Lecture-37
The Purpose of APIs

(Refer Slide Time: 00:04)
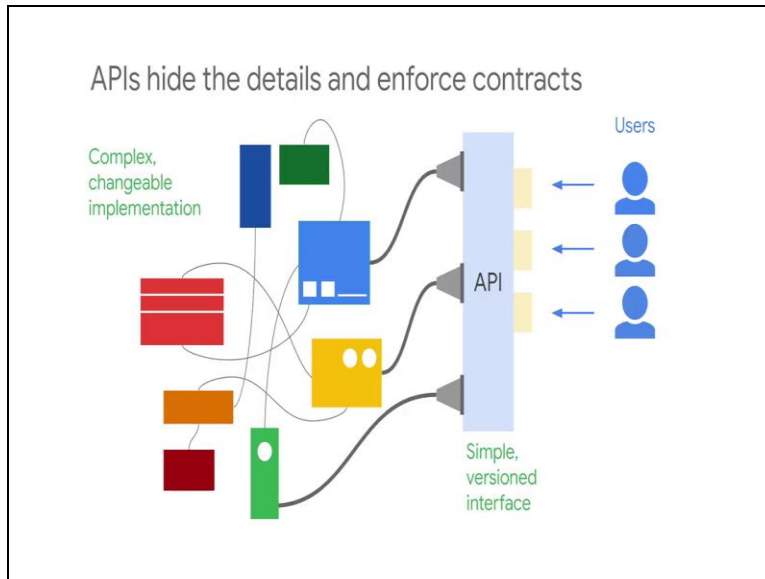
Let us start by discussing the purpose of API's. An API is a software structure written so that presents a clean well-defined interface that abstract away needless detail. API's are used to simplify the way different disparate software resources communicate by using a universal structure of communications we open up a wide range of opportunities.

(Refer Slide Time: 00:26)

What is a REST API?

- A set of constraints a service must comply with.
- An API that uses HTTP requests to GET, PUT, POST, and DELETE data.
- Designed to set up a format for applications to communicate.
- Great for cloud applications because they are stateless.
- Authentication via OAuth and security by leveraging tokens.

{REST}

REpresentational State Transfer

Representational state transfer or REST is currently the most popular architectural style for surfaces. REST outlines a key set of constraints and agreements that the service must comply with. If a service complies with these REST constraints it is said to be restful. The web is HTTP based and provides an architectural structure that scales well and stands the test of time. REST transfers the ideas that work so well for the web and applies them to services.

API is intended to be spread widely to consumers and deployed to devices with limited computing resources like mobile are well-suited to a REST structure. REST api is use HTTP requests to perform get, put, post and delete operations. Having different software services leverage a universal communication channel ensures applications can get updated or rewritten and still be able to work with other applications as long as they conform to the agreed-upon API standard.

What are the main reasons rest API is work well with the cloud is due to their stateless nature. State information does not need to be stored or referenced for the API to run. Authentication can be done through OAuth and security can be used by leveraging tokens.

(Refer Slide Time: 01:59)

Deploying and managing APIs can be difficult
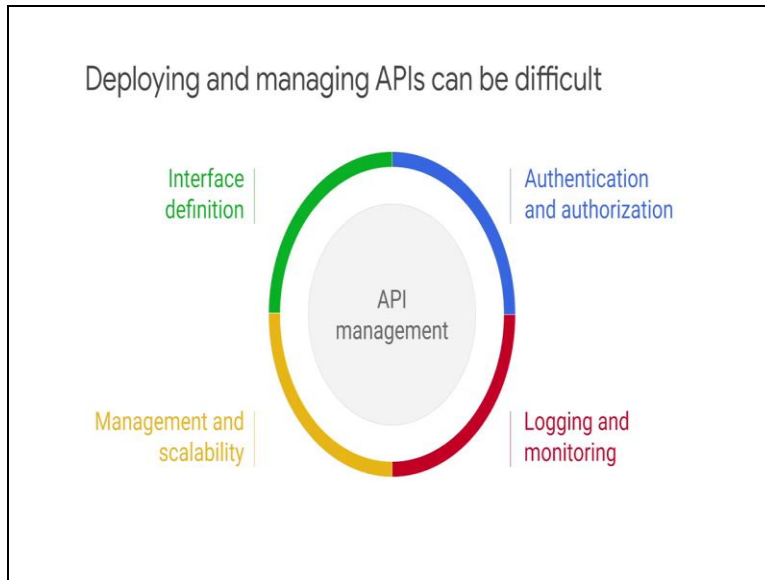
When deploying and managing API's on your own there are a number of issues to consider for example the language or format you will use to describe the interface. How you will authenticate services and users who invoke your API. How you will ensure that your API scales to meet demand and whether your infrastructure log details API invocations and provides monitoring metrics.