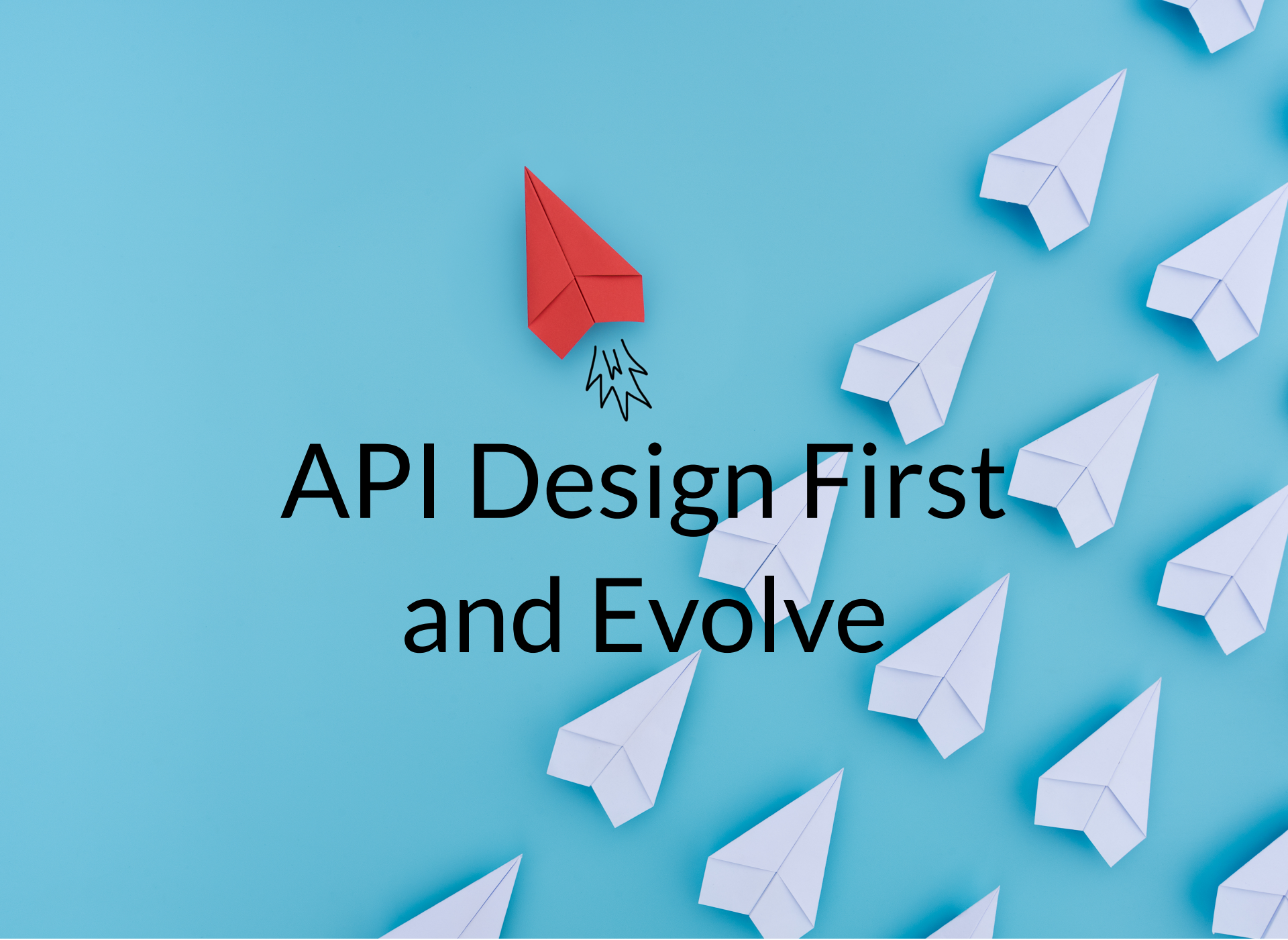

API Design First
and Evolve

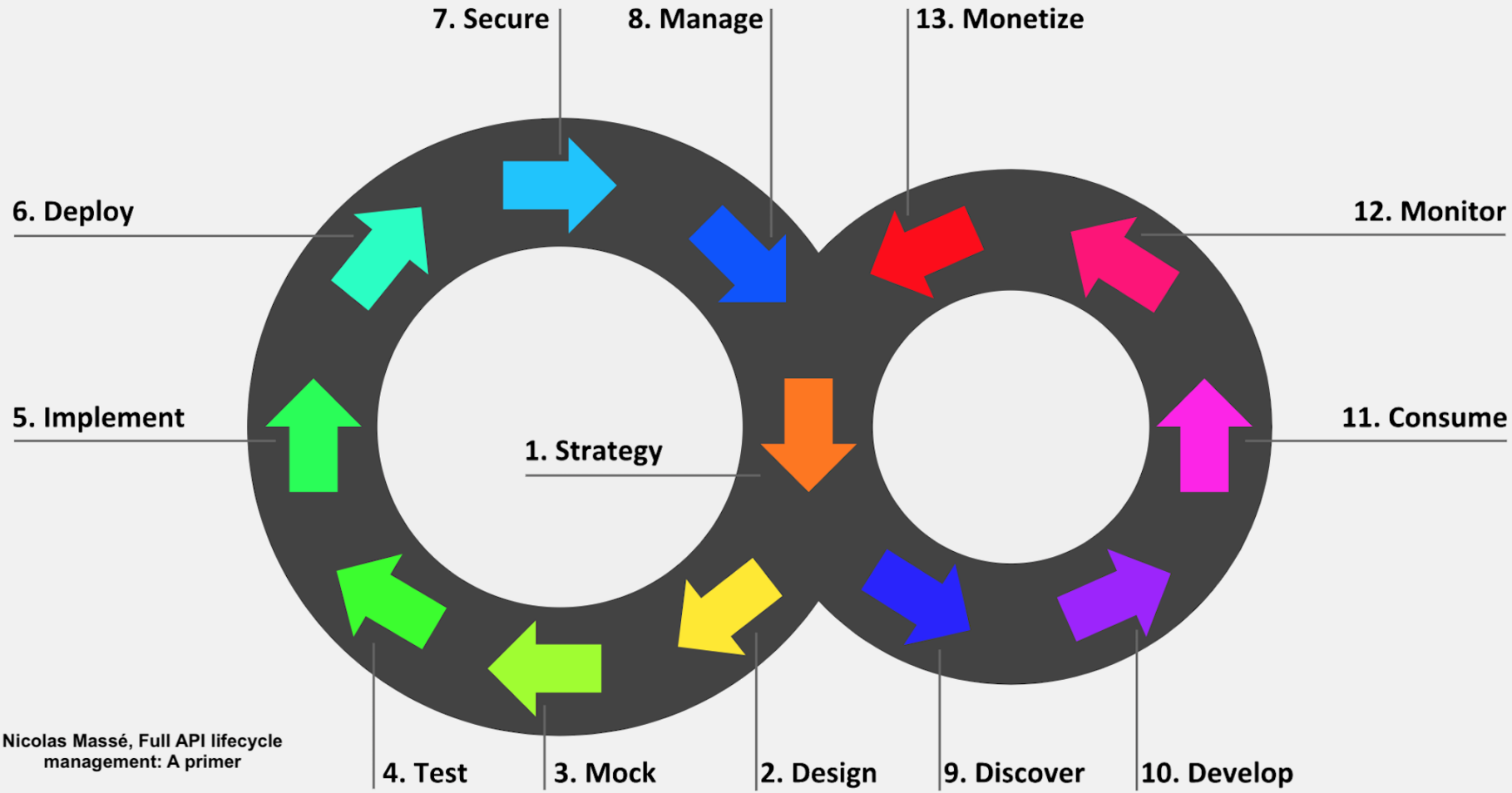
HELLO!

m Phil Sturgeon

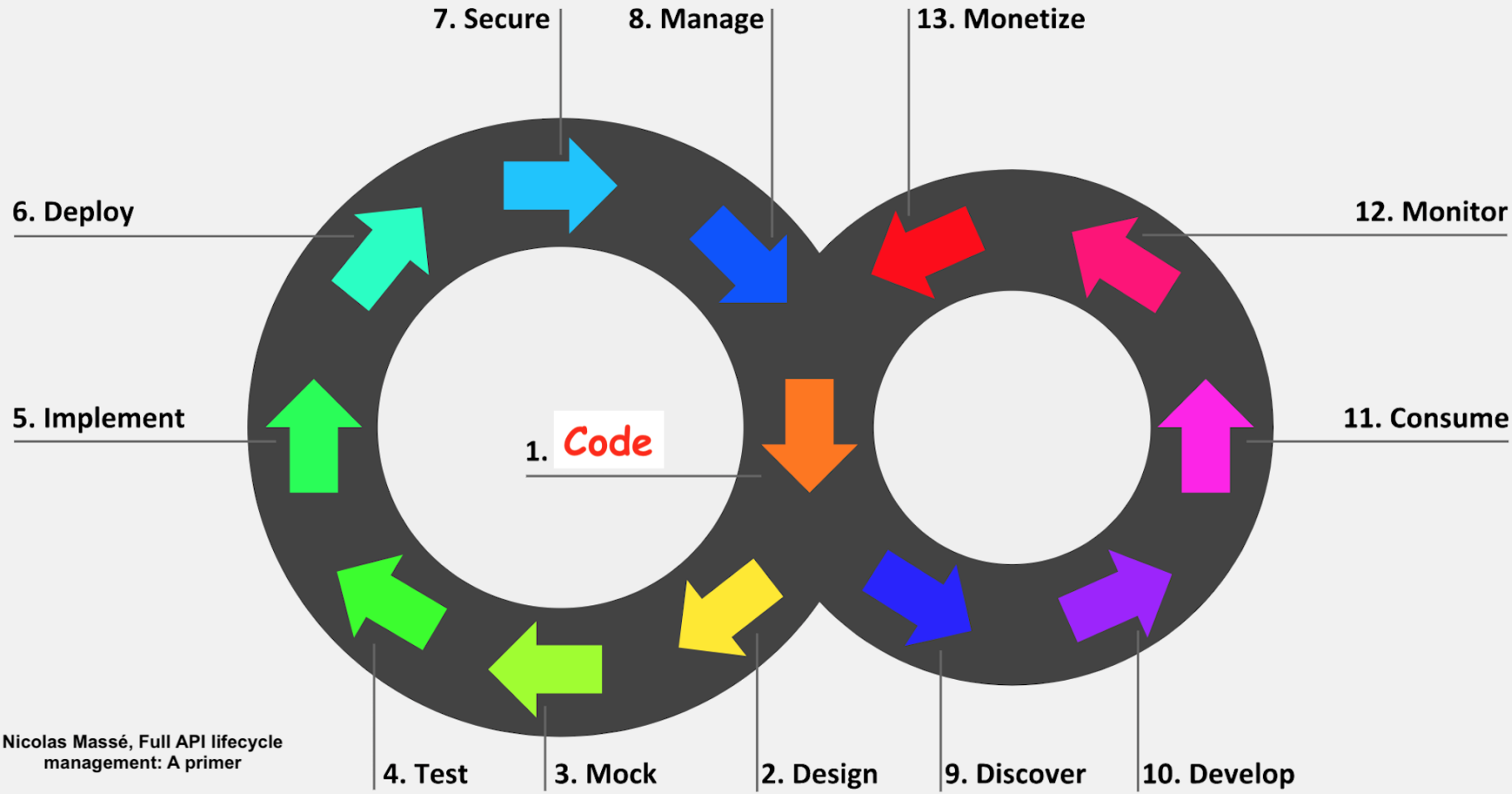
ve to talk about APIs,
shing bikes, and
ving the planet.

u can find me at
hilsturgeon

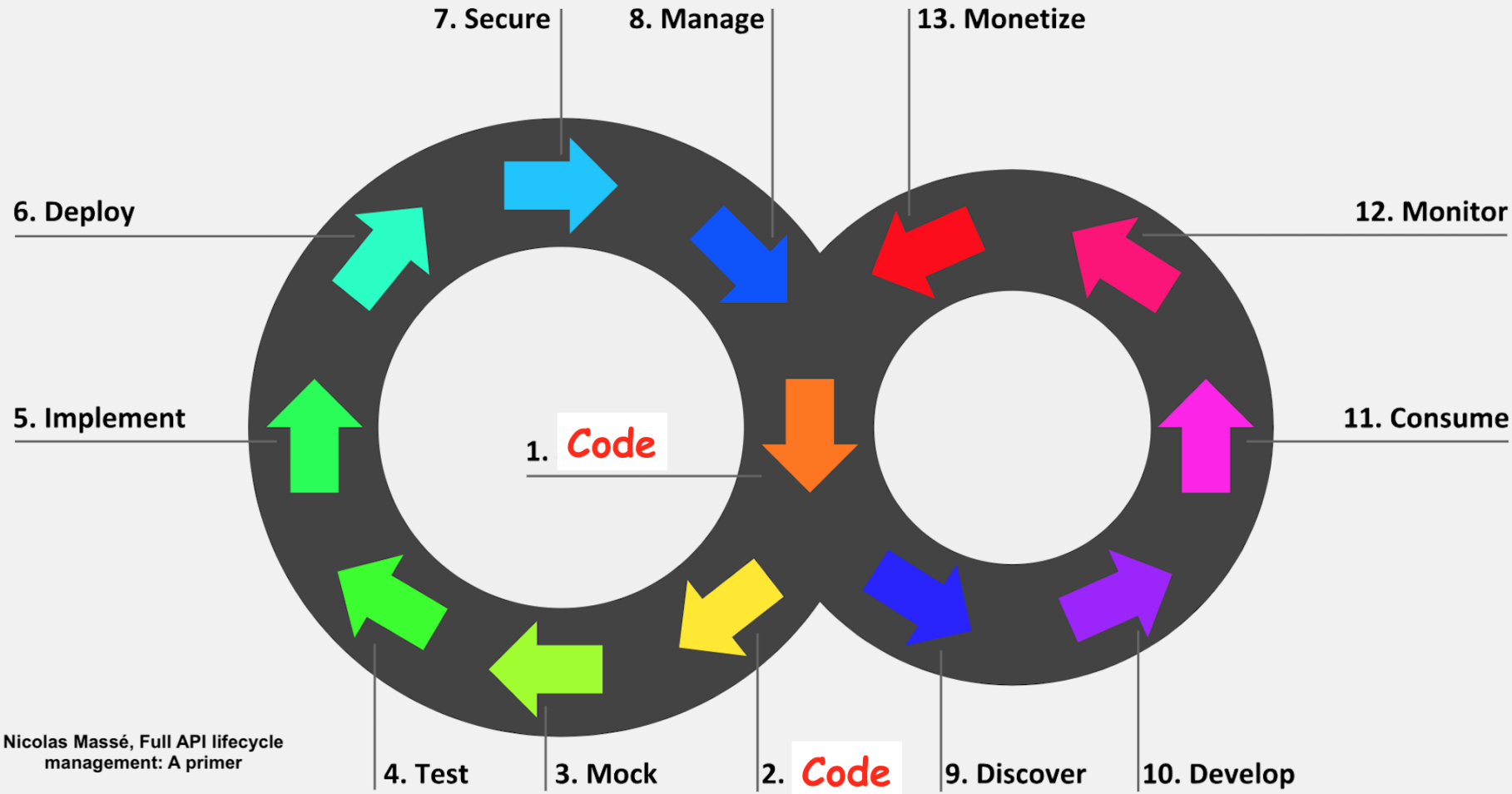

wework
MARKET

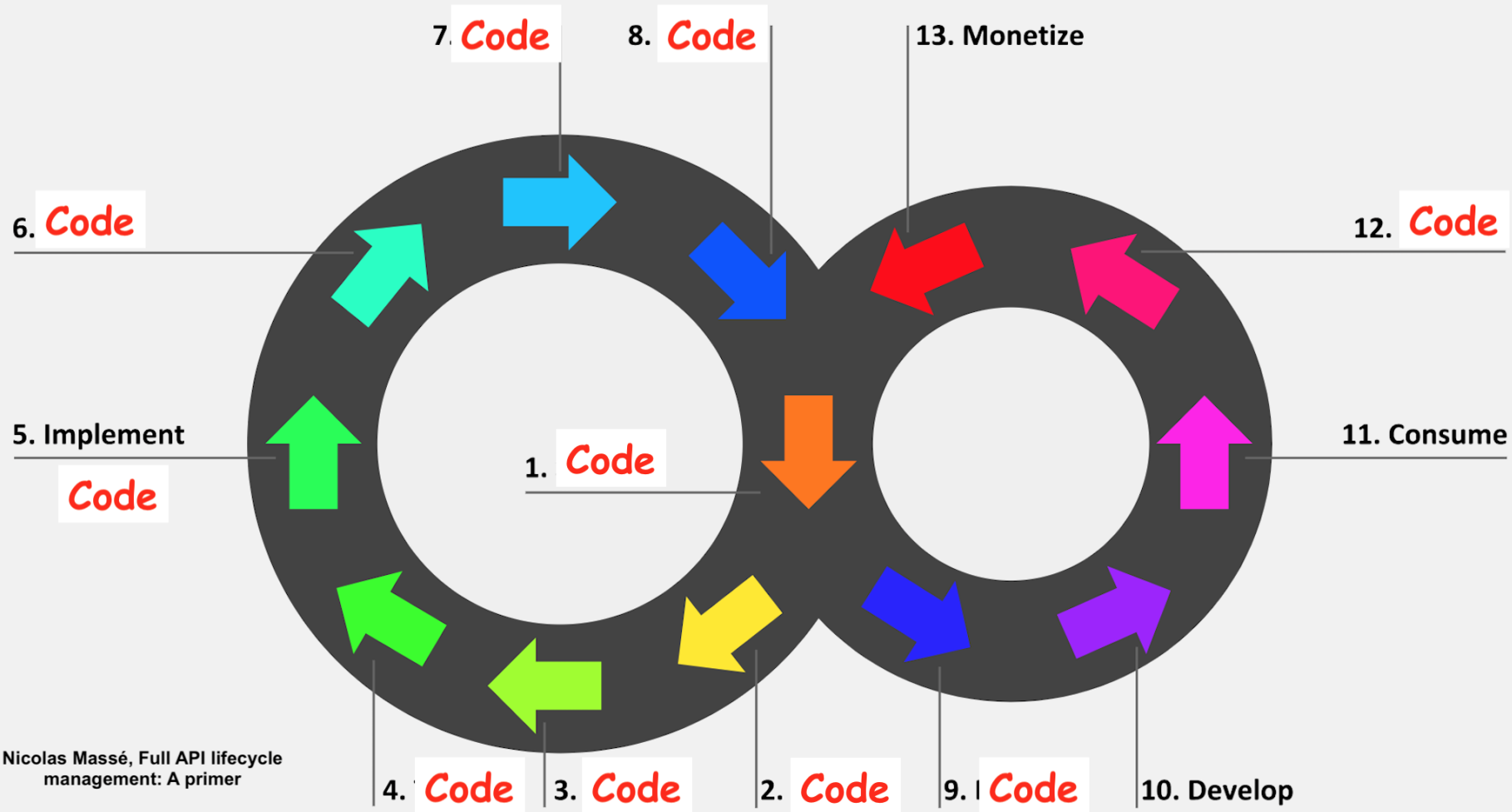


Nicolas Massé, Full API lifecycle management: A primer



Nicolas Massé, Full API lifecycle management: A primer


7. Secure
8. Manage
13. Monetize
6. Deploy
12. Monitor
5. Implement
1. Code
11. Consume
4. Test
3. Mock
2. Code
9. Discover
10. Develop
Nicolas Massé, Full API lifecycle management: A primer



Nicolas Massé, Full API lifecycle management: A primer

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?
3. Design First or Code First?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?
3. Design First or Code First?
4. Where are the visual editors?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?
3. Design First or Code First?
4. Where are the visual editors?
5. How/when do we create documentation?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?
3. Design First or Code First?
4. Where are the visual editors?
5. How/when do we create documentation?
6. How/when do we create mocks?

API WORKFLOW QUESTIONS

1. Swagger, OpenAPI v3, API Blueprint, RAML?
2. Annotations or YAML?
3. Design First or Code First?
4. Where are the visual editors?
5. How/when do we create documentation?
6. How/when do we create mocks?
7. How do we keep code and docs in sync?

THE API DESCRIPTION FORMAT

OpenAPI is an API Description Format for creating HTTP API Descriptions.

You can describe your API with a Description Document.

e.g.: `openapi.yaml`

```yaml
openapi: 3.0.3

info:
  version: v1.0
  title: RemoteOk Jobs API
  contact:
    email: support@remoteok.io
  description: >
    Remotely interested in an a job? This is a JSON feed from Rem
    all the latest jobs. You can't search by category or tag as f
    tell, but... you can sift through!

servers:
  - url: 'https://remoteok.io/api'
    description: Production
  - url: 'https://sandbox.remoteok.io/api'
```

```yaml
type: object
properties:
  name:
    title: Name
    type: string
    description: Users full name supporting unicode but no emojis
    maxLength: 20
  email:
    title: Email
    description: Like a postal address but for computers.
    type: string
    format: email
  date_of_birth:
    title: Date Of Birth
    type: string
    description: 'Date of users birth in the one and only date st
```


3scale BY RED HAT
42crunch
ably
Acumatica The Cloud ERP
APIFORTRESS
APImetrics API INTELLIGENCE
Assertible
Atlassian
Band Protocol
Bloomberg
ebay
Google
HM Government
IBM
IFS
Intento
Interzoid
isa
Kong
Liferay
LinksRez Linking You to the World
Microsoft
MuleSoft
NHSX
opentravel enabling the future
ORACLE + apiary
RepreZen
SAP
SMARTBEAR
software AG
Stoplight
talend
TeejLab DATA | RESEARCH | SOFTWARE | INNOVATIONS
TORO CLOUD
Tyk

ANNOTATIONS OR YAML?

```
class UserController {
  @OpenApi(
      path = "/users",
      method = HttpMethod.POST,
      // ...
  )
  public static void createUser(Context ctx) {
      // ...
  }
}
```

```
/**
  * @OA\Get(path="/2.0/users/{username}",
  *   operationId="getUserByName",
  *   @OA\Parameter(name="username",
  *     in="path",
  *     required=true,
  *     description=Explaining all about the username parameter
  *     @OA\Schema(type="string")
  *   ),
  *   @OA\Response(response="200",
  *     description="The User",
  *     @OA\JsonContent(ref="#/components/schemas/user"),
  *     @OA\Link(link="userRepositories", ref="#/components/links
  *   )
  * )
  */
```

```
/**
 * @swagger
 * /users:
 *   get:
 *     description: Returns users
 *     produces:
 *      - application/json
 *     responses:
 *       200:
 *         description: users
 *         schema:
 *           type: array
 *           items:
 *             $ref: '#/definitions/User'
 */
app.get('/users', (req, res) => {
```

Code comments are just facts waiting to become lies.

Vinai Kopp, A talk at #MM18IT


MILK
88¢

CODE FIRST VS DESIGN FIRST

Crashy McCiderface 🚴‍♂️🌍🔥
@philsturgeon

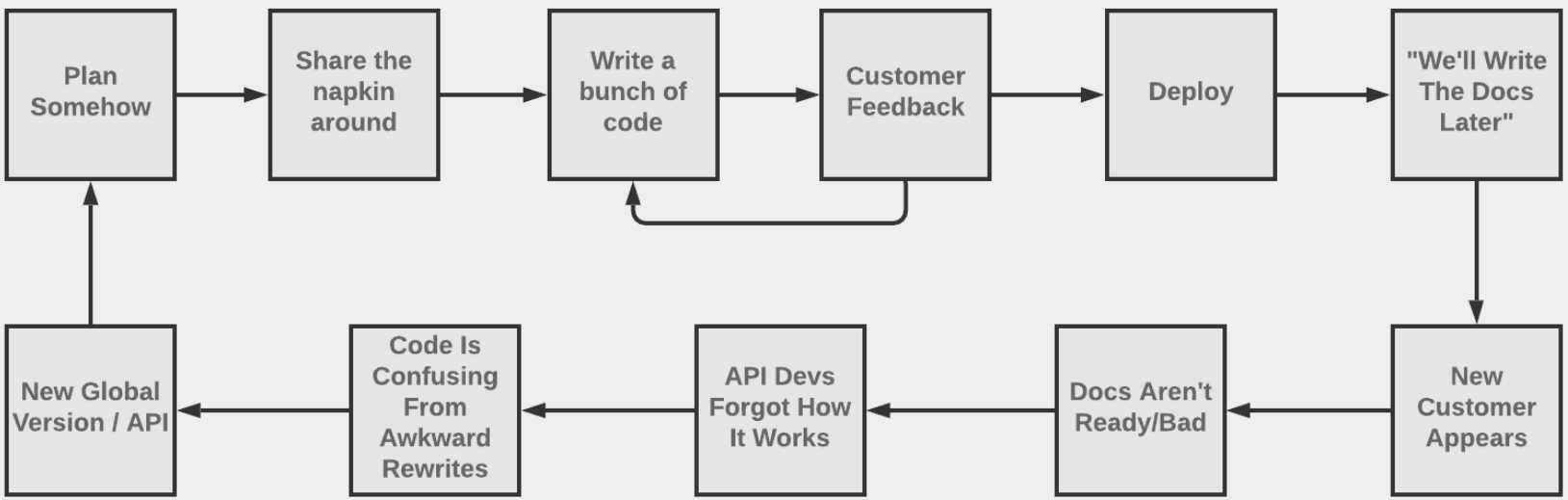
API Devs: Would you summarize the API development culture at your organization as:
A) entirely Design First
B) entirely Code First
C) used to be Code First, adopting Design First for new APIs
D) Some other awkward combination

A) Design-First	20.9%
B) Code First	23.3%
C) Adopting Design First	20.3%
D) Awkward Combination	**35.5%**

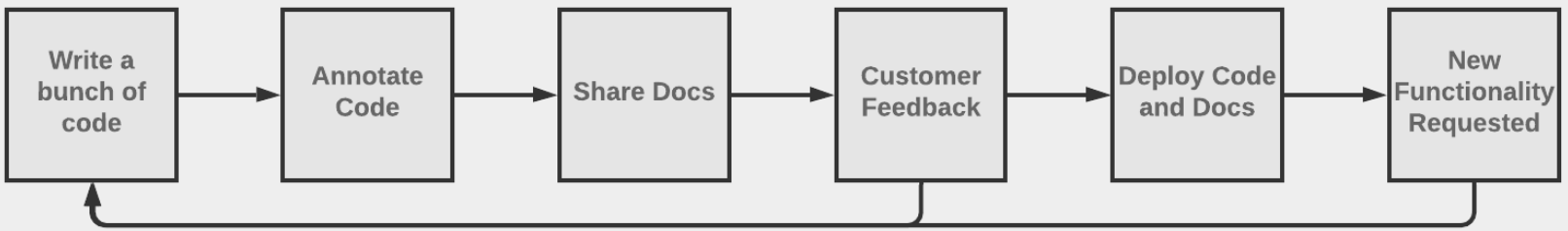
296 votes · Final results

1:28 PM · Mar 2, 2020 · Twitter Web App

Code First, Docs "When We Have Time"

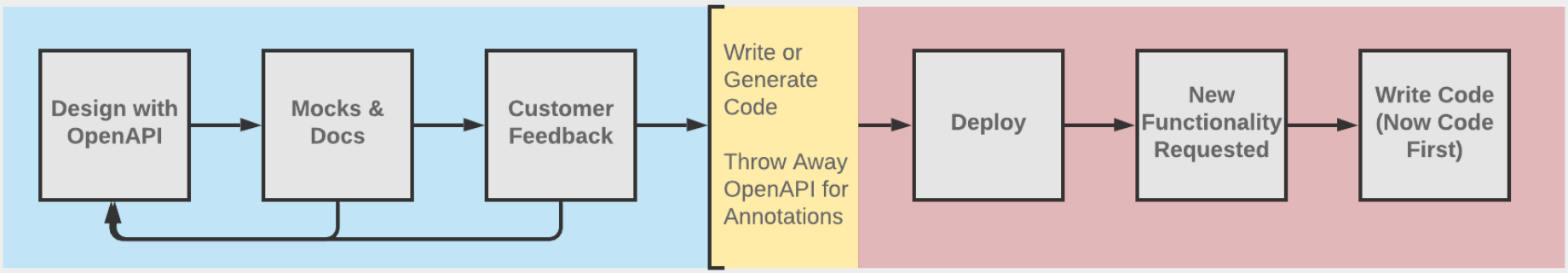


Code First, Generate Docs

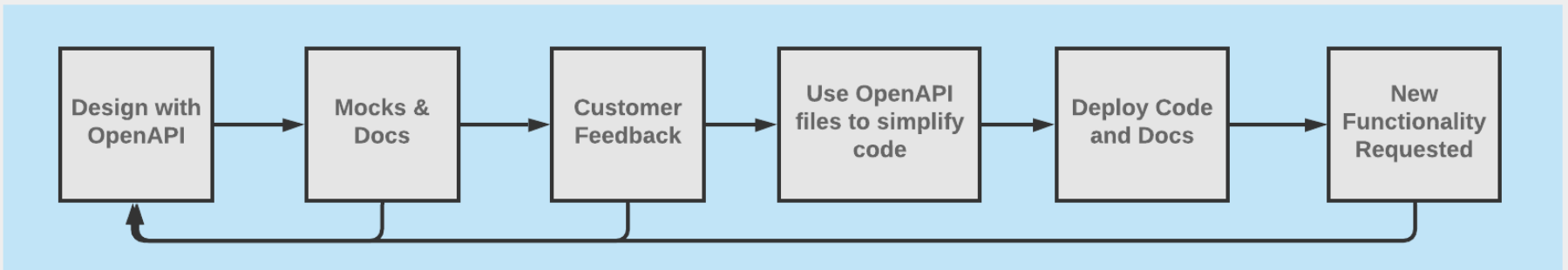


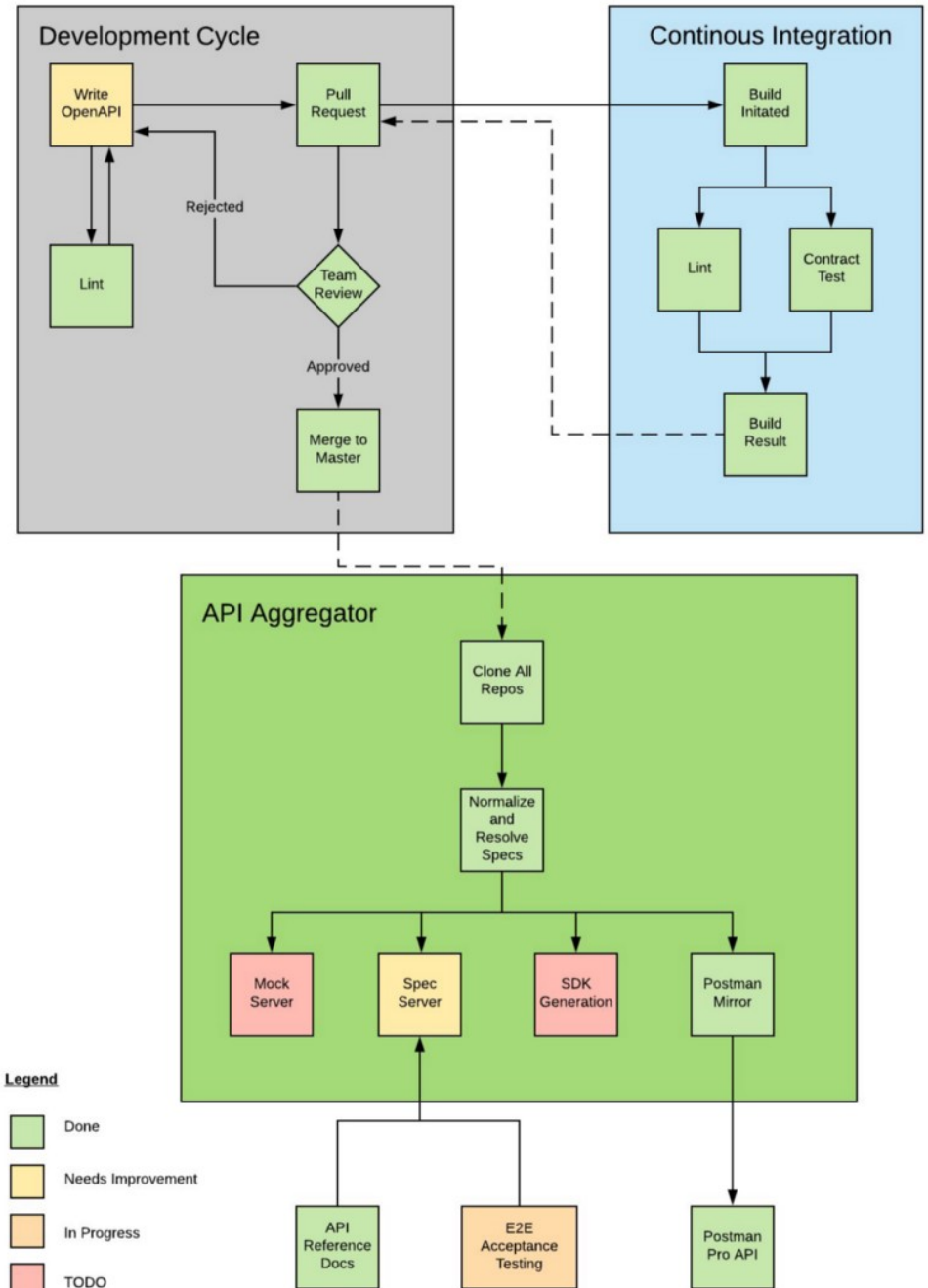
Most API planning is not machine readable. Especially napkins napkins.

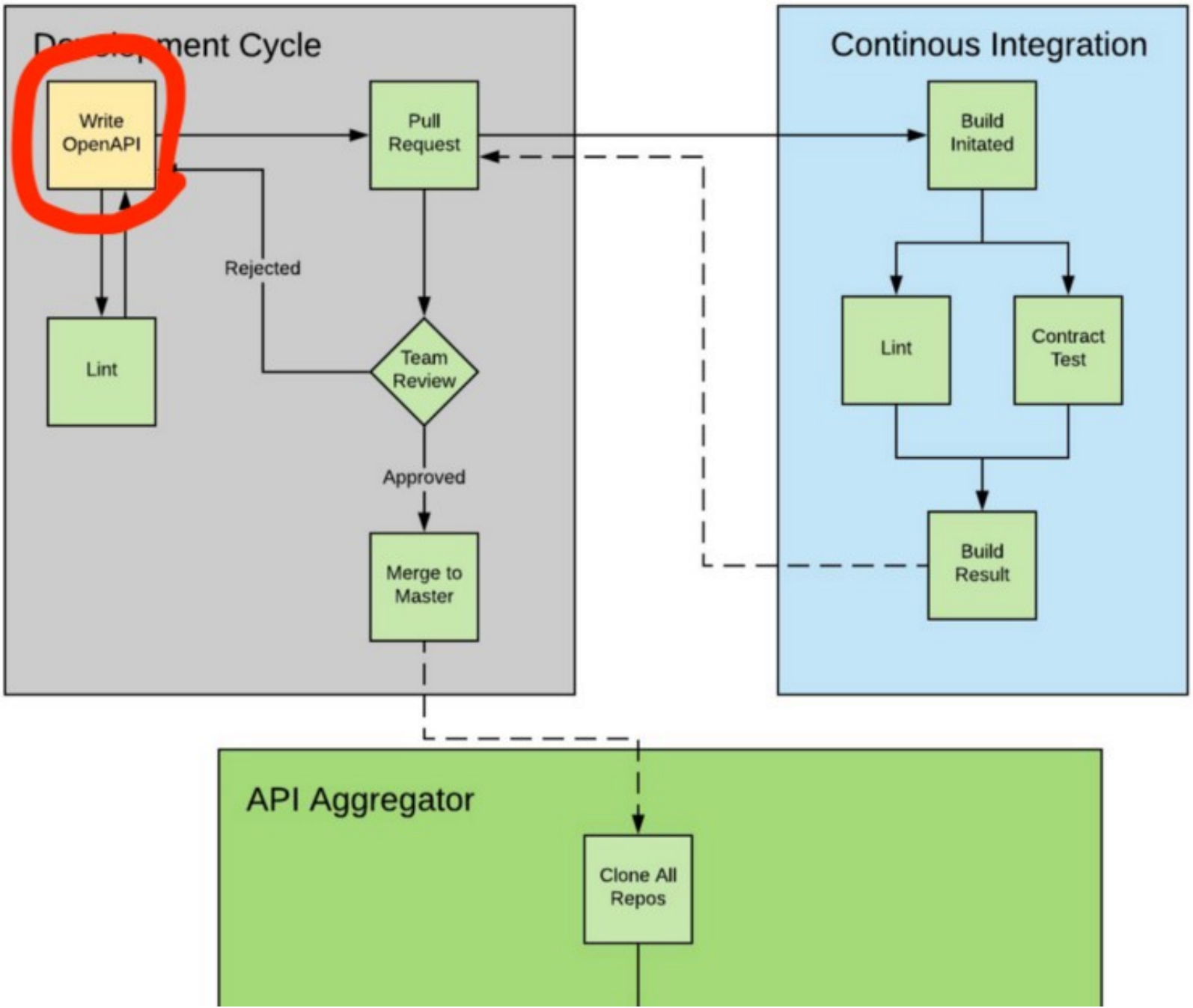
Design First, Ditch For Code First



Design First, Evolve With Code




Development Cycle
Write OpenAPI
Pull Request
Rejected
Lint
Team Review
Approved
Merge to Master
Continous Integration
Build Initated
Lint
Contract Test
Build Result
API Aggregator
Clone All Repos
Normalize and Resolve Specs
Mock Server
Spec Server
SDK Generation
Postman Mirror
Legend
Done
Needs Improvement
In Progress
TODO
API Reference Docs
E2E Acceptance Testing
Postman Pro API


ment Cycle
Write OpenAPI
Pull Request
Rejected
Lint
Team Review
Approved
Merge to Master
Continous Integration
Build Initated
Lint
Contract Test
Build Result
API Aggregator
Clone All Repos

While OpenAPI is great for describing APIs, ... ***it's definitely not a great experience to write OpenAPI documents from scratch.***

Sebastien Armand, "Making OpenAPI Bearable With Your Own DSL"

```
;;; ENTITIES
(define pet-entity
 (entity "Pet"
    'race (string "What kind of dog / cat this is (labrador, gold
    'origin (string "Country of origin" "Egypt")
    'birthday (datetime "Birth date of the pet" "2017-10-20T00:14
    'species (string "What kind of animal is this" "dog" #:enum '
(define $pet (schema-reference 'Pet pet-entity))

;;; RESPONSES
(define list-pets-response (jsonapi-paginated-response "List of p

;;; REQUESTS
(define pet-request (json-request "Pet Request Body" ($pet)))

;;; MAIN DOC
```

Where are the visual editors?

openapi.tools/#gui-editors

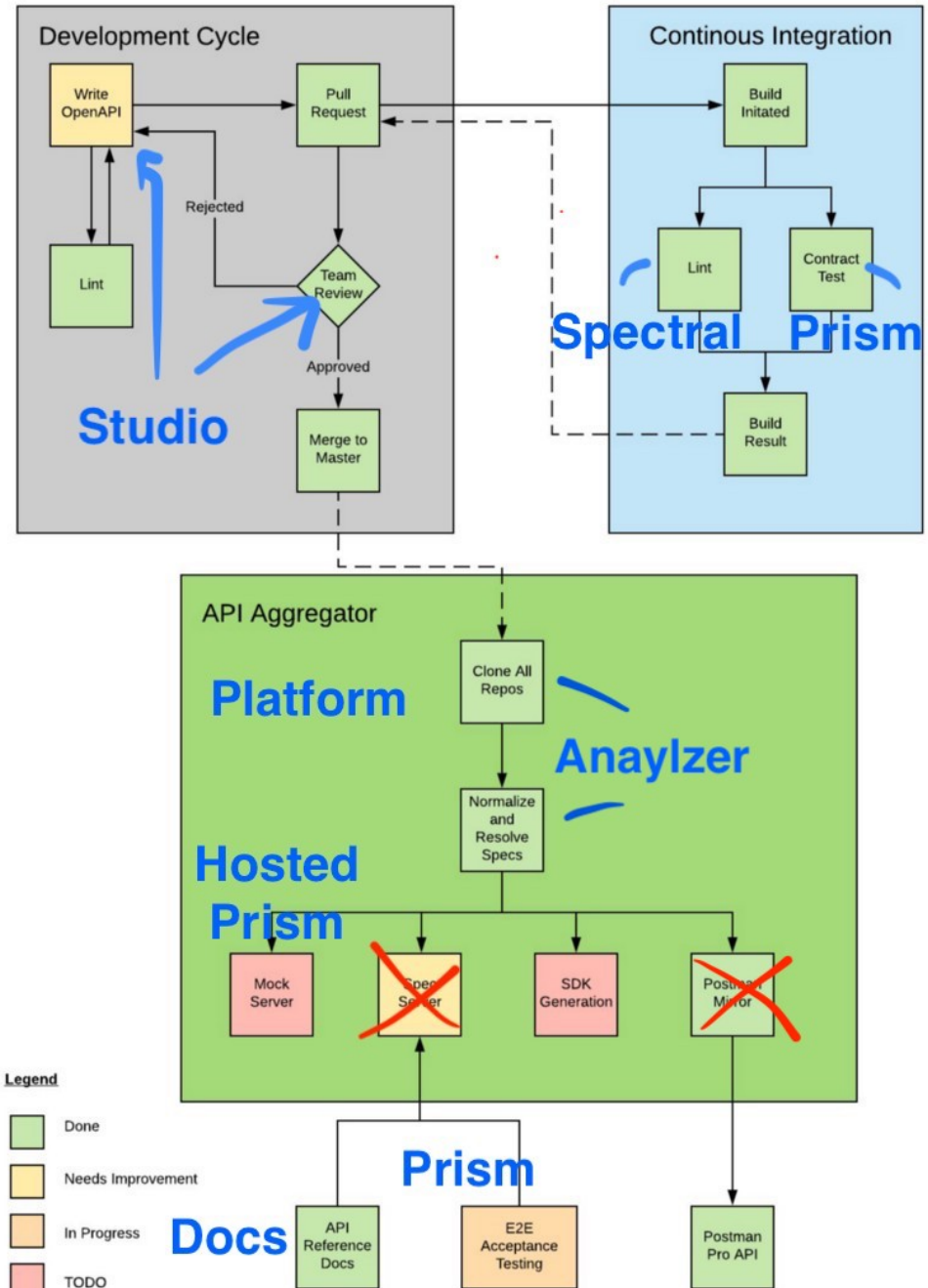
Stoplight

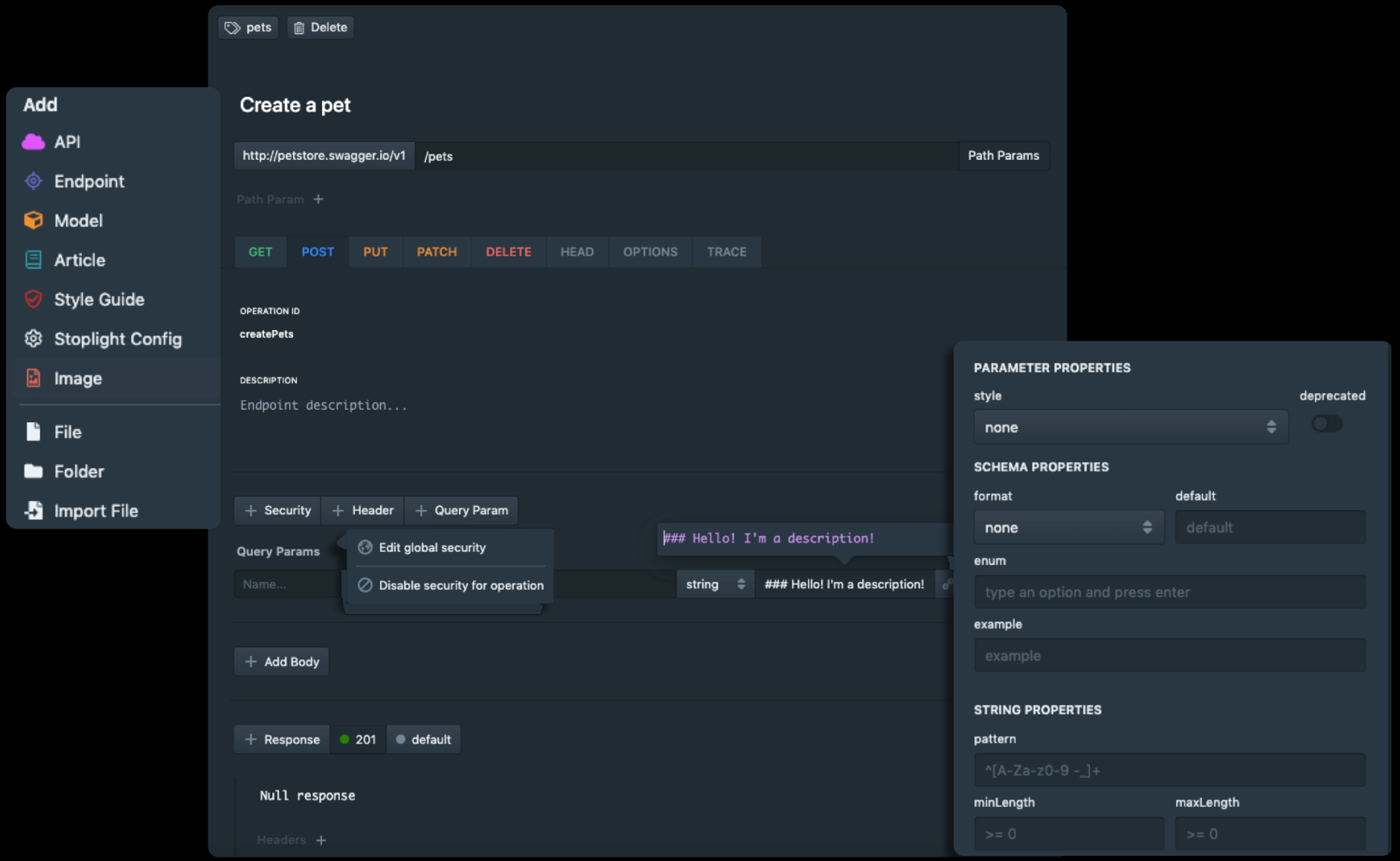
Use Cases Pricing Enterprise Learn Open Source Log in Get Started

Build Quality APIs Faster

Supercharge your API lifecycle with best-in-class tooling that integrates seamlessly into your Git workflows.

Get Started For Free

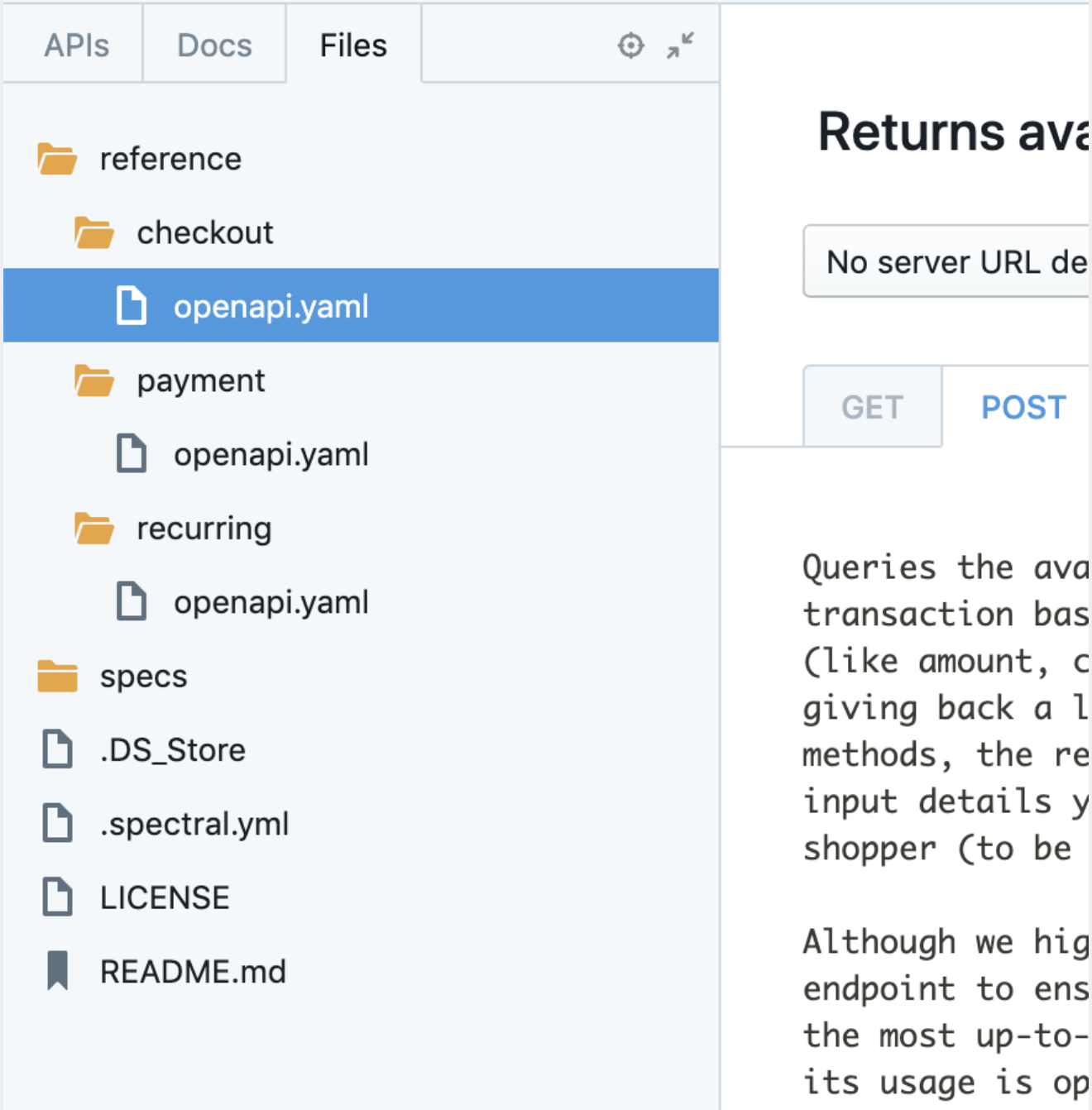

Development Cycle
Write OpenAPI
Pull Request
Rejected
Lint
Team Review
Approved
Merge to Master
Studio
Continous Integration
Build Initated
Lint
Contract Test
Spectral
Prism
Build Result
API Aggregator
Platform
Clone All Repos
Anaylzer
Normalize and Resolve Specs
Hosted Prism
Mock Server
SDK Generation
Legend
Done
Needs Improvement
In Progress
TODO
Prism
Docs
API Reference Docs
E2E Acceptance Testing
Postman Pro API

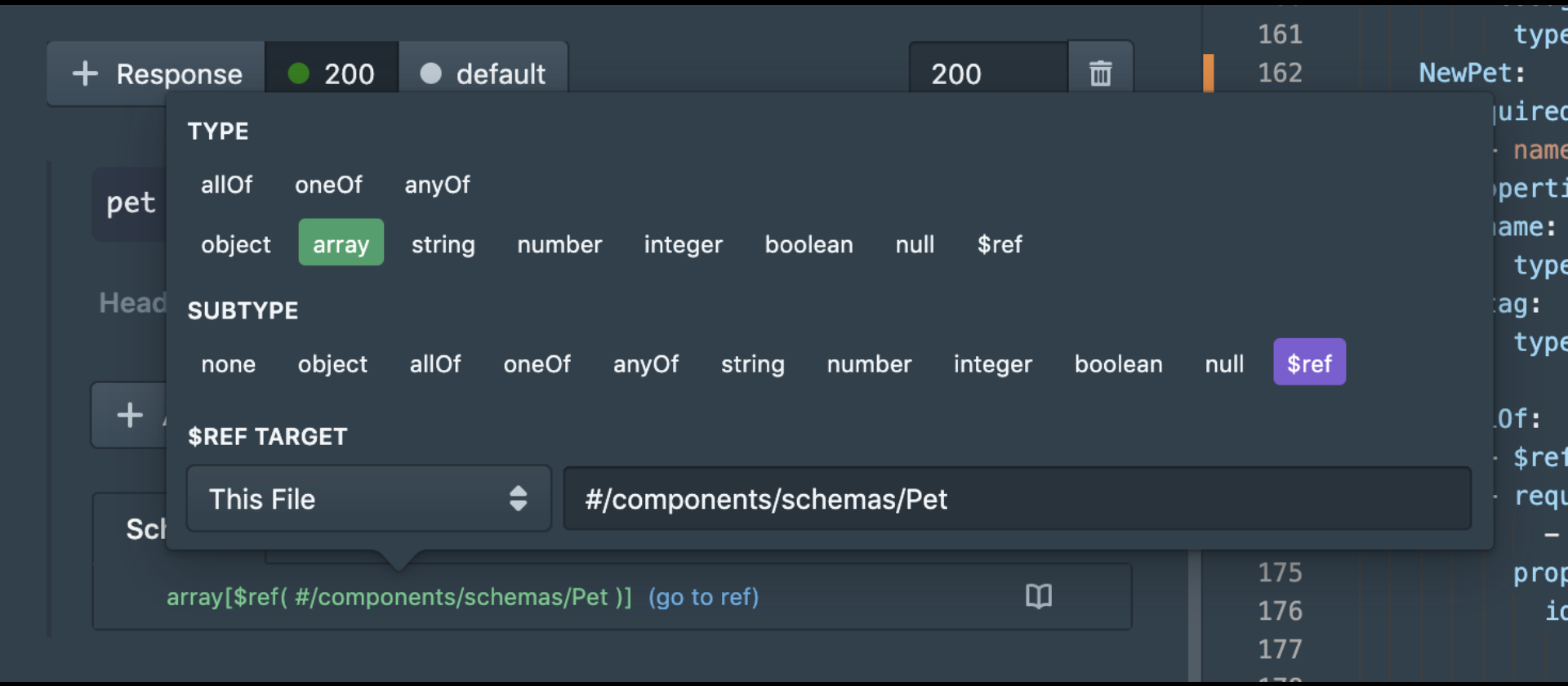

Add
API
Endpoint
Model
Article
Style Guide
Stoplight Config
Image
File
Folder
Import File
pets
Delete
Create a pet
http://petstore.swagger.io/v1
/pets
Path Params
Path Param +
GET
POST
PUT
PATCH
DELETE
HEAD
OPTIONS
TRACE
OPERATION ID
createPets
DESCRIPTION
Endpoint description...
+ Security
+ Header
+ Query Param
Query Params
Edit global security
Disable security for operation
Name...
string
Hello! I'm a description!
Hello! I'm a description!
+ Add Body
+ Response
201
default
Null response
Headers +
PARAMETER PROPERTIES
style
deprecated
none
SCHEMA PROPERTIES
format
default
none
default
enum
type an option and press enter
example
example
STRING PROPERTIES
pattern
^[A-Za-z0-9 -_]+
minLength
maxLength
>= 0
>= 0

Collaborate Flexibly with Git Workflows




Collaborate on API designs easily using your existing Git repositories. Stoplight fits seamlessly into your current development workflows.

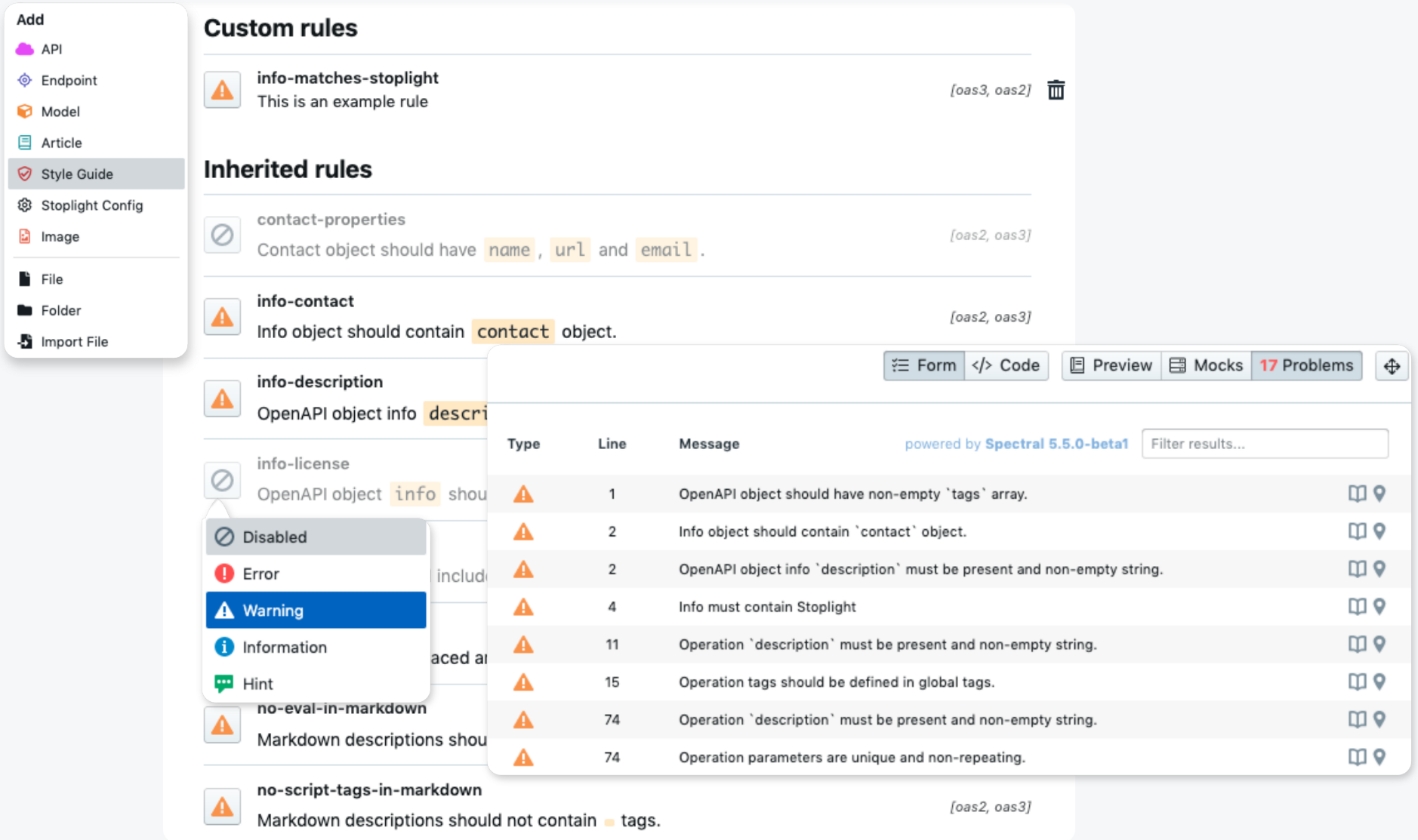

APIs
Docs
Files
reference
checkout
openapi.yaml
payment
openapi.yaml
recurring
openapi.yaml
specs
.DS_Store
.spectral.yml
LICENSE
README.md
Returns ava
No server URL de
GET
POST
Queries the ava
transaction bas
(like amount, c
giving back a l
methods, the re
input details y
shopper (to be
Although we hig
endpoint to ens
the most up-to-
its usage is op


+ Response
200
default
200
TYPE
allOf oneOf anyOf
object array string number integer boolean null $ref
SUBTYPE
none object allOf oneOf anyOf string number integer boolean null $ref
$REF TARGET
This File
#/components/schemas/Pet
pet
array[$ref(#/components/schemas/Pet)] (go to ref)
161
162
NewPet:
175
176
177

```
40        description: OK - the request has succeeded.
41        schema:
42          $ref: '#/definitions/PaymentMethodsResponse'
```

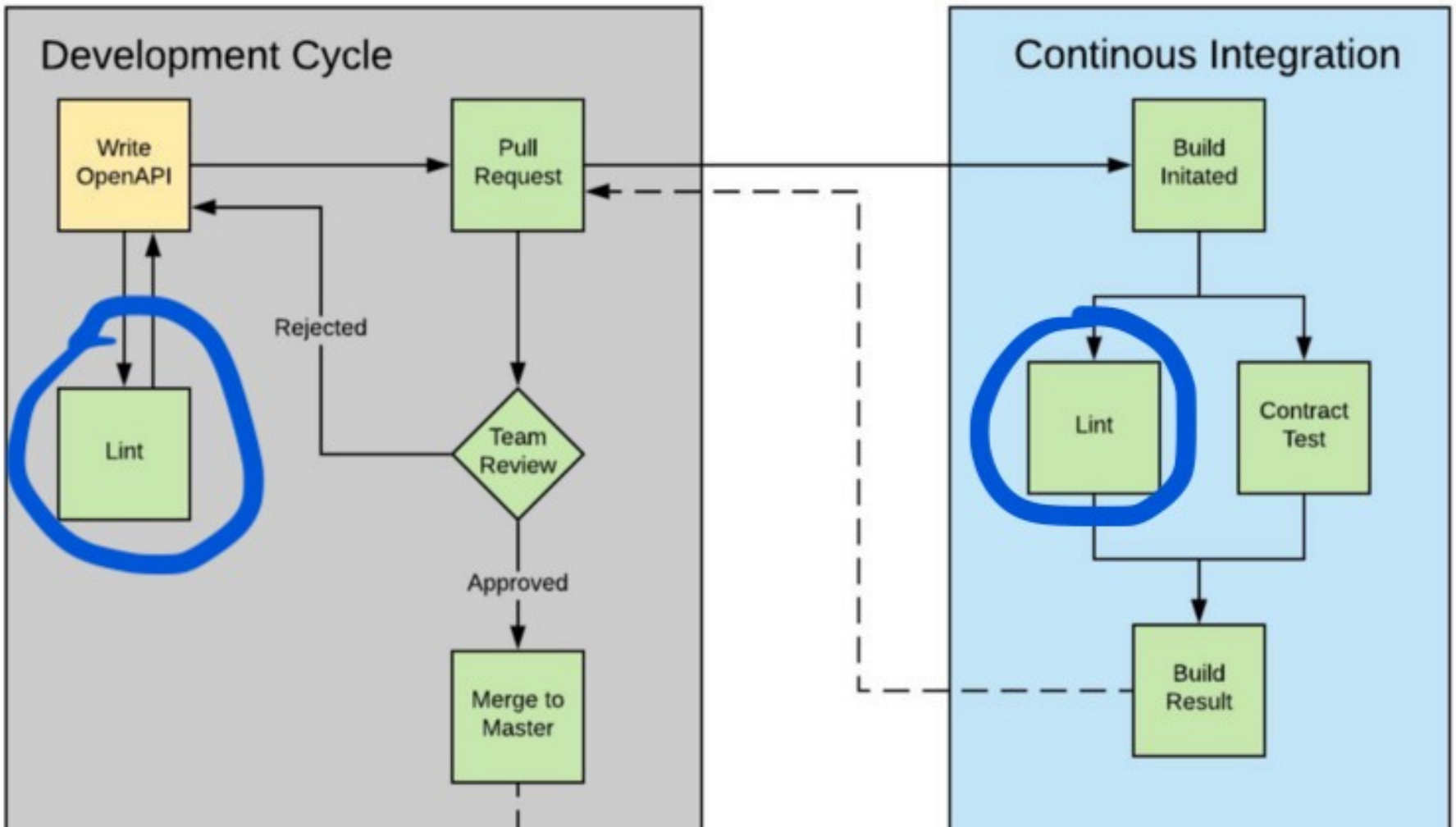
Type	Line	Message	powered by Spectral
⚠	84	Operation should have non-empty `tags` array.	
⚠	86	Operation summary should be succinct, no full stops, and less than ...	
⚠	86	Operation summary should be succinct, no full stops, and less than ...	
⚠	96	Parameters must have a description.	
⚠	96	Parameters must have a description.	
⚠	96	Parameters must have a description.	
⚠	96	Parameters must have a description.	
⚠	96	Parameters must have a description.	

Mock Server | 244 Issues


Add
API
Endpoint
Model
Article
Style Guide
Stoplight Config
Image
File
Folder
Import File
Custom rules
info-matches-stoplight
This is an example rule
[oas3, oas2]
Inherited rules
contact-properties
Contact object should have name , url and email .
[oas2, oas3]
info-contact
Info object should contain contact object.
[oas2, oas3]
info-description
OpenAPI object info descri
info-license
OpenAPI object info shou
Disabled
Error
Warning
Information
Hint
no-eval-in-markdown
Markdown descriptions shou
no-script-tags-in-markdown
Markdown descriptions should not contain tags.
[oas2, oas3]
Form
Code
Preview
Mocks
17 Problems
Type
Line
Message
powered by Spectral 5.5.0-beta1
Filter results...
1 OpenAPI object should have non-empty `tags` array.
2 Info object should contain `contact` object.
2 OpenAPI object info `description` must be present and non-empty string.
4 Info must contain Stoplight
11 Operation `description` must be present and non-empty string.
15 Operation tags should be defined in global tags.
74 Operation `description` must be present and non-empty string.
74 Operation parameters are unique and non-repeating.

API DESCRIPTION LINTING

a.k.a "API Style Guides"


Development Cycle
Write OpenAPI
Pull Request
Lint
Rejected
Team Review
Approved
Merge to Master
Continous Integration
Build Initated
Lint
Contract Test
Build Result

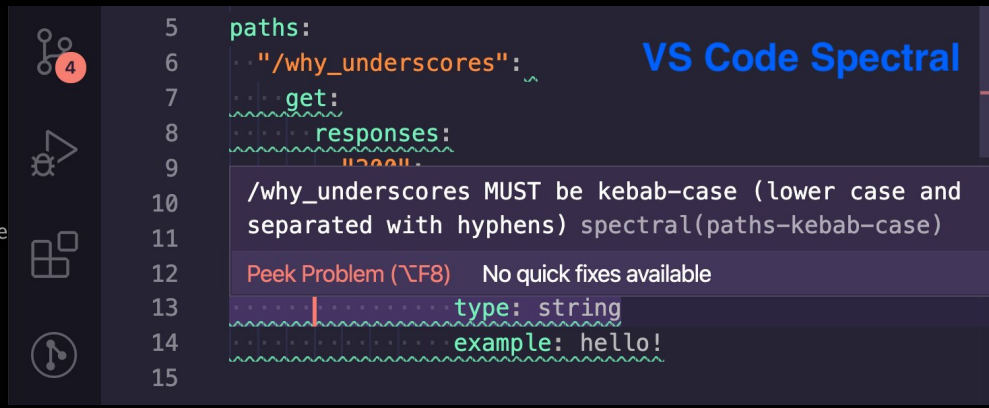
Spectral CLI

```
▶ spectral lint -r apisyouwonthate.yml example.yaml
OpenAPI 3.x detected

/Users/phil/src/openapi-contrib/style-guides/example.yaml
 6:22  warning  paths-kebab-case  /why_underscores should be kebab-case (lowe
nd separated with hyphens)

✖ 1 problem (0 errors, 1 warning, 0 infos, 0 hints)
```

VS Code Spectral



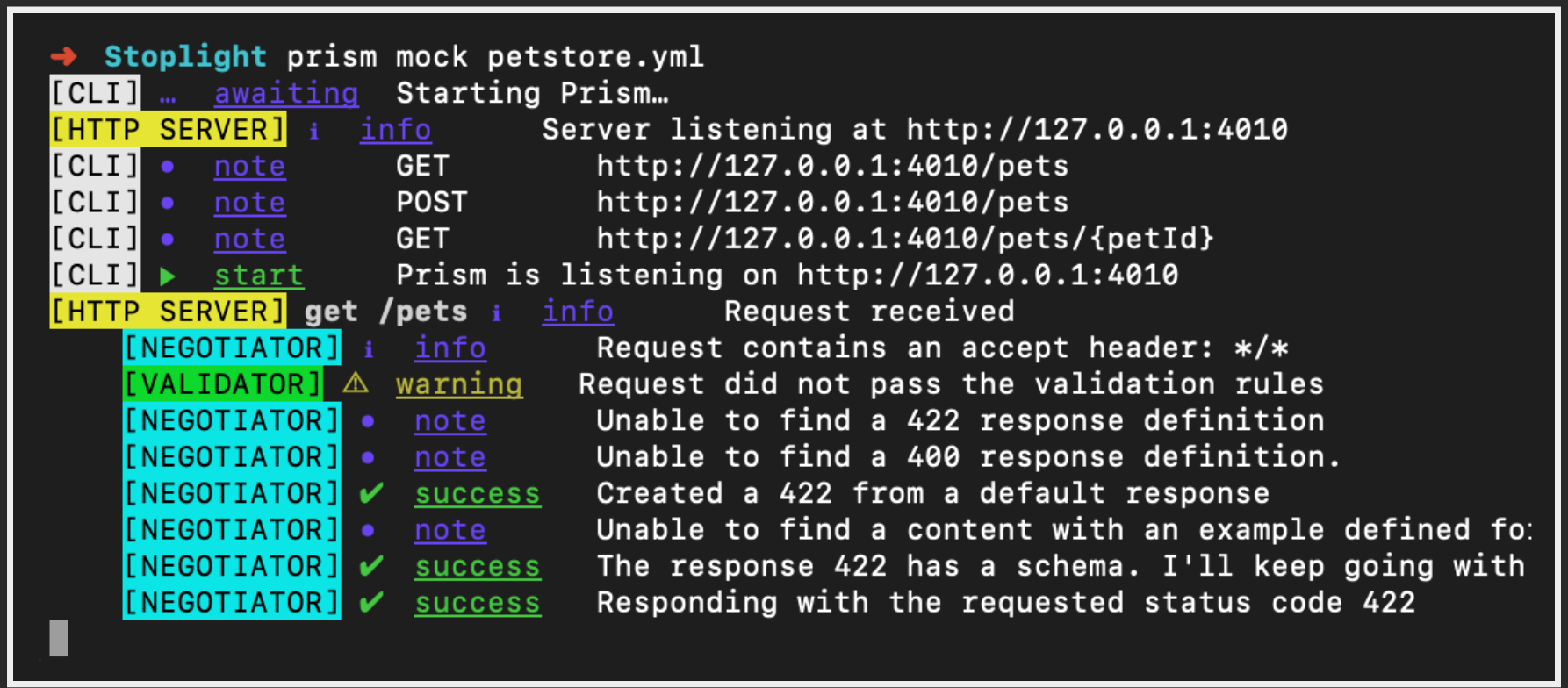

5 paths:
6 "/hello":
7 get:
8 responses:
9 "400":
10 content:
11 nonsense/json:
12 schema:
13 type: string
14 example: hello!
15
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 1 Filter. E.g.: text, **/*.ts, !**/node_modules/**
example.yaml 1
Every error response SHOULD support either RFC 7807 or the JSON:API Error format. spectral(unknown-error-format) [11, 27]

How/when do we create mocks?

1. Local Mock Servers
2. Hosted Mock Servers

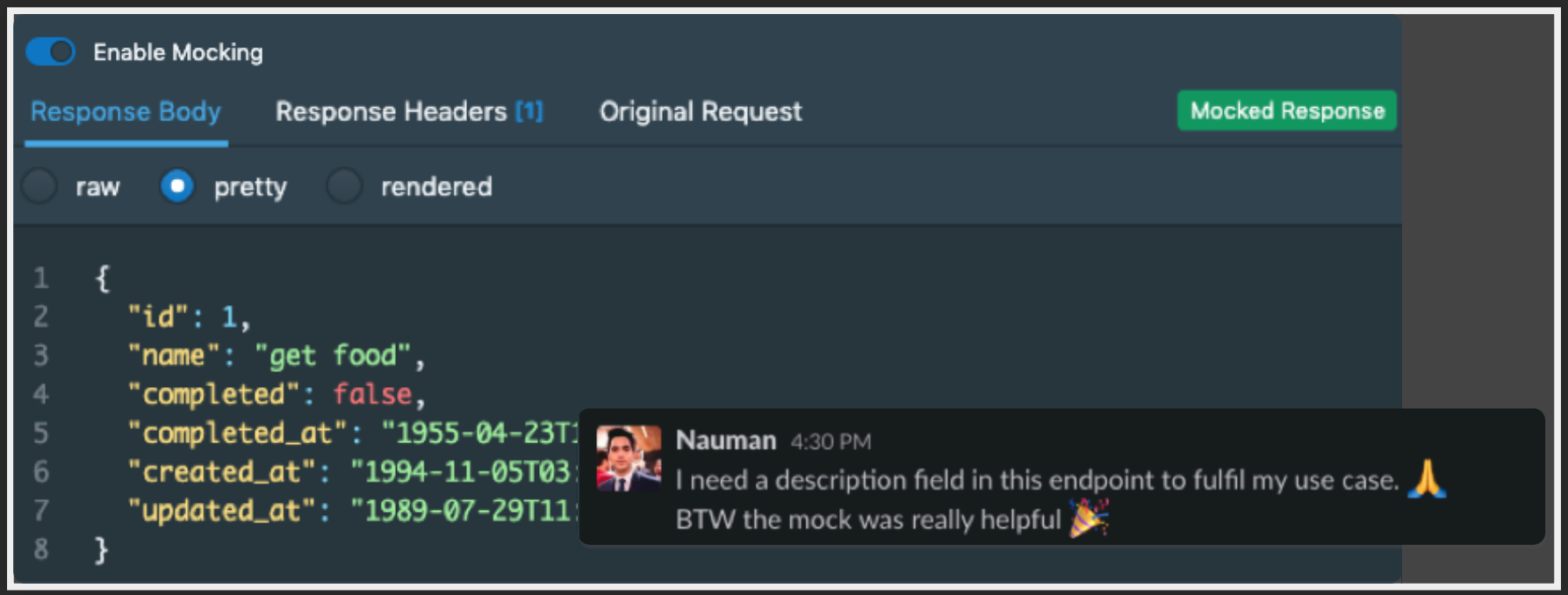
http://openapi.tools/#mock

Local Mock Server



Hosted Mock Server

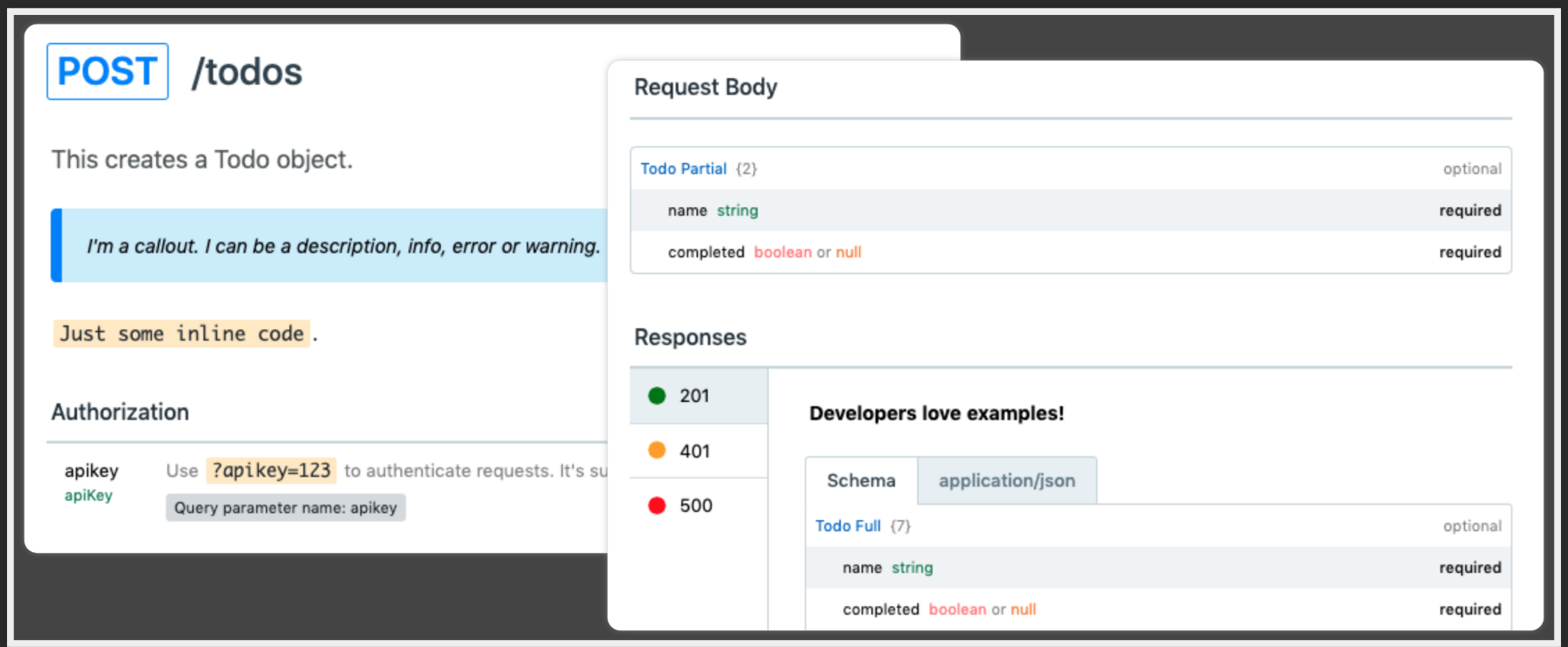
`https://acme.stoplight.io/mocks/widgets`



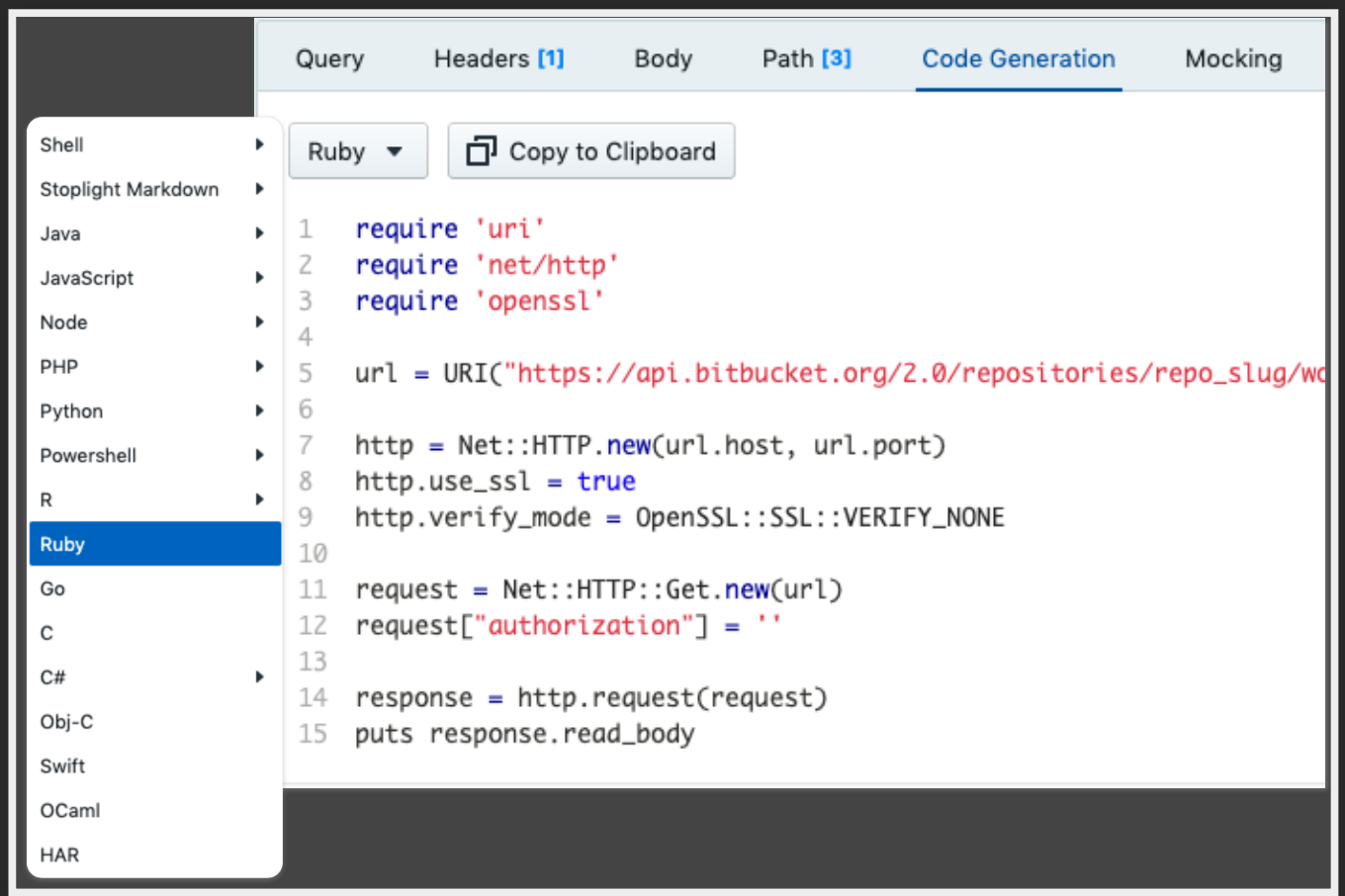
How/when do we create documentation?

1. CI deploys HTML to S3
2. Read from Git repo on push

Automatic generated API Reference docs



Automatic generated Code Samples



⛔ How do we keep code and API docs in sync?

API

name
string
max length: 20

email
format: email

age
type: integer
minimum: 18

Ref Docs

name
string
max length: 20

email
format: email

age
type: integer
minimum: 18

✅ How to re-use API descriptions as code!

SERVER-SIDE VALIDATION

- Model?

SERVER-SIDE VALIDATION

- Model?
- Controller?

SERVER-SIDE VALIDATION

- Model?
- Controller?
- View? 😱

SERVER-SIDE VALIDATION

- Model?
- Controller?
- View? 😱
- Service?

SERVER-SIDE VALIDATION

- Model?
- Controller?
- View? 😱
- Service?
- Contract?

SERVER-SIDE RUBY VALIDATION

```
class NewUserContract < Dry::Validation::Contract
  params do
    required(:email).filled(:string)
    required(:age).value(:integer)
  end

  rule(:email) do
    unless /\A[\w+\-.]+@[a-z\d\-]+(\.[a-z\d\-]+)*\.[a-z]+\z/i.mat
      key.failure('has invalid format')
    end
  end

  rule(:age) do
    key.failure('must be greater than 18') if value < 18
  end
end
```

SERVER-SIDE JAVASCRIPT VALIDATION

```
import Joi from 'joi';

export default {
  createUser: {
    body: {
      username: Joi.string().regex(/^[0-9a-fA-F]{24}$/).required(
      email: Joi.string().required(),
      age: Joi.number()
    }
  }
};
```

LOOKS LIKE AN API DESCRIPTION

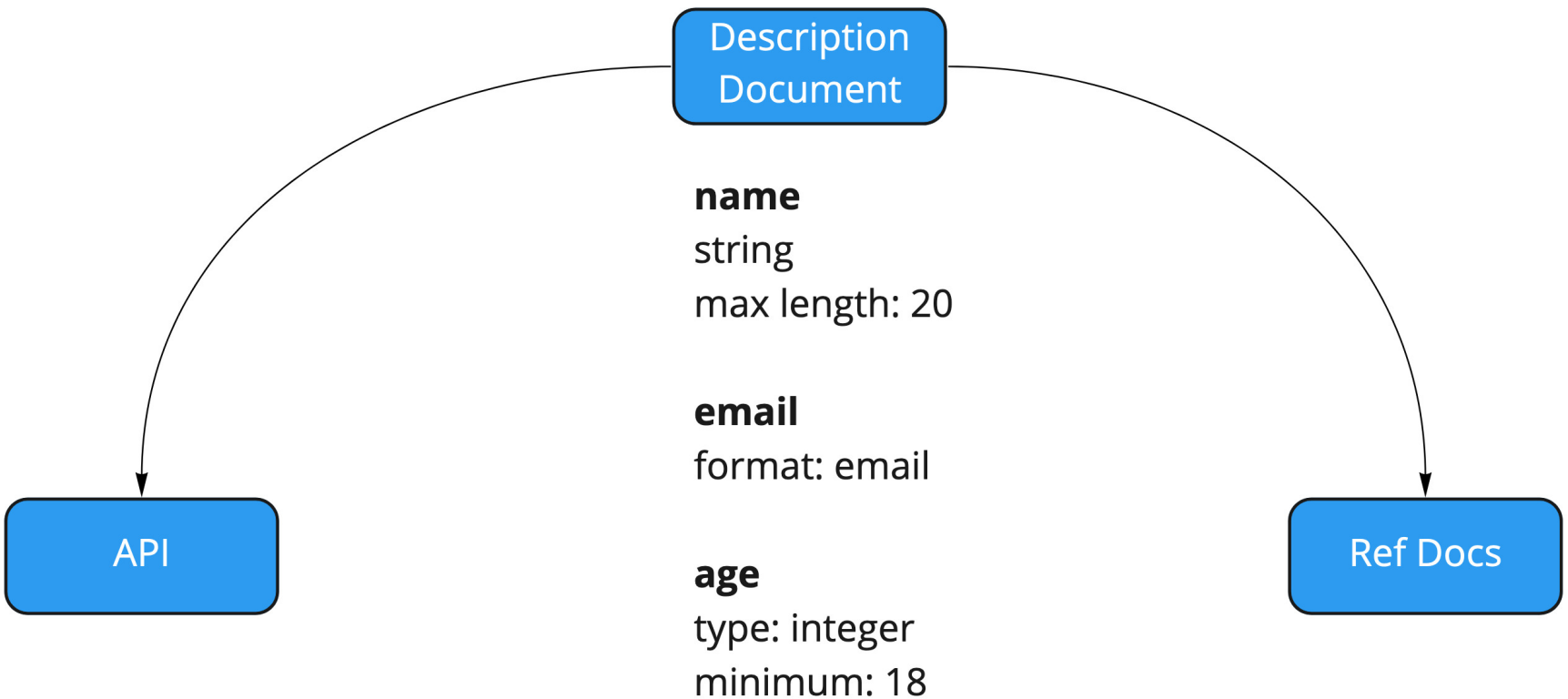
```yaml
type: object
properties:
  username:
    type: string
  email:
    type: string
    format: email
  age:
    type: integer
    minimum: 18
required:
- email
- age
```

Multiple Sources Of Truth = Checking For LIES

Dredd, takes a lot of work and breaks so much teams just turn it off

https://dredd.org/

REUSE API DESCRIPTIONS FOR VALIDATION


Description Document
name
string
max length: 20
email
format: email
age
type: integer
minimum: 18
API
Ref Docs
miro

OpenAPI Validation Middlewares can reject invalid requests with "no code".

```
# config/application.rb

config.middleware.use Committee::Middleware::RequestValidation,
  schema_path: 'openapi.yaml'
```

```php
use League\OpenAPIValidation\PSR15\ValidationMiddlewareBuilder;
use League\OpenAPIValidation\PSR15\SlimAdapter;

$psr15Middleware = (new ValidationMiddlewareBuilder)
  ->fromYamlFile('openapi.yaml')
  ->getValidationMiddleware();

/** @var \Slim\App $app */
$app->add(new SlimAdapter($slimMiddleware));
```

```
{
  "errors": [
    {
      "status": "422",
      "detail": "The property '#/widget/price' of type string did
      "source": {
        "pointer": "#/widget/price"
      }
    }
  ]
}
```

- **Ruby:** committee
- **PHP:** league/openapi-psr7-validator
- **NodeJS:** fastify or express-ajv-swagger-validation
- **Java/Kotlin:** openapi-spring-webflux-validator
- **Python:** connexion
- **Mojolicious:** Mojolicious

If you have validation code, you can delete it.

If this is a new application, you **don't need to write it**.

"Data-store Validations" like "is email unique" still need to happen 👍

Your integration/e2e test suite proves the **requests** work as expected.

Documented requests are now *proven* to be correct.

What about responses?

Middlewares *can* validate responses, but why waste time in prod?

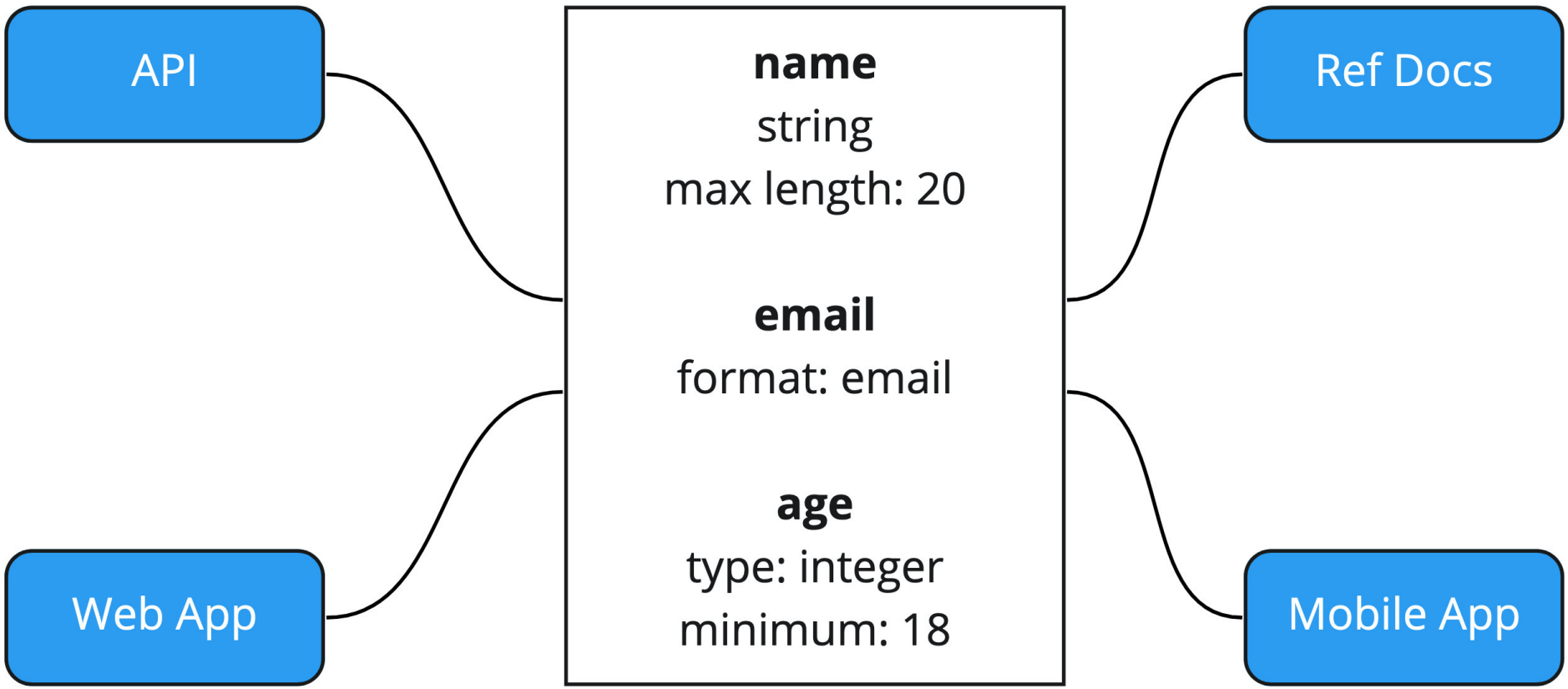
Your test suite can contract test **responses** with OpenAPI.

```yaml
responses:
  "200":
    description: OK
    content:
      application/json:
        schema:
          $ref: ./models/user.json
```

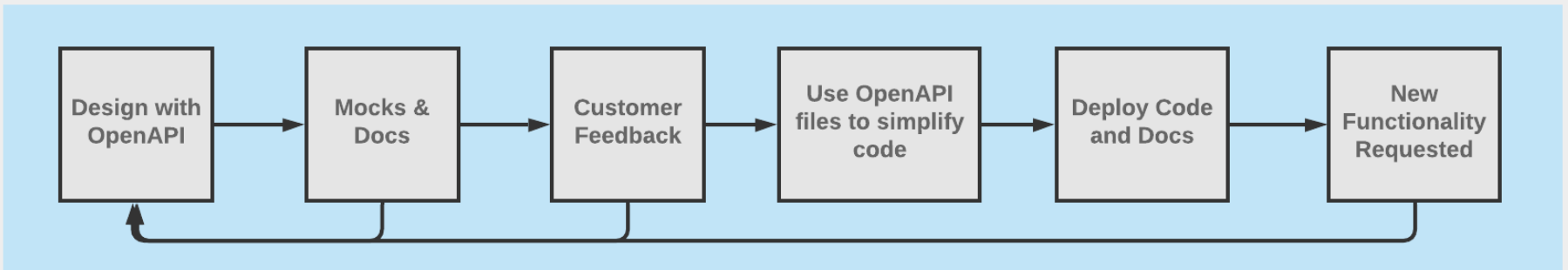
```
JsonMatchers.schema_root = "api/models/"

it 'should conform to user schema' do
  get "/users/#{subject.id}"
  expect(response).to match_json_schema('user')
end
```

https://apisyouwonthate.com/blog/writing-documentation-via-contract-testing


API
Web App
name
string
max length: 20
email
format: email
age
type: integer
minimum: 18
Ref Docs
Mobile App

Design First, Evolve With Code



THANK YOU!

ApisYouWontHate.com

Stoplight.io

@philsturgeon