

API Resources | Best Practices in API Governance



By Janet Wagner

The number of companies integrating APIs with applications and systems is growing by the day. According to the Cloud Elements 2018 State of API Integration report, 61% of the API industry professionals surveyed think that API integration is critical to their business strategy, and 57% said that they consider their organization a platform provider.

For companies actively deploying API-driven business strategies, governance is a necessity.

Governance is especially beneficial for organizations that have an API design program or microservices architectures. This article highlights why API governance is important and covers a few API governance best practices.

The Primary Goal of API Governance: Consistency

The number of companies that consider themselves a platform provider is increasing, and so is the number of companies building APIs and applications. And the larger an organization becomes, the greater number of APIs the company will likely build and consume. Applications built by organizations often use both internal and third-party APIs.

As we explain in our article about an [API-first approach to building products](#), consistency is key. A governance process can help ensure that APIs are consistent across an entire organization. Governance can also help prevent problems such as duplicate code, tight coupling between components, unreliability, and too many services.

Do You Know Where Your APIs Are?

Kin Lane, API Evangelist, recently [published an article](#) about API governance in which he says that he often runs into companies that “don’t even know where all of their APIs are.” Lane recommends that before you can implement and maintain API governance, you need to do a number of

things. Among those things are know where your APIs are, take an inventory of your existing APIs, and create an org chart.

You need to know where you stand first regarding your APIs before you can plan where you want to go with your APIs.

What Should API Governance Include?

Once you have taken inventory of your APIs, what should API Governance include?

1. Centralization

A central point where policies are created and enforced. Usually, a team is created that handles API governance across the organization.

2. API Contract

Some companies are adopting an API-first approach to application development which we discuss in another blog article. Establishing a contract helps ensure that APIs are consistent and reusable. It also allows development teams to work in parallel- teams can create mocks, test, and deploy APIs.

3. Style Guidelines

Standardizing API design ensures that all APIs built by an organization remain consistent. One way to standardize API design is by establishing and enforcing style guidelines for all APIs. SwaggerHub features [built-in](#)

[style validation](#) so that organizations can maintain style consistency across all APIs.

4. Reusability

Standardizing API design not only ensures that APIs are consistent across an organization but also contain reusable components. And [SwaggerHub Domains](#) can help improve reusability of design resources.

5. Automation

API contracts, documentation, and tracking are processes that can be automated and should be part of overall API governance. And there are many tools available today that automate API design and governance processes. For example, SwaggerHub provides [tools for standardizing design styles](#) and managing common models.

6. Versioning

Versioning helps developers keep track of and maintain different versions of APIs. You could use versioning to deprecate an old version of an API when it's replaced by a new version. Content negotiation is another way to keep track of different versions of an API. Versioning or content negotiation could work depending on your specific situation.

7. Deprecation Policy

Versioning effectively helps make deprecating APIs much easier. An [effective deprecation policy](#) benefits the consumers of your APIs because it provides stability. Developers will know what to expect as APIs are

deprecated, and you can control when and how APIs are eventually sunsetted.

8. Tracking

Tracking the many aspects of an API should be part of your API governance. Track where APIs are deployed, who are using the APIs, how the APIs are being used, routing information, and other elements of the API lifecycle.

9. API Discovery

As the number of APIs and governance tooling grows, it becomes necessary to provide a means to easily search for and discover deployed APIs and the tooling currently in use. A method for depicting dependencies also becomes necessary for analyzing the future impact of apps consuming your APIs. For example, Capital One has implemented a [service discovery portal](#), and has enabled automatic publishing to that portal.

So Why Implement API Governance?

API governance helps save time and money because it enables consistency across APIs, allows components to be reused, and ensures that APIs are built proactively to achieve specific goals and bring value to the business. API governance also helps companies make intelligent decisions regarding API programs and establish best practices for building, deploying, and consuming APIs.

When everyone at an organization is on the same page regarding APIs, the more efficient, valuable, and successful your API programs will be.

Looking to implement an API governance policy? [Learn how SwaggerHub can help.](#)

Thanks for reading! Looking for more API resources? Subscribe to the Swagger newsletter. Receive a monthly email with our best API articles, trainings, tutorials, and more. [Subscribe](#)