



Esri CityEngine for Developers

Markus Lipp, Ph.D.

**Esri Developer Summit
Middle East & Africa**

19-21 November 2013
Park Hyatt Dubai



CityEngine

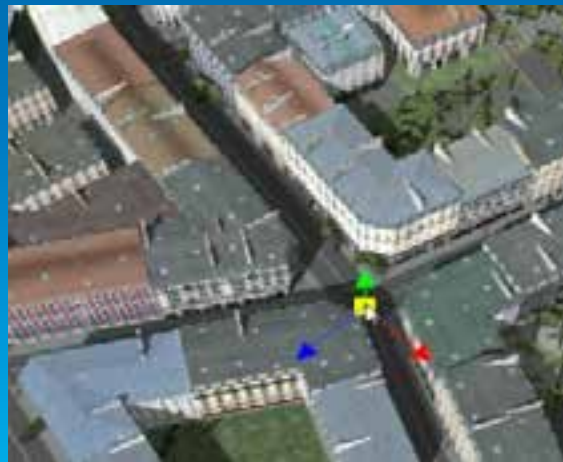
<http://www.esri.com/software/cityengine>

3D procedural modeling and design solution

- **Procedurally generate 3D urban content**
 - From 2D GIS geometry and attributes
 - Using algorithms and parametric rules



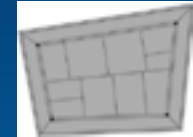
- **3D City Design**
 - Iterative design
 - Real-time feedback
 - Street sketching



Procedural modeling

3D model creation using rules / algorithms

- Base geometry



- Procedural rules



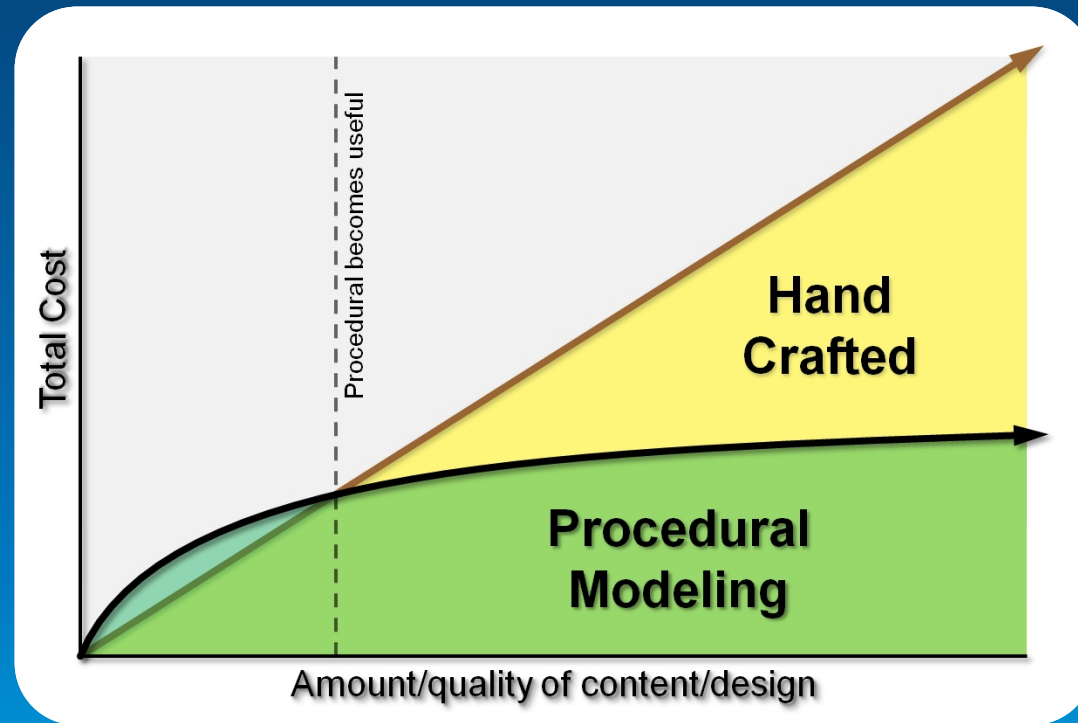
Base geometry



Final 3D model

Iteratively refine a design by creating more and more detail

Procedural Modeling vs. Manual Modeling



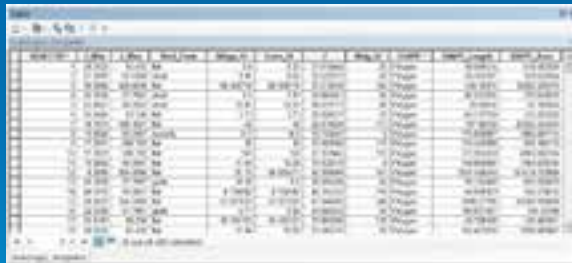
Time reduction / cost saving

GIS Data as Input

ArcGIS example



Geometry (parcels, footprints, streets)

A screenshot of a GIS attribute table showing various data fields. The table has multiple columns with headers and rows of data. The data appears to be related to the geometry shown in the map above, such as parcel area, street width, and building height.

Attributes (height, roof type, street width)

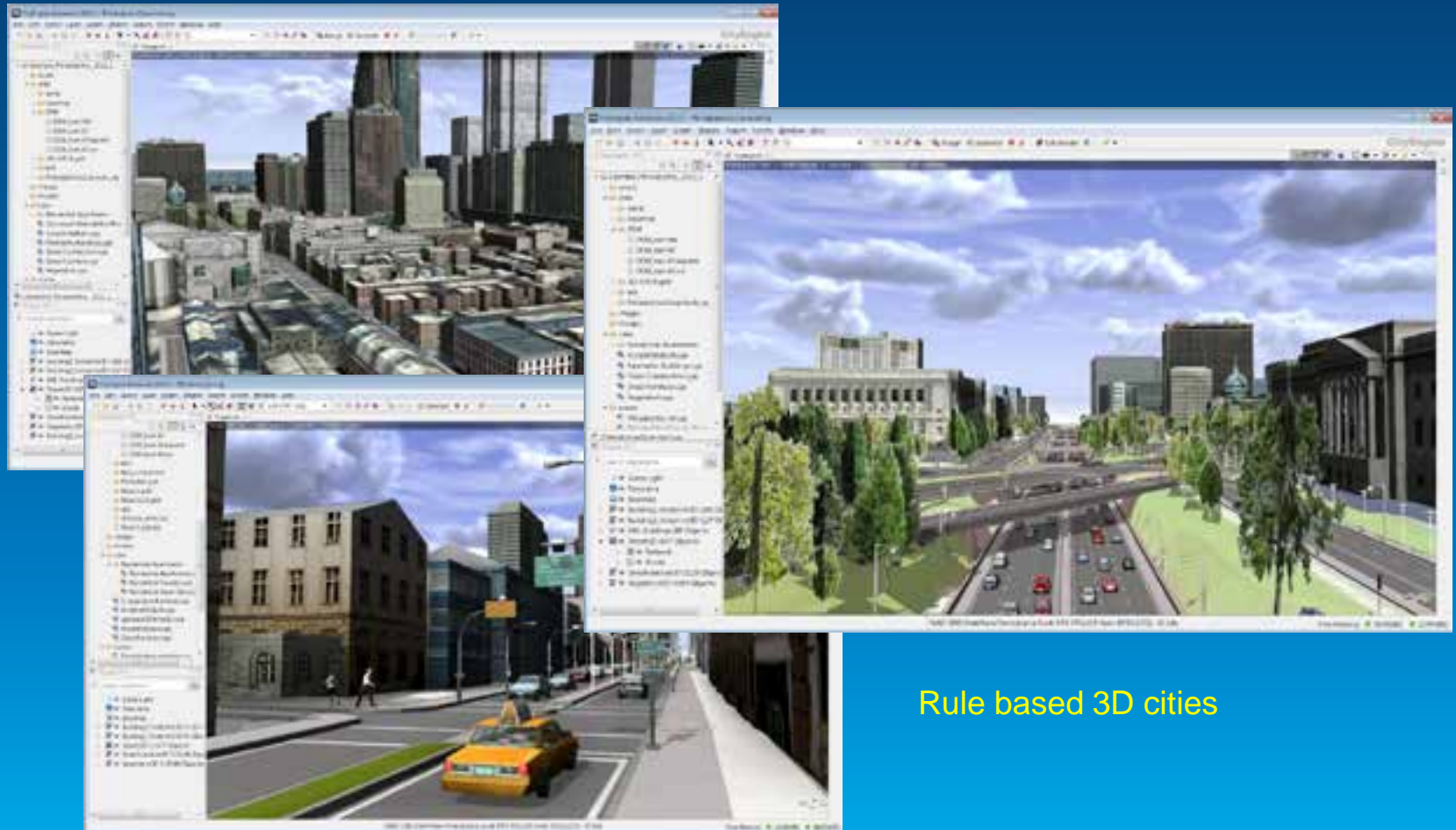


Rules



3D city content from GIS data

procedural city modeling



Rule based 3D cities

3D City Design – Procedural Approach

Urban planning example



Add a floor



Add a roof

New development – draw streets

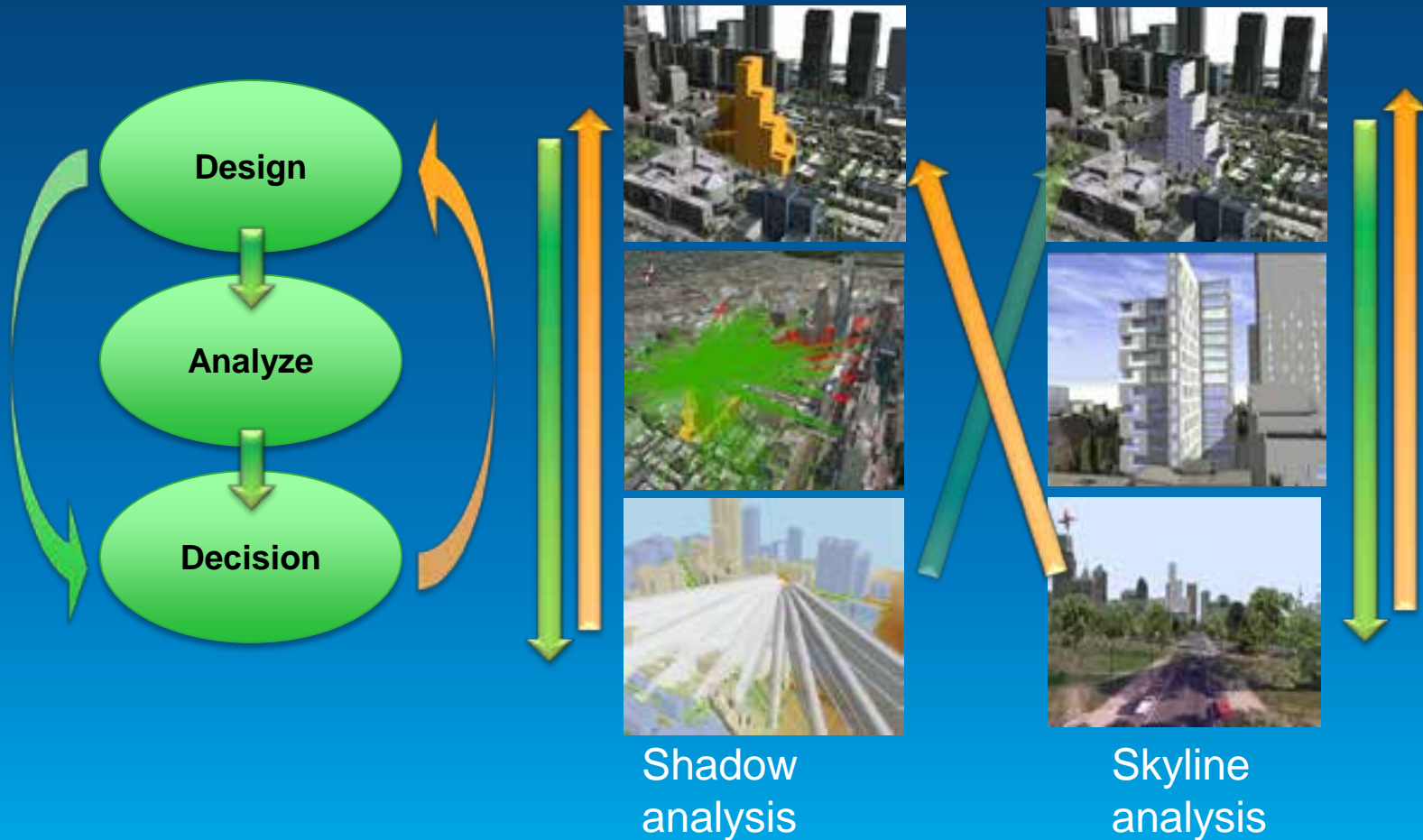


Reporting (area ratios...)



3D City (Geo)design

Iterative analysis while designing





Navigator

- models
- rules
 - 3dcimZoningEnvelopes.cga
 - Esri Vegetation Library with Lurr
 - Esri Vegetation Library with Lurr
 - Facade Textures.cga
 - International City.cga
 - International City1.cga
 - Roof Textures.cga
 - Street Construction Simple.cga

Intern...



***Scene**

search expression

- Scene Light
- Panorama
- Heightmap
- Obstacle
- Streetnetwork (2126 Objects)



Inspector

Shapes (115) Blocks (10)

Name Shape

Shape Parameters

Block Parameters

shapeCreation	true	Off
type	?	Recurs
lotWidthMin	?	
simplify	0.25	
cornerAlignme...	Street width	Street s
lotAreaMin	150	
irregularity	0	
shallowLotFrac	1.5	
alignment	Uneven	Uneven
seed	?	

Street Parameters

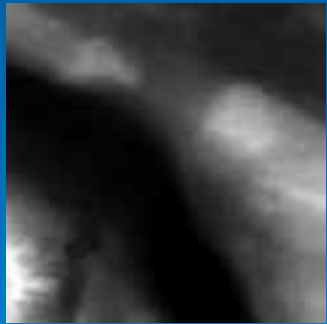
shapeCreation	true	Off
streetWidth	?	
streetOffset	0	
sidewalkWidth...	?	
sidewalkWidth...	?	
precision	0.5	
laneWidth	3.5	

Block Parameters

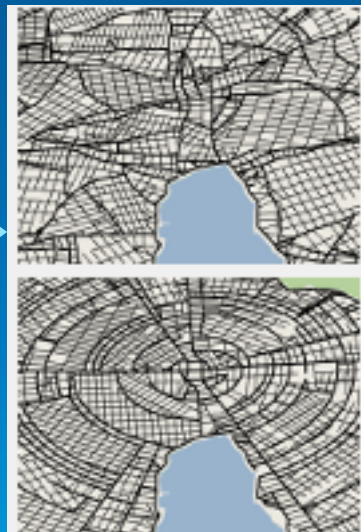
shapeCreation	true	Off
type	?	Recurs
lotAreaMin	500	
lotAreaMax	1500	

Procedural Workflow

- Transforming parameters and maps to 3D cities
- Based on street growing algorithms and rules
- I/O with ArcGIS at every step possible



Height map



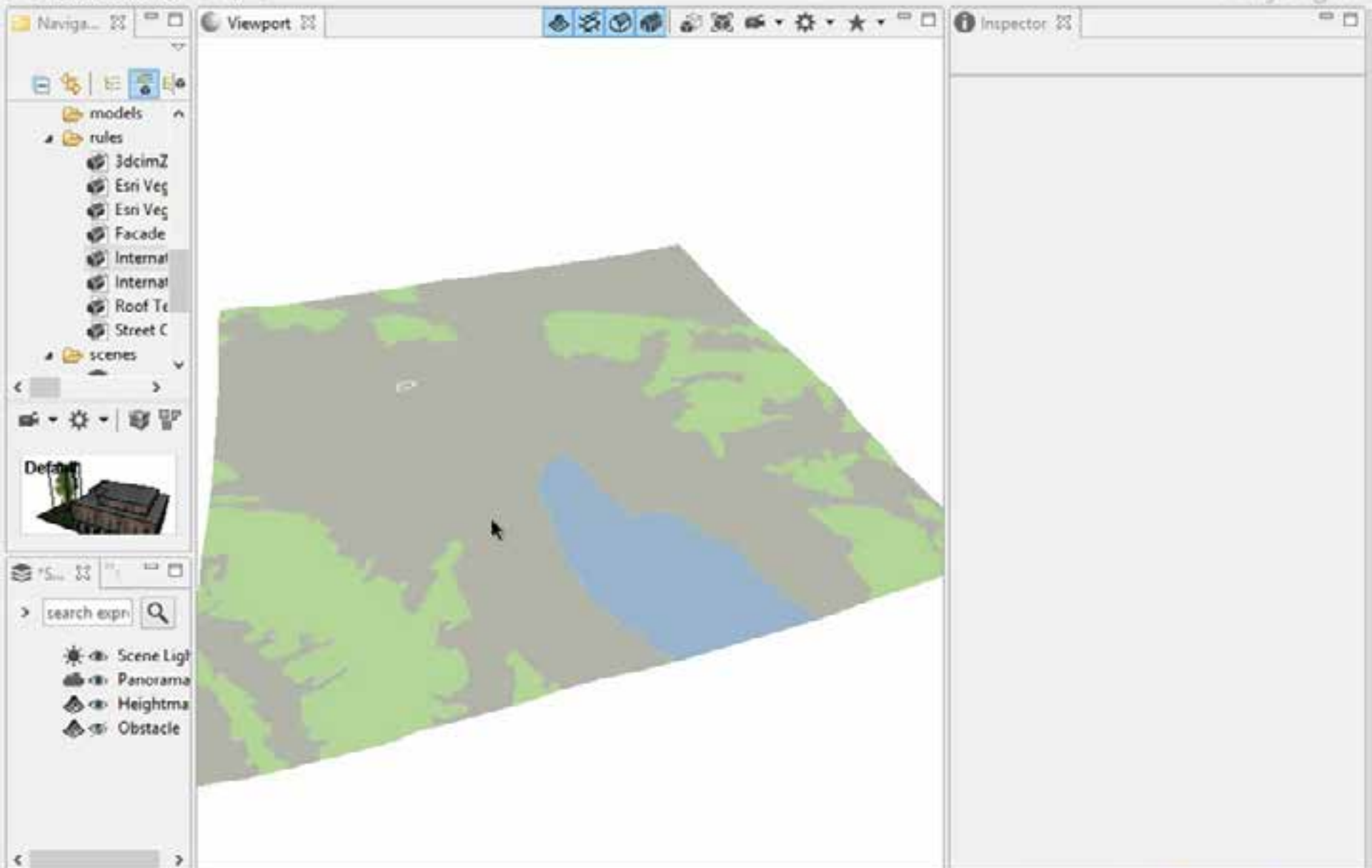
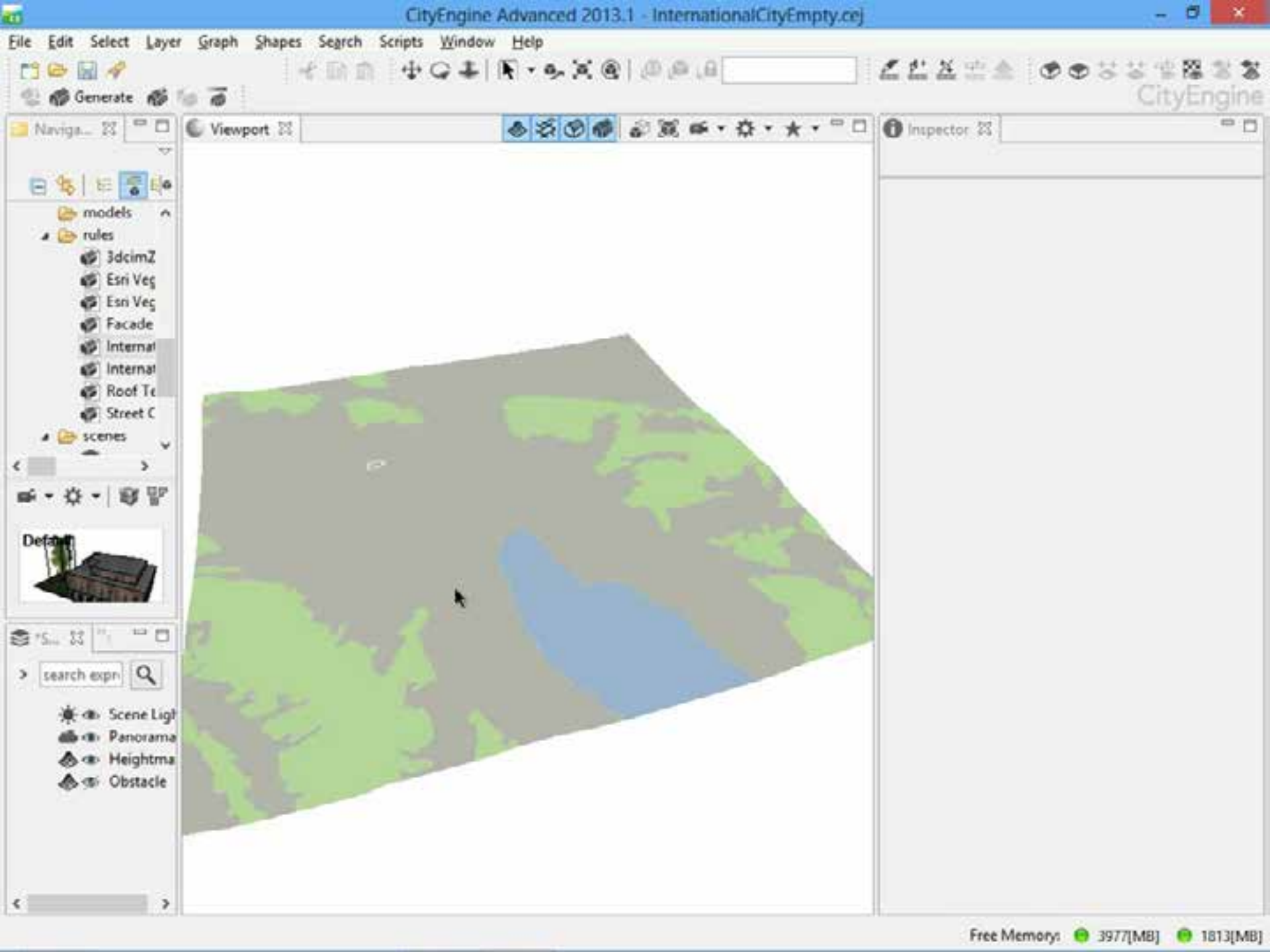
Street network



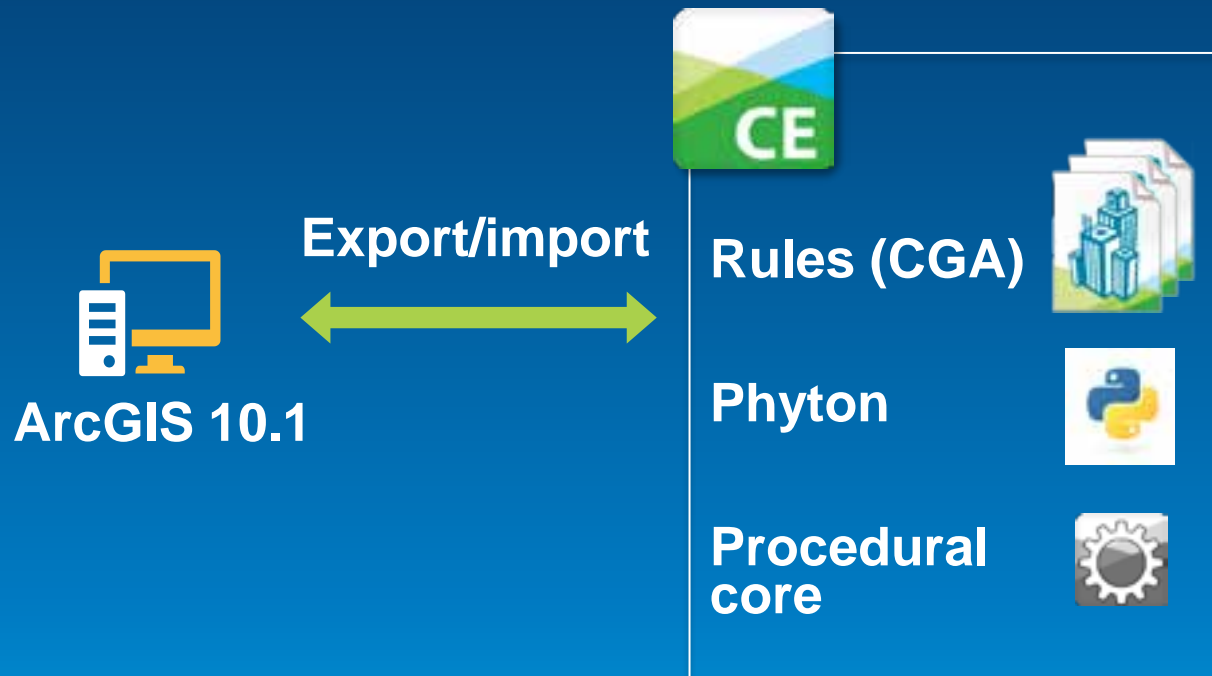
Blocks and parcels



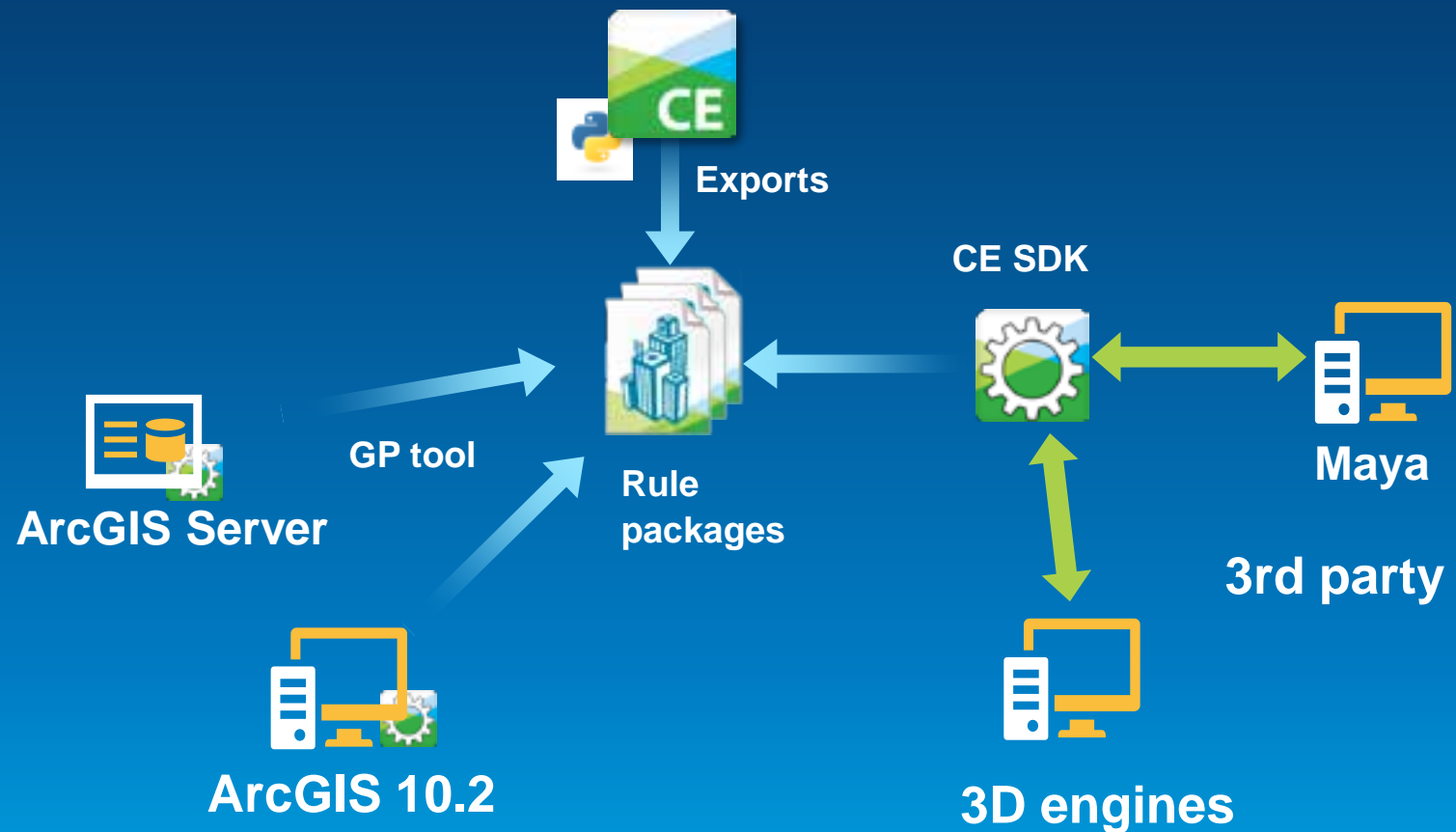
3D geometry



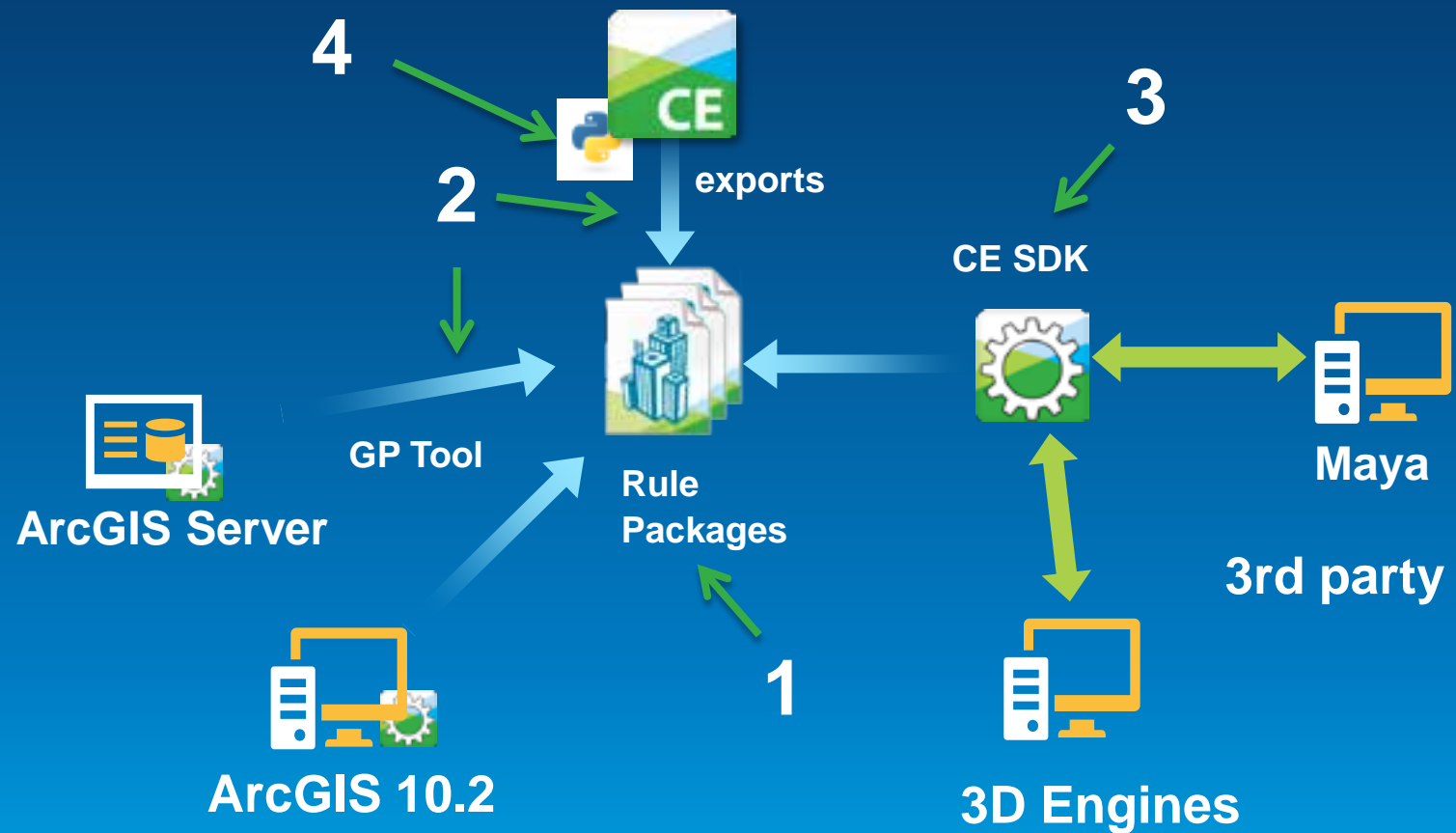
CityEngine 2012 – Opportunities for Developers



CityEngine 2013 for Developers



CityEngine 2013 for Developers



1. Rules, Rule Packages, CGA

- Rule: description of shape refinement



- Rule Package: multiple rules & assets

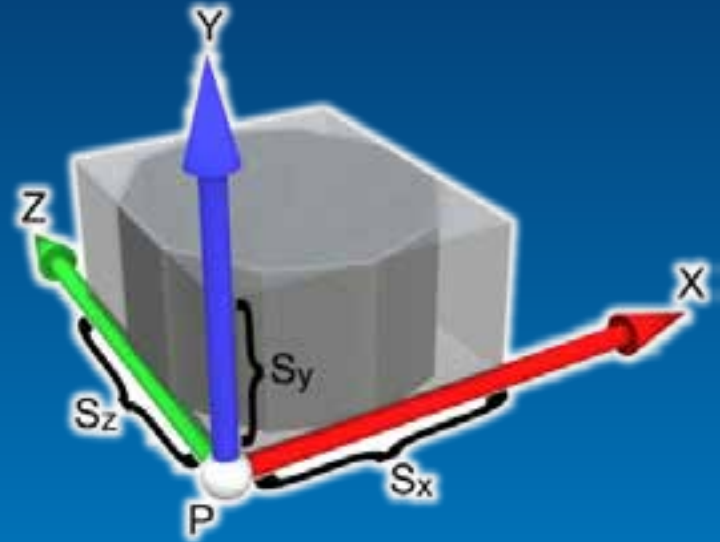


- CGA: «scripting language for shapes»

```
Mass (h, roofType) -->  
    extrude (h*HeightFactor) Stories comp (f) {top: Roof (roofType) }
```

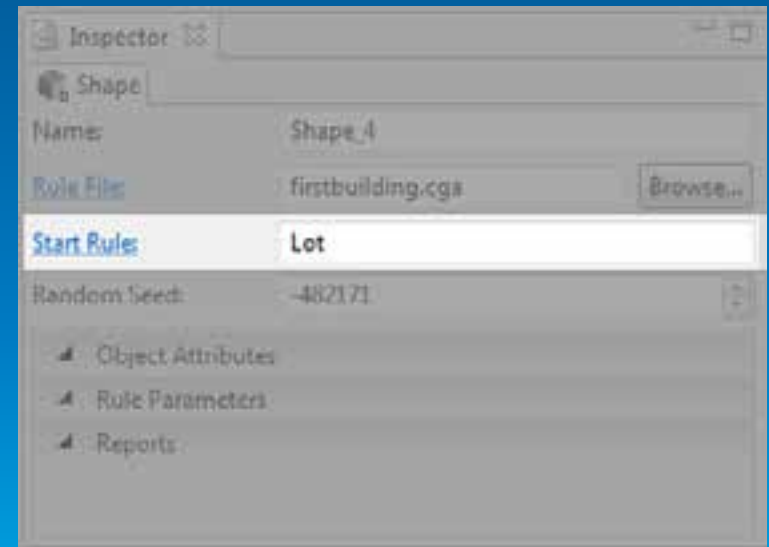
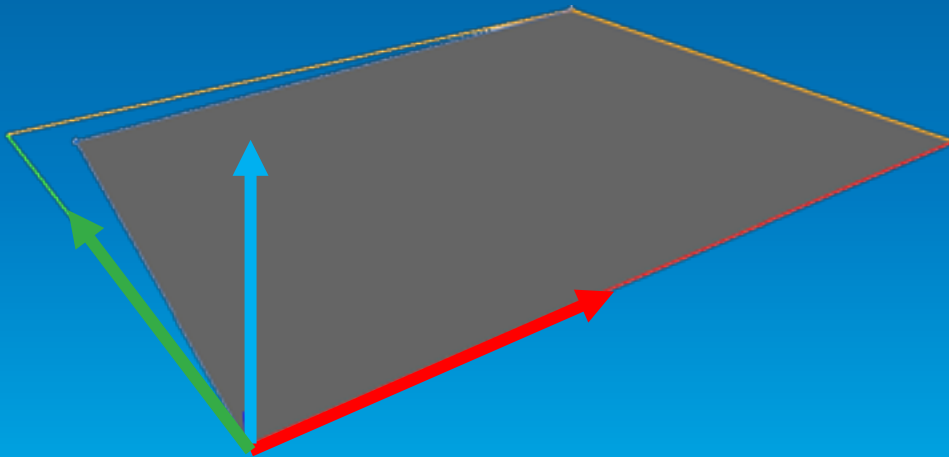
CGA Shape Grammar - Definition

- A *shape* consists of:
 - Symbol
 - Attributes
 - Geometry (polygonal mesh)
 - Oriented bounding box called *scope* (numeric attributes)
- Initial shape: *axiom*
- A *rule* describes the transformation of a shape into one or more successor shapes



GIS Lot as Initial Shape

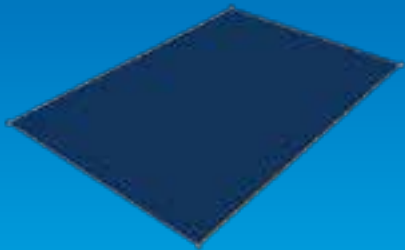
- Symbol = start rule
- Attributes: height, zoning...
- Geometry = only one face
- Scope oriented on first edge



Rule Example

Lot --> extrude(10) Mass

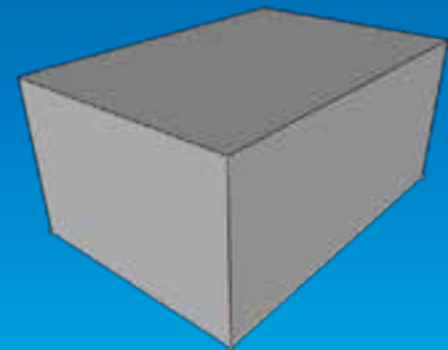
- *Lot* and *Mass* are shapes
- A modified copy of shape *Lot* becomes shape *Mass*
- *Mass* is called a leaf shape
- Output geometry = all leaf shapes



Lot with shape symbol **Lot**



Rule application (generation)



Resulting shape **Mass**
Displayed geometry

Multiple rules

Lot --> extrude(10) Mass

Rule #1

Mass --> C D

Rule #2

- Rule #2 is a matching rule for shape Mass
- Shape Mass is replaced by shapes C and D
- Mass NOT leaf shape here

CGA Syntax Example

```
attr height = 20
const heightG = 8.5
Lot --> extrude(height) Mass
Mass --> comp(f) { top : Roof.
           | front : Frontfacade
           | side : Facade }
Facade -->
  split(y){heightG: Groundfloor
    / ~1 : UpperFloors}
Groundfloor -->
  case scope.sx > 10 :
    color("#cccccc")
  else : color("#ffcccc")
```

- Rules (may have parameters)
Lot, Mass, ...
- User-defined attributes and constants: height, heightG
- Boolean, float and string expressions
20, 8.5, ("#cccccc"),
scope.sx > 10
- CGA-specific keywords
attr, top, front, case
- CGA operations (may have parameters)
extrude(height),
comp(f)

CGA operations overview

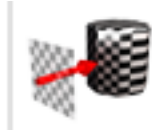
Geometry creation



Geometry subdivision



Texturing



Transformations



User Interface in CityEngine

- Example building rule file



The screenshot displays the CityEngine interface with the following components:

- Rule File Editor:** Shows the code for a rule named 'Lot'. The code defines attributes for windowWidth, windowHeight, and floorheight, and uses the 'extrude' function to create the building's facade and floors.
- 3D View:** A small 3D preview of the building model is visible in the background.
- Inspector Panel:** A panel on the right side of the interface showing the current object's attributes and rule parameters. It includes a 'Shape' section, a 'Rule Parameters' table, and a 'Reports' section.

Rule Parameters Table:

Name	Source	Value
floorheight	Rule	4
height	Rule	18.9770...
tilewidth	Rule	3
windowHeight	Rule	2.800000
windowWidth	Rule	2.200000

2. Exporting and Using Rule Packages

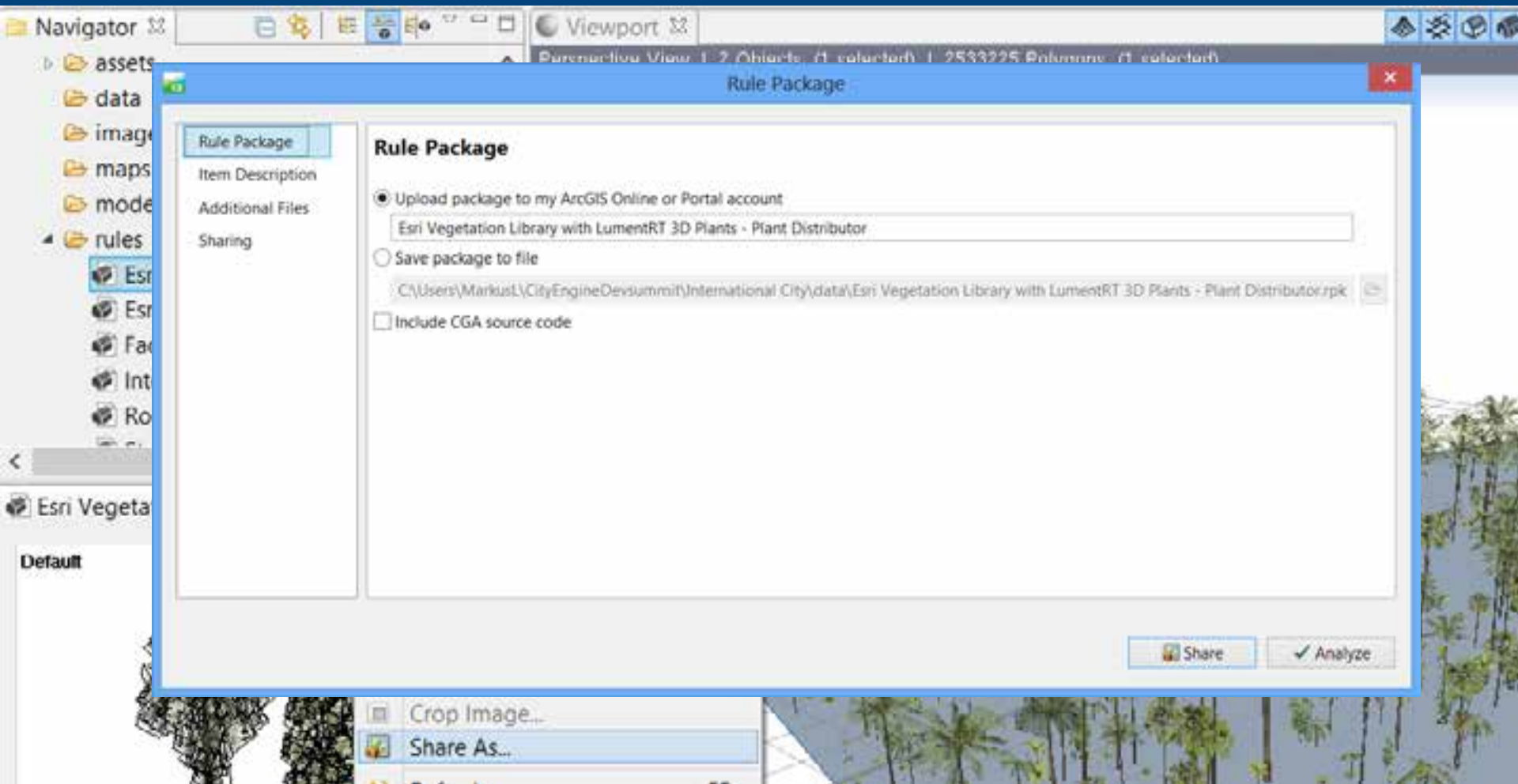
Recap: Rule package is:

- Combination of CGA rules with assets
 - Textures, meshes
- Author in CityEngine, used in GP Tools or SDK

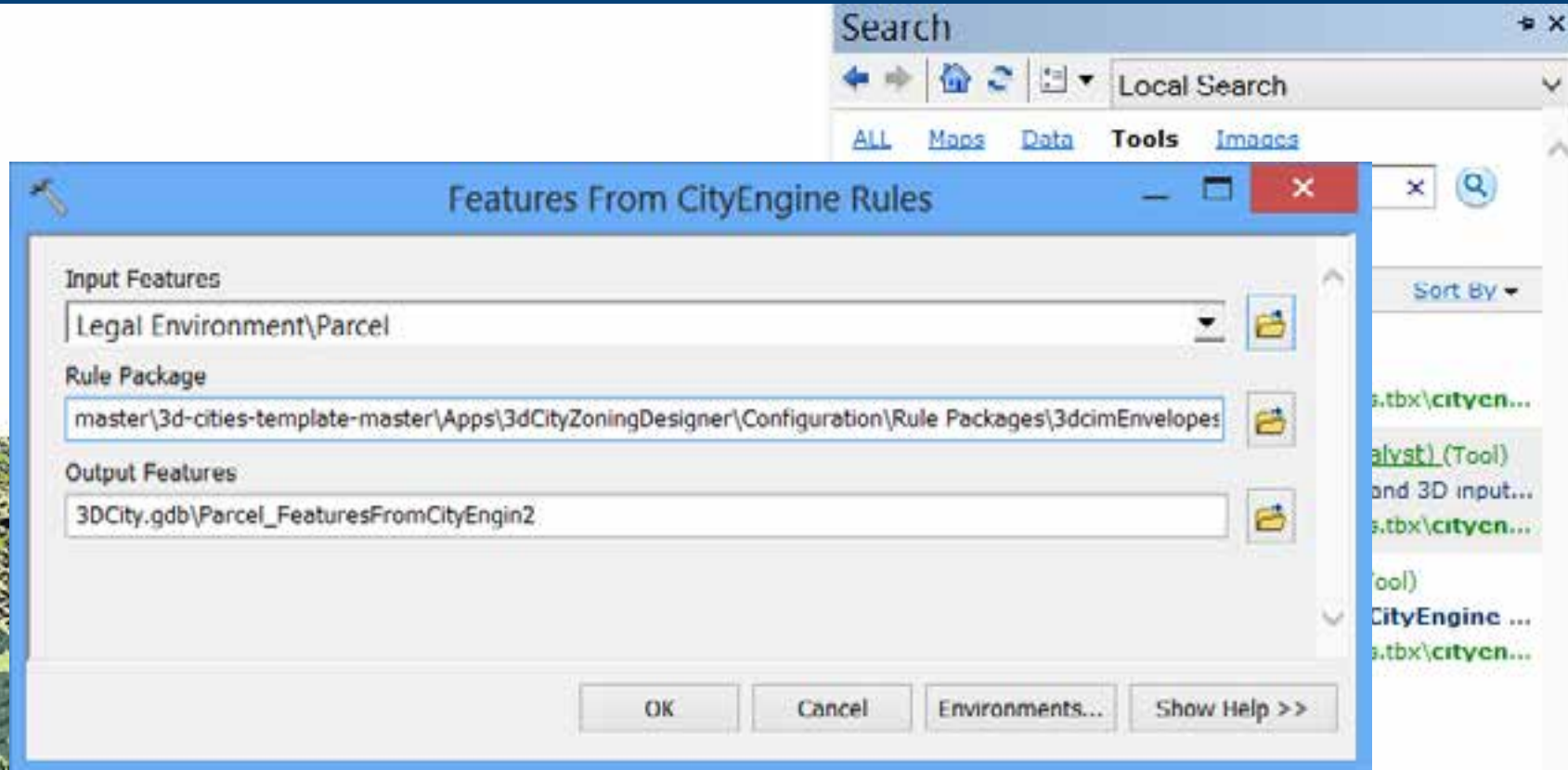


Export from CityEngine

right click on rule, “Share As...”

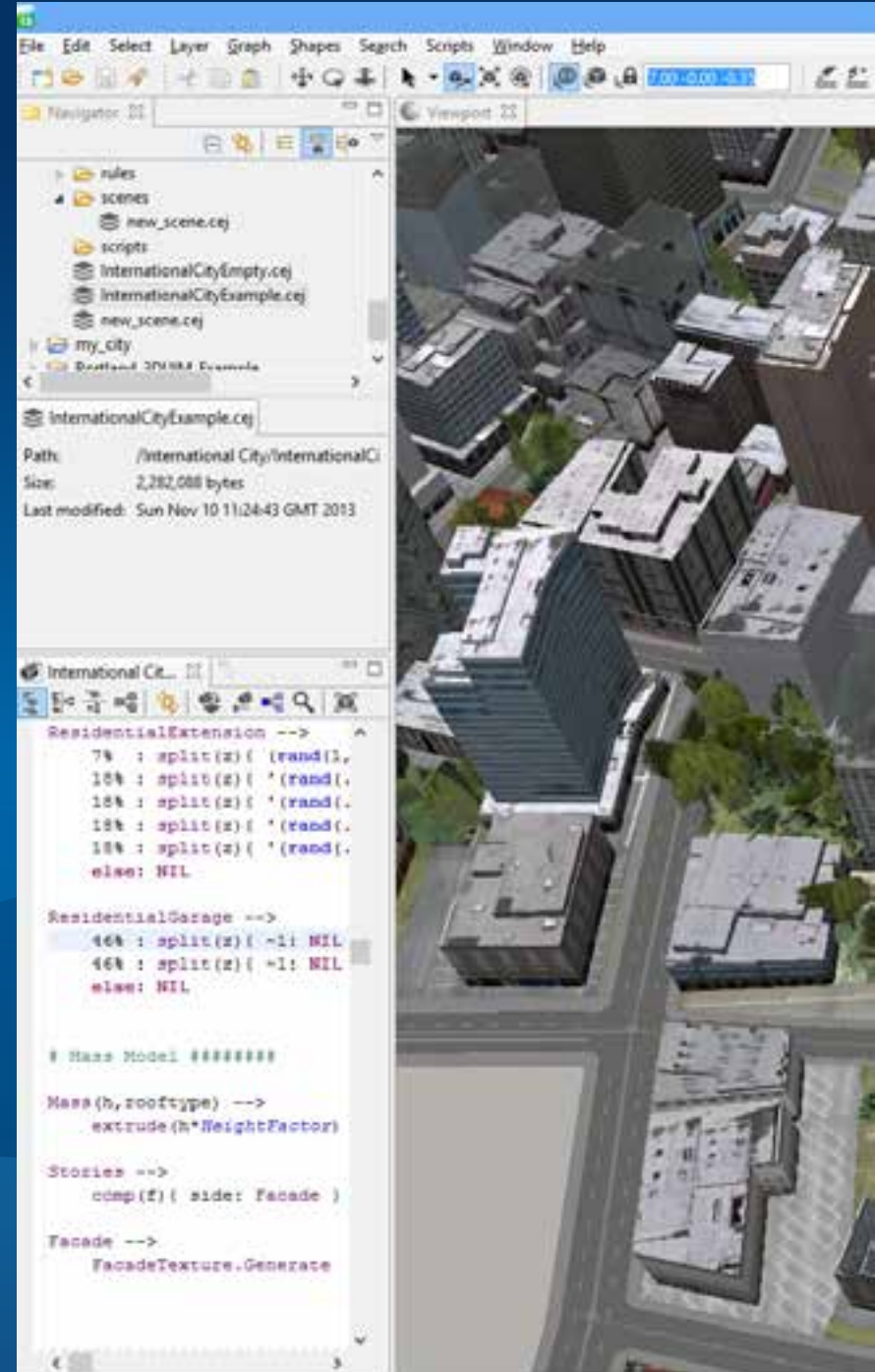


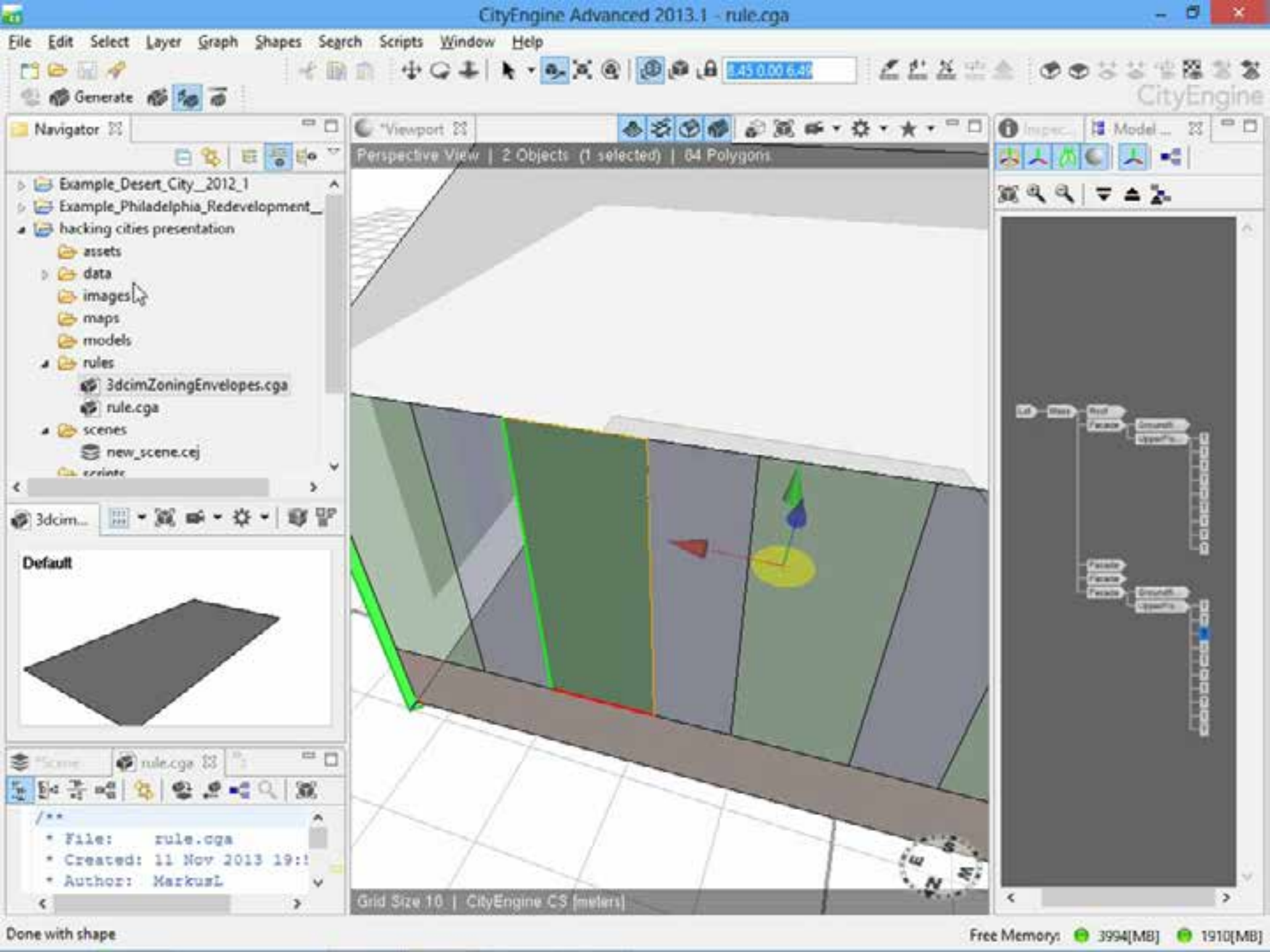
Using in ArcScene - CityEngine GP Tool



Demo

CityEngine 2013

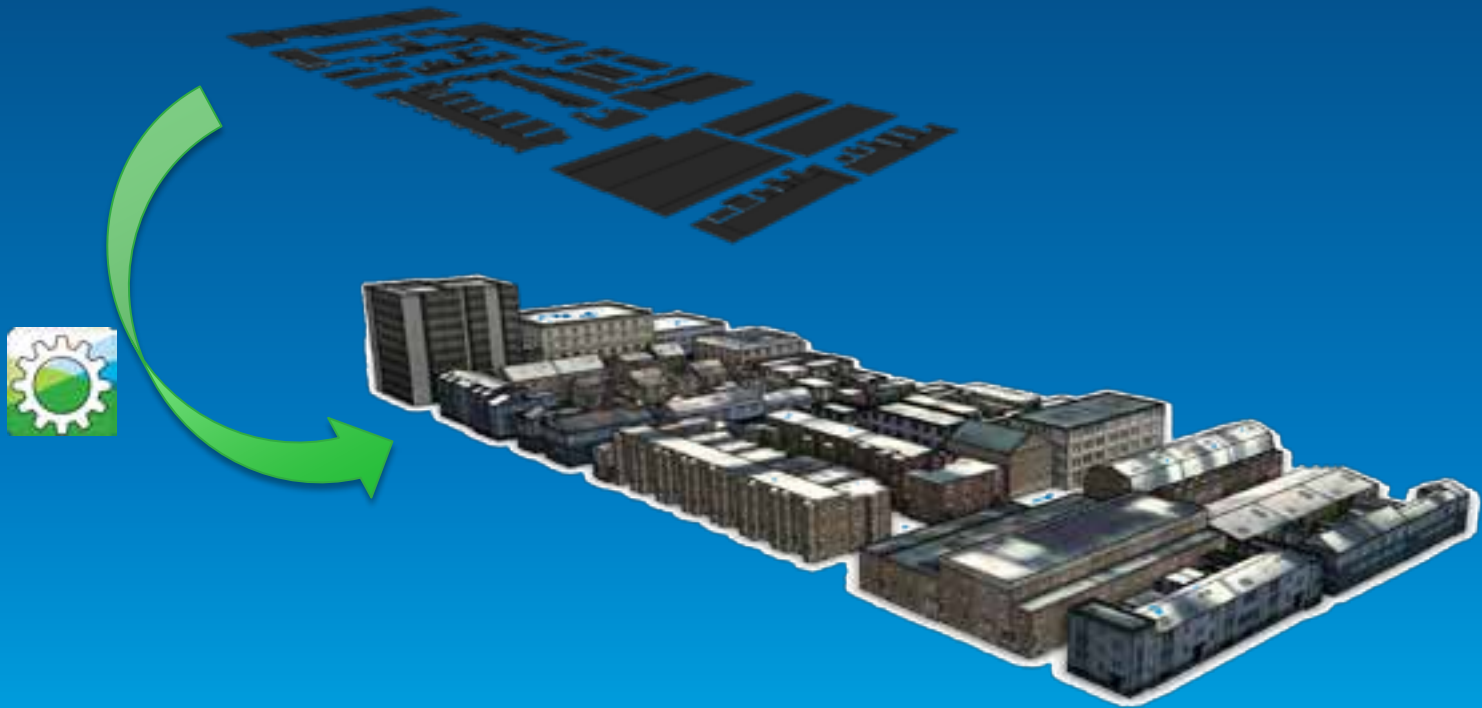




CityEngine GP Tool

Use Cases

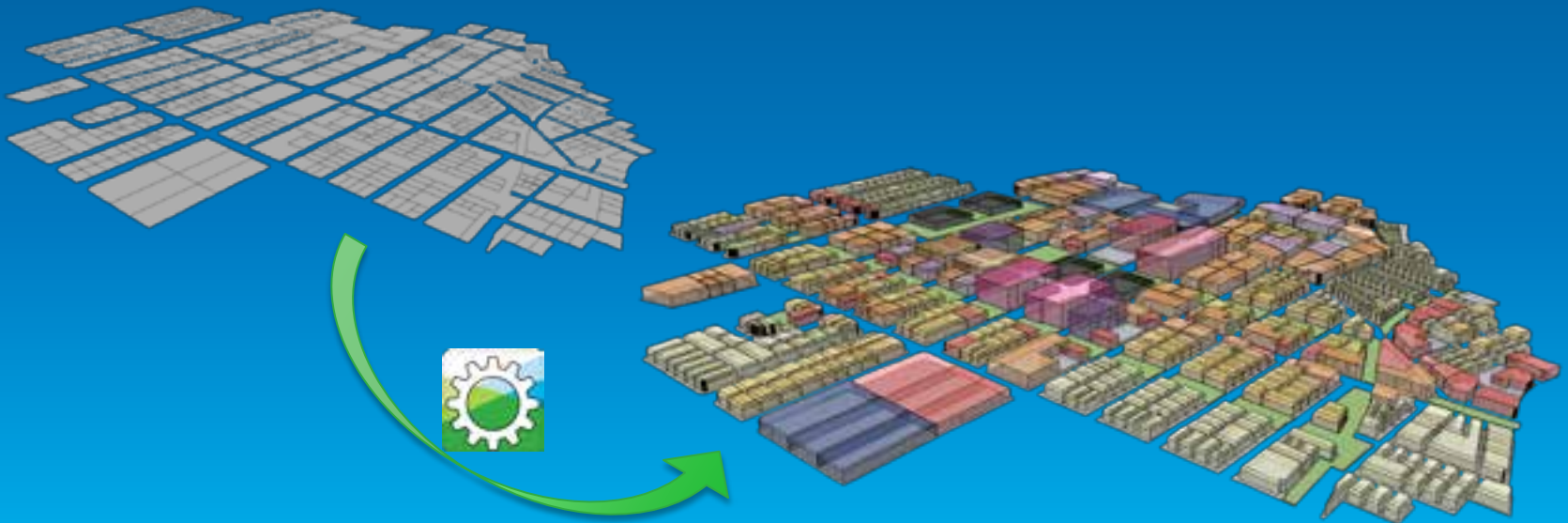
- 2D to 3D: automatic building generation from data model
 - E.g. visualize new development options



CityEngine GP Tool

Use Cases

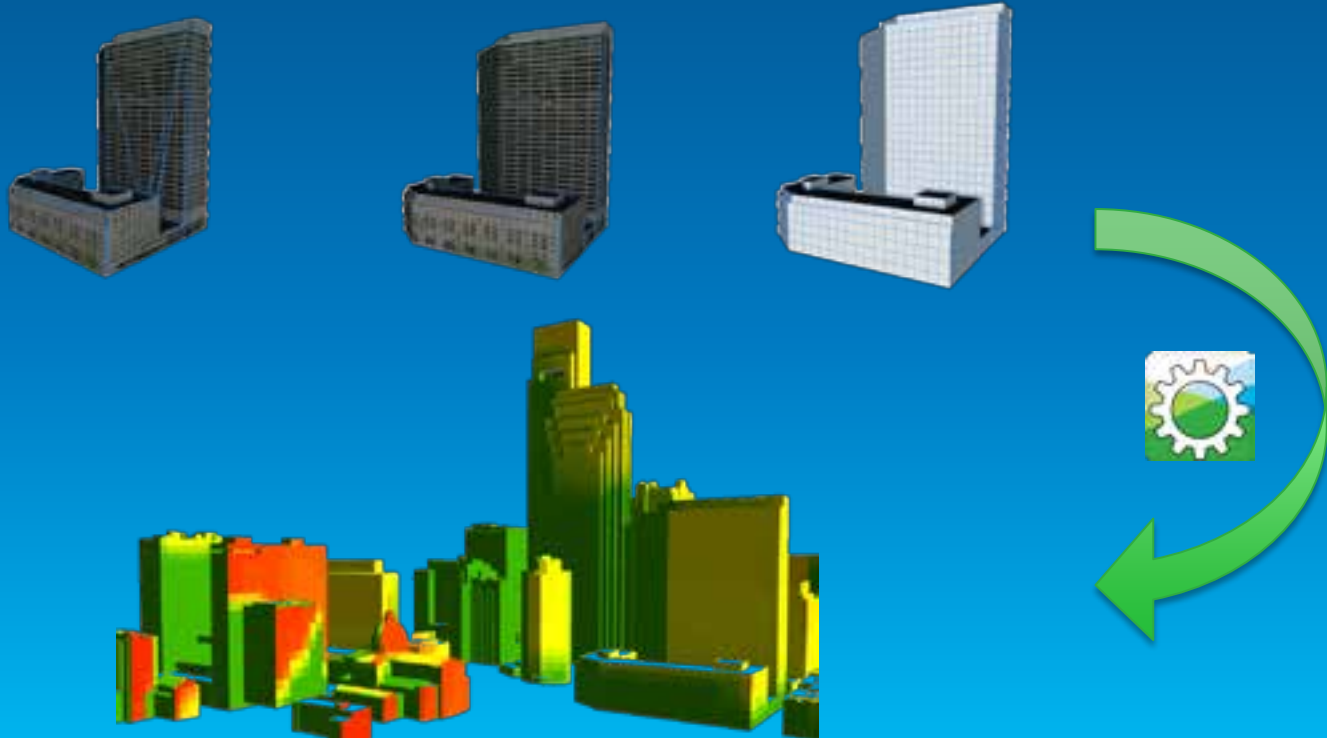
- **2D to 3D: generation of zoning volumes from data model**
 - Intuitive visualization of zoning regulations
 - Analyze impact of regulation changes



CityEngine GP Tool

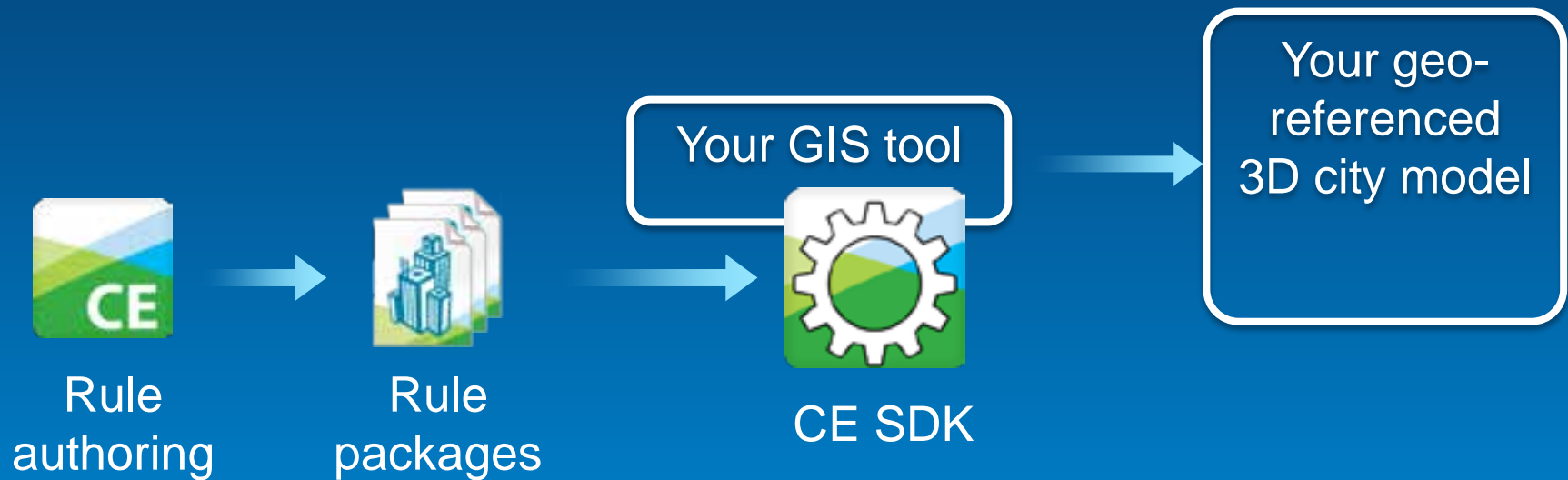
Use Cases

- **3D to 3D: Generate panels on 3D multipatches**
 - Generic rule that subdivides geometry, places point features and/or generates attributes
 - Distribute patches on 3D geometry



3. CityEngine SDK

“Proceduralize” your in-house modeling pipeline



CityEngine SDK

Basis for an Eco-System



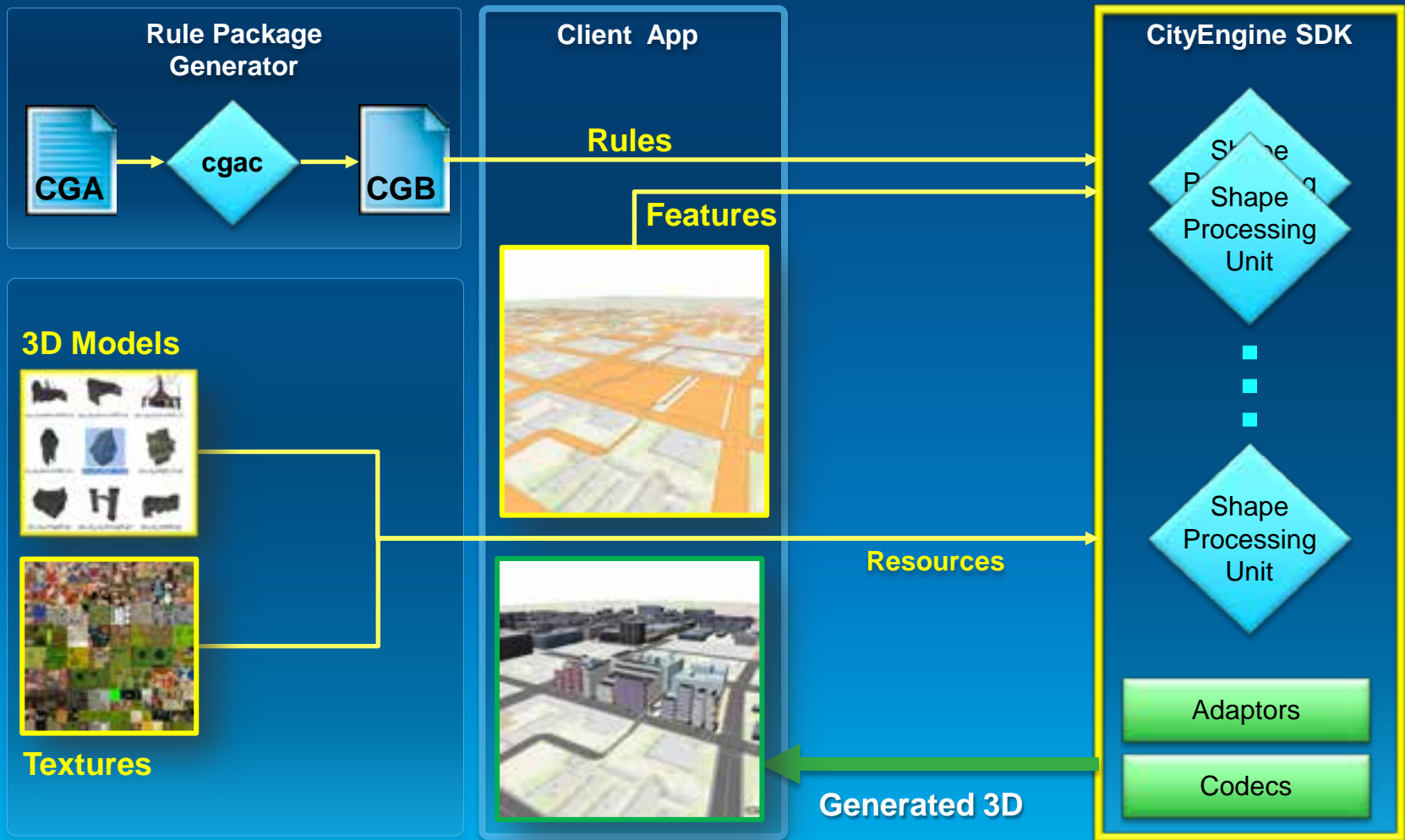
CityEngine SDK

System Architecture

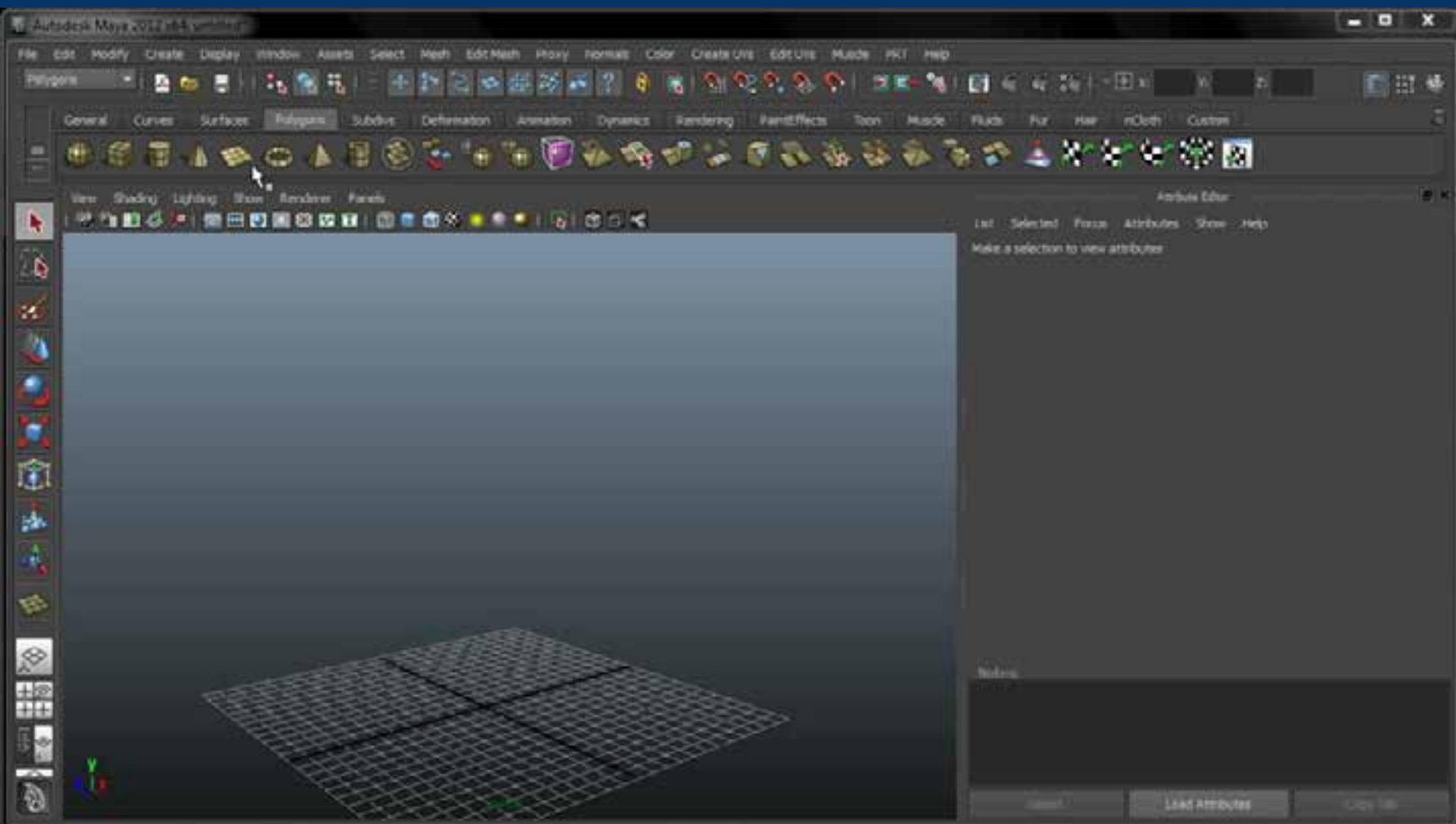


CityEngine SDK

Data & Control Flow



SDK Usage Example – Maya Plugin

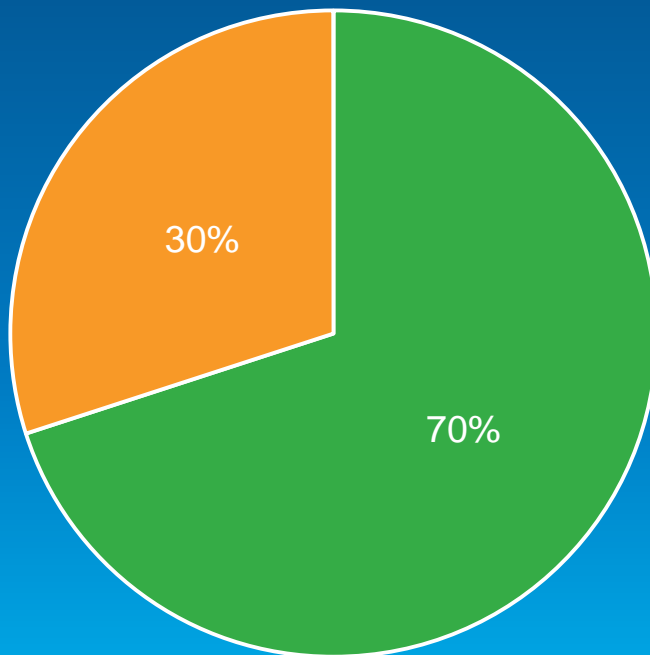


4. Python Scripting



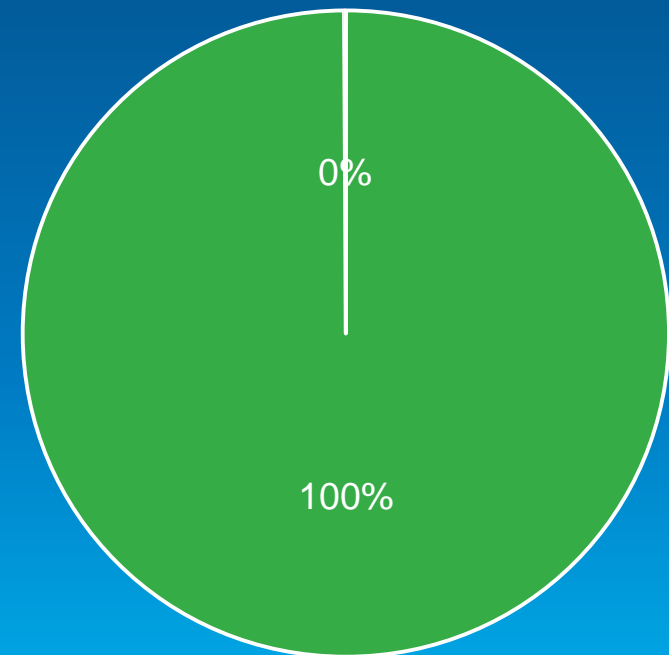
- Automate UI tasks
- CE 2013: All of functions accessible in Python

CityEngine 2012



■ Tools available in Python ■ Unsupported tools

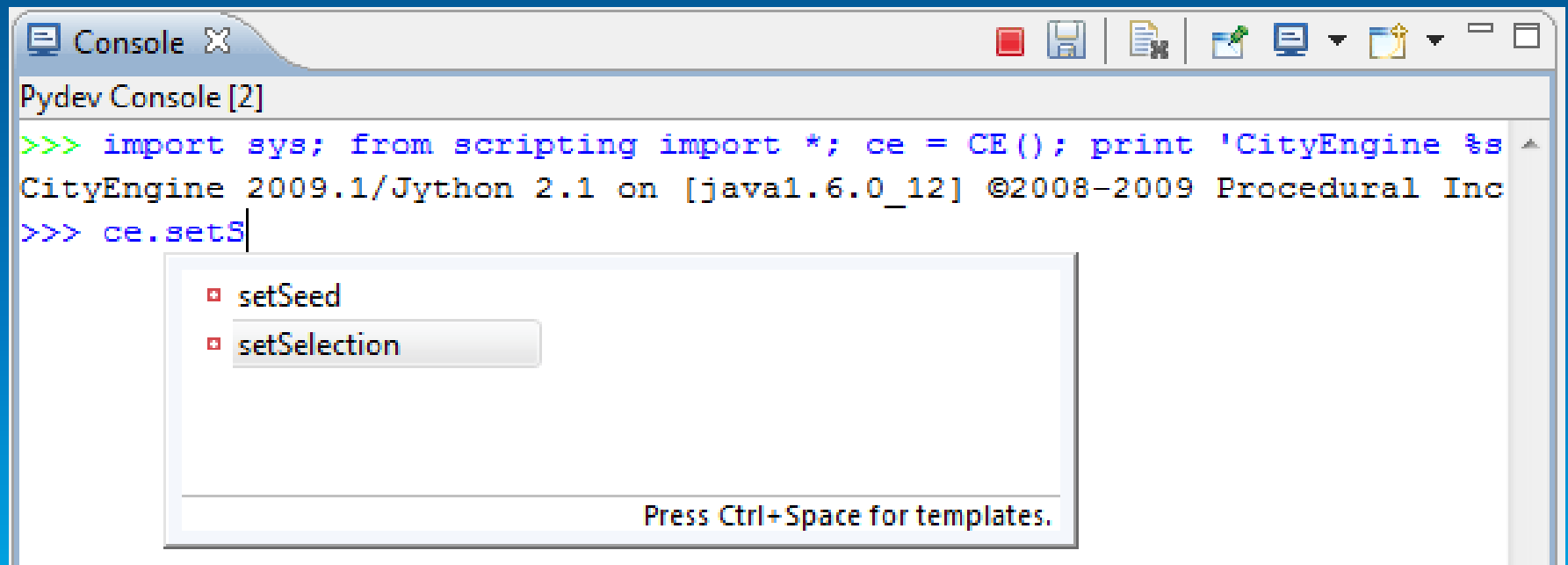
CityEngine 2013



■ Tools available in Python ■ Unsupported tools

Python Scripting

- Python Console:
 - Call CE or conventional Python commands interactively
 - Command completion



The screenshot shows a Pydev Console window with the title 'Console'. The console output displays the following Python code and its result:

```
>>> import sys; from scripting import *; ce = CE(); print 'CityEngine %s'
CityEngine 2009.1/Jython 2.1 on [java1.6.0_12] ©2008-2009 Procedural Inc
>>> ce.setS
```

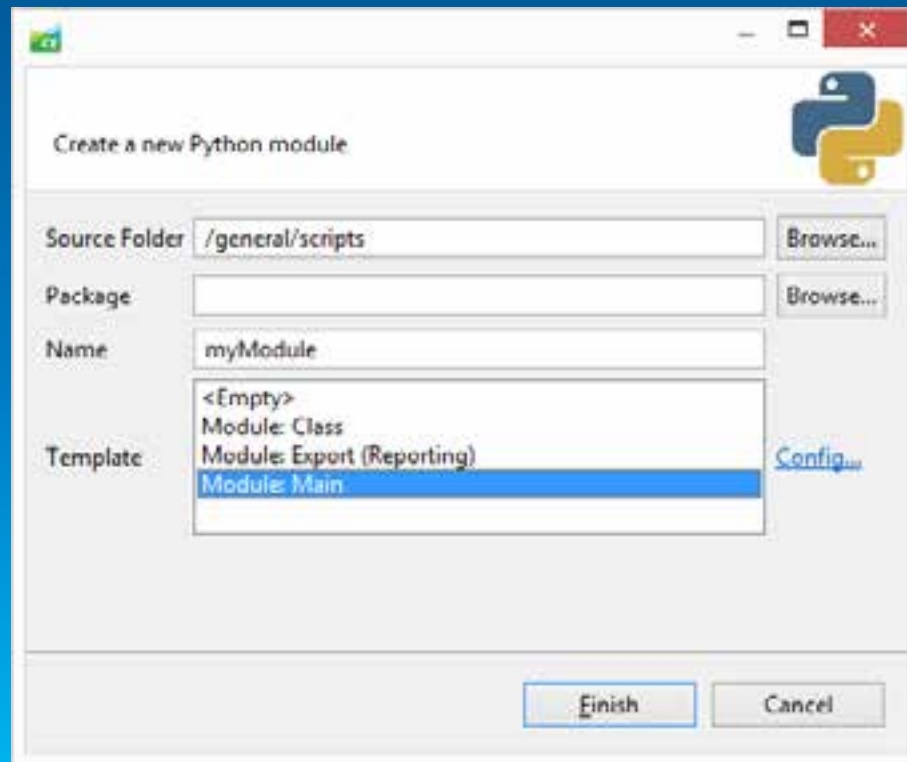
A command completion popup menu is visible, showing two options:

- setSeed
- setSelection

At the bottom of the popup, it says 'Press Ctrl+Space for templates.'

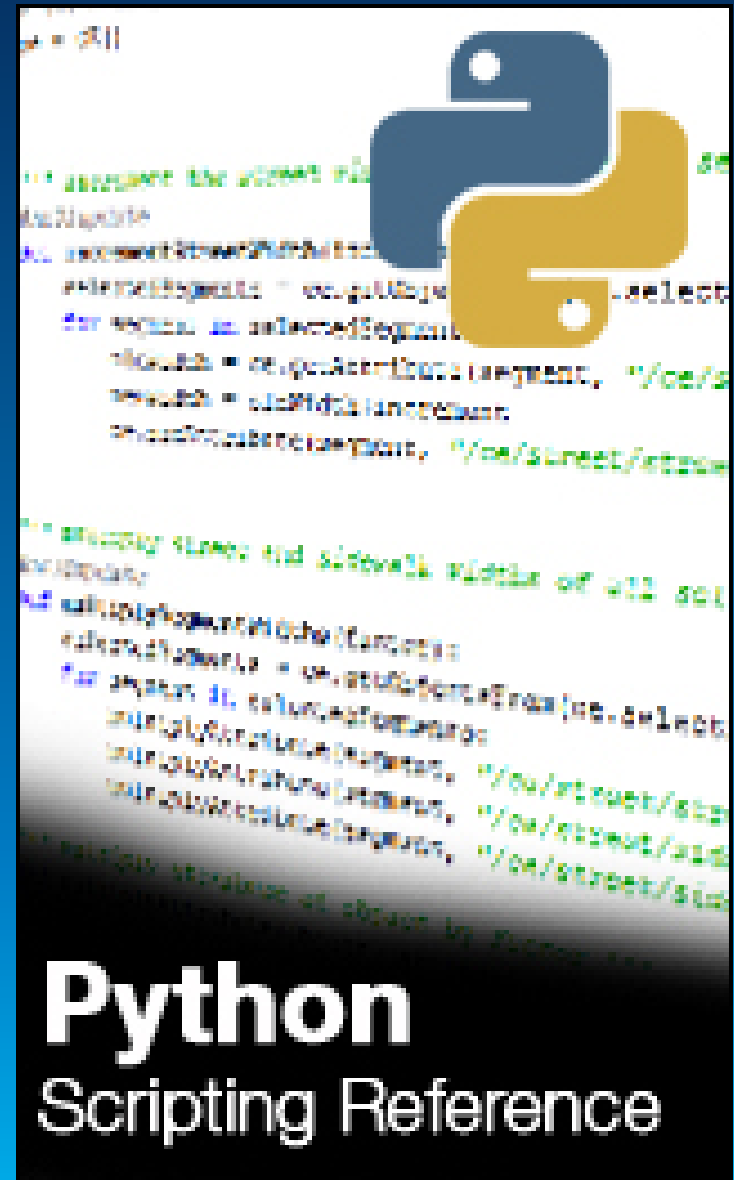
Python Scripting

- Python Editor
 - Convenient editor
 - Edit and execute



Python Scripting

- Extensive command set
see CityEngine Help for reference
- Use your own Python modules



Python: Export via script

```
def exportToObj(shapes, exportName):  
    # create new export settings class, define  
    export format  
    objExportSettings = OBJExportModelSettings()  
    # specify export settings  
    objExportSettings.setGeneralName(exportName)  
    # do the export  
    ce.export(shapes, objExportSettings)  
  
if __name__ == '__main__':  
    exportToObj("pythonExported")
```



scripts/export.py

Python: Export to a set of files

```
def exportMulti(shapes, exportName):  
    for i in range(10,20):  
        # set value of height attribute  
        ce.setAttribute(shape, "/ce/rule/height", i)  
        # call export function  
        exportToObj(shape, exportName + str(i))  
  
if __name__ == '__main__':  
    exportMulti("pythonExported")
```



scripts/export.py

Python: Script Based Export

- Python scripts can run parallel to the export
- Can process arbitrary report data via callback functions
- Powerful mechanism in combination with CGA
`report()`



```
# Called before the export starts.
def initExport():

# Called for each initial shape before
generation.
def initModel():

# Called for each initial shape after
generation.
def finishModel():

# Called after all initial shaped are
generated.
def finishExport():
```

Python: Write report data to file 1

```
def finishModel(exportContextUUID, shapeUUID,
               modelUUID):
    shape = Shape(shapeUUID)
    model = Model(modelUUID)

    # get report variable 'LotArea' of generated model
    reports = model.getReports()
    shapeName = ce.getName(shape)
    lotAreaSum = sum(reports['LotArea'])

    # storing data to global variable
    global REPORT
    REPORT += "%s,%f\n" (shapeName, lotAreaSum)

def finishExport(exportContextUUID):
    # write collected report data to file
    global REPORT
    filename = ce.toFSPath("data/report_LotAreas.txt")
    file = open(filename, "w")
    file.write(REPORT)
    file.close()
```

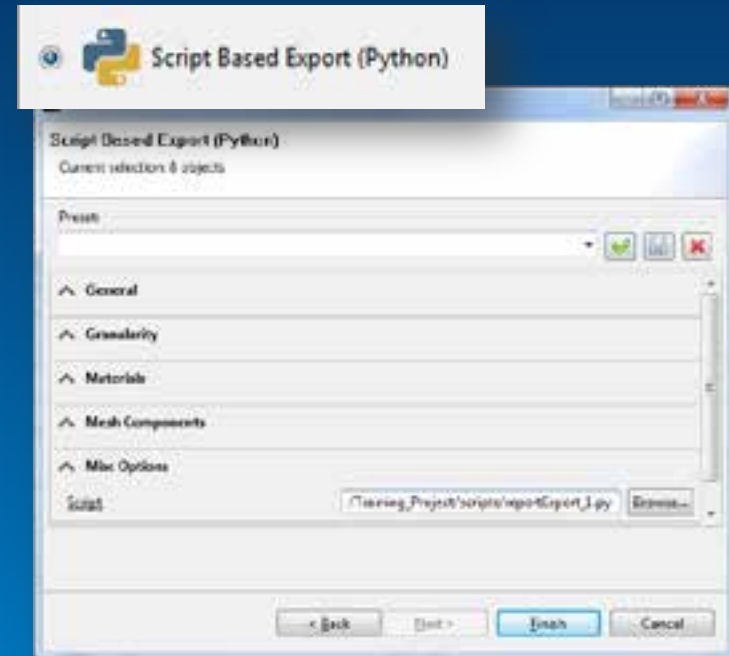


scripts/reportExport_1.py

Python: Write report data to file 2

- Start the script based exporter with python script containing the callback functions
- Collected report data is written to file *data/report_LotAreas.txt*

```
Lot_3,2615.475098  
Lot_2,2573.790283  
Lot_7,1753.116943  
Lot_4,2815.327881  
Lot_1,1365.432495  
Lot_6,2164.343994  
Lot_5,2069.638184  
Lot_0,2551.697510
```



Further New Features in CE2013

Highlights

New Wizard City

International city with over 400 categorized façade images

à Quick city modeling and/or texturing

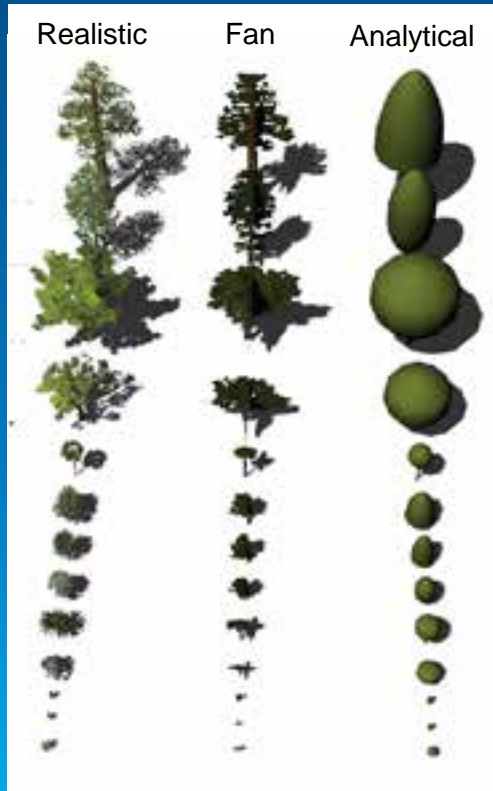
à Serves as template for custom geotypical façade sets



3D Plant Library

Esri vegetation library with 80 LumenRT plants by e-on software

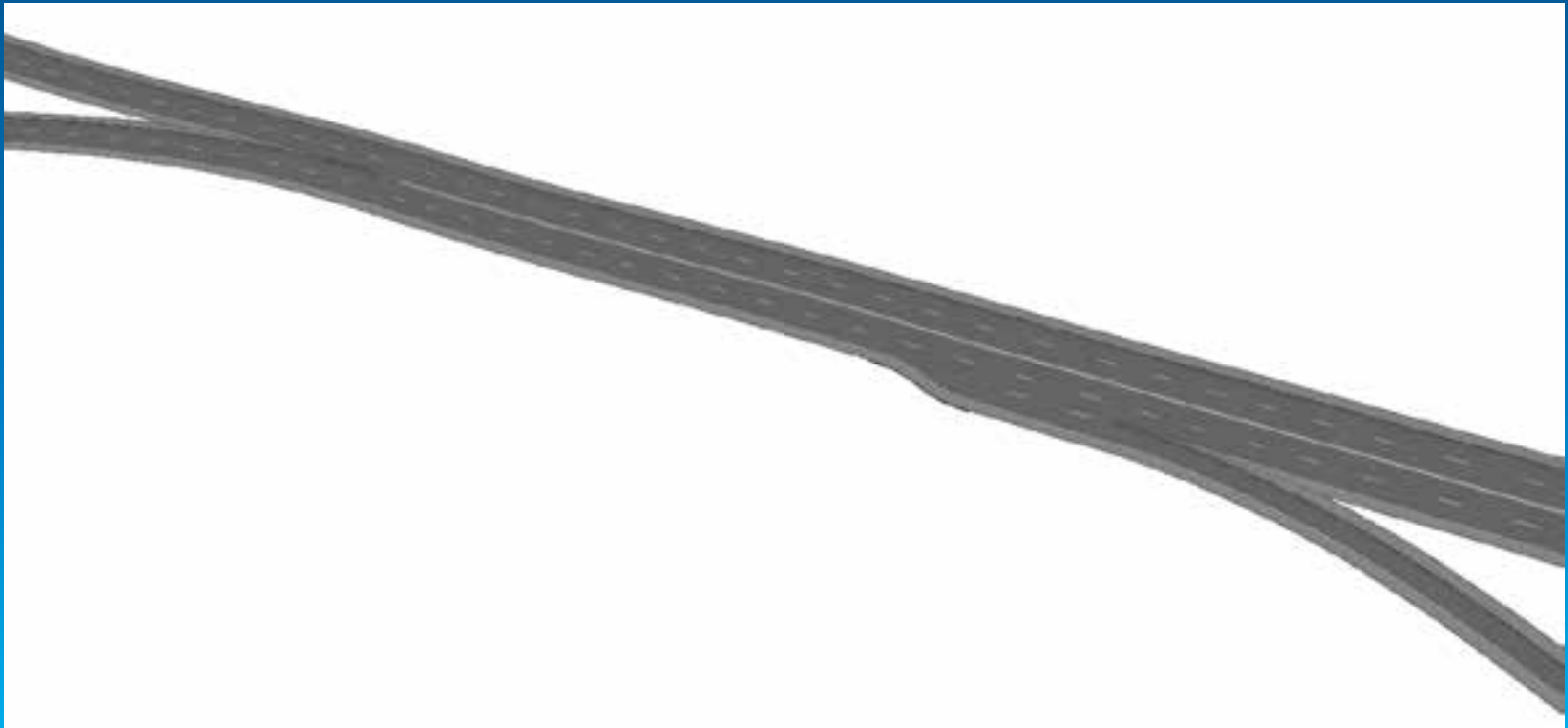
à Design & modeling of visually appealing green spaces



Freeway Intersections

New intersection type for streets incoming/outgoing at acute angles

à Easy modeling of slip roads and freeway interchanges

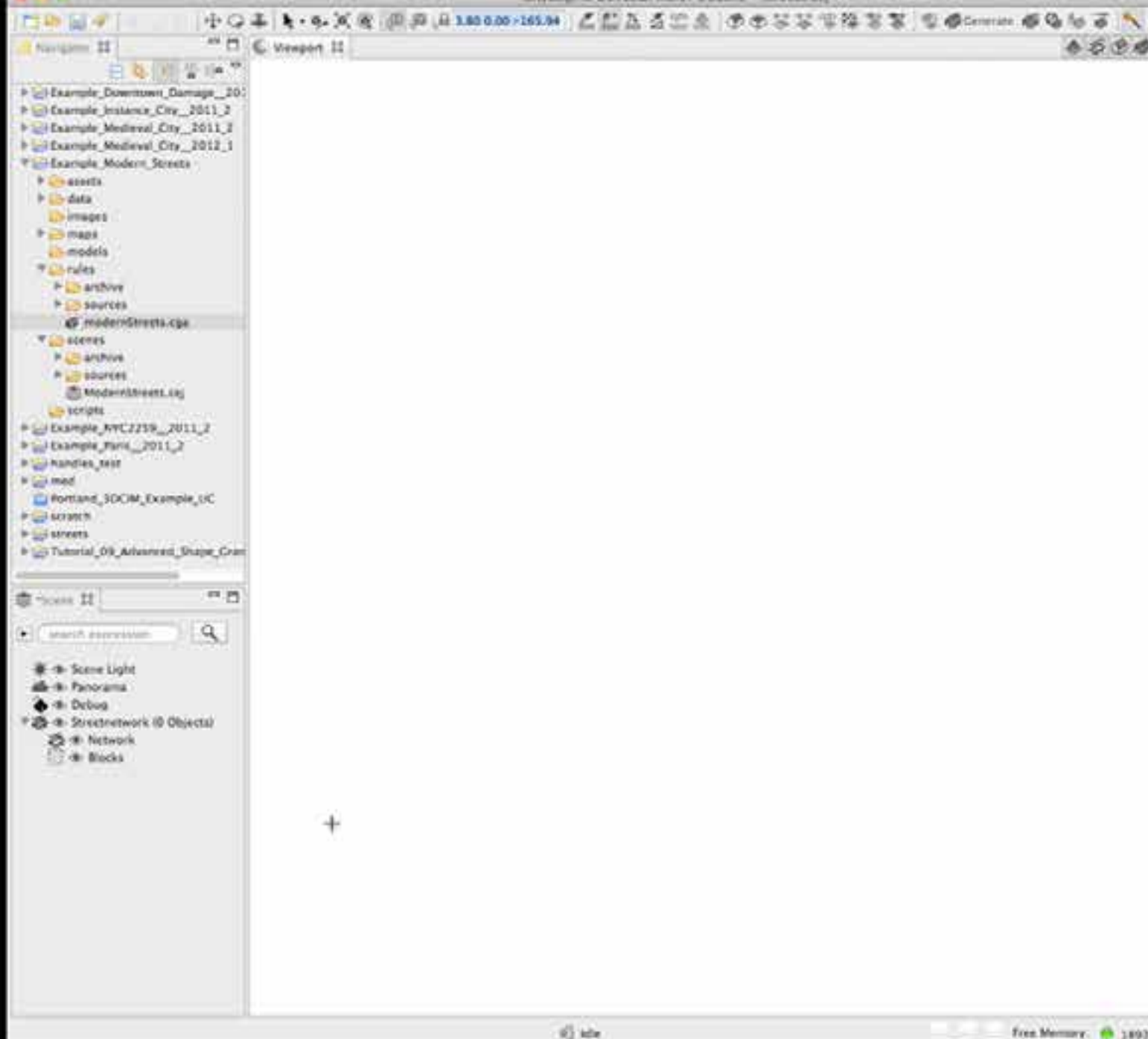


Improved OpenStreetMap & FileGDB Importer

Automatic graph simplification, junction settings & bridge generation

à Easily visualize complex 2D street network in high-quality 3D

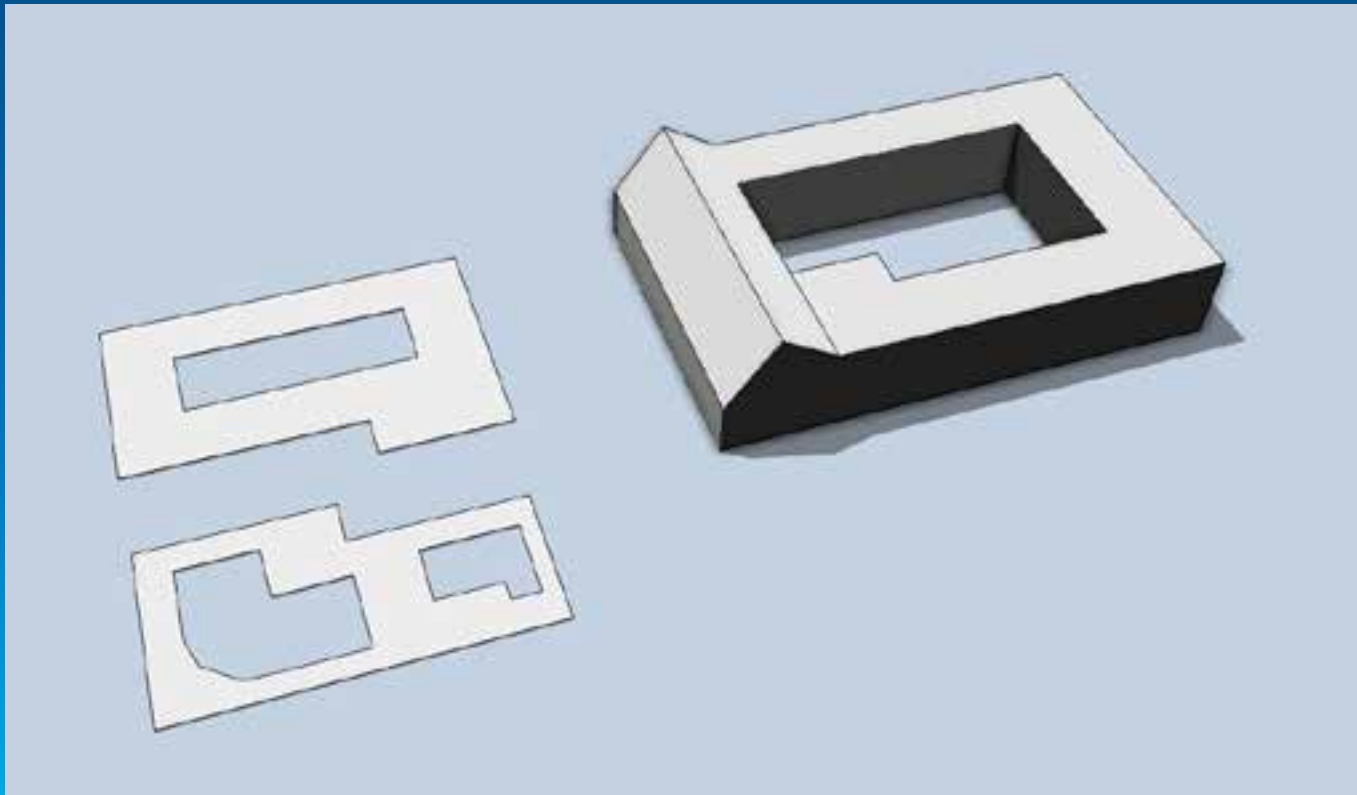




Polygons with Holes

CityEngine now supports shapes/features & models with hole-polygons

à Flawless import, editing & export of geometries with holes

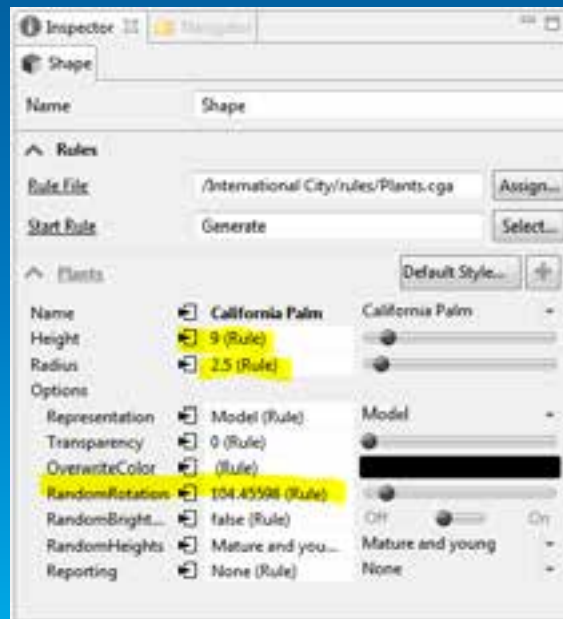


Accurate Rule Attribute Interface

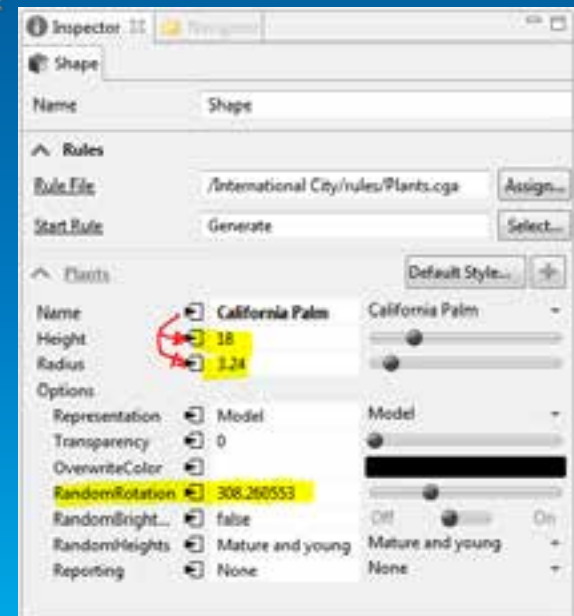
Evaluated values of rule attributes are now shown in Inspector window

à Intuitive WYSIWYG user experience for the rule interface

CityEngine 2012



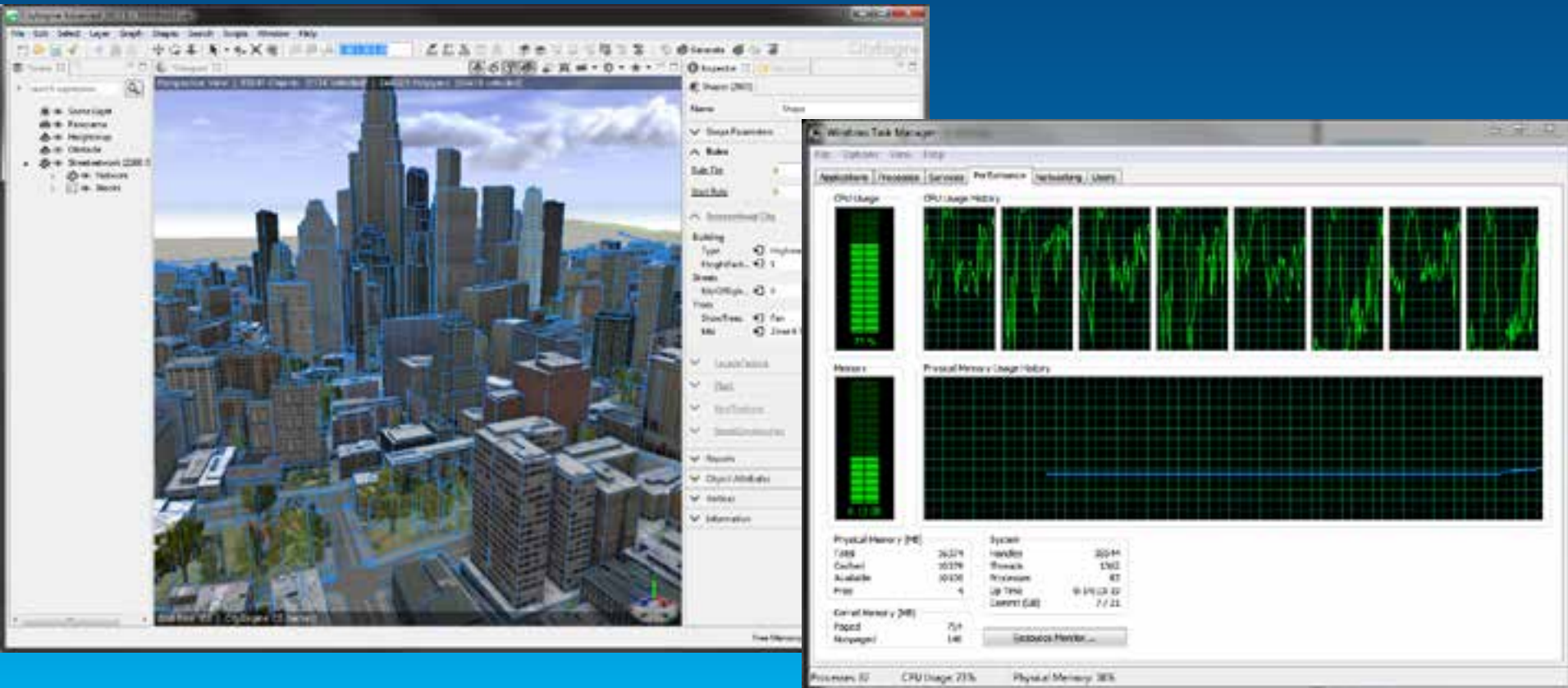
CityEngine 2013



Multi-Threading

CGA now makes use of multi-core processors

à Faster generation of procedural geometries in large scenes



More Compact WebScenes

CityEngine WebScenes now support object instancing

à Smaller file-sizes & faster loading times in browser



CityEngine 2012

45MB

CityEngine 2013

18MB

High-Quality Plant Visualization in Browser

CityEngine WebScenes now support leaf-card materials

à Same plant rendering technology as in games



Support for Retina MacBooks

MacBook Pro with Retina display is now detected and supported

à CityEngine fast and razor-sharp on the latest MacBooks



... and much more:

Terrain supports now no-data values i.e. creates holes instead of wrong surface

Improved Geo-Tiff support for terrain now using GDAL

Map interpolation can now be switched off e.g. for accurate sampling

FileGDB import can read related tables

Improved material mgmt in 3D exporters e.g. consistent when multiple files

Envelope operation now supports also non-planar, slant parcels

In CGA, groups do not have to be listed for every attribute anymore

Simpler workflow to import/link CityEngine projects into Workspace

Intelligent visibility switches e.g. no disappearing objects when in isolate mode anymore

Inspector keeps previous state i.e. no jumping inspector panes anymore

Faster Example and Tutorial Download

...

International City, Plants & Plants in Browser Demo

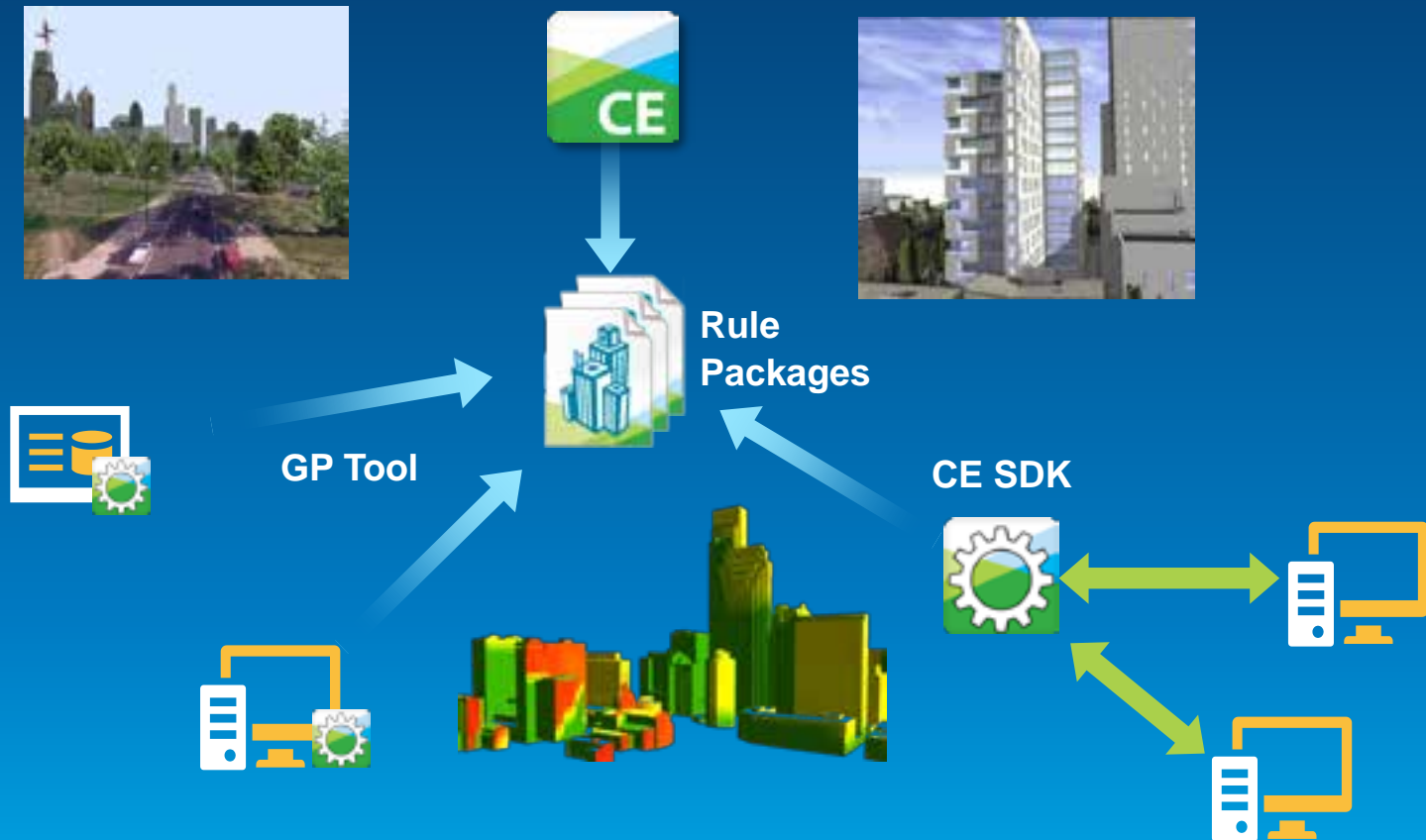
(if time)

CityEngine 2013 timeline

- **November**
- **SDK: Binaries in CE2013 – coming in Nov**
- **SDK Headers, Documentation, Examples (incl Maya Plugin) TBR in GIT repository over the next 2-3 months**



Summary – CityEngine 2013 great for Developers





Understanding our world.