

GOING TO

INFINITY AND BEYOND

DOCUMENTATION WITH

OpenAPI Specification

@taylor_atx



@taylor_atx

Single Source
of
TRUTH

@taylor_atx

So you've
built an API...

@taylor_atx

What is the
OpenAPI Specification?
(formerly known as Swagger)

Standard, structured approach
for describing HTTP APIs that
is both human and machine
readable

info

servers

security

paths

tags

externalDocs

Components:

responses, parameters, examples, links,
request Bodies, headers, callbacks,
schemas, securitySchema

An OpenAPI document looks like:

Openapi: 3.0.0

info:

version: 1.0.0

title: "Corgi API"

license:

name: MIT

servers:

- url: <http://api.corgis.io/v1>

paths:

/corgis:

get:

Summary: List all corgis
operationId: listCorgis

tags:

- pets

parameters:

- name: limit

* Not a complete document

Components:

schemas:

Corgi:

required:

- id

- name

properties:

id:

type: integer

format: int64

name:

type: string

tag:

type: string

Corgis:

type: array

items:

\$ref: "#/components/schemas/Corgi"

Error:

required:

- code

- message

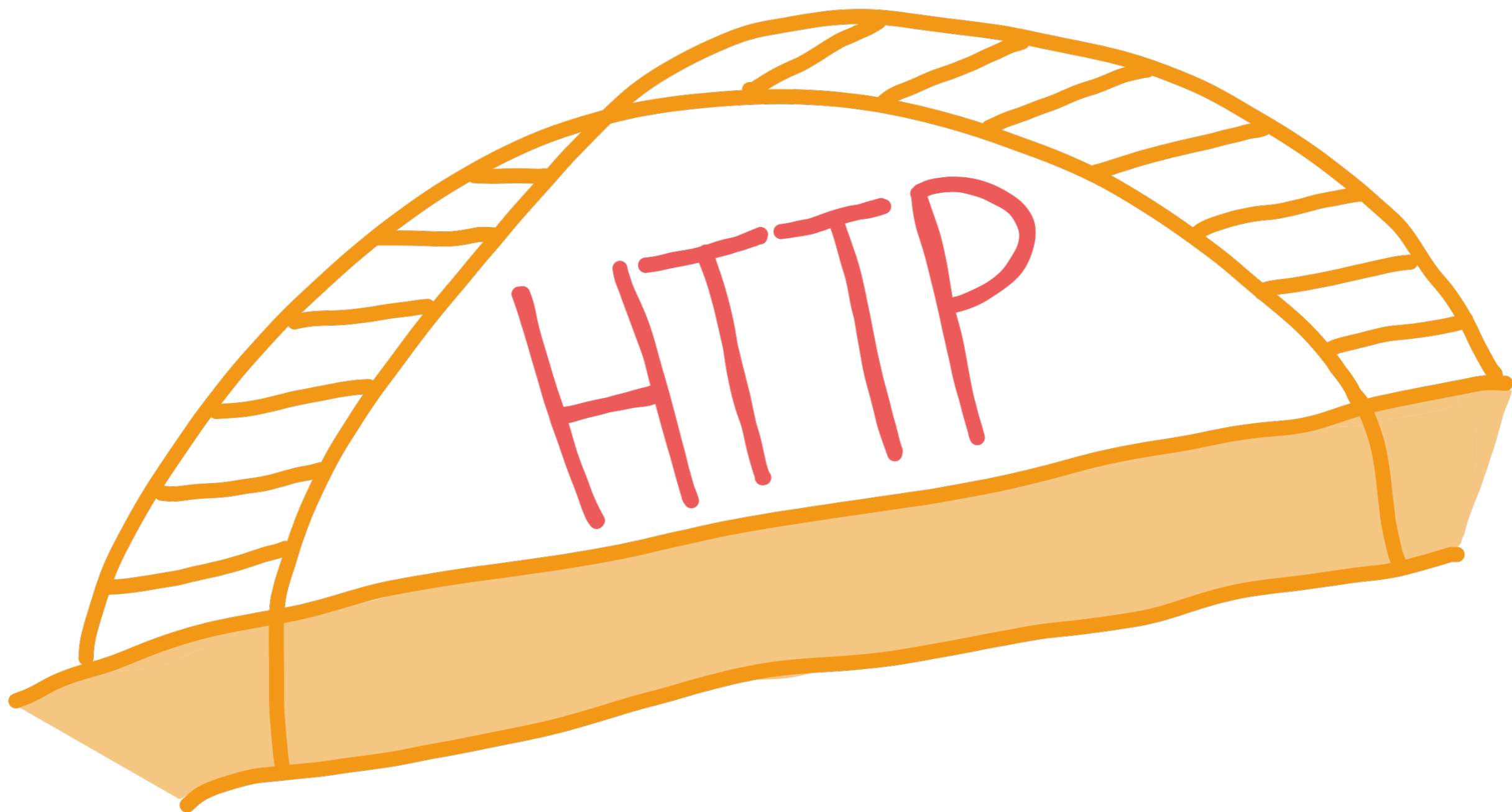
properties:

* Not a
complete
document

Who am I?



twitter → @taylor_atx



@taylor_atx

Hidden OpenAPI

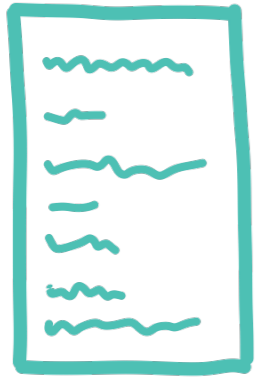
Benefits



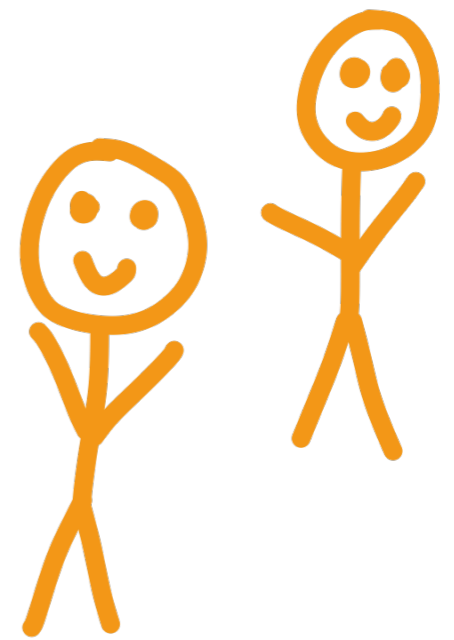
It is NOT just for
API References

- Development contract
- Prototype and mocking
- Client libraries and SDKs
- Testing
- Server stubs

More than a technical
representation



Tool for
Collaboration





@taylor_atx

#1 Design-First

#2 Mocking and Feedback

#3 Testing

#1 Design-First

- Consistency 
- Code-First is expensive 
- Gain full benefits of OpenAPI

#2 Mocking and Feedback

OpenAPI Document 



Request →



Response

Mock Server

If you use or build internal APIs (including libraries/SDKs), you can...

- Build faster
- Parallelize development
- More options



@taylor_atx



Consumer



Consumer



Consumer



If you write docs or tutorials,
you can...

- Start writing sooner
- Give feedback
- Have interactive docs

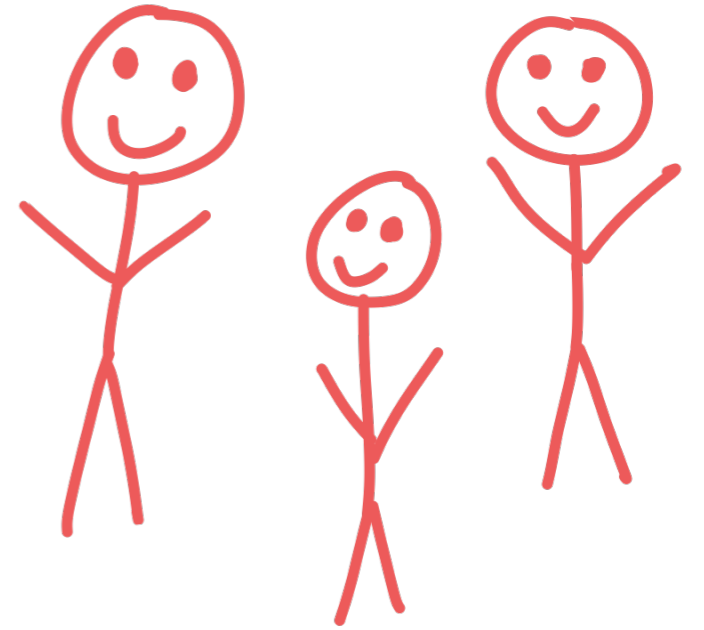
@taylor_atx

I you care about developer experience or advocate for users, you can...

- Power feedback loops
- Put real endpoints in front of users

@taylor_atx

Start with humans



Questions to ask:

What are the tasks users
want to accomplish?

Will the API help users
achieve these tasks?

Example: What calls do they need
to make to achieve a
task?



Are the endpoints confusing?

Example: Do you need insider info to use certain endpoints? Consistent with other APIs? Standards?

@taylor_atx

Public OpenAPI

Specification documents



@taylor_atx

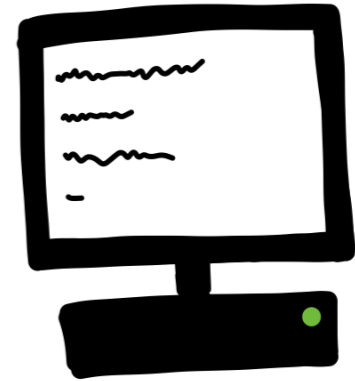
"No matter how great your architecture is and how maintainable your code is, if the API doesn't do what the design says it should, you've failed."

- Continuous API Management

@taylor_atx

#3 Testing

- Machine Readable



- Linting

- Contract Testing



What is Contract
Testing?




{ "user": "ada"
...
}

OpenAPI Document



Response
Body



Validation
Server



PASS
OR
FAIL

Less broken

SDKs

Example:

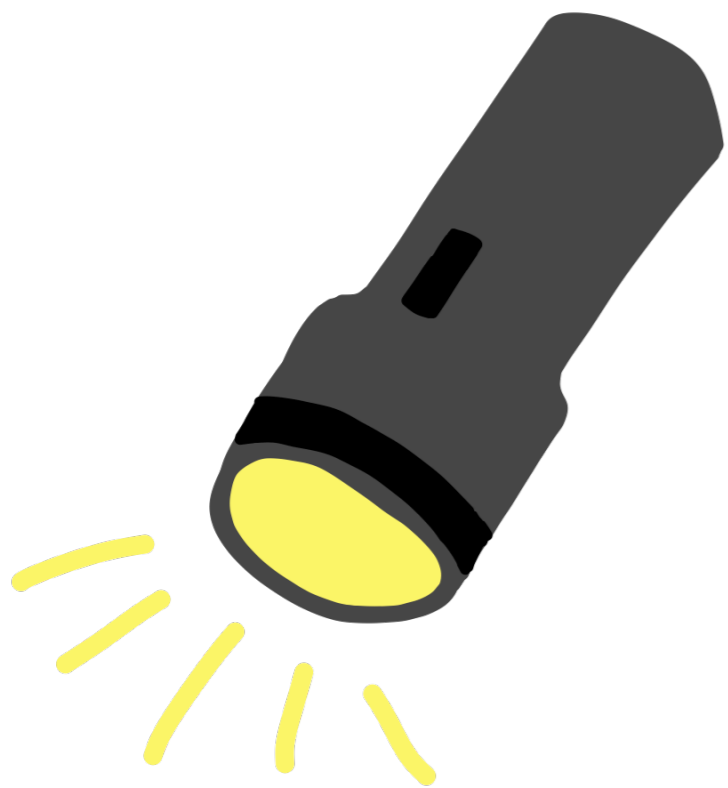
Update SDK code

↳ Open pull request

↳ CI runs → Contract tests

↳ PASS
OR
FAIL

The Reality



@taylor_atx

Ideation



Design



Mocking



Implementation



Testing



Management



Monitoring



Ideation

Design

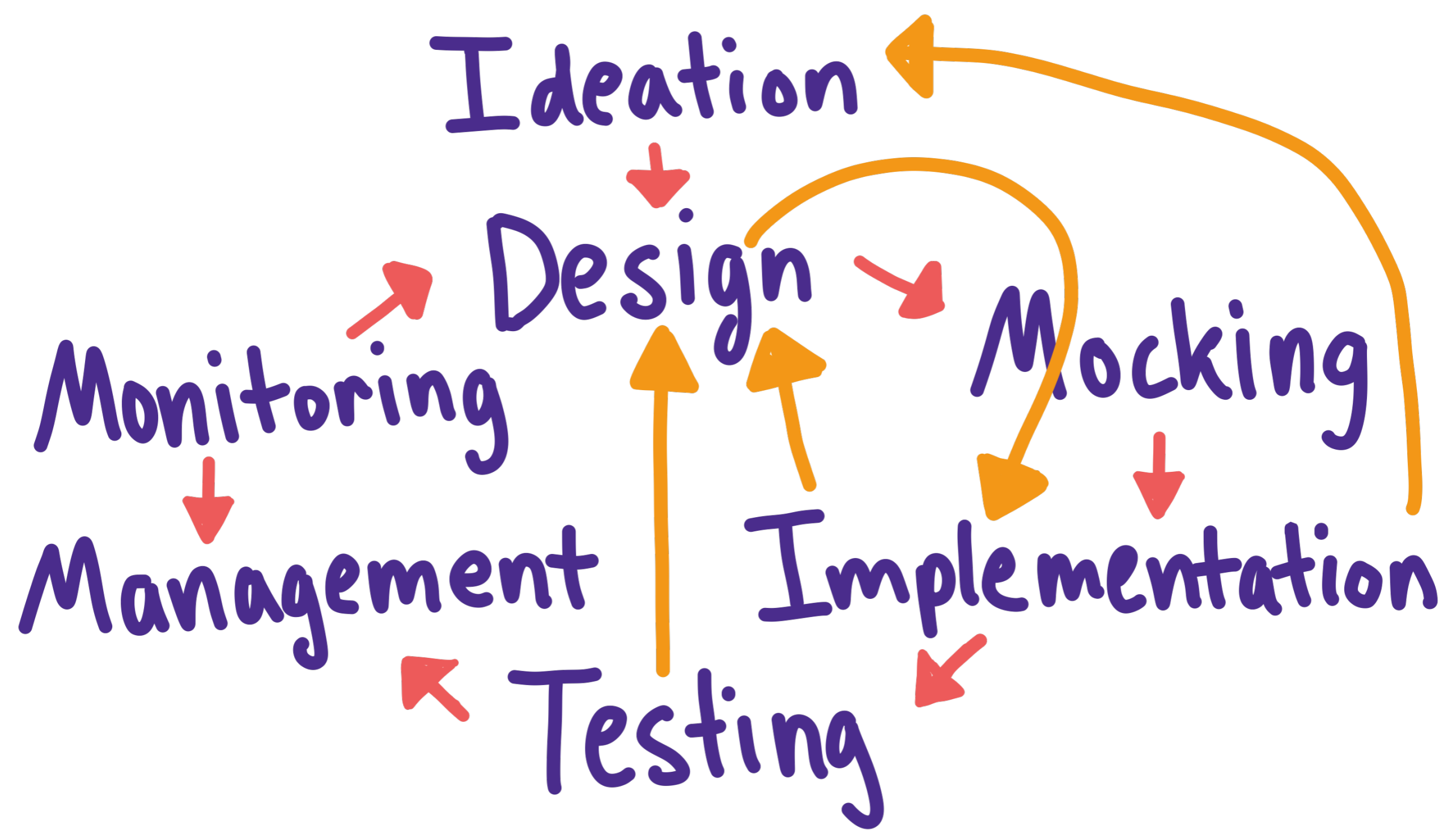
Mocking


Monitoring

Management

Implementation

Testing



What would allow  you to focus more on the API Experience?

twitter → @taylor_atx

slides → noti.st/tbarn

tools → openapi.tools



API Specifications



Conference



Oct 15-17th, 2019



Vancouver, BC



apispecs.io

