SAFE SOFTWARE™

THE ULTIMATE GUIDE
TO EVERYTHING API

BY STEWART HARPER

TABLE OF CONTENTS

Part 4: Building a Scalable, Secure API (without writing code)
- Why should you care about creating APIs?
- When is the codeless API a good fit?

Conclusion

Ask questions and get answers from our brilliant community of customers & partners.

FME KNOWLEDGE CENTER

Q&A Forum

Ideas Exchange

Knowledge Base

FME Hub

APIs: WHAT THEY ARE & WHY THEY MATTER FOR YOUR DATA

APIs ARE REVOLUTIONIZING THE WAY WE SHARE AND ACCESS DATA.

By allowing web services to communicate with each other and existing business systems, APIs drill through data silos and open up huge possibilities for data integration.

In this section, we'll walk through what APIs are, why they should be at the center of your data integration strategy, and how you can use them to share and access data.

WHAT IS AN API?

To be a successful business, you need to provide a way for clients to interact with your product and services. For example, a restaurant might offer their customers the convenience of ordering food by phone or website for delivery, as well as dining-in.

In software and cloud technology, an API (application programming interface) is another way companies can serve their tools and services. And APIs are becoming a primary point of interaction — for good reason. APIs make it easy for software to communicate and share data. They allow partners and customers to access core business systems, whenever they want, in a stable and secure way.

For example, Salesforce provides an API that allows you to move customer data into and out of the CRM and sync with other services that use customer data, like Zendesk or Zopim Live Chat.

If a cloud company does not have an API, it will become impossible for clients to integrate the service with their business systems. In fact, the market is now so competitive, it is not whether cloud companies have an API, but their success may depend on how usable and intuitive it is.

MORE SPECIFICALLY WE'LL TAKE A LOOK AT:

1. What is an API is and why it is so important to the modern enterprise.

2. How to move data between web services, and why this is going to be the new norm.

3. An example of how we used APIs to migrate our knowledge content—over 1000 articles and 10,000 discussions—from Salesforce Knowledge to AnswerHub and Auth0.

4. How (and why) to build your own API without hosting any of your own infrastructure, using the AWS API Gateway and FME Cloud.
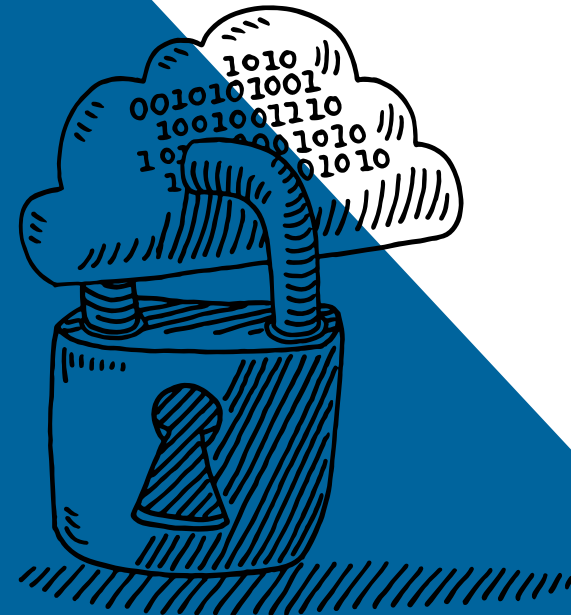
WHY ARE APIs BECOMING SO POPULAR?

APIs used to be niche technology, created by tech companies such as Salesforce, AWS and Google—the pioneers of APIs. This is no longer the case. As a result of software permeating nearly every industry and product, APIs are now mainstream. ProgrammableWeb reports there are nearly **14,000 public APIs**. There are several reasons for this:

THE RISE OF CLOUD COMPUTING pushed infrastructure and data to the cloud. This dramatically increased the requirement for APIs, both for the initial migration and integration with other systems.

BARRIERS HAVE DROPPED DRAMATICALLY for building and consuming APIs . You can now create a managed, secure, documented API without hosting any infrastructure and writing minimal code.

The rise of **MOBILE PHONES AND DEVICES EMBEDDED WITH SENSORS** compliments this services based architecture perfectly. Developers can focus on the user experience and integration with device features, while the more complex business logic can be offloaded to web services.

The jostle to remain competitive in the modern enterprise revolves around **THE NEED TO BE AGILE**. APIs give you flexibility, allowing you to quickly leverage and use the services that make sense there and then, dramatically lowering risk and allowing greater innovation.

Web technology companies that adopted an API first strategy caused **DISRUPTION OF ENTIRE SECTORS** (think Salesforce, Ebay, Amazon and Twitter). Startups copying this strategy have successfully captured market share in many sectors, leaving larger incumbents scrambling to catch up.
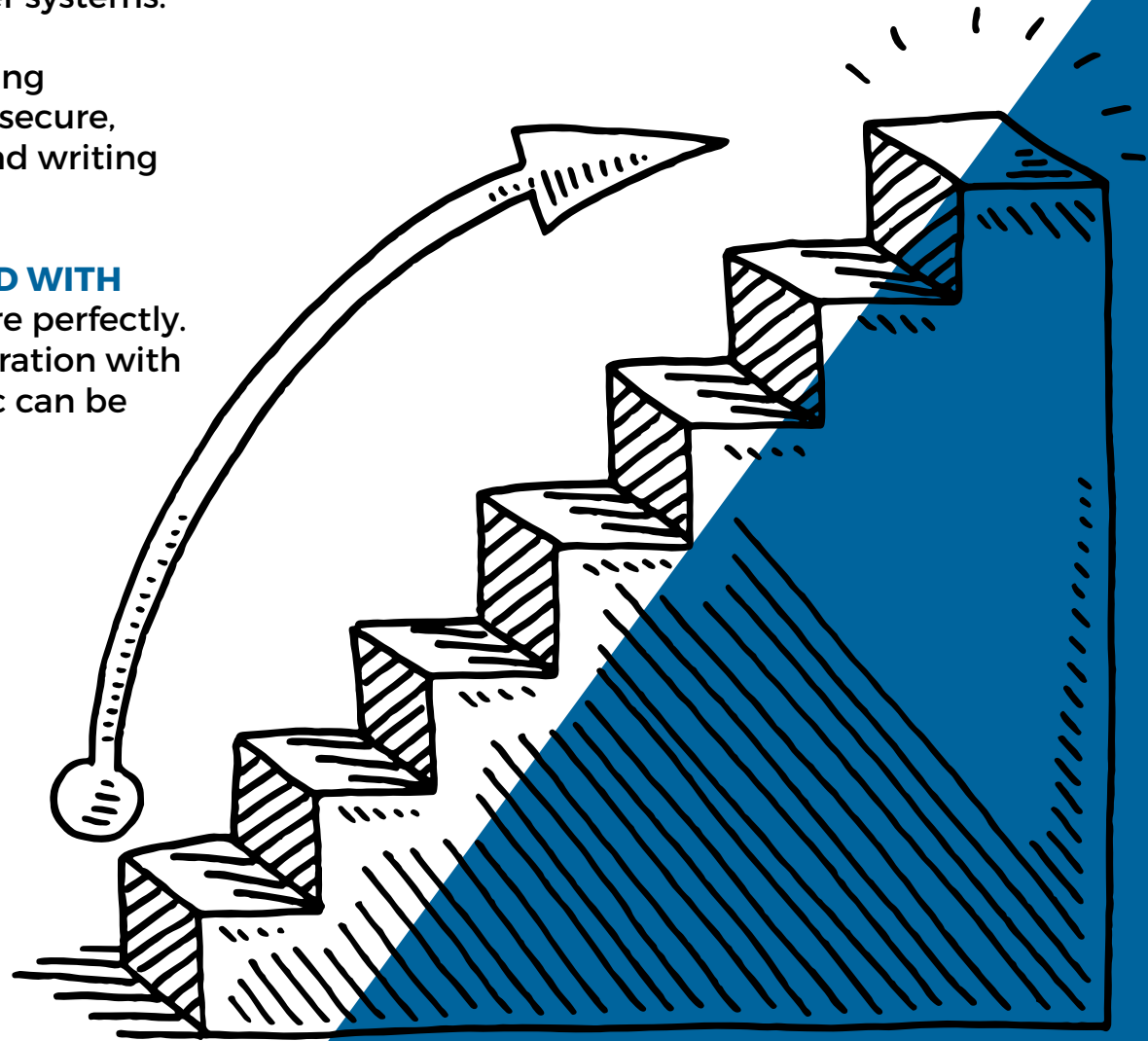
WHAT ABOUT "DISPOSABLE APIs"?

APIs can also be hugely beneficial on a smaller scale; producing internal APIs can transform and streamline internal business processes. With tools now available to create a fully functioning scalable API in less than a day, the next wave of API revolution is coming—internal disposable APIs.

These 'disposable' temporary APIs are created to solve a specific problem that might only persist for a short time—say for the duration of a migration project.
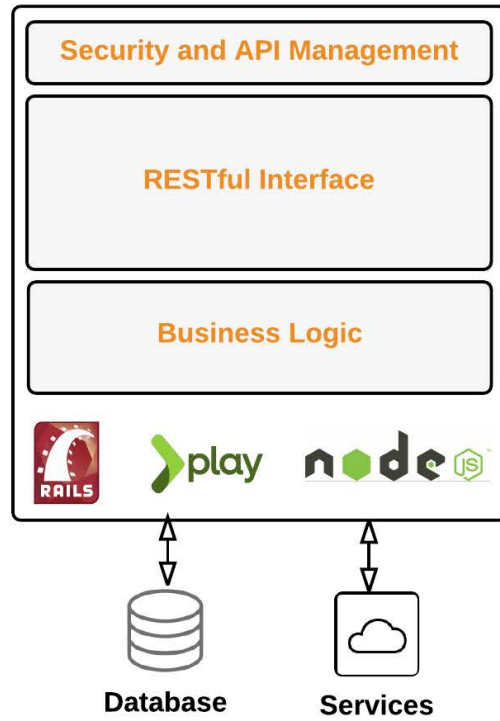
WHAT COMPRISES AN API?

Wikipedia states that an API is a set of routines, protocols, and tools for building applications. But from a business perspective, an API can be treated as a product with three core functional components:
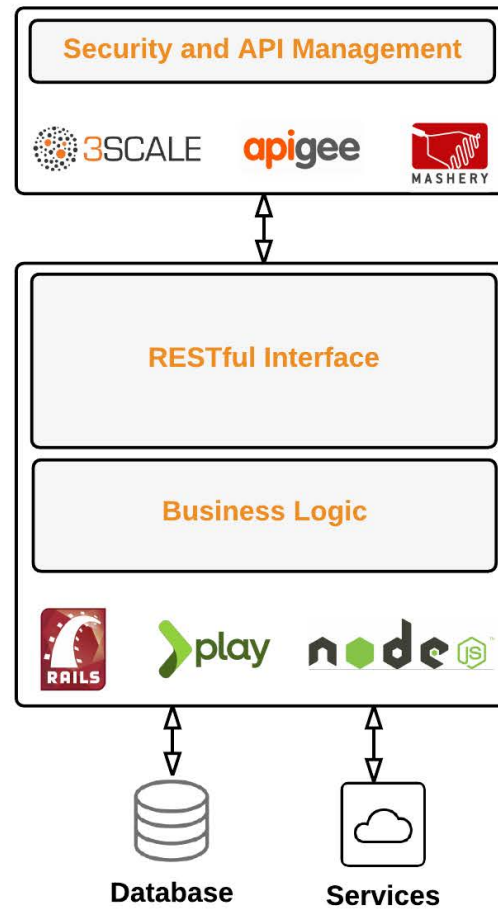
1. API management and security
2. The interface itself (resources, methods etc).
3. The business logic that is tied to each resource.

There are other important elements too—such as monitoring, analytics and threat protection—but these are not required to deliver an API, especially on a small scale.

Self-hosted API

Security and API Management

RESTful Interface

Business Logic

Database   Services

Managed API

Security and API Management

3SCALE   apigee   MASHERY

RESTful Interface

Business Logic

Database   Services

Server-less & code-less APIs

Authorization and Authentication

Auth0   Cognito

API Management   apigee   3SCALE

RESTful Interface

AWS API Gateway

Business Logic

AWS Lambda   FME Cloud

Database   Services
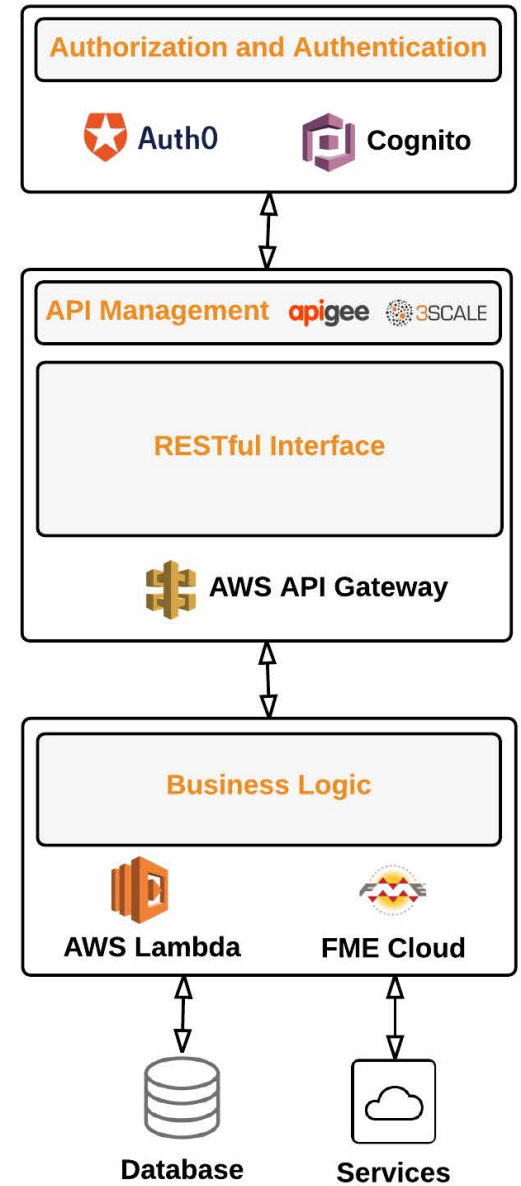
The API has evolved over the years, and companies now have a lot of choice on how they build and deploy APIs. The decision depends on the requirements of the project.

SELF-MANAGED API – This is the most flexible option, but also requires a strong development capacity and the ability to deploy, monitor and maintain a web stack.

MANAGED API – This option takes away a large amount of the pain around running a production API. You still need to create your API using your technology of choice; but the management, security, analysis and usability of the API is handled by the service.

SERVERLESS AND CODELESS API – This takes away the pain of managing and running your own infrastructure. All you have to worry about is the business logic. Authorization and authentication can be handled by a service. You design and manage the API using the AWS API gateway. Then you can either use FME Cloud and/or AWS Lambda to deliver the business logic and interact with the data.

WATCH THE WEBINAR

How to Connect to Any API (Without Writing Any Code)

Over 14,000 APIs are available publicly, making it easier than ever to connect to open data, applications, and business systems. Learn how to access, move, and integrate data using APIs with FME. No coding required.

HOW TO MOVE DATA IN THE CLOUD WITH APIs

It is now a case of 'survival of the fittest' for web services. APIs give you the ability to migrate data and build integrated systems so you can take advantage of services with new functionality or lower costs. This flexibility keeps cloud companies on their toes, and means you can use the best services for your needs.

In this section we share an example of how we used APIs to migrate our knowledge content. In Part 3, we share how and why to build your own API.

WHAT ARE THE MAIN CHALLENGES WHEN MOVING DATA IN THE CLOUD?

1. THE INITIAL BULK UPLOAD

(How to get data from an on-premise source into the cloud, or move data from an existing cloud service)

Migrating data in bulk, either from on-premises infrastructure or from another service, can take significant effort. And it's crucial to ensure as much data as possible is mapped from the original data source to the new service. Considerations include:

- Renaming attributes
- Cleaning and validating data
  - Removing duplicates
  - Truncating data
  - Removing special characters
- Merging data from multiple sources into one new schema

2. SYSTEM INTEGRATION

(How to connect new services with existing business processes once the new service is up and running)

Enterprises are leveraging web services to save time and money, but the penalty is a highly fragmented enterprise. It is therefore imperative that you are able to connect these web services. The following are important when integrating services:

- A scheduling tool that enables you to automate your connection workflows
- A cloud based deployment so you don't have to worry about managing further infrastructure
- Fault tolerance and monitoring

For example, our iPaaS platform FME Cloud leverages over 15 services. Without this combination of services, we wouldn't be able to deliver the platform as it is. This resulted in fragmentation of our data, but fortunately they each have an API so we were able to build an integrated system.

SERVICES LEVERAGED TO DELIVER FME CLOUD

WHAT KINDS OF DATA INTEGRATION TOOLS ARE AVAILABLE TO WORK WITH APIs?

POINT-TO-POINT SOLUTIONS

With the explosion of web services came the explosion of tools to help you move data between services. The majority of these tools provide point-to-point integration, a solution that solves one specific challenge. There are, however, significant limitations with point-to-point solutions.

1. LIMITED DATA TRANSFORMATION CAPABILITIES
The data transformation component is often baked into the connector with little or no control over transforming the data as it moves between applications. This puts you at risk because if you wish to transform the data in a non-standard way, you will need to do custom development.

2. SHORT TERM SAVINGS, LONG-TERM PAIN
Once you have spent time training staff in how to use the software, there is no way to reuse that knowledge and apply the logic to other integration workflows.

3. MAINTAINABILITY
In the short-term, point-to-point offers an enticing promise of lower IT maintenance costs, but as more one-off integrations pile up, complexity and costs rise dramatically.

4. MONITORING
As you add more and more point to point integrations it becomes increasingly hard to monitor and ensure reliability.

FLEXIBLE SOLUTIONS

Flexible data integration tools are single investments that can accommodate multiple new applications without users having to learn new concepts or build new components. Flexible tools also allow you to transform data as it moves, which means you can use your data exactly how it's needed. Choosing a tool that provides flexibility, like FME, is a crucial part of delivering long-term data integration architectures.

WATCH A WEBINAR

Amazon Web Services: Lessons for Architecting Data in the Cloud

Learn tips for transitioning your data architecture into the cloud. We'll explore storing data, automating data processing, and delivering data with AWS, with lessons from our experience helping clients deploy their data into Amazon. You'll discover the standard design patterns we live by, and learn best practices for Amazon services including S3, RDS, Lambda, SNS, and SQS. Plus, get a peek at the future of data delivery as we take a look at AWS API Gateway.

MIGRATING TO NEW SYSTEMS WITH APIs

At Safe, we are always looking for ways to improve our customer service, and we recently upgraded our user community to provide a new **Knowledge Center** where FME users can ask questions, access resources, and submit ideas straight to our developers all in the same place.

Our old system used a combination of Salesforce and Trello, and APIs made it possible for us to move all of our data to our new **AnswerHub** community — without losing a thing.

UNDERSTANDING REQUIREMENTS AND CHALLENGES

Migrating data between services can be a complex process. And one of the most challenging parts is understanding the data models to construct an accurate mapping. But the good news is once that's finished, flexible data transformation tools, like FME, make the actual migration very straightforward.

Three main tasks needed undertaking in our migration:
1. Moving over 1,500 knowledge articles and 3,000 Q&As from Salesforce
2. Mapping ideas submitted from customers on Trello boards to the Ideas component in AnswerHub.
3. Making it possible for users to sign-in to the whole site with one login using Auth0 for site-wide authentication. To achieve this, user information needed to be loaded to both AnswerHub and **Auth0** for sign-in to work automatically.
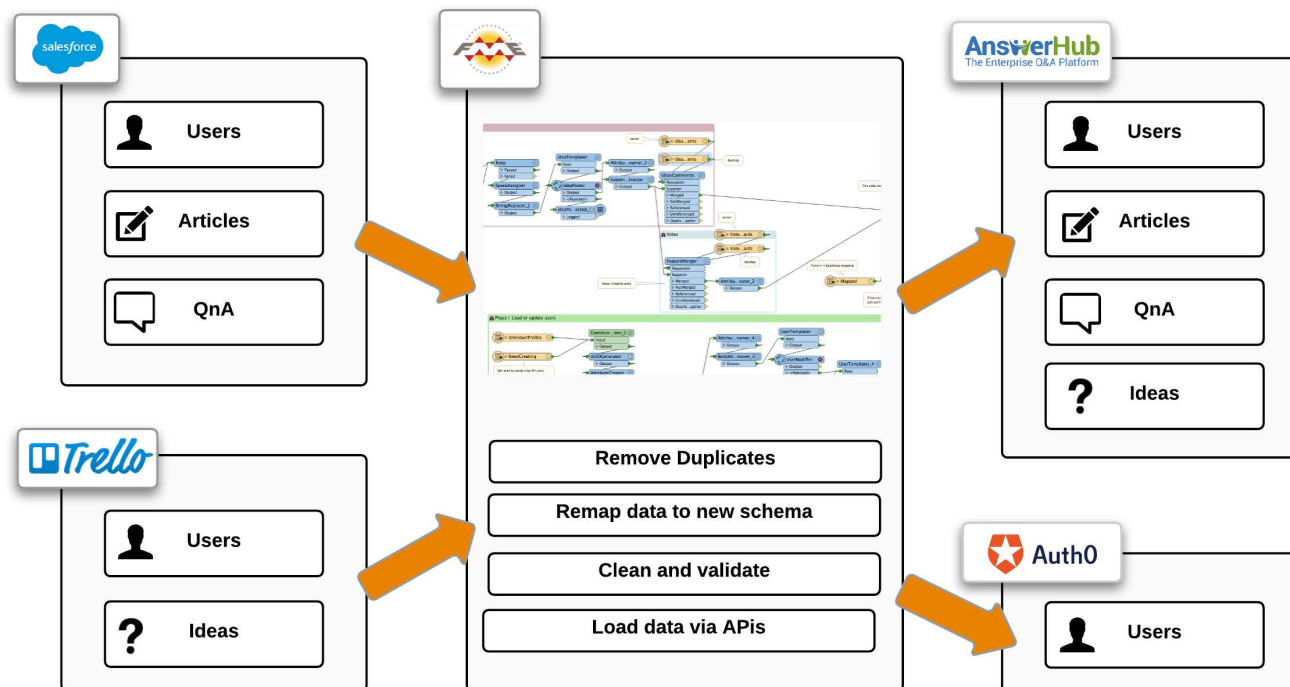
GENERAL DATA MIGRATION CHALLENGES

1) CONNECTING TO THE APIS — AUTHENTICATION

To access services, you first need to determine the authentication mechanism: token, OAuth2 or maybe HTTP Basic. And each service usually interprets the standard slightly differently. For example, with token based authentication, does the token go in the query string or in the header?

The complexities around authentication, especially if the service is using OAuth2, make it one of the biggest barriers for working with web services. But thankfully FME supports OAuth2, token and basic authentication — so once we worked out the intricacies of authentication for the three services, we were able to set and forget in our FME data transformation workflows.

2) CREATING REPEATABLE MIGRATION PROCESS

A major difference between loading data via API calls and a direct read-and-write method is that the loading process can easily become a multi-phase process. One piece of data can be loaded and the resulting object, now immediately available through the API, can be used in the next phase of the migration. This does require a bit of a shift in approach; creating a repeatable migration process is more about defining a set of steps than about mapping out an exact target dataset.

For example, once a Q&A question was loaded into AnswerHub, its URL was returned in the response header. From here, we could extract the URL and use it to post comments and answers to the question.

3) API ERRORS

API errors are a fact of life. They can be caused by network timeouts, improperly formatted requests, or various server errors. At a minimum, it's important to log these to confirm that information doesn't get lost. Ideally, it is possible to identify what caused the error and resubmit the requests with only the failed content. Again, thankfully, we were able to configure this in our transformation workflows.

DATA SPECIFIC CHALLENGES

Every migration is unique, and for this particular example, we had a different set of challenges for each set of data we had to load:

1) Loading Users

Users had to be loaded to Auth0 and AnswerHub from Salesforce and Trello. There were several challenges:

- Attribute mapping to match the new schema

- Truncating usernames if they were over 12 characters

- Removing special characters from usernames as Auth0 only supports alphanumeric.

- Testing for duplicate usernames as the public display name was not necessarily unique in the old community.

2) LOADING KNOWLEDGE ARTICLES

All existing Salesforce Knowledge Base articles needed to be migrated to the new site. Some of the challenges encountered:

- **Different handling of attachments and images:** Salesforce stored attachments and images within their infrastructure, whereas AnswerHub allows you to store images and attachments anywhere. We therefore used the migration as an opportunity to migrate all images and attachments to AWS S3.
- **Moving from category-based to tag- and search-based organization:** Work was required here to identify how to organize content under a new paradigm, but once that was done FME was used to tag everything using the API.

3) IDEAS

Ideas were pulled from two existing Trello boards into AnswerHub's Ideation system. Trello is a general-purpose tool for visual organization, while Ideation is specifically targeted to getting customer feedback, and indicating progress on implementation of ideas.

Challenges with this migration:

- Mapping the Trello users that suggested ideas with the AnswerHub users that had been imported from Salesforce.
- Identifying topic tags for the ideas
- Authenticating with Trello required creating an app profile in Trello and getting an authorization token from the Trello site to use in FME
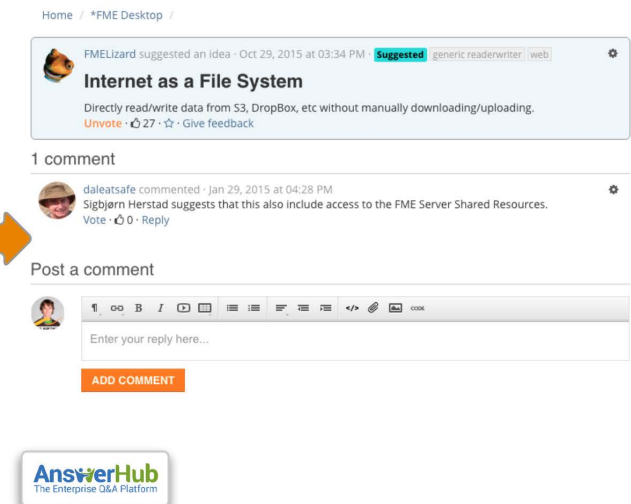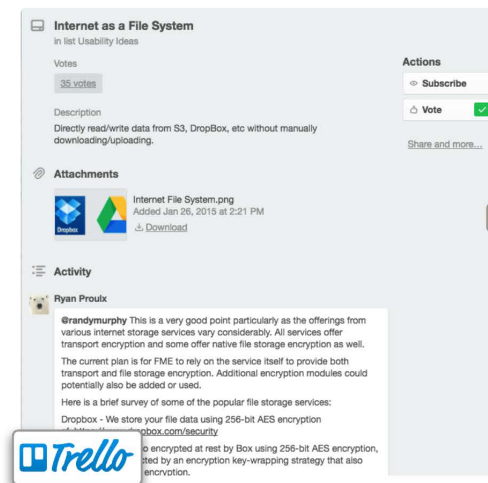
4) QUESTION AND ANSWER SITE

All existing Salesforce Knowledge Base articles needed to be migrated to the new site. Some of the challenges encountered:

- **Associating content types:** A multi-step process was required. Replies had to be associated with questions, and questions had to be associated with users, so it was not possible to load all content at once. Also, many questions and replies had images or other files attached. These needed to be loaded to Amazon S3, and the URLs integrated into the content HTML.
- **Bringing over social interactions:** We wanted to preserve as much of the flow of the original conversation, while taking advantage of the features of the new platform. This required posting content as the original user, but making use of their new ID on AnswerHub to map 'Likes' and 'Best Answers' across.
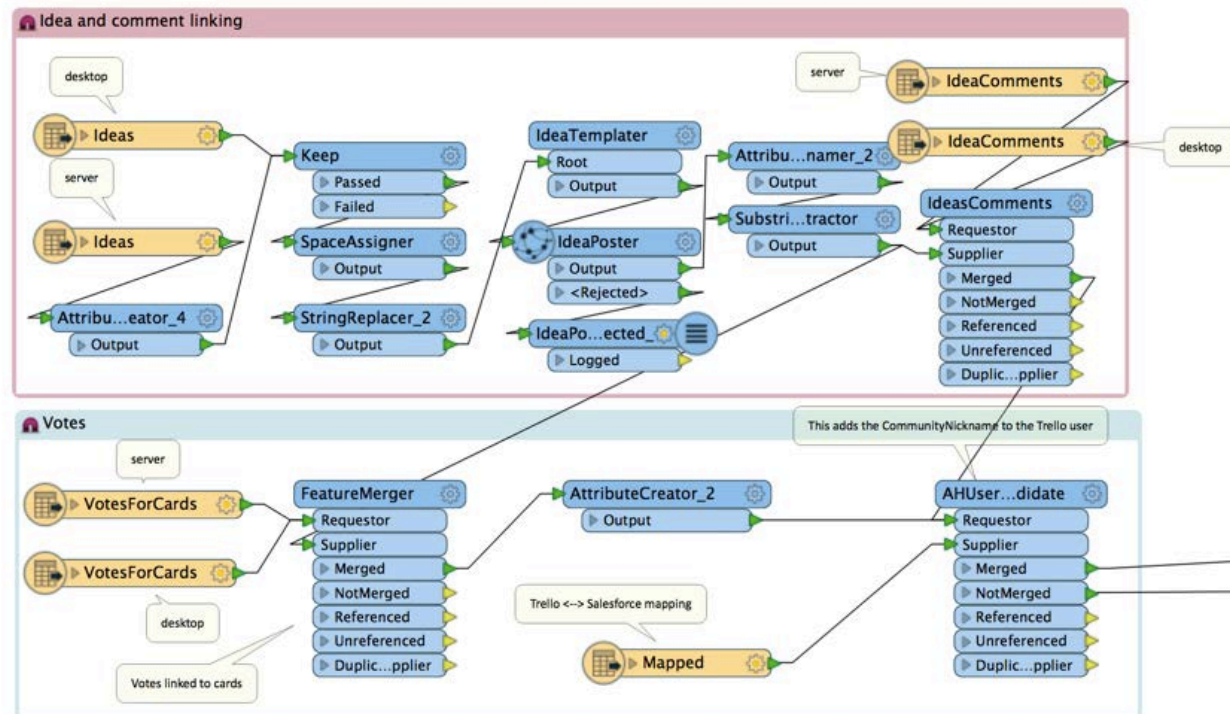
After identifying our requirements and understanding the data models, we developed several workflows in FME that connected to Salesforce and Trello via APIs, transformed the data as it was migrated, and loaded it into AnswerHub and Auth0 via API calls.

KEY TRANSFORMERS USED

FME transformers are used to read data into the workflow, validate the data and correct it. Several key transformers were used when undertaking this bulk migration:

- **HTTPCaller** – all API communication is an HTTP request – this transformer allows you to make a request to a specific URL
- **JSONTemplater** – data sent to the the APIs was all JSON, this transformer is used to generate the request body from attribute values
- **FeatureMerger** – used for linking different content types together by foreign key, and for linking up newly-created article metadata stored in spreadsheets
- **S3Uploader** – attachments and embedded images were uploaded to S3 right in the workflow
- **StringSearcher** – used for sanitizing usernames according to a regular expression pattern

CONNECTING TO VARIOUS APIs

Authentication is a crucial part of working with APIs. FME allows you to authenticate via token, OAuth 2.0 or HTTP Basic which covers the most popular forms.

While designing the migration workflow, we tested out our ideas in a staging environment. When we were satisfied everything was running smoothly, we switched the target over to the production environment. Rather than carrying out the tedious task of changing the URL, username and password in every HTTPCaller, the URL was set to a published parameter, and the authentication information was stored as a web service. This has several advantages:

- We were able to easily switch between environments
- The migration was easily be run with a different user account
- Credentials were kept separate from the migration workspace

ASSESSING THE RESULT

APIs give you the flexibility to choose the best services for your needs. In our migration example, we successfully moved data from Salesforce and Trello into AnswerHub/Auth0 and upgraded our user community to provide even better customer service.

THE KEY TAKEAWAY? You're never stuck in one system. With APIs and data transformation technology to get your data exactly how it's needed, you have the freedom to choose whichever services are best for your needs.

GET A FREE TRIAL OF FME

Join tens of thousands of data pros using FME, the data integration platform with the best support for location data.
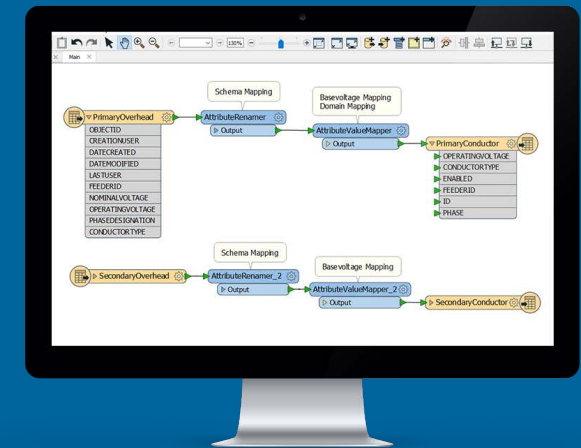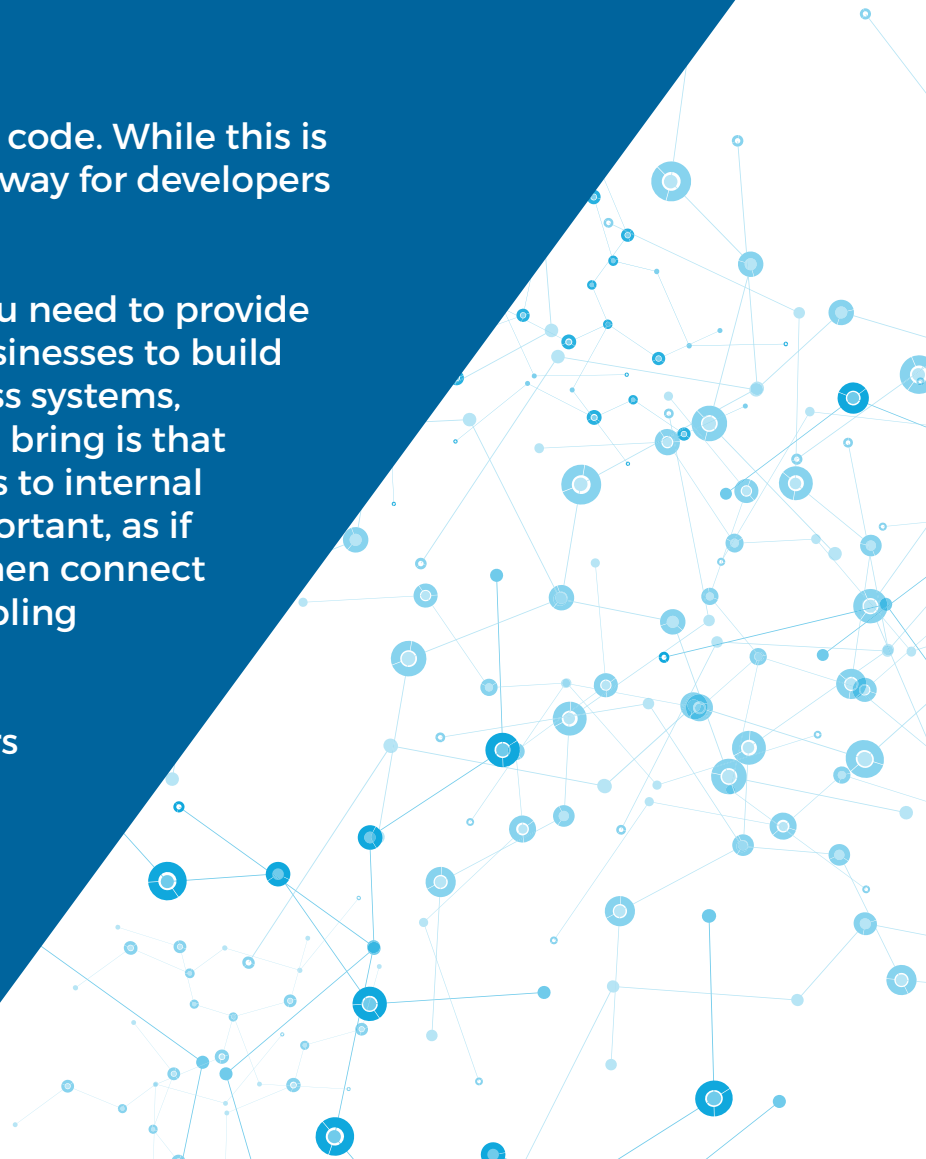
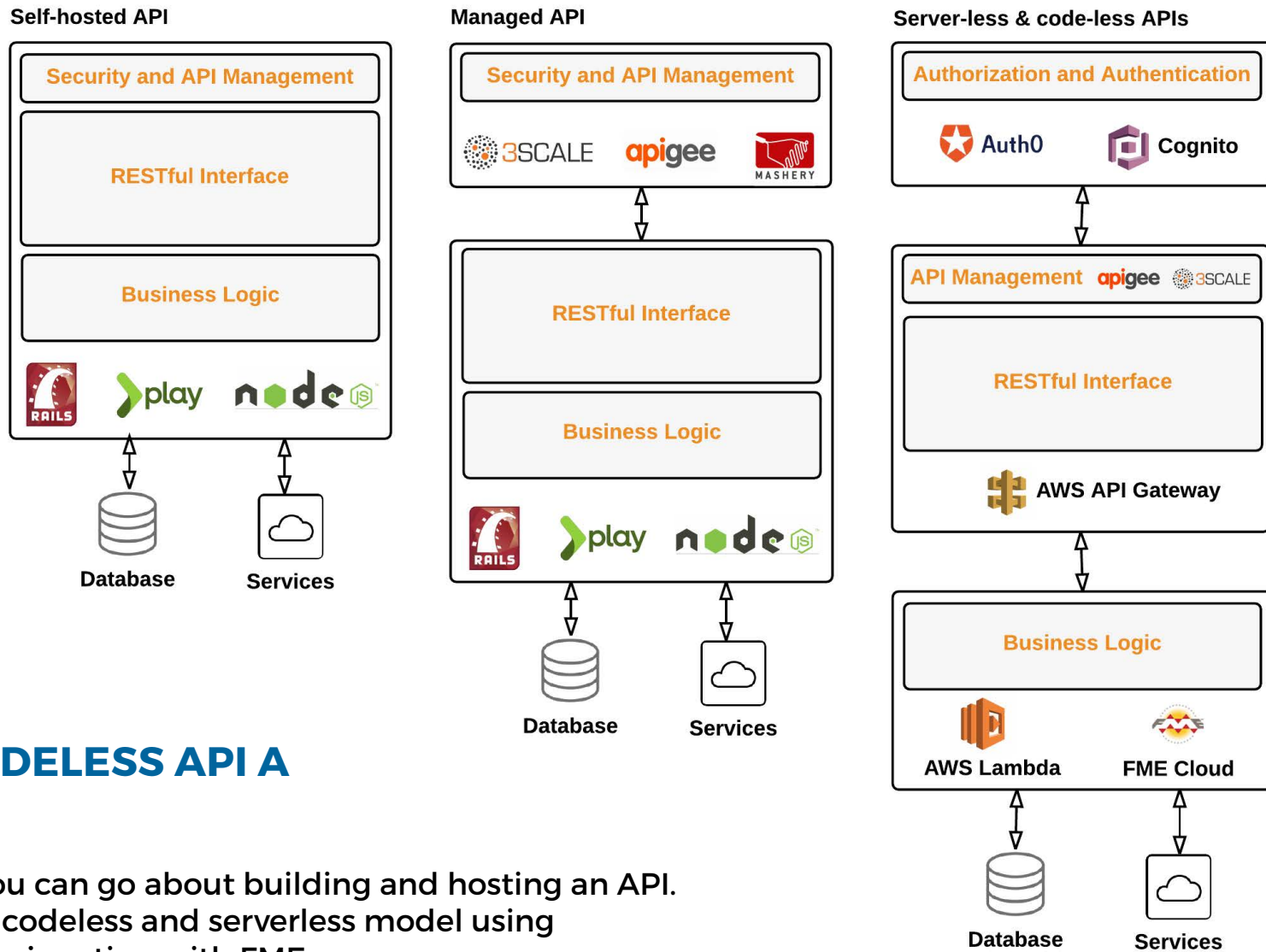# 04 BUILDING A SCALABLE, SECURE API (WITHOUT WRITING CODE)

WHY SHOULD YOU CARE ABOUT CREATING APIs?

FME allows you to connect data and applications without writing any code. While this is extremely powerful, the workflows you create do not provide an easy way for developers to interact with your data.

As discussed previously, to be successful in the modern enterprise, you need to provide a stable, clear interface to your business for developers. APIs allow businesses to build platforms that partners and customers can use to access core business systems, whenever they want, in a stable and secure way. Another benefit APIs bring is that they abstract the internal implementations, so you can make changes to internal behaviour without impacting customer implementations. This is important, as if you decide to migrate the data store, you can migrate the data and then connect it to the original API, and the user can keep using the API. This decoupling lowers the risk considerably for consumers of the API.

Creating an API has now turned into a commodity, with many vendors such as AWS and Azure providing services. The complexity therefore lies not in creating the API, but how to connect the APIs to the data. This part is not trivial and was traditionally done with code, but with FME you can connect an API to hundreds of data sources without writing any code.

Self-hosted API

Security and API Management

RESTful Interface

Business Logic

RAILS play node js

Database Services

Managed API

Security and API Management

3SCALE apigee MASHERY

RESTful Interface

Business Logic

RAILS play node js

Database Services

Server-less & code-less APIs

Authorization and Authentication

Auth0 Cognito

API Management apigee 3SCALE

RESTful Interface

AWS API Gateway

Business Logic

AWS Lambda FME Cloud

Database Services

WHEN IS THE CODELESS API A GOOD FIT?

There are many ways you can go about building and hosting an API. We are focusing on the codeless and serverless model using **AWS API Gateway** in conjunction with FME.

PICTURE: Evolution of the API, from a fully managed coded web stack to a serverless, codeless solution.

To assess if this is a good fit for your scenario, here is a checklist:

- You wish to allow developers to access your data and processes.
- You don't have access to developers and want to do everything within a GUI.Credentials were kept separate from the migration workspace
- Agility is key and you wish to create disposable APIs that might only last a short duration—say the lifetime of a project.
- You are prototyping a new service. Don't just put a website up for beta users—get an API in front of them. APIs are much stickier than web apps. If you can get beta users to integrate your solution into their workflows, you will have a higher chance of retaining them.

A serverless and codeless API is probably not a good fit if you want to create a large complex API that is going to serve a significant user base with millions of requests, as you will need more control to ensure you can optimize.

Watch the webinar recording, where we use FME Cloud and Amazon API Gateway to create a stable, scalable API on top of a PostGIS database without writing code. The code samples for the webinar are available on **GitHub**.

WATCH THE WEBINAR

How to Connect to Any API
(Without Writing Any Code)
—— Stewart Harper and Don Murray ——

How to Create an API Without Coding [Technical Workshop]

Create APIs the quick way. Find out how to build APIs on top of legacy systems, transforming data on the fly as API requests come in — without writing a single line of code. See how to use FME Cloud and AWS API Gateway so you can give your customers and other teams the ability to connect to your data and business logic.
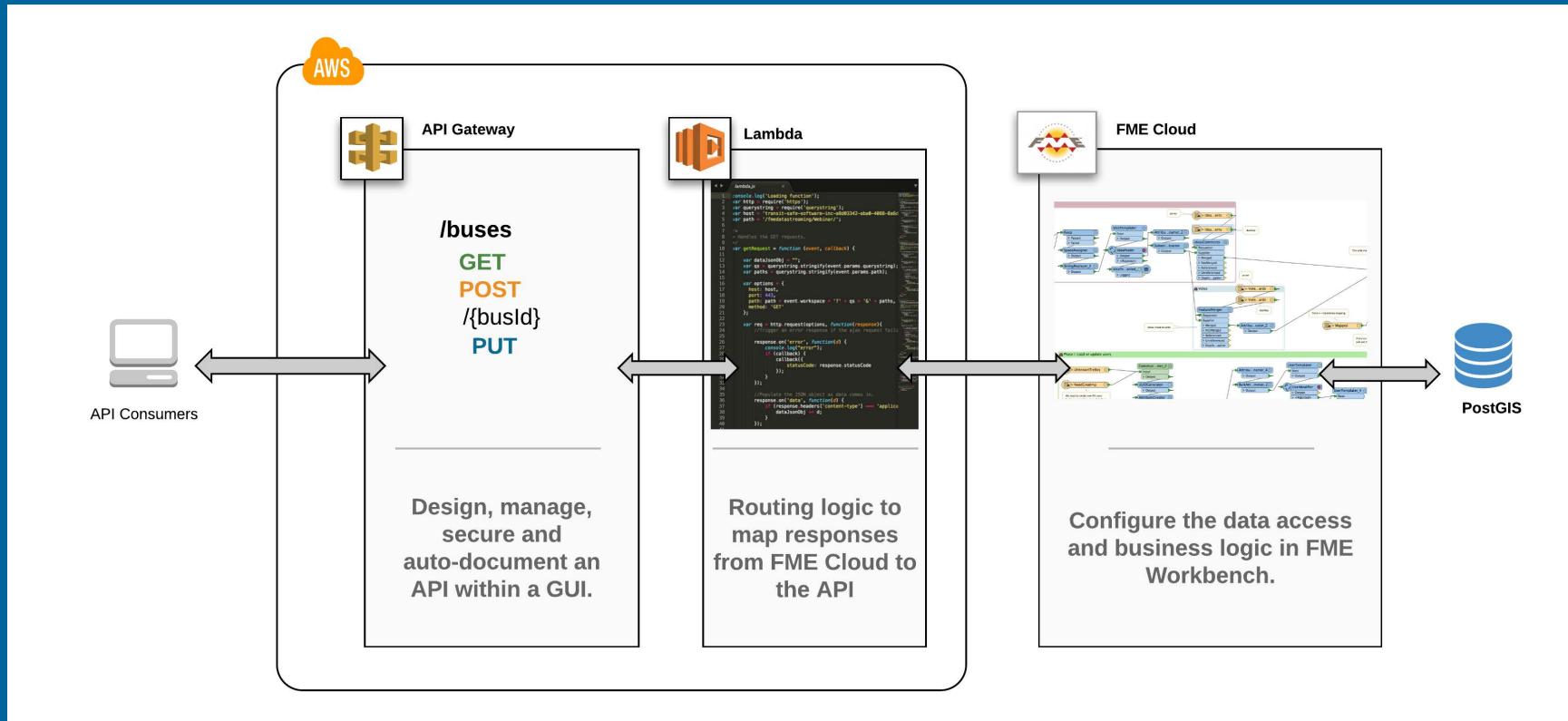
TRANSIT API WE BUILD IN THE WEBINAR.

CONCLUSION:
A DATA REVOLUTION

The need for flexibility, stability, and security makes APIs critical to the modern enterprise. In this ebook, we looked at leveraging APIs to move data between web services, plus how to build your own API to allow clients to integrate with your service. Companies like Salesforce, AWS, and Google, who adopted an API-first strategy, have led a competitive market in which success may depend on how usable and intuitive your API is.

There are various options for building an API and we hope to have given you a better idea of how to get started.

WHAT IS THE FME HUB?

FME Hub provides access to hundreds of free FME tools to improve your productivity. It also lets you privately upload your own creations for easy access by you wherever you are.

Anyone can contribute to FME Hub. Sign up for a profile and upload custom transformers, templates, custom formats, and web connections to share with the community. Items can be downloaded and installed in seconds – no sign up required.

WATCH A WEBINAR

Web Services & Web Data – Best Practices for Integration

Integrating with web services and web data is easier than ever with FME 2017. This webinar will demo FME's built-in support for web services, the concept of using the web as a file system, and how to create custom workflows to connect to any web service. You'll learn how to keep up with the fast-paced world of web-based applications and data – all without coding.

Web Services & Web Data
Best Practices for Integration

Stewart Harper and Don Murray

SAFE SOFTWARE