Computational Design and BIM tools for big infrastructural projects

Arch. Christian Pallaria

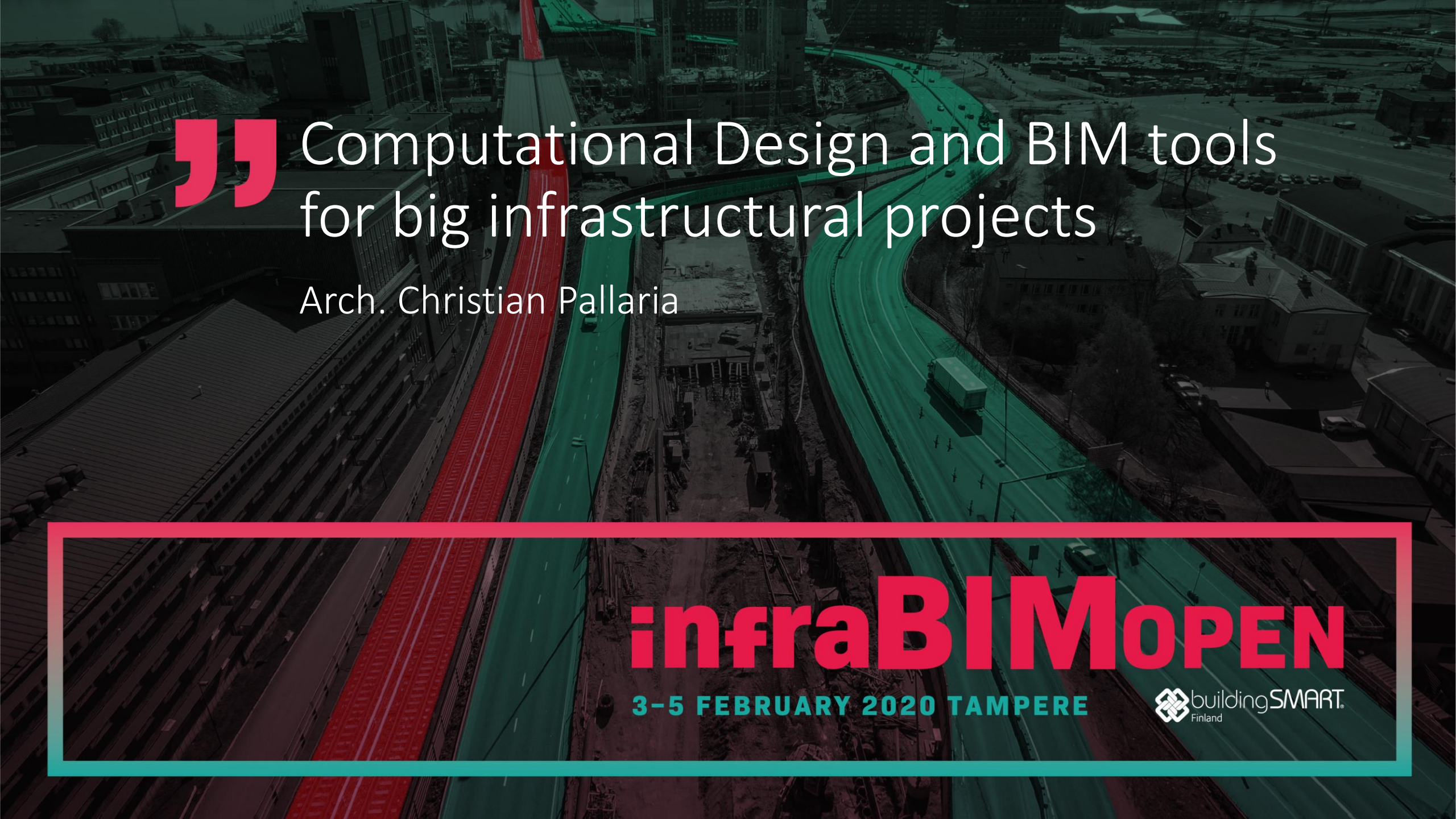
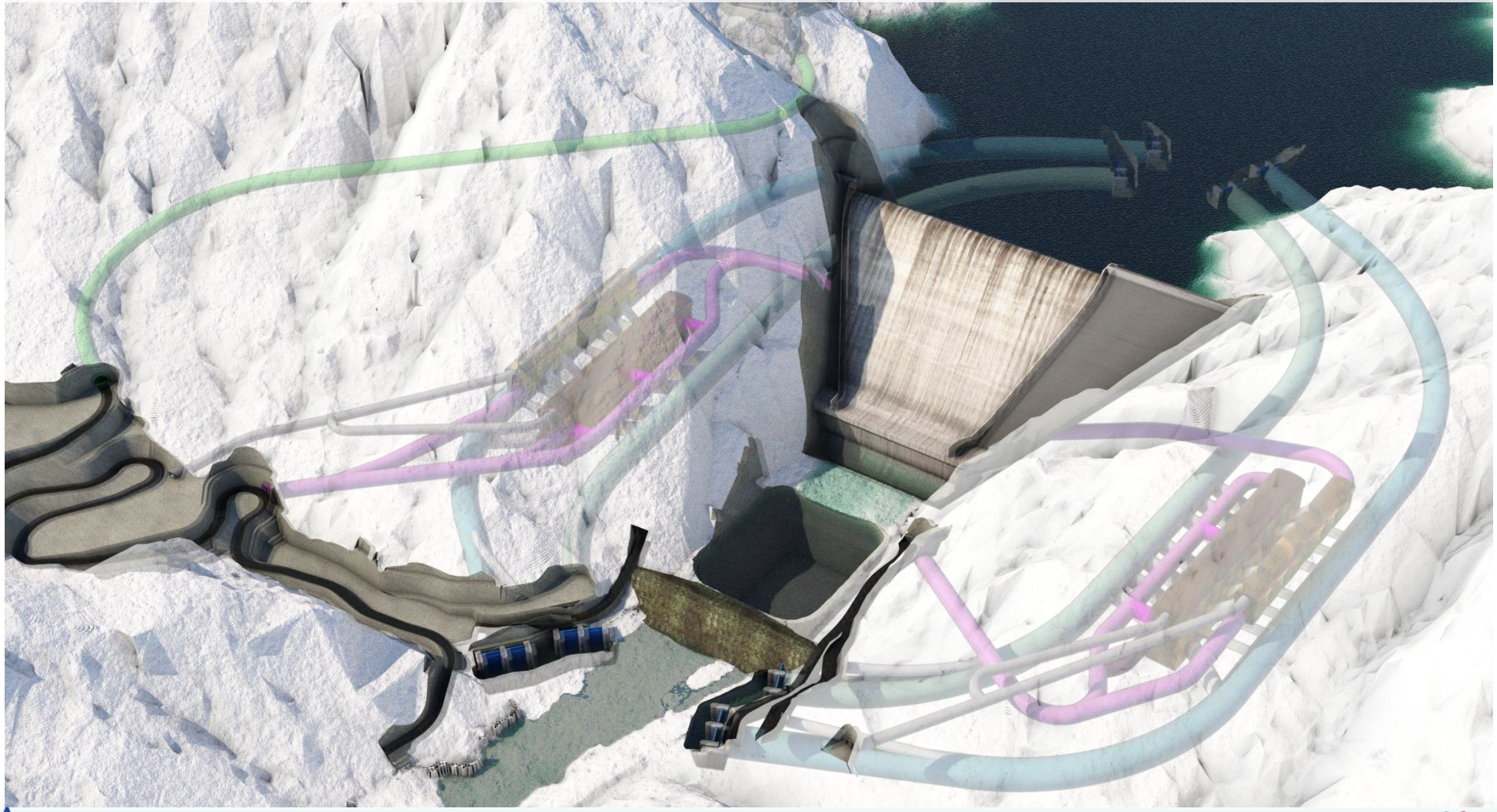
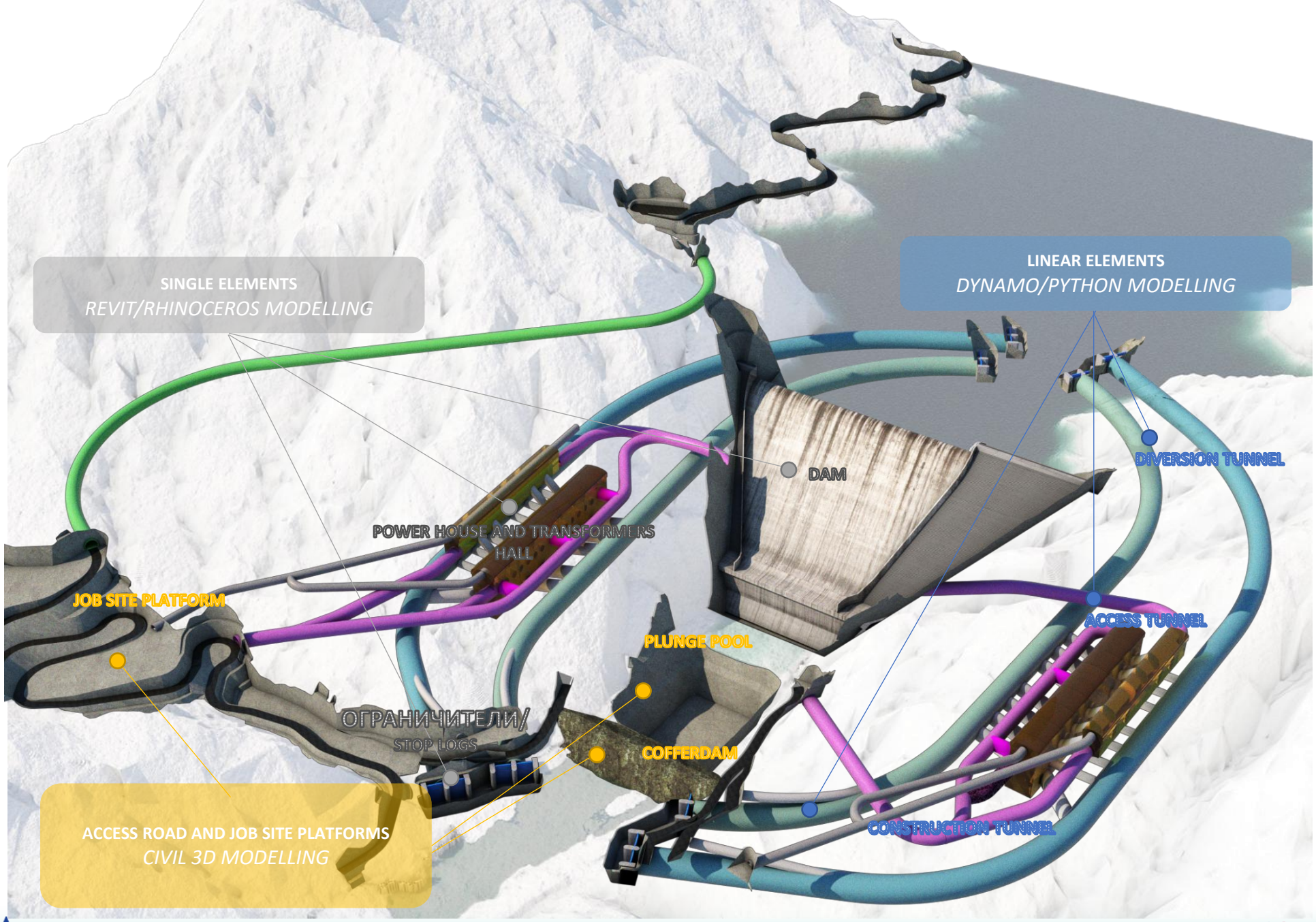


Table of contents

Table of contents

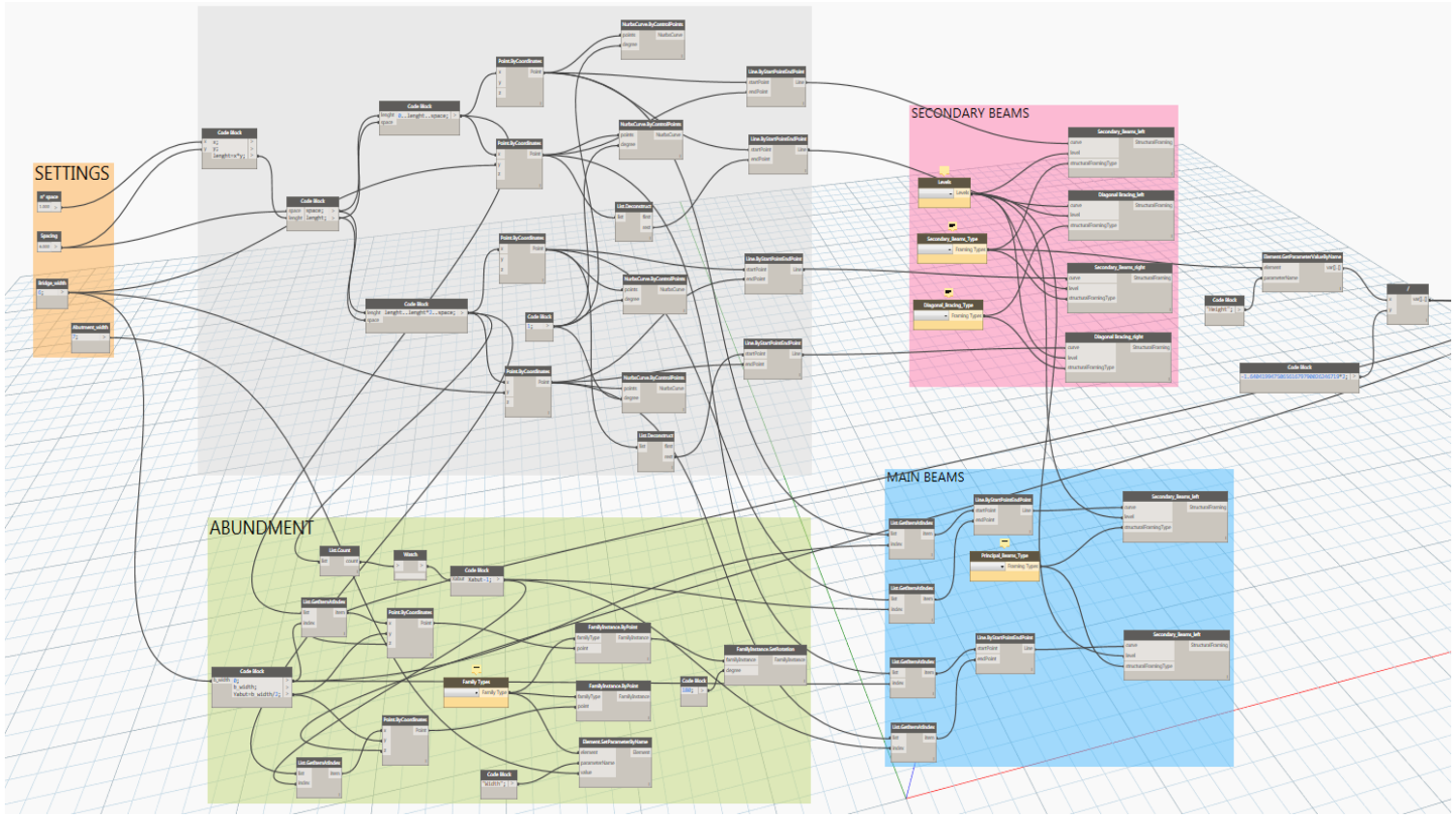
Hydroelectric power plants




SINGLE ELEMENTS
REVIT/RHINOCEROS MODELLING
LINEAR ELEMENTS
DYNAMO/PYTHON MODELLING
DAM
DIVERSION TUNNEL
POWER HOUSE AND TRANSFORMERS HALL
JOB SITE PLATFORM
ACCESS TUNNEL
PLUNGE POOL
ОГРАНИЧИТЕЛИ/
STOP LOGS
COFFERDAM
CONSTRUCTION TUNNEL
ACCESS ROAD AND JOB SITE PLATFORMS
CIVIL 3D MODELLING

Dynamo...our best friend

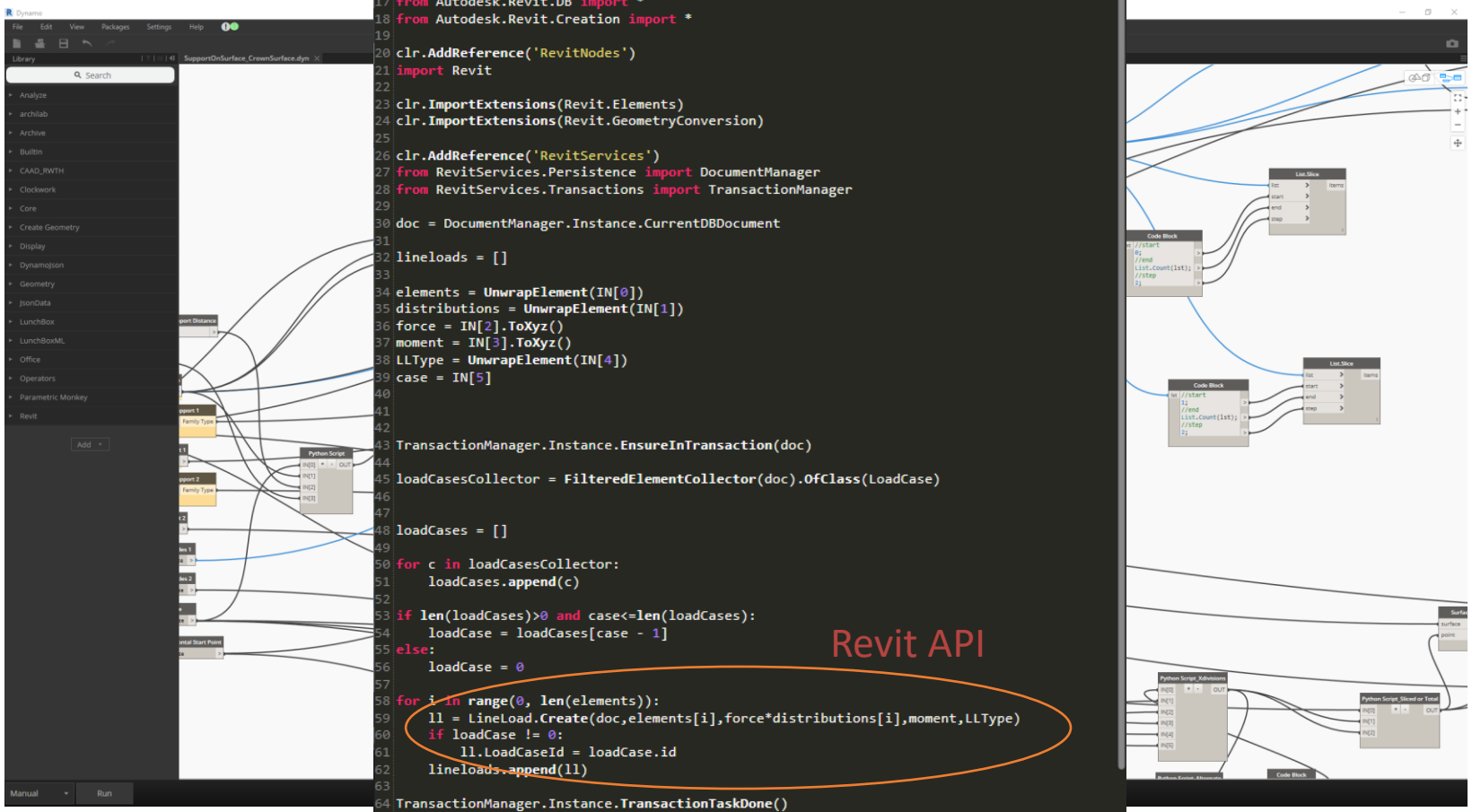
- Dynamo is a visual programming language. It consists of standard nodes that allow you to do many operations. But, when the nodes doesn't do what you need, you can program your custom node using Python programme language



Dynamo...

Python is a simple program … nodes that are not already compiled directly in …

It's a bit difficult compared … knowledge, but permits to enter inside the software …



```
#Declaration section#

import sys
import clr

pyt_path = r'C:\Program Files (x86)\IronPython 2.7\Lib'
sys.path.append(pyt_path)
import os

clr.AddReference('ProtoGeometry')
from Autodesk.DesignScript.Geometry import *

clr.AddReference('RevitAPIUI')
from  Autodesk.Revit.UI import *

clr.AddReference('RevitAPI')
from Autodesk.Revit.DB import *
from Autodesk.Revit.Creation import *

clr.AddReference('RevitNodes')
import Revit

clr.ImportExtensions(Revit.Elements)
clr.ImportExtensions(Revit.GeometryConversion)

clr.AddReference('RevitServices')
from RevitServices.Persistence import DocumentManager
from RevitServices.Transactions import TransactionManager

doc = DocumentManager.Instance.CurrentDBDocument

lineloads = []

elements = UnwrapElement(IN[0])
distributions = UnwrapElement(IN[1])
force = IN[2].ToXyz()
moment = IN[3].ToXyz()
LLType = UnwrapElement(IN[4])
case = IN[5]



TransactionManager.Instance.EnsureInTransaction(doc)

loadCasesCollector = FilteredElementCollector(doc).OfClass(LoadCase)


loadCases = []

for c in loadCasesCollector:
    loadCases.append(c)

if len(loadCases)>0 and case<=len(loadCases):
    loadCase = loadCases[case - 1]
else:
    loadCase = 0

for i in range(0, len(elements)):
    ll = LineLoad.Create(doc,elements[i],force*distributions[i],moment,LLType)
    if loadCase != 0:
        ll.LoadCaseId = loadCase.id
    lineloads.append(ll)

TransactionManager.Instance.TransactionTaskDone()
```

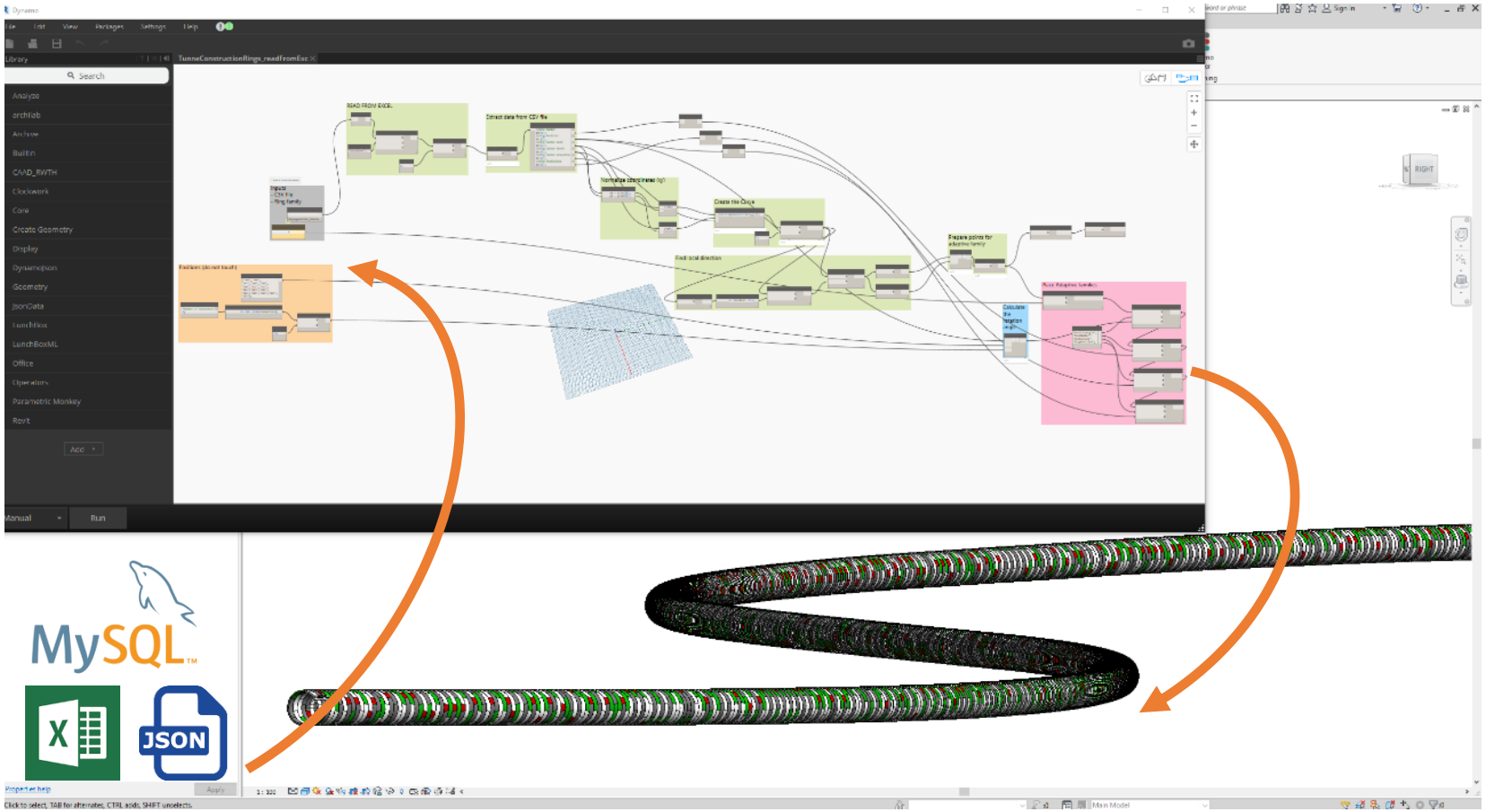
Python Script

Run | Save Changes | Revert

Revit API

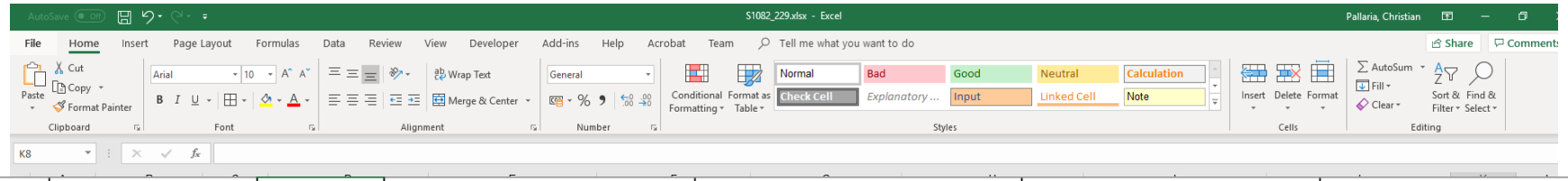
TBM modelling using Dynamo

Using Dynamo we can read a Database (excel, JSON, csv, etc…)
model a TBM tunnel and write parameters inside the model



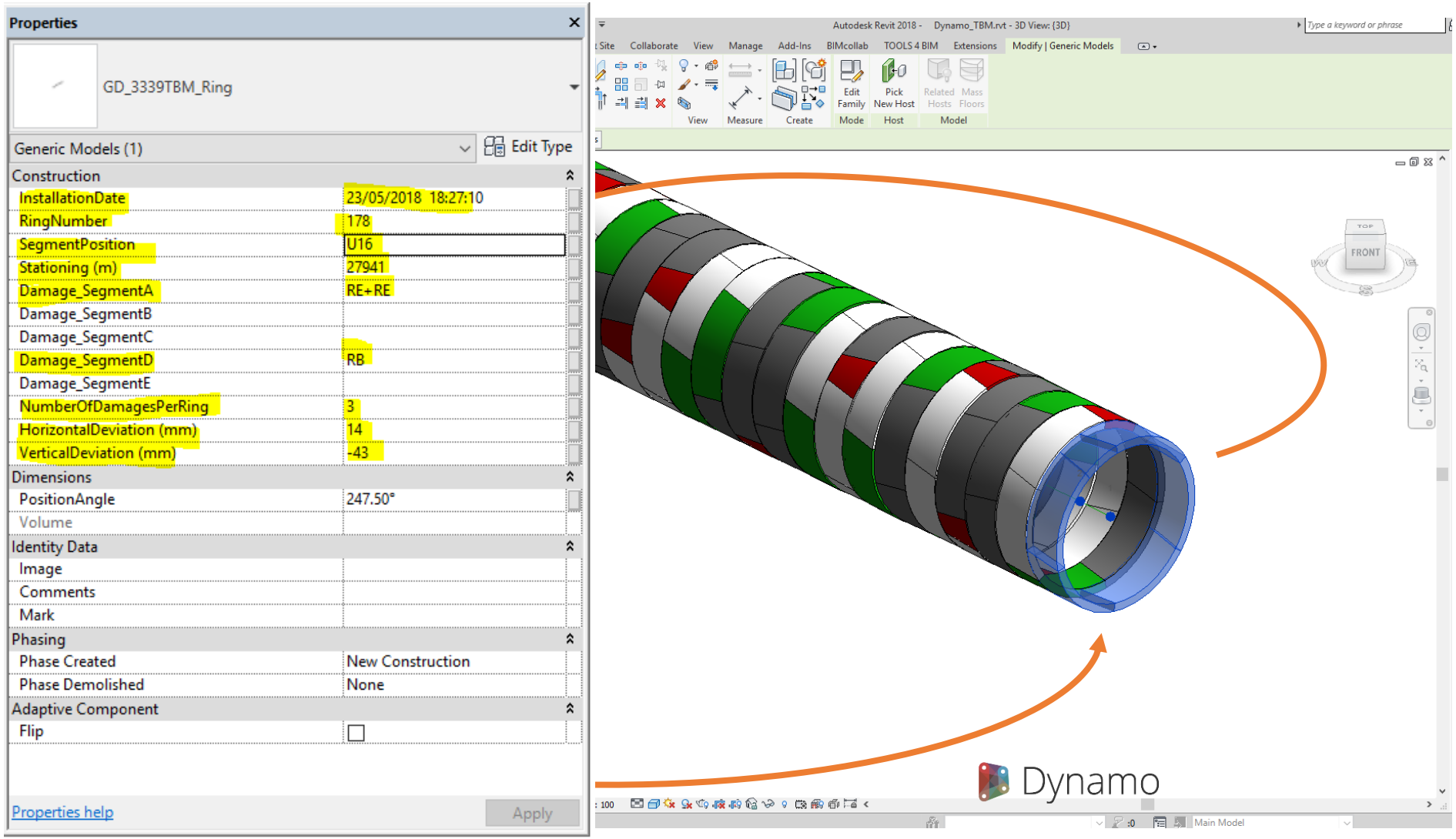
TBM modelling using Dynamo

The excel file has got information about ring number, date and time, name, position in XYZ, stationing and any other information that can be read and written in the model as information for each element

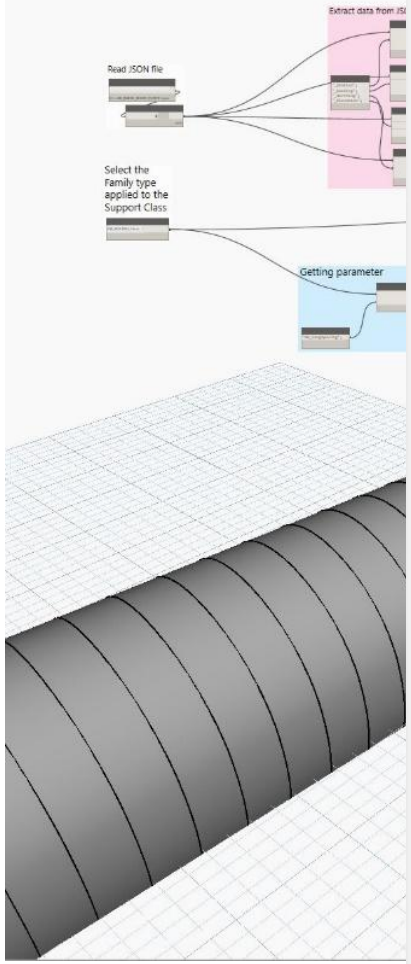


Ring Number	Date and time (Ring selection)	Ring	Ring center EAST value [m]	Ring center NORTH value [m]	Ring center HEIGHT value [m]	Ring center Stationing [m]
1	23.05.2018 18:27:10	U13	400217,058	4568159,562	51,136	27941,38
2	24.05.2018 07:00:38	U08	400218,015	4568158,407	51,147	27939,88
3	24.05.2018 15:09:40	U13	400219,004	4568157,270	51,144	27938,37
4	24.05.2018 22:14:33	U15	400220,011	4568156,157	51,185	27936,87
5	26.05.2018 15:53:50	U10	400221,019	4568155,044	51,181	27935,37
6	26.05.2018 23:44:07	U05	400222,031	4568153,936	51,197	27933,87
7	27.05.2018 06:28:25	U07	400223,026	4568152,809	51,206	27932,37
8	28.05.2018 16:17:58	U12	400224,026	4568151,695	51,200	27930,87
9	29.05.2018 03:52:33	U04	400225,026	4568150,579	51,204	27929,37
10	29.05.2018 11:37:20	U09	400226,022	4568149,456	51,188	27927,87
11	29.05.2018 18:07:55	U01	400227,012	4568148,326	51,178	27926,37
25	24	01.06.2018 22:27:04	U01	400239,920	4568133,675	51,182	27906,84	36,04	5	2
26	25	02.06.2018 02:53:39	U09	400240,913	4568132,553	51,186	27905,35	37,53	4	5
27	26	02.06.2018 06:28:20	U14	400241,858	4568131,384	51,201	27903,84	39,04	68	20
28	27	02.06.2018 11:01:49	U06	400242,887	4568130,287	51,180	27902,34	40,54	23	-2
29	28	02.06.2018 14:57:09	U01	400243,863	4568129,142	51,187	27900,83	42,05	49	4
30	29	03.06.2018 23:04:20	U09	400244,832	4568128,035	51,191	27899,36	43,52	56	7
31	30	04.06.2018 02:01:08	U14	400245,855	4568126,897	51,186	27897,83	45,05	41	0
32	31	04.06.2018 05:31:47	U16	400246,835	4568125,812	51,199	27896,37	46,51	25	12
33	32	04.06.2018 09:45:40	U05	400247,870	4568124,667	51,210	27894,83	48,05	7	21
34	33	04.06.2018 13:34:03	U10	400248,870	4568123,547	51,212	27893,33	49,55	-2	21
35	34	04.06.2018 16:14:56	U08	400249,866	4568122,454	51,210	27891,85	51,03	-25	17

TBM modelling using Dynamo



NATM modelling using Dynamo

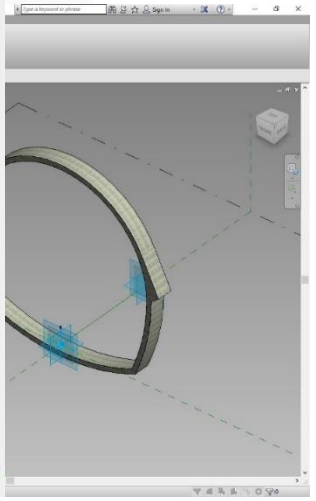


Tipi di famiglia

Nome del tipo: PW3_15cm

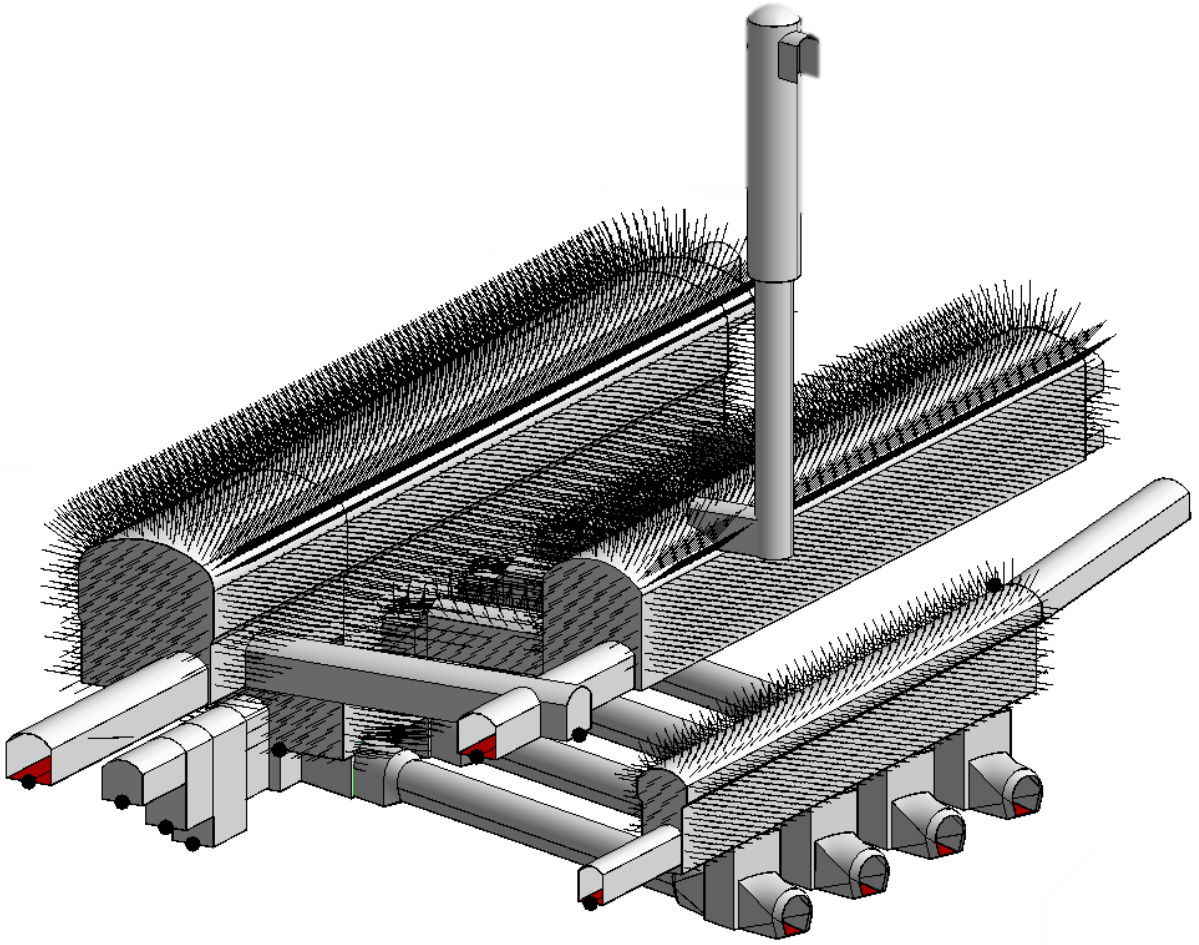
Parametri di ricerca

Parametro	Valore	Formula	Blocca
Costruzione			
GD_LongSpacing (default)	2.0000	=	☐
Materiali e finiture			
GD_Material	Concrete, Cast-in-Place, Gray	=	
Dati			
GD_STD_NumberOfElement	16	=	☐
GD_STR_NumberOfElement	0	=	☐
GD_TEI_ElementVolume	0.000000	=	
GD_WIM_ElementArea.	0.000000	=	
Visibilità			
GD_ProfileSelector<Modelli gene	PRF_PW3	=	
Dati identità			
GD_CreatedBy	Alberto Bolzonello	= "Alberto Bolzonello"	
GD_ElementDescription	SHC: SFR40 - 15cm THICK	=	
Immagine tipo		=	
Nota chiave		=	
Modello	RCK	= "RCK"	
Produttore	Geodata Engineering	= "Geodata Engineering"	
Commenti sul tipo	SHC(T826.2) + STD(I62.1)	=	
URL		=	
Descrizione		=	
Codice assieme		=	
Costo		=	



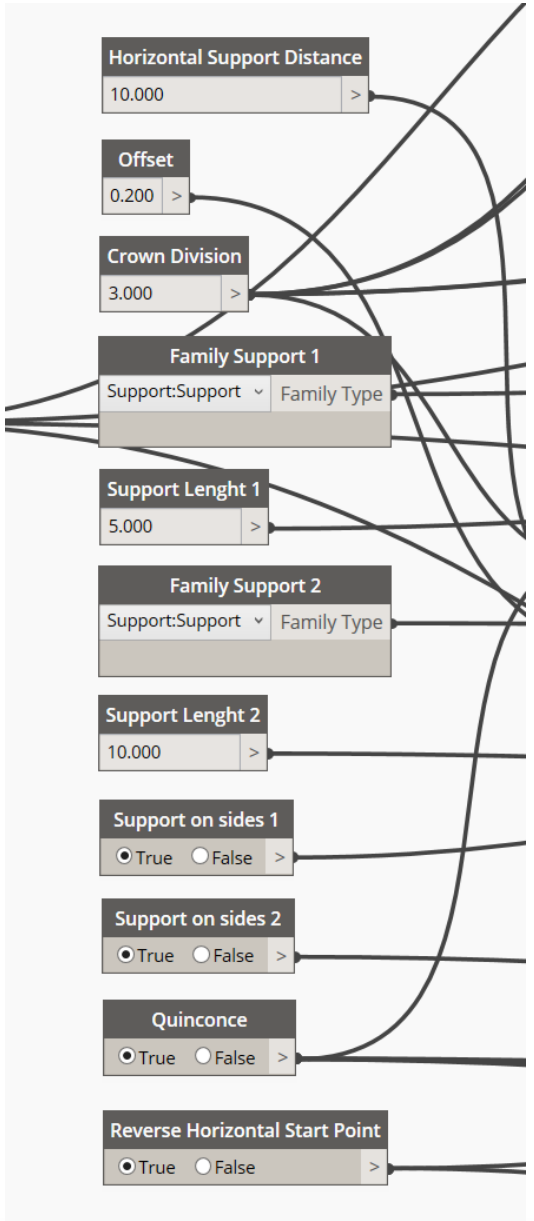
Support modelling

- Dynamo has been used also for modelling of supports in the Power House of the plants.
- Clicking the surface, defining input data, supports were modelled automatically



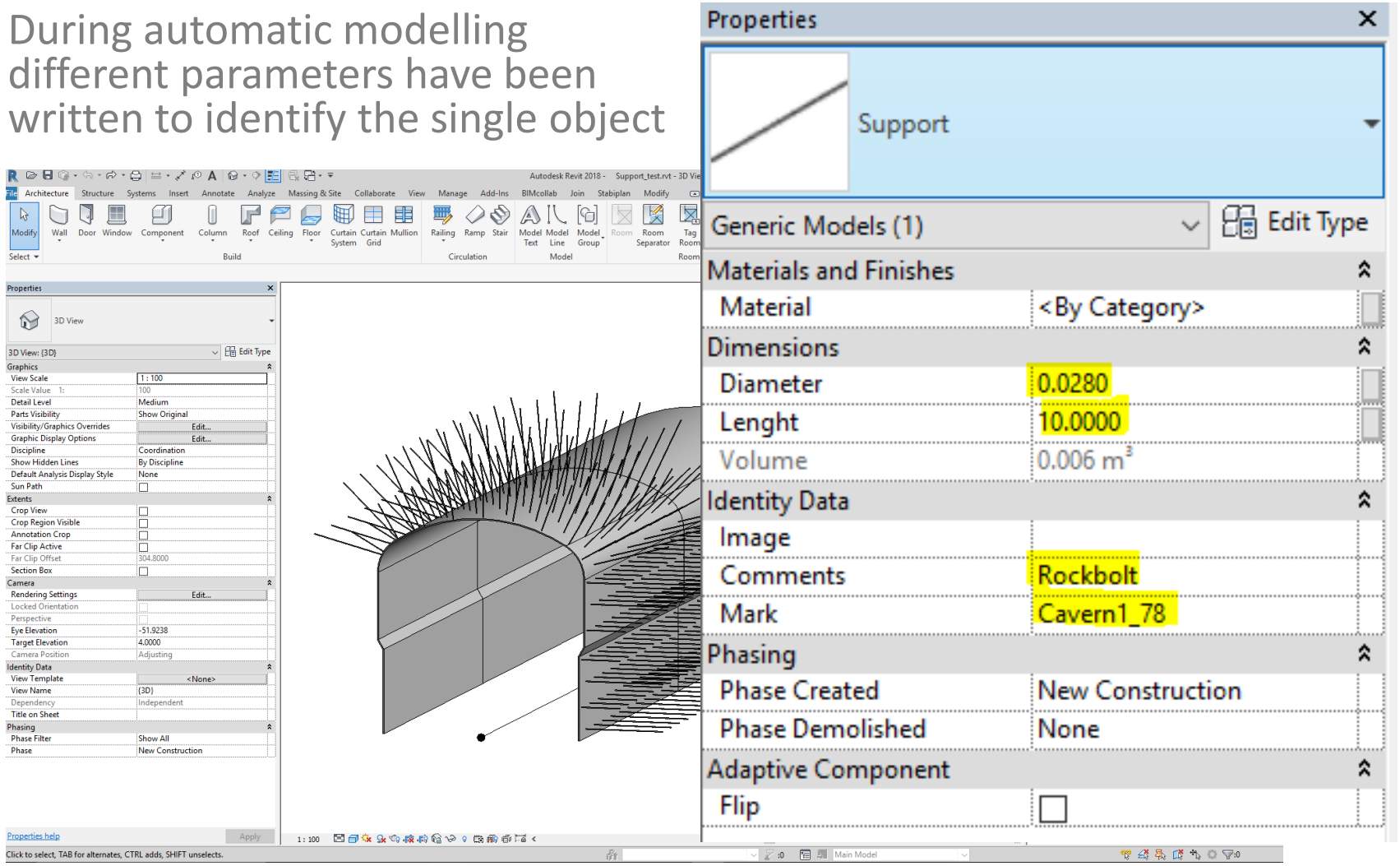
Support modelling

- The code allows to control:
  - Initial offset of the starting point;
  - X support distance
  - Y support distance
  - Support length
  - Support radius
  - Support material and tipology
  - Set the type of support disposition (quinconce or normal)



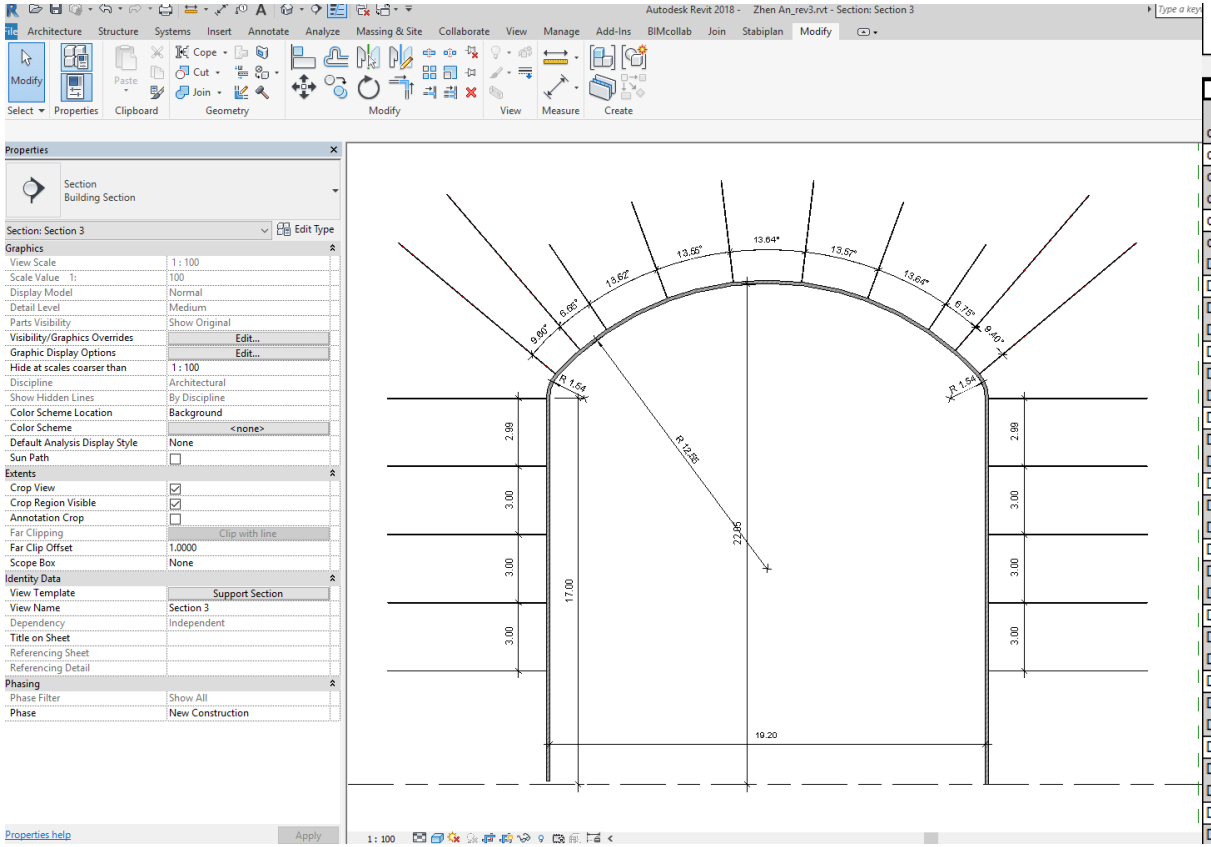
Support modelling

During automatic modelling different parameters have been written to identify the single object



Support modelling

The model has been used to realize 2D drawings and BoQ



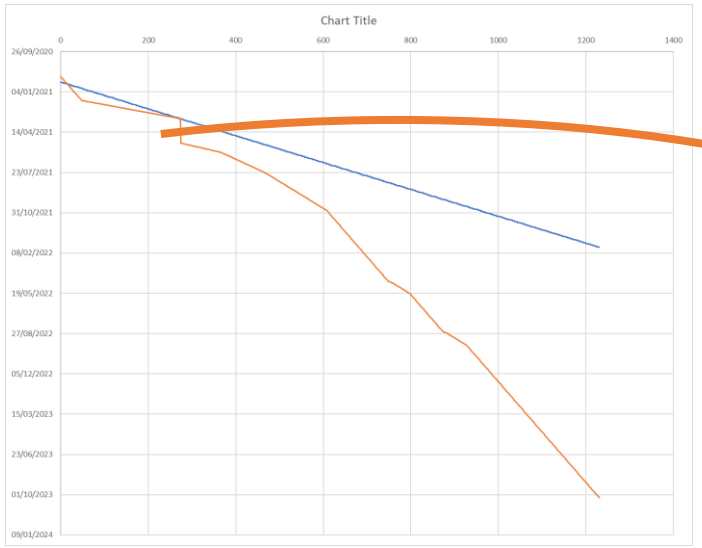
<BoQ_Supports>

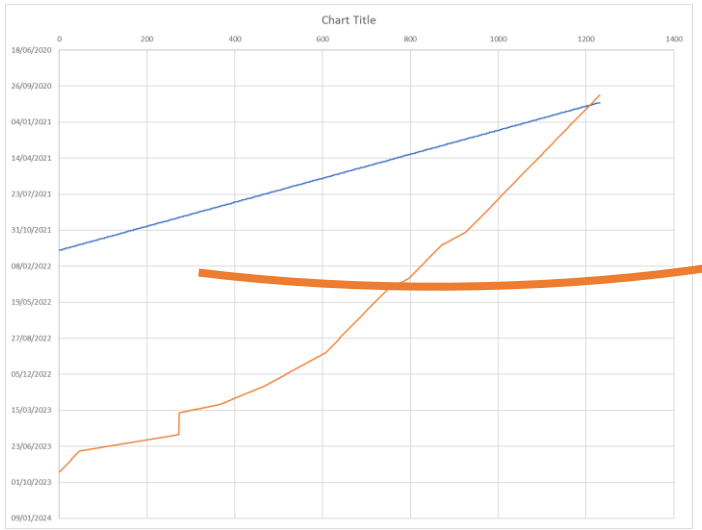
A	B
Type	Lenght
cable-bolt-L20	
cable-bolt-L20	880.00 m
cable-bolt-L20: 44	880.00 m
cable-bolt-L25	
cable-bolt-L25	2400.00 m
cable-bolt-L25: 96	2400.00 m
D25/L3_rock-bolt	
D25/L3_rock-bolt	1776.00 m
D25/L3_rock-bolt: 592	1776.00 m
D25/L4.5_rock-bolt	
D25/L4.5_rock-bolt	711.00 m
D25/L4.5_rock-bolt: 158	711.00 m
D25/L5_rock-bolt	
D25/L5_rock-bolt	3145.00 m
D25/L5_rock-bolt: 629	3145.00 m
D28/L4.5_rock-bolt	
D28/L4.5_rock-bolt	12640.50 m
D28/L4.5_rock-bolt: 2809	12640.50 m
D28/L5.5_rock-bolt	
D28/L5.5_rock-bolt	1826.00 m
D28/L5.5_rock-bolt: 332	1826.00 m
D28/L7_rock-bolt	
D28/L7_rock-bolt	4011.00 m
D28/L7_rock-bolt: 573	4011.00 m
D28/L9_rock-bolt	
D28/L9_rock-bolt	15318.00 m
D28/L9_rock-bolt: 1702	15318.00 m
D28/L9_rock-bolt	
D28/L9_rock-bolt	3141.00 m
D28/L9_rock-bolt: 349	3141.00 m
D32/L9_pre-stressed	
D32/L9_pre-stressed	16569.00 m
D32/L9_pre-stressed: 1841	16569.00 m

4D and 5D simulations

Temporary lining

Final lining






```
newPk = 0
newSpace = 0

LimMin = 20

LimMax = 50


for dataA in Tlin:

    for dataB in Texc:
        if Texc.index(dataB)+1 < len(Texc):
            if dataA == dataB or datetime.datetime.strptime(dataA, '%d/%m/%Y') >= datetime.datetime.strptime(dataB, '%d/%m/%Y') and datetime.datetime.strptime(dataA, '%d/%m/%Y') < datetime.datetime.
                if PKexc[Texc.index(dataB)] - PKlin[Tlin.index(dataA)] <= LimMin :
                    print (Tlin[Tlin.index(dataA)])
                    newPk = PKlin[Tlin.index(dataA)] + LimMax
                    print(newPk)
                    for pk in PKexc:
                            if newPk <= pk and newPk > PKexc[PKexc.index(pk)-1]:
                                print(pk)
                                newSpace = abs(datetime.datetime.strptime(Texc[PKexc.index(pk)], '%d/%m/%Y') - datetime.datetime.strptime(Tlin[Tlin.index(dataA)], '%d/%m/%Y'))
                                Tlin[Tlin.index(dataA)] = Texc[PKexc.index(pk)]
                                print (newSpace)

                    for i in range(Tlin.index(dataA),len(Tlin)):
                            Tlin[i]= (datetime.datetime.strptime(Tlin[i], '%d/%m/%Y')) + datetime.timedelta(newSpace.days)
                            Tlin[i] = Tlin[i].strftime("%d/%m/%Y")
                break

for i in range (1,(max(len(Tlin),len(Texc)))+1):
    if i <= len(Tlin):
        foglio1.cell(i,1).value = datetime.datetime.strptime(Tlin[i-1], '%d/%m/%Y')
        foglio1.cell(i,2).value = PKlin[i-1]
    if i <= len(Texc):
        foglio1.cell(i,4).value = datetime.datetime.strptime(Texc[i-1], '%d/%m/%Y')
        foglio1.cell(i,5).value = PKexc[i-1]

#if (len(Tlin)>len(Texc)):
#    foglio1.cell(len(Tlin)+1,1).value = datetime.datetime.strptime(Tlin[len(Tlin)], '%d/%m/%Y')
#    foglio1.cell(len(PKlin)+1,2).value = PKlin[len(PKlin)]
#else:
```

Elenco errori

0 Errori | 0 Avvisi | 0 Messaggi | Compilazione + IntelliSe | Elenco errori di ricerca

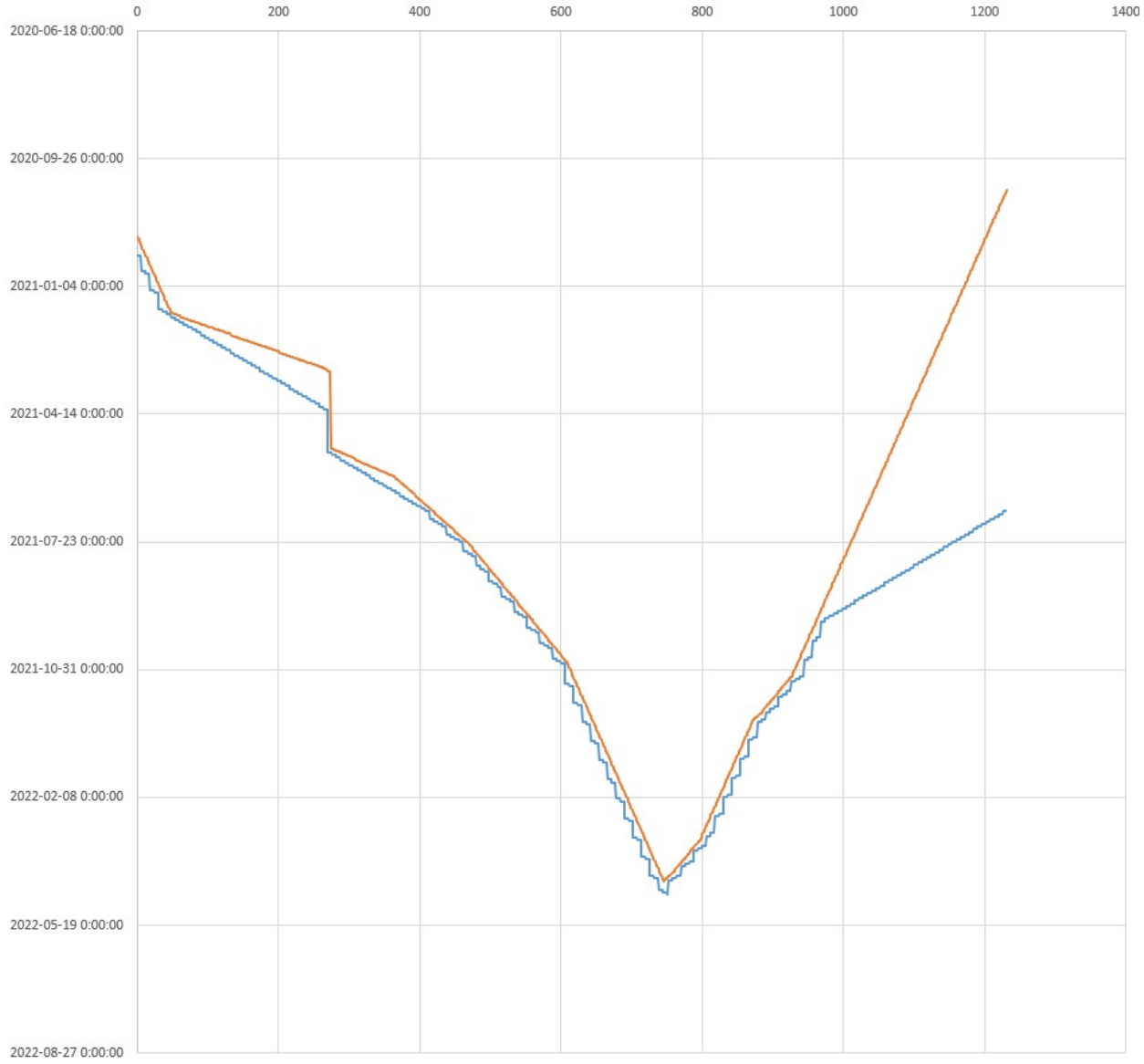
Codice | Descrizione | Progetto | File | Ri... | Stato eliminazione

4D simulation_test interattivo | Python 3.7 (64-bit) interattivo | Debug interattivo | Elenco errori

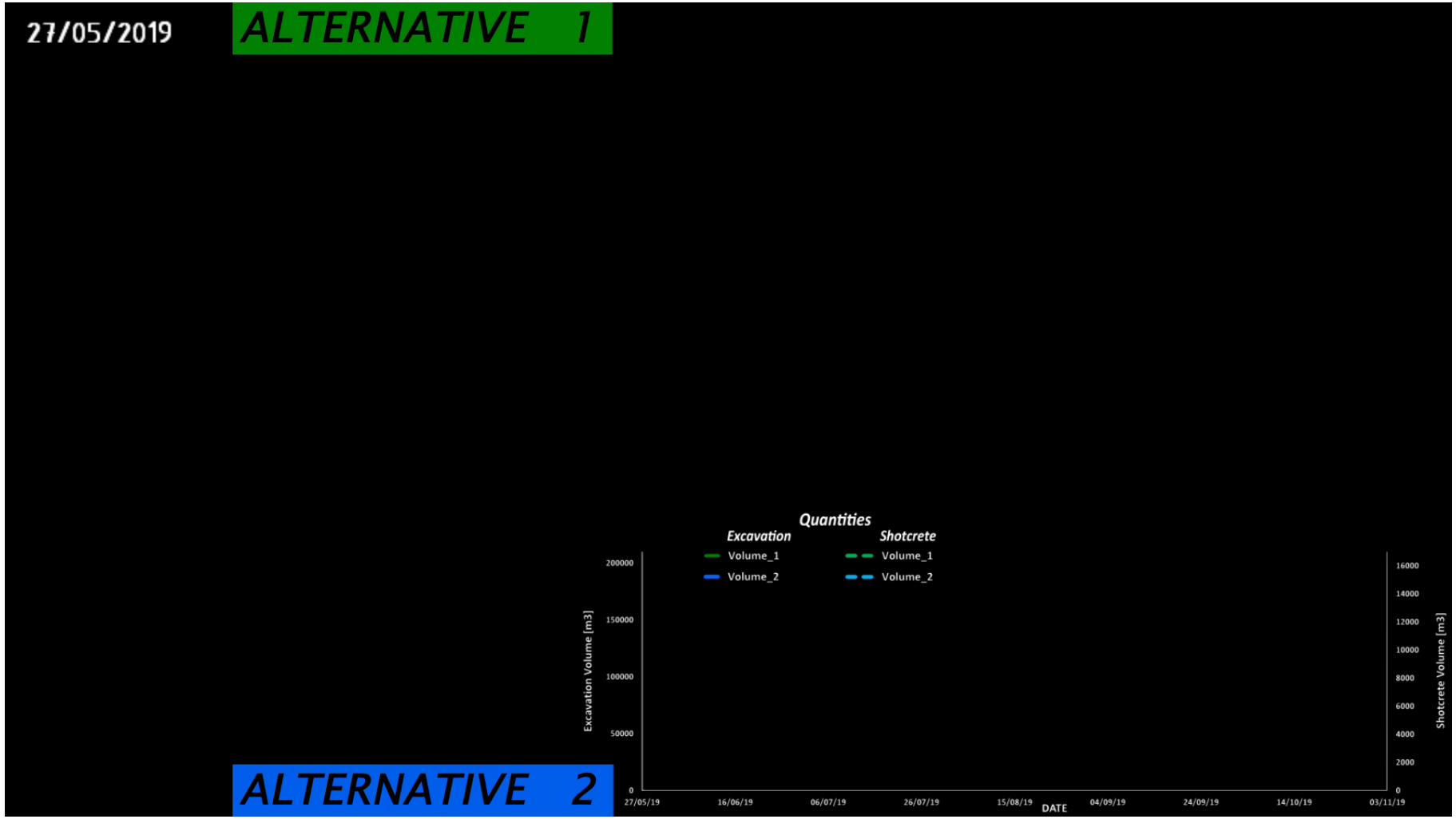
4D and 5D simulations

Today we insert the starting date of the excavation and the code is able to calculate the start and end date of each element of the temporary excavation. At the same time it calculates also the final lining considering the distance limits from the excavation.

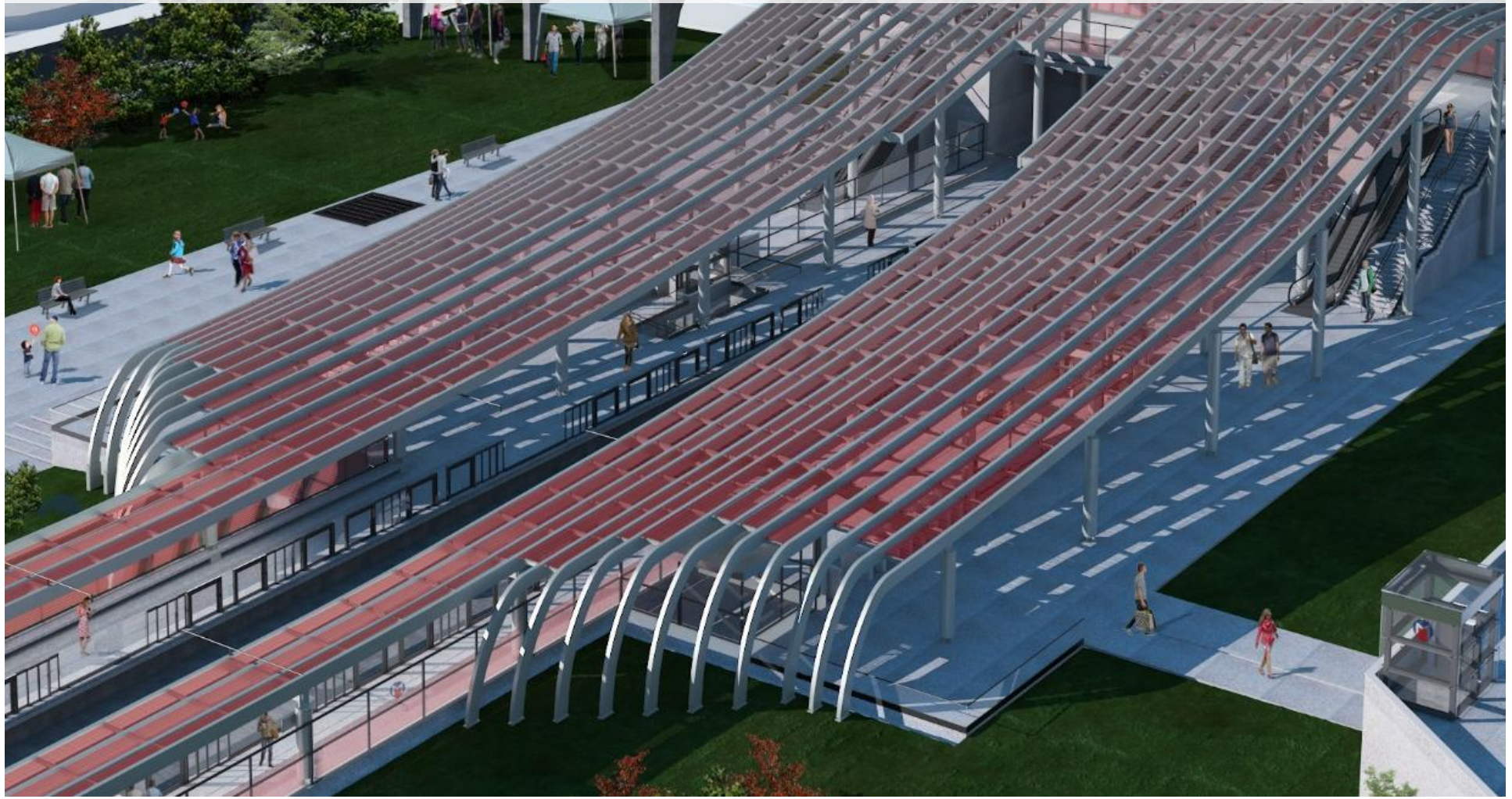
The result is not only a model full af these information that can be imported directly into Navisworks realizing a 4D simulation automatically, but it gives also a graph into an excel file with all information, which can be read by everyone



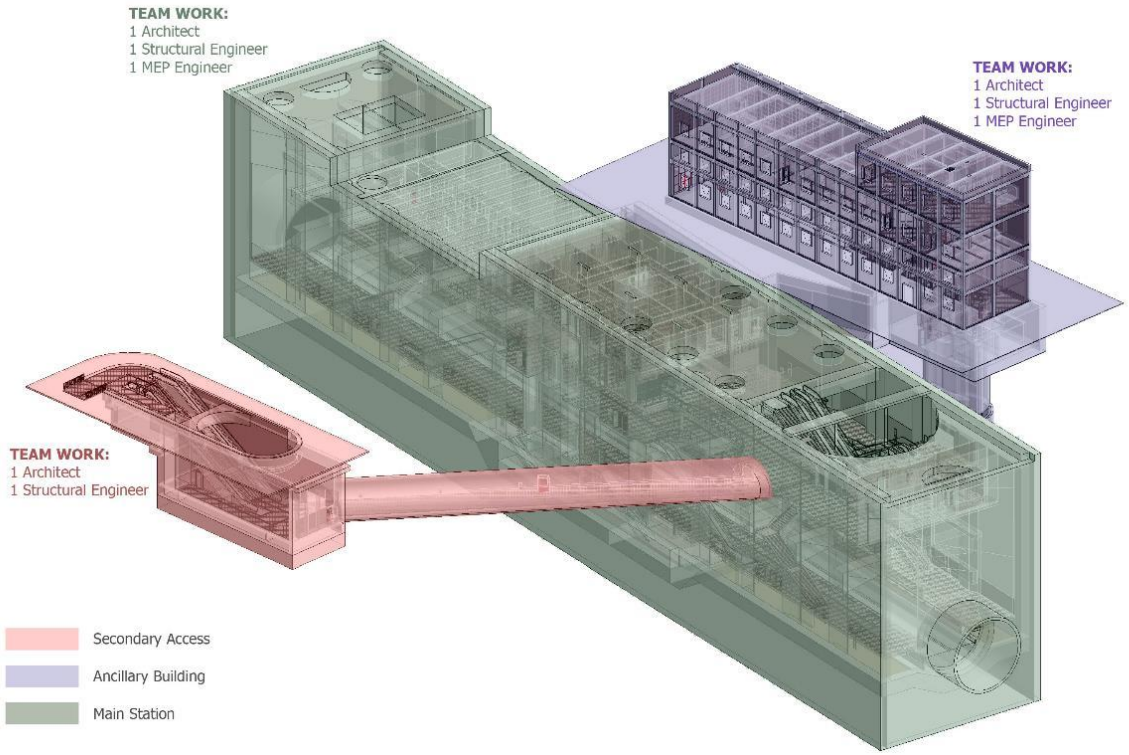
4D and 5D simulations: video



Metro and railways

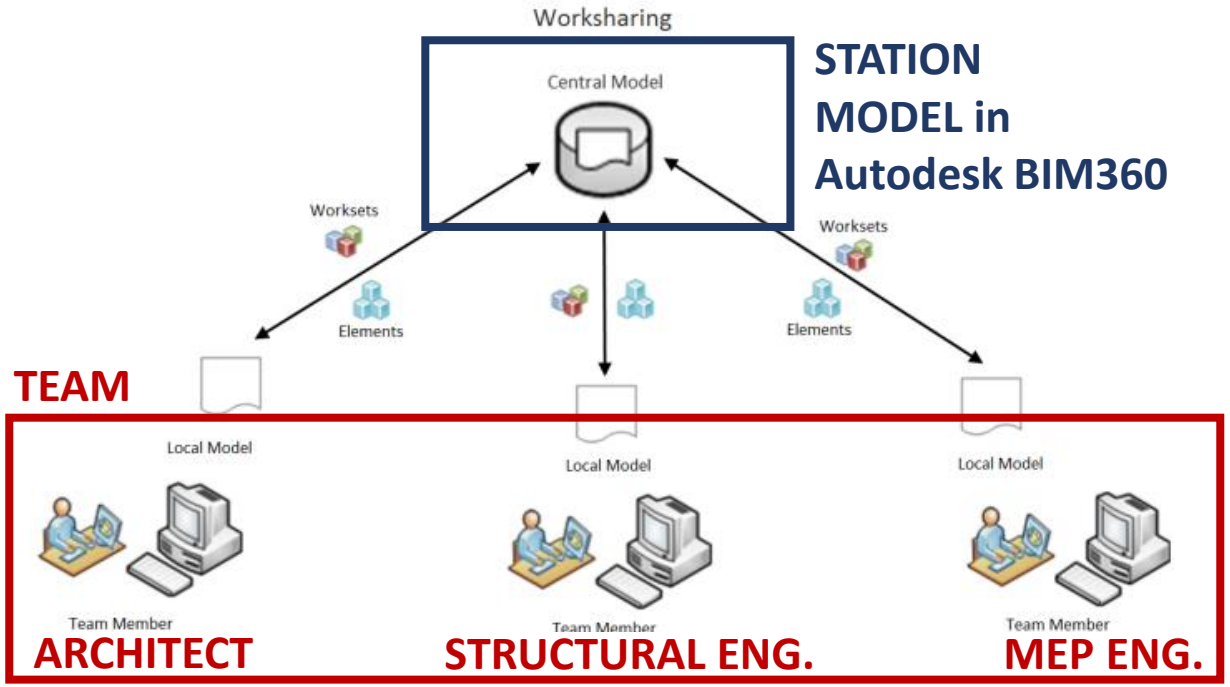


Architectural and functional design



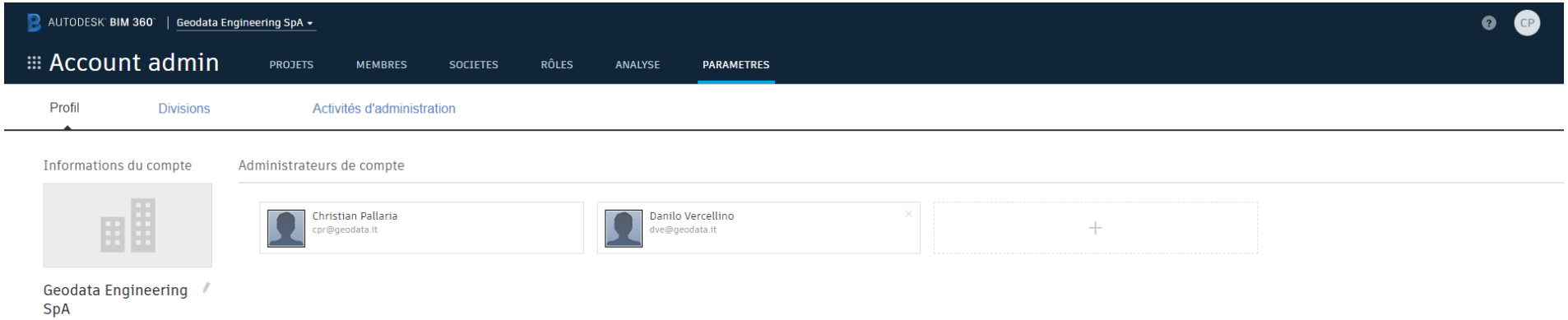
The use of the BIM approach has improved for the design management both within Geodata and towards other Stakeholders. In particularly, it has been extremely useful for **sharing the same model simultaneously with structural, architectural and MEP designers**.

Interaction and interface in a teamwork



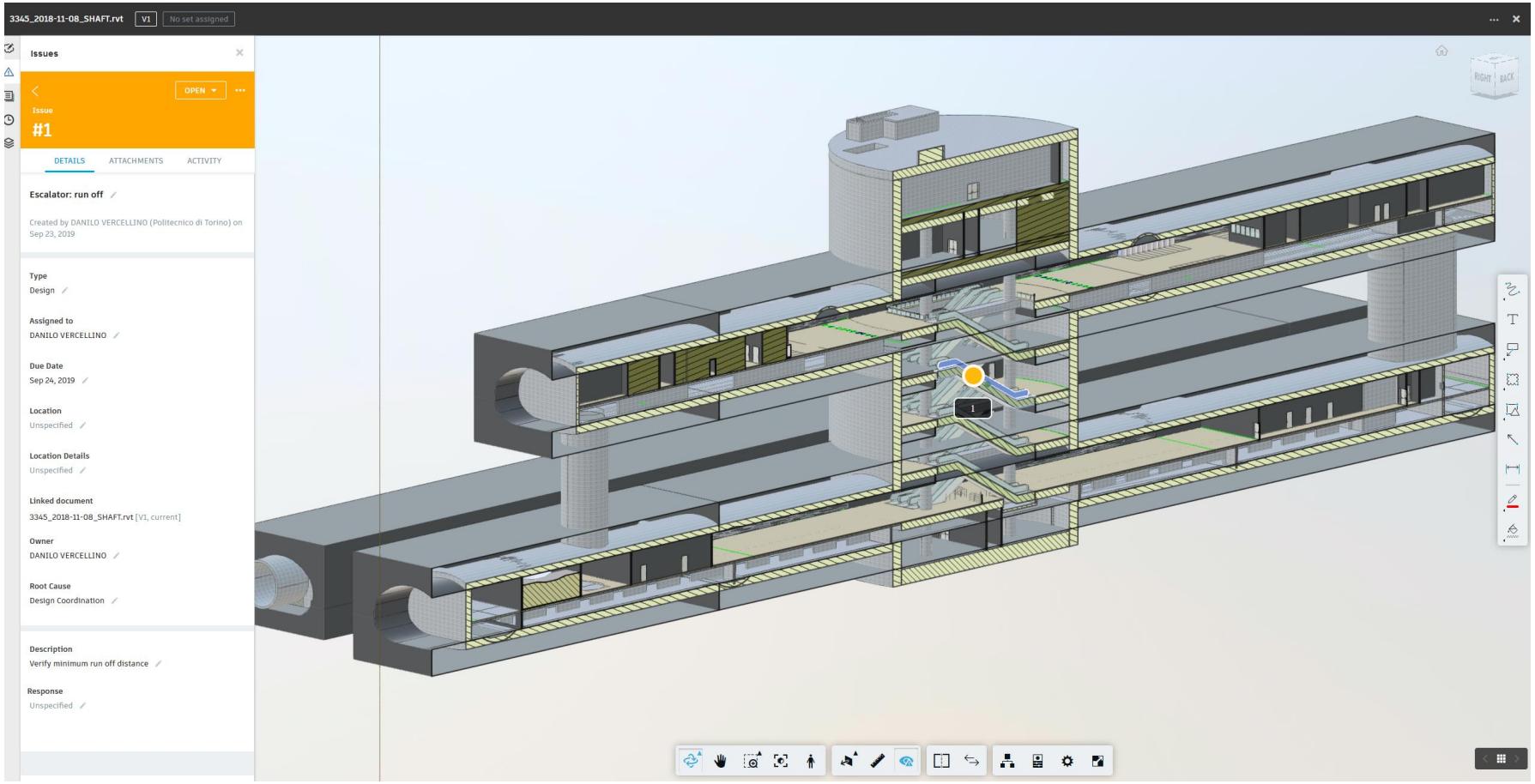
The **interaction and the interface with all the other related disciplines, structural design, MEP design, architectural finishes, is a must**. The greater the interaction between the disciplines is and the minors the negative impacts are during the construction phase.

Autodesk BIM360 for stakeholder collaboration



Autodesk BIM360 was used at the construction level, in order to give a full access of the design documentation to all stakeholders, with **different access rules** according to the user. Everyone was able to insert comments, even uploading images **directly from the station sites**. All this allowed a **simultaneous coordination between stakeholders**, especially between the Construction Supervision, the Contractor and the designers.

Autodesk BIM360 for stakeholder collaboration



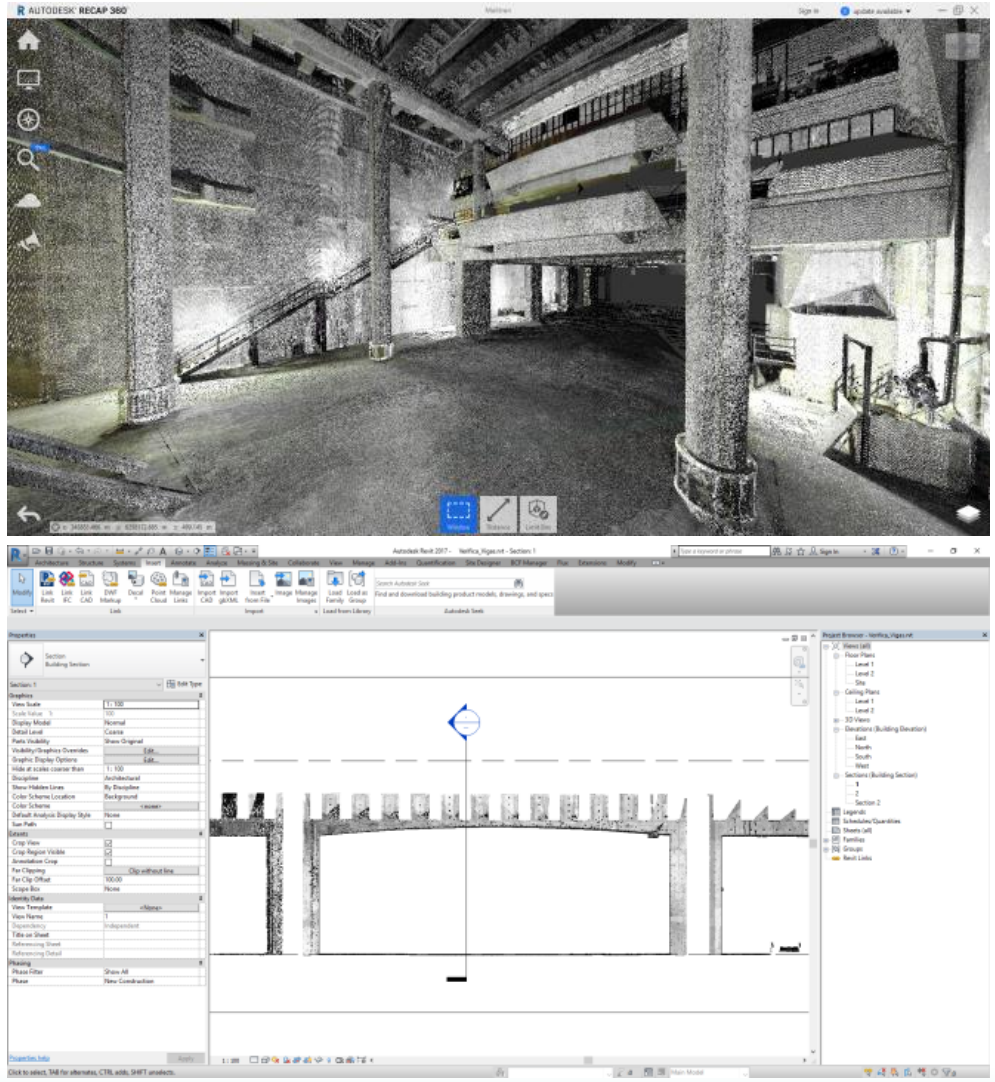
Santiago-Batuco Railway

Working with point cloud in Revit

An existing station in Santiago of Chile, built in 2004 with a empty space realized for the future cross line.

Design a new station in an exiting context is not easy with traditional CAD technology

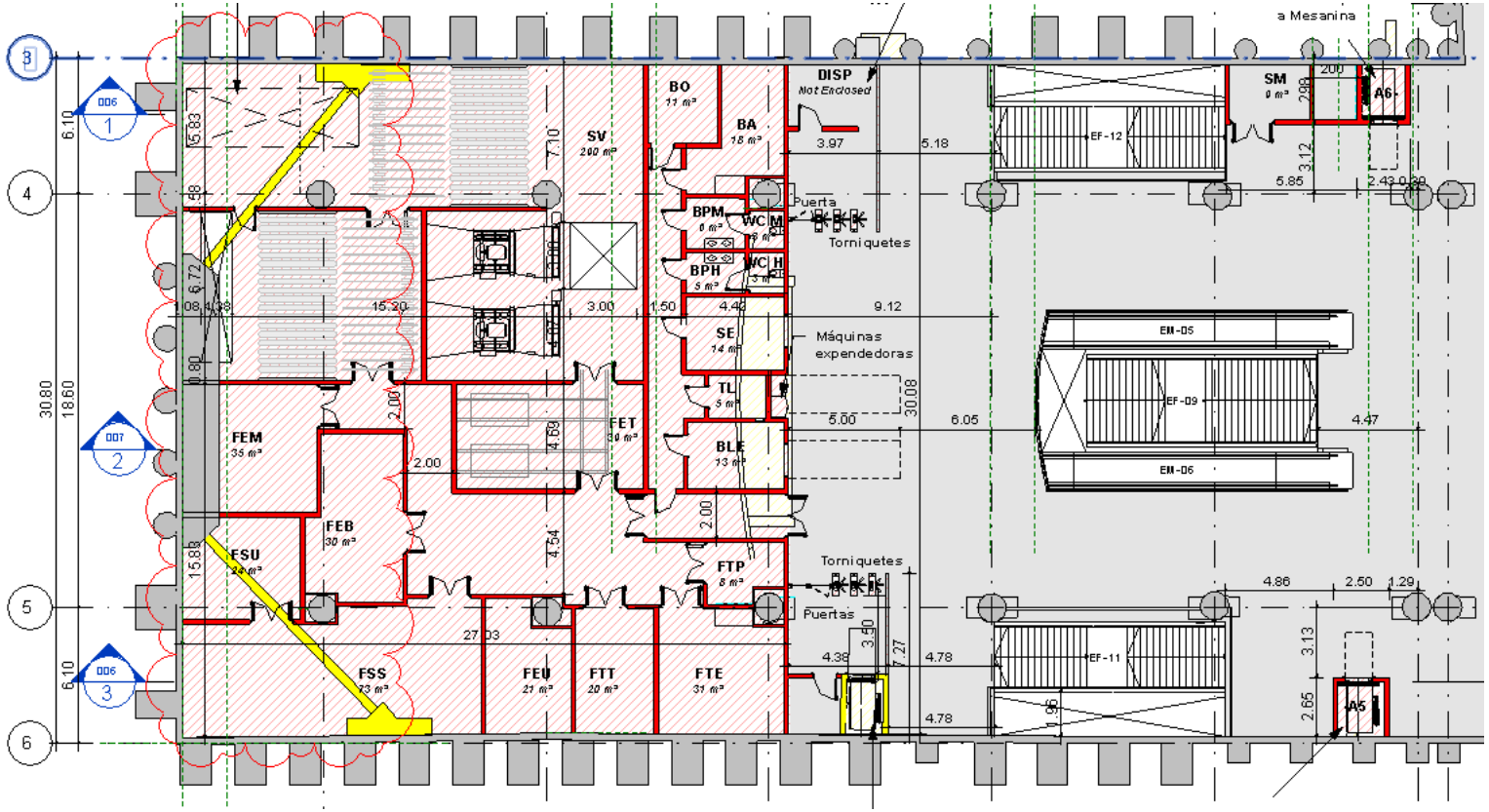
Working with point cloud in Revit



- A laser scan is very helpful and “easy to use” in a Revit model
- The point cloud is managed inside ReCap360 and then imported into Revit
- This process permitted to realize an “as built” model as it is in reality, with a very low error range

Working with phases in Revit

Then it was possible to represent in different colours demolishing and new construction elements



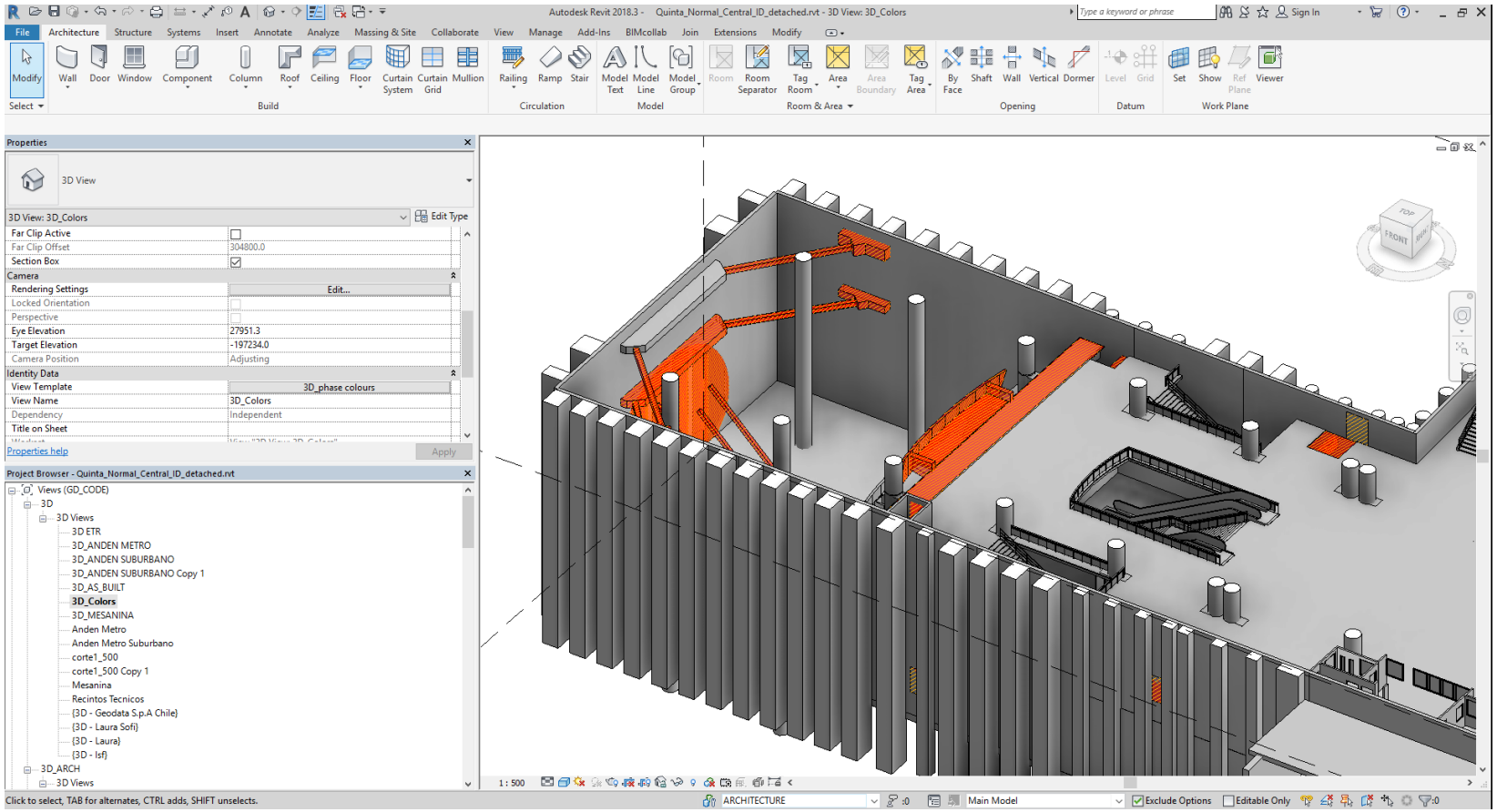
New elements

Demolishing

Existing structure

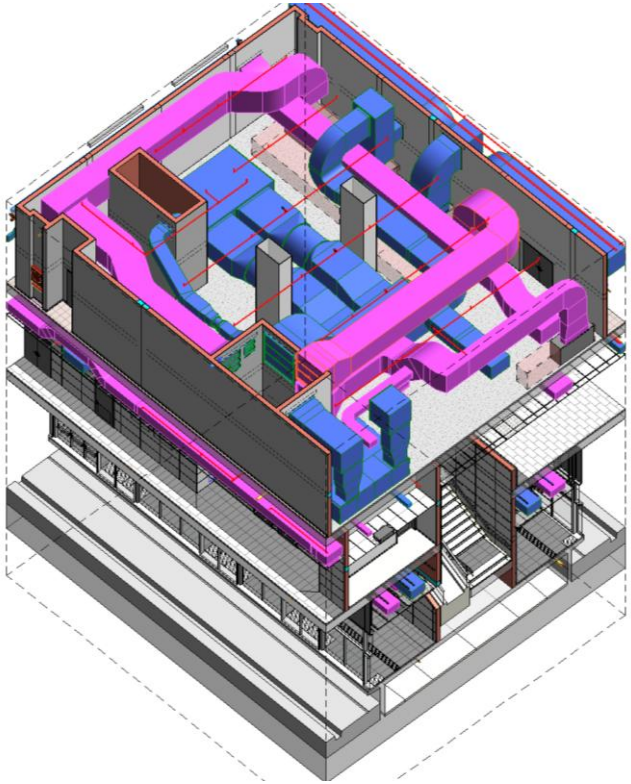
Working with phases in Revit

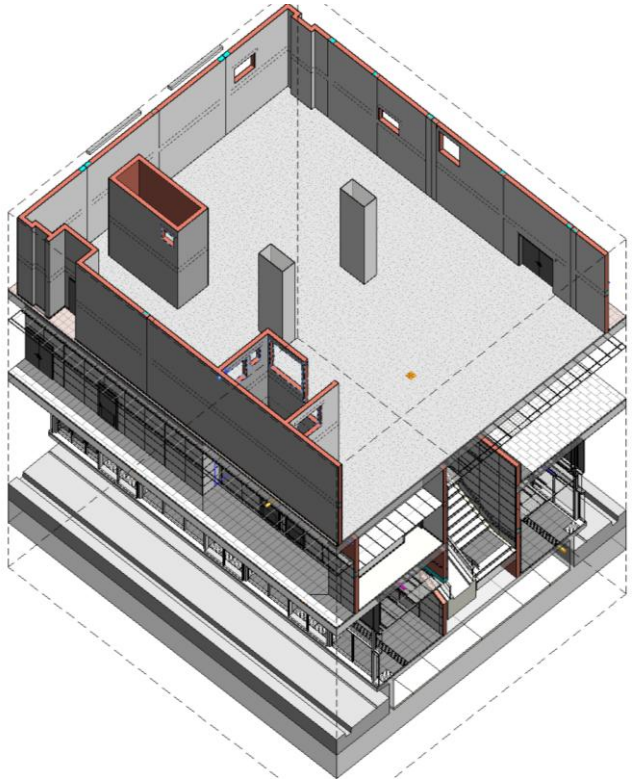
3D view of demolished elements



Revit MEP and clash detection

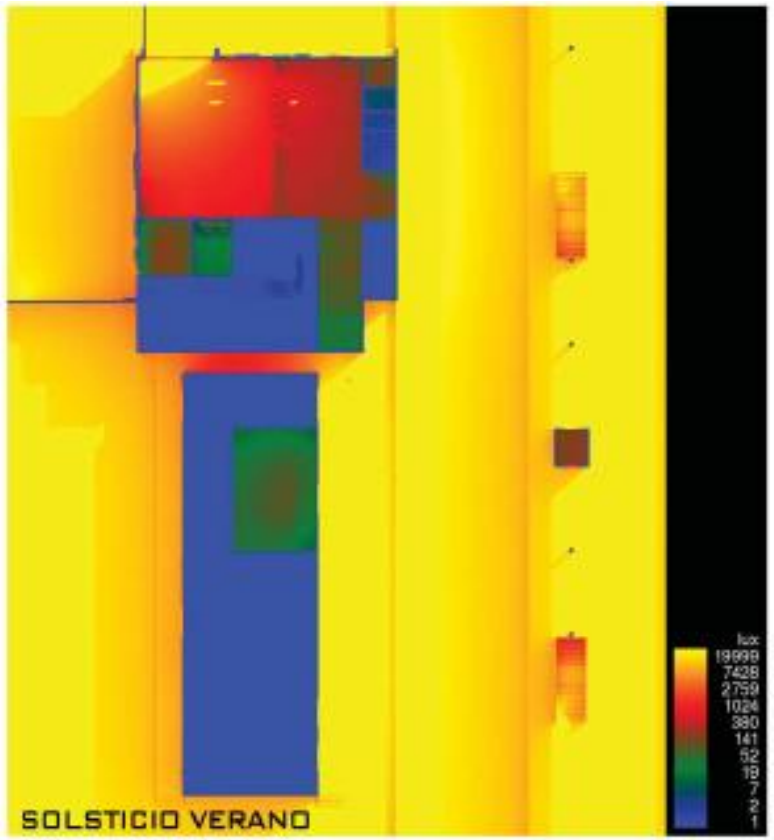
Use of specific Add In for realizing cut openings in the Architectural model. We also use a specific dynamo code to realize cut opening where the Add In does not work properly

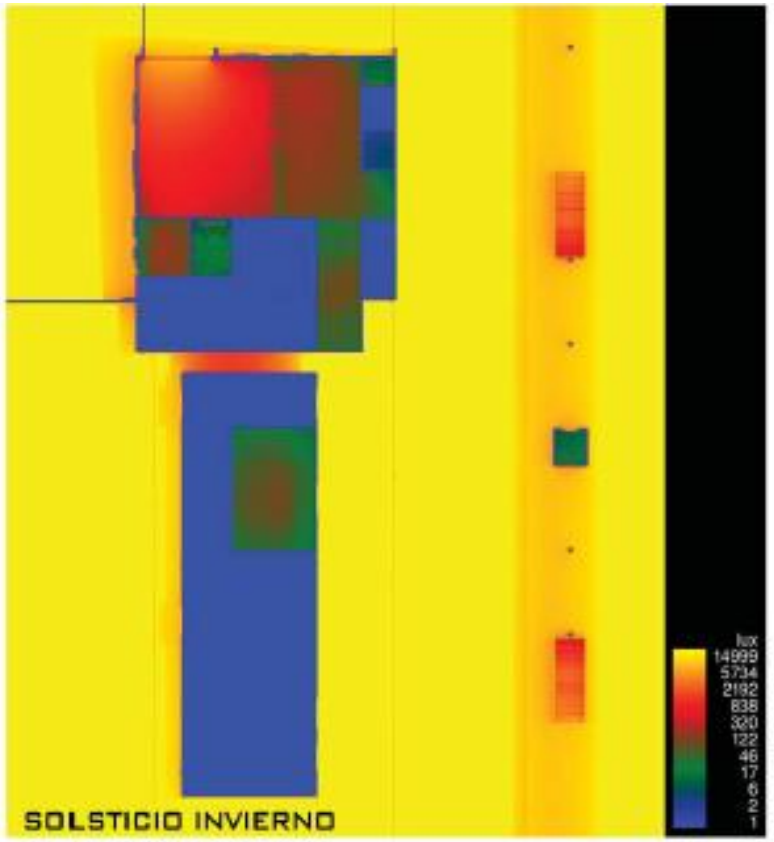




Day Light simulation

The powerful of BIM was exploited also for day light simulation, because the model has got its specific geo-localization



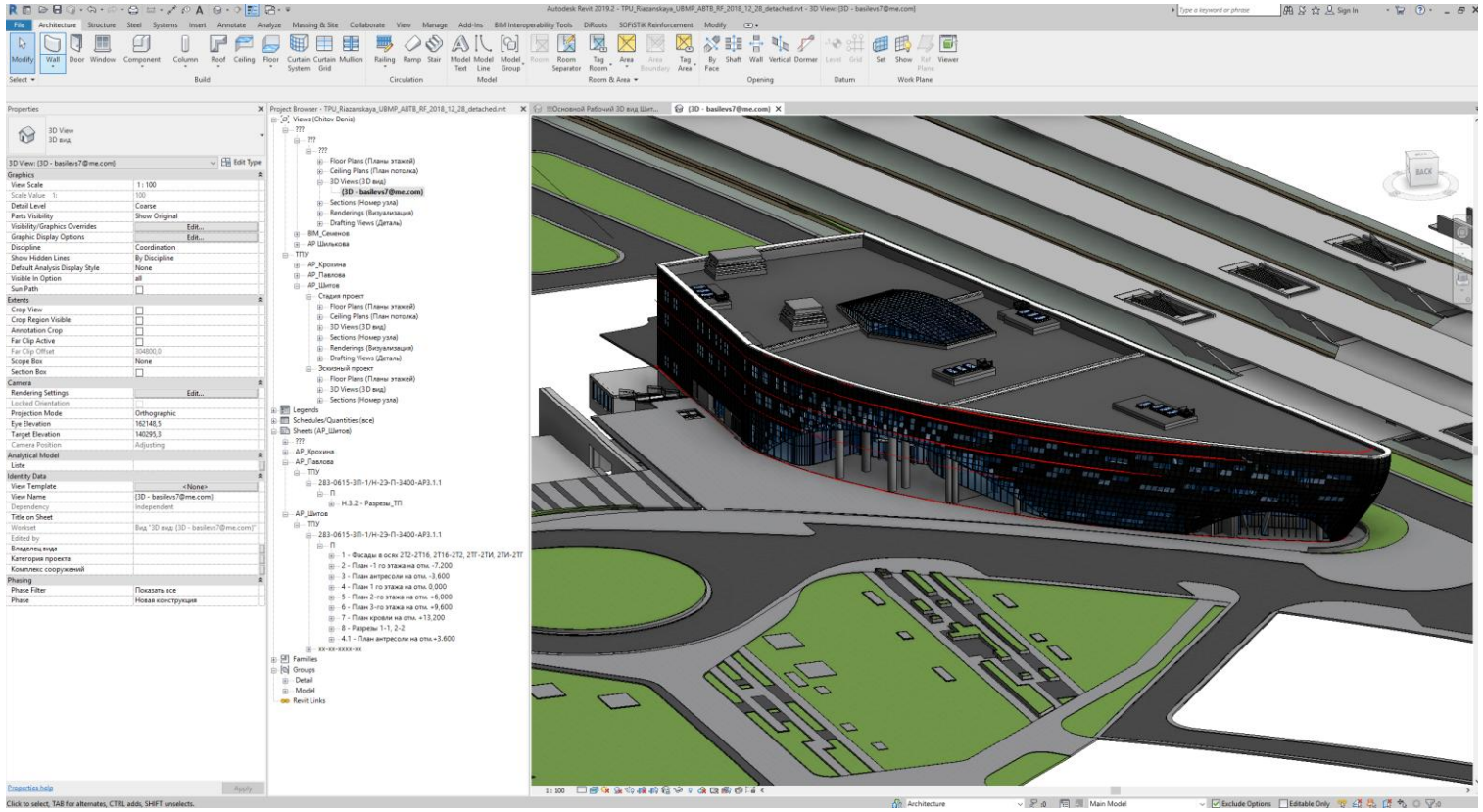


Revit-Robot interoperability

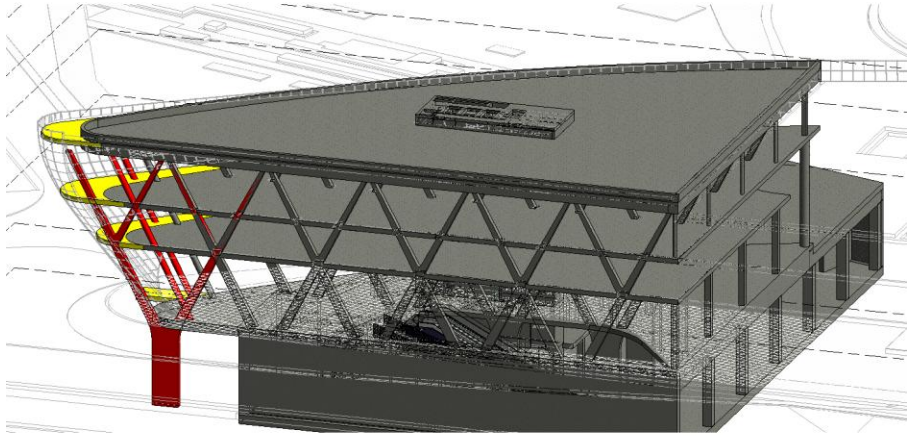
AUTODESK REVIT

ROBOT STRUCTURAL ANALYSIS PROFESSIONAL

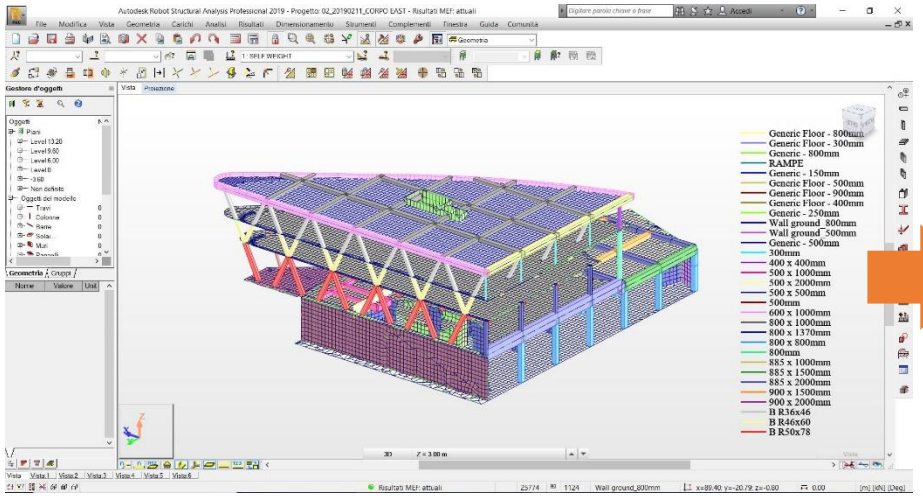
When the model start to be complex, the use of BIM process is a MUST (Architectural, Structural and MEP projects shall be done and managed all inside a unique platform)

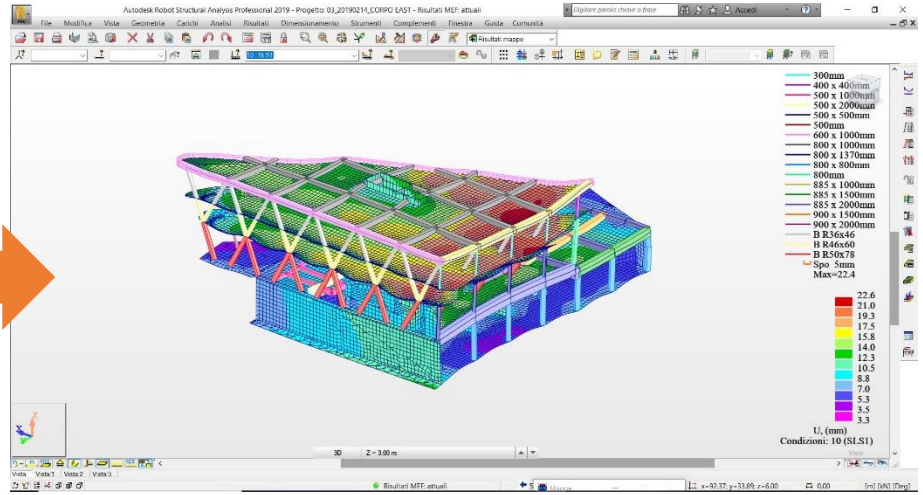


Revit-Robot interoperability



- Structural model has been realized for the entire structure defining loads and load combinations directly inside Revit.
- The Revit model has been imported inside Robot. The analytical model is read by Robot Structural Analysis (with some exception that shall be managed differently) Robot Structural Analysis do the calculation and give results. All updates in future can be managed automatically by the integration between Revit and Robot.

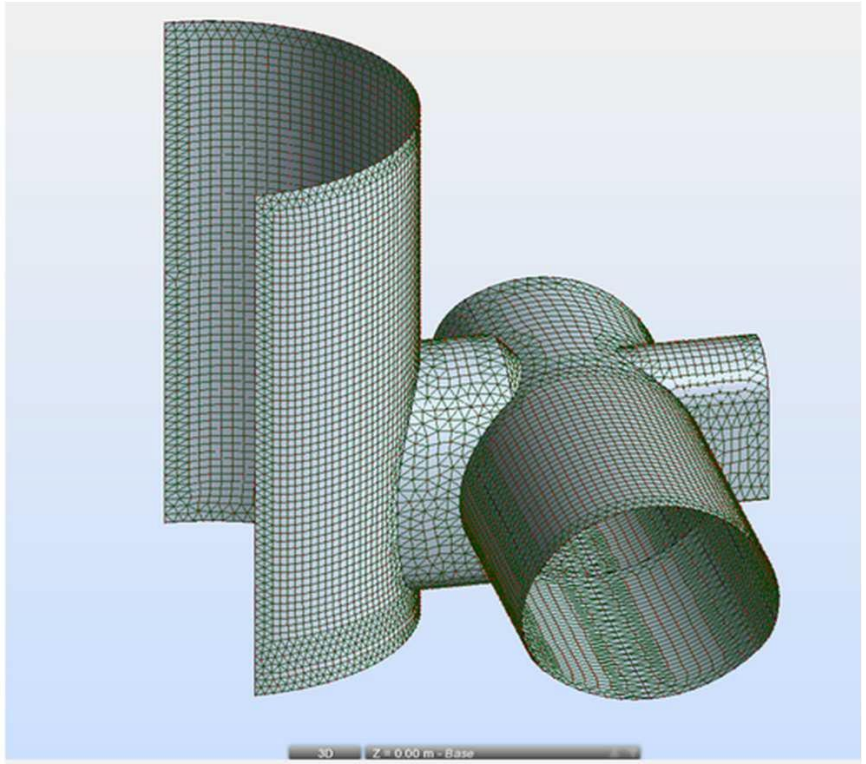


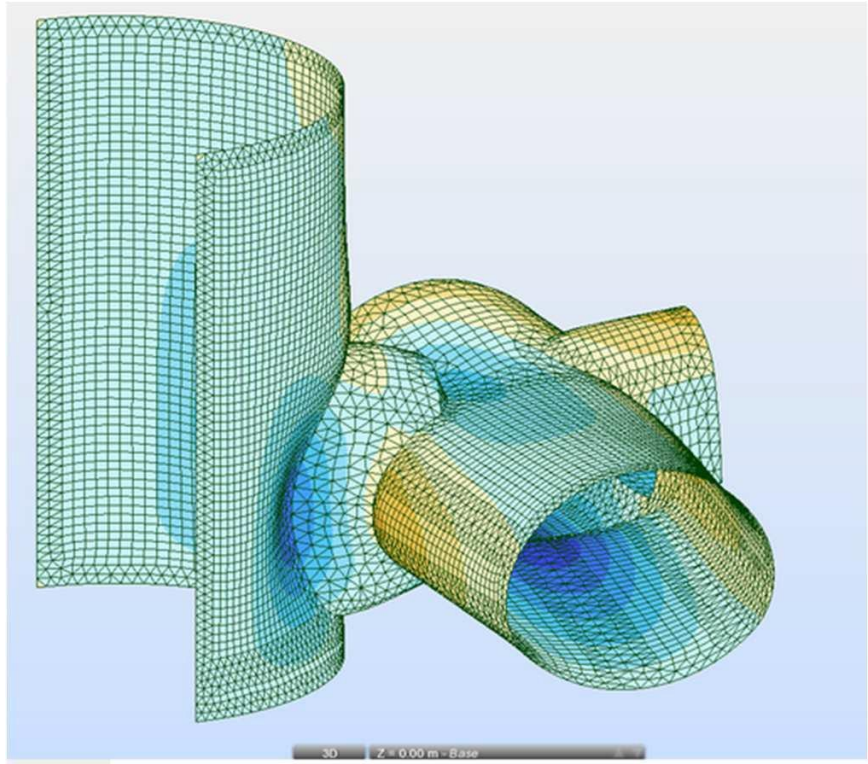


Complex geometries

AUTODESK REVIT

ROBOT STRUCTURAL ANALYSIS PROFESSIONAL

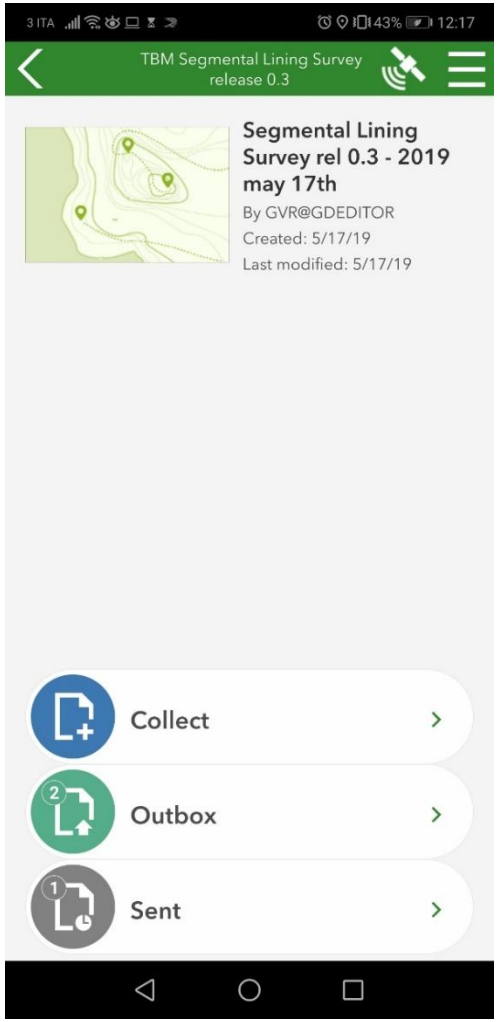


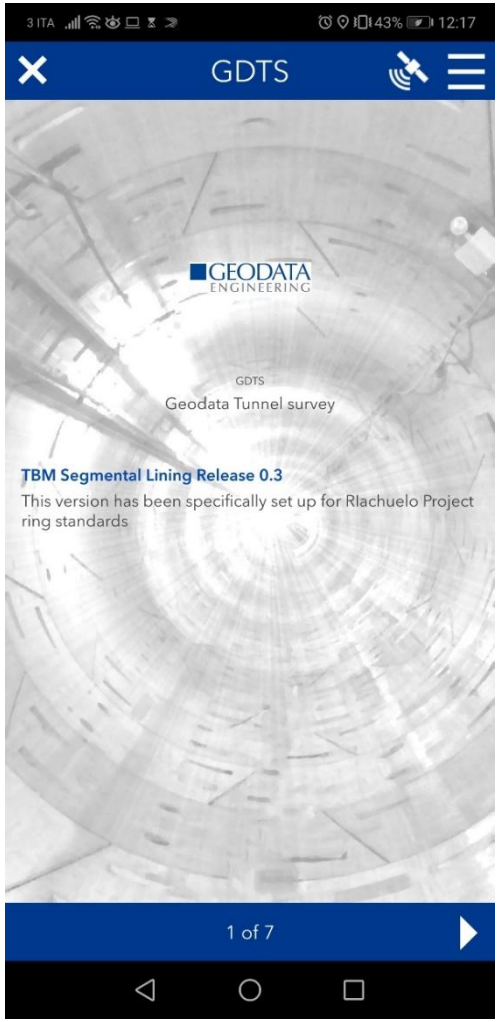


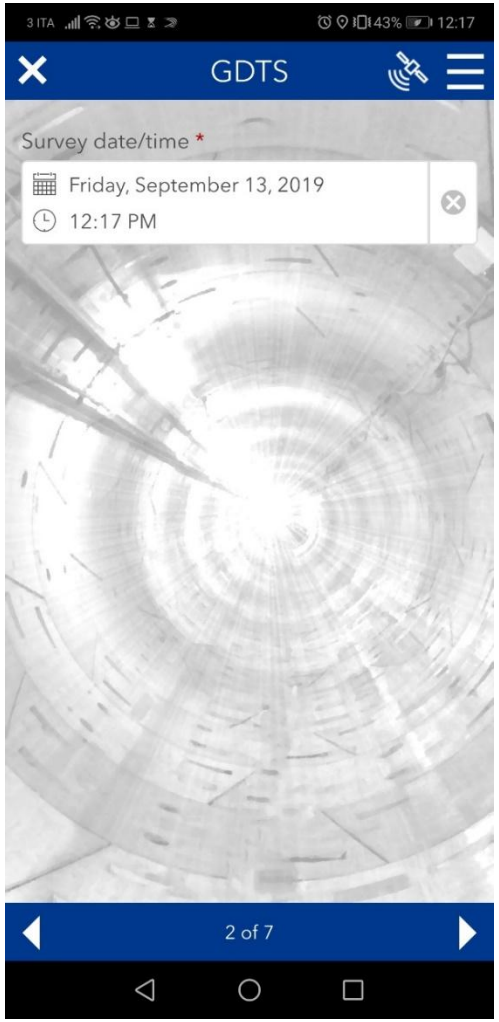
TBM Segmental Lining Survey

Segmental Lining Survey App

Input Data

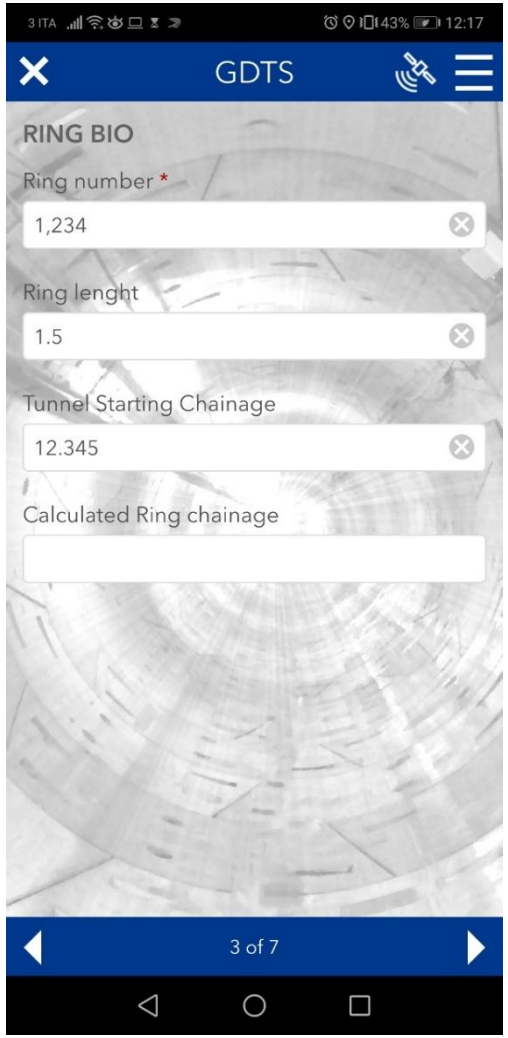


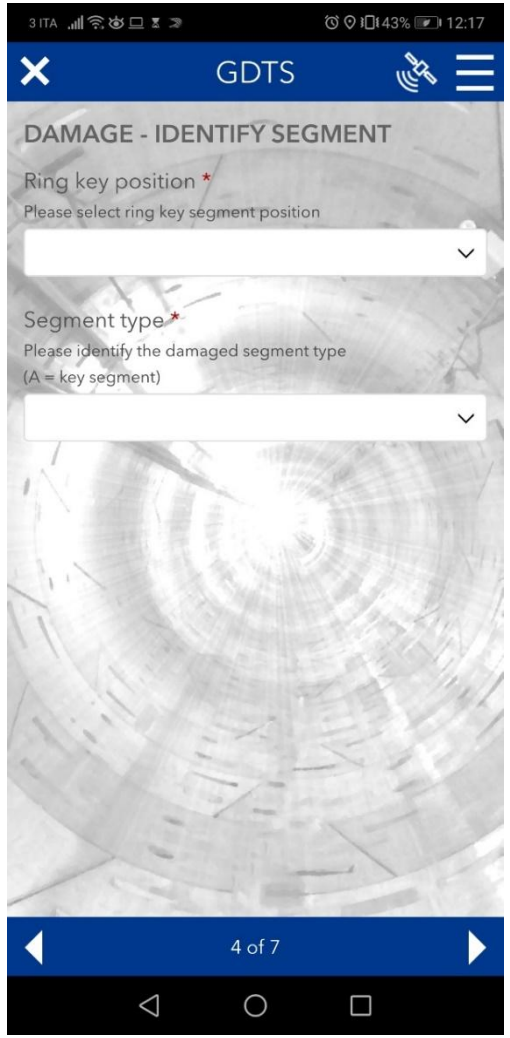


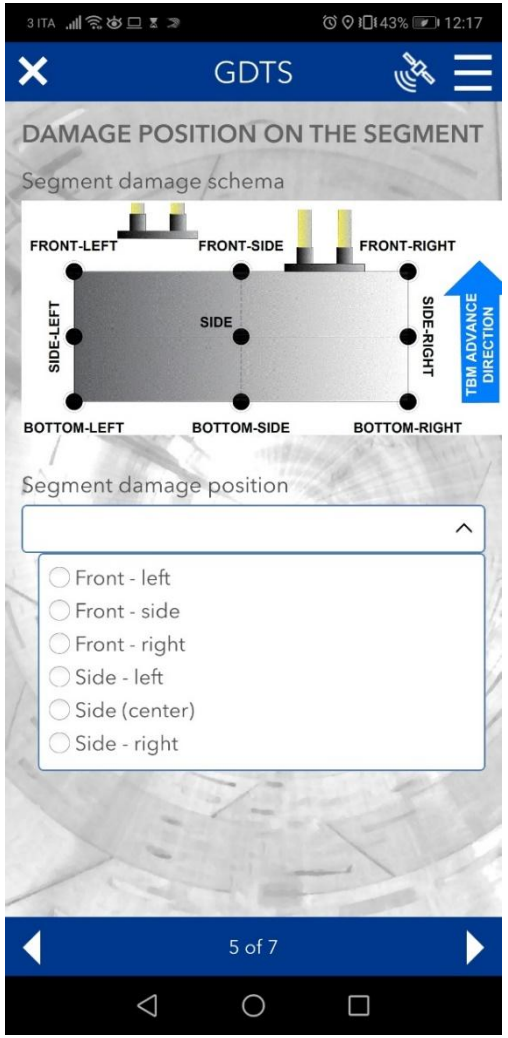


Segmental Lining Survey App

Input Data

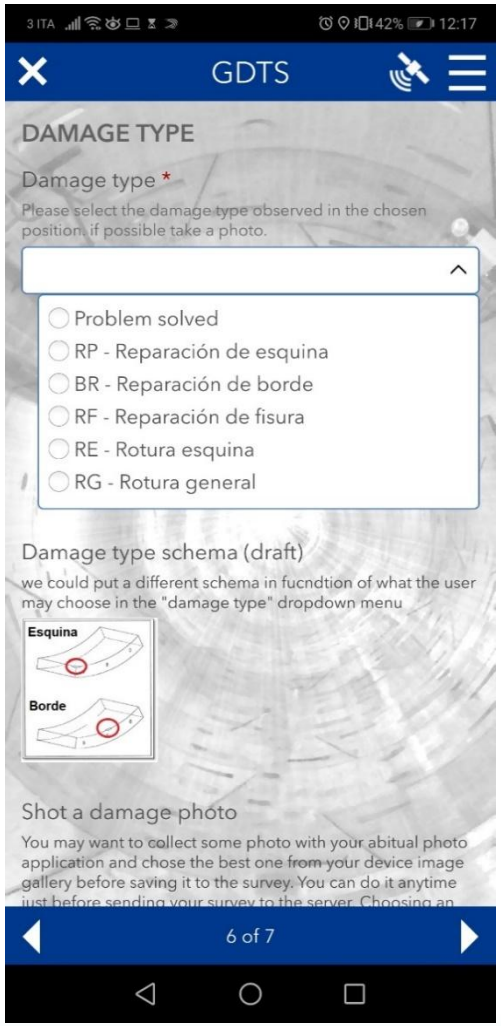


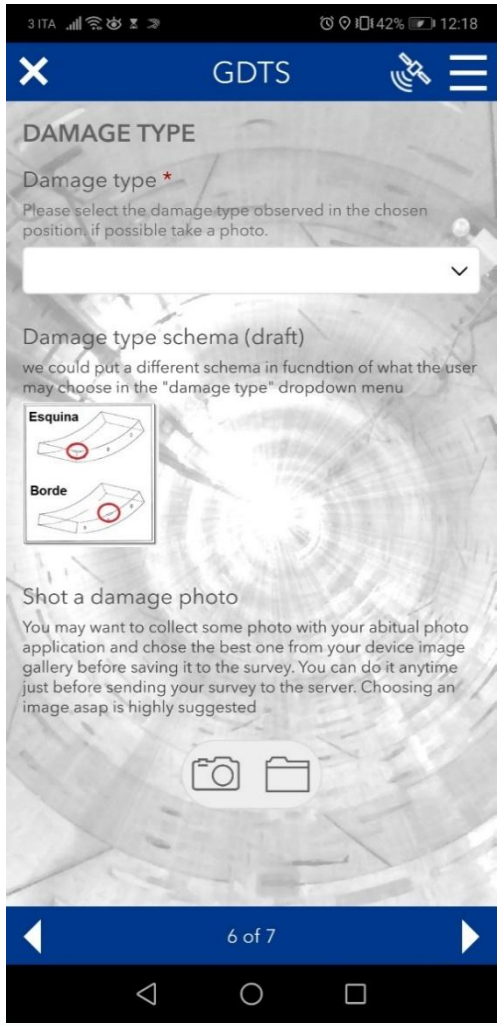




Segmental Lining Survey App

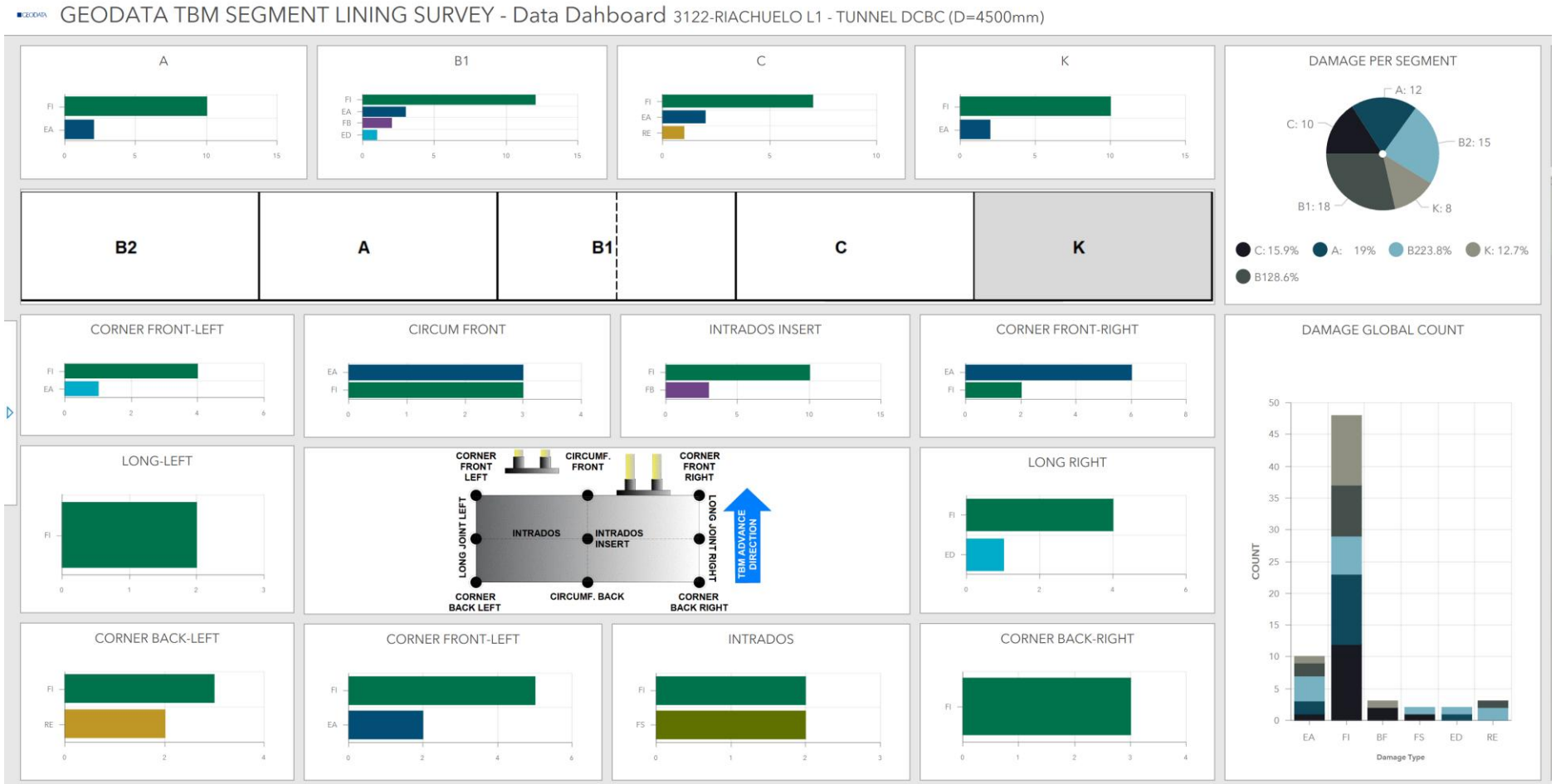
Input Data





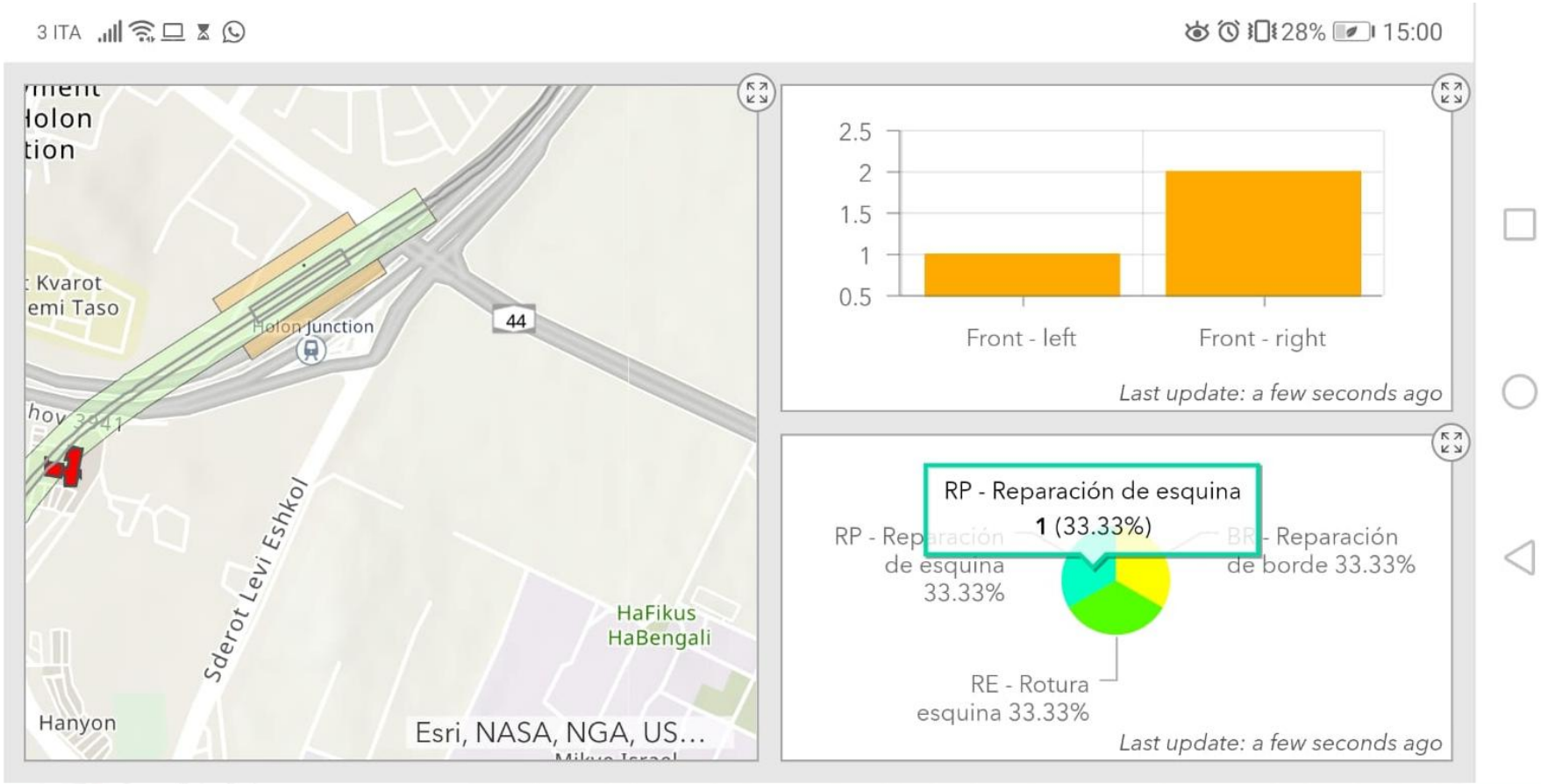
Segmental Lining Survey App
Output

All data are available in a custom dashboard that allows to track the construction process - WEB ACCESS SCREENSHOT



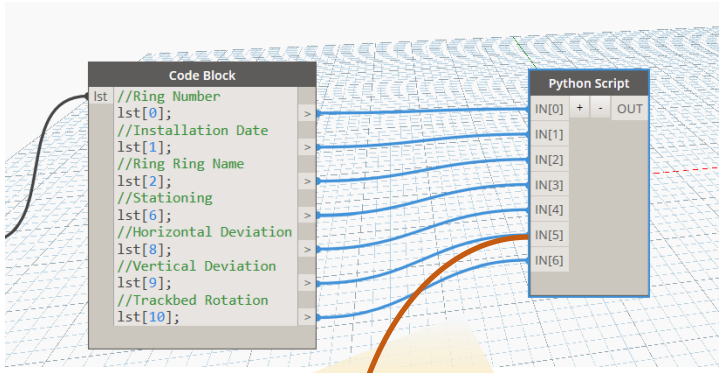
Segmental Lining Survey App
Output

All data are available in a custom dashboard that allows to track the construction process - MOBILE SCREENSHOT



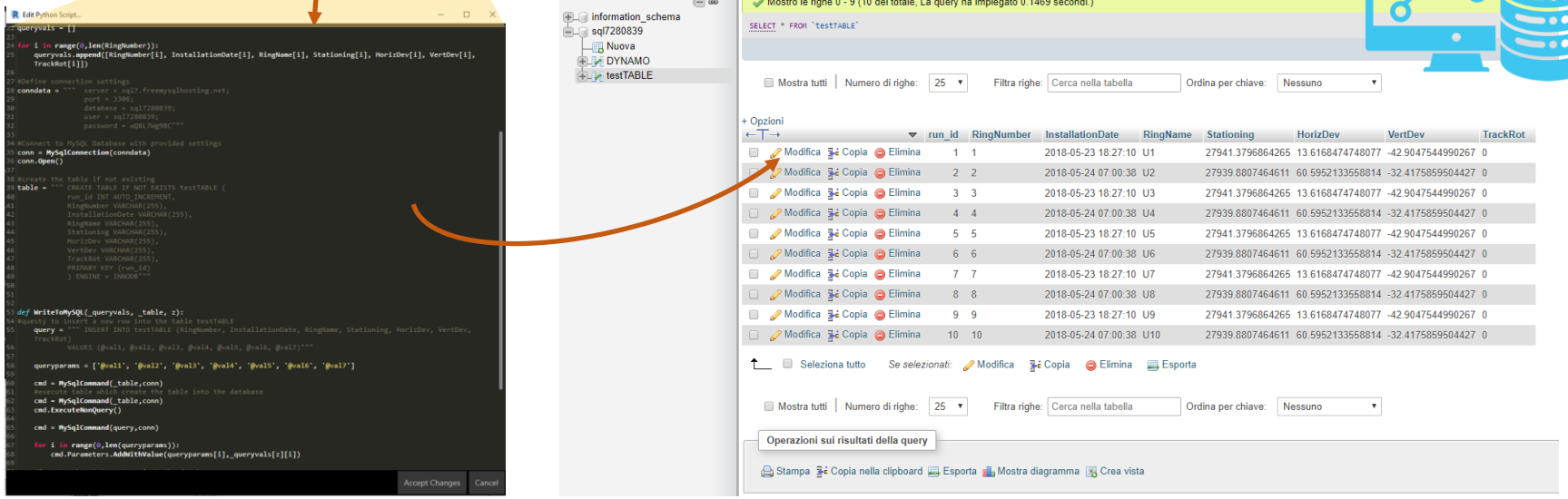
Segmental Lining Survey App

Connection to the DB – the TWIN Model



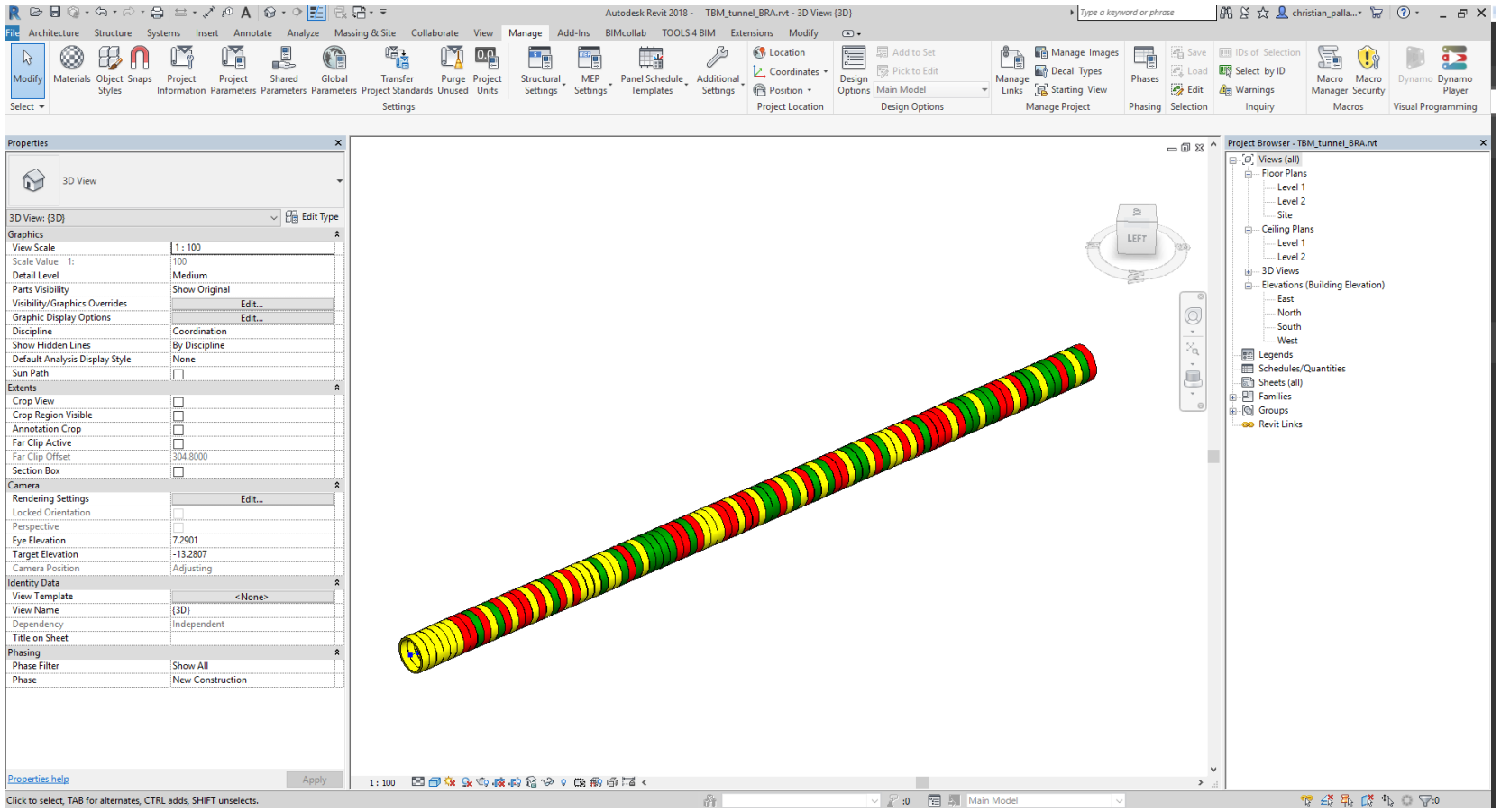
A Dynamo code can be connected to a MySQL database to store data (i.e. about the usage of the code) and read data (i.e. the VMT data)

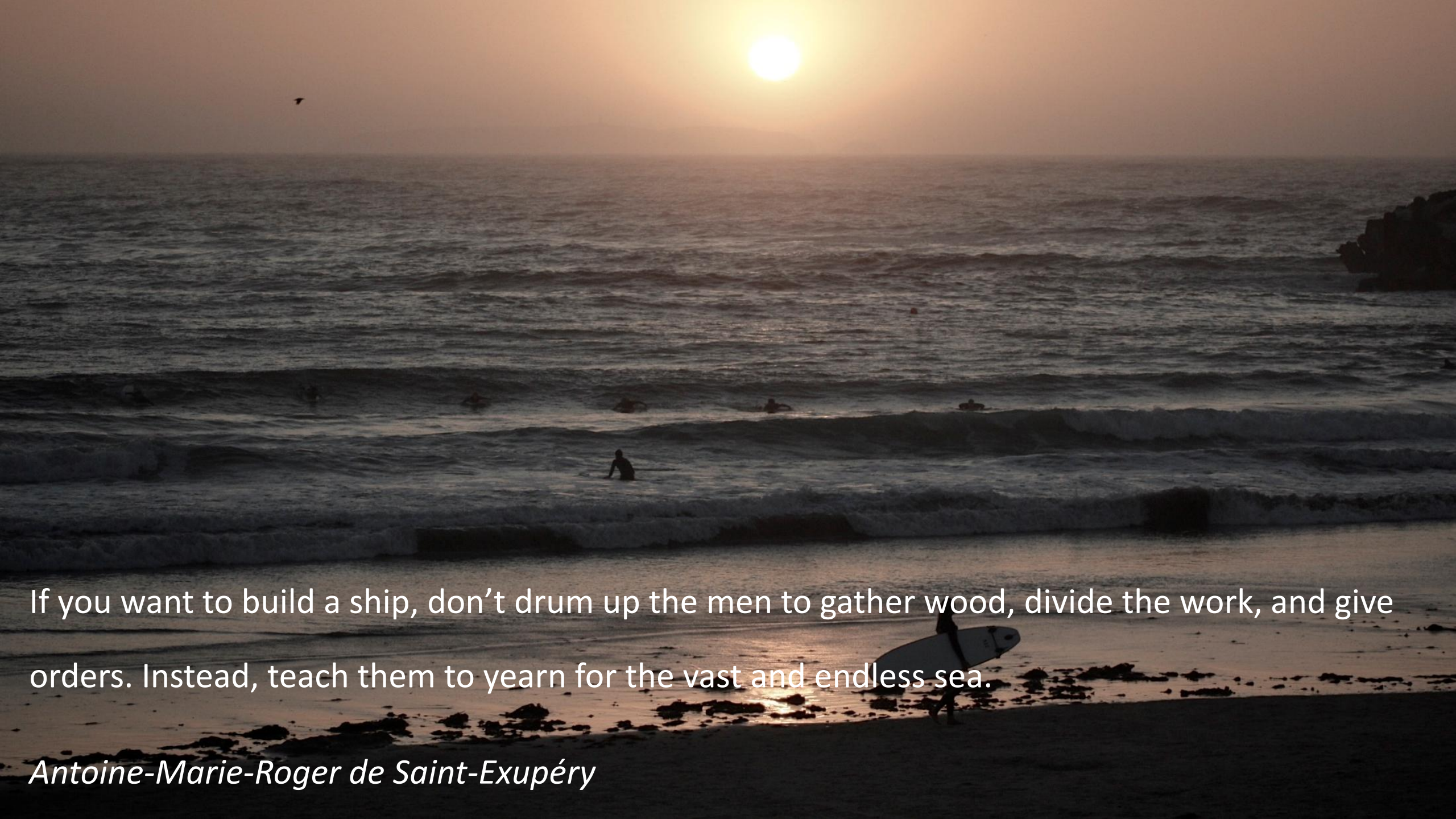
This allow to have a real time updating of the model, realizing a “twin model” representing the reality



Segmental Lining Survey App

Connection to the DB – the TWIN Model





If you want to build a ship, don’t drum up the men to gather wood, divide the work, and give orders. Instead, teach them to yearn for the vast and endless sea.

Antoine-Marie-Roger de Saint-Exupéry