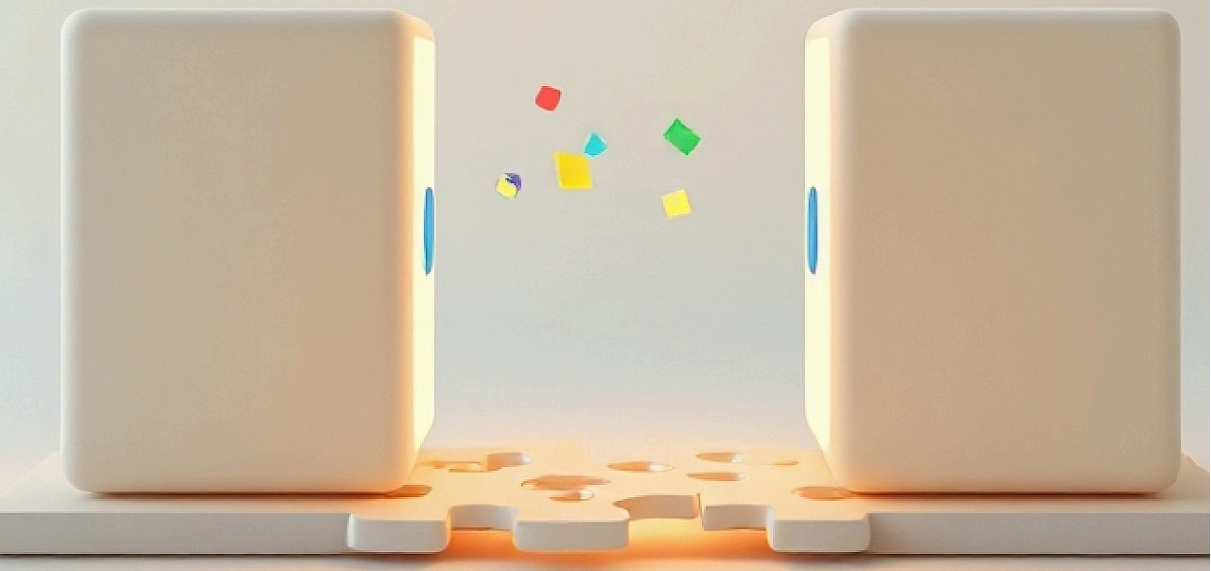
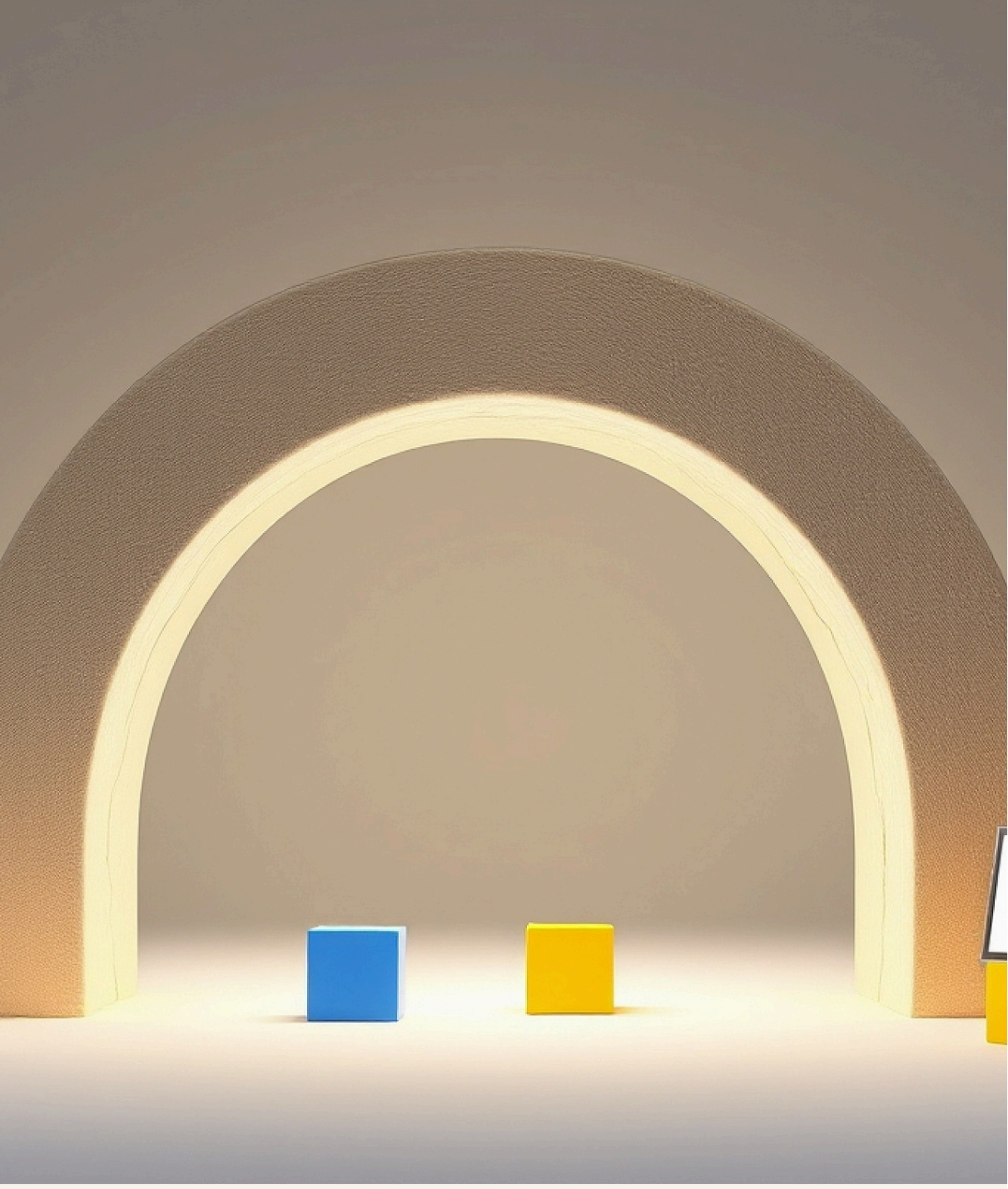
What is an API?

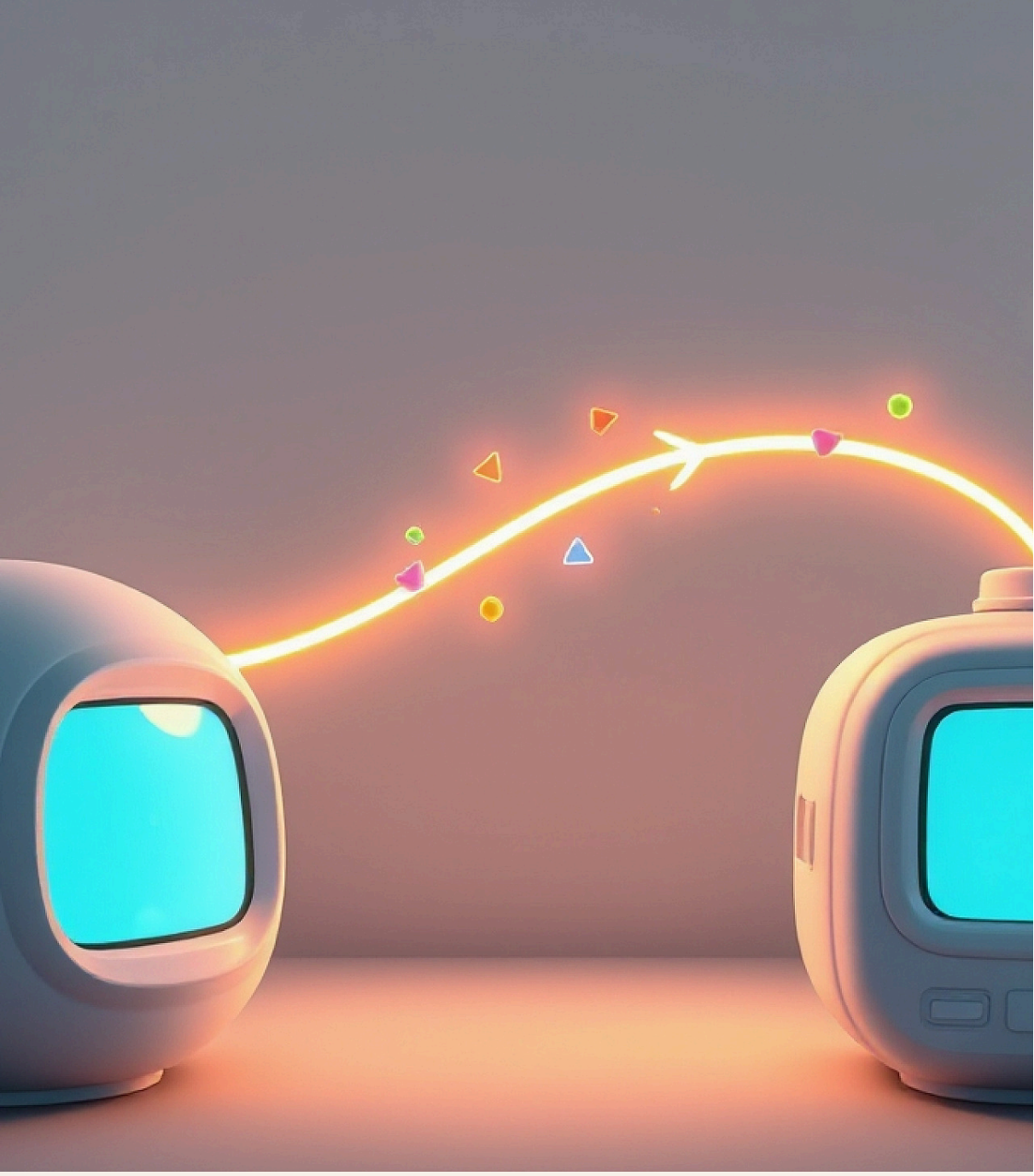


www.yzokul.com.tr

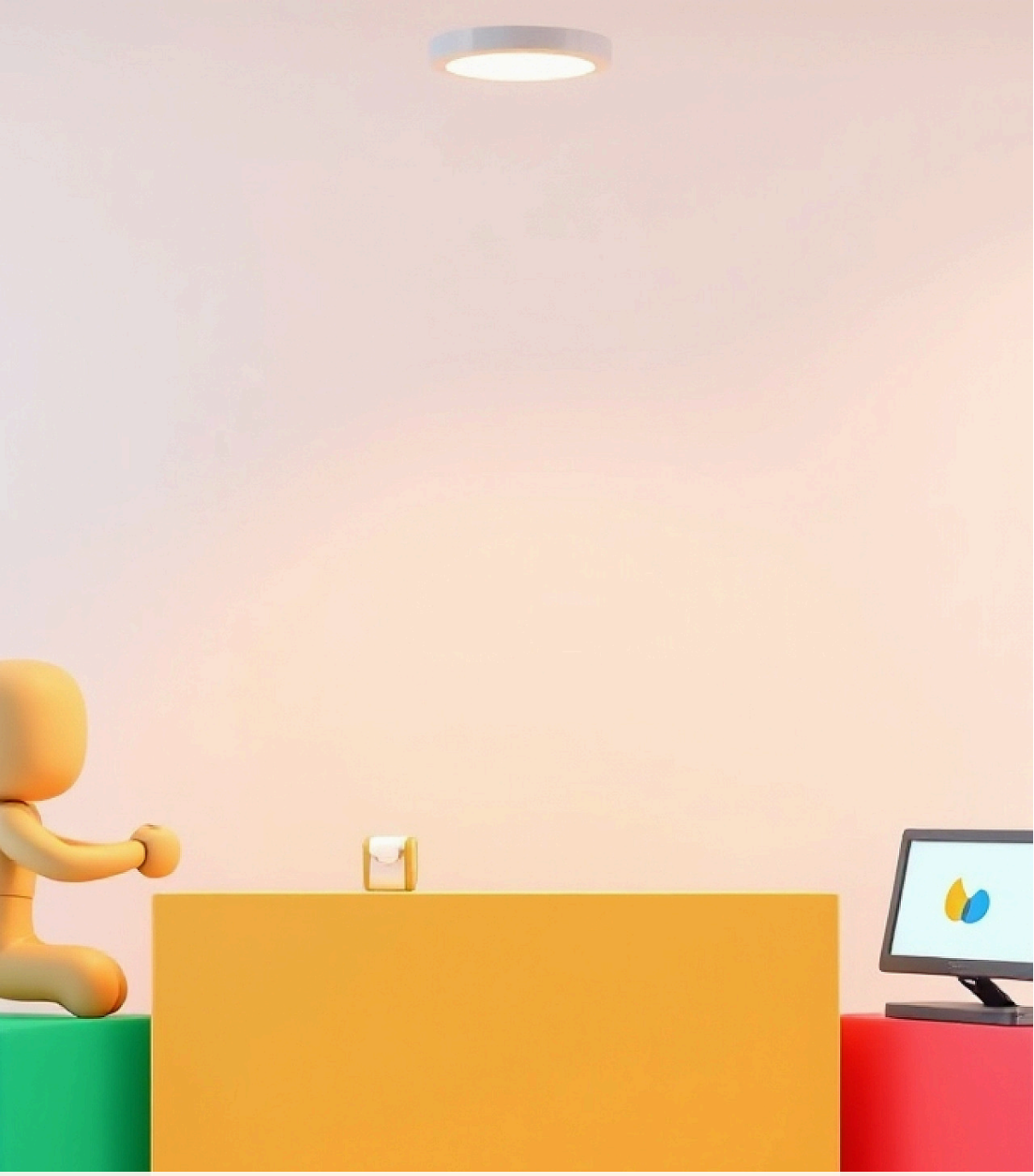
What is an API?



You have probably heard the term API and wondered what it means. The short answer: an API is an interface that lets one piece of software talk to another using specific rules. That is all you need to start. The rest of this book explains each part of that answer step by step.



Picture two separate programs that need to exchange something. Think of two people who want to talk but do not share a language yet. They need a way to pass messages back and forth. Communication between software is the whole point of an API. Without it, programs would stay isolated and could not help each other.



An interface is the agreed meeting point where two sides interact. Imagine a counter where you place an order and receive what you asked for. The API is that meeting point between two programs. One side comes to the interface to make a request, and the other side responds through the same interface. Nothing else needs to be shared.



For two programs to understand each other, they must follow the same rules, like speaking a shared language. The API is exactly this set of agreed rules. It tells both sides how to format messages and how to interpret them. With the same rules, the programs can work together even if they were built by different people.



An API defines which actions one program may ask another to perform. Only the listed operations are possible; nothing else is allowed. For example, an API might allow asking for a list of items or adding a new one. If an action is not in the list, the request will not work. This keeps communication predictable and safe.



The API also defines which information is sent and in what form. Both sides know exactly what to expect, such as a name, a number, or a date. This avoids confusion about what a message means. For instance, the API might say a request must include a city name as text. Then both programs handle it the same way.



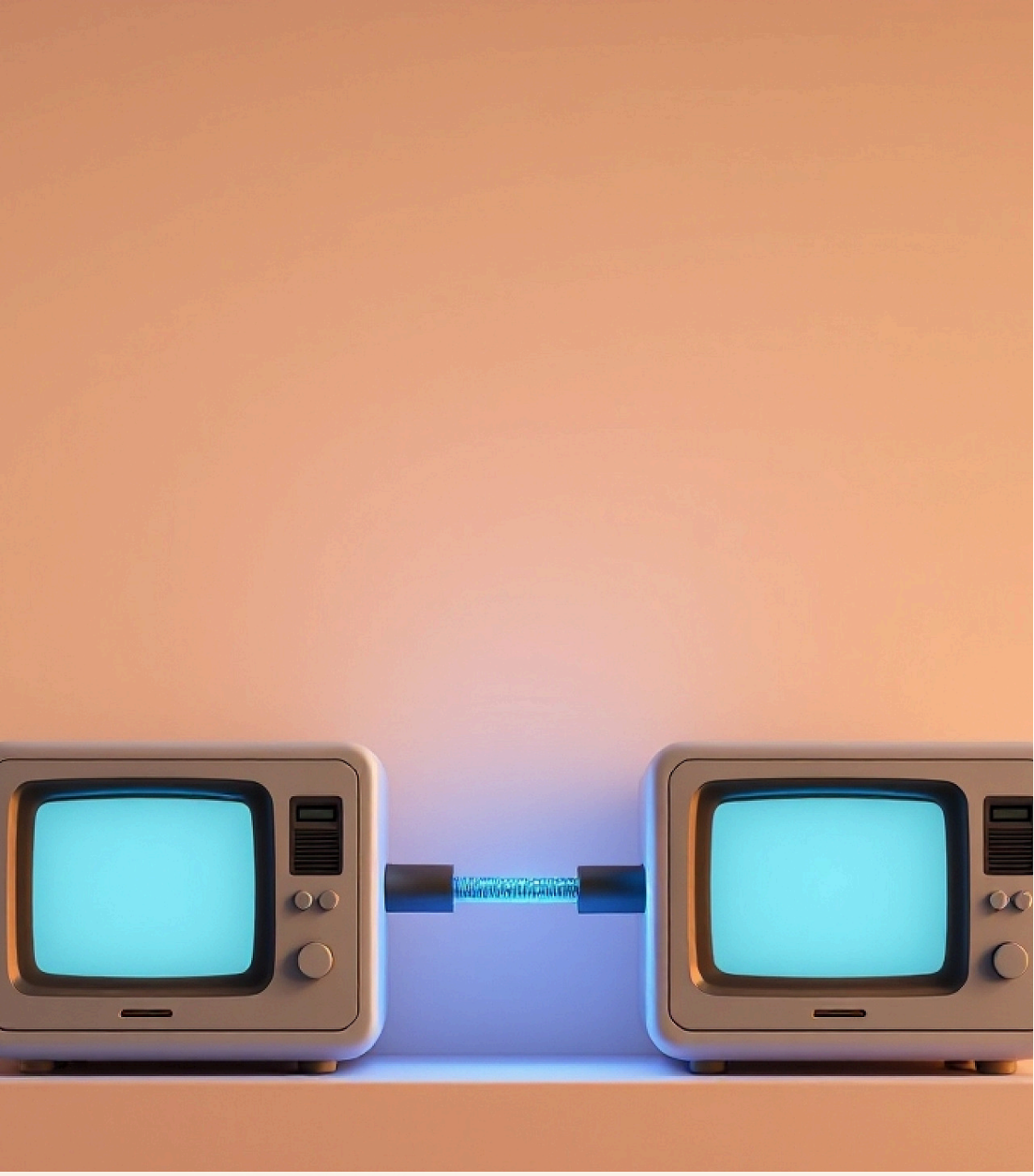
Consider an app that shows weather. The app asks a weather service for the forecast of a specific city. The request names the city, and the service sends back the weather data. The API defines how the city is named and how the forecast is returned. This makes the exchange smooth and reliable.



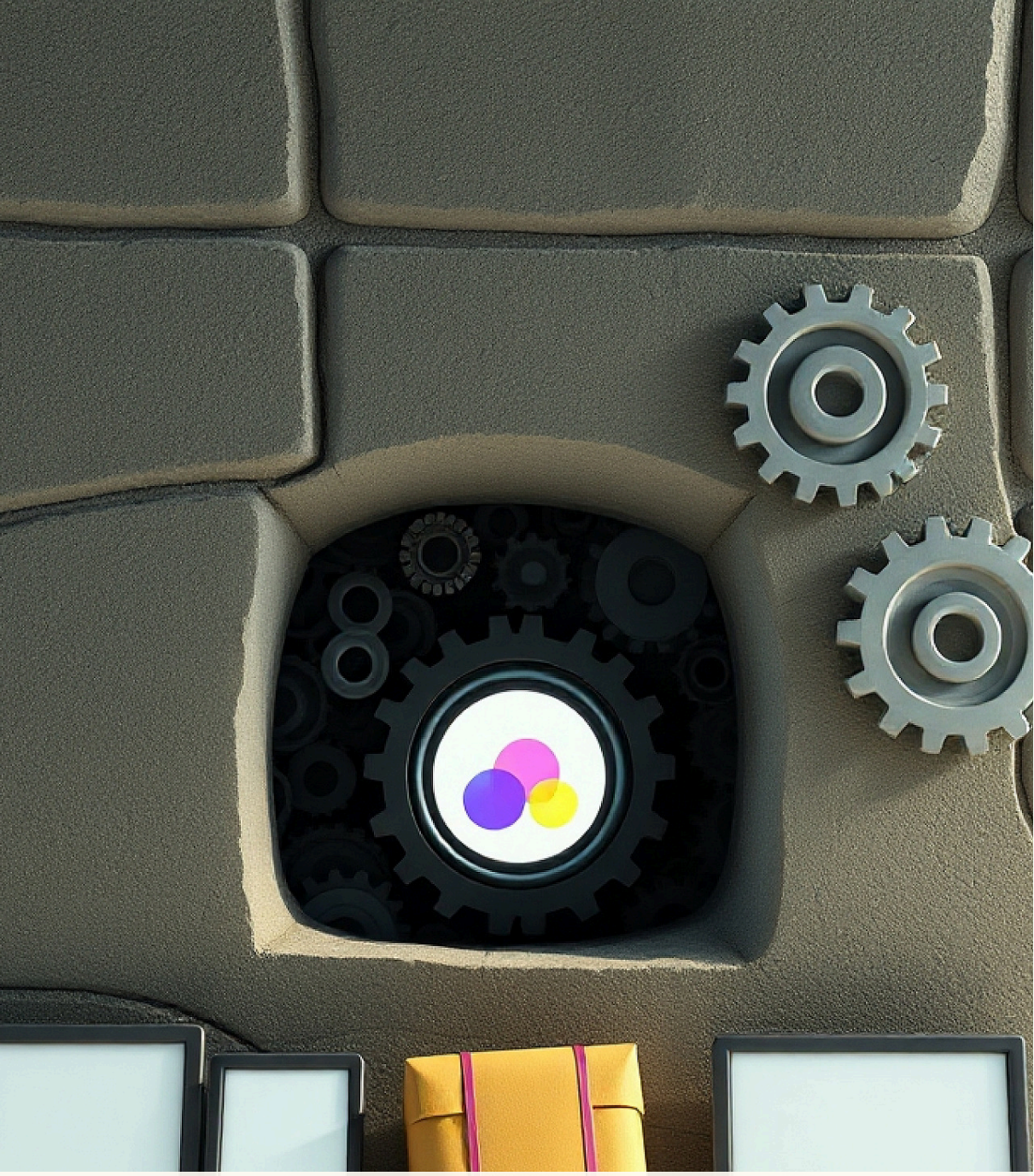
The weather example is a simple back-and-forth: one program asks, the other answers. The API shapes both the question and the answer. The asking program formats its request according to the rules. The answering program reads it, prepares a reply, and sends it back in the agreed form. This request-reply pattern is the heart of many APIs.



APIs do not require the internet. Two software components on the same computer can also talk through an API. For example, a program might use an API to ask another part of the same system for a calculation. The rules and the request-reply pattern stay the same, only the location changes. This shows how flexible the idea is.



Picture two programs running on one machine. They use an API just like the weather example, but locally. One component requests data, and the other provides it. The same rules apply: allowed operations, defined information, and a clear reply. The API works the same way whether the programs are far apart or side by side.



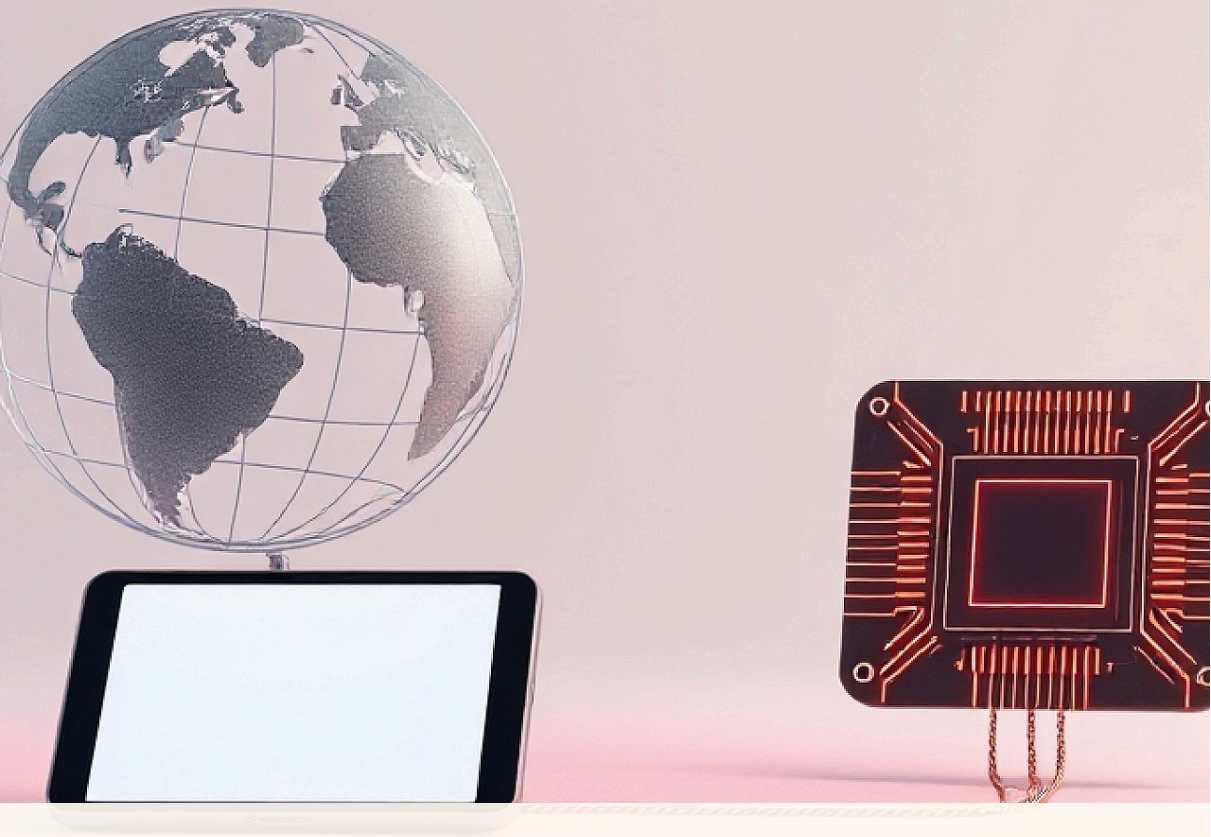
Fixed rules let programs work together without knowing each other's internals. Each side only needs to know the interface, not how the other is built. This is like ordering at a counter without entering the kitchen. The API hides complexity and makes cooperation simple. You can change one program as long as the interface stays the same.



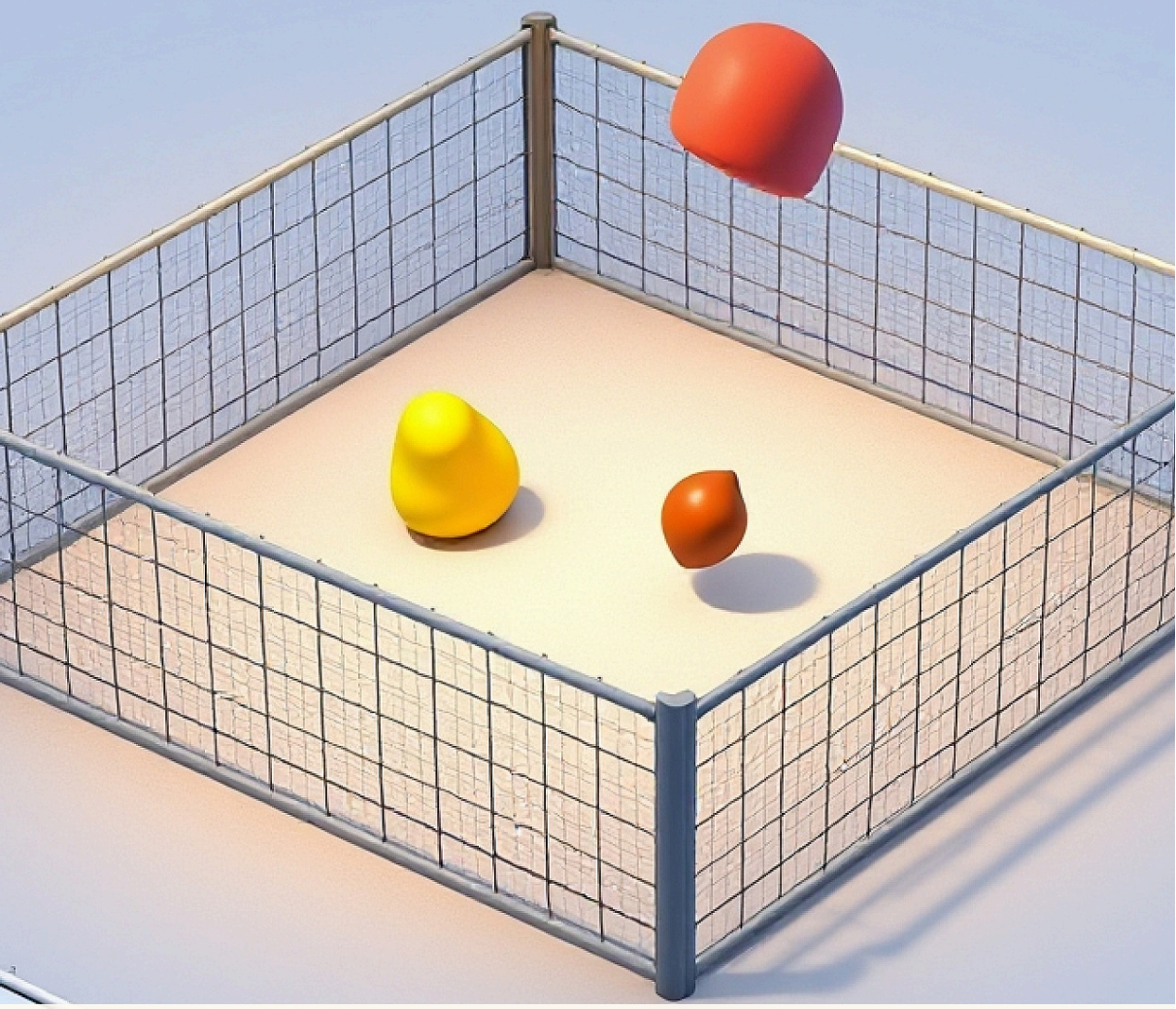
In any API exchange, there are two roles: the program that makes the request and the program that provides the answer. The API sits between them, defining how the request is made and how the answer is given. One side asks, the other answers, and both follow the same rules. This clear division keeps communication organized.



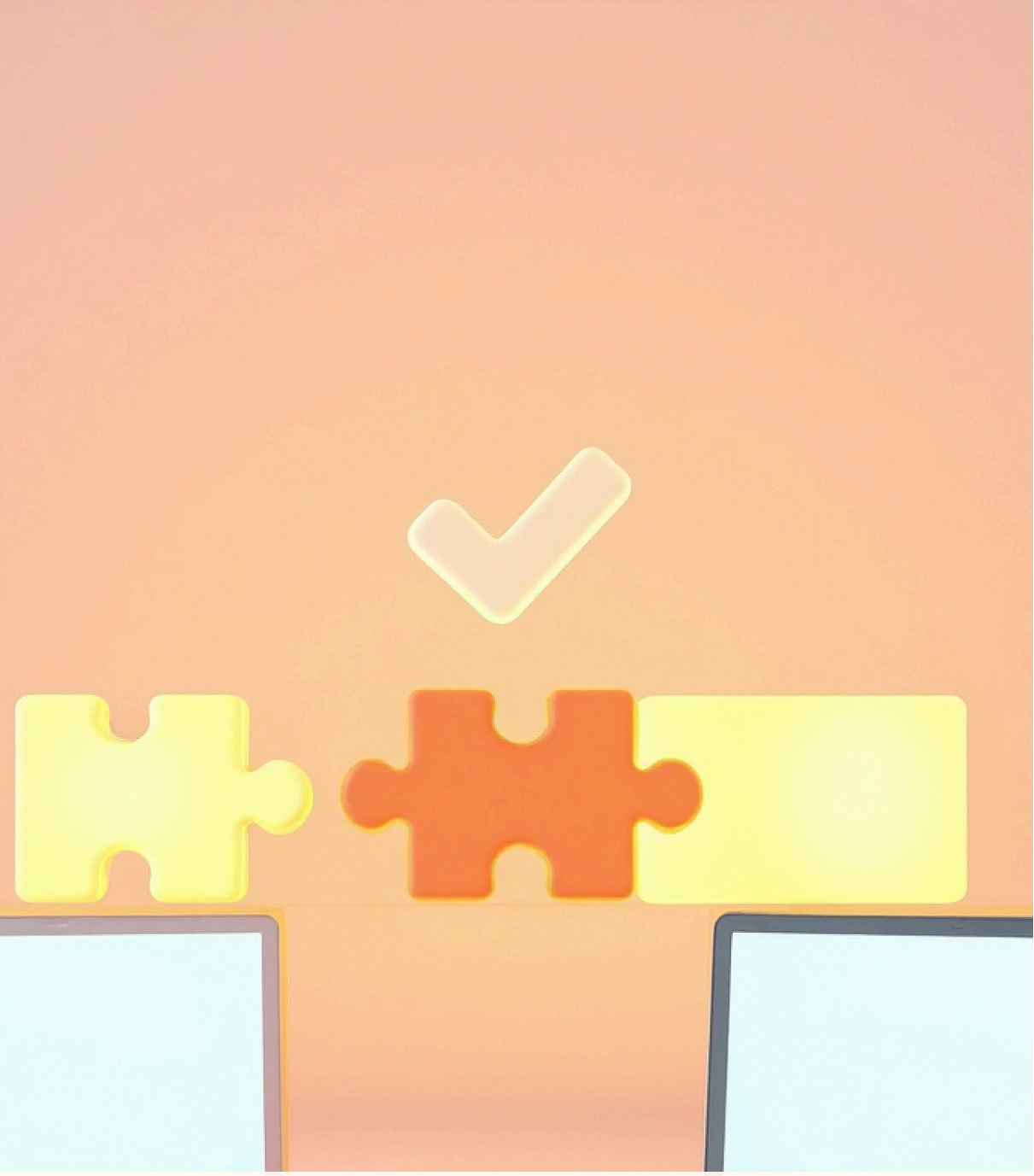
A common confusion is thinking the API is the whole program or its data. The API is not the program itself. It is only the agreed way to ask for and receive things. The program behind the API does the actual work, but you interact only through the interface. Knowing this helps you use APIs correctly.



Another common confusion is that APIs are limited to web services. APIs are not limited to the internet. They also work between components on the same computer. The same principles apply whether the programs are across the world or inside one machine. So do not assume an API always means a remote service.



You cannot ask for anything you like through an API. Only the operations and information the API defines are possible. If you request something outside those rules, it will not work. For example, if the weather API only gives forecasts, you cannot ask it for stock prices. Respecting the defined boundaries is essential.



To sum up, an API is an interface with specific rules that lets software communicate. It defines allowed operations and the information sent, and it works over the internet or on one computer. Remember that the API is not the program, not limited to the internet, and not open to any request. These key points cover what an API is.