Rationale BIM-Programmierung mit Forge und Vergleich mit der Revit API

Jeremy Tammik

The Building Coder, Autodesk

English Handout

This presentation is in German.
The slide deck is in English.
For full English documentation, please
refer to the class handout at

http://thebuildingcoder.typepad.com/blog/2017/10/rational-bim-programming-at-au-darmstadt.html

Key learning objectives

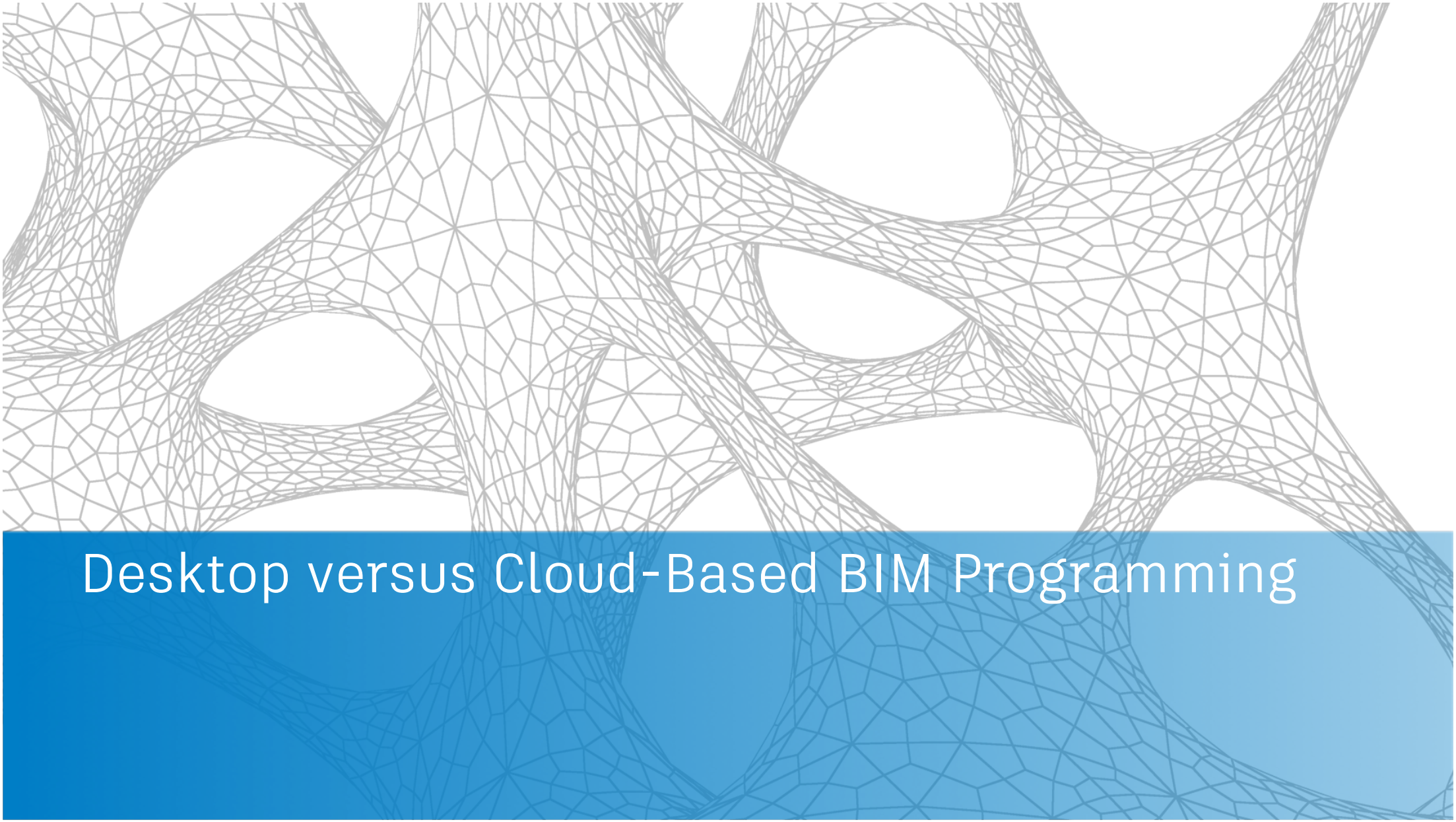
- Envision the most efficient, productive, optimal BIM workflow
- Split software portfolio between desktop add-ins and cloud apps
- Optimal use of Revit, Forge and other programming environments
- Understand pros and cons of commercial, open source and DIY
- Foster ubiquitous connectivity between components
- Enable informed software architecture analysis and design
- Overview of existing samples

Agenda

- Desktop versus Cloud-Based BIM Programming
- Autodesk Forge and Design Automation for Revit
- Using Web Technologies for BIM Programming
- Connecting Desktop BIM and Cloud
- Technologies and Implementation Details
- Resources

Shorter Still...

- Generic BIM programming using web technologies versus Revit-specific add-ins
- BIM connections bridging desktop and cloud to make best use of both worlds



Desktop versus Cloud-Based BIM Programming

Takeaway

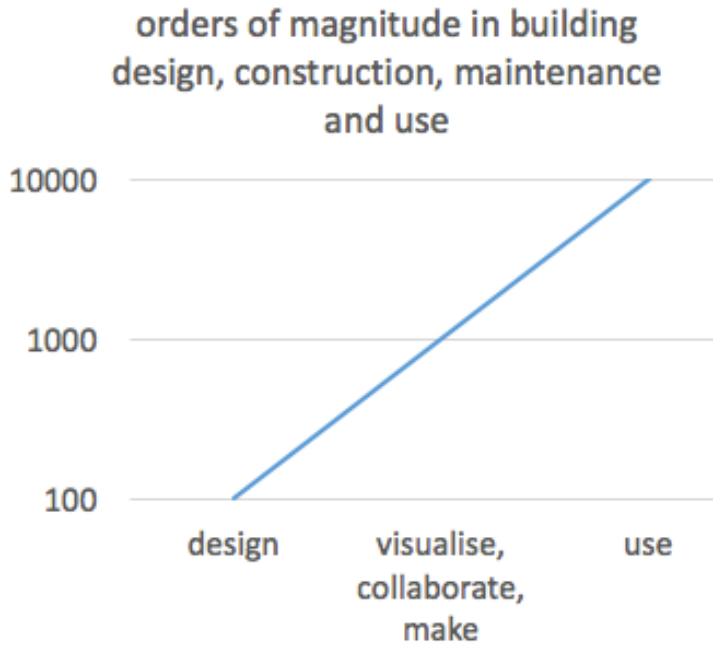
- Split wisely between Revit API and cloud based programming
- Revit is for BIM design
- Many BIM related workflows are read-only
- Many BIM related programming tasks do not require Revit
- Design Automation API for Revit is coming soon

Optimal BIM Workflow

- What is your task?
- Who needs what?
- Where, when, and how?
- Is your user involved in hard-core BIM design work?
- Or, more commonly: view, annotate, adjust, construct, mobile, field work?

BIM Collaboration Roles

- Participant counts grow by orders of magnitude
- Building design, construction, maintenance, use
  - design - architect, engineer - Revit
  - visualise - client, everybody - Viewer
  - collaborate - management – Glue, Plan
  - make - construction – Field, Layout
  - use - inhabit, maintain, FM



Trends, Tools and Technologies

- Revit and BIM Design
  - Expensive
  - Shrinking numbers of desktop computers
- Forge and BIM Use
  - Partially free and open source
  - Growing number of mobile devices
- Glue code, connected custom components
  - Open source libraries
  - JavaScript, HTML5, SVG, WebGL

What Cloud? How Secure?

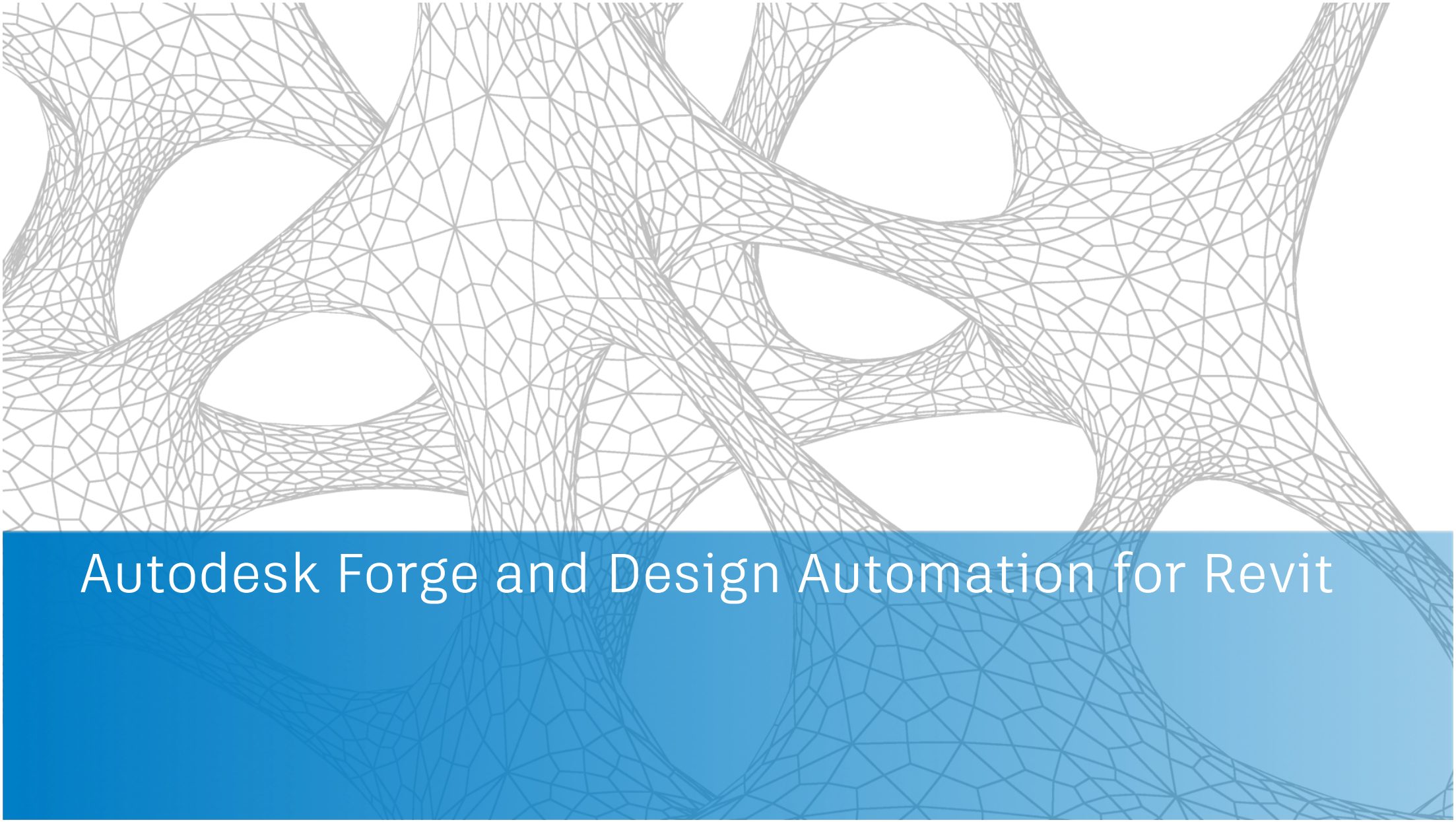
- Locally installed, totally private
  - 2D room editor uses private CouchDB web server
- Global, open source, with standard security measures
  - FireRatingCloud uses node.js web server on Heroku and MongoDB database on mongolab
- Forge OAuth2 user access authorisation
  - Roomedit3dv3

Keep It Simple!

- Simplify your data
  - Graphics? 2D? 3D? Properties?
  - Customise, optimal workflow, minimal complexity
  - Based on 'need to know'
- Use existing components
  - Minimise add-ins, custom components, glue code
  - Open source
  - Forge

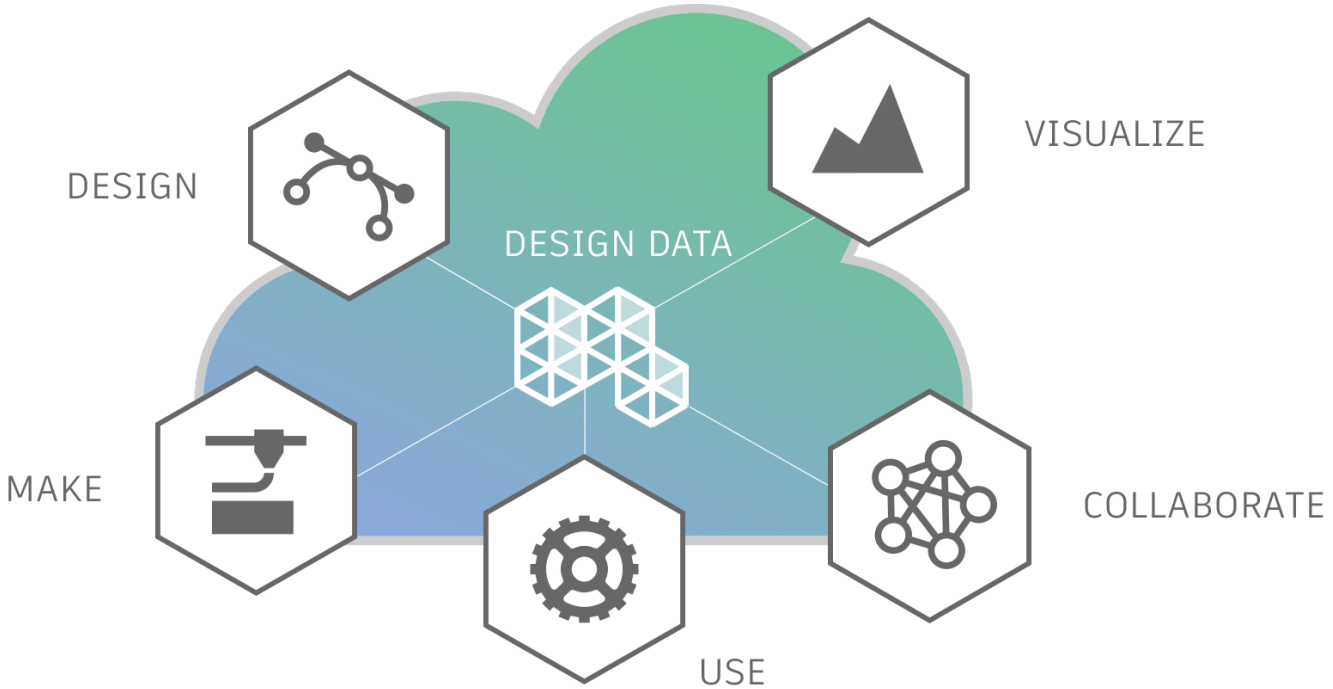
Quotes on Three Fundamental Aspects

- Perfection
  - Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away – Antoine de Saint-Exupéry
- Lazy
  - … develop the three great virtues of a programmer: laziness, impatience, and hubris – Larry Wall
- Simple
  - Simplicity is the ultimate sophistication – Leonardo da Vinci
  - There is no greatness where there is no simplicity – Leo Tolstoy
- KISS


Autodesk Forge and Design Automation for Revit

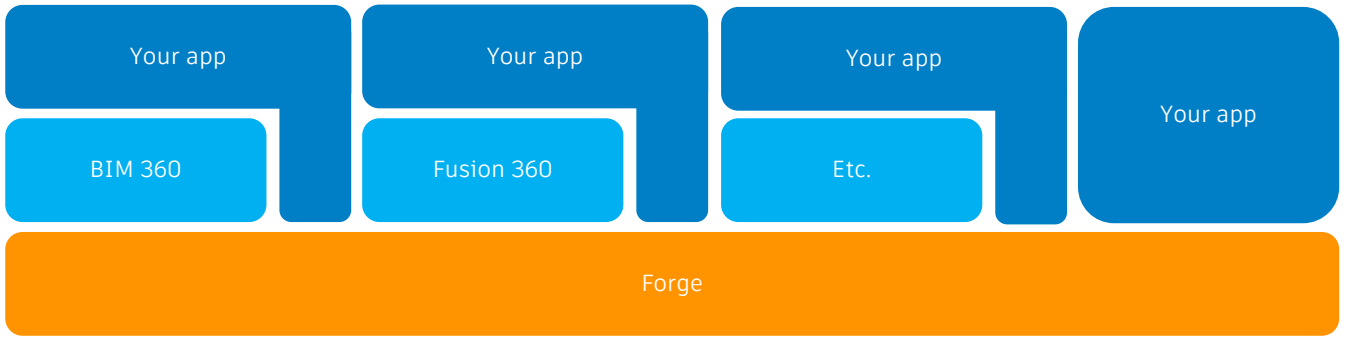
Forge Empowers Developers to Empower Users

- Forge is a platform to empower developers
- They can in turn empower their users
  - design
  - visualise
  - collaborate
  - make
  - use



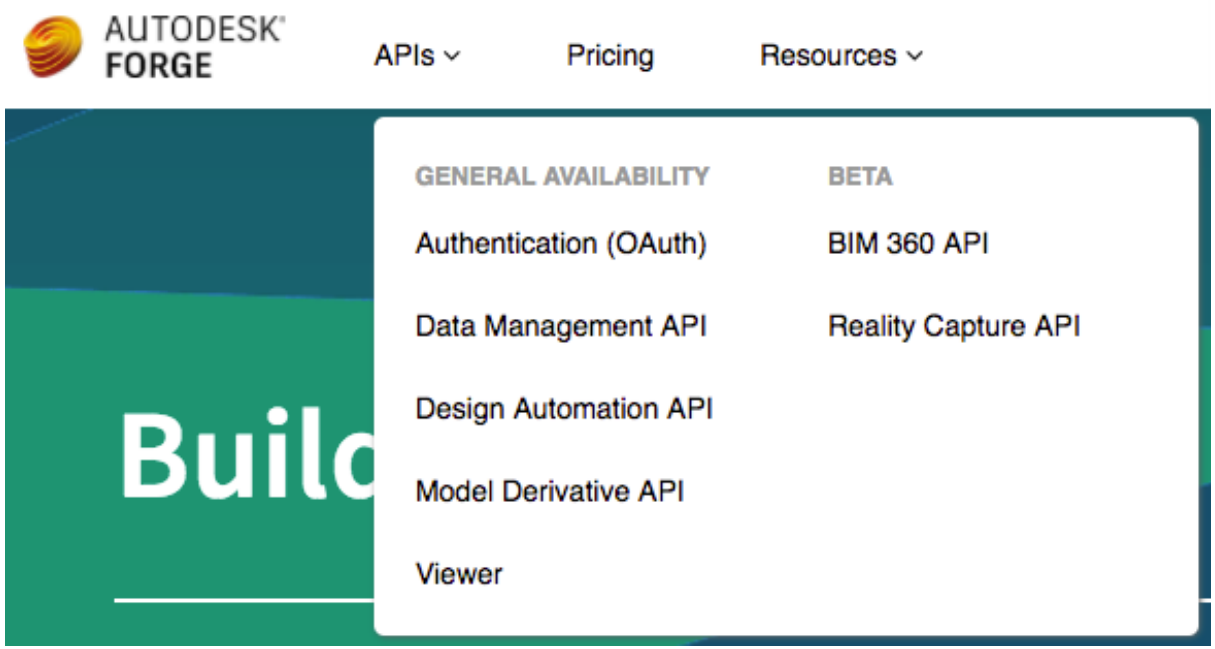
Forge is a Platform

- Autodesk uses Forge to build its products
- Your application has access to the same APIs



Forge APIs

https://autodesk-forge.github.io



Create or Manipulate BIM in the Web

- Do not misuse Revit as a server
- It is an end user product
- If you have such a requirement, talk with Autodesk
  http://thebuildingcoder.typepad.com/blog/about-the-author.html#5.28b

Forge APIs in the Past

- 3 years ago

Share	Collaborate	Create
Translation API		
Viewer		

Forge APIs at Present

- Today, 'Forge 2.0'

Share	Collaborate	Create
Model Derivative API	BIM 360 HQ API	Design Automation - AutoCAD
Viewer	Data Management API	

Forge APIs Near Future

- Next year, 'Forge 3.0'

Share	Collaborate	Create
Model Derivative API	Data Management API	Design Automation - AutoCAD
Viewer	Webhooks API	Design Automation - Revit
	BIM APIs (HQ, RFIs, Issues, Checklists)	Design Automation – Inventor
		Photo-to-3D API
		Forge IDX

Design Automation for Revit

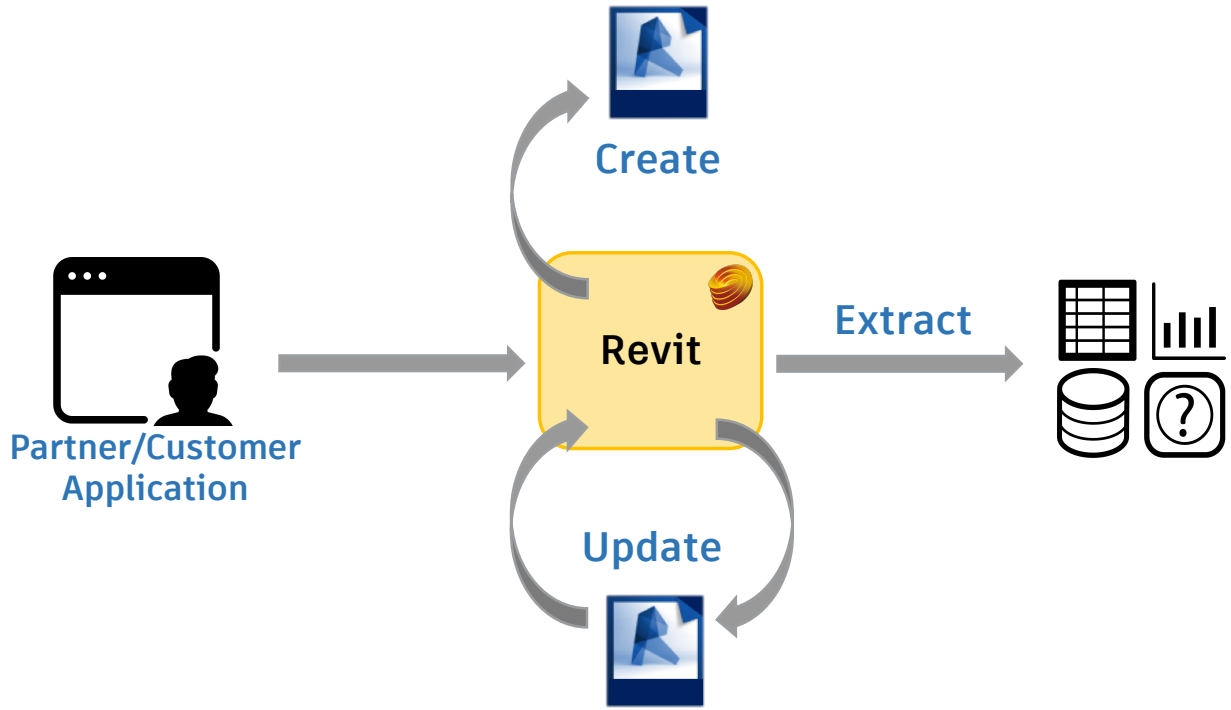
- What is the value?

For Partners	• Create and deploy services and apps that leverage partner's core competencies to **enhance customer workflows**
For Customers	• Quickly extend Revit-based workflows to **solve specific business problems**
For Autodesk	• Facilitate the creation of **new cloud-based solutions** to common customer problems

What Can and Can't It Do?

- What can it do?
  - Full access to Revit DB API outside Revit via cloud services
  - Execute .NET add-ins and Dynamo scripts
  - Access custom functionality while operating on a Revit model
  - Read data from anywhere, save data anywhere
- What can’t it do?
  - No “sticky sessions” – only batch operations
  - Access live Revit data in a RESTful way
  - No Revit UI API access – there is no UI!

Workflows


Create
Partner/Customer
Application
Revit
Extract
Update

Workflow 1 – Create

Use Cases

- Create families (RFAs) from a catalog
- Create RVTs from layout app
- Stair layout generator
- Convert from third party format to RVT/RFA
- Generate full documentation from specification

Hypothetical Example

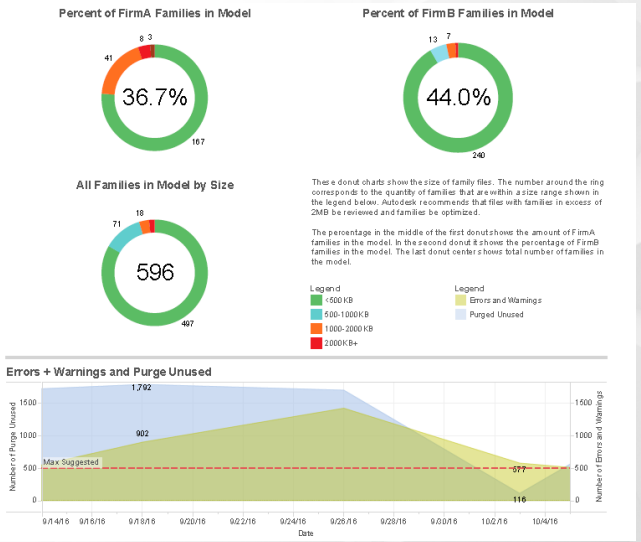
AUTODESK® FORGE

Workflow 2 – Extract

Use Cases

- Custom extraction (Excel, private database, unsupported formats, etc)
- Automated extraction (e.g. nightly PDF or DWG)
- Model QA/QC & code checks
- Clash detection
- Share model information with consultants without need for Revit

Hypothetical Example



AUTODESK® FORGE

Workflow 3 – Update

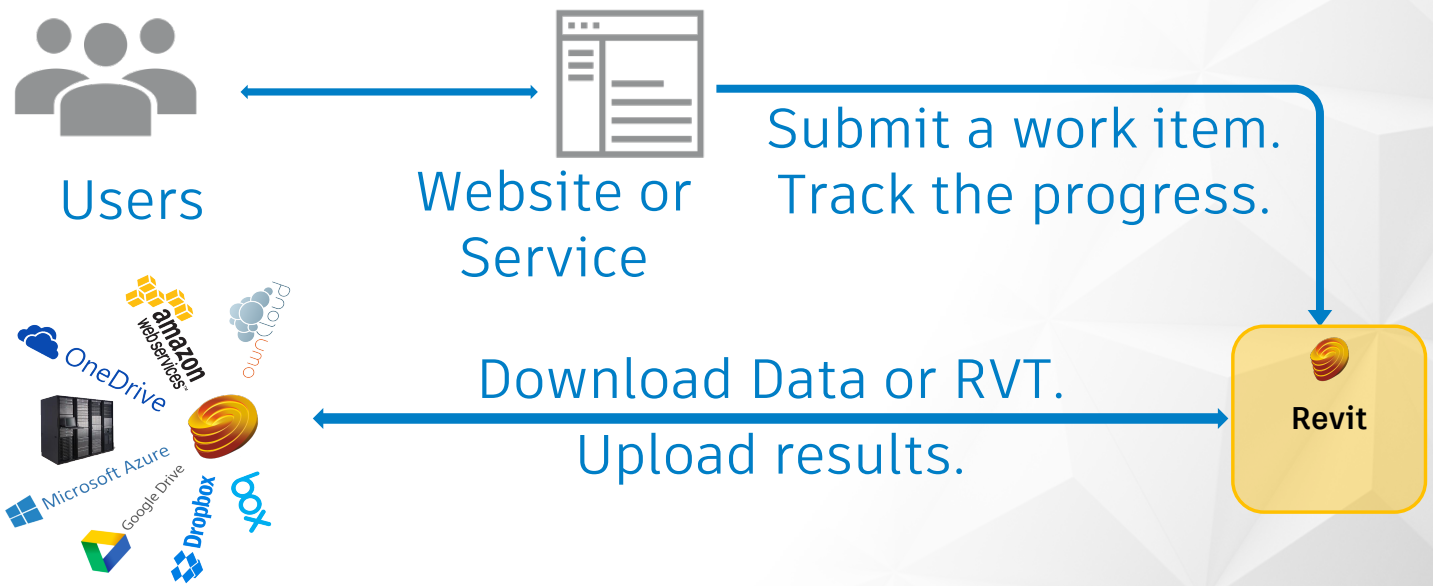
Use Cases

- Execute nightly model tasks on C4R models
- Fix common modeling and documentation mistakes
- Batch upgrades & updates
- Automatically replace out-of-date content
- Scheduled Dynamo scripts
- Design option generation
- And many more!

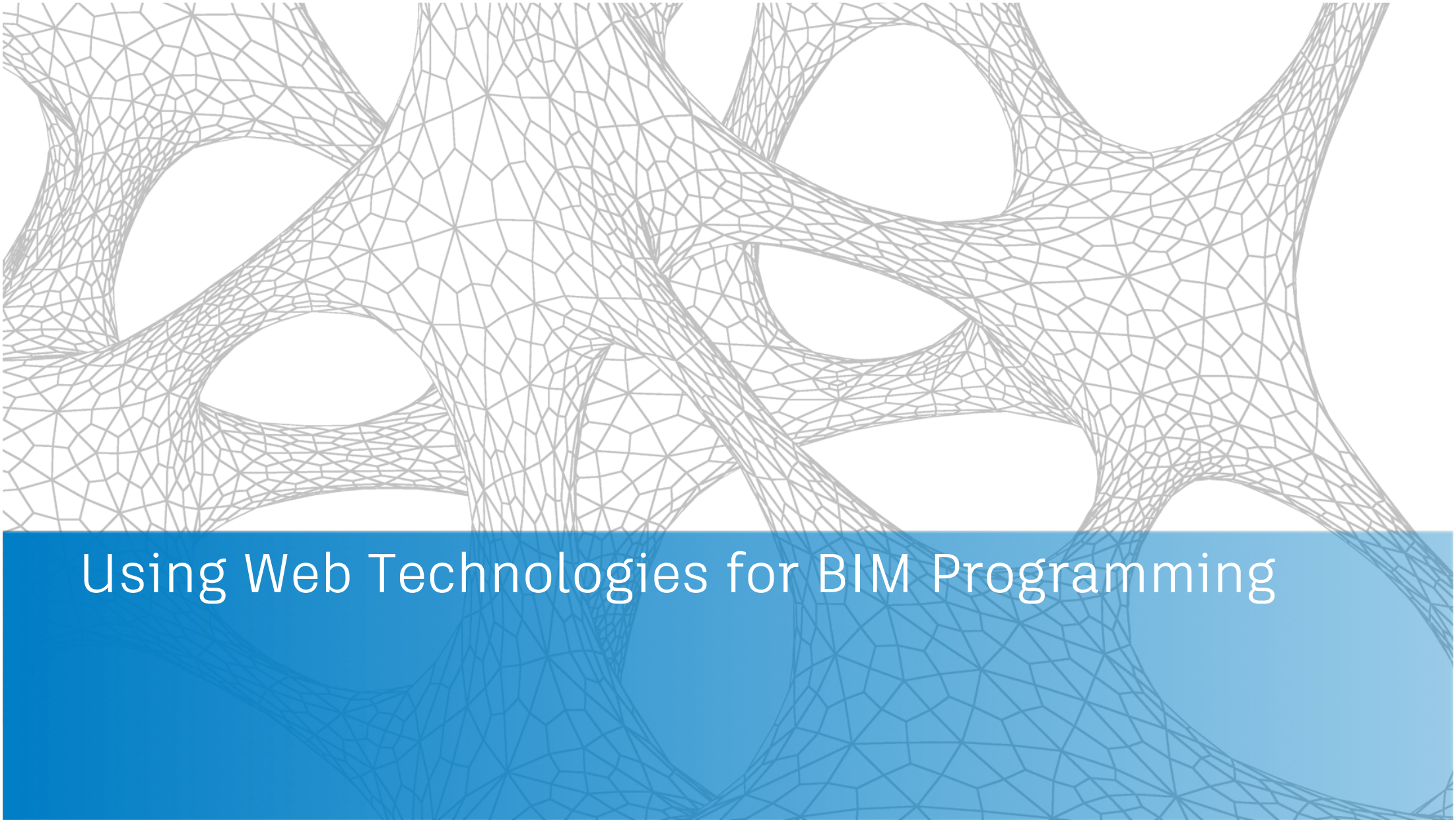
Hypothetical Example

AUTODESK® FORGE

How Does It Work?



AUTODESK® FORGE



Using Web Technologies for BIM Programming

Web-Based BIM Programming Opportunities

- Revit API is BIM and end user product specific
- Many BIM related tasks can be solved more generically

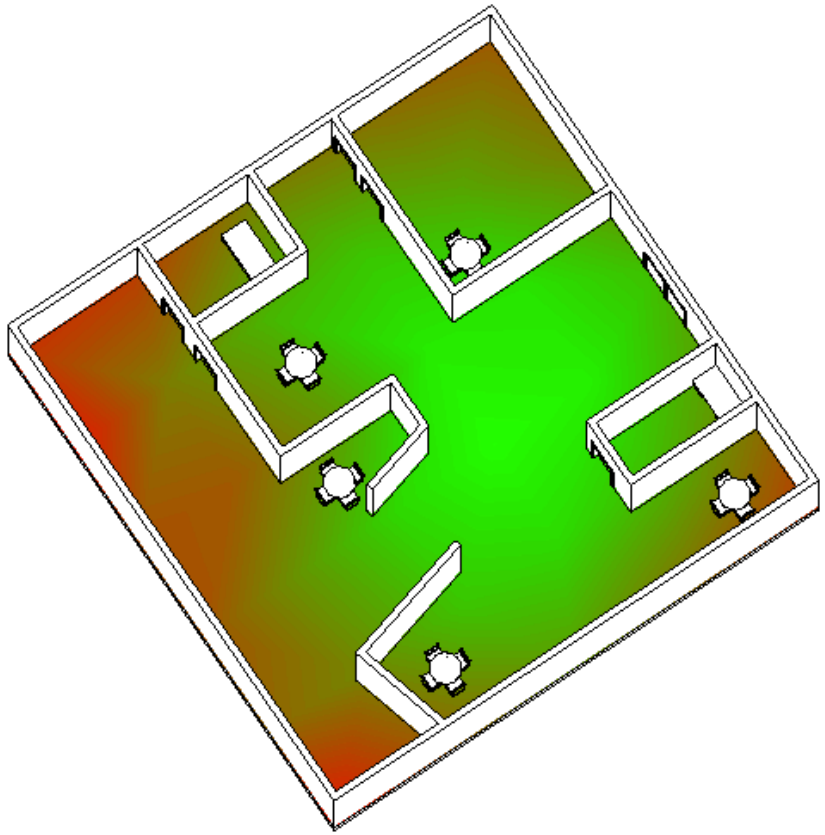
Revit API Characteristics and Limitations

- Revit API is BIM specific
- Very powerful in certain areas
- Leverages Revit product functionality

Therefore:

- Making use of it requires Revit product understanding
- It works only within a running Revit session
- It is completely event driven, hard to 'drive from outside'
- It is limited to the UI oriented Revit end user product
- Programming it requires special training
- Using a Revit add-in requires a Revit installation

Ex: Calculate and Display Signal Attenuation



RvtFader

- C# .NET Revit add-in
- Calculate and display signal attenuation

https://github.com/jeremytammik/RvtFader

- The Analysis Visualisation Framework AVF
- Ray tracing using ReferenceIntersector

ForgeFader

- Forge viewer extension app

  https://github.com/jeremytammik/forgefader

- Demo

  https://forge-rcdb.autodesk.io/configurator > Meta Properties
  https://forge-rcdb.autodesk.io/configurator?id=59041f250007f5c0eef482f2

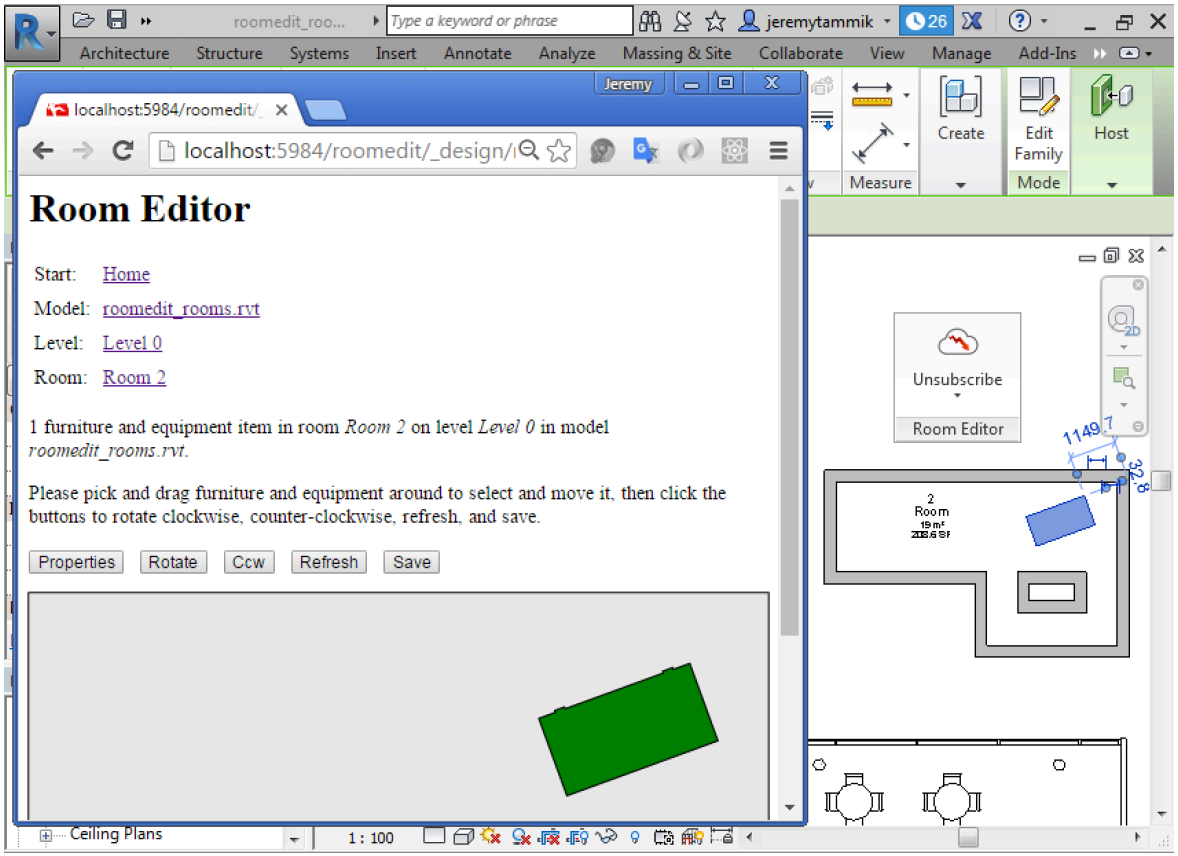
- JavaScript
- Three.js
- Custom shader

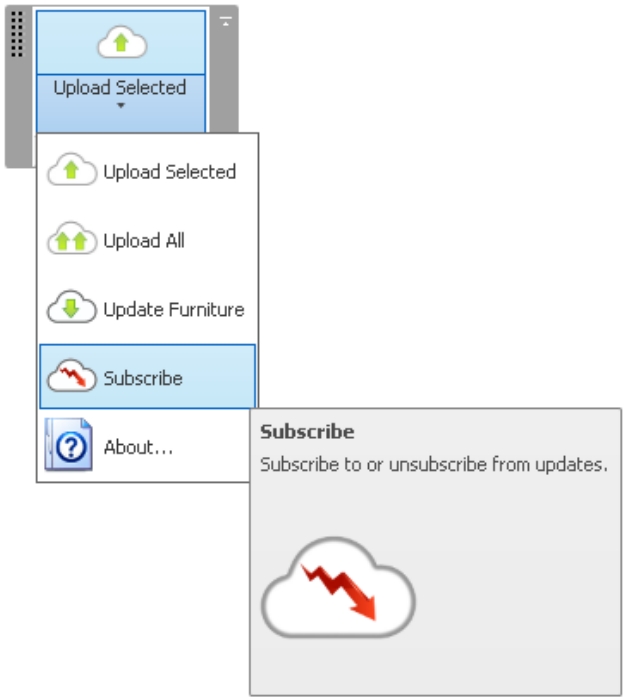

Connecting Desktop BIM and Cloud

Sample Overview

- Simplified 2D BIM room editor, SVG graphics
- FireRating in the Cloud, the simplest sample
- Roomedit3d, Forge based 2D + 3D simplified round-trip BIM editor
- Forge meta property editor + RvtMetaProp

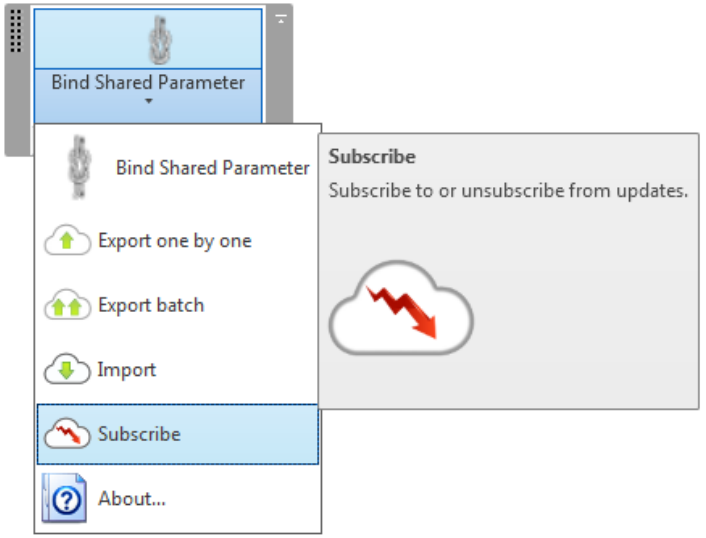
2D Room Editor



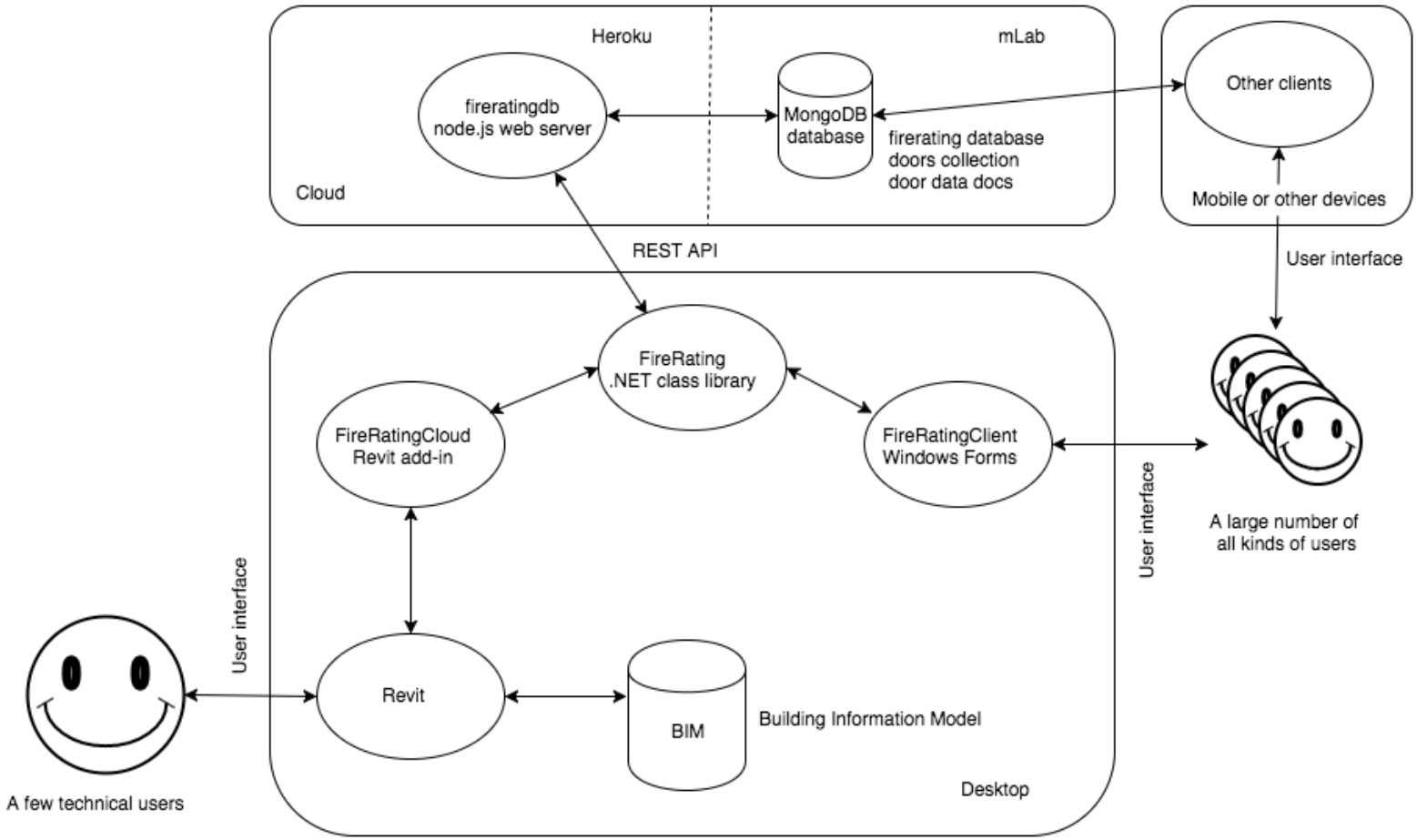


FireRating in the Cloud Commands

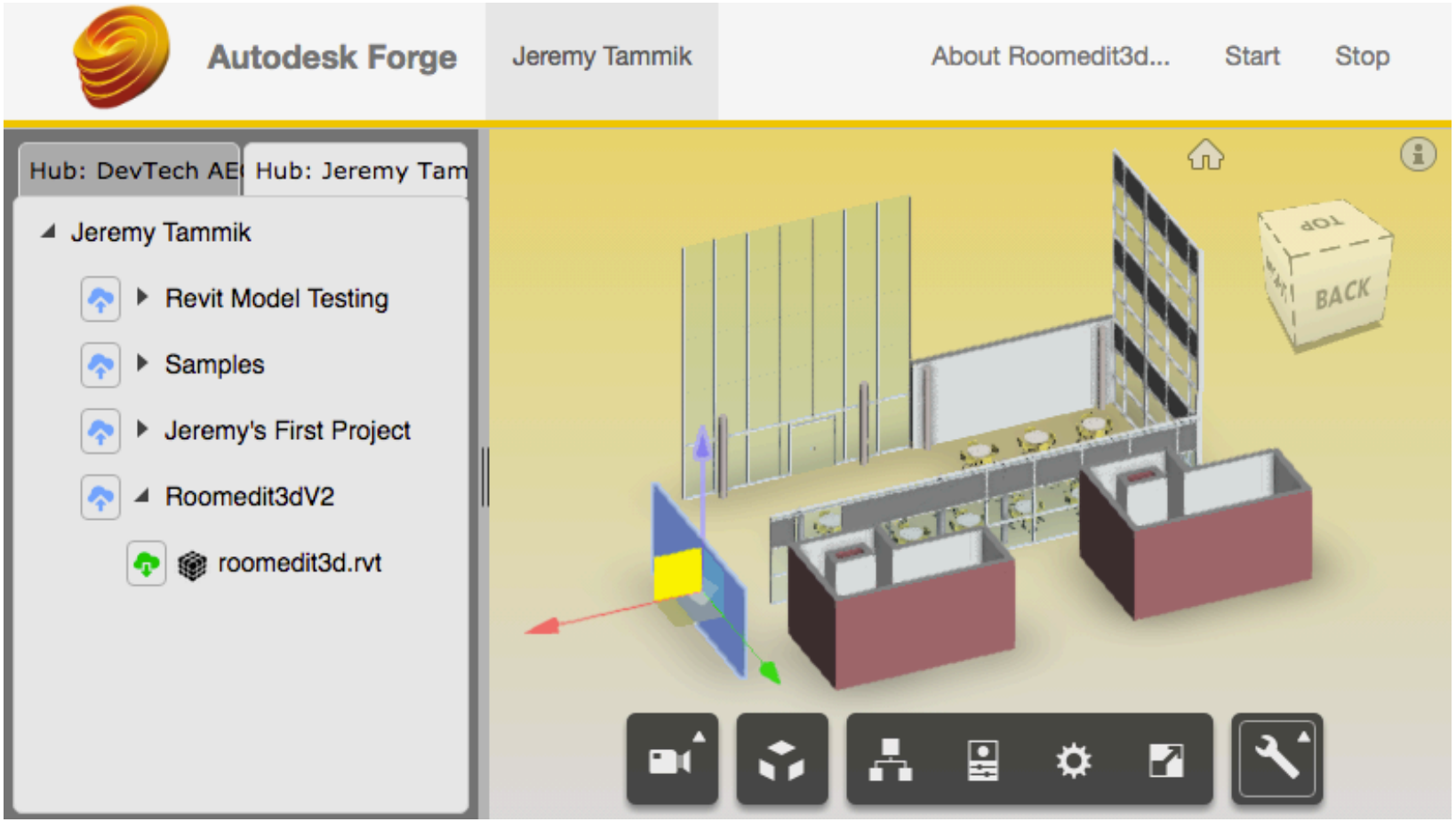
- Create the shared 'Fire Rating' parameter
- Export fire rating values for all doors
- Import the modified values back into BIM
- Store data for multiple projects
  - Cloud database, Revit UniqueId
- Subscribe to changes



FireRating in the Cloud Architecture



Forge Based BIM Editor



Advantages of a Forge Based App

- Realistic model rendering in both 2D and 3D, optionally linked
- Complete access to all BIM data, geometry, structure, properties
- Not bound to any specific model
- Secure authenticated access
- Embedded in a full ecosystem of mature CAD related web services
- Minimal amount of coding based on boilerplate sample code

Forge Meta Property Editor

- https://forge-rcdb.autodesk.io/configurator
- Meta Properties
- Use Model: Office

https://forge-rcdb.autodesk.io/configurator?id=59780eec17d671029c53420e

- Add and edit custom properties on elements in Forge viewer
- Download custom property data in CSV or JSON

RvtMetaProp

https://github.com/jeremytammik/rvtmetaprop

- Read custom property data in CSV or JSON
- Associated with individual building elements
- Modify existing element parameter data
- Generate shared parameters for new
- Requires suitable category sets


Technologies and Implementation Details

NoSQL

- “Not only SQL”
  - Next generation database paradigm
- Some characteristics
  - Non-relational, distributed, open-source, scalable, huge data
- Frequent other characteristics
  - Schema-free, easy replication support, simple API, eventually consistent, i.e., BASE, not ACID
    - http://nosql-database.org
    - https://en.wikipedia.org/wiki/NoSQL
    - http://www.mongodb.com/nosql-explained

CAP and ACID versus BASE

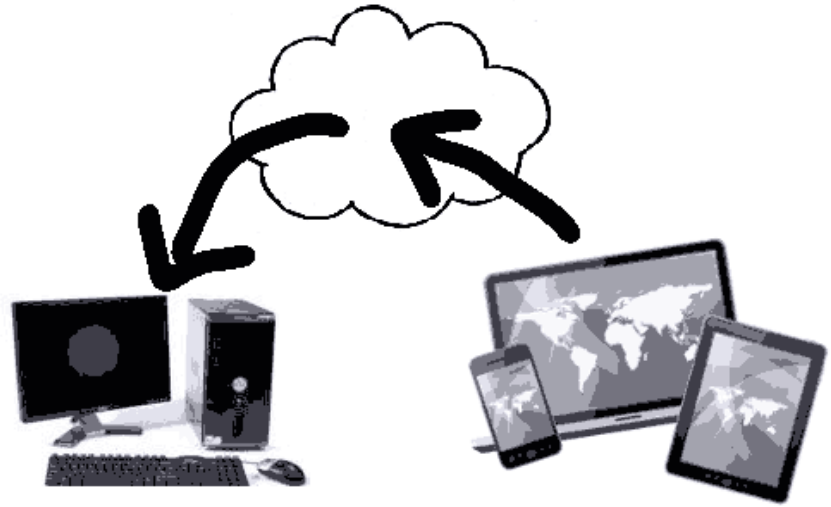
- ACID
  - Atomicity, Consistency, Isolation and Durability guarantee that database transactions are processed reliably
- CAP Theorem
  - The ACID paradigm cannot simultaneously guarantee consistency, availability and partition tolerance (distributed system)
- BASE
  - Basic Availability, Soft-state, Eventual consistency

http://thebuildingcoder.typepad.com/blog/2014/05/views-displaying-given-element-svg-and-nosql.html#5

Data Source, Repository and Consumer Client

- BIM – Building Information Model
- Cloud-based data repository
- 2D rendering on mobile device

Real-time Edit Triggers Database and BIM Update



- Graphical room editor on mobile device
- Update cloud database
- Reflect real-time changes in BIM

CouchDB Database Implementation

- Everything is a document
- All documents are JSON
- Every document has built-in id and revision
- The database design is also a document
- The design defines views and attachments

BIM Model

- Model – a RVT project file
- Level
- Room
- FamilyInstance – furniture or equipment
- FamilySymbol – geometry

BIM Object Graphics

- Room has boundary loops and can contain holes
- FamilySymbol has a single boundary loop
- FamilyInstance has a 2D placement
  - Translation
  - Rotation

BIM Object Relationships

- CouchDB ids are Revit unique ids
- Family instance → room → level → model
- Family instance → symbol

NoSQL Database Structure

- DbObj base class
- DbModel
- DbLevel
- DbRoom
- DbFurniture
- DbSymbol

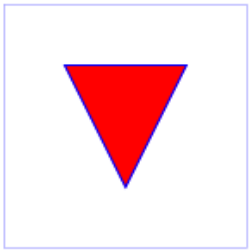
Database Object Relationships

- DbFurniture.symbolId → DbSymbol
- DbFurniture.roomId → DbRoom
- DbRoom.levelId → DbLevel
- DbLevel.modelId → DbModel

Database Object Graphics and Placement

- All graphics represented by SVG path element data

```
<svg width="4cm" height="4cm" viewBox="0 0 400 400"
    xmlns="http://www.w3.org/2000/svg" version="1.1">
  <rect x="1" y="1" width="398" height="398"
      fill="none" stroke="blue" />
  <path d="M 100 100 L 300 100 L 200 300 z"
      fill="red" stroke="blue" stroke-width="3" />
</svg>
```



JSON Symbol Database Document

- Family symbol
- Define geometry


```
{
  "_id": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f5fc",
  "_rev": "1-d575ca095533db4ccbed9f7ab2607a12",
  "loop": "M-191 922 L190 922 216 862 190 859 -191 859 -216 862 -216 919Z",
  "type": "symbol",
  "description": "FamilySymbol Furniture <390652 Table ronde a chaises>",
  "name": "Table ronde avec chaises - 01"
}
```

JSON Instance Database Document

- Furniture doc represents family instance and defines
- Relationship to room and family symbol
- Placement = transform = translation + rotation


```
{
  "_id": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f65b",
  "_rev": "1-c1b4fc969181267b55dab4c6857fc5d7",
  "roomId": "cbe571b0-0593-4350-a8e6-abf3c9239325-00061210",
  "symbolId": "11cc6e52-519e-49b2-9813-c9561b59a1fd-0005fe6d",
  "transform": "R-90T-10429,1020",
  "type": "furniture",
  "description": "FamilyInstance Furniture <390747 Canapé...
  "name": "Canapé 3 places"
}
```

CouchDB Views

- A view defines a map and optional reduce function
- The map produces key-value pairs
- Reduce produces an accumulation

```
exports.models = {
  map: function (doc) {
    if( 'model' == doc.type ) {
      emit(doc, null);
    }
  },
  reduce: function (key, values, rereduce) {
    return sum(values);
  }
}
```

Room Editor Views

- models
- levels
- rooms
- furniture
- symbols
- map_room_to_furniture
- map_level_to_room
- map_model_to_level

```
rooms = {
  map: function (doc) {
    if( 'room' == doc.type ) {
      emit(doc, null);
    }
  }
};
```

```
map_level_to_room = {
  map: function (doc) {
    if( 'room' == doc.type ) {
      emit(doc.levelId, doc);
    }
  }
};
```

Minimal Predefined HTML Scaffolding

- <h1>Room Editor</h1>
-
- <ul id="navigatorlist">
-
-
- <script type="text/javascript" src="modules.js"></script>
- <script type="text/javascript" src="raphael-min-jt.js"></script>

- Populated entirely using JavaScript adding HTML and SVG nodes using jquery, raphael and db for CouchDB queries

Resources

Full Documentation

Please refer to the class handout documentation at

http://thebuildingcoder.typepad.com/blog/2017/10/rational-bim-programming-at-au-darmstadt.html

Old Sample Repositories

- 2D RoomEditorApp and roomeditdb

  https://github.com/jeremytammik/RoomEditorApp
  https://github.com/jeremytammik/roomedit

- FireRatingCloud and fireratingdb

  https://github.com/jeremytammik/FireRatingCloud
  https://github.com/jeremytammik/firerating

- Roomedit3dApp, roomedit3d and roomedit3dv3

  https://github.com/jeremytammik/Roomedit3dApp
  https://github.com/jeremytammik/roomedit3d
  https://github.com/Autodesk-Forge/
  forge- boilers.nodejs/tree/roomedit3d

New Sample Repositories and Demos

- RvtFader
  https://github.com/jeremytammik/RvtFader
- ForgeFader
  https://github.com/jeremytammik/forgefader
  https://forge-rcdb.autodesk.io/configurator > Meta Properties
  https://forge-rcdb.autodesk.io/configurator?id=59041f250007f5c0eef482f2
- RvtMetaProp
  https://github.com/jeremytammik/rvtmetaprop

AUTODESK®

Make anything.