



90%

De los developers usan APIs

19th Developer Economics Survey - Slashdata

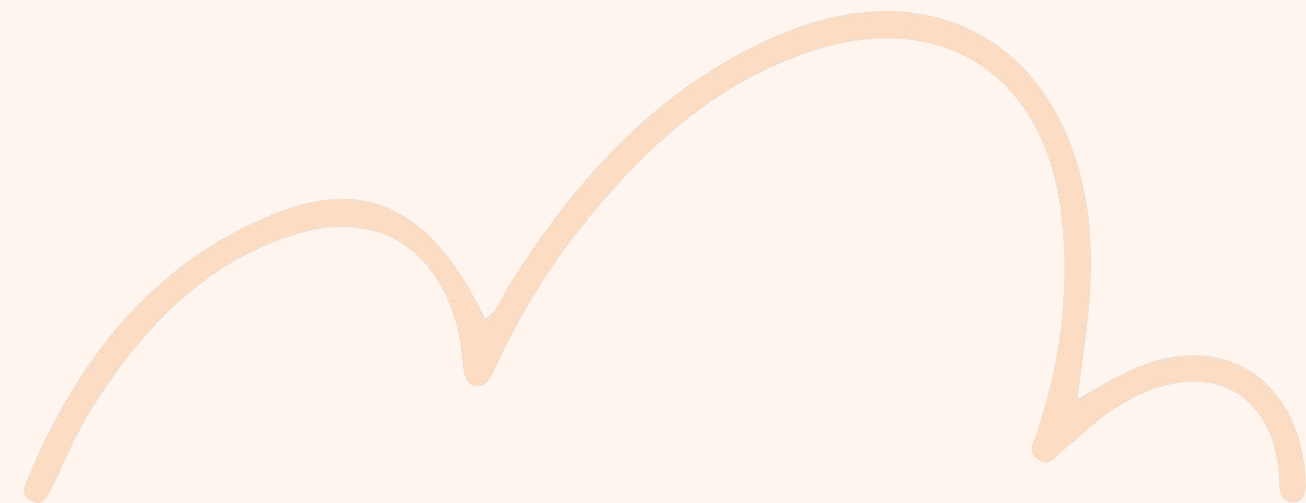
OpenAPI Specification: Presente y futuro



Benjamin Granados *(he/him)*

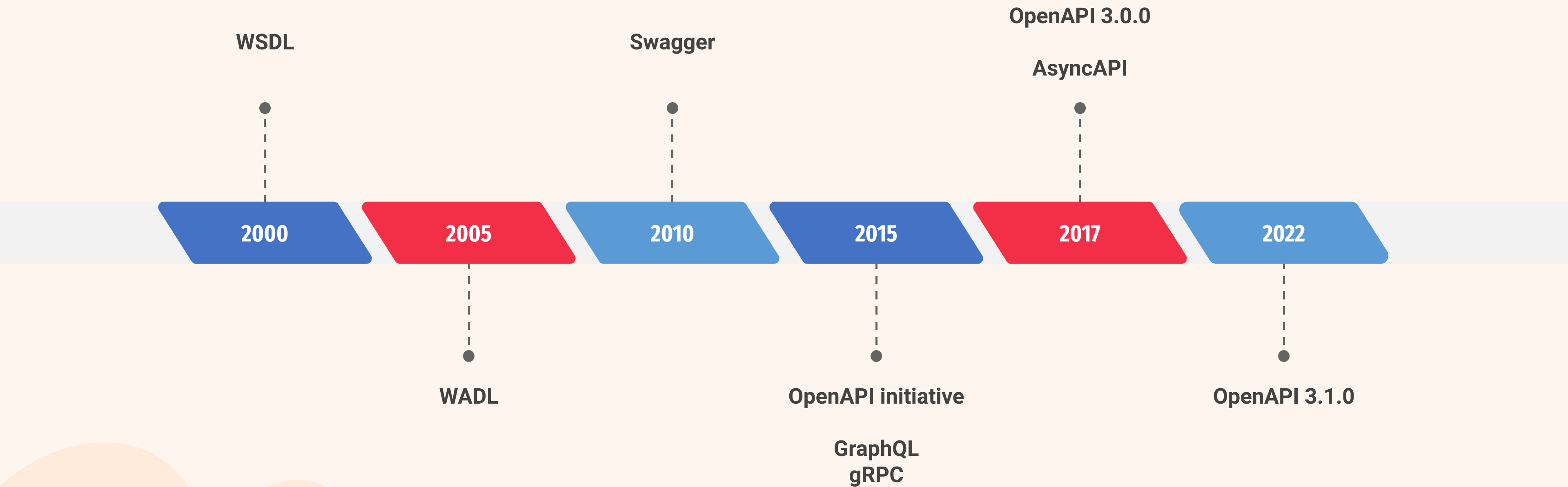
Developer Evangelist  **twilio**

@benjagm



**Las APIs hacen la
vida más fácil a los
developers**

Un poco de historia





OpenAPI Specification

OpenAPI v3.0

info

servers

security

paths

tags

externalDocs

components

YAML

```
openapi: 3.0.0
info:
  title: Sample API
  description: Multiline/single-line description in Common Mark or HTML.
  version: 0.1.9

servers:
- url: http://api.example.com/v1
  description: Optional description, e.g. Main (production) server
- url: http://staging-api.example.com
  description: Optional description, e.g. Internal staging server

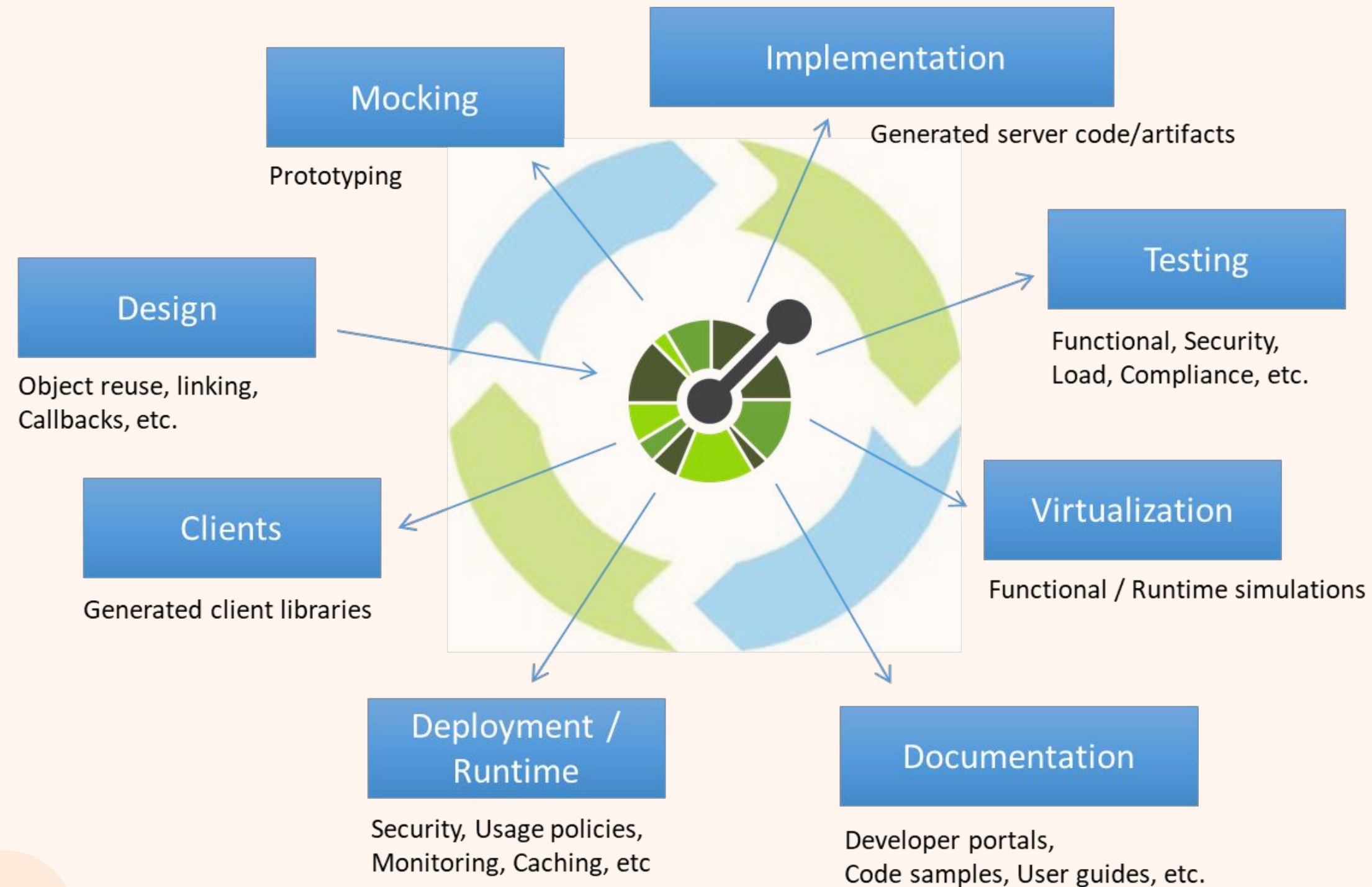
paths:
  /users:
    get:
      summary: Returns a list of users.
      description: Optional extended description (Common Mark/HTML).
      responses:
        '200': # status code
          description: A JSON array of user names
          content:
            application/json:
              schema:
                type: array
                items:
                  type: string
```

¿Por qué OpenAPI es tan importante?

- ✓ **Base de usuarios enorme**
- ✓ **Estabilidad**
- ✓ **Mejor DX**
- ✓ **Facilita el gobierno y la colaboración**



OpenAPI Specification alimenta el ciclo de vida de las APIs



Encontrando las herramientas adecuadas

<https://openapi.tools/>

Tool Types

- **Auto Generators:** Tools that will take your code and turn it into an OpenAPI Specification document.
- **Converters:** Various tools to convert to and from OpenAPI and other API description formats.
- **Data Validators:** Check to see if API requests and responses are lining up with the API description.
- **Description Validators:** Check your API description to see if it is valid OpenAPI.
- **Documentation:** Render API Description as HTML (or maybe a PDF) so slightly less technical people can figure out how to work with the API.
- **DSL:** Domain Specific Language to write OpenAPI in your language of choice.
- **GUI Editors:** Visual editors help you design APIs without needing to memorize the entire OpenAPI specification.
- **Miscellaneous:** Anything else that does stuff with OpenAPI but hasn't quite got enough to warrant its own category.
- **Mock Servers:** Fake servers that take description document as input, then route incoming HTTP requests to example responses.
- **Parsers:** Loads and read OpenAPI descriptions, so you can work with them programmatically.
- **SDK Generators:** Generate code to give to consumers, to help them avoid interacting at a HTTP level.
- **Security:** By poking around your OpenAPI description, some tools can look out for attack vectors you might not have noticed.
- **Server Implementations:** Easily create and implement resources and routes for your APIs.
- **Testing:** Quickly execute API requests and validate responses on the fly through command line or GUI interfaces.
- **Text Editors:** Text editors give you visual feedback whilst you write OpenAPI, so you can see what docs might look like.
- **Learning:** Whether you're trying to get documentation for a third party API based on traffic, or are trying to switch to design-first at an organization with no OpenAPI at all, learning can help you move your API spec forward and keep it up to date

Presente de OpenAPI (Version 3.1)

- ✓ **Compatibilidad completa con JSON Schema**
- ✓ **Mejor especificación de Webhooks**
- ✓ **Identificación mejorada (SPDX)**
- ✓ **Reusabilidad**

Que retos enfrenta OpenAPI

- ✓ **Migración de Swagger a OpenAPI**
- ✓ **Evolución de las herramientas y el propio estándar**
- ✓ **Convergencia vs Coexistencia con AsyncAPI**



Thank you