AZURE DAY

Better API governance with Azure API Center

Massimo Crippa

Thanks to

Microsoft

avanade

eng

software one

4wardPRO
AN IMPRESOFT COMPANY

BCSoft
consulenza sistemi informativi

LOBRA

UNIKEY
Bringing IT knowledge to the people

Packt

JET BRAINS

NET CODE

Massimo Crippa

Lead Architect at Codit

Help customers throughout their app innovation journey, from envisioning to the cloud solution architecture.

https://massimocrippa.com/blog

https://www.linkedin.com/in/massimocrippa

https://github.com/massimoc

MVP Microsoft® Most Valuable Professional

Building an API program is complex

IDEA conception

API Strategy and Design

Security

Scalability & Performance

Documentation

Testing & QA

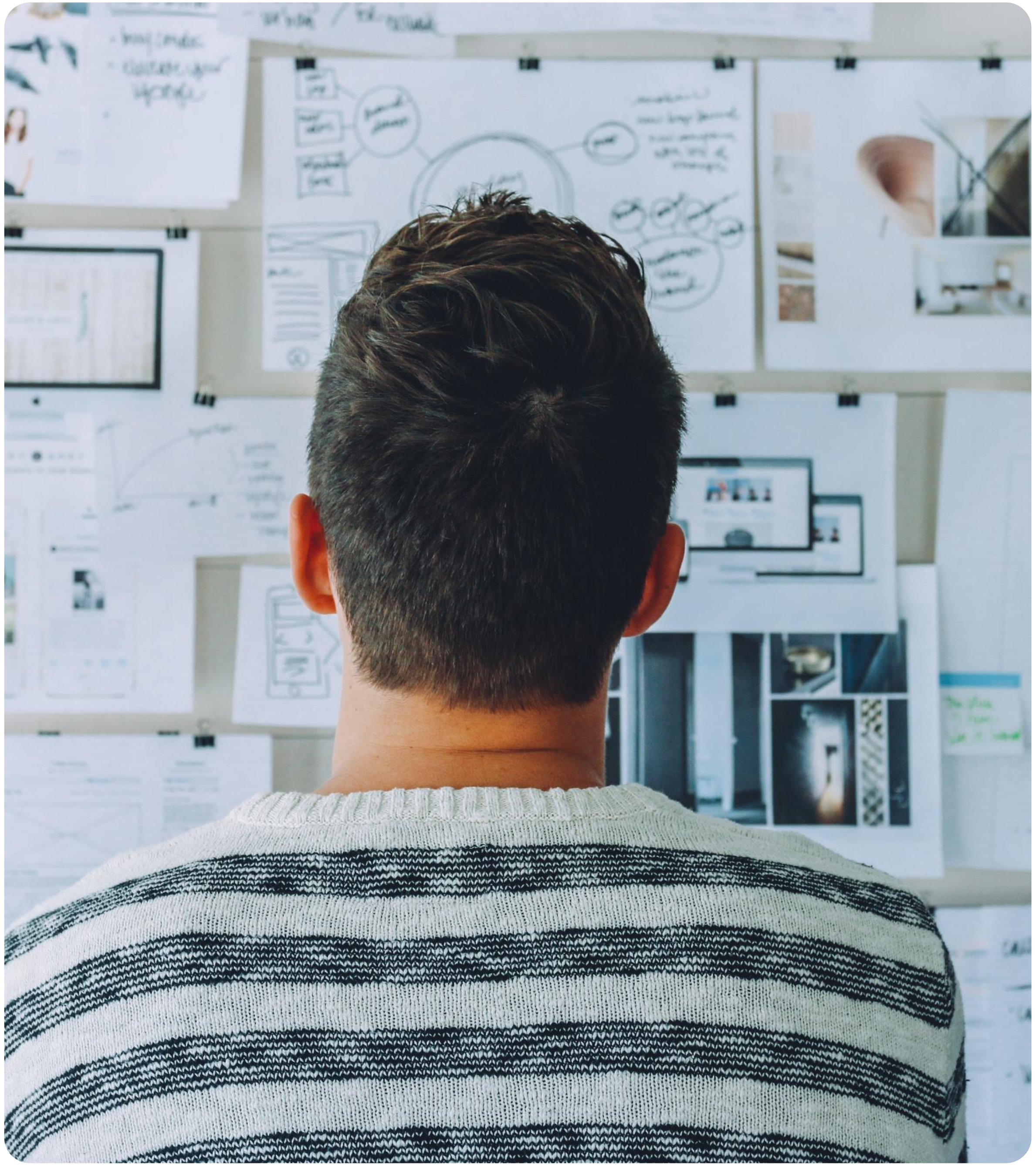
Observability & Analytics

Developer Experience

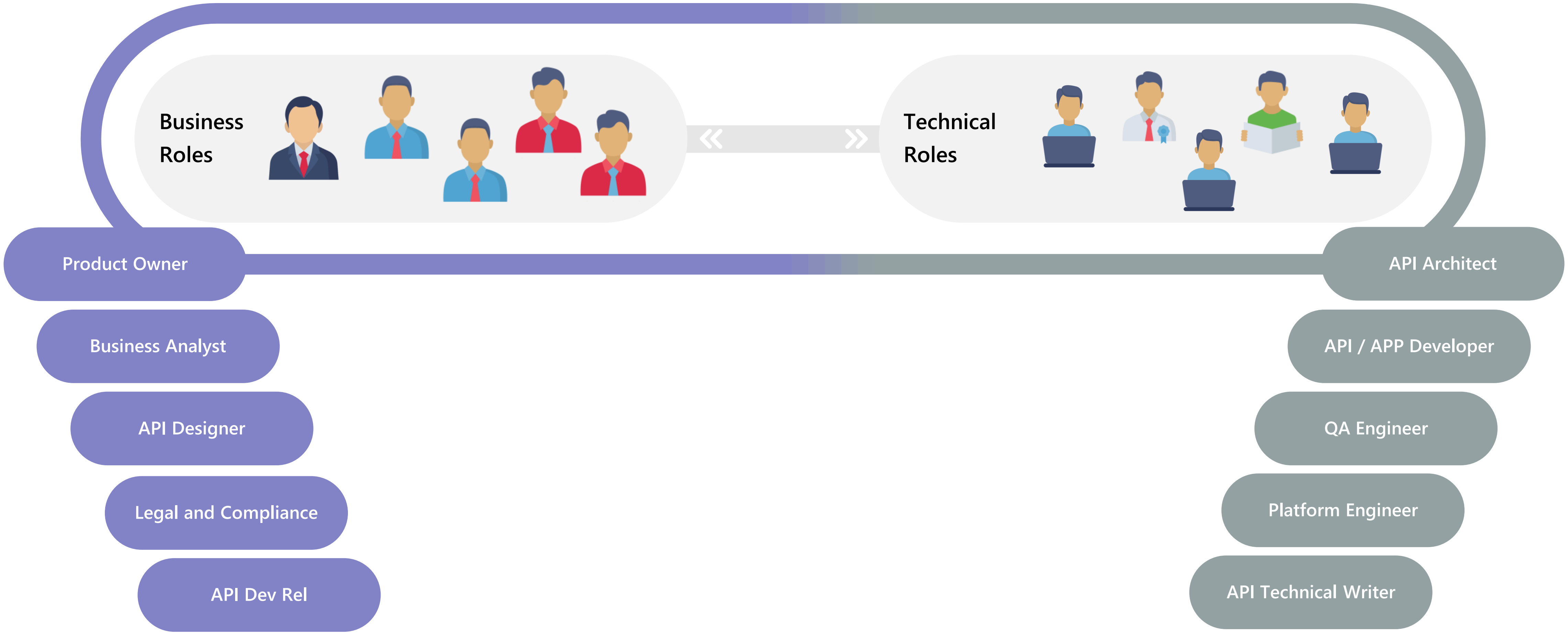
Delivery & Ops

Governance & Lifecycle Management

Compliance & Legal



Roles involved in an API program



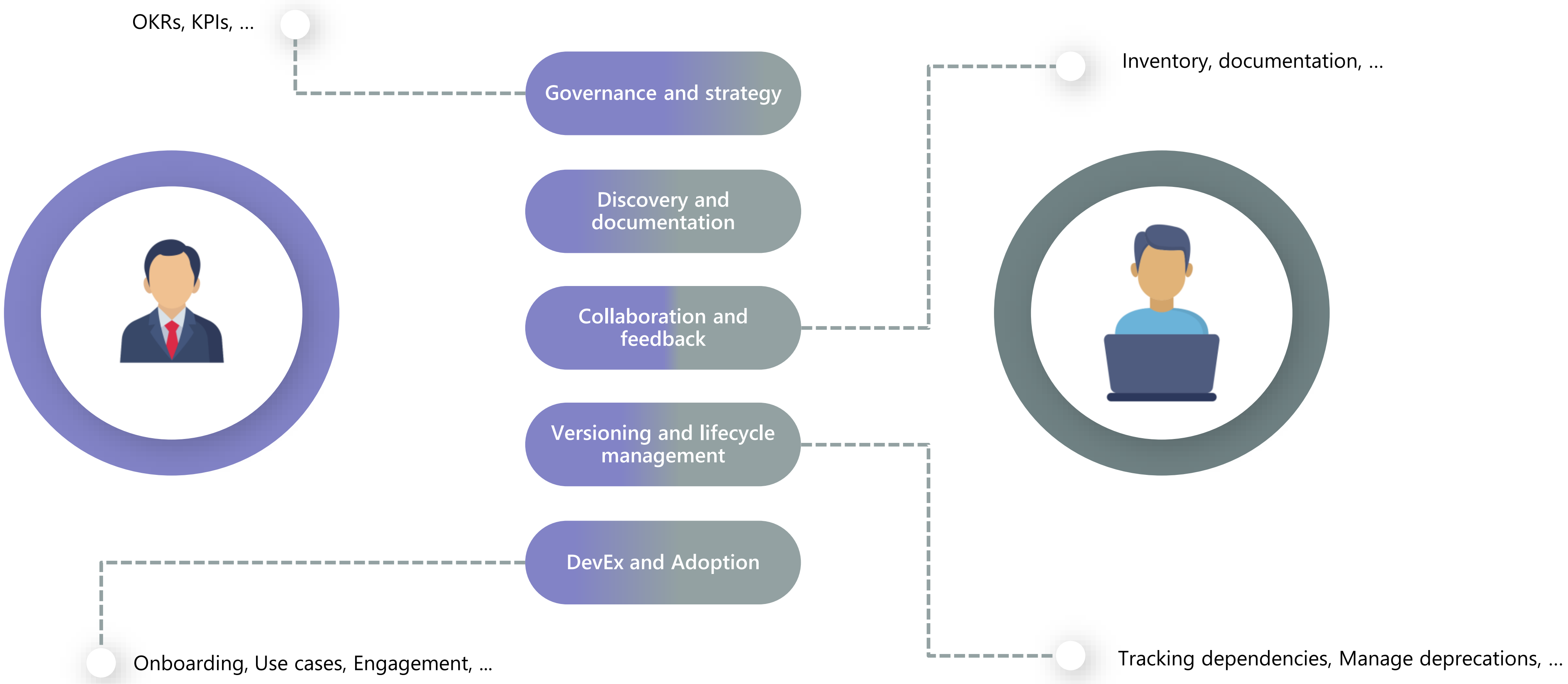
Business Roles

Technical Roles

- Product Owner
- Business Analyst
- API Designer
- Legal and Compliance
- API Dev Rel

- API Architect
- API / APP Developer
- QA Engineer
- Platform Engineer
- API Technical Writer

Roles and objectives



OKRs, KPIs, …

Governance and strategy

Discovery and documentation

Collaboration and feedback

Versioning and lifecycle management

DevEx and Adoption

Onboarding, Use cases, Engagement, …

Inventory, documentation, …

Tracking dependencies, Manage deprecations, …

Working towards a shared goal : a better API Governance

consistency and standardization

follow the same set of standards (e.g. style guidelines)

security

protect APIs and data from unauthorized access (e.g. OWASP top 10 API)

compliance

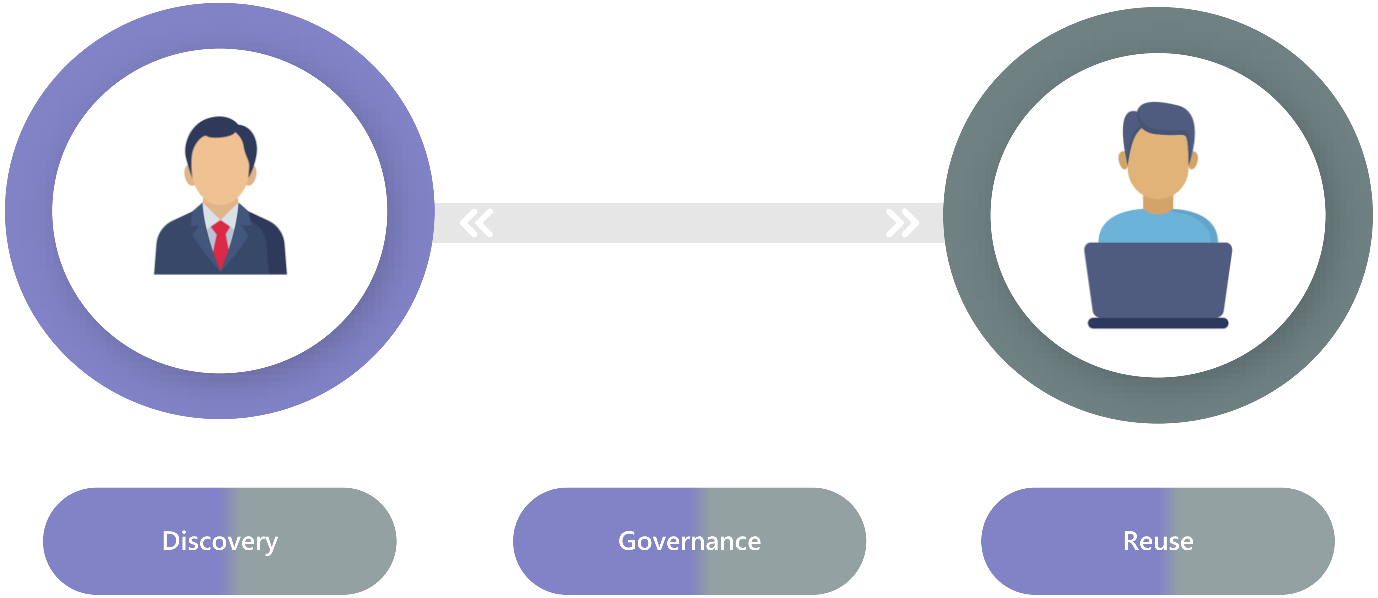
respect a specific industry or country regulations (e.g. GDPR)

agility

well-defined versioning and change management (innovate without impact on consumers)

The need for an API catalog

Bring together the people, technology and process



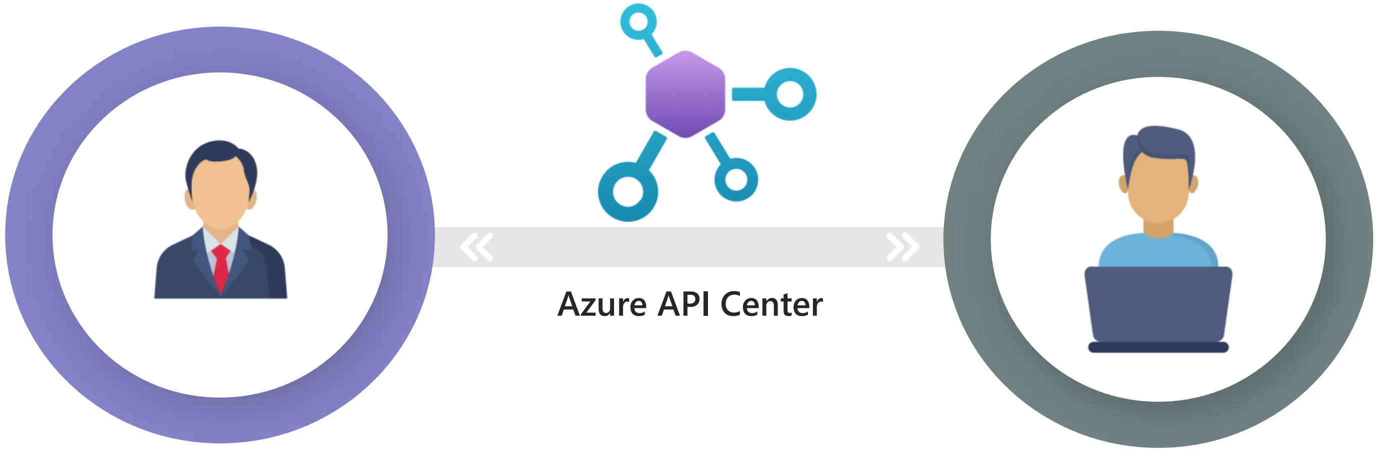
Discovery

Governance

Reuse

The need for an API catalog

Bring together the different roles involved in an API program



Azure API Center

Discovery

Avoid fragmented API inventories and prevent unmanaged APIs

Governance

Shadow and ungoverned APIs are low-hanging fruit for attackers

Reuse

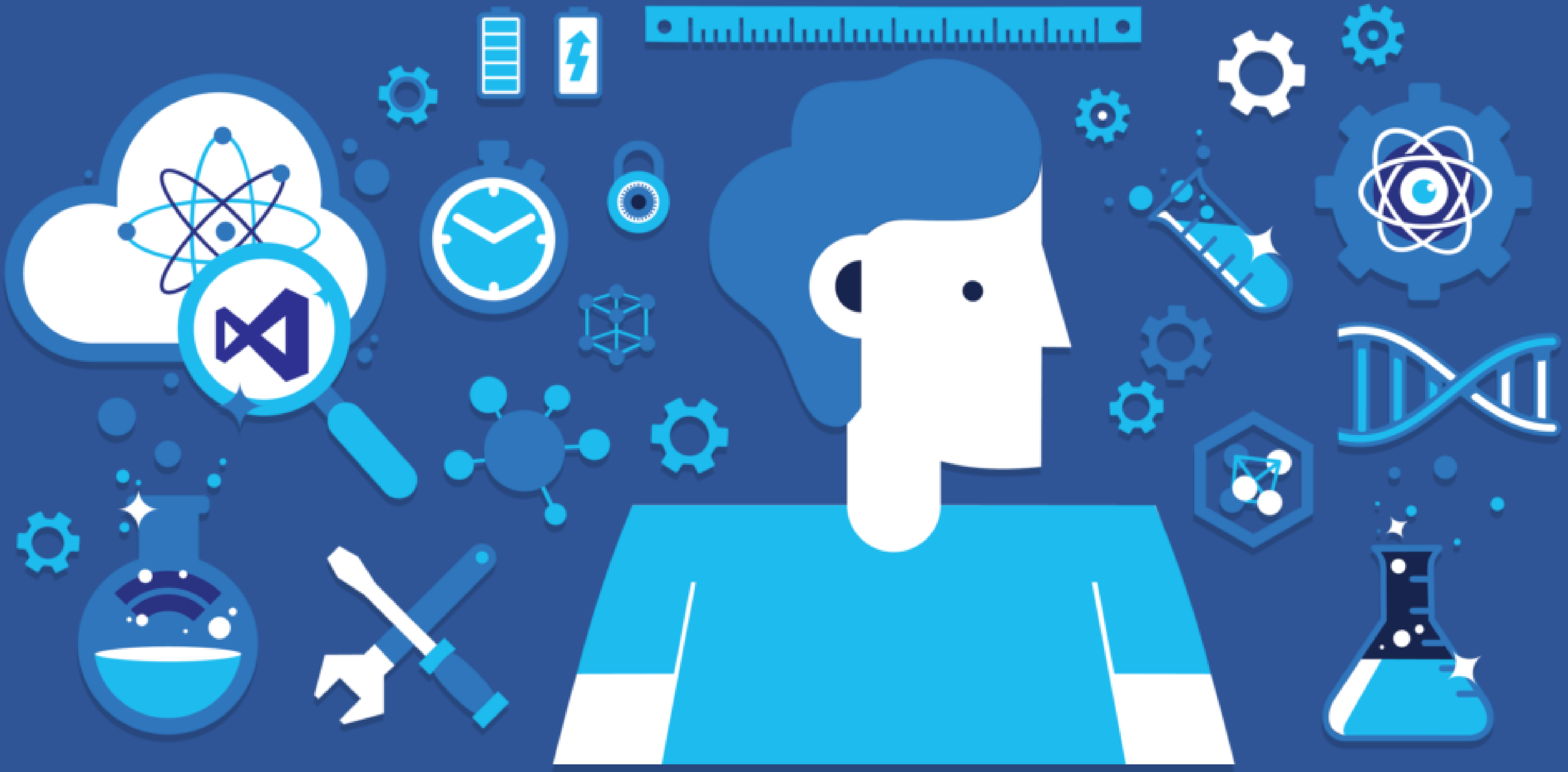
Increase efficiency and productivity

Welcome Azure API Center

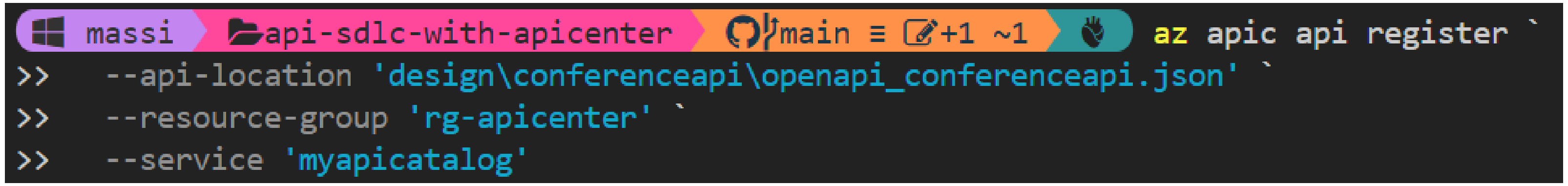
- Create API Inventory no matter where are located
- Track information you need according to your governance model
- Discover the different initiatives in your organization
- Maximise the developer productivity

AZURE DAY

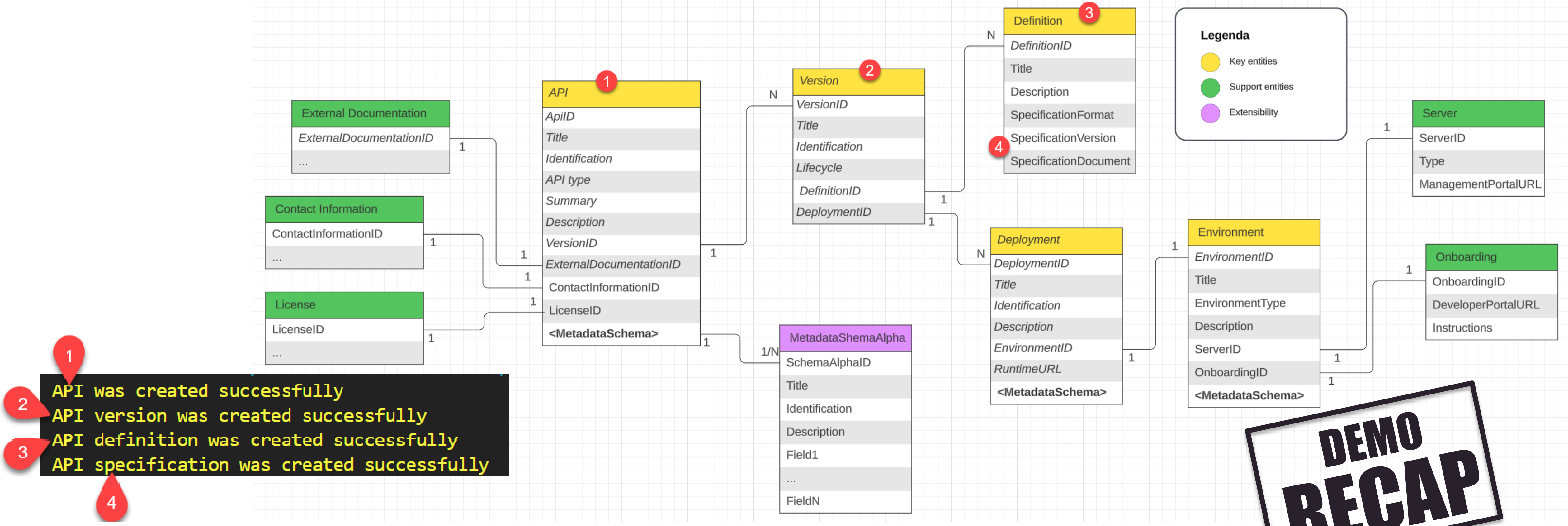
Demo : API Center first sight



First look : az apic api register



```
massi  api-sdlc-with-apicenter  main ≡ +1 ~1   az apic api register `
>>    --api-location 'design\conferenceapi\openapi_conferenceapi.json' `
>>    --resource-group 'rg-apicenter' `
>>    --service 'myapicatalog'
```



External Documentation
ExternalDocumentationID
...

Contact Information
ContactInformationID
...

License
LicenseID
...

API
ApiID
Title
Identification
API type
Summary
Description
VersionID
ExternalDocumentationID
ContactInformationID
LicenseID
<MetadataSchema>

Version
VersionID
Title
Identification
Lifecycle
DefinitionID
DeploymentID

Definition
DefinitionID
Title
Description
SpecificationFormat
SpecificationVersion
SpecificationDocument

MetadataShemaAlpha
SchemaAlphaID
Title
Identification
Description
Field1
...
FieldN

Deployment
DeploymentID
Title
Identification
Description
EnvironmentID
RuntimeURL
<MetadataSchema>

Environment
EnvironmentID
Title
EnvironmentType
Description
ServerID
OnboardingID
<MetadataSchema>

Server
ServerID
Type
ManagementPortalURL

Onboarding
OnboardingID
DeveloperPortalURL
Instructions

Legenda
● Key entities
● Support entities
● Extensibility

```
API was created successfully
API version was created successfully
API definition was created successfully
API specification was created successfully
```

DEMO RECAP

Let’s focus on two personas

API Product Manager

make sure all the moving parts come together as expected

API Developer

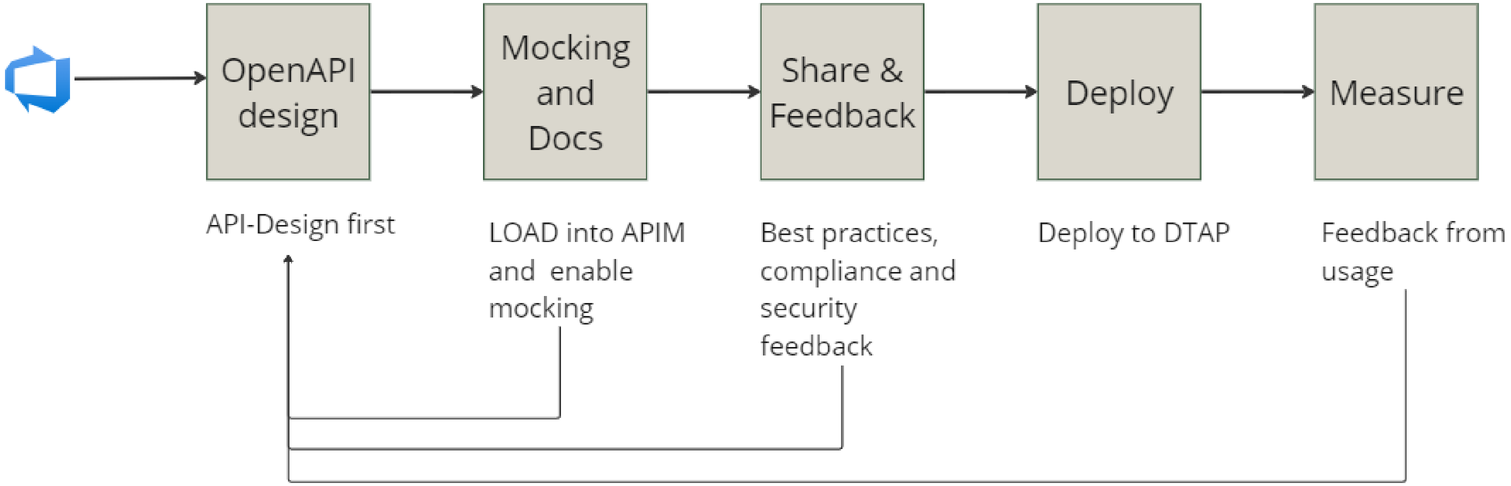
reduce the time to market (discover, select, build, run)

Linda

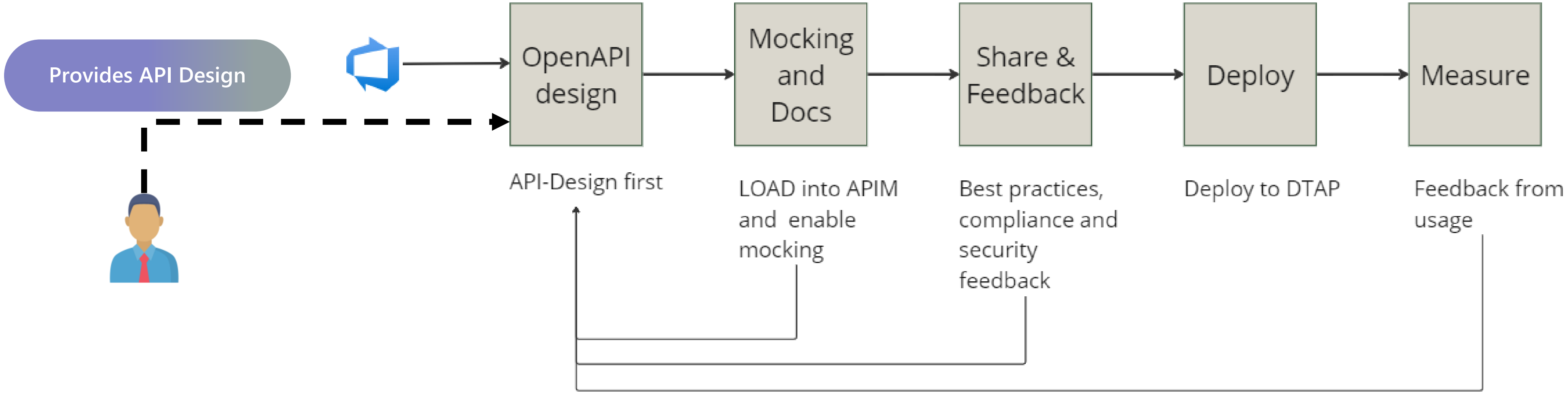
API Product Manager

API Design-First workflow and API Center

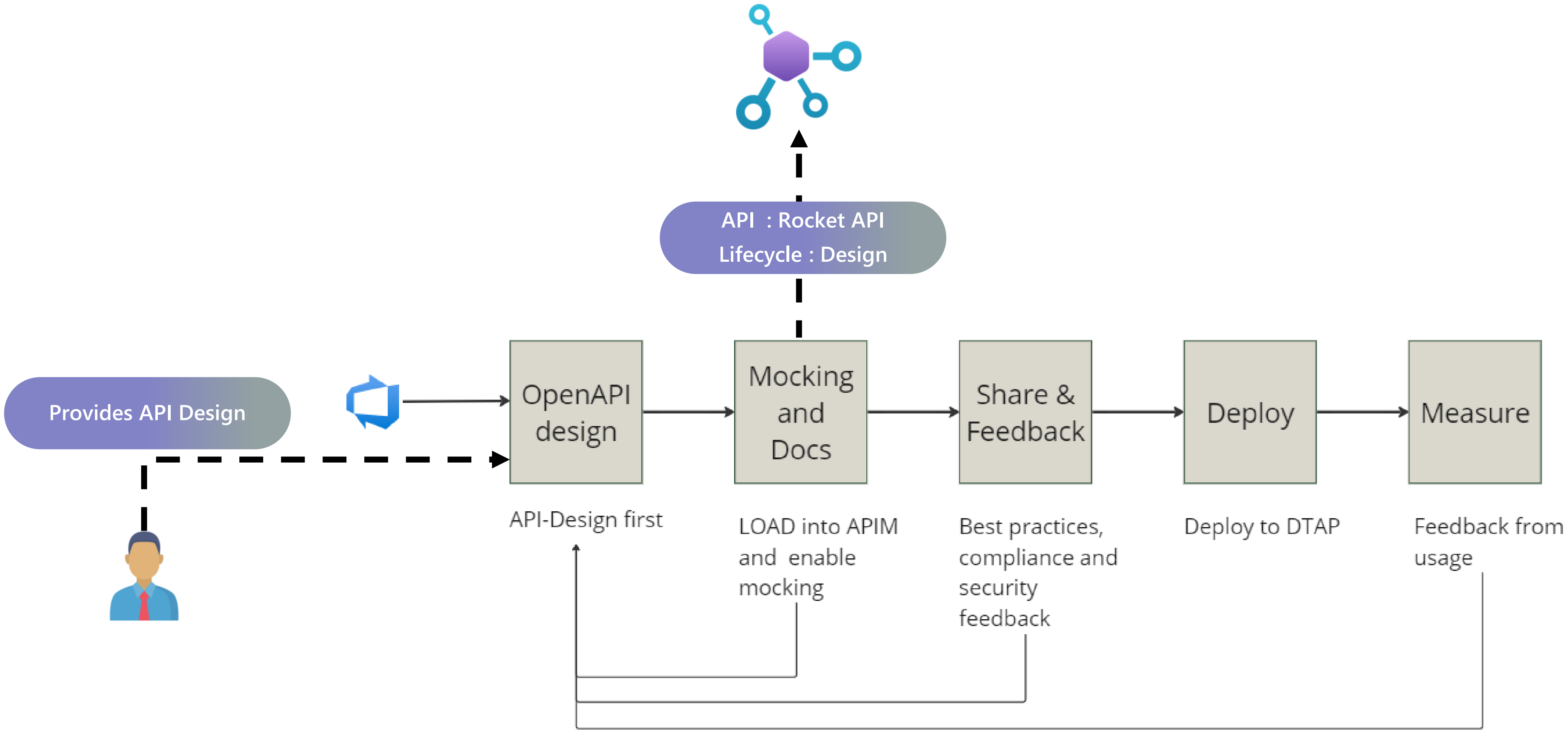
Open API as source of truth, closing the feedback look and using API Center to govern the lifecycle



API Design-First workflow and API Center



API Design-First workflow and API Center



Provides API Design

API : Rocket API
Lifecycle : Design

OpenAPI design

Mocking and Docs

Share & Feedback

Deploy

Measure

API-Design first

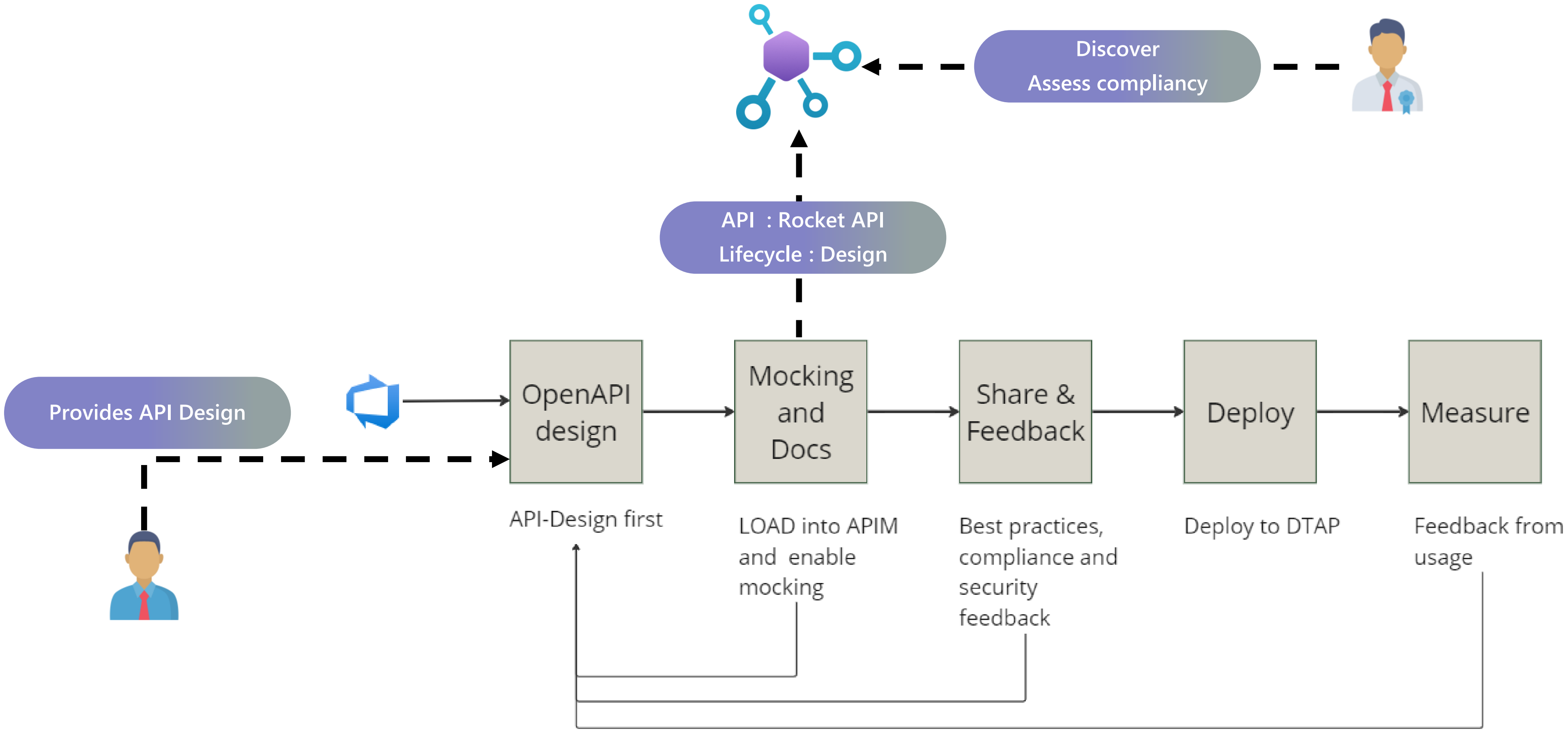
LOAD into APIM and enable mocking

Best practices, compliance and security feedback

Deploy to DTAP

Feedback from usage

API Design-First workflow and API Center



Discover
Assess compliancy

API : Rocket API
Lifecycle : Design

Provides API Design

OpenAPI design → Mocking and Docs → Share & Feedback → Deploy → Measure

API-Design first

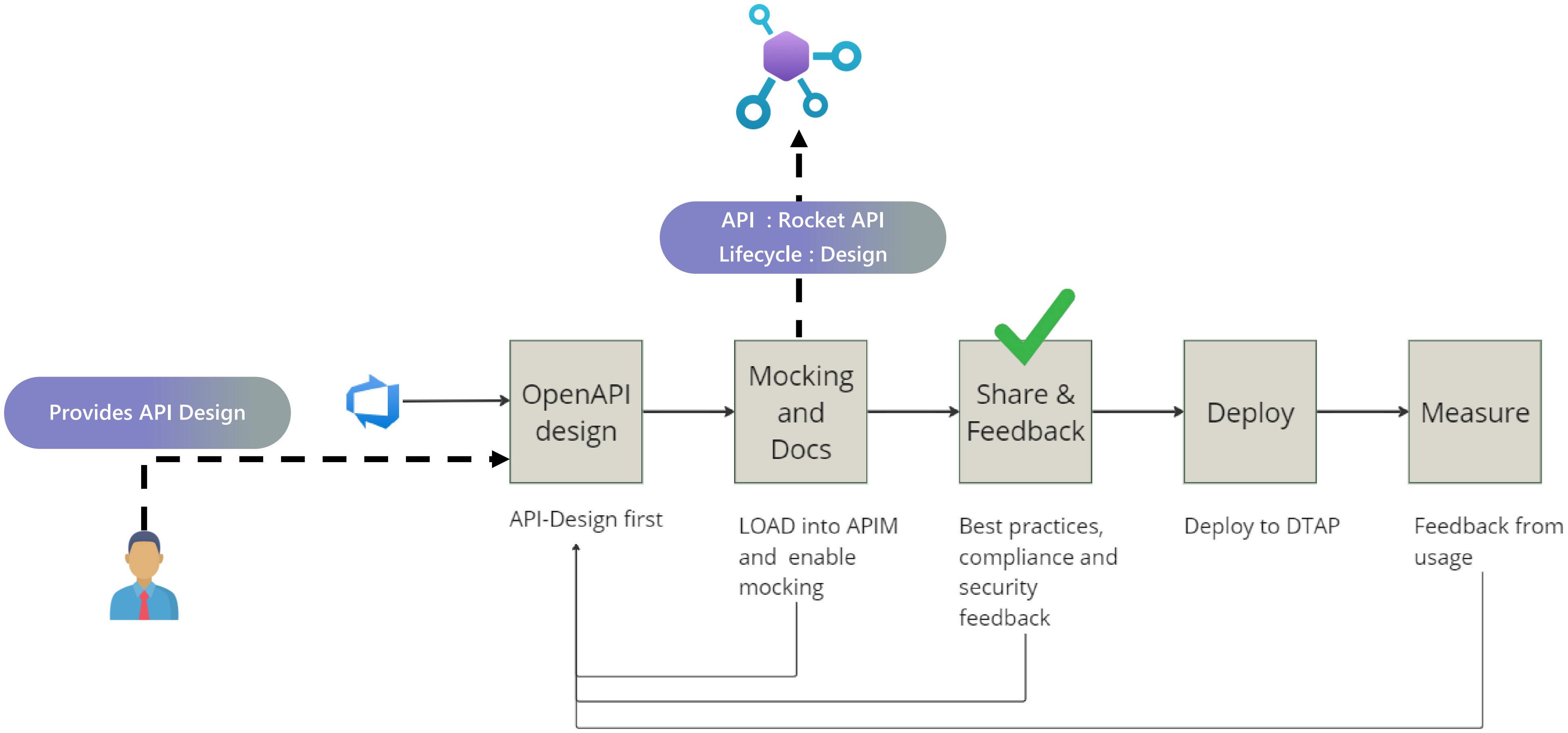
LOAD into APIM and enable mocking

Best practices, compliance and security feedback

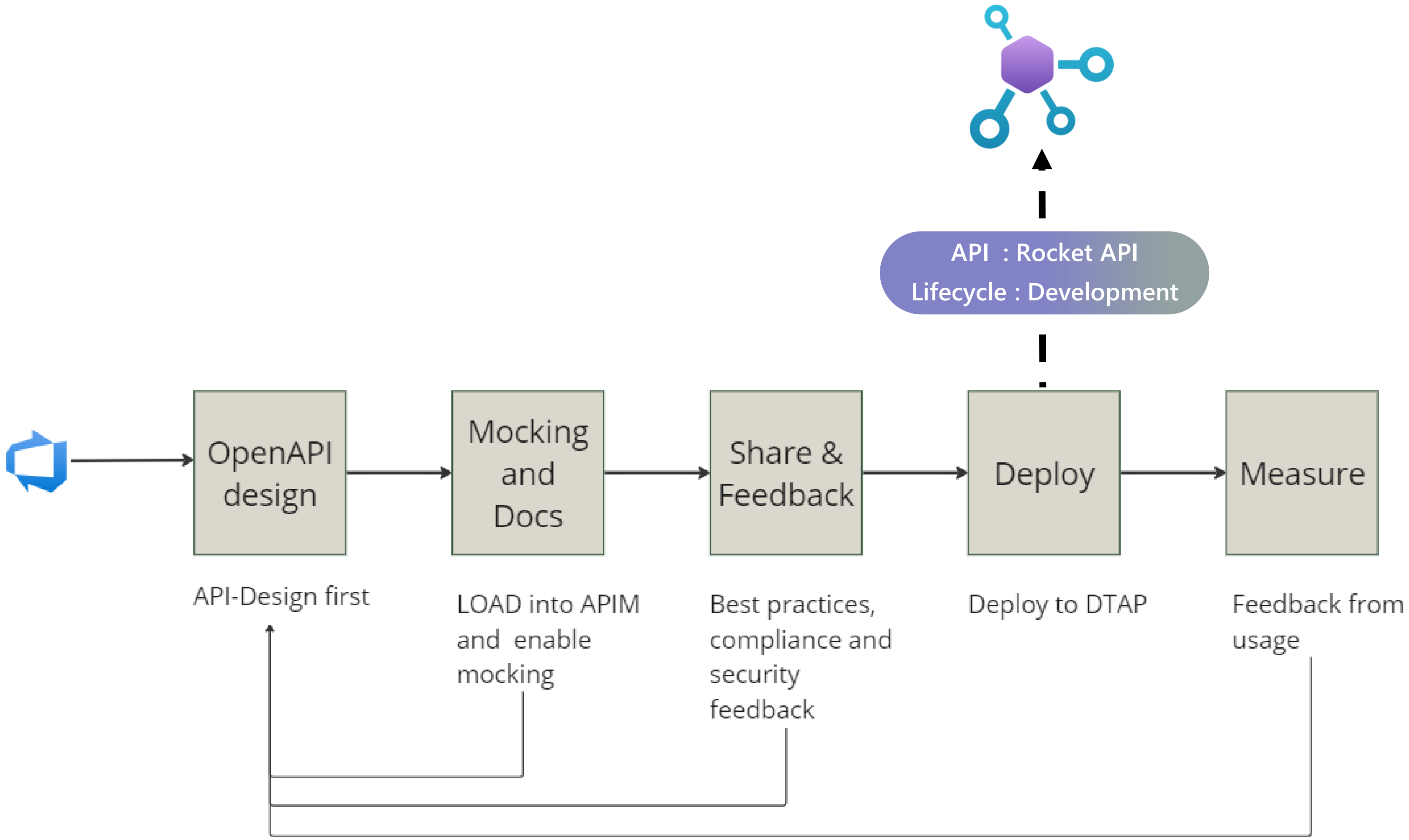
Deploy to DTAP

Feedback from usage

API Design-First workflow and API Center



API Design-First workflow and API Center



API : Rocket API
Lifecycle : Development

OpenAPI design

Mocking and Docs

Share & Feedback

Deploy

Measure

API-Design first

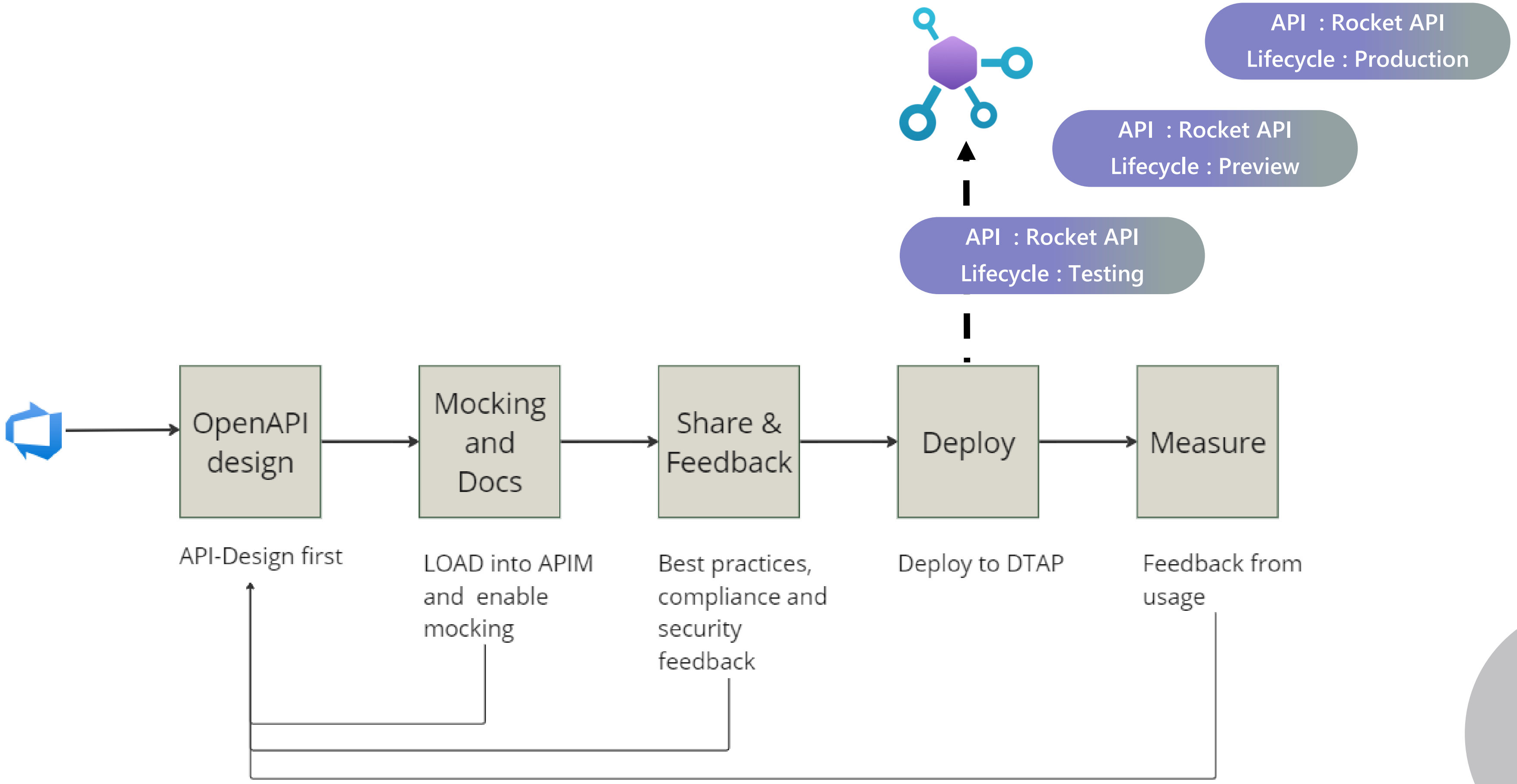
LOAD into APIM and enable mocking

Best practices, compliance and security feedback

Deploy to DTAP

Feedback from usage

API Design-First workflow and API Center



API : Rocket API
Lifecycle : Production

API : Rocket API
Lifecycle : Preview

API : Rocket API
Lifecycle : Testing

OpenAPI design

Mocking and Docs

Share & Feedback

Deploy

Measure

API-Design first

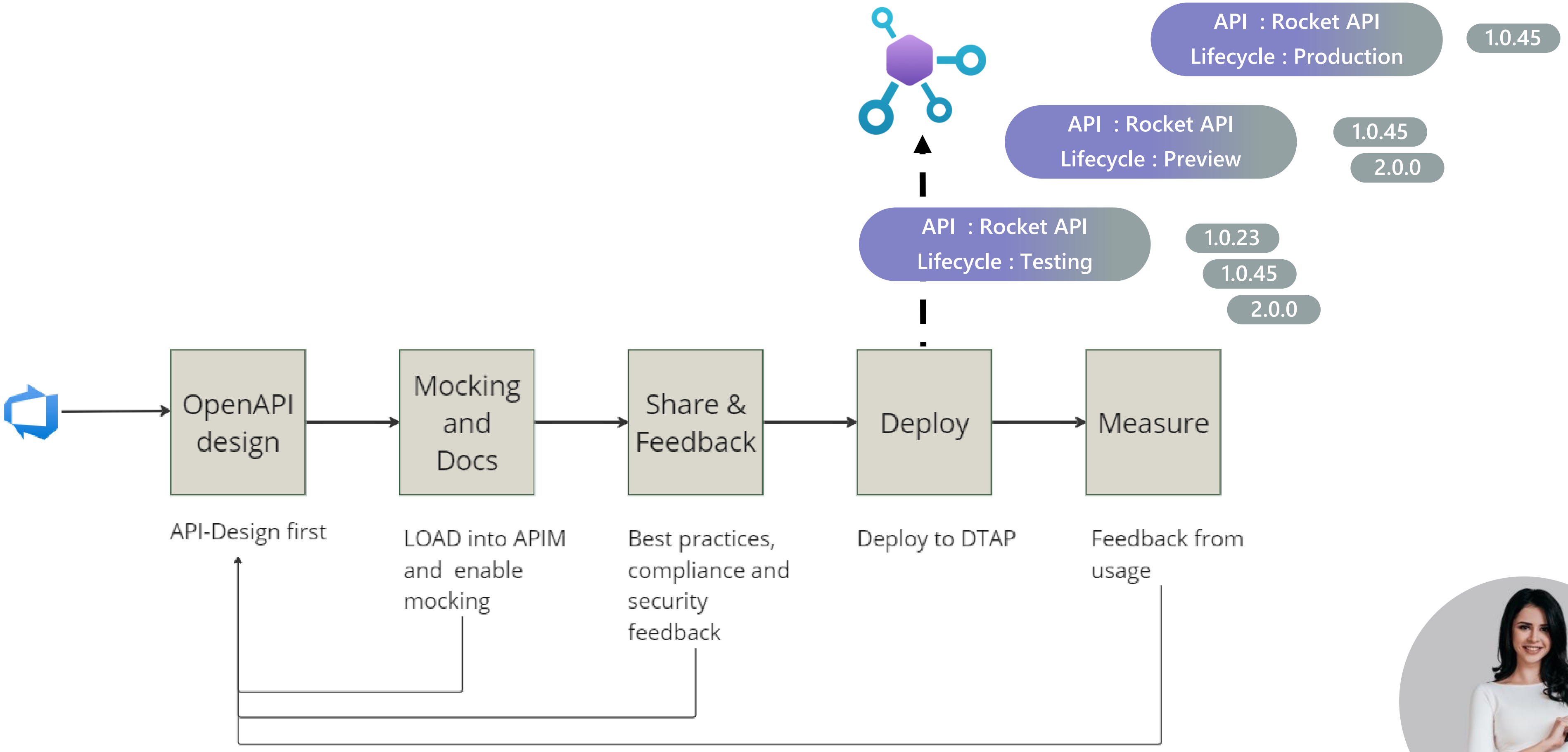
LOAD into APIM and enable mocking

Best practices, compliance and security feedback

Deploy to DTAP

Feedback from usage

API Design-First workflow and API Center



OpenAPI design

Mocking and Docs

Share & Feedback

Deploy

Measure

API-Design first

LOAD into APIM and enable mocking

Best practices, compliance and security feedback

Deploy to DTAP

Feedback from usage

API : Rocket API
Lifecycle : Testing

1.0.23
1.0.45
2.0.0

API : Rocket API
Lifecycle : Preview

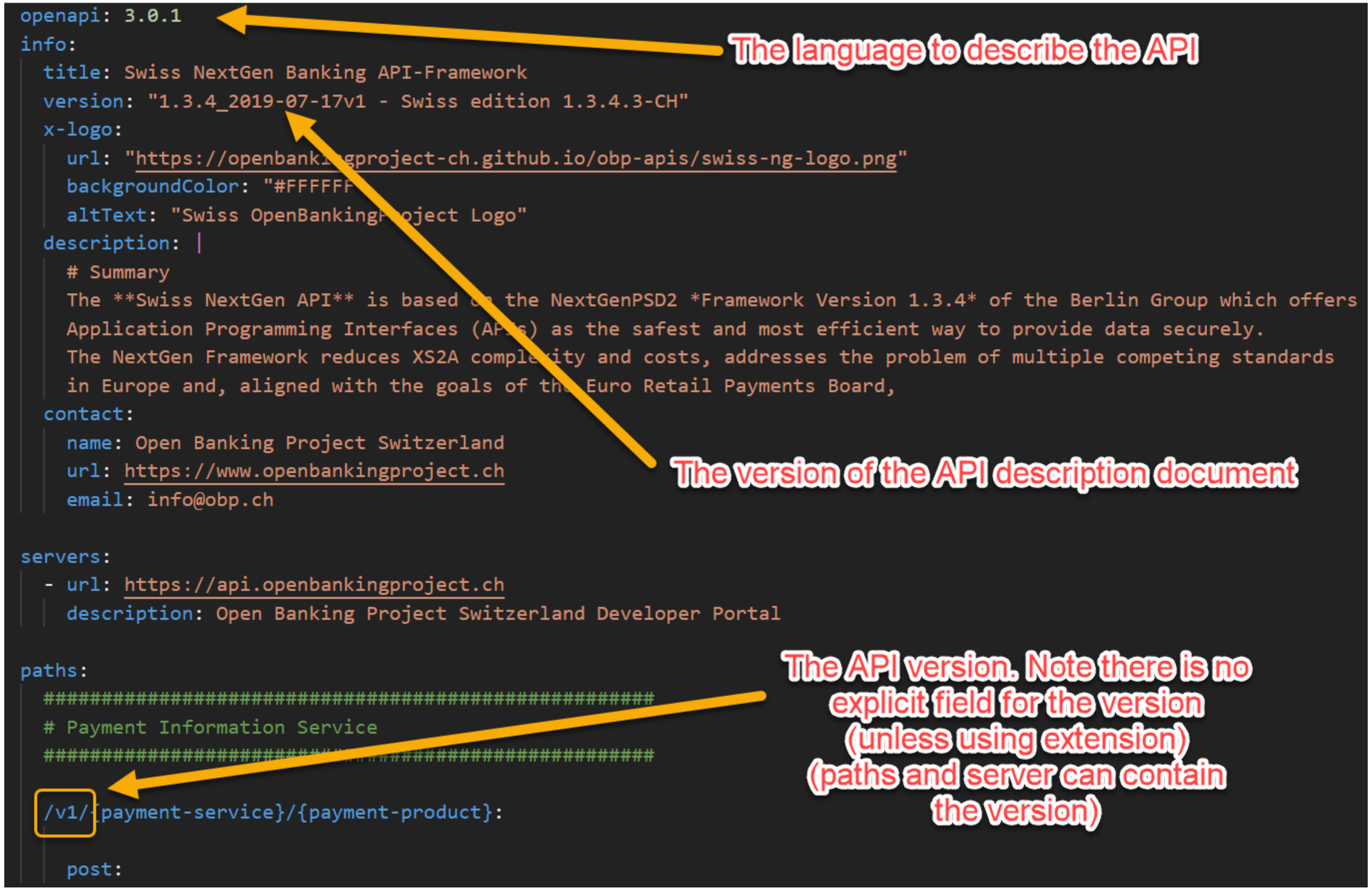
1.0.45
2.0.0

API : Rocket API
Lifecycle : Production

1.0.45

Versioning & Open API

OpenAPI, the description document, the API or the implementation?



```yaml
openapi: 3.0.1
info:
  title: Swiss NextGen Banking API-Framework
  version: "1.3.4_2019-07-17v1 - Swiss edition 1.3.4.3-CH"
  x-logo:
    url: "https://openbankingproject-ch.github.io/obp-apis/swiss-ng-logo.png"
    backgroundColor: "#FFFFFF"
    altText: "Swiss OpenBankingProject Logo"
  description: |
    # Summary
    The **Swiss NextGen API** is based on the NextGenPSD2 *Framework Version 1.3.4* of the Berlin Group which offers
    Application Programming Interfaces (APIs) as the safest and most efficient way to provide data securely.
    The NextGen Framework reduces XS2A complexity and costs, addresses the problem of multiple competing standards
    in Europe and, aligned with the goals of the Euro Retail Payments Board,
  contact:
    name: Open Banking Project Switzerland
    url: https://www.openbankingproject.ch
    email: info@obp.ch

servers:
  - url: https://api.openbankingproject.ch
    description: Open Banking Project Switzerland Developer Portal

paths:
  #####################################################
  # Payment Information Service
  #####################################################

  /v1/{payment-service}/{payment-product}:

    post:
```

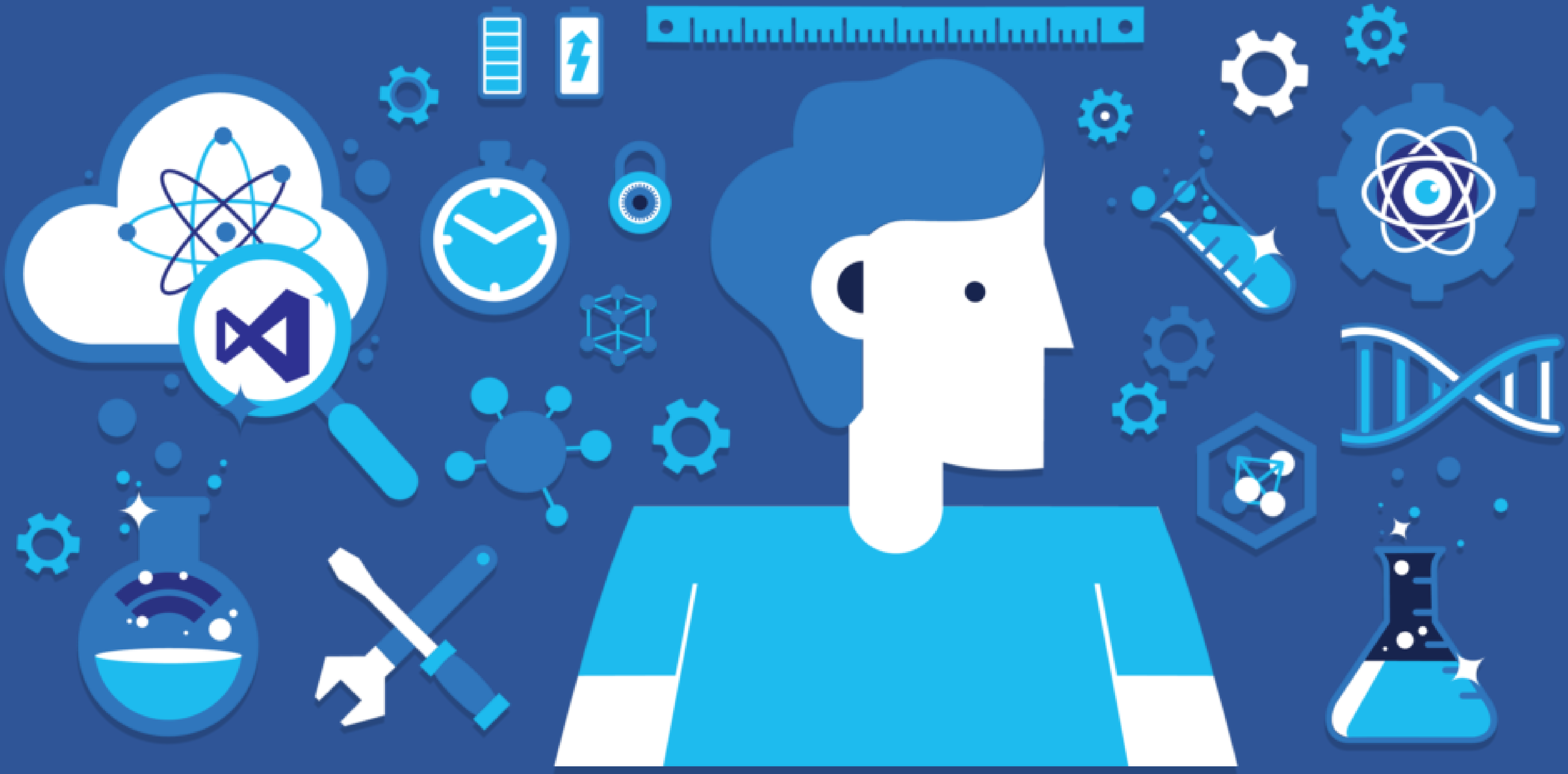
The language to describe the API

The version of the API description document

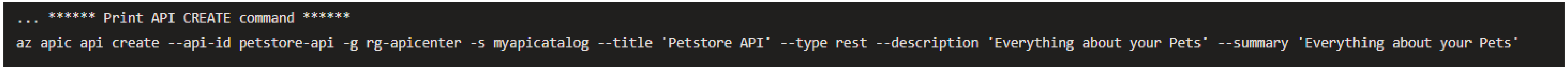
The API version. Note there is no explicit field for the version (unless using extension) (paths and server can contain the version)

AZURE DAY

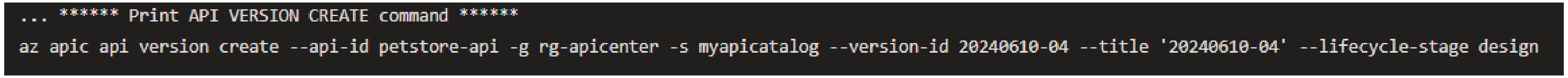
Demo : API Governance with API Center



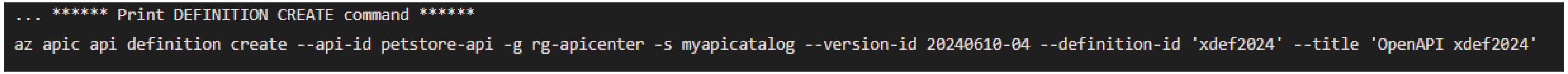
Add API to the catalog & register a deployment



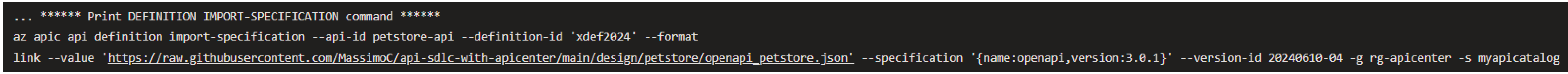
```
... ****** Print API CREATE command ******
az apic api create --api-id petstore-api -g rg-apicenter -s myapicatalog --title 'Petstore API' --type rest --description 'Everything about your Pets' --summary 'Everything about your Pets'
```



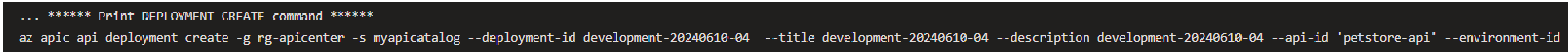
```
... ****** Print API VERSION CREATE command ******
az apic api version create --api-id petstore-api -g rg-apicenter -s myapicatalog --version-id 20240610-04 --title '20240610-04' --lifecycle-stage design
```



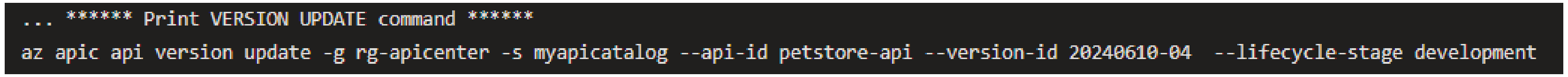
```
... ****** Print DEFINITION CREATE command ******
az apic api definition create --api-id petstore-api -g rg-apicenter -s myapicatalog --version-id 20240610-04 --definition-id 'xdef2024' --title 'OpenAPI xdef2024'
```



```
... ****** Print DEFINITION IMPORT-SPECIFICATION command ******
az apic api definition import-specification --api-id petstore-api --definition-id 'xdef2024' --format
link --value 'https://raw.githubusercontent.com/MassimoC/api-sdlc-with-apicenter/main/design/petstore/openapi_petstore.json' --specification '{name:openapi,version:3.0.1}' --version-id 20240610-04 -g rg-apicenter -s myapicatalog
```



```
... ****** Print DEPLOYMENT CREATE command ******
az apic api deployment create -g rg-apicenter -s myapicatalog --deployment-id development-20240610-04  --title development-20240610-04 --description development-20240610-04 --api-id 'petstore-api' --environment-id
```



```
... ****** Print VERSION UPDATE command ******
az apic api version update -g rg-apicenter -s myapicatalog --api-id petstore-api --version-id 20240610-04  --lifecycle-stage development
```

DEMO
RECAP

About applying quality checks

Three options you can combine to see what works best for you

Shift-left to developers

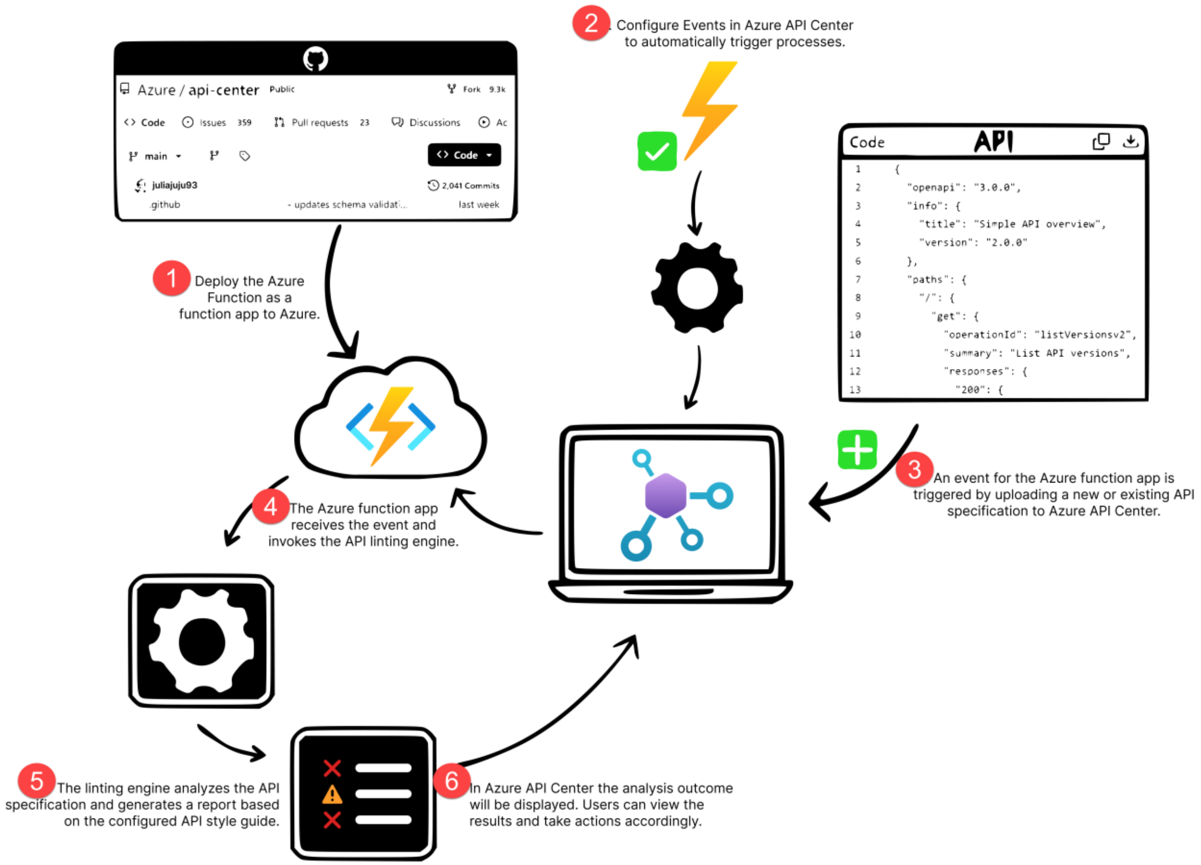
Get the company rules and linting in VSCode

Continuous check in CICD

Linting in pipelines (what we saw in the demo)

Check across entire API Catalog

EventGrid integration and API Analyzer



```
{
  "openapi": "3.0.0",
  "info": {
    "title": "Simple API overview",
    "version": "2.0.0"
  },
  "paths": {
    "/": {
      "get": {
        "operationId": "listVersionsv2",
        "summary": "List API versions",
        "responses": {
          "200": {
```

1 Deploy the Azure Function as a function app to Azure.

2 Configure Events in Azure API Center to automatically trigger processes.

3 An event for the Azure function app is triggered by uploading a new or existing API specification to Azure API Center.

4 The Azure function app receives the event and invokes the API linting engine.

5 The linting engine analyzes the API specification and generates a report based on the configured API style guide.

6 In Azure API Center the analysis outcome will be displayed. Users can view the results and take actions accordingly.

https://github.com/Azure/APICenter-Analyzer/

Emma

API Developer

(API consumer standpoint)

SDKs are the developers’ painkiller

“I do not waste my time in serialization and deserialization code”

“Please abstract me away
from your complex authentication mechanism”

“I Love it!
It’s so easy to understand, easy to use”

I do not want to learn a new SDK every time a new API integration is needed

“the SDK is not available in my language of choice”

“it is out of date”

“it is poorly implemented”

“I do not want to intake dependencies on libraries from dozens of API producers”

“I want a single tool to get consistent experience across APIs no matter the API provider”

- Emma -

API Center extension for VS Code

Discover APIs, get the documentation, generate SDKs and more …

API : Rocket API
Lifecycle : Production

1.0.45

API : Rocket API
Lifecycle : Preview

2.0.0

API : 1Password API
Lifecycle : Testing

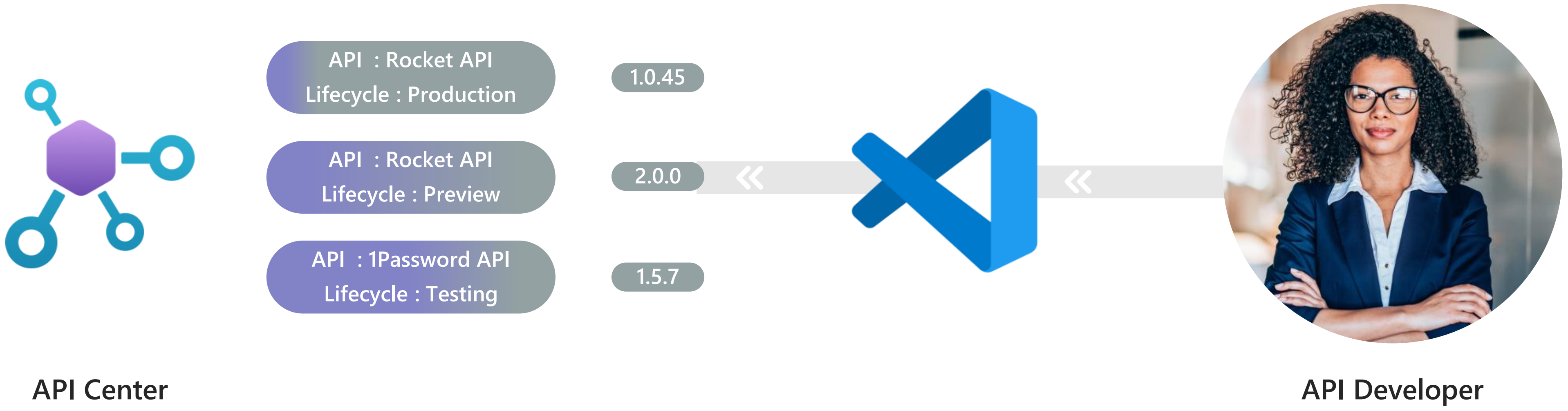
1.5.7

API Center

API Developer

API Center extension for VS Code

Discover APIs, get the documentation, generate SDKs and more …



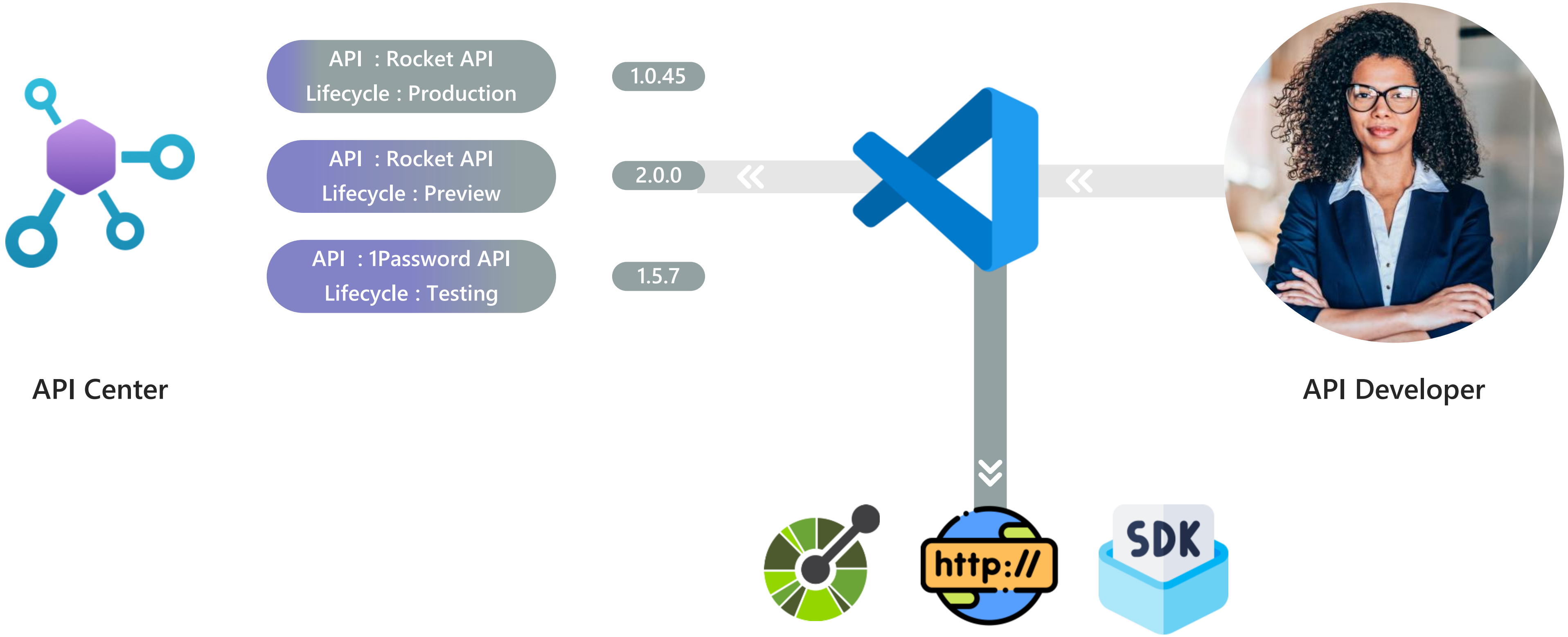
API : Rocket API
Lifecycle : Production

API : Rocket API
Lifecycle : Preview

API : 1Password API
Lifecycle : Testing

API Center extension for VS Code

Discover APIs, get the documentation, generate SDKs and more ...



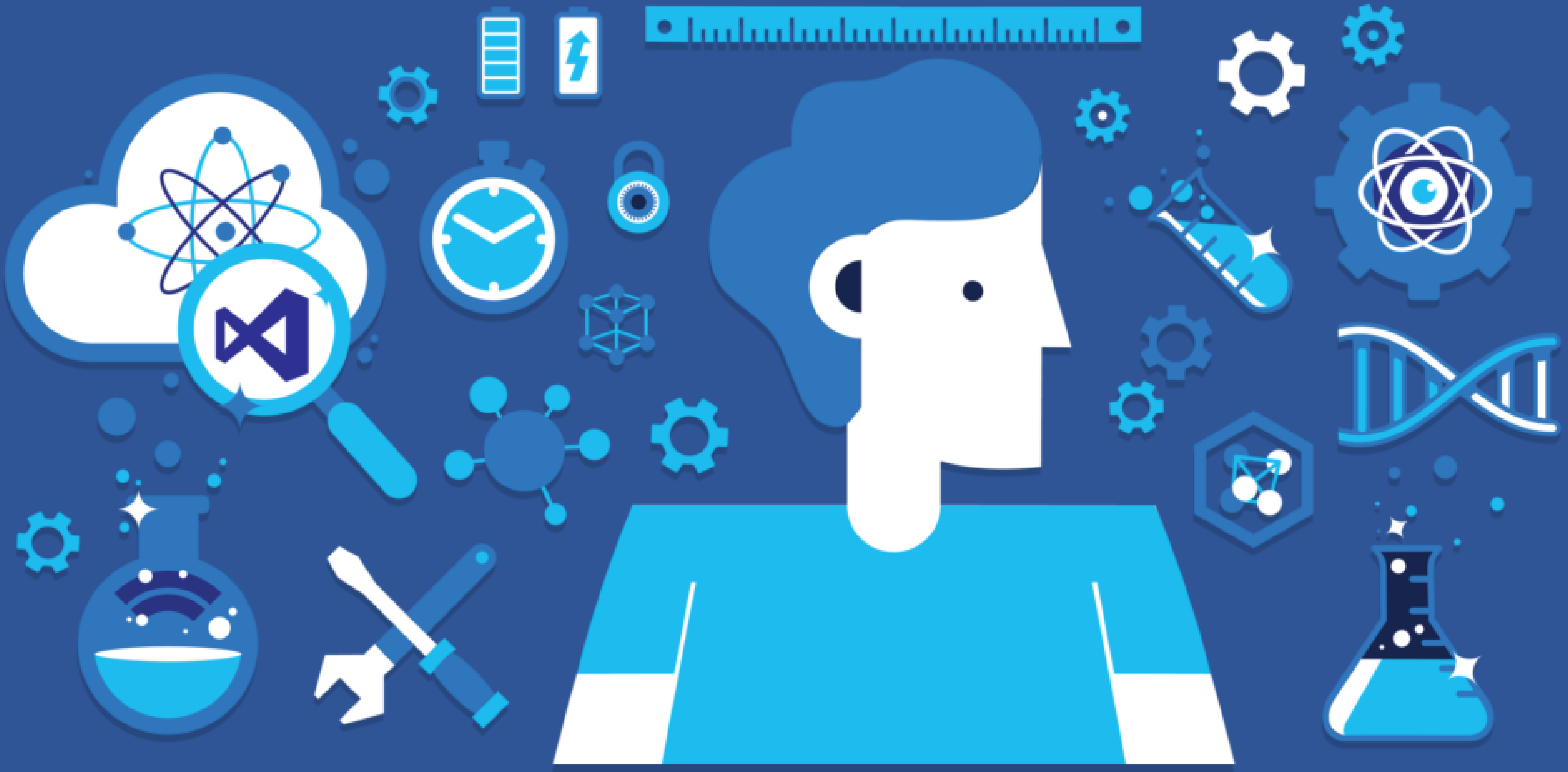
API : Rocket API
Lifecycle : Production

API : Rocket API
Lifecycle : Preview

API : 1Password API
Lifecycle : Testing

AZURE DAY

Demo : Discover and consume APIs

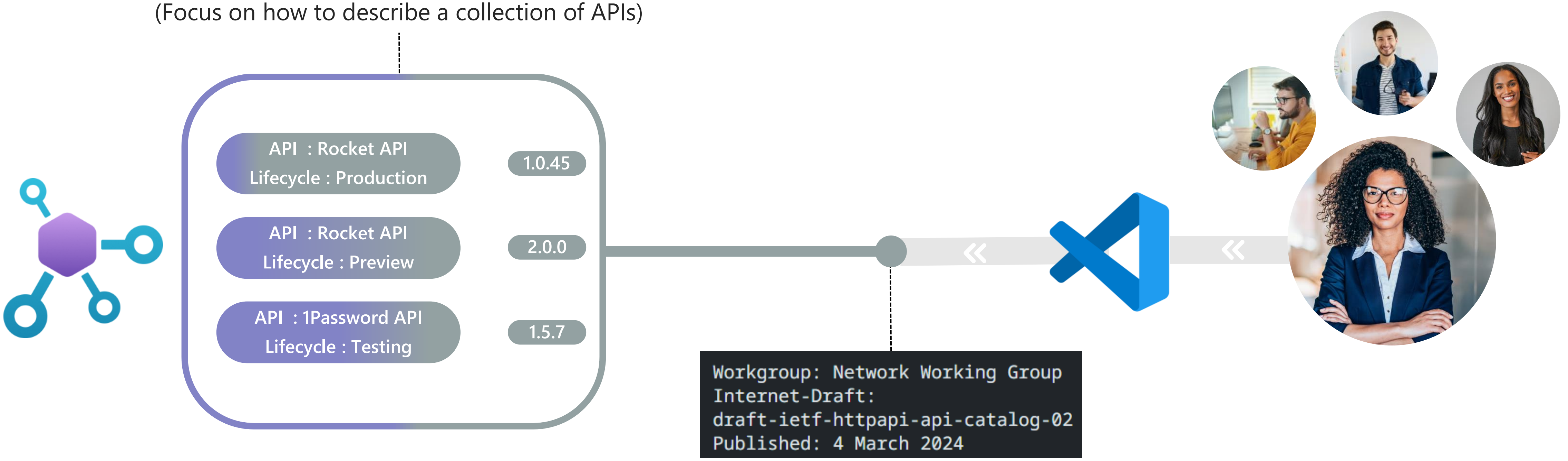


Towards an API Catalog standardization ?

Fully automated API discovery

APIs.json

(Focus on how to describe a collection of APIs)



API : Rocket API
Lifecycle : Production

1.0.45

API : Rocket API
Lifecycle : Preview

2.0.0

API : 1Password API
Lifecycle : Testing

1.5.7

```
Workgroup: Network Working Group
Internet-Draft:
draft-ietf-httpapi-api-catalog-02
Published: 4 March 2024
```

api-catalog IETF draft

(Define the location where to find the api-catalog)
https://apis.example.com/.well-known/api-catalog

AZURE DAY ROME 2024

VOTA LA SESSIONE

Facci sapere se questa sessione ti piace.
Inquadra il QR code e esprimi una tua opinione.
Ci aiuterai a migliorare.