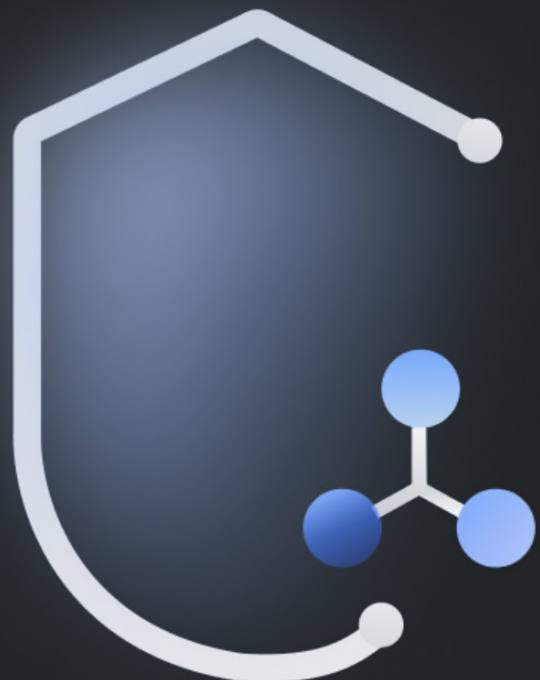


The CISO's Guide to API Governance

Best practices for developing and safeguarding
your API landscape.



Application programming interfaces (APIs) provide a framework that allows apps to communicate with one another to present you and others with information.

Every time you run an app to process an order or access information, you interact with an API. Your apps rely on these APIs for data.

Even within organizations, APIs are valuable components in business operations and they require their own governance, API governance. They make it easier for developers to use technologies, applications, or back-end systems when building the business. For example, if you want Google Maps in your application, your developer will need the Google Maps API. This will give your app access to all the API's features and data. An API's main objective is to deliver value to its users.

API Governance

Now that we've covered what an API is, let's focus on API governance. This involves sticking to a set of principles when building an API. It's crucial because different applications, organizations, and developers will use the API.

This article will take a straightforward approach to the subject. We'll start with a more in-depth definition before getting into the details. After that, we'll talk about how effective API governance benefits your organization.

Why Does API Governance Matter?

Let's go back to the Google Maps API. This API is valuable because it gives developers access to Google's real-world knowledge. At first, it was mainly in mobile phones. But as the consumer channel grew, it scaled to the point where other channels could access it—for example, a car's GPS. This led to increased revenue and innovation, which has been good for the organization. It did, however, demand changes to the API's original design.

This is what we call *API lifestyle management*. To make these changes to the API without



breaking the releases, you'll need to follow a standardized convention during development and maintenance.

Let's take this a step further, shall we? After all, in the real world, things aren't as simple as it seems in the Google Maps example.

Different consumers are using your API, and your organization is likely using a variety of APIs as well. Also, consumers of your API are using your API to create an experience for their end users. This is what we call the [API landscape](#).

Your API landscape is made up of all of your organization's existing API services as well as all of the APIs connected to them. Therefore, how can you properly manage your API landscape without compromising user experience and security? To do this, you'll need to have a defined set of principles called API governance.

API governance involves ensuring that your organization's APIs adhere to a set of guidelines. These guidelines include:

- Following a standardized approach to API documentation
- Meeting specific security standards
- Auditing across the API landscape

One of the significant benefits of API governance is the ease with which people can implement the API-first architecture approach in an organization

How Can API Governance Benefit You?

Today, your business processes likely hinge on the availability of data that you get from hundreds of APIs. This is good because they should be at the core of every business. With API governance, you can manage them responsibly.

One of the significant benefits of API governance is the ease with which people can implement the API-first architecture approach in an organization. This approach makes it easy for developers to add features and integrate their services with third-party applications.

API-FIRST ARCHITECTURE
IS A SOFTWARE DESIGN
TECHNIQUE THAT
PUTS THE API FIRST IN
BUILDING APPS THAT CAN
COMMUNICATE WITH ONE
ANOTHER EFFORTLESSLY

Designing your API with some principals in mind can help you accomplish API governance. We'll go over this in the following section.

API Governance Principles

Every organization should follow API governance principles. They enable you to create APIs whose value can be maximized across the entire API landscape. Let's look at these principals one-by-one.

Principle 1: Consistency

The primary goal of API governance is to create a consistent experience for end users. But just as user experience (UX) is important, so is developer experience (DX). Someone needs to discover, use, and develop applications using this API.

From a developer or API consumer perspective, a good DX means:

- Easy-to-use API
- API with standardized data dictionaries and versioning
- API with consistency in its endpoint and parameter nomenclature

For an API provider, this means having an *API-first mindset*. It also means *developing APIs that are consistent and reusable across the entire landscape*. This will save everyone from duplicated codes, smelly code inconsistencies, redundancy, and tight coupling between components.

Principle 2: Predictability (Managing Complexity)

Organizations appreciate predictability. And they should! It allows them to worry less and focus more on running their businesses. And predictability is one way to significantly increase API adoption across organizations.



Let's picture the API landscape. It includes organizations developing and maintaining a slew of private and public APIs for different applications with varying needs. Things will undoubtedly become complex, and disruption will happen. What matters is how often disruption happens and what occurs next.

How can you attain predictability? Focus on reliability, scalability, and maintainability.

We can define reliability as your API's fault-tolerance ability. Fault tolerance refers to a system's ability to continue operating without failure, even in the face of challenges.

Scalability refers to the ability to deal with the API's growth. This is in terms of increased loads, data volume, and traffic volume.

Maintainability refers to the ease with which a developer can make changes to an API when the need arises. For example, the developer will likely need to fix bugs, keep things operational, and add features.

Reducing complexity can greatly improve the reliability, scalability, and maintainability of your API. You risk losing your API consumers to a more agile competitor if you can't manage complexity.

Due to their interoperability, APIs are frequently the source of data breaches and leaky data in organizations



Principle 3: Security and Compliance

Technology has advanced in recent years. Unfortunately, so have **threats**. Thus, the most crucial principle to evaluate is security and compliance.

Due to their interoperability, **APIs are frequently the source of data breaches and leaky data in organizations**. To prevent unauthorized access to sensitive information, organizations must commit to a security strategy.

When it comes to API security, the first step is to understand all of the APIs exposed, either directly or indirectly, in your organization's API landscape. Here are some API security checklists we recommend you follow:

- » **Visibility into all your APIs.** As previously stated, knowing your landscape is the first step. To truly understand your landscape, you must be aware of all available APIs, including dependencies and third-party APIs. Label them, tag them, and figure out how they're related.
- » **A solid API secure design and development process.** This includes setting secure configuration standards that are appropriate for your technology stacks. It also means implementing security criteria when developing and integrating APIs. This encompasses monitoring, analytics notions, identity-based security, and network security concepts.

- » **API security testing.** Check for insecure dependencies in your API code by testing and using API discovery and risk management applications.
- » **Threat analytics.** Learn how data flows into, out of, and within your APIs. This will allow you to detect and respond to suspicious activity quickly. You should consider non-security use cases, such as API performance. [Traceable AI](#), thankfully, offers all these services.

Security and compliance is a broad topic, especially given how technical it is. To understand it better, read this ebook: [The Practical Guide to API Security](#) by [Aaron Lieberman](#), a cloud practice manager/architect at [Big Compass](#). He explains how to assess, implement, and scale your API security in detail.

Just as Google Maps API does, your APIs should drive innovation by providing value to consumers

Principle 4: Interoperability (Value and Business Alignment)

View your API as more than just dozens of code blocks. Technically, it should be able to solve a common problem. Just as the Google Maps API does, your APIs should drive innovation by providing value to consumers. Each API should also be in line with the organization's goals, as this will make its adoption much easier.

It's up to you to come up with a revenue model for your API if that's the end goal. Overall, a rich API with high interoperability value will

enable enterprises to access data. This will result in revenue generation, either through a revenue model or by allowing API consumers to build applications that match the organization's needs.

Principle 5: Quality API Documentation System

Irrespective of how valuable your API is, it'll go unnoticed and unused if potential consumers don't understand how to use it. To get your API out there, you need to have well-written and well-structured API documentation.



API consumers should be able to learn about the API's capabilities, arguments, design reviews, and integration abilities from the documentation. Documentation maintenance and updates should also be a part of the API life cycle. You can use maintenance tools and text editors for API documentation maintenance. Or you may choose to automate the process using API description formats like [OpenAPI Specification](#).

Besides increasing the awareness of your API, documentation reduces churn and saves your organization support time and costs. Proper documentation helps create a fantastic user and developer experience. It also tells how much you care about the end users.

Summing Up: Risks and Safeguards

The consequences of neglecting some of these principles can be disastrous. Service degradation, customer churn, downtime, and outages are all potential risk. As a result, your

organization's API governance should be in order, especially as it scales and gets more integrated.

To learn more about developing and safeguarding your API's landscape, remember to check out the [Big Compass Practical Guide to API Security](#).

About Traceable AI

Traceable is the industry's leading API security platform that discovers, manages and secures all of your APIs across your development lifecycle. The platform applies the power of distributed tracing and machine learning models to understand the DNA of all your applications, understand any changes, and detect anomalies in order to detect and block API attacks, making businesses more secure and resilient.

Learn more at <https://www.traceable.ai>.

Meet with a security expert

Our crack security research team is happy to meet with you to talk about your API security challenges.

Schedule meeting

About us.

Traceable AI was founded by third-time entrepreneur Jyoti Bansal and Sanjay Nagaraj. Bansal and Nagaraj saw the massive adoption of cloud-native architectures firsthand during their time at AppDynamics and founded Traceable AI as a result to protect applications from next-generation attacks.

Traceable AI applies the power of machine learning and distributed tracing to understand the DNA of the application, how it is changing, and where there are anomalies in order to detect and block threats, making businesses more secure and resilient.

Traceable.ai

