



Please

**Ask questions
through the app**

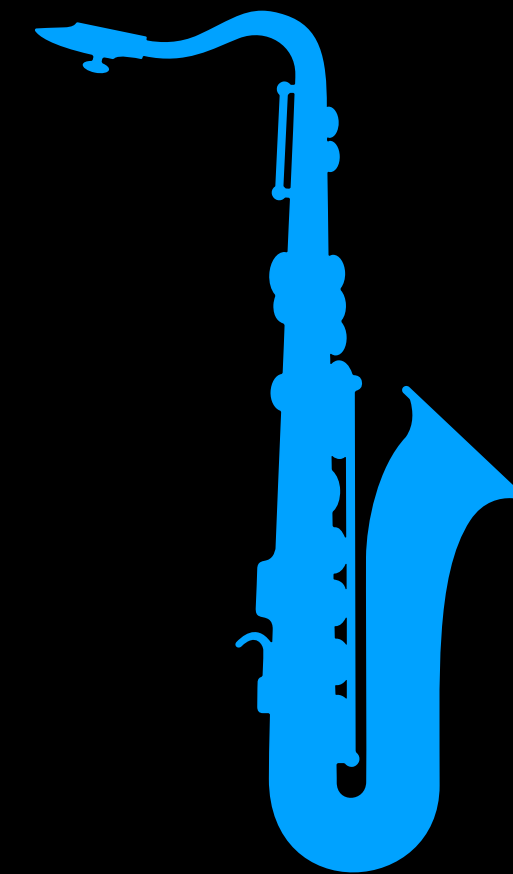


Rate Session

Thank you!

Microservice API Management, Discovery, Documentation and all that Jazz

Graham Brooks



ThoughtWorks®



 [@grahamcbrooks](https://twitter.com/grahamcbrooks)

 graham@grahambrooks.com

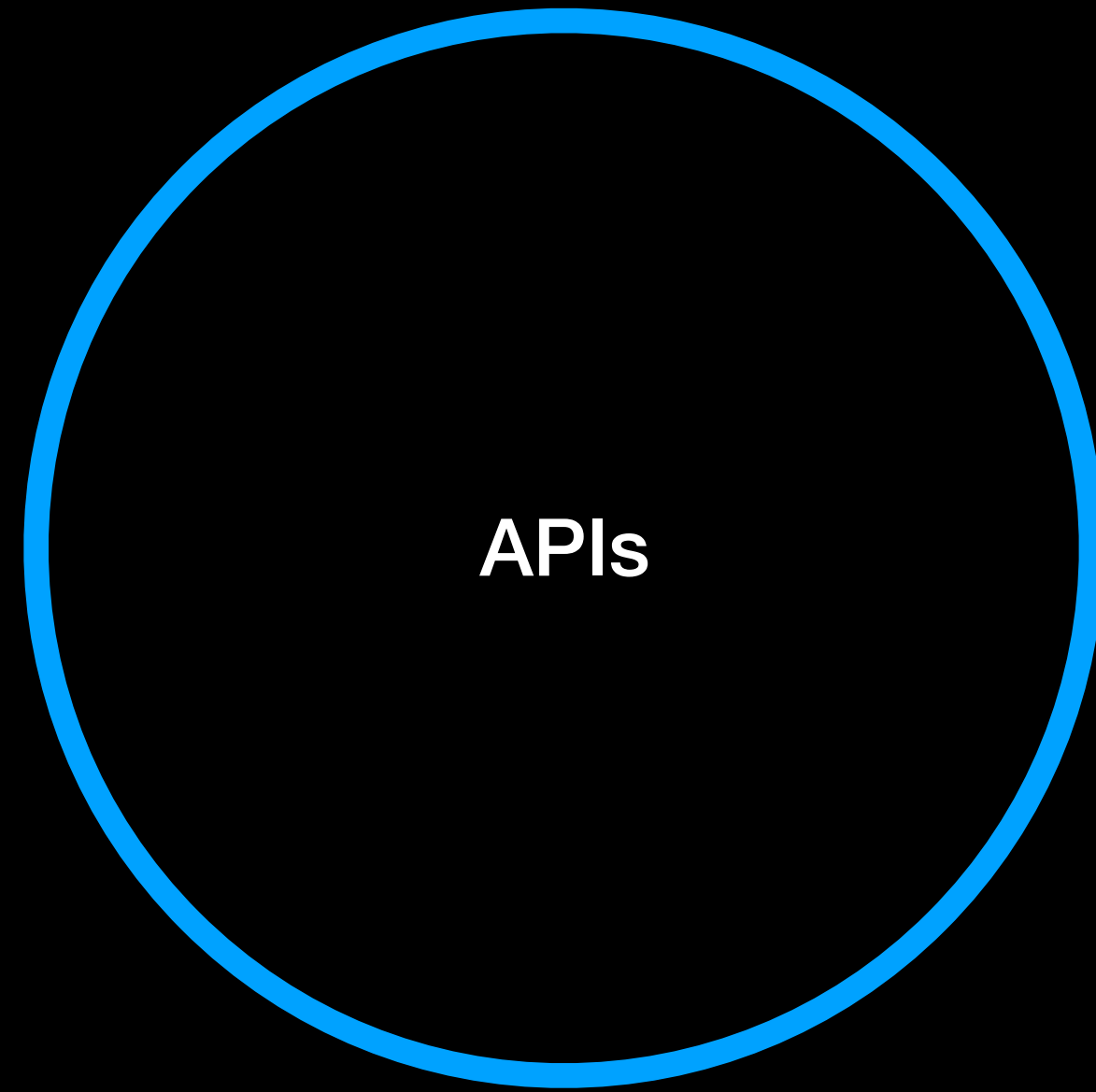
 gbrooks@thoughtworks.com

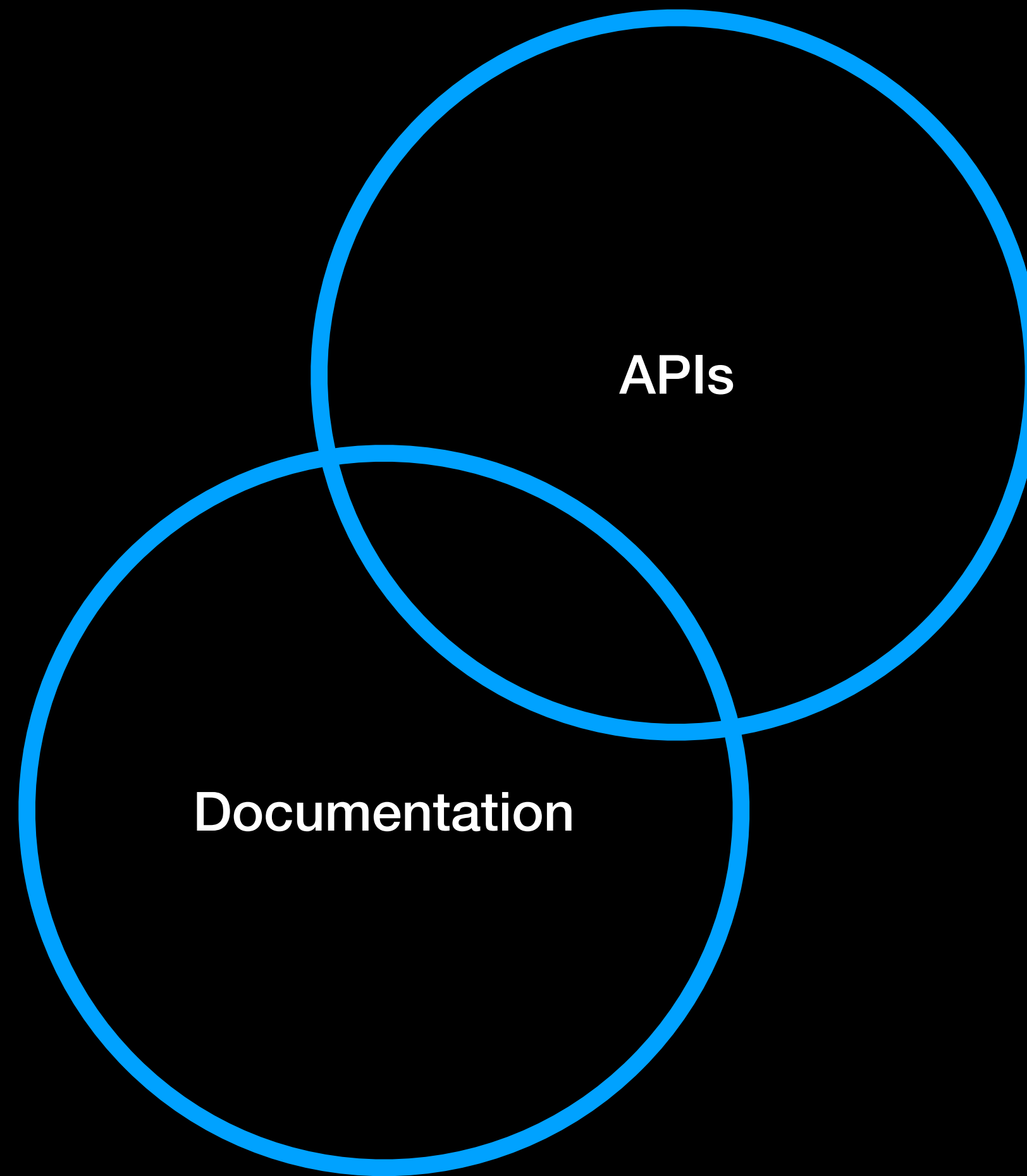
 grahambrooks.com/talks

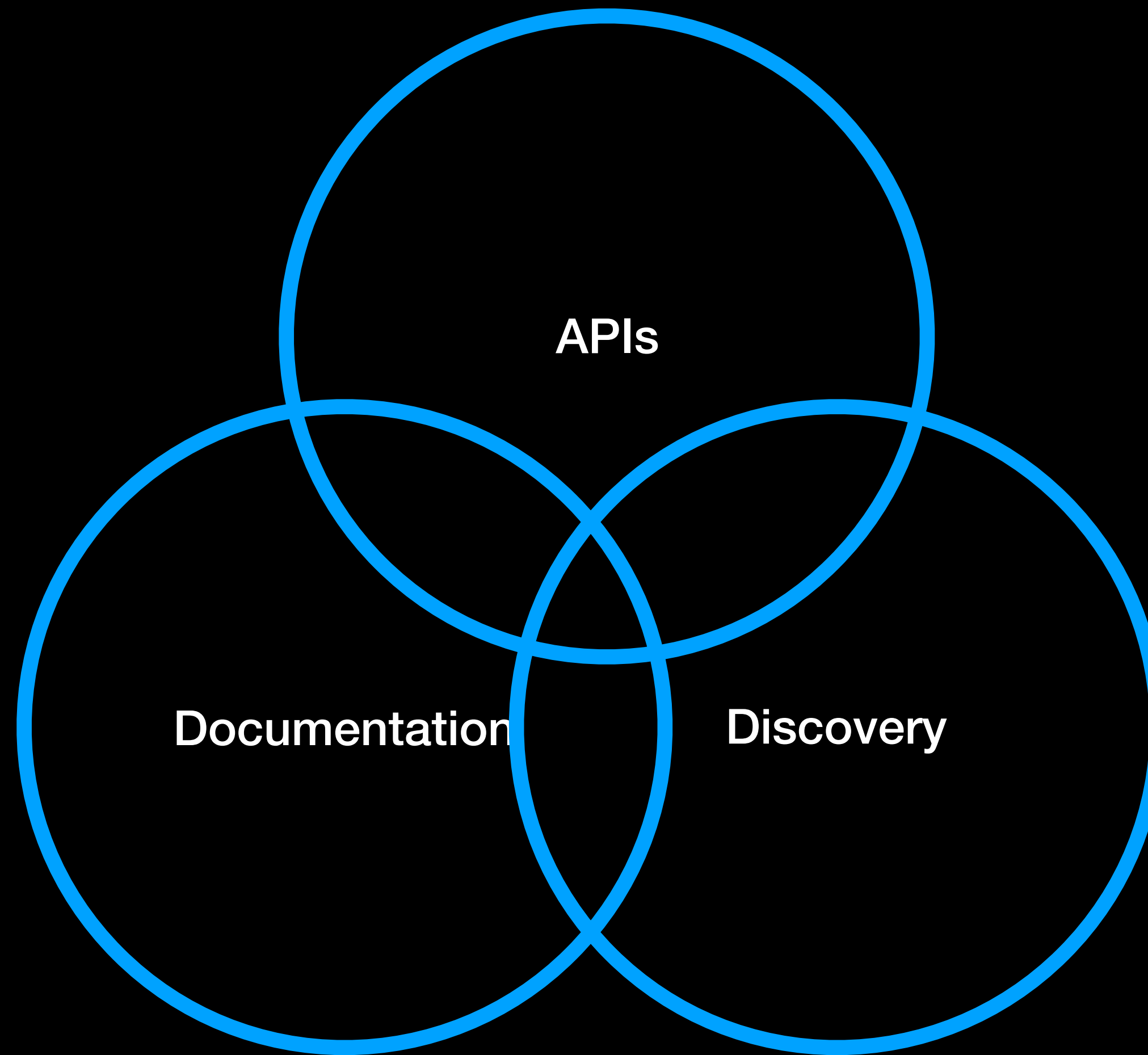


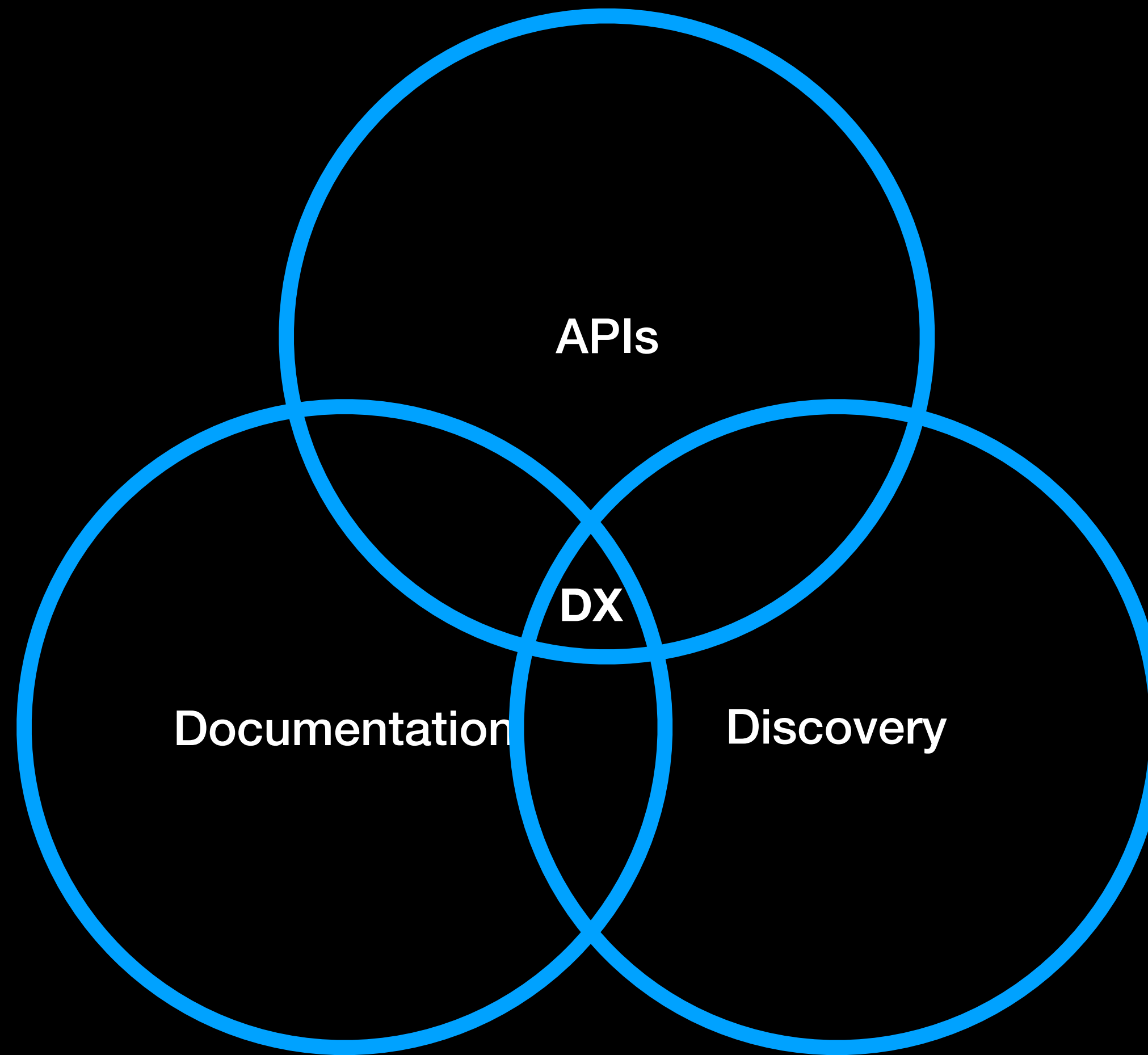
ThoughtWorks®











DX

Developer eXperience

1993



1993

Distributed Component Object Model

Distributed Component Object Model - DCOM



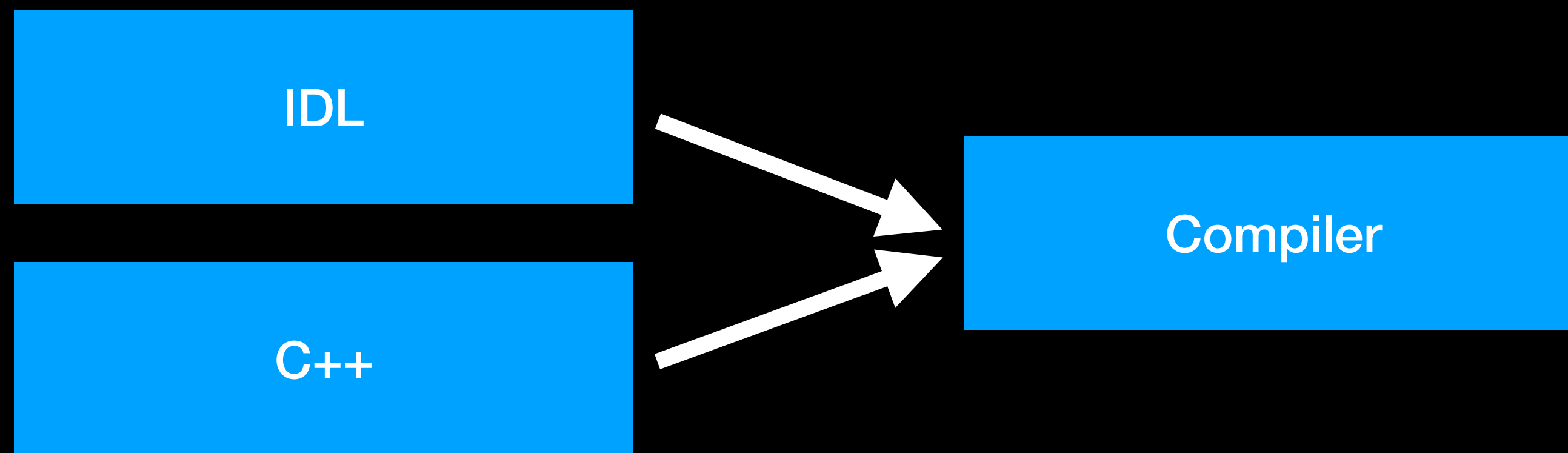
IDL

Distributed Component Object Model - DCOM

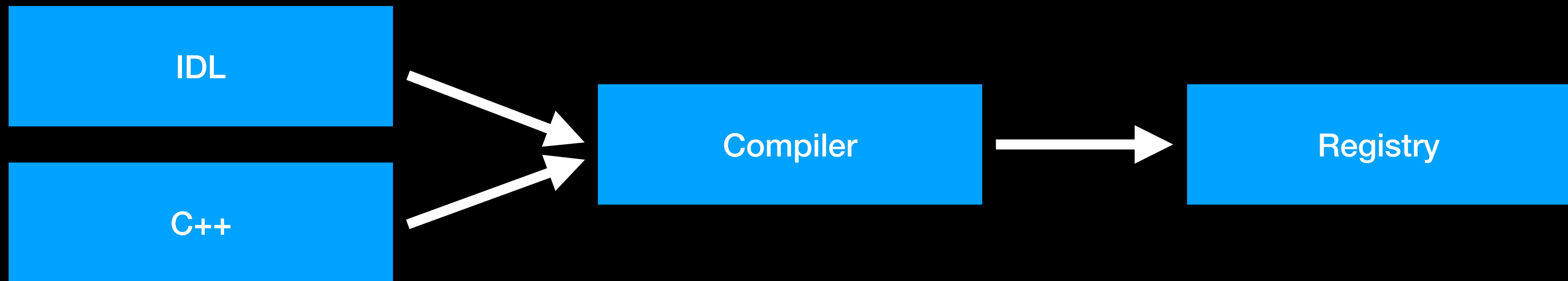
IDL

C++

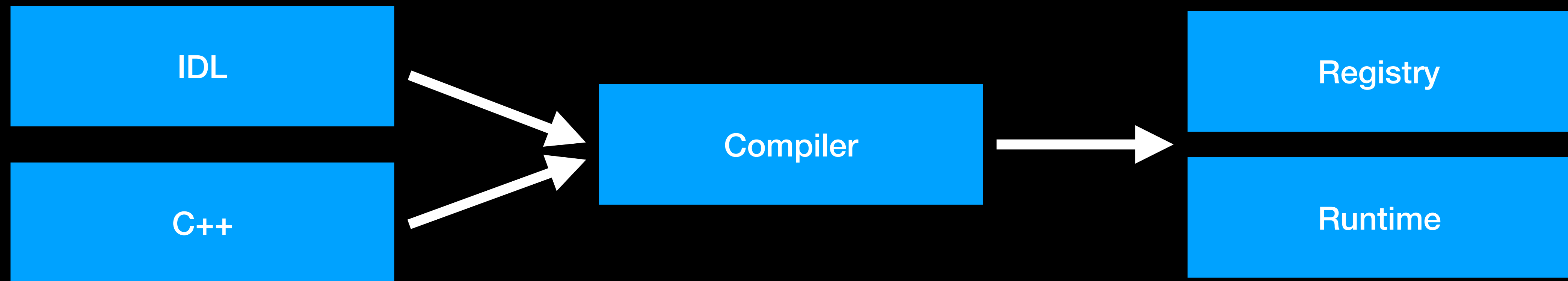
Distributed Component Object Model - DCOM



Distributed Component Object Model - DCOM



Distributed Component Object Model - DCOM



Closed





Tech Specific



Tight Coupling



Poor Distributed lifecycle



API

Monolith

Authentication

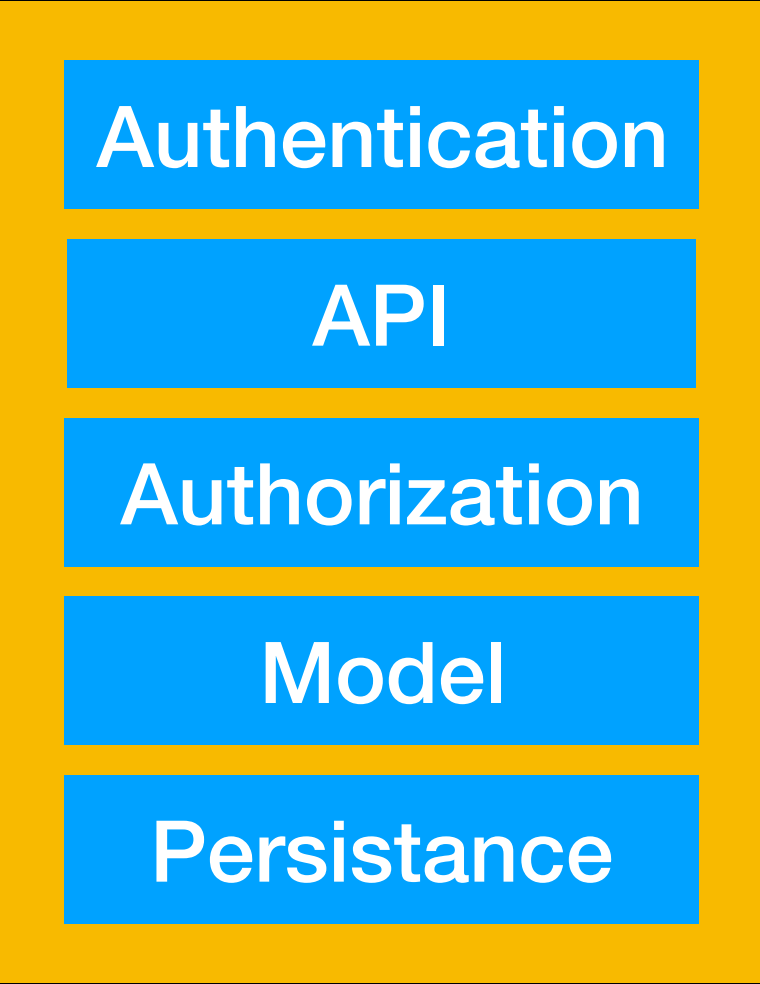
API

Authorization

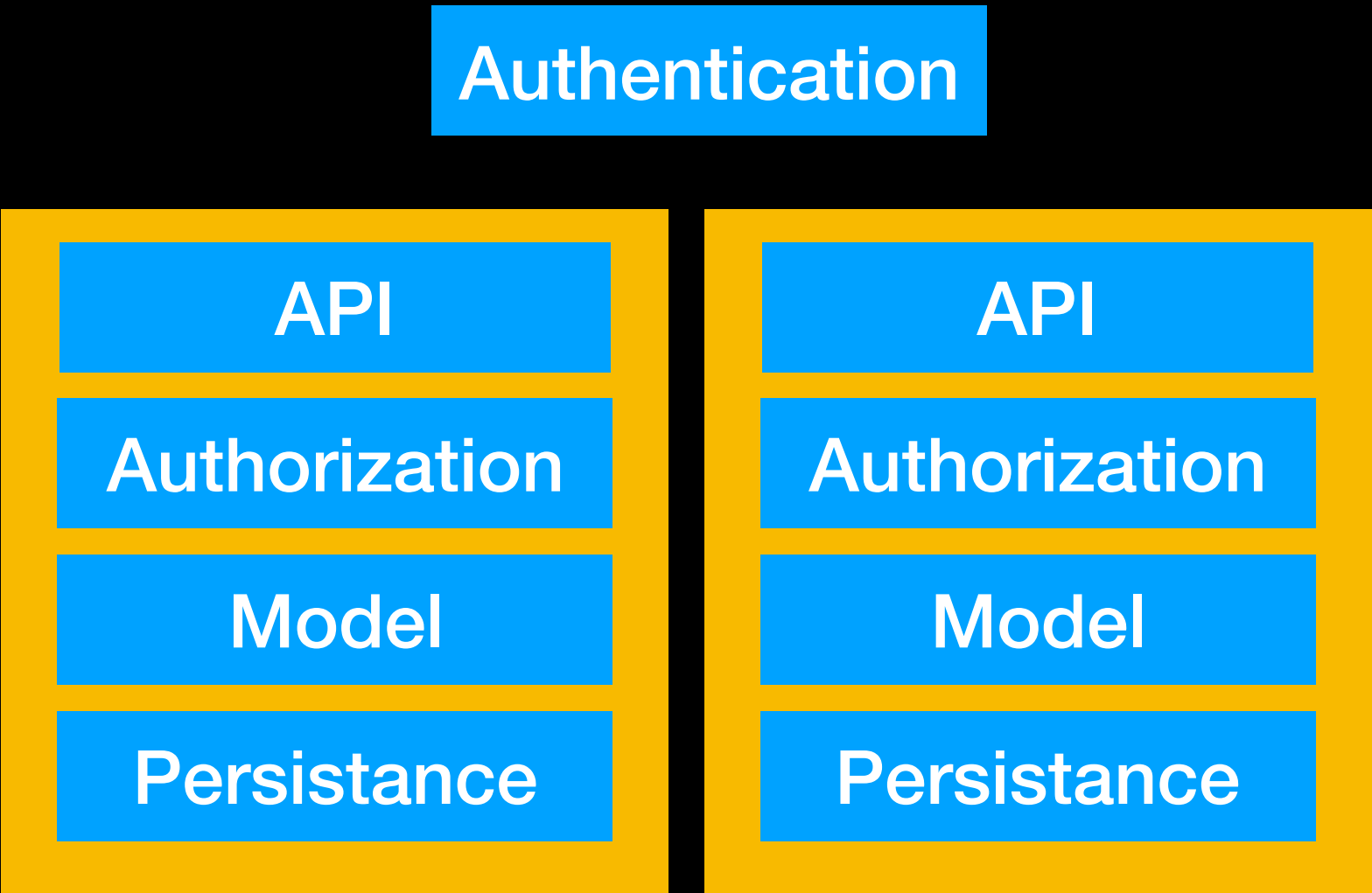
Model

Persistence

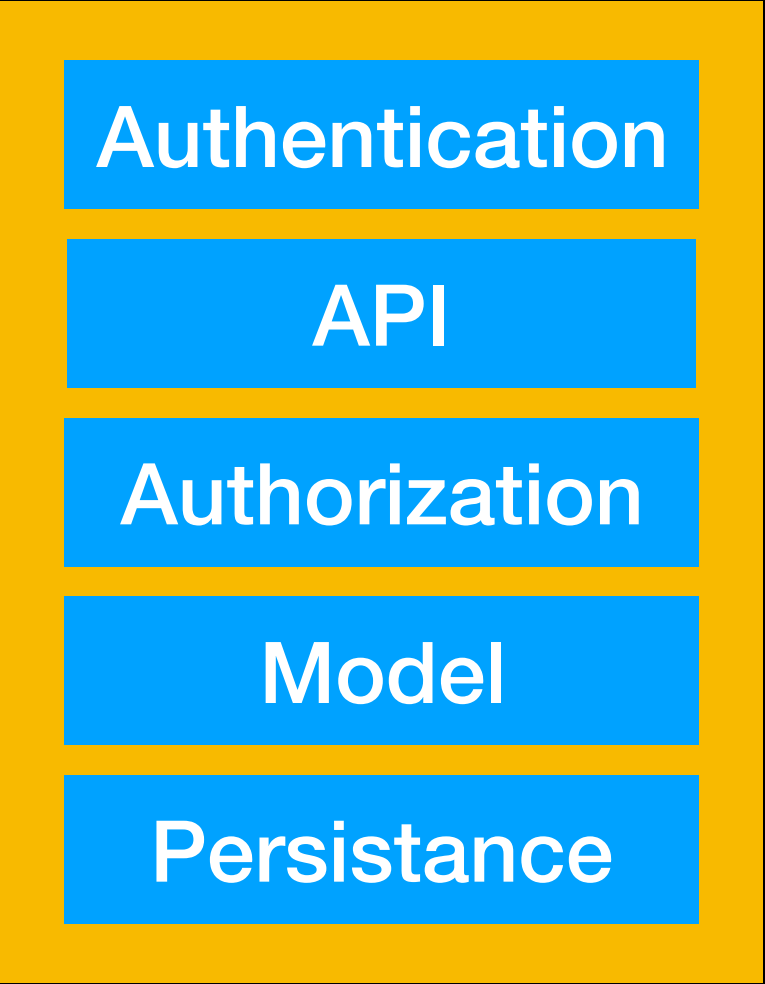
Monolith



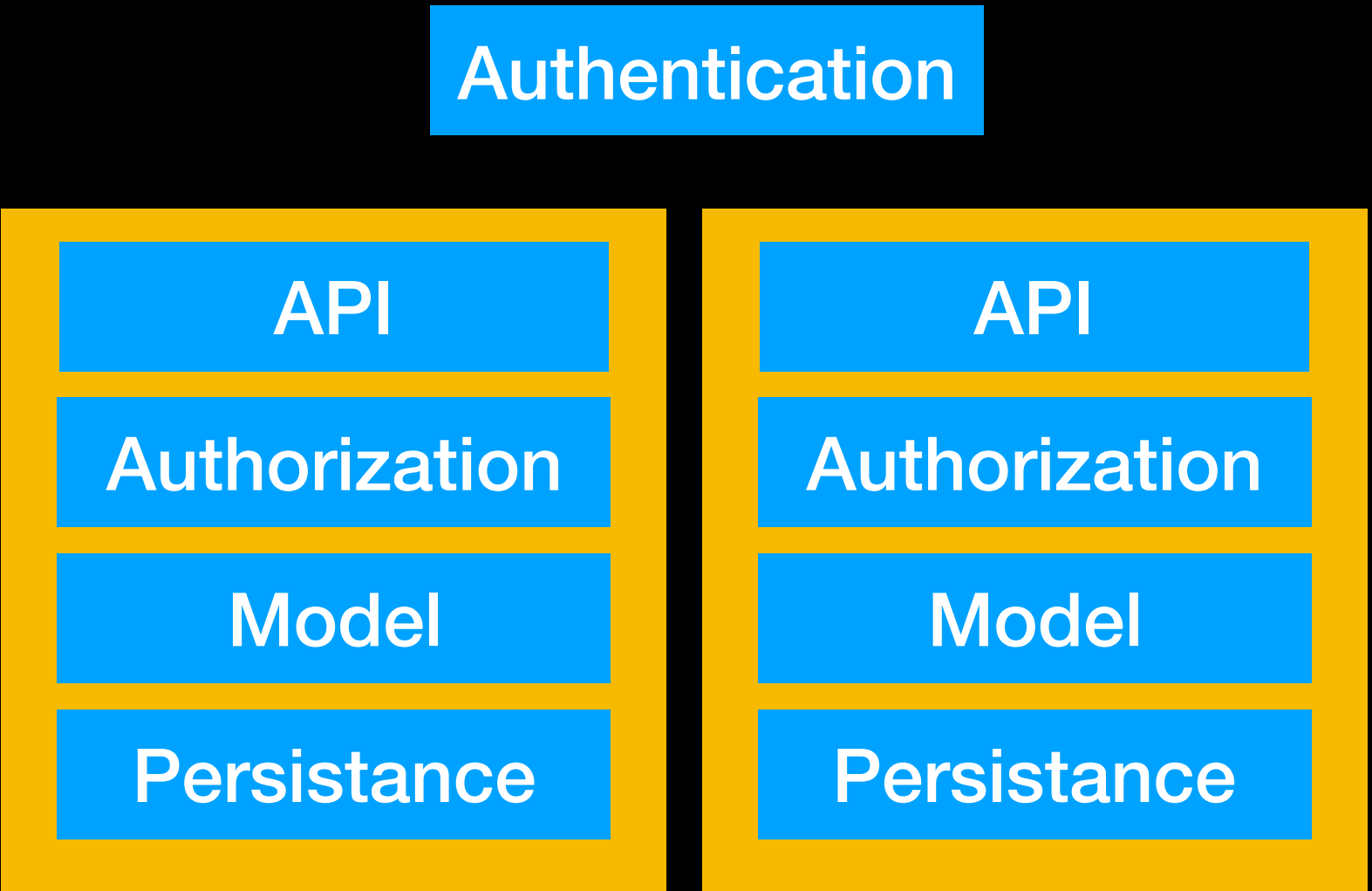
Microservice



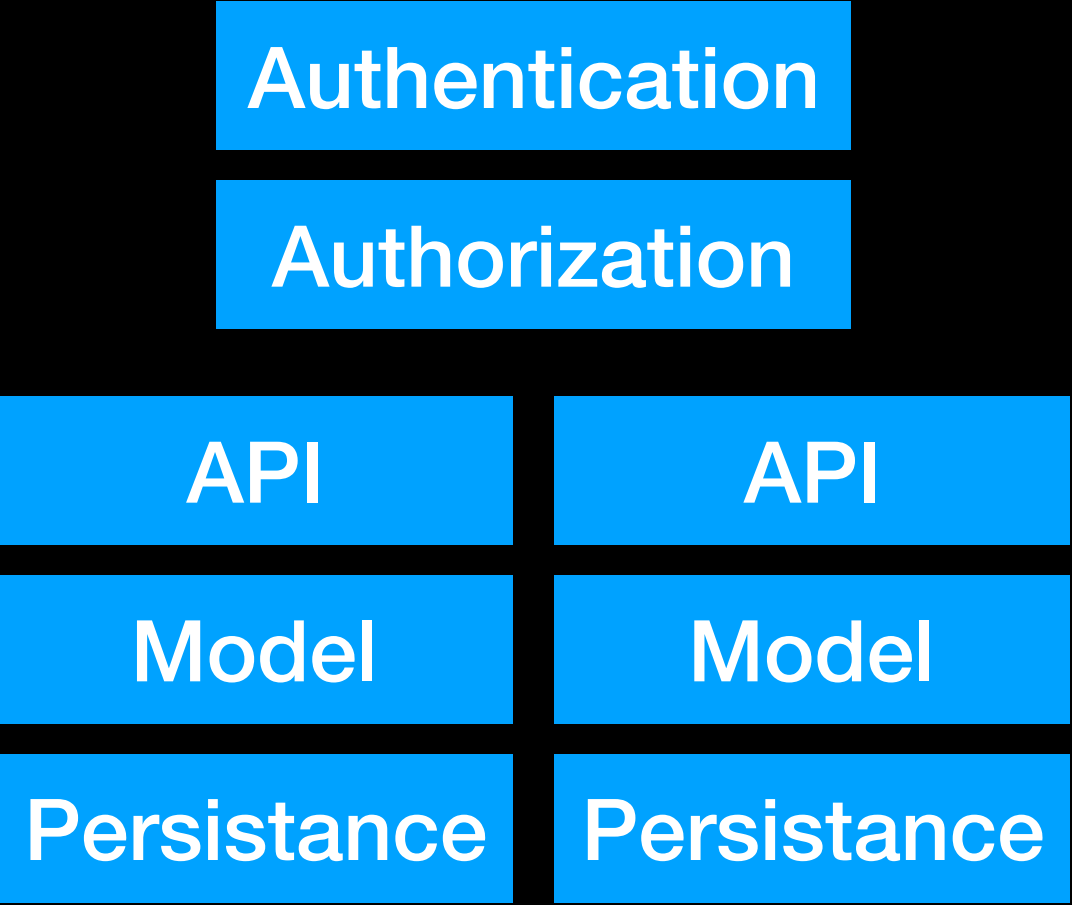
Monolith



Microservice



Serverless



API Gateway

Authentication

Authorization

DoS Protection

Rate Limiting

Microservice

Serverless

API

Authorization

Model

Persistence

API

Authorization

Model

Persistence

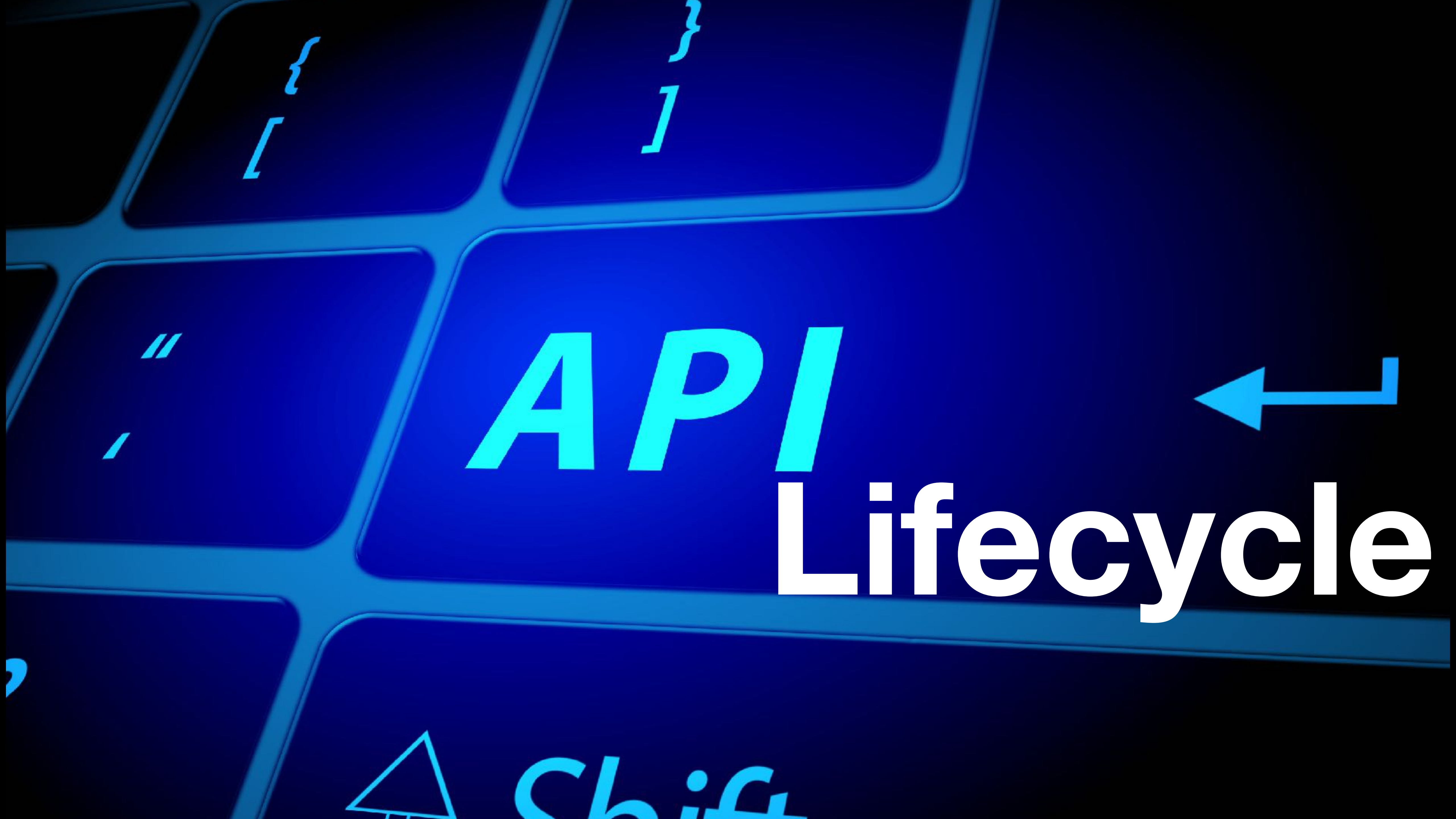
API

Model

Model

Persistence

Aspect	Monolith	Microservice	Serverless
Scaling	Complex	Simple	N/A
Endpoints	100-1000	10-100	1
Instances	10s	100s	1000s
Repositories	1	100s	1000s
Deployment	Infrequent	Continuous	Continuous
Developer Contention	Often	Infrequent	Rare
Asset Management	Simple	Difficult	Complex



API

Lifecycle

Experiment

Stub

Document

Experiment

Implement

Stub

Verify

Document

Document

Generate

Experiment

Implement

Realize

Stub

Verify

Define (Gateway)

Document

Document

Explore

Generate

Experiment

Implement

Realize

Maintain

Stub

Verify

Define (Gateway)

Evolve

Document

Document

Explore

Extend

Generate

The simplest way to design APIs

RESTful API Modeling Language (RAML) makes it easy to manage the whole API lifecycle from design to sharing. It's concise - you only write what you need to define - and reusable. It is machine readable API design that is actually human friendly.

Write once. Use many. Creative laziness encouraged.


[RAML 1.0 Specification](#)
[RAML 0.8 Specification](#)


- For every API, start by defining which version of RAML you are using, and then document basic characteristics of your API - the title, baseURI, and version.
- Create and pull in namespaces, reusable libraries containing data types, traits, resource types, schemas, examples, & more.
- Annotations let you add vendor specific functionality without compromising your spec
- Traits and resourceTypes let you take advantage of code reuse and design patterns
- Easily define resources and methods, then add as much detail as you want. Apply traits and other patterns, or add parameters and other details specific to each call.
- Describe expected responses for multiple media types and specify data types or call in pre-defined schemas and examples. Schemas and examples can be defined via a data type, in-line, or externalized with include.
- Write human-readable, markdown-formatted descriptions throughout your RAML spec, or include entire markdown documentation sections at the root.

```

1  #%RAML 1.0
2  title: World Music API
3  baseUri: http://example.api.com/{version}/
4  version: v1
5
6  uses:
7    Songs: libraries/songs.raml
8
9  annotationTypes:
10   monitoringInterval:
11     parameters:
12       value: integer
13
14  traits:
15   secured: !include secured/accessToken.raml
16
17  /songs:
18   is: [ secured ]
19   get:
20     (monitoringInterval): 30
21     queryParameters:
22       genre:
23         description: filter the songs by genre
24   post:
25   /{songId}:
26     get:
27       responses:
28         200:
29           body:
30             application/json:
31               type: Songs.Song
32             application/xml:
33               schema: !include schemas/songs.xml
34               example: !include examples/songs.xml

```

Songs Library

```

1  #%RAML 1.0 Library
2  types:
3    Song:
4     properties:
5       title: string
6       length: number
7    Album:
8     properties:
9       title: string
10      songs: Song[]
11    Musician:
12     properties:
13       name: string
14     discography: (Song | Album)[]

```

songs.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <root xmlns="http://www.example.com/2012/01/01/songs"
3  xmlns:base="http://example.com/2012/01/01/songs"
4  xmlns:secured="http://example.com/2012/01/01/songs"
5  >
6    <secured:filter genre="rock">
7      <secured:filter genre="rock">
8        <secured:filter genre="rock">
9          <secured:filter genre="rock">
10           <secured:filter genre="rock">
11             <secured:filter genre="rock">
12               <secured:filter genre="rock">
13                 <secured:filter genre="rock">
14                 <secured:filter genre="rock">
15               </secured:filter>
16             </secured:filter>
17           </secured:filter>
18         </secured:filter>
19       </secured:filter>
20     </secured:filter>
21   </root>

```

💬 There are many advantages to taking an API-first development approach. And with the help of tools like RAML, APIs have become more elastic and easier to use. 💬

Swagger

The Best APIs are Built with Swagger Tools



Design

Design and model APIs according to specification-based standards



Build

Build stable, reusable code for your API in almost any language



Document

Improve developer experience with interactive API documentation



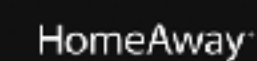
Test

Perform simple functional tests on your APIs without overhead



Standardize

Set and enforce API style guidelines across your API architecture








Swagger for Everyone

Swagger open source and pro tools have helped millions of API developers, teams, and organizations deliver great APIs.



Swagger Open Source

Get up and running with your API development with the Swagger open source tools



Swagger Inspector

Quickly test APIs and generate API definitions from code — great for teams of 1 or 100+



SwaggerHub

Work better together in the leading API development platform for teams & enterprises

Swagger and OAS Go Hand-in-Hand

Swagger offers the most powerful and easiest to use tools to take full advantage of the OpenAPI Specification.

[Learn More](#)


```
1 swagger: "2.0"
2 info:
3   description: "This is a sample server Petstore server. You can find out more about Swagger at [http://swagger.io](http://swagger.io) or on [irc.freenode.net, #swagger](http://swagger.io/irc/). For this sample, you can use the api key `special-key` to test the authorization filters."
4   version: "1.0.0"
5   title: "Swagger Petstore"
6   termsOfService: "http://swagger.io/terms/"
7   contact:
8     email: "apiteam@swagger.io"
9   license:
10    name: "Apache 2.0"
11    url: "http://www.apache.org/licenses/LICENSE-2.0.html"
12 host: "petstore.swagger.io"
13 basePath: "/v2"
14 tags:
15 - name: "pet"
16   description: "Everything about your Pets"
17   externalDocs:
18     description: "Find out more"
19     url: "http://swagger.io"
20 - name: "store"
21   description: "Access to Petstore orders"
22 - name: "user"
23   description: "Operations about user"
24   externalDocs:
25     description: "Find out more about our store"
26     url: "http://swagger.io"
27 schemes:
28 - "http"
29 paths:
30 - /pet:
31   post:
32     tags:
33     - "pet"
34     summary: "Add a new pet to the store"
35     description: ""
36     operationId: "addPet"
37     consumes:
38     - "application/json"
39     - "application/xml"
40     produces:
41     - "application/xml"
42     - "application/json"
43   parameters:
44   - in: "body"
45     name: "body"
46     description: "Pet object that needs to be added to the store"
47     required: true
48     schema:
49       $ref: "#/definitions/Pet"
50   responses:
51     405:
52       description: "Invalid input"
53   security:
54   - petstore_auth:
55     - "write:pets"
56     - "read:pets"
57 - /pet:
58   put:
59     tags:
60     - "pet"
61     summary: "Update an existing pet"
62     description: ""
63     operationId: "updatePet"
64     consumes:
65     - "application/json"
66     - "application/xml"
67     produces:
68     - "application/xml"
69     - "application/json"
69     parameters:
70     - in: "body"
71       name: "body"
72       description: "Pet object that needs to be added to the store"
```

Swagger Petstore

1.0.0

[Base URL: petstore.swagger.io/v2]

This is a sample server Petstore server. You can find out more about Swagger at <http://swagger.io> or on [#swagger](http://irc.freenode.net). For this sample, you can use the api key **special-key** to test the authorization filters.

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Schemes

HTTP

Authorize



pet Everything about your Pets

Find out more: <http://swagger.io>



POST

/pet Add a new pet to the store



Parameters

Try it out

Name

Description

body required

Pet object that needs to be added to the store

(body)

Example Value Model

```
{
  "id": 0,
  "category": {
    "id": 0,
    "name": "string"
  },
  "name": "doggie",
  "photoUrls": [
    "string"
  ],
  "tags": [
    {
      "id": 0,
      "name": "string"
    }
  ],
  "status": "available"
}
```

Parameter content type

application/json

Responses

Response content type

application/xml

Code

Description

405

Invalid input

Experiment

Implement

Realize

Stub

Verify

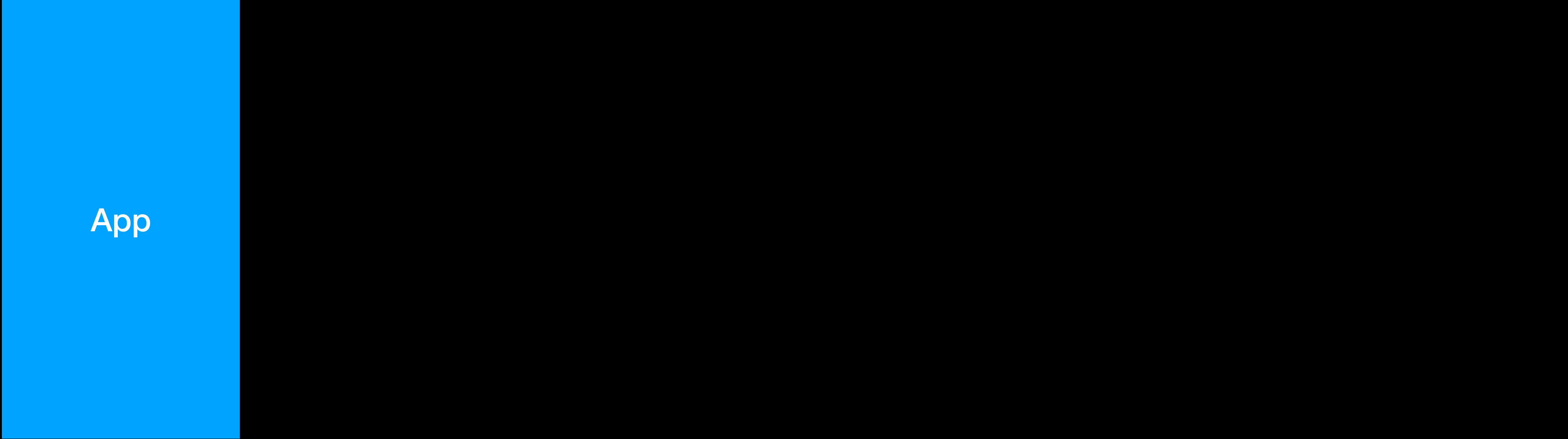
Define (Gateway)

Document

Document

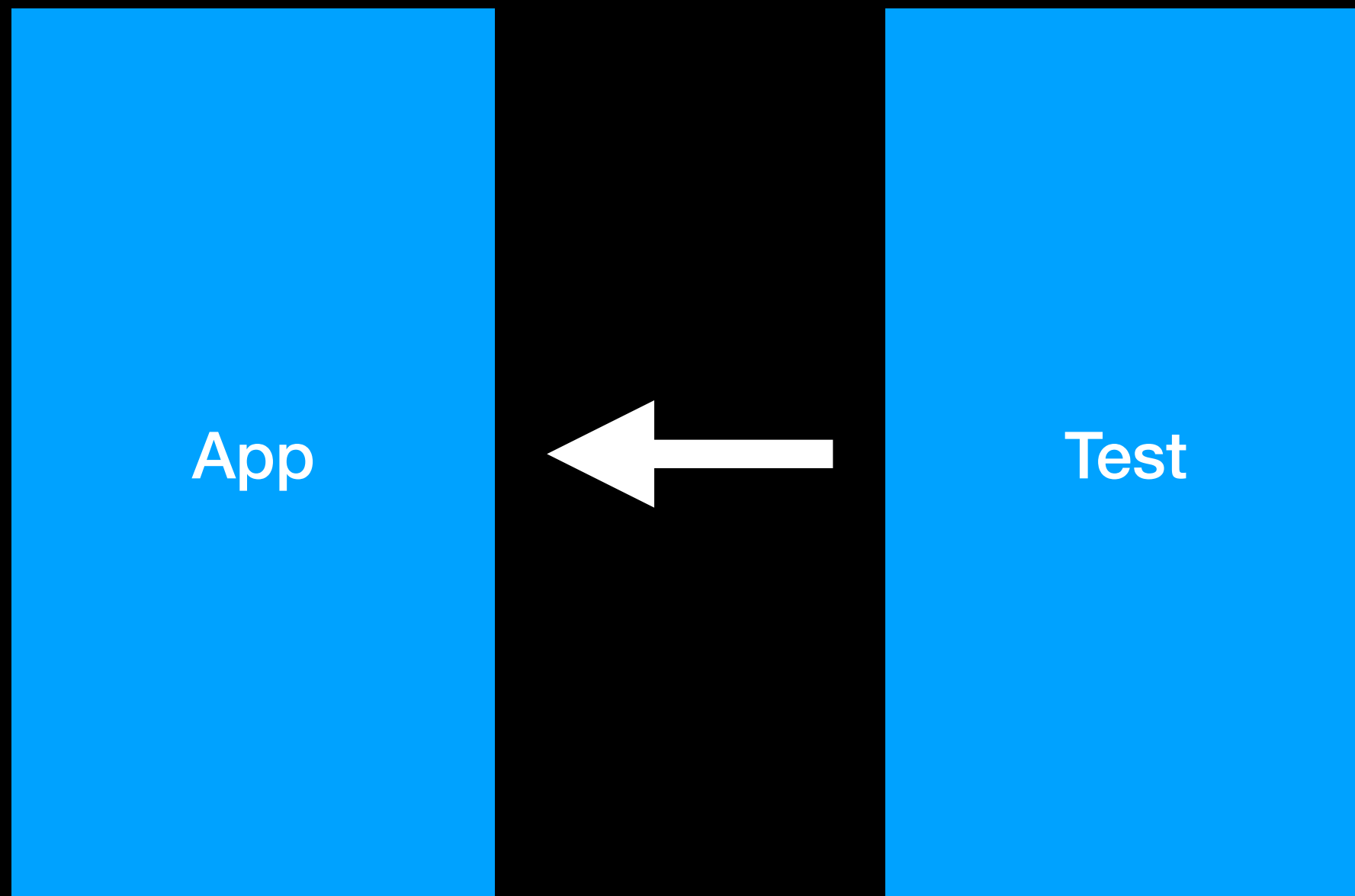
Generate

Spring REST Docs

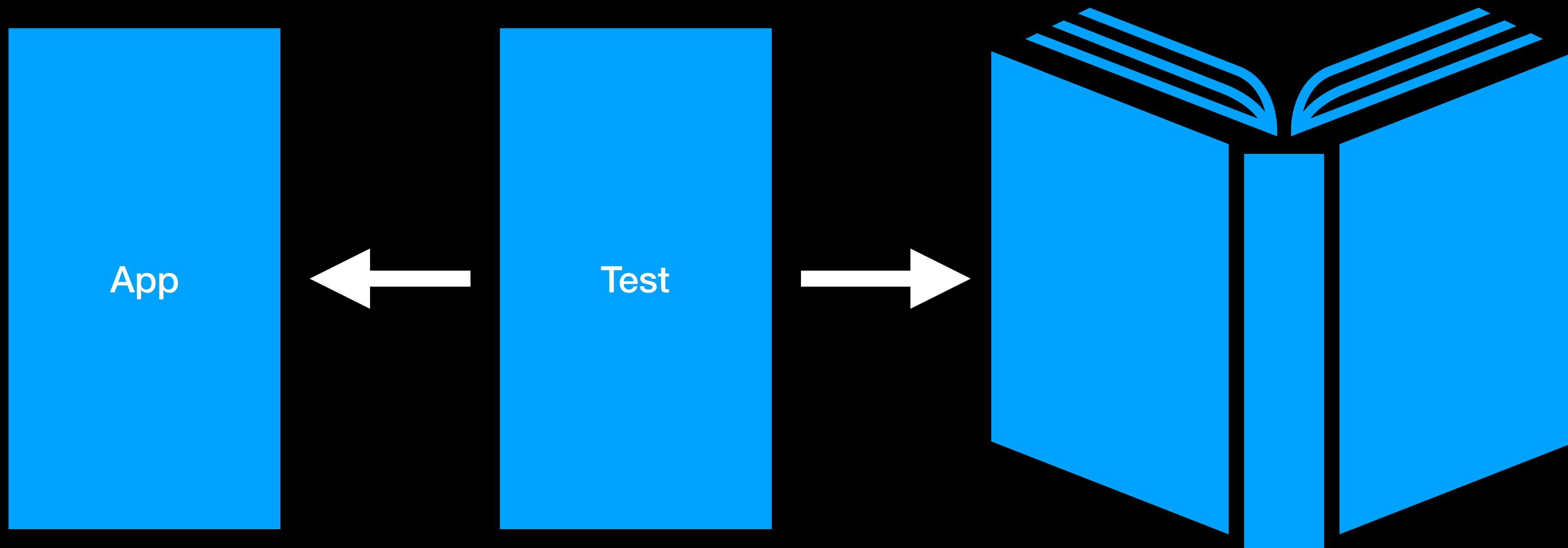


App

Spring REST Docs



Spring REST Docs



Text Based

High Quality Documentation

Executable

Excessive Coupling

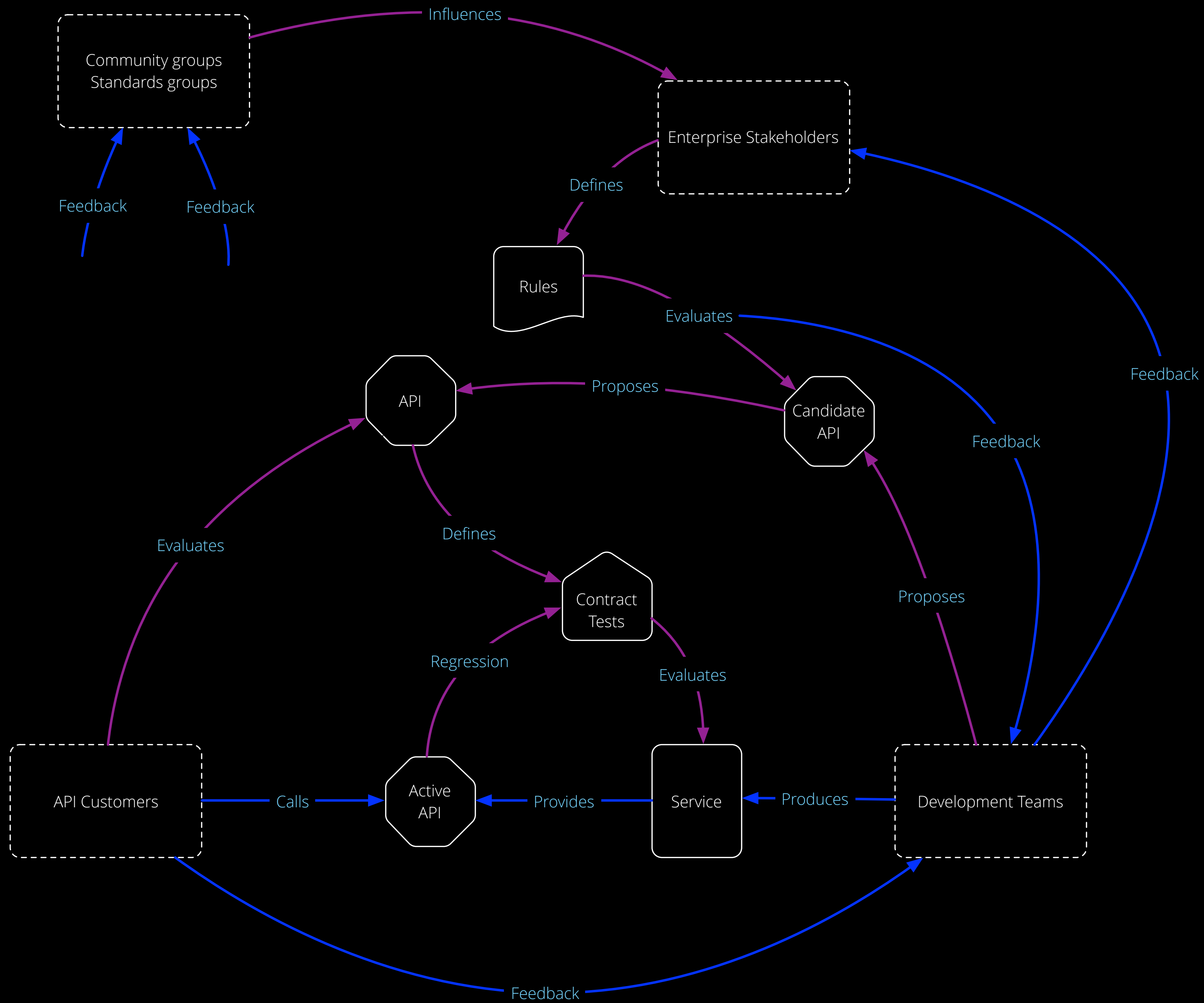
Versionable



Consumer Driven Contracts

Consumer Contracts

- Test - Executable
- Written by consuming development teams
- Specific to content
- Most useful for internal APIs
- Regression test for producer
 - Blocks promotion
- Encourages and supports environment availability



Spec and Checks

- API Specifications
 - Experimentation
 - Verification & Regression
 - Versioned
 - Code Generation
- Contracts
 - Internal
 - Primary verification
 - Evolve

Documentation



Text based

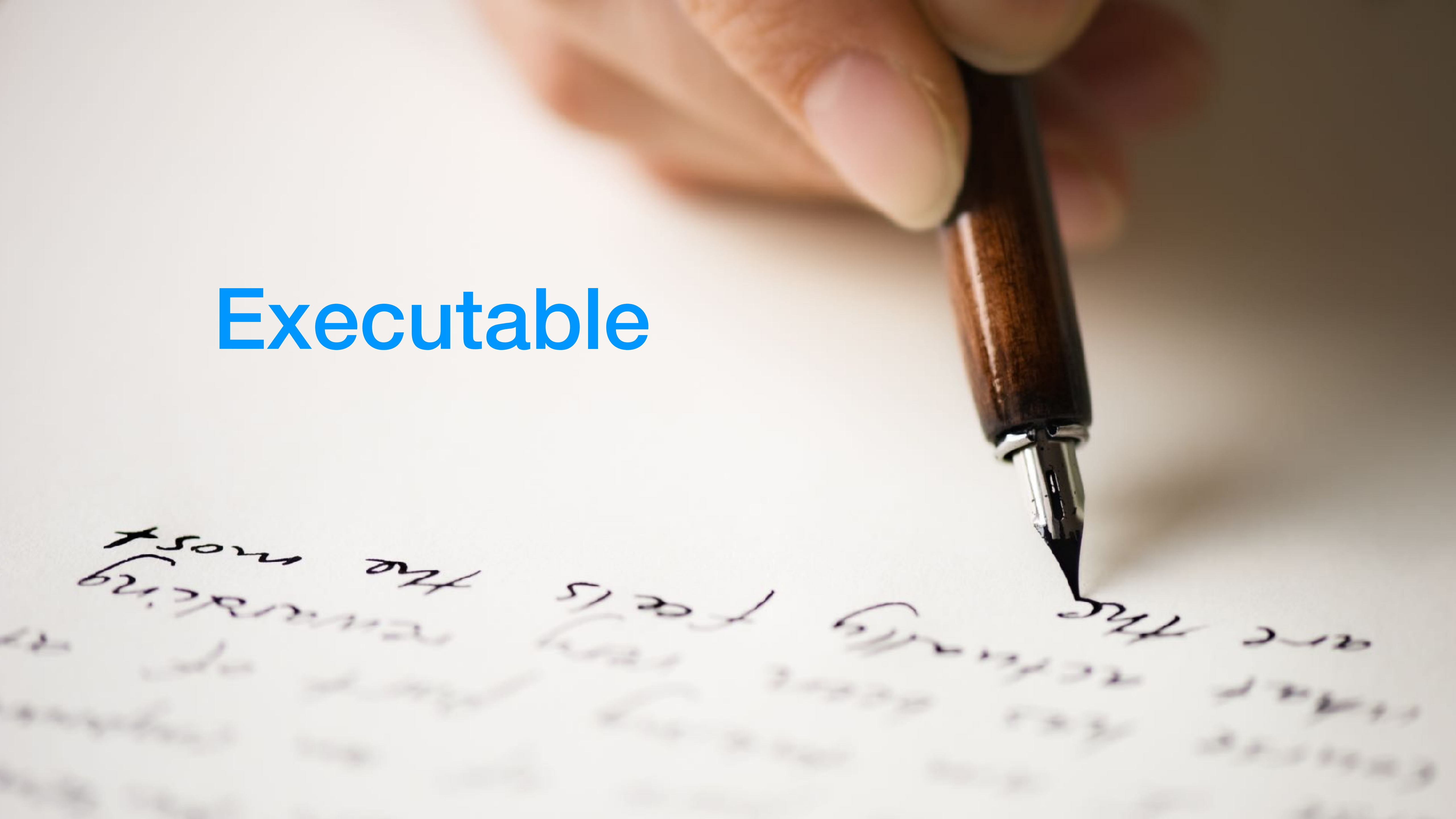


Diagramming

are the
the
most

A close-up photograph of a hand holding a dark wooden fountain pen, poised to write on a piece of paper. The paper already has some handwritten text in cursive, which is slightly out of focus. Overlaid on the left side of the image is the word "Versioned" in a large, bold, blue sans-serif font. The background is a soft, out-of-focus light beige color.

Versioned

A close-up photograph of a hand holding a dark wooden fountain pen, poised to write on a piece of white paper. The paper already has some cursive handwriting on it, which is slightly out of focus. The word "Executable" is overlaid in a large, bold, blue sans-serif font on the left side of the image. The background is a soft, out-of-focus light brown or beige color.

Executable

JavaDoc and friends

```
// import statements
```

```
/**
```

```
 * @author      Firstname Lastname <address @ example.com>
```

```
 * @version     1.6                      (current version number of program)
```

```
 * @since       1.2                      (the version of the package this class was first added to)
```

```
 */
```

```
public class Test {
```

```
    // class body
```

```
}
```

HttpClientBuilder	disableContentCompression (boolean disableContentCompression) Disable support of decompression of responses
HttpClientBuilder	name (String environmentName) Use the given environment name.
HttpClientBuilder	using (org.apache.http.client.CredentialsProvider credentialsProvider) Use the given CredentialsProvider instance.
HttpClientBuilder	using (org.apache.http.conn.DnsResolver resolver) Use the given DnsResolver instance.

What

How

~~What~~

How in 30 minutes or less

~~What~~

Alternatives

are the
most
the
facts
are the
most
the
facts

asciidoc

Running as a developer should be very simple and quick. Long manual installations are error prone and a waste of everyone's time. If you find problems with the developer setup please raise an issue so we can address it.

Currently the installation scripts have only been tested on OSX and rely on [homebrew](https://brew.sh) (<https://brew.sh>).

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) ([Brewfile](#)) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Sometimes elasticsearch may not have started. If you have used homebrew to get things rolling.

```
brew services elasticsearch restart
```

to get things rolling.

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Running as a developer should be very simple and quick. Long manual installations are error prone and a waste of everyone's time. If you find problems with the developer setup please raise an issue so we can address it.

Currently the installation scripts have only been tested on OS X and rely on [homebrew](https://brew.sh) (<https://brew.sh>).

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) ([Brewfile](#)) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Sometimes elasticsearch may not have started. If you have used brew services elasticsearch restart

```
brew services elasticsearch restart
```

to get things rolling.

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Running as a developer should be very simple and quick. Long manual installations are error prone and a waste of everyone's time. If you find problems with the developer setup please raise an issue so we can address it.

Currently the installation scripts have only been tested on OS X and rely on [homebrew](https://brew.sh) (<https://brew.sh>).

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) ([Brewfile](#)) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Sometimes elasticsearch may not have started. If you have used homebrew to get things rolling.

```
brew services elasticsearch restart
```

to get things rolling.

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Running as a developer should be very simple and quick. Long manual installations are error prone and a waste of everyone's time. If you find problems with the developer setup please raise an issue so we can address it.

Currently the installation scripts have only been tested on OS X and rely on [homebrew](https://brew.sh) (<https://brew.sh>).

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) ([Brewfile](#)) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

Once everything is setup use the build script to build all the components:

```
./build
```

Sometimes elasticsearch may not have started. If you have used

```
brew services elasticsearch restart
```

to get things rolling.

Prerequisites

Kuona runs on the Java VM. You will need to install Java 8 JDK to build and run the kuona tools.

Check out the [Brewfile](#) for a full list of dependencies.

OS X/macOS Setup

```
git clone https://github.com/kuona/kuona-project.git
cd kuona-project
sh dev-setup
```

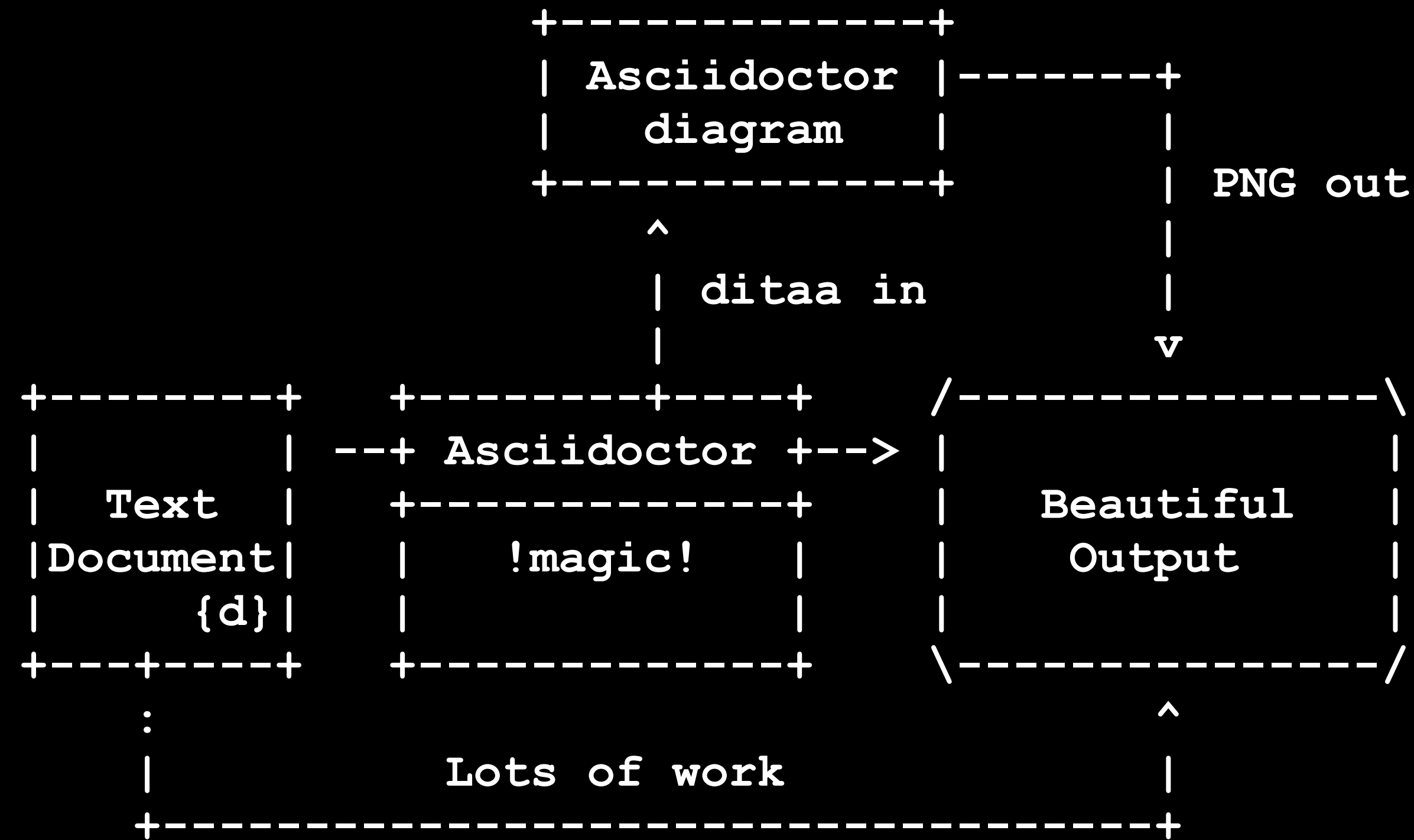
Once everything is setup use the build script to build all the components:

```
./build
```

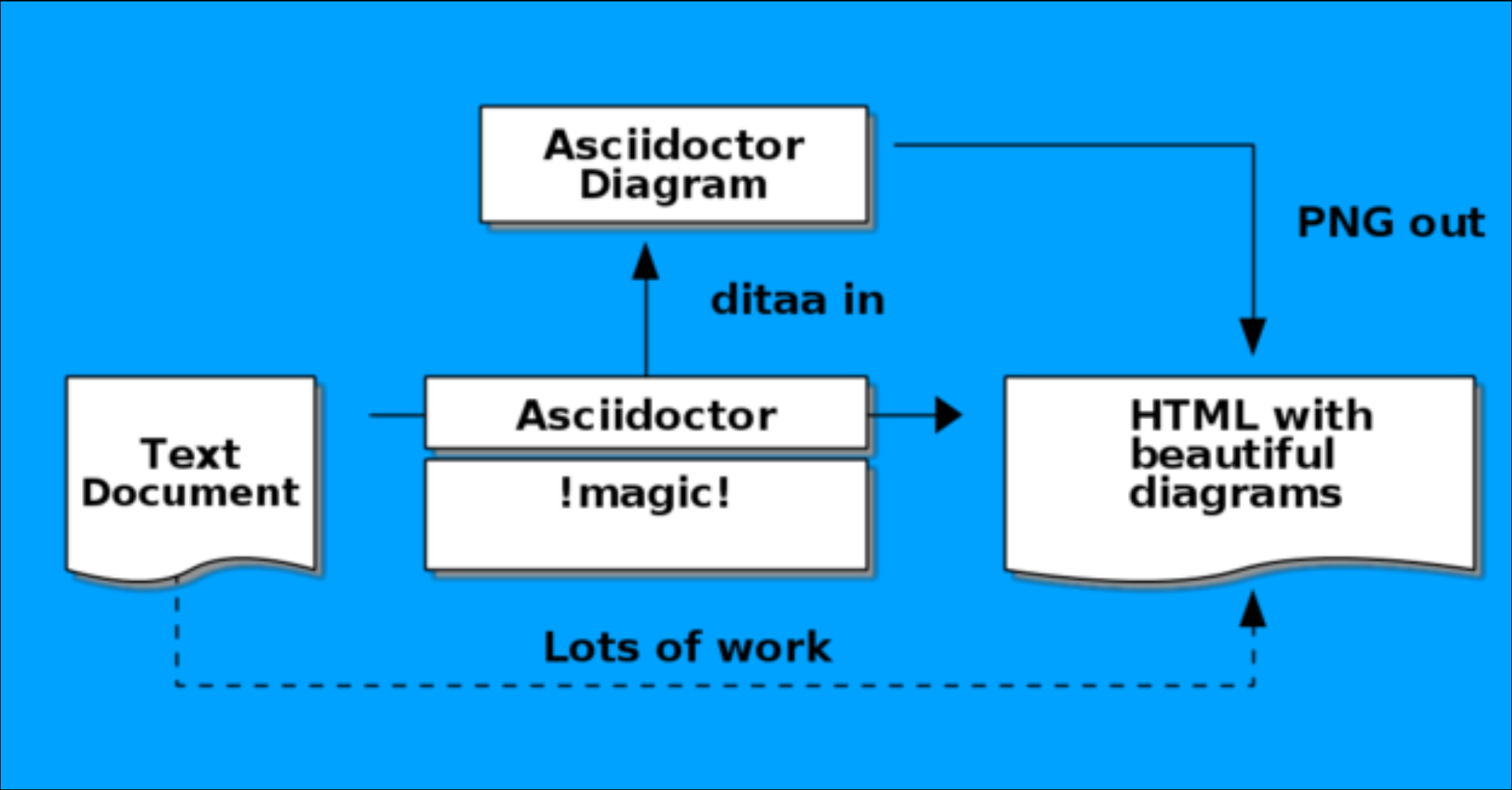
```
asciidoc <- asciidoctor + asciidoctor diagram
```

[ditaa]

....



....



PlantUML Diagram Syntax

```
[plantuml, diagram-classes, png]
```

1 2 3

```
....
```

```
class BlockProcessor
```

```
class DiagramBlock
```

```
class Ditaablock
```

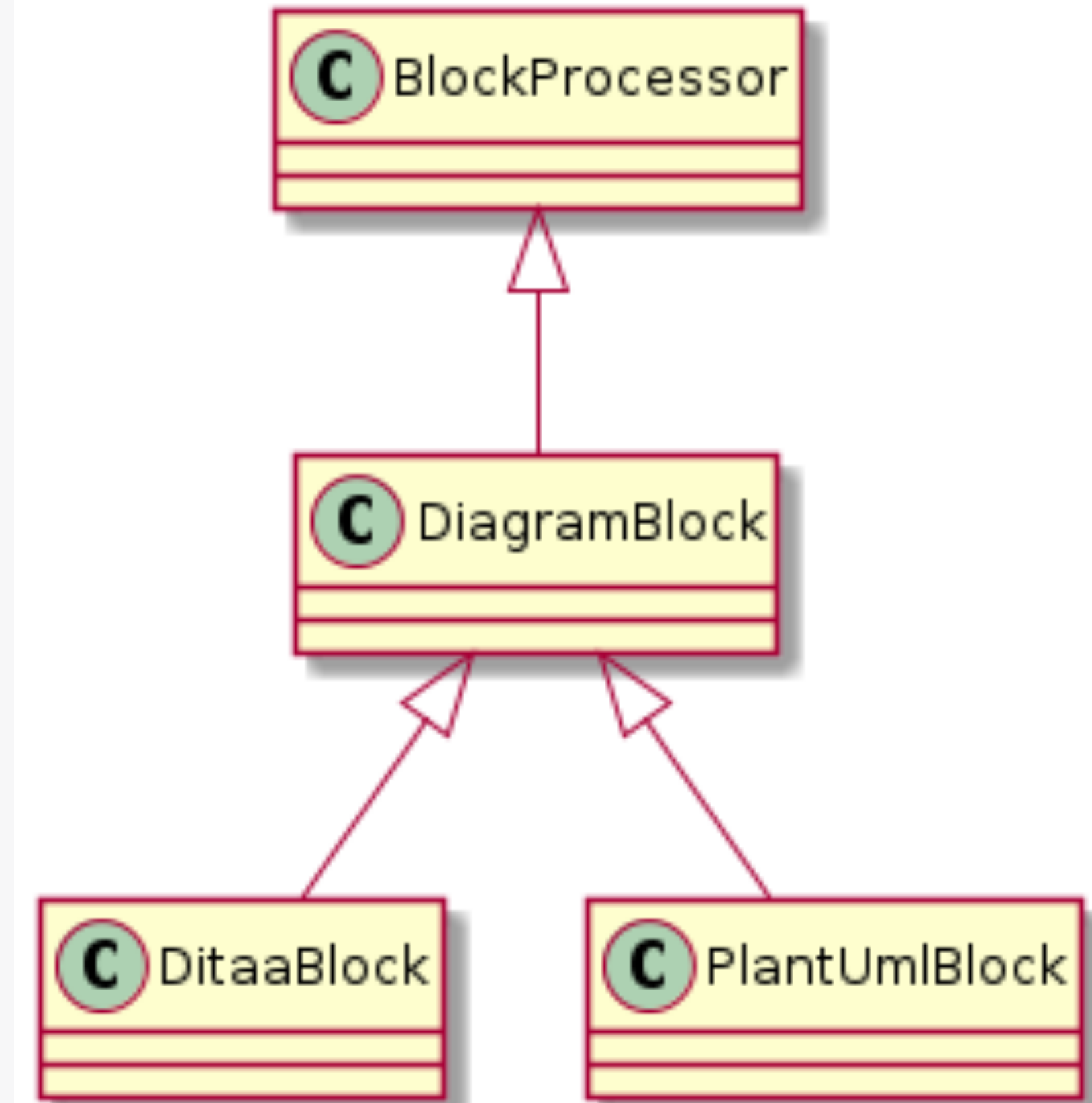
```
class PlantUmlBlock
```

```
BlockProcessor <|-- DiagramBlock
```

```
DiagramBlock <|-- Ditaablock
```

```
DiagramBlock <|-- PlantUmlBlock
```

```
....
```

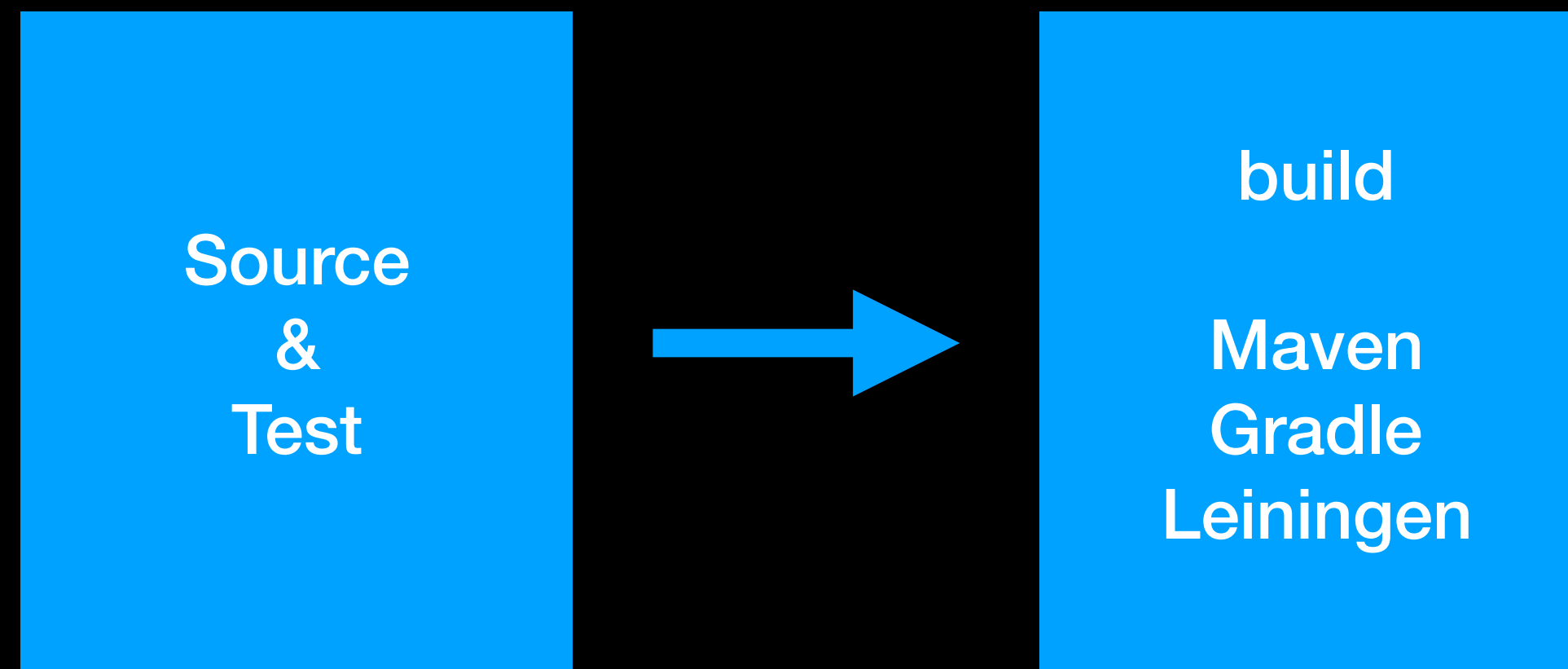


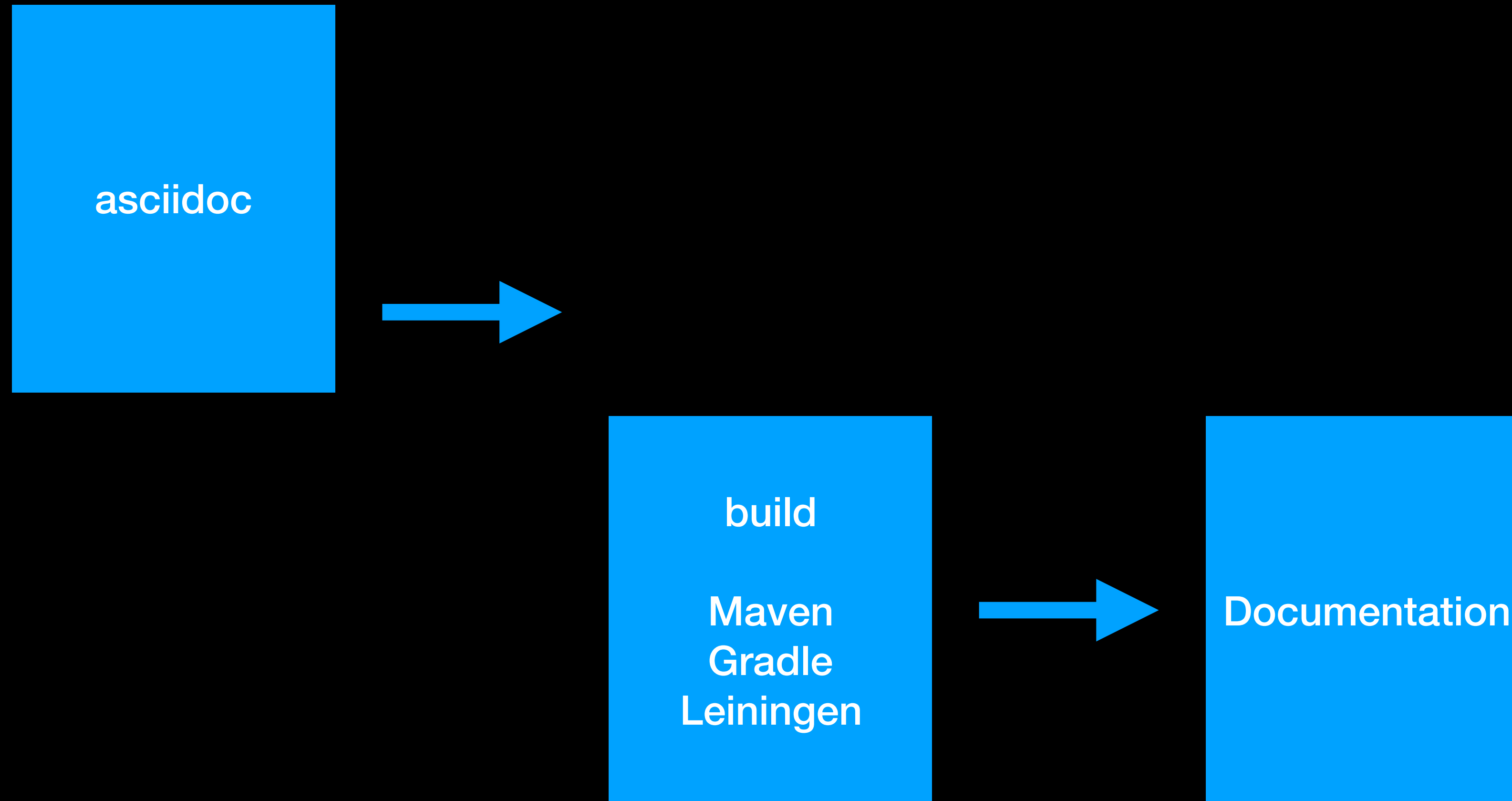
- 1 The diagram is written in PlantUML so the first positional attribute is assigned the `plantuml` diagram type.
- 2 The name of the generated diagram file (target) is written in the second positional

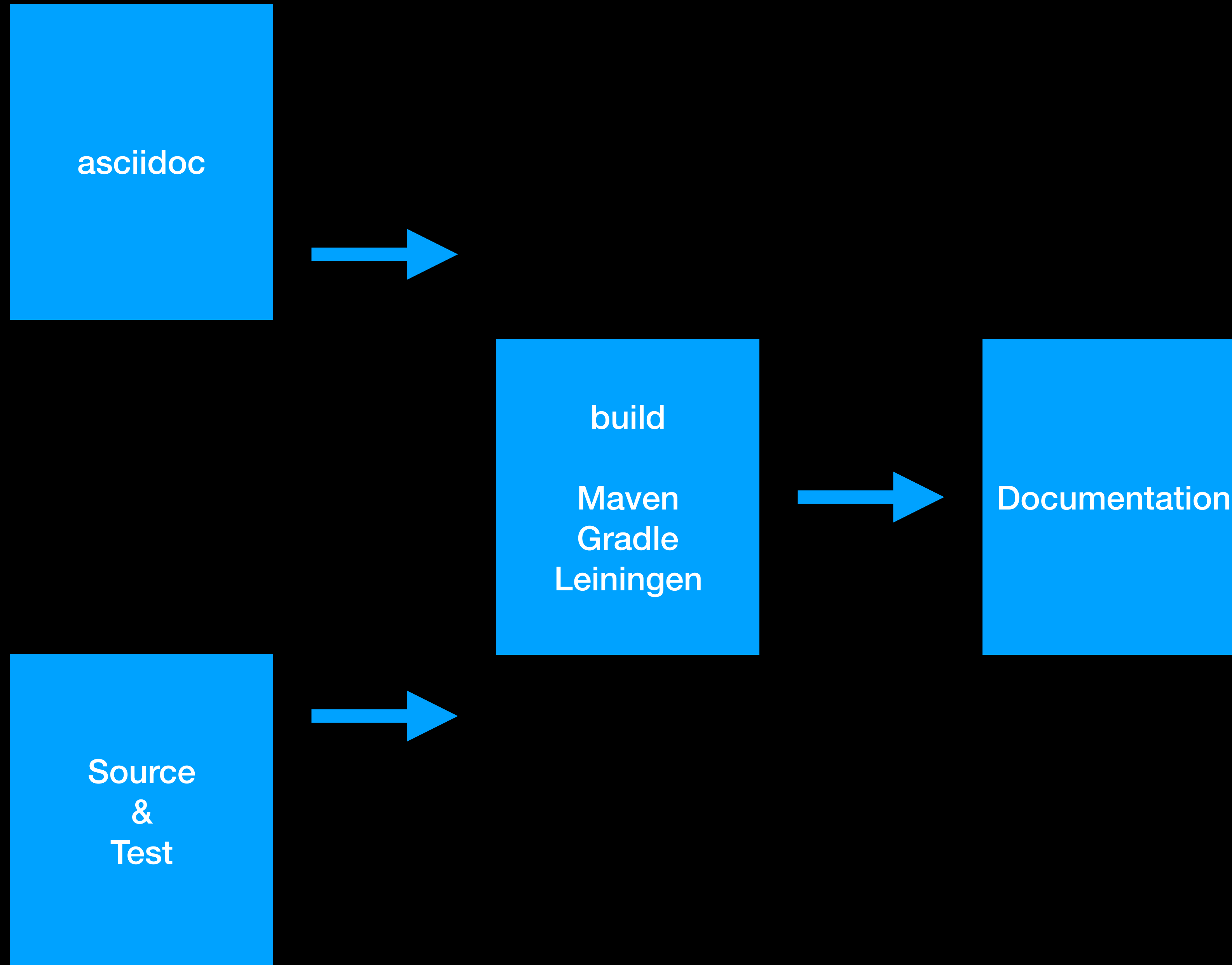
- Text Based ✓
- Diagramming ✓
- Versioned ✓
- Executable ?

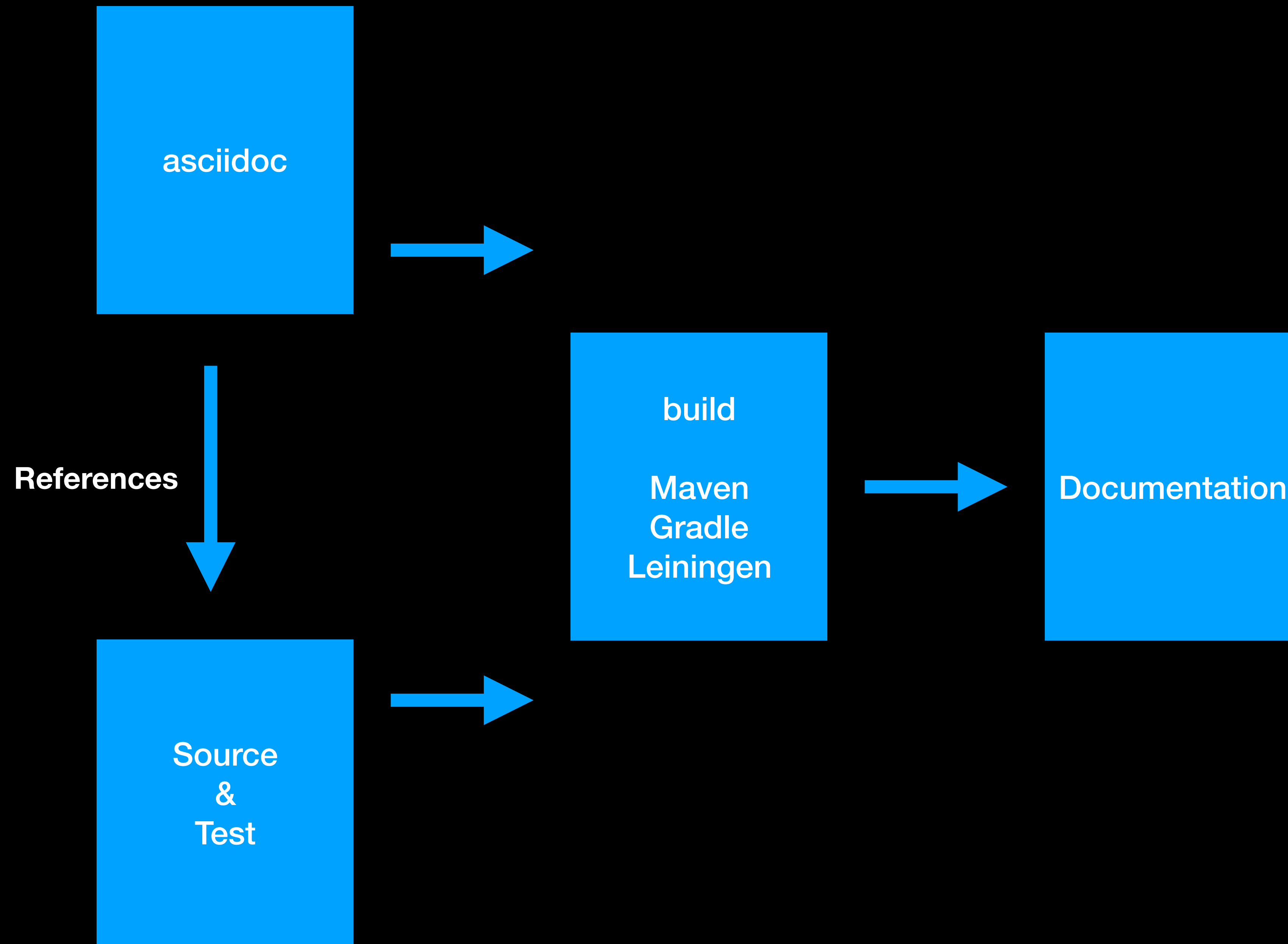
Executable

are the
most
the
making
facts
are the
most
the
making
facts





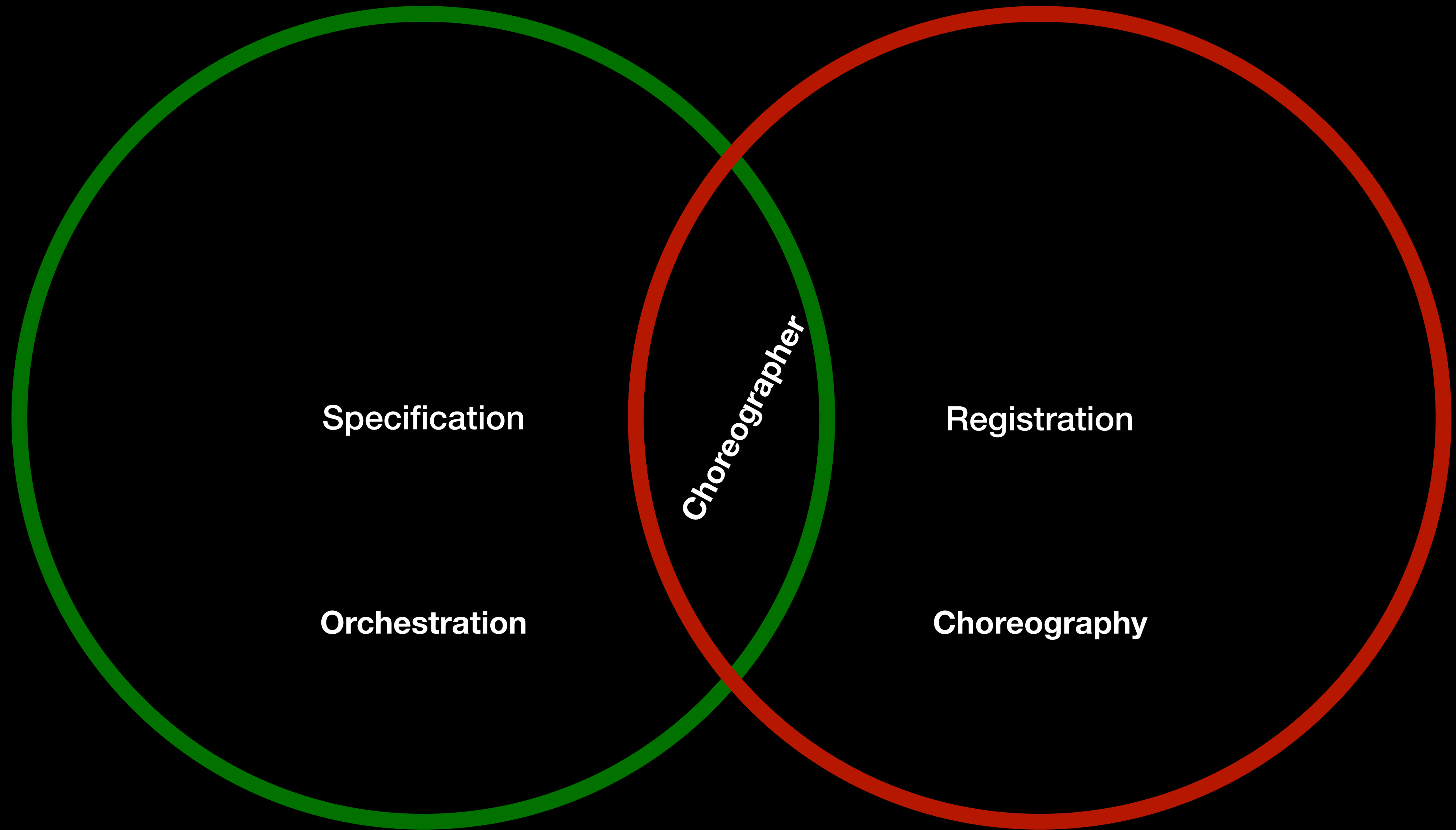




Demo

Discovery





Specification

Orchestration

Choreographer

Registration

Choreography

Version Control

Monorepo

Pipeline

Pipeline

Runtime

Environment

Deployable

App

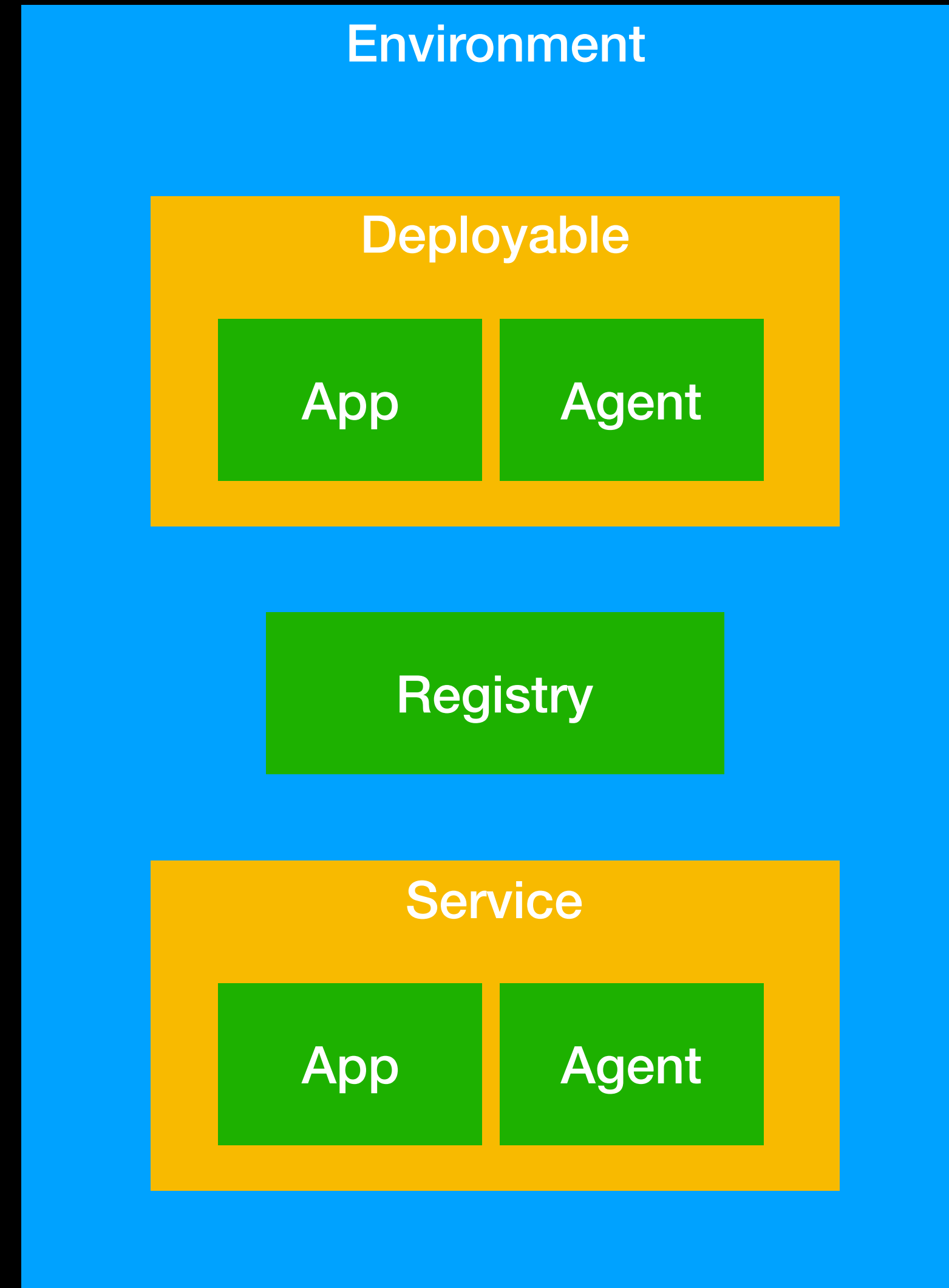
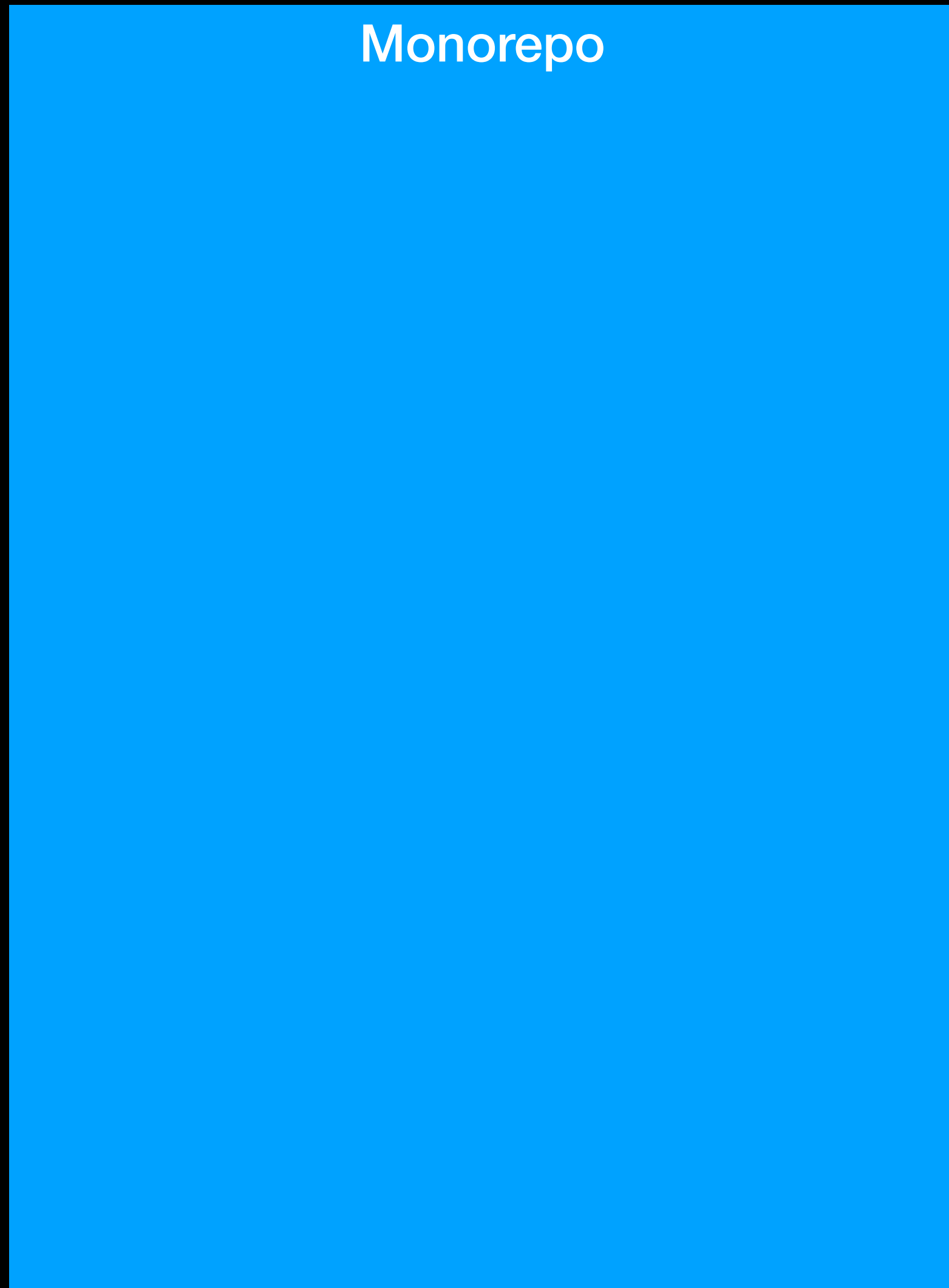
Agent

Registry

Service

App

Agent



Version Control

Repo

Pipeline

Repo

Pipeline

Runtime

Environment

Deployable

App

Agent

Registry

Service

App

Agent

```
— CHANGELOG.md
— LICENSE
— README.md
— classes
— dependency.dot
— dev-resources
  — log4j.properties
— doc
  — core.adoc
  — core.html
  — intro.md
— kuona-core.iml
— pom.xml
— project.clj
— src
  — kuona_core
— target
— test
  — kuona_core
  — pom.xml
  — test-manifest.yaml
— unit.yml
```

find

grep

???

```
projects % tree -L 1 .
```

```
.
— OSX\ playground.playground
— Restweb
— angular
— api-gateway-sample
— bcus
— clojure
— compojure
— cpprestsdk
— el-dorado-ui
— event-drive-enterprise
— grahambrooks.com
— insperna-nginx.conf
— islands_engine
— java-nio-server
— java-nio-socket-server
— kuona-project
— kuona-project.local
— kuona-project.original
— kuona-workspace
```

Successful Recipes

Site Search

F E S S

admin

codelibs

Q

Sort Score Result 20 results Languages All

Results 1 - 20 of 26 for codelibs (0.02 sec)

codelibs/corelib · GitHub

Features Enterprise Pricing Watch 6 Star 2 Fork 0 codelibs/corelib Code Issues 0 Pull requests 0 Pulse Graphs...information. src .gitignore README.md pom.xml README.md codelibs-core Core library fo...
github.com/codelibs/corelib
Registered: 2015-12-04 01:16 - 32.2K bytes - Viewed (0)

codelibs/solrlib · GitHub

Features Enterprise Pricing Watch 6 Star 2 Fork 0 codelibs/solrlib Code Issues 1 Pull requests 0 Pulse Graphs
github.com/codelibs/solrlib
Registered: 2015-12-04 01:16 - 34.4K bytes - Viewed (0)

development.txt

<https://github.com/codelibs> Developer Guide <dev/getting-started> JavaDocs
<http://fess.codelibs.org/apidocs/index...org/apidocs/index.html> Issues
<https://github.com/codelibs/fess/issues?state=...
fess.codelibs.org/ sources/development.txt

DATE RANGE

Past 24 Hours6

Past Week26

Past Month6

Past Year26

SIZE

- 10kb9

10kb - 100kb16

100kb - 500kb1

Site Search

F E S S

admin

codellibs

Search

Sort **Score** Result **20 results** Languages **14**

Results 1 - 20 of 26 for codellibs (0.02 sec)

codellibs/corelib - GitHub

Features Enterprise Pricing Watch 6 Star 2 Fork 0 codellibs/corelib Code Issues 0 Pull requests 0 Pulse Graphs...information, src .gitignore README.md pom.xml README.md codellibs-core Core library fo...
github.com/codellibs/corelib
Registered: 2015-12-04 01:16 - 32.2K bytes - Viewed (0)

codellibs/solrib - GitHub

Features Enterprise Pricing Watch 6 Star 2 Fork 0 codellibs/solrib Code Issues 1 Pull requests 0 Pulse Graphs
github.com/codellibs/solrib
Registered: 2015-12-04 01:16 - 34.4K bytes - Viewed (0)

development.txt

<https://github.com/codellibs> Developer Guide <dev/getting-started> JavaDocs
<http://ress.codellibs.org/apidocs/index...org/apidocs/index.html> Issues
<https://github.com/codellibs/loss/issues?state=...
ress.codellibs.com ->https://development.txt

DATE RANGE

Past 24 Hours 1

Past Week 20

Past Month 6

Past Year 25

SIZE

- 10kb 1

10kb - 100kb 15

100kb - 500kb 5

Code Search

searchcode server

localhost:8080

searchcode

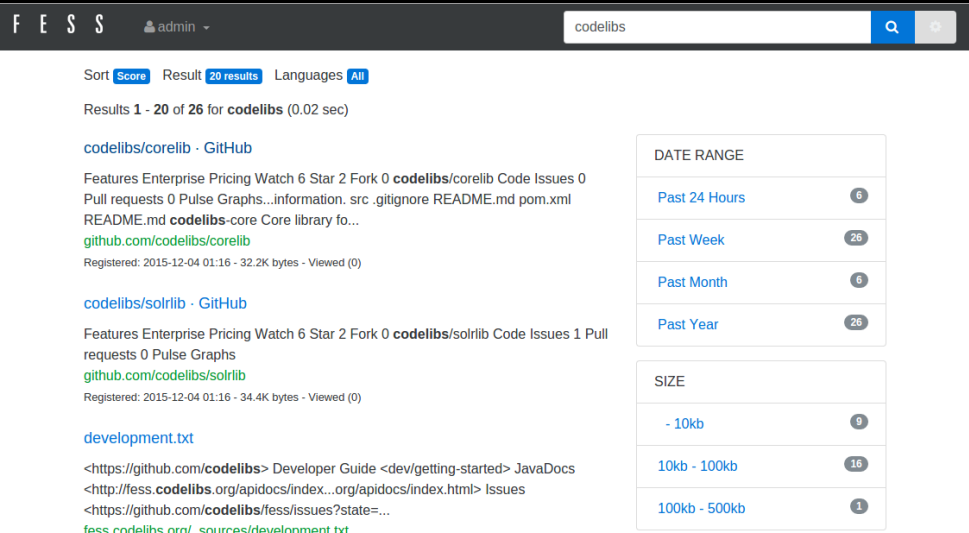
Repositories Documentation Admin

search

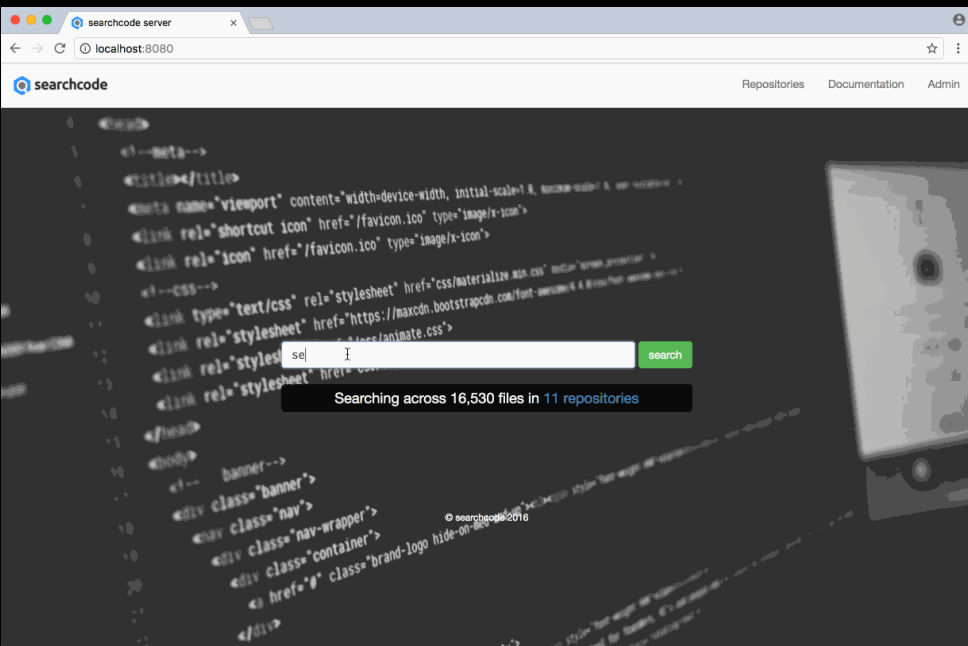
Searching across 16,530 files in 11 repositories

© searchcode 2016

Site Search



Code Search



Discoverable

Thank You

Graham Brooks

ThoughtWorks®



 [@grahamcbrooks](https://twitter.com/grahamcbrooks)

 graham@grahambrooks.com

 gbrooks@thoughtworks.com

 grahambrooks.com/talks



Please

**Remember to
rate this session**

Thank you!