SD468705

There Is an Easier Way to Do That: Lessons Learned in Three Years of Forge Dev

Thiago Almeida
Blackbird Industries

Raphael Rodrigues
Onbox

Learning Objectives
- Learn how to create inclusive AECO solutions with collaboration in mind.
- Learn how to use your existing code base to power up the creation of cloud-based solutions.
- Learn how Forge can be used to deliver high-quality, feature-rich applications with less effort.
- Discover lessons learned from taking existing desktop applications to the cloud.

Description

Is your organization using BIM (Building Information Modeling) but still stuck consuming and developing desktop-based apps and unsure of how to make the move to the cloud? This class will guide you through the journey of more than three years of challenges, changing the mindset from developing single-user desktop-based BIM solutions to a more collaborative and inclusive cloud-based environment. Learn how our team overcame difficult decisions, and how Forge empowered our business to develop feature-rich apps with less effort. We will go beyond tutorials and getting-started guides to explore how you and your team can take similar steps and build a pathway to port your existing solutions to run off-premises, just like Autodesk did. We will take a tour of several Forge APIs—OAuth, Data Management, Model Derivative, and Design Automation—to build applications that are ready for this new era of automation and collaboration.

Speaker(s)

Thiago Almeida has a Bachelor Degree in Civil Engineering and in Product Design. Technician Degree in Software Analysis. Strong knowledge of engineering, mathematics and Autodesk products including official Autodesk certifications on Revit Architecture, Revit Structure, Revit MEP and 3d Studio Max. For the past 7 years started to dive into the Revit API to explore opportunities and to leverage customizable Building Information Modeling workflows for clients and partners, resulting in 6 products running in production to thousands of active users.

Raphael Rodrigues has a Bachelor Degree in Information Systems. With a full-stack background, already worked with mobile, desktop and web applications in the past. For the past 2 to 3 years have been working providing solutions with Revit Plugins and Forge based applications.
Know when it’s time to change to cloud

Cloud applications are changing the way we communicate and work, specially after the events of this year with the pandemic and everything, but even before that, the cloud movement is increasing fast and multiple companies are adopting this technology and even developing solutions in this new environment. This couldn't be different with the AECO industry, even though this is a challenging and demanding area, it is possible to provide new solutions using the cloud environment, Autodesk itself have been making this change in the past few years and releasing a powerful set of tools to boost innovations, so to answer the question “When it’s time to change to cloud?”, the best time is now. With that in mind, this section will talk about the advantages and disadvantages of cloud applications in order to make this crystal clear and maybe help companies that are thinking about making the change.

Advantages of Cloud Applications

Clouds Applications are an easy to use environment for the end users. With all evolution of technologies and people using social media on a daily basis, users already know how to logging into a web application and not only that, web development’s main focus are user experience and user interface, meaning that it empowers even the most non tech savvy users.

Another advantage is that it allows companies to take advantage of mobile devices like phones and tablets. Everyone has a smartphone in their pockets these days and clients are able to access their applications on the run, everywhere they go. Not only that, this also decreases cost because of the reduced need of expensive professional hardware, when done right, web applications can be run on almost any device with an internet connection. All this leads to a better data interaction and availability, information can be accessed when needed by the user that needs it, web applications can eliminate completely the need of transferring files via email or flash drives, changes are saved on globally available databases getting rid of Data Silos.

Companies like Microsoft, Amazon and Google provide ready-to-use infrastructure that can be accessed on demand, decreasing the need for robust physical servers and maintenance, your application and databases can all be hosted by a web server with a great set of tools to handle errors, logs, load balance and even providing a way to test different UI and see how each one perform. All that shortens the development cycle and help companies to deploy better solutions and prototypes with less time, effort and costs while providing a faster turnover. Those environments are completely controllable and scalable, companies could go from an infrastructure capable of dealing 20 users to 20000 pretty fast, changing the virtual machines memory and processing power on demand and allowing different integrations with different services at the same time, providing new business opportunities and growth.

All the advantages mentioned can be done on secure connections and with encryptions on the communication, making security another great advantage of going Cloud. The issue is that depending on how the company approaches security, it can be either a great advantage or the worst disadvantage of all, but we’ll talk about the disadvantages now.

Disadvantages of Cloud Applications

Security is a huge concern, mostly because of past scandals throughout the years involving hacking invagions and data breach, leading to malicious gathering and usage of information.
From time to time we receive an email or we see on the news about those attacks and about vulnerabilities that can be exploited, so this topic is always brought to the table when talking about cloud solutions. With that concern in mind, even governments are trying to mitigate this by increasing regulations on cloud services and they can be quite complex, not only that, since each government can have a different approach when it comes to regulations, this can be a really difficult task to achieve.

Another disadvantage is that the change can be quite overwhelming, on the AECO area the companies are usually specialized in desktop development so changing the environment will require a lot of understanding of new technologies and philosophies when providing solutions. Investments of money and time can be an issue at first, there will be a lot of code to port, a lot of knowledge to gather and a lot of infrastructure to understand and setup to guarantee the quality and take full advantage of the cloud. Autodesk itself had to invest a lot of time and money when they ported Revit to the cloud. All of this is linked to the need of a multi-disciplinary team to handle all different areas of the solution like specialists in security, databases, load balancing, user experience and maybe even mobile specialists since the web application can be open on phones and tablets.

The last disadvantage is related to Latency and Internet Speeds, AECO solutions stack really quickly on the amount of data generated and transfer that globally can be a challenge depending on the infrastructure of the companies. This ends up being more of a concern than a disadvantage as internet connections are improving really fast around the world.

**How can Forge empower your applications**

Forge was designed and developed with our industry in mind, trying to boost all of the advantages of cloud applications to a new level while softening the disadvantages. Even though Forge is a relatively new technology, Autodesk made sure to bring all of the baggage acquired from all the years of managing and developing it’s industry leading applications like AutoCAD and Revit while improving the way that clients interact with them, with the ability to use them on-demand.

With all of the available API’s to access features, Forge adds a powerful base to our applications. Imagine to have to write a web application to manage your AECO projects and extract information about your models from scratch, it would be a nearly impossible task, but with Forge and Autodesk’s help it isn’t. The available API’s take care of most of the work for you and Autodesk is always investing into new features and maintenance, it feels like having their engineers working for us.

The interaction with data and the collaboration are improved using Forge, instead of having to manually open your desktop app, export files and then email then to your co-workers, you can have all this setup on a web environment and trigger manually or automatically depending on your company needs. Not only that, this interaction can be done in a way that empowers non tech savvy users and allow them to see their data and the pretty picture of their work, but also all metadata that comes with it, this is a massive boost to collaboration in your company.

Imagine having to visualize an entire Revit model manually, with a couple hundred megabytes of data, this is possible in a fairly quick way in a web viewer provided by Forge. The optimization
and speed to load is granted by the way that the workflow is specified, all files are translated to
the proper format and optimized before being able to open on the viewer, this ensures that the
access to information is quick and reliable.

Since the industry is always looking for new ways to improve and evolve, developing Forge
applications is also a great way to have partnership opportunities. It is impossible to solve all
problems specially when they are from specific niches, because of that Autodesk is always
available to help us succeed creating those partnership relations in events like AU and Forge
Partner talks.

Forge has no upfront costs, meaning that your company can start developing without a huge
investment. This is super helpful because allow companies to create prototypes and test
multiple solutions for their problems, seeing what works and what doesn’t for your company.

Tour of the Forge API

This section will guide the user on the capabilities and strengths of each Forge API with the goal
to help companies to understand what to use to solve their problems. At the end of this section
we will list some useful tips and tricks that we learned along the way. Some of the topics can be
a bit confusing at first to beginners, thinking about that we recorded a demo talking about the
basics of Forge and Authentication, make sure to check it out.

Authentication API

The first API that developers need to access, it works as an entrance door to the other Forge
APIs, to keep it simple for now, let’s use the door analogy. Imagine that there is a presential
event happening and your company want you to attend to that event, so your company contact
the event organization about it and they send back a credentials card to allow you in the
building, when you get to the door they validate your credentials card and let you in. This is the
two-legged authentication, in the analogy your company is the application and the event
organization is the Forge platform, they communicate directly and the result of that
communication is an authentication token, the credentials card that will allow you to access all
endpoints and resources that doesn’t require an end user permission, this type of authentication
is called “two-legged” because the communication happen only between the app and the Forge
platform, each one being a “leg”. Continuing with a event analogy, imagine you’re going to
attend to a class inside the event but the class have limited seats and the access is limited so
your credentials card will not be enough, you will actually have to confirm that your name is on
the list and you will have to confirm your identity, if everything is confirmed correctly you will be
able to enter the room and attend to the class. This adds another “leg” to the flow, being the
end-user identity and permission, so this type of authentication is called “three-legged” and we
can accomplish that in two different ways, the authorization code or the implicit grant. In the
authorization code method, after the user authorizes the application to the forge platform, the
forge platform returns a code that can be exchanged to an Access Token and a Refresh Token
that the application can store on the database and keep the access alive, using the Refresh
token to generate a new Access token once the previous expires. On the implicit grant method,
the app just receives an Access token, meaning that once it expires the user will need to give
permission again.
Data Management API
This API allows the developer to control and manage data across all Autodesk’s cloud products like BIM360, Fusion Team and the Object Storage Service in an unified and consistent way, turning the Data Management API a mandatory tool to empower your application and to scale into the cloud environment. Let’s say your company uses BIM360 in a daily basis to control projects information and progress, the development team could create an application to generate Bill of Materials and use this API to send the files to the BIM360 project and have all information centralized in one place. The basic rule is, everytime you need to access, share and send files between your applications, the Data Management API is the way to do it, and you have the certainty that the communication will be consistent. It is worth noticing that even if your company does not use any Autodesk Software as a Service application (SaaS) like BIM360 or Fusion Team, you can still use the Object Storage Service to handle your files, and the way that it work is by creating cloud buckets to store your files, following a specific retention policy and giving you full control to create, modify, delete and download your files. There are three different types of retention policy that basically dictates the period that your file will be kept saved, Transient its like a cache storage that lasts for 24 hours, Temporary stores your files for 30 days and finally the Persistent retention policy will store your file until it’s deleted.

Model Derivative API
The Model Derivative API is the one used to handle file translations and metadata extractions, there are over 70 different file formats that can be translated to the SVF format to be opened in the Viewer on a web application or, instead of the SVF, they can be translated into other formats like DWG, FBX, IGES, OBJ, STEP, STL and be downloaded. This API also provides the ability to read those files and extract Metadata and parts of the model, generating a new OBJ file with just the selected portions of the model for instance.

It is important to notice that the developer will end up using this API directly or indirectly when preparing models to be displayed in the Viewer. The direct use would be sending requests on
the API endpoints to translate, download or extract metadata from files and the indirect use would be using a service like BIM360 Docs to host your files, they will be internally translated and will be ready to be displayed on the Viewer.

**Forge Viewer API**
This is the API that allow developers to embed a 2D and 3D Viewer on a web application to display projects and to interact with them in multiple different ways, not only that, the API offer a powerful set of tools to customize the viewer and even create totally new features and ways to interact and provide valuable data about projects.
With the Viewer API, developers can access a good amount of information about the loaded model, they can access the properties database for instance and view the properties of different parts on the model. They can navigate between multiple views, 3D and 2D in an efficient way, for example, it is possible to develop a dialog inside the viewer that lists all sheets of your project and displays it on a list with the name and thumbnail of each sheet, allowing the user to load and visualize each sheet. Another possible example is to use not one but multiple viewers inside your web application allowing to improve the last example and allow the users to view sheets on a different dialog and have both the 3D and 2D side-by-side synchronizing the selection of parts and highlighting the selected part on both viewers independently of where the user selected it. The possibilities are infinite and this API provides a big set of classes and methods to allow this to happen.

The Forge Viewer was built using a javascript library called ThreeJS, this library covers a lot of ground when it comes to graphics visualization on the browser like loading 3D models, applying materials and textures to models, setting up and running animations, playing audios, positional audios and manipulating the camera. Because of that, developers could enhance the viewer even more by developing in ThreeJS directly, just keep in mind that this adds an extra level of difficulty and Autodesk probably added everything you’ll need to create your solutions with the Viewer API, using ThreeJS directly is recommended only for very specific cases.

**BIM360 API**

BIM360 is already well known and stablished, it is a cloud platform that works like a centralized Hub for projects, containing documents, cost management, coordination, design collaboration,
analytics and more, all in one place and available globally. BIM360 is developed on top of Forge and use it internally on its features, the BIM360 Docs for instance uses the Data Management API to handle file transfers and if you upload a Revit file for example, the file will be automatically translated using the Model Derivative API.

The last image shows the Document Management UI, if the user click that .rvt file, BIM360 opens the model on an instance of a Viewer that is already integrated with other services provided like Issues and markups. You can see how much BIM360 takes advantage of Forge in the background and help users to boost their project management.
After this brief introduction of BIM360, let's talk about the Forge BIM360 API, it basically allows developers to synchronize and use the BIM360 power inside your own application. With it you can manage projects, add users, create and retrieve folders and files, manage project issues, manage checklists and so on. It is worth noticing that if your company already uses BIM360, your web application could manage your files and projects in a way that you don’t even have to pay for the translation of files, all that cost is included with your BIM360 subscription.

BIM360 API Features list.

- Account Admin
- Issues
- RFIs (beta)
- Document-related (Pushpin) Issues and RFIs (beta)
- Checklists
- Document Management
- Model Coordination
- Cost Management (beta)
- Relationships

BIM360 API Example
Let’s take a look on a working example using the BIM 360 API on the Shedmate App, the startup company that we work on. To start there is a form that collect information about a project, client name, email, address and so on, since this is just a basic web form, it can be filled by someone in the office or it can be sent to the client itself to fill the information, whatever suits the need of the company. The main point of interest of having the form separated like this is that the people using it don’t even need to know what BIM360 is, their concern is just to fill the form with the needed information about the project. Another interesting feature is the ability to collect the information about the project automatically using the Google Geolocation API, the user can fill in the address or the cliente can just select the option “Use my current location”, super accessible and easy to use.
Shedmate client information form.

<table>
<thead>
<tr>
<th>Client</th>
<th>Site Address</th>
<th>Builder</th>
<th>Job</th>
</tr>
</thead>
<tbody>
<tr>
<td>Back</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **This client is a Company**
- **First Name**
- **Autodesk**
- **Last Name**
- **Demo**
- **Email**
  - demo@autodesk.com
- **Phone**
  - 99 999 999
- **Phone 2 (Optional)**
- **Get from my location**

**Full Address**
- **San Francisco, CA, USA**

**Street Number**

Shedmate geolocation form.

- **Postal Code**

- **Country**
  - United States

- **State**
  - California

- **County**
  - San Francisco County

- **Coordinates**
  - 37.7749265, -122.4194155
After all the information is gathered, people with BIM360 access in your company will have the ability to synchronize data from Shedmate to BIM360 enabling the possibility to create the project, manage users, manage files and the Revit project. The final result is a complete project saved on BIM360 that was created entirely from simple Web-based forms.
**Design Automation API**

This API is all about Automation, as the name states, and the amount of possibilities turns this API in one of the most powerful that Forge has to offer but the workflow can be a little confusing at first. To give an introduction, this API allows the developer to use Autodesk’s core products as cloud services, meaning that they could automate almost any task that originally would need a desktop and the software installed to run. The Design Automation API workflow is composed by Activities, App Bundles, WorkItems and Engines. Activities are specifications of actions that will be executed using an Engine, on this actions the developer need to specify the number of inputs and outputs as well as the App Bundle to that will be used. The App Bundle is the module that will be used to execute functions, generally speaking this will be the DLL along with some other auxiliar files to be used for example, the App Bundle can be a revit plugin dll with some revit template files. Workitems execute Activities, the developer will need to supply all of the input data that was specified in the Activity and at the end of the process Design Automation will generate the output files in the way that they were also specified by the activity. You can think on Engines being the software that will run your action, they can be AutoCAD, 3dsMax, Inventor and Revit on different versions aswell like Revit 2020.

Using Design Automation can massively reduce costs and at the same time help the company to scale their operations, this is huge since everytime we think about scale we have to think about increased costs and this does not happen when you automate your processes with DA. On a fully automated workflow for example, the company could have software licenses only for the development team and the end-users would access and interact with the workflow using a web application from multiple different gadgets with internet access. The possibilities are endless.
Porting an existing Desktop App
In this sub-section we are going to talk about the workflow of porting an application to be used on Design Automation, but first let's take a look on an example of a Revit plugin that was ported.

![Revit addin to be ported to DA.](image)

The last image shows a set of buttons for an addin that is going to be ported to be accessed via DA, there are features there to frame and clad walls, adjust frames and collect reports, diagrams and cutlists.

![Revit addin feature example.](image)

This app allows users to go about cladding walls, there was both an auto and manual feature, users could click walls and roofs one by one or just run the command to do all in one go. After that it is possible to adjust the fine details and generate a complete Bill of Materials of the project, all of this inside the Revit UI, opening WPF windows and interacting with the model.
Web Application version of the Plugin.

The previous image is the new application created, using all Web-based capabilities, it is running on an IPad. The top priority was to have an interface that was simple but powerful to enable users without previous knowledge of Revit, while maintaining the user experience and interaction with the model. In order to do that, we decided to create our 3D Engine on top of ThreeJS, it was a ton of work but that allowed us to have our custom environment to manipulate as we want while being compatible with any device with an internet browser.
Since the application runs completely independent of Revit UI, there was the need of implementing custom versions of essential features like the ability to place windows, doors and families. One interesting thing to notice here is that even though we developed this awesome configurator, the true is that is impossible to have those sheds 100% complete without the need of fine adjustments, so in order to do this we added a feature to serialize the shed information and send it to Design Automation to create everything in a Revit project that the company can then proceede to adjustments and fine details. Not only that, but DA also enables the ability to export industry standard files like dxf and IFCs.
After the shed is created and customers are happy with the result, we are able to export all information to Design Automation and then receive a complete set of files about that specific shed.

Resulting Revit file from DA.
The final result are all needed files generated correctly and returned to the company, but with the example explained, how was it possible to port the application and what were the issues found?

The first focus point is the software architecture, at the time of this development, the MVVM pattern was adopted, it means Model View ViewModel, and the focus is to break down the code of your application concentrating each responsibility in the right place, separating UI code from the business logic code, making the application more flexible, extensible and modular. This was the first big issue faced, we couldn’t reach a perfect implementation of MVVM and ended up breaking the pattern on some features, so it was very time consuming to fine tune the business logic and coding to really fit the pattern.
This is the final result of the project, the image shows the Solution Explorer on Visual Studio, notice all the different projects with their own responsibility, developers can then use just the needed project for the environment they want to work with. Following the pattern, we have all responsibilities concentrated in their own space, so there is a Design Automation project, a Revit App project, a Revit UI project, a WPF project and etc. The core of all Revit interactions is handled by the RevitApp project, the hook with Revit UI and the ExternalApplication entry point is handled by the UIApp, and the responsible to run the design automation and handle workitems is the DA project, the only dependency that it have is the RevitAPI, there is no need to have any code related to UI on this project.

The Design Automation workflow starts with all information needed as a JSON file, then this file is deserialized into a data model, this data model is used to create site plans and all its details, after that the 3D model of the shed is created, then plans, elevations and section detailing. After everything is properly created, the needed data is placed on sheets with correct scales and the file is saved and returned as an output.

There is one important note to take from all this, the key is Modularity, having the core of the business logic separated, your solutions can access and use it independently of the environment, either the Design Automation or the Desktop app can be run with the same code base. Not only that, this unlocks the potential of scalable and testable applications, and boost the productivity of your development team.

Forge API Tips and Tricks
This sub-section of the Forge Tour will be dedicated to share some of the useful tips and tricks learned along the way of those years of development. The first one is to always check the API version at the top left corner of the documentation, sometimes the documentation defaults to
other versions, this can lead to some timeloss or confusion when looking for possible features on the API.

For the Forge API Documentation version.

Before sending API requests, check for the needed authentication token, since depending on the resource you're trying to access, you'll need to send either the 2-legged or the 3-legged tokens, the documentation can be used to easily detect which one need to be sent. The requests documentation also contains sample requests and responses, use it to help on the call setup.

Token type on Forge API requests.
There are some softwares like Postman or Insomnia that are dedicated to help setup API requests, they are super useful to test and debug requests before actually implementing them on your application, because of that is recommended that developers use them to boost productivity.

During the development of your applications, it will be common to face issues and questions that you might not know the answer to, but with the community constantly growing the chances are that other people had the same questions and got their answers on forums like StackOverflow. Always check it when you face issues or are unsure of the best way to approach a solution, and if the question was never asked before, feel free to ask, someone will know the answer.