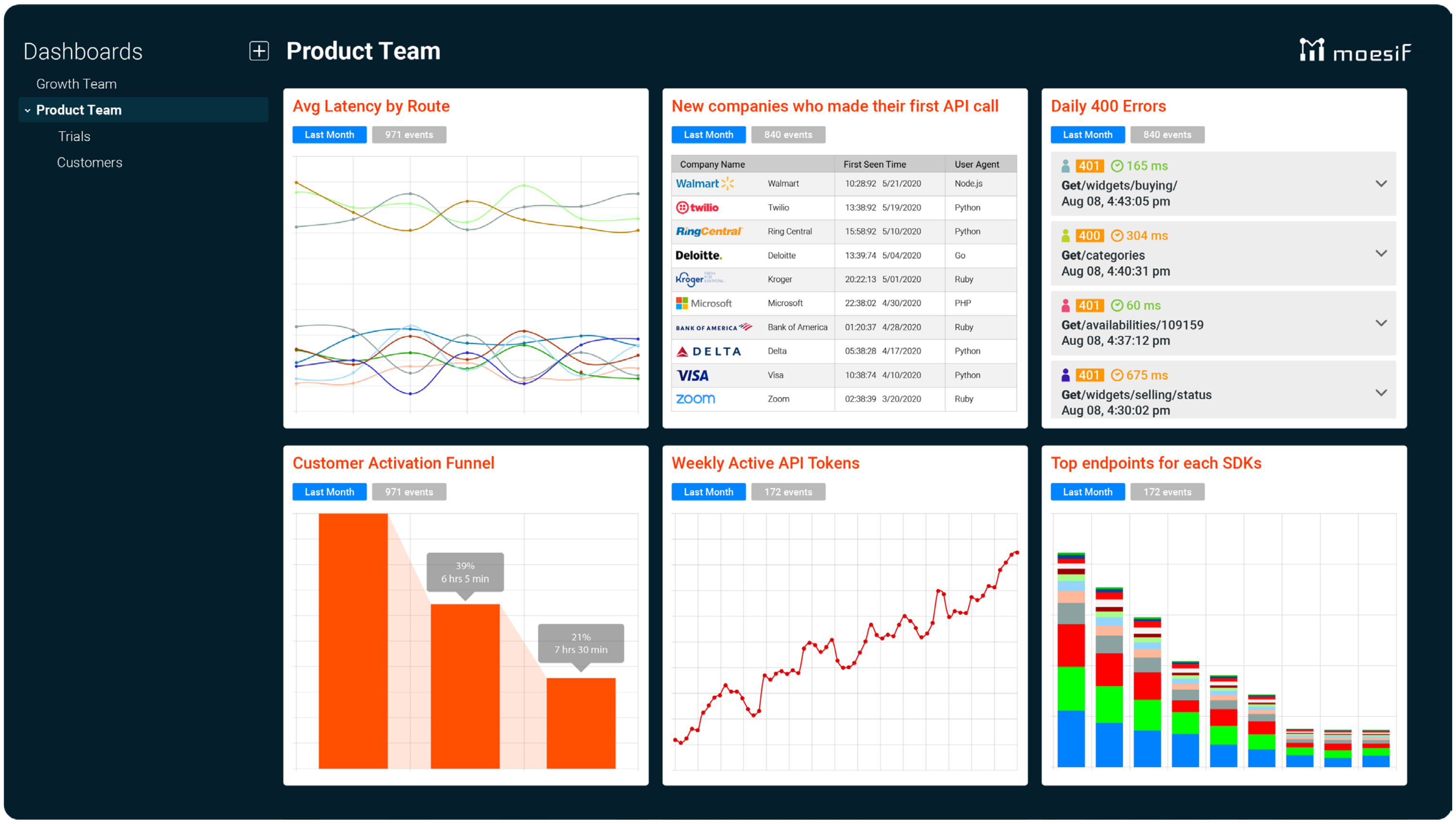
The Cost of Building an Enterprise API Analytics Platform



Company Name		First Seen Time	User Agent
Walmart	Walmart	10:28:92 5/21/2020	Node.js
twilio	Twilio	13:38:92 5/19/2020	Python
RingCentral	Ring Central	15:58:92 5/10/2020	Python
Deloitte.	Deloitte	13:39:74 5/04/2020	Go
Kroger	Kroger	20:22:13 5/01/2020	Ruby
Microsoft	Microsoft	22:38:02 4/30/2020	PHP
BANK OF AMERICA	Bank of America	01:20:37 4/28/2020	Ruby
DELTA	Delta	05:38:28 4/17/2020	Python
VISA	Visa	10:38:74 4/10/2020	Python
zoom	Zoom	02:38:39 3/20/2020	Ruby

401 165 ms
Get/widgets/buying/
Aug 08, 4:43:05 pm

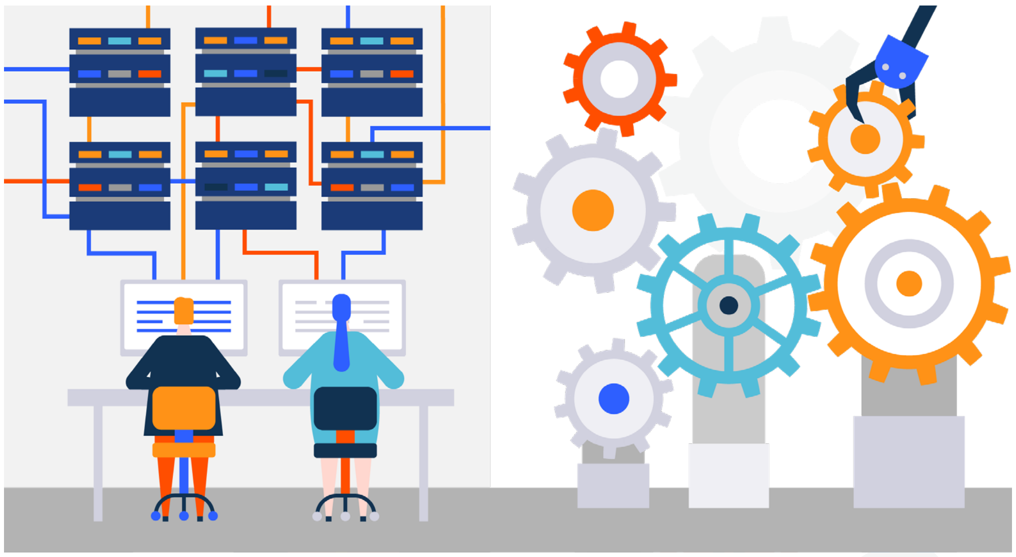
400 304 ms
Get/categories
Aug 08, 4:40:31 pm

401 60 ms
Get/availabilities/109159
Aug 08, 4:37:12 pm

401 675 ms
Get/widgets/selling/status
Aug 08, 4:30:02 pm

Deciding whether to hire and build an API analytics platform vs purchasing from a third party vendor can be a daunting task. Not only do you need to investigate ROI, you also have to navigate politics and may run into *Not Invented Here* syndrome, among other things. In the long run, by purchasing a ready made solution like Moesif, your product and engineering teams will be able to focus on what they do best: building products that customers love.

Cost Analysis of an Internal Build



Initial build Cost

The cost of building an API analytics system depends on data volume and feature complexity but can be broken down into three areas:

01 Data processing infrastructure

Designing and building a system capable of tens or hundreds of millions of API calls is not easy. It requires investment in good architecture for high-availability data collection, data pipelines and aggregation, and storing data securely. Care needs be given to performance to ensure your analytics system does not slow down your APIs or cause an outage, leading to lost revenue.

From what we've seen, the most common requirements for API analytics systems are handling 25 million API calls a month, 100 million API calls a month, and over 1 billion API calls a month. As the analytics system scales to higher volume, it can easily cost over a million dollars.

Monthly Value	Engineers Required	Build Time	Build Cost for Infrastructure
25 million API calls	6 engineers	13 weeks	$154,296
100 millions API calls	13 engineers	26 weeks	$669,616
1 billion API calls	19 engineers	33 weeks	$1,240,302

The cost of each team member is based on the national average salary for a data engineer which was $102,864 / year in February 2020 on Glassdoor.

02 Visualization, reporting, and integrations

Besides data volume, initial cost is also dependent on who is leveraging the reporting and will the data be truly actionable. Will only a few engineers be using the analytics system and are happy querying via raw SQL, or do you expect other users? Many engineering leaders looking to implement an API analytics system would like the data accessible by decision-makers across the company including product, marketing, support, and customer success. Unless the analytics system has infrastructure required to enable self-service access even by non-technical users, the data team will still become the bottleneck, slowing decision making and experimentation which can cause lost market share. This can include both a visualization tool that enables teams to create and experiment with their own dashboards and metrics, but also connectors with tools like Salesforce, HubSpot, and Segment, to make the data actionable.

There are multiple options when choosing a business intelligence tool which sit on top of your processed data in your warehouse.

User Count	Person-weeks for visualization	Connectors Required	Person-weeks for connectors	Build Cost for visualization and Integrations
10 users	5 weeks	2	6 weeks	$21,758
50 users	5 weeks	7	14 weeks	$51,431
150 users	5 weeks	21	42 weeks	$134,513

A BI visualization tool like Tableau or Looker requires an estimated 5 person-weeks to implement with your data infrastructure and set up the reporting. Each additional connector to a tool like Salesforce or Hubspot requires an estimated 3 person-weeks. We've usually seen each team of 7 requires a new connector for their tool of choice.

If the analytics system requires real-time alerting and anomaly detection, these should also be factors. If it takes 24 hours for a product owner to realize something is wrong with a customer flow, then revenue can be lost.

03 Security and compliance

If your business has security and compliance requirements, these costs should be accounted for also. This can include legal requirements such as having security audit logs and data breach detection along with mechanisms in place to comply with regulatory requirements such as General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) and ways to scrub Personally identifiable information like credit card numbers. This means you may need to involve legal and security teams to review your analytics system to ensure you're not exposing too much risk to the business.

Your company may also have other requirements for internal systems such as enterprise single-sign on and breached password detection to aid account management. Because your analytics system contains a large amount of customer data, good practices include leveraging role based access control and custom permissions to prevent data leakage along with systems in place to monitor if a breach does occur. Many visualization tools have RBAC in place, but can be easily circumvented by anyone who has direct access to the data infrastructure.

Below are typical costs for security related features:

Feature	Person-week	Build Cost for Security and Compliance
Security Audit Logs	3 weeks	$5,934
Data Scrubbing / GDPR & CCPA Support	4 weeks	$7,912
Data Breach Detection	4 weeks	$7,912

Ongoing Maintenance Cost

Once the system is live, there are still high ongoing costs for a system processing terabytes of data and usually come from three areas: Compute and storage cost, License fees for visualization tools, Ongoing maintenance and fixes.

Contrary to popular assumption, maintenance cost can grow significantly over time due to engineers moving on to other projects requiring ramp up time. Many companies implement an API analytics system to drive adoption and growth of the API business, yet this growth can directly correlate to additional compute cost. If technical debt grows, the system will soon require a follow up investment to handle the additional load by adding compute resources and leveraging newer methodologies in data processing.

01 Compute cost

From our own measurement, 100 Million API Calls requires on average 1 Terabyte in storage. If one year's worth of data history is required, 100 Million API Calls / month will require 12 Terabytes annually.

As of February 2020, AWS Redshift on-demand pricing is $0.85 per Hour for a 2TB HDD, 4 core VM and $6.80 per Hour for a 16TB HDD, 36 core VM.

Monthly volume	Storage Required	Annual Cost for Compute and Storage
25 million API calls	4 Terabytes	$14,902
100 millions API calls	12 Terabytes	$44,706
1 billion API calls	120 Terabytes	$418,308

02 Visualization cost

Majority of cost for visualization and reporting is in license fees for a Business Intelligence tool such as Tableau or Looker.

As of February 2020, each Tableau Creator seat costs $70 per month, billed annually.

User Count	Annual Fee for Visualization Tooling
10 users	$8,400
50 users	$42,000
150 users	$126,000

03 Maintenance cost

Ongoing annual maintenance is typically 25% of initial build time for a complex system such as an analytics service, but can go higher. This maintenance includes bug fixes (including critical security fixes), feature requests from various business units, upgrading outdated software, performance optimization as API traffic increases, and fixing data quality issues.

Support is not included, but this should be accounted for especially if the analytics system will be used company-wide by business users.

Monthly volume	Engineers Required	Person Days	Annual Cost for Maintenance
25 million API calls	6 engineers	16 Days	$37,980
100 millions API calls	13 engineers	32 Days	$164,582
1 billion API calls	19 engineers	41 Days	$308,196

Total Cost

First Year Build Cost

The all in first year cost includes the cost to build the data infrastructure, cost to build the necessary integrations to connect with existing tools, compute and storage costs, and any annual license fees for visualization tools.

Monthly API Call Volume	User count	Infrastructure	Visualization and Integrations	Compute and Storage	Visualization Tooling	Total First Year Cost
25 million	10 users	$154,296	$21,758	$14,902	$8,400	**$199,356**
1 billion	50 users	$668,616	$51,431	$44,706	$42,000	**$806,753**
100 million	150 users	$1,240,302	$134,513	$418,308	$126,000	**$1,919,123**

Ongoing Yearly Maintenance Cost

Ongoing yearly cost include maintenance cost to maintain the internal system includes bug fixes and security fixes, along with annual fees in compute, storage, and visualization tools.

Monthly API Call Volume	User Count	Maintenance	Compute and Storage	Visualization Tooling	Total Annual Cost
25 million	10 users	$37,980	$14,902	$8,400	**$61,282**
100 million	50 users	$163,582	$44,706	$42,000	**$251,288**
1 billion	150 users	$308,196	$418,308	$126,000	**$852,504**

Why Purchase Moesif

Companies purchasing Moesif over a homegrown build can realize 5x to 10x in cost savings in both the initial setup along with ongoing annual cost. As a best-in-class API analytics service, Moesif is constantly adding advanced features like conversion funnels and retention analysis along with leveraging machine learning to empower teams to make good decisions quickly without getting buried in metrics and reporting.

By purchasing Moesif, Engineering teams have extra bandwidth to focus on what they do best: building a great product customers love, instead of getting buried supporting legacy homegrown analytics systems.

Moesif customers also gain our expertise in growing API platforms. Enterprise customers have access to our API experts to aid your internal teams in defining their success criteria while supporting them when they have questions on how to best use the platform.

When to Build vs When to Buy

This section digs into when it makes sense to build vs buy ready-made analytics solution and provide a point based framework for evaluating API analytics solutions and perform the proper diligence.

The first decision a company should make is whether they want to build the infrastructure or purchase a ready-made solution. There are benefits and risks to both. In general, purchasing shortens the delivery of a well-polished analytics solution with lower cost in time and money compared to homegrown, but a homegrown gives you greater control over what is tracked and presented.

Answer the following questions starting with a point scale of 0.

When to buy?

Add 1 point if you answer yes to any of the following:

☐ Are you resource constrained?

Buying a ready-made analytics solution is almost always cheaper than building and maintaining a homegrown solution yourself. Vendors can amortize the R&D cost to build high performance infrastructure that's scales to billions of API calls across many customers. In addition, ready-made SaaS solutions have almost zero maintenance overhead whereas homegrown solutions accrue technical debt over time due to engineer turn-over, product neglect, and evolving business demands. According to SAP, 78% of homegrown enterprise apps are abandoned after first use. Feature and bug fixes (even critical security fixes) remain unresolved after the initial delivery due to engineering resource reallocation. If you don't have a data infrastructure team dedicated to maintaining your homegrown solution, it almost always makes sense to purchase.

☐ Are you time constrained?

Due to feature creep and unexpected onion peeling, building a usable analytics solution can drag on for months or years even for the fast moving teams. Fully-managed SaaS solutions take less than a day to get up and running. Every day your team is building analytics infrastructure is time not spent on building customer requested features that drive your bottom line growth. Buying allows you to invest your efforts into your core competencies rather than reporting infrastructure. If it takes six months to deliver your first iteration of your analytics solution, then that's a six month delay in data collection and having access to proper data causing suboptimal planning and execution.

☐ Is real-time performance important?

Do you care if new data shows up the next day or are you looking for real-time monitoring and anomaly detection capabilities? Vendors can invest larges amounts of resources to make their solution high performance and real-time which are necessary if you want real-time alerts on API and account level health. To shorten schedule, homegrown systems are usually built on existing technologies like SQL and Hadoop, where as a vendor can invest much more resources into custom-built data stores and infrastructure.

☐ Do many teams need to access the data?

Company-wide tools require strong access controls, audit logs, data management, along with a easy to use interface to make data accessible by non-technical users that a ready-made solution already has built in. A homegrown API analytics solution is usually designed by engineers for engineers, forcing non technical users to rely on making requests to an already overloaded data team. This slows down time-to-insight drastically.

In addition homegrown solutions usually don't have any security features in place due to aggressive schedules to build quickly. Majority of security incidents are due to insiders rather than third parties, sometimes inadvertently due to employees sharing passwords, employees unaware of proper IT-security policies, and no automatic threat detection of homegrown systems. Just because it's behind the corporate firewall doesn't mean it's immune.

☐ Is your team unfamiliar with privacy and compliance laws?

With regulation like GDPR, CCPA, and the upcoming ePrivacy Regulation being introduced, enterprises are now having to navigate more regulation than ever. Homegrown analytics systems rarely have frameworks and features in place to deal with things like GDPR's right to erasure and methods to stop collection of a specific user or company. Ready-made API analytics solutions from a third party vendor usually have additional features and frameworks already in place to make this much easier for their customers. After all, their main business is selling compliant API analytics software.

☐ Do you require specific integrations?

Will your API analytics solution be used by engineering only, or do you expect other teams like product, marketing, and customer success to also leverage your organization's API data? If so, each team may have their own workflows and tools of choice. For example, engineering might be using PagerDuty and Jira, product using Amplitude and Segment,customer success using Zendesk and Salesforce, and finally finance using Tableau and Slack bots. Will you be building integrations with each of these products? Otherwise you may have data silos which decreases the usefulness of API analytics and slows down your organization's productivity. A ready-made solution usually has an ecosystem of plugins and integrations.

☐ Do need expertise in data science and creating the right API metrics?

Because of their experience helping many companies leverage API analytics, vendors usually have deep expertise that they can share with their customers on best practices and ways to leverage API analytics at your organization. They're able to provide a framework to build a better platform and can point you to tutorials and other know how that can assist your organization. Homegrown systems are usually built without any input from experts in growing API platforms and what to measure. A partner with deep expertise in API analytics can help you get started tracking core metrics like TTFHW (Time to First Hello World) and walk you through best practices for API cohort retention analysis while understand what custom metrics are important for your business.

When to build?

Subtract 1 point if you answer yes to any of the following:

☐ Are you in a highly-regulated industry?

Certain highly-regulated industries like healthcare and banking have very specific requirements on where data is held and the type of data that can be collected such as with HIPAA compliance. In these cases, it makes sense to build your own system to have complete control over data and not rely on a third party vendor. While most third party vendors already follow general best practices for SOC 2 and ISO compliance and have hardened infrastructure, but may not have industry-specific compliance like HIPAA. If they do, it may require a special plan Regardless, it's imperative that you're homegrown system is also compliant. Don't fall into the trap of assuming internal systems are immune to compliance.

☐ Do you want to open-source your analytics infrastructure?

An overlooked benefit of building your own analytics infrastructure is you can also open-source it. If your company has a friendly policy for open sourcing internal tools, it can be a great way to boost recruiting efforts especially if you're trying to recruit more data engineers. Since you have full control over the analytics platform, you can do as you please. If you do plan on open-sourcing, make sure you make that decision early. Open source projects have a higher bar to code cleanliness and also need to be able to run as a single container without require many dependent services to be stood up. If your analytics infrastructure depends on many components like a SQL database, Hadoop, Spark, and Kafka, it may be harder to open-source.

☐ Do you want complete control over the look and feel of information?

Building an analytics solution gives you complete control over how information is presented to end-users. While many ready-made solutions do provide ways to customize dashboards and white-label for customer facing portals, if you have very specific requirements, you will need to build that in house.

☐ Do you have a very narrow use case?

If you have very specific use case and don't intend to grow from there, building sometimes makes sense because you can over optimize on compute and disc space. This is one of the hardest ones to decide on since it's hard to plan for unknowns. If you want to track one single metric, and know that's all you need, then buying a solution may not make sense. However, be careful of the fallacy that today's use case seems specific, since as more teams start using your homegrown solution you'll discover new ways to leverage API data at your organization.

☐ Do yo have a specific road map of analytics features and metrics to track?

Does your organization have a well defined road map of what analytics features are important now and in the future. Do you believe the vendor cannot keep up adding more advanced features? Does your team have certain trade secret metrics that no other company is tracking? If you're unsure what's the best way to measure, a read-made solution will come with a well defined framework so you don't have to think about which analytics features are important, but this can limit you in measuring very specific things.

☐ Do you have dedicated data engineering resources?

Do you have a dedicated data engineering teams and data science teams to build and maintain this? Have they already built and supported analytics infrastructure before at your organization? A dedicated team is not constantly context switching between feature development and building analytics infrastructure. This gives them more time to spend on maintenance and improvements.

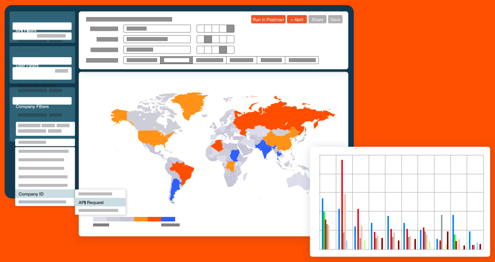
☐ Does having analytics in-house give you a competitive advantage?

If you're building and selling a payments API and your main competitive advantage is that payments can be made in every global currency and can occur even offline without internet service, it would make sense to build your payments infrastructure in-house. On the other hand, unless you're in the business of selling analytics solutions, it makes sense to partner with a vendor that already has deep expertise in analytics that can supplement your own engineering expertise rather than reinvent the wheel. The vendor can bring not only their infrastructure, but also know how on what are the best metrics to be tracking and how to track them better.

Concluding Thoughts

Building vs buying an API analytics solution can be an overwhelming process, but the best companies get the most ROI when a a methodology is applied like above. Sometimes it's recommended you try both. After all, building a solution may take 6 or 12 months but a purchased solution may take only a few hours to set up. In this case, it makes sense to purchase first, even if shorter contract to have something up and running while your company looks into what it takes to build.

The awesome thing about today's SaaS based solutions is you don't need to commit multiple years using a product. Purchase a solution, learn how the product works, and then go build it after seeing what features your team uses the most and what they don't use.



Moesif for API Engineering

Leverage user-centric API analytics to explore issues impacting your APIs and customer experience. Drill into what a user did with your API and why their experience suffered without spending hours in manual log search.