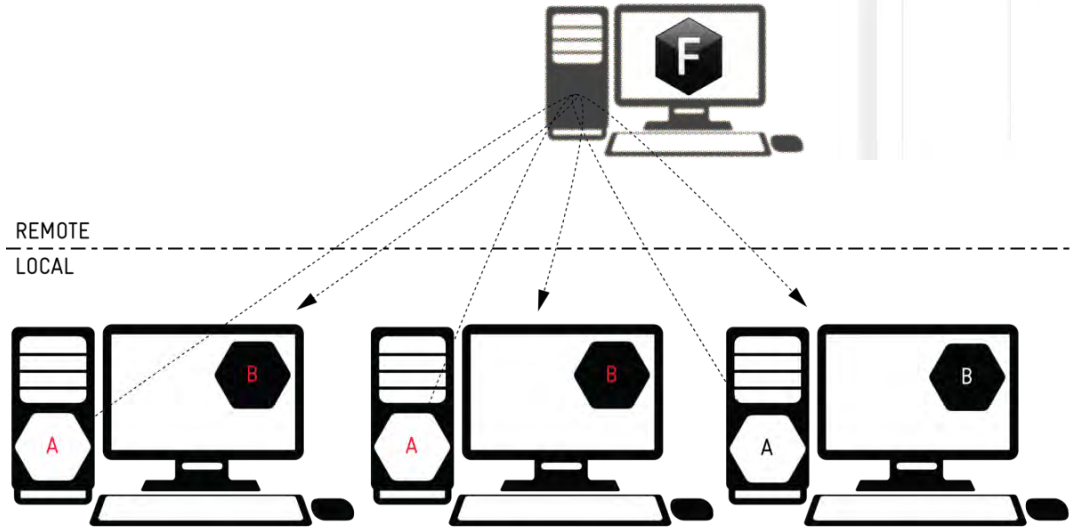
From Geometry to Data

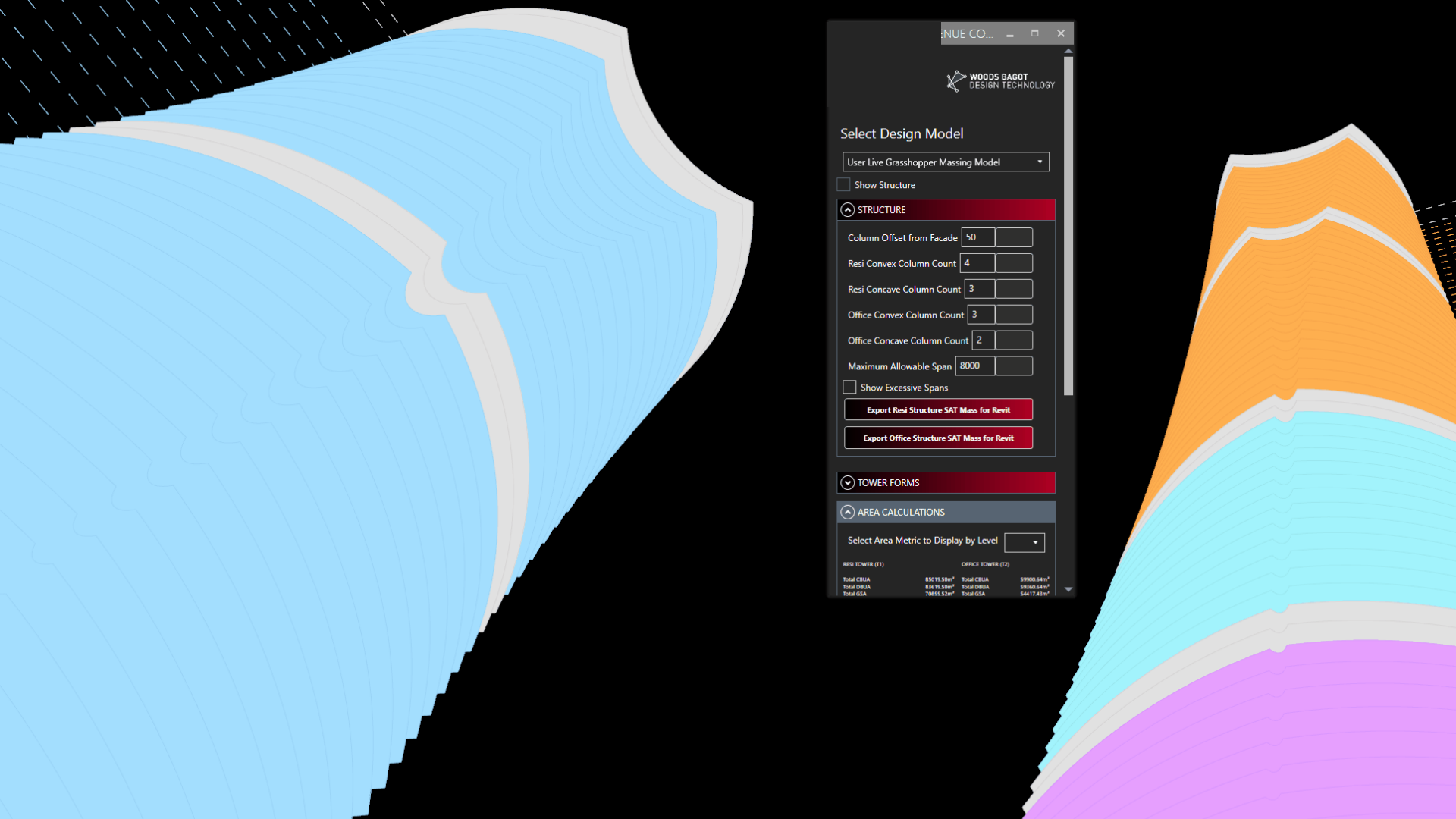


Interop Data Paradigms

Cloud-based Visual Programming for Designing Information Exchange

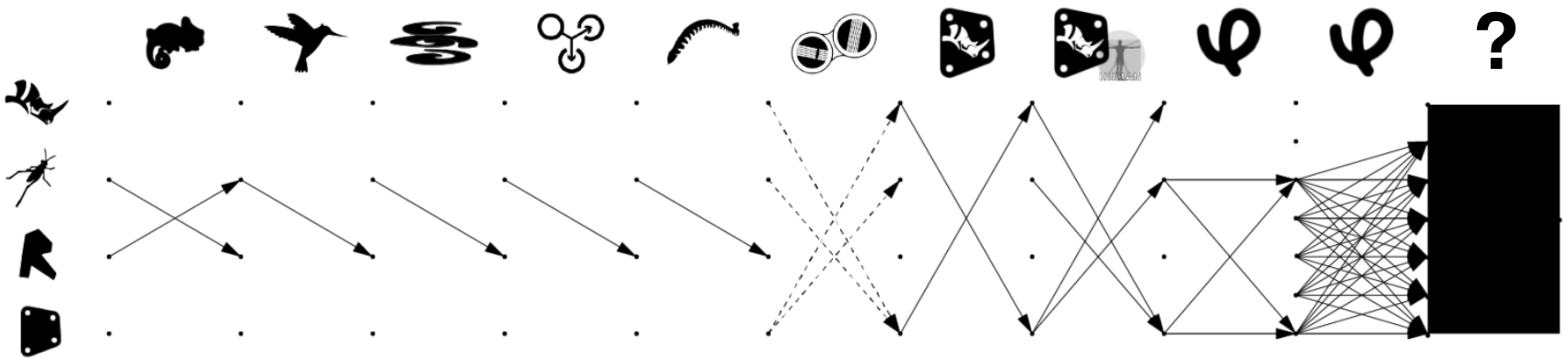





ENUE CO...
WOODS BAGOT
DESIGN TECHNOLOGY
Select Design Model
User Live Grasshopper Massing Model
Show Structure
STRUCTURE
Column Offset from Facade 50
Resi Convex Column Count 4
Resi Concave Column Count 3
Office Convex Column Count 3
Office Concave Column Count 2
Maximum Allowable Span 8000
Show Excessive Spans
Export Resi Structure SAT Mass for Revit
Export Office Structure SAT Mass for Revit
TOWER FORMS
AREA CALCULATIONS
Select Area Metric to Display by Level
RESI TOWER (T1)
OFFICE TOWER (T2)

From Platform Limitations to Open Development

Open Formats for Authoring and Sharing Information



Thank you.