Connecting any Desktop Application or Revit Add-in with Autodesk Forge and the Cloud

Jeremy Tammik
Autodesk Inc.

AUTODESK

http://thebuildingcoder.typepad.com/blog/2016/10/connecting-desktop-and-cloud-at-rtc-material.html

RTC EUROPE 2016

Session Summary

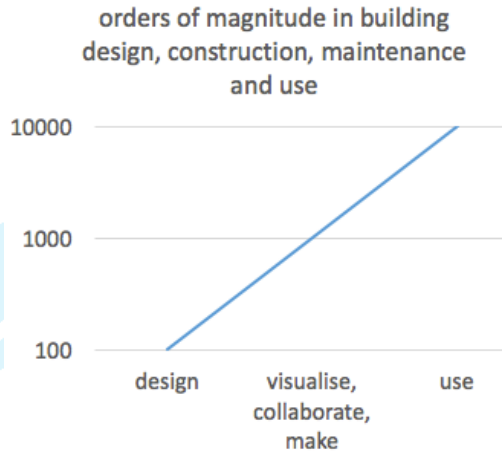
- Free your BIM data – address all BIM participants!
- Why connect? What cloud? How about security?
- It's easy, and almost everything open source
- The 2D SVG cloud-based round-trip room editor
- NoSQL databases, CAP theorem and ACID versus BASE
- FireRating in the Cloud
- Forge overview
- Forge-based round-trip BIM editor
  - Roomedit3dv3

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



RTC EUROPE 2016

Trends, Tools and Technologies

- Revit and BIM
  - Shrinking numbers of desktop computers
- A growing number of participants
  - Growing numbers of mobile devices
- Glue code, connected custom components
  - Internet, cloud
  - HTML5, SVG, WebGL
  - Forge

What Cloud? Private? Secure?

- Local and totally private
  - 2D room editor uses private CouchDB web server
- Global with Internet security
  - FireRatingCloud uses node.js web server on Heroku and MongoDB database on mongolab
- Forge OAuth
  - Roomedit3dv3 uses Forge OAuth

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

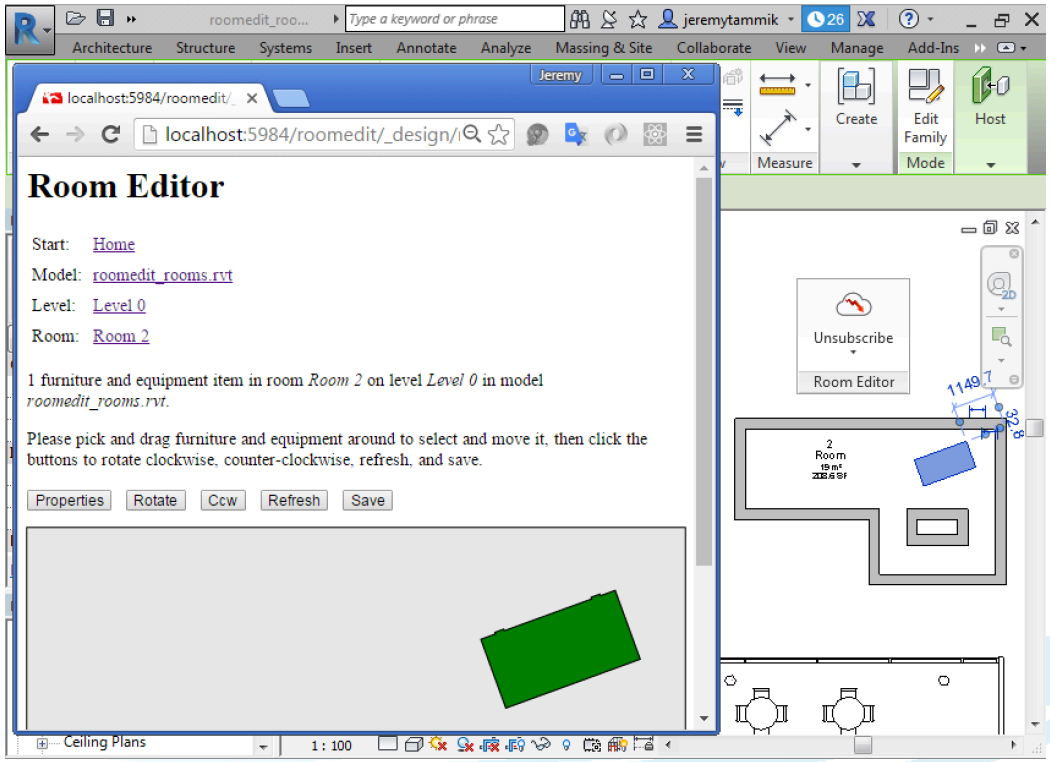
Sample Overview

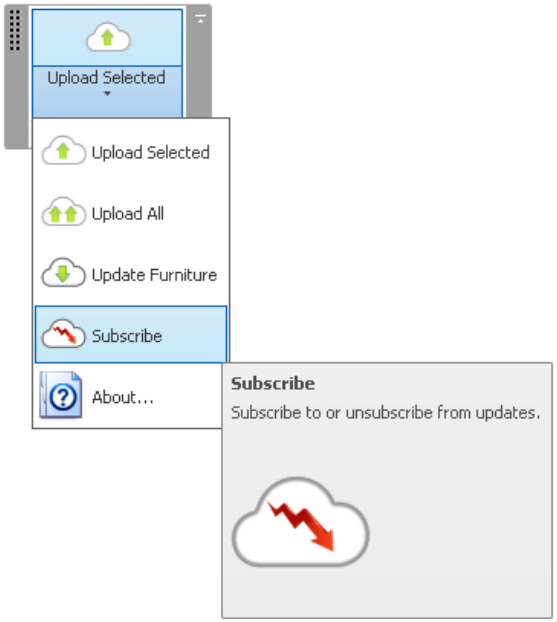
- Simplified 2D BIM room editor, SVG graphics
- FireRating in the Cloud, the simplest sample
- Roomedit3d, Forge based

Caveat

- I am not suggesting modelling in the web
- If you want to do so, talk with Autodesk
  - Revit I/O
  - http://thebuildingcoder.typepad.com/blog/about-the-author.html#5.28b

2D Room Editor





RTC EUROPE 2016

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

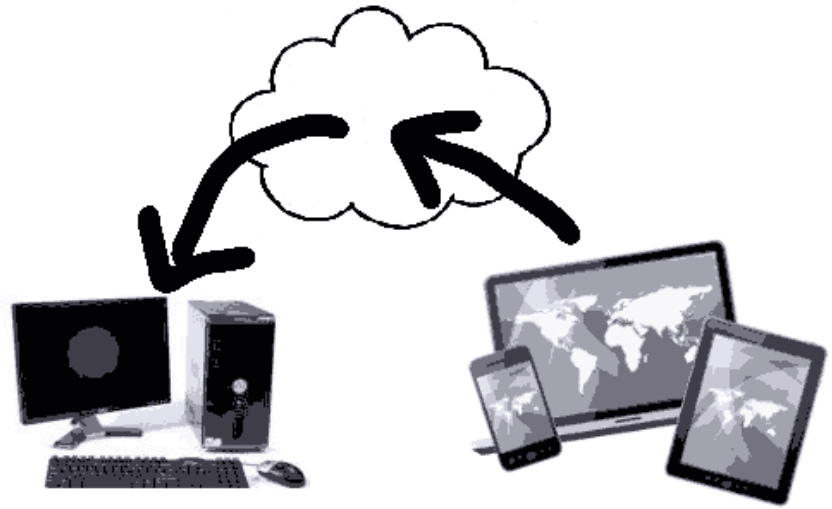
- ACID
  - Atomicity, Consistency, Isolation and Durability guarantee that database transactions are processed reliably
- CAP Theorem
  - The ACID paradigm cannot simultaneously guarantee consistency, availability and partition tolerance (distributed system)
- BASE
  - Basic Availability, Soft-state,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

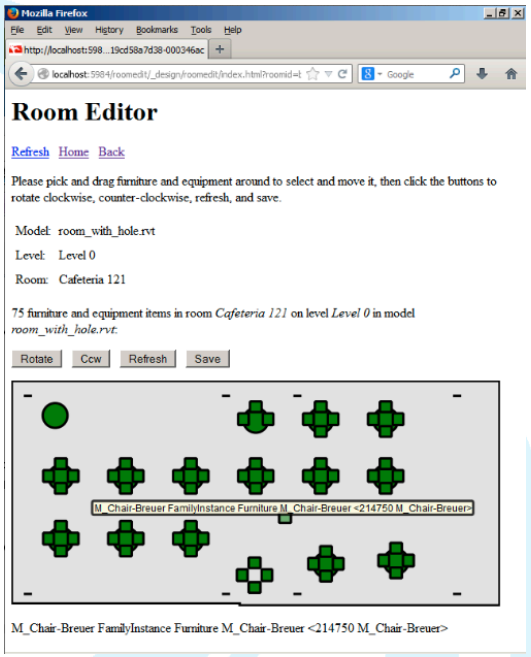


- Graphical room editor on mobile device
- Update cloud database
- Reflect real-time changes in BIM

RTC EUROPE 2016

Sixty Second First Impression Demo

- Simple navigation
  - Home page: list all models, select one
  - Model selected: list all levels, select one
  - Level selected: list all rooms, select one
  - Room selected: display 2D graphical editor, click and drag furniture
  - Furniture selected: display and



RTC EUROPE 2016

Free and Simple

- 100% open source
- 200 hours research
- < 8 hours to rebuild
  - Install components
  - Re-implement from scratch
  - Two implementation files
    - index.html (474 lines)
    - roomedit.js (358 lines)

Update Splits Functionality

- Index.html
- Roomedit.js
- One day to extract data from Revit
- One day to enhance database, editor, update etc.

Base Technologies

- BIM – Revit
- Data repository – NoSQL
- Rendering and editing – HTML and SVG

Implementation Environment

- Revit BIM – Revit .NET add-in
- NoSQL Database – Apache CouchDB
- HTML, SVG – JavaScript, jquery, db, Raphaël

Simpler Still

- Same origin policy
- Server-side scripting
- Two components instead of three

The Two and Only Projects

- Revit add-in
- CouchDB database

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

CouchDB Database Interaction

- Relax!
- All interactions are REST
- Management console is Futon
  http://127.0.0.1:5984/_utils/
- Access all documents
  http://127.0.0.1:5984/roomedit/_all_docs
- Include full doc data, not just id and revision
  http://127.0.0.1:5984/roomedit/_all_docs?include_docs=true

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

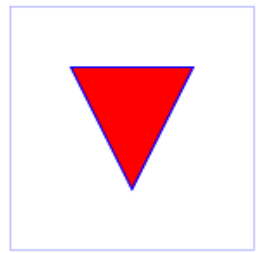
Database Object Relationships

- 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);
    }
  }
};
```

RESTful CouchDB Document and View Access URLs

- Specific document
  - http://127.0.0.1:5984/roomedit/11cc6e52-519e-49b2-9813-c9561b59a1fd-0005f5fc
- Specific view
  - http://127.0.0.1:5984/roomedit/_design/roomedit/_view/models
- Specific key in view

Entire CouchDB Application Definition

- Kanso loads CouchDB

```
roomedit/data/room_model_9.json
roomedit/index.html
roomedit/kanso.json
roomedit/lib/app.js
roomedit/lib/views.js
roomedit/raphael-min-jt.js
roomedit/roomedit.js
```

http://kan.so

- kanso.json

```
{
  "name": "roomedit",
  "attachments": ["index.html",
    "raphael-min-jt.js"],
  "modules": ["lib"],
  "load": "lib/app",
  "dependencies": {
    "attachments": null,
    "modules": null,
    "properties": null,
    "db": null,
    "jquery": null
  }
}
```

Index.html

- HTML scaffolding
- JavaScript query section
- Raphaël SVG generation and interaction
- RESTful database updates
- Called with
  - No argument: list all models
  - Model id: list all levels in model
  - Level id: list all rooms on level

Minimal Predefined HTML Scaffolding

```
<h1>Room Editor</h1>

<div id="content"></div>

<ul id="navigatorlist"></ul>

<div id="editor"></div>

<p id="current_furniture"></p>

<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

List all Models

- No input parameter
- View ‘models’, no key
- Populate navigator list

```
db.getView('roomedit', 'models',
  function (err, data) {
    if (err) {
      return alert(err);
    }
    var n = data.rows.length;
    for (var i = 0; i < n; ++i) {
      var doc = data.rows[i].key;
      var s = url + '?modelid=' + doc._id;
      $('<li/>').append($('<a>')
        .attr('href',s)
        .text(doc.name))
      .appendTo('#navigatorlist');
    }
    var p = $('<p/>').appendTo('#content');
    p.append( $('<a/>').text('Home').attr('href',url) );
    p.append( document.createTextNode( three_spaces ) );
    p.append( $('<a/>').text('Back').attr('href',url) );

    $('<p/>')
      .text( 'Please select a model in the list below.' )
      .appendTo('#content');

    $('<p/>')
      .text(n.toString() + ' model' + pluralSuffix( n ) + dotOrColon( n ) )
      .appendTo('#content');
  }
);
```

List all Levels in Selected Model

- Input parameter ‘modelId’
- Get model document itself
  - Display current selection
- View ‘map_model_to_level’ with model id key
  - Populate navigator list

List all Rooms on Selected Level

- Input parameter ‘levelId’
- Get level document itself → model id
- Get model document
  - Display current selection
- View ‘map_level_to_room’ with level id key
  - Populate navigator list

Display and Edit a Selected Room

- Input parameter ‘roomId’
- Get room document itself → level id
- Get level document → model id
- Get model document
  - Display current selection
- View ‘map_room_to_furniture’ with key room id
- Retrieve family instances → symbol ids

Nested Database Queries and Callback Functions

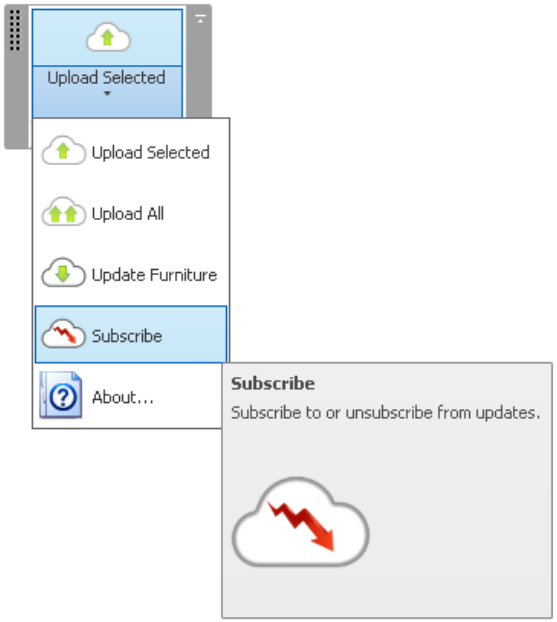
- Nested asynchronous database queries
- Each nested query depends on previous results
- Page cannot be displayed until all complete
- Nested callback functions

SVG Room Editor

- Using jquery and Raphaël
- Input room + furniture populated with symbol SVG path
- Determine size and aspect ratio
- Instantiate paper = canvas
- Place room and attach tooltip events mouseover + out
- Place furniture and attach identification, drag and tooltip events

Revit Add-in

- Determine 2D boundary polygon loops
- Upload model to database
- Download changes from database
- Subscribe to real-time updates
- Timer and external event



RTC EUROPE 2016

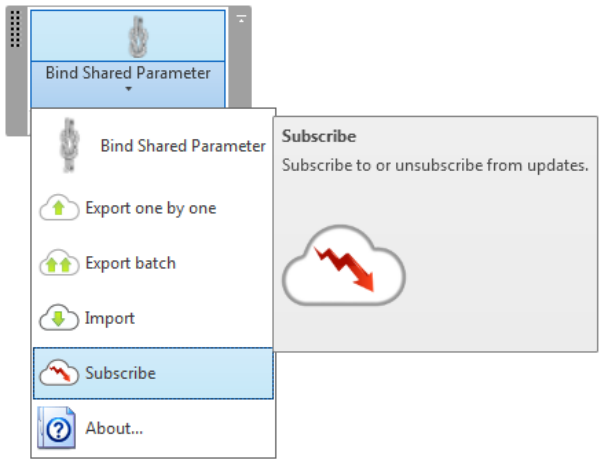
Revit Add-in Interaction with CouchDB

- How does the Revit add-in access CouchDB?
  - Upload model data
  - Retrieve database changes
  - Subscribe to changes
- DreamSeat

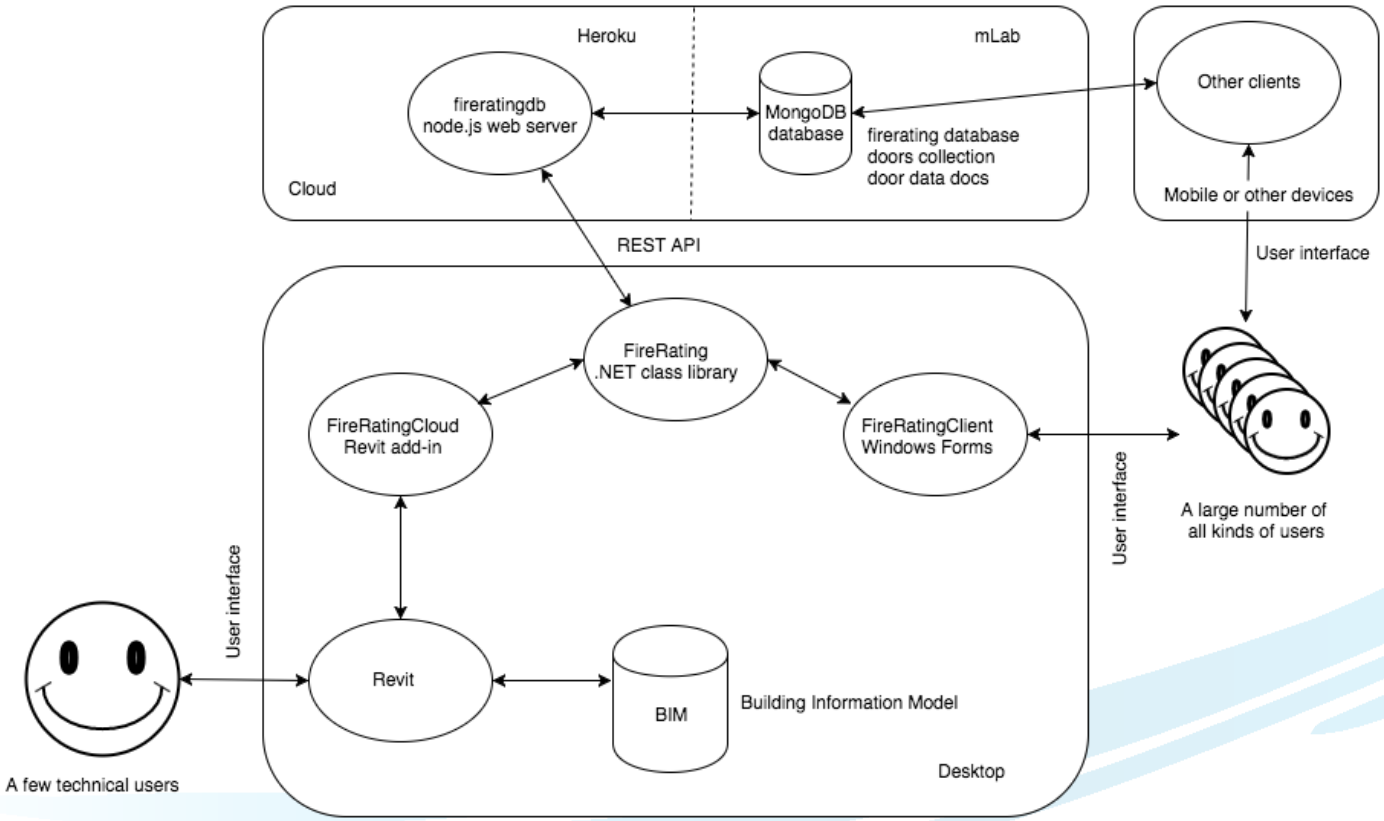
  https://github.com/vdaron/DreamSeat
  - Easy!
  - Good job!

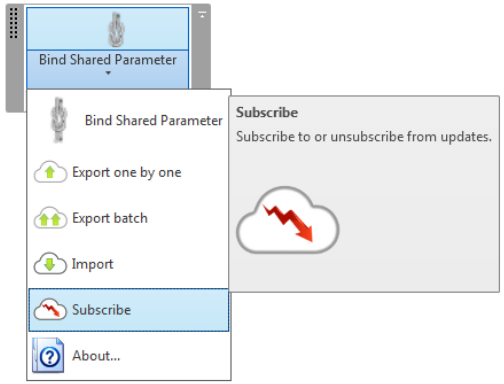
FireRating in the Cloud Commands

- 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





RTC EUROPE 2016

FireRating in the Cloud C# REST Client

- Revit add-in
- Stand-alone

FireRating in the Cloud Node.js MongoDB Server

- /a/src/web/mongo/firerating/server.js

FireRating in the Cloud Mongolab Database

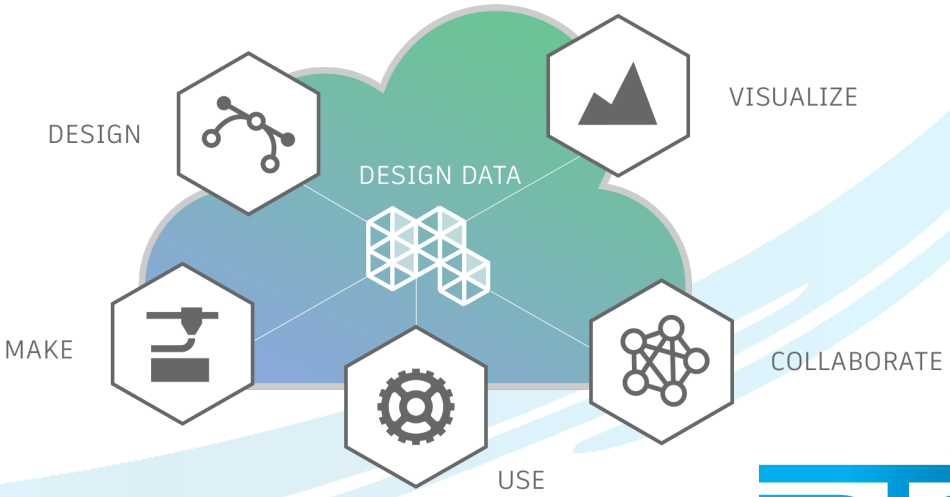
https://mlab.com

https://mlab.com/databases/firerating

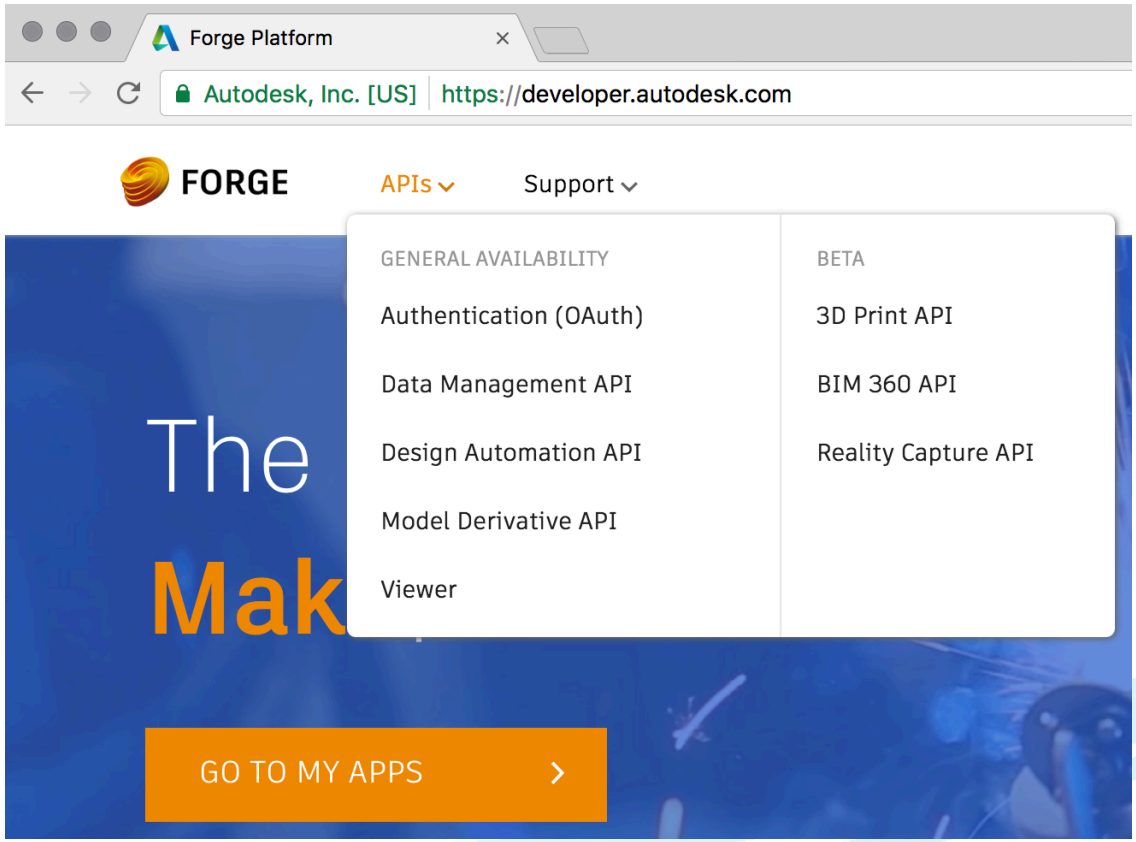
https://mlab.com/databases/firerating/collections/doors?q=%7B%22tag%22%3A%22jeremy%22%7D

Forge Platform Empowers Developers

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

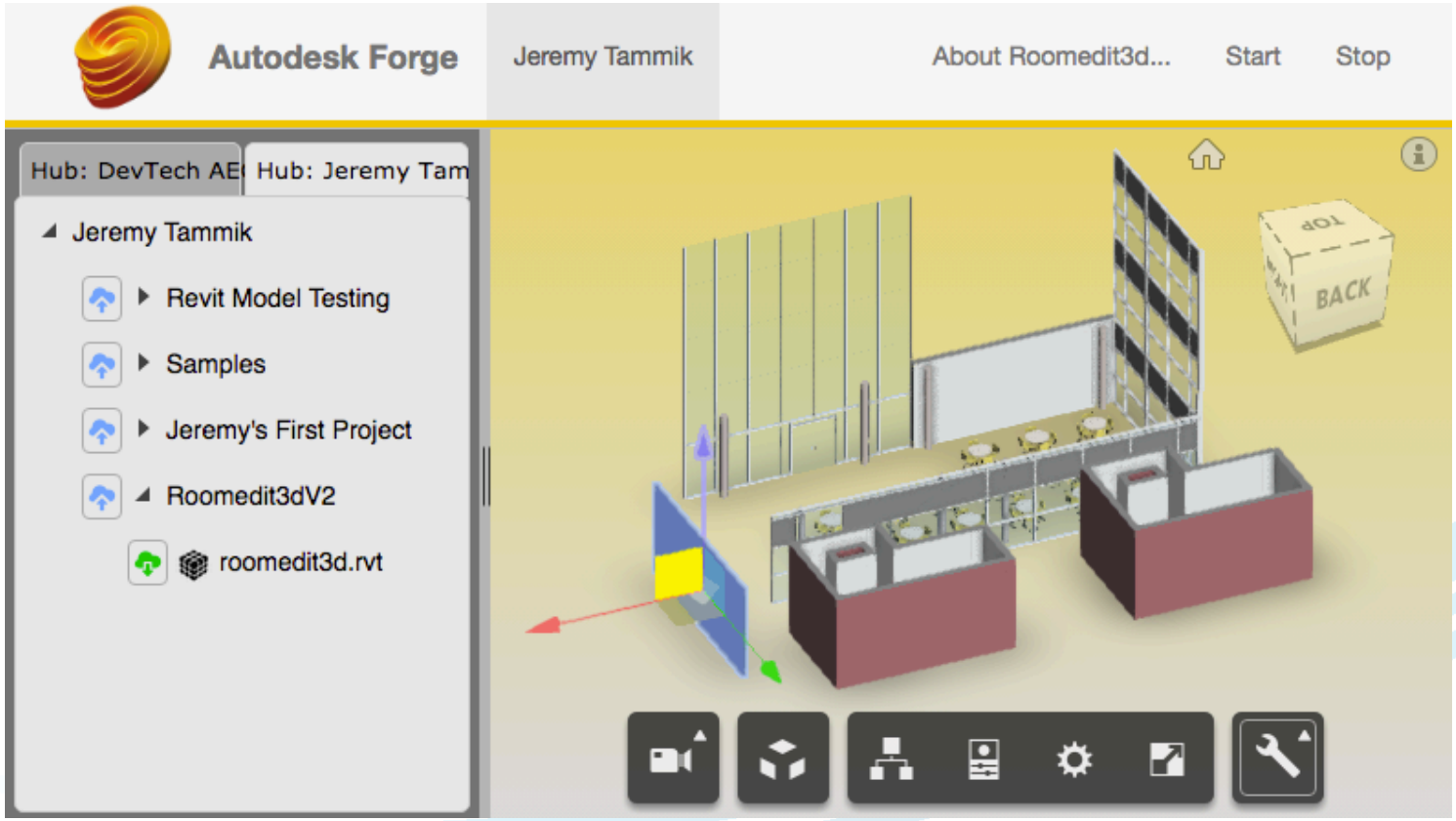


Forge Components



RTC
EUROPE 2016

Forge Based BIM Editor



RTC EUROPE 2016

Roomedit3dv3 Evolution

- Roomedit3d pre-DevCon – hard-coded
- Roomedit3dv2 for DevCon – A360, deprecated
- Roomedit3dv3 – boilerplate + a dozen lines
  - https://roomedit3dv3.herokuapp.com

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

RTC EUROPE 2016

Sample Repositories

- 2D RoomEditorApp and roomeditdb
  - https://github.com/jeremytammik/RoomEditorApp
  - https://github.com/jeremytammik/roomedit
- Properties FireRatingCloud and fireratingdb
  - https://github.com/jeremytammik/FireRatingCloud
  - https://github.com/jeremytammik/firerating
- Forge Roomedit3dApp, roomedit3d and roomedit3dv3
  - https://github.com/jeremytammik/Roomedit3dApp
  - https://github.com/jeremytammik/roomedit3d
  - https://github.com/Autodesk-Forge/
    forge- boilers.nodejs/tree/roomedit3d

Questions?

Session 1.4 – Connecting Desktop and Cloud

Jeremy Tammik

Autodesk Inc.

jeremy.tammik@autodesk.com – @jeremytammik