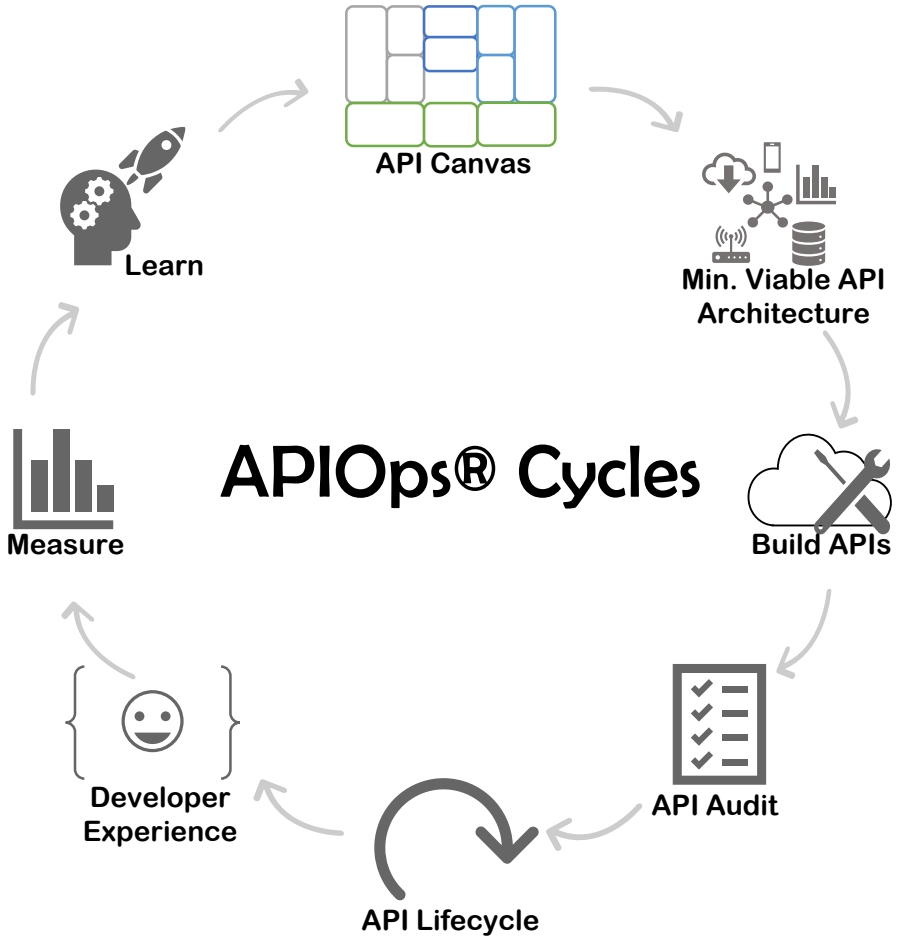


”name”:
”**APIOps® Cycles Design Book**”,

”description":
”APIOps® Cycles is a Lean Method for Designing and Managing API Products",

”audience”: [”Business Developers”, ”Product Managers”, ”Architects”, ”Developers”, ”Integration specialists”],

”version": ”1.1-beta",
”authors":[”Marjukka Niinioja, Digia Oyj”, ”Turo Hyppönen Digia Oyj”]

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

Overview

What is APIOps® Cycles?

The APIOps© Cycles is a set of tools and methods originally created by Digia. The tools and methods are designed to help all organizations and individuals to build great APIs.

APIOps©Cycles method uses a re-interpreted version of an API Canvas. We have added our experience to the model which is a form of popular business model canvas.

You can learn and adapt the method to your API needs either on your own or with API consultants.

The APIOps© Cycles method uses modern and proven frameworks with a twist of our experience to fit them for APIs. All methods have a Lean management base: Business Model Canvas, Value Proposition Canvas, Lean Startup, Minimum Viable Architecture and DevOps .

The method fits also in an organization where the working methods are not yet very lean or agile. Or at least there is a clear separation of development and operations. The main benefit of the methods is that they enforce communication. APIOps© Cycles helps communication between different business roles and different IT roles.

First method is API Canvas, i.e. making sure we know what benefits come and to who for using the API. The Canvas helps to define the key features of the API. It also clarifies what we need to build to make the API real.

What can I do with API Canvas?

Main benefit of API Canvas compared to for example project scoping tools is that it treats the API as a product. This means that it tries to find the needs of several API Consumer segments, not just the one at hand.

It also directs user to think how to communicate and support the API Consumers and give them access to the API. It also focuses on how to make money, added value like customer retention or cost savings with the API.

Start with the API Value Proposition Canvas. It's a **great interviewing tool**. Use it when finding and validating requirements with API Consumers.

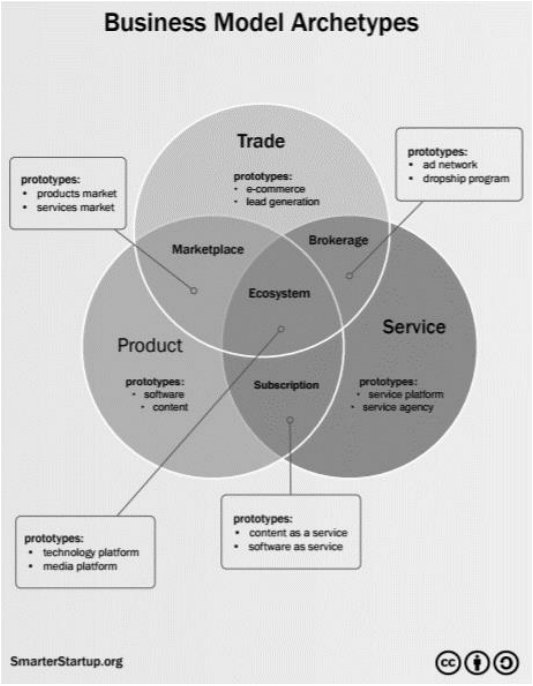
Digia API Canvas has 9 areas, which have almost the same titles as in Business Model Canvas. **API Canvas is the master document for next phases, including architecture design.**

You should share the API Canvas with all relevant stakeholders. This includes external developers and other partners you need to work with. The simpler API Value Proposition Canvas helps to create value proposition and related information to the API Canvas.

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

Why APIs are a business issue?

- Because APIs make all business models possible
- APIs can bring additional value to customers and partners allowing them to access their own data or use your services via their own interfaces
- API is a product which can be sold, commonly with a subscription or value based model
- Ecosystem can provide and consume APIs to create a unified, complete customer journey and share value
- APIs can be sold on a marketplace. They enable physical and digital products and services to be sold on a marketplace
- APIs used by ecosystem to create their own innovation solutions which can be used as a lead generation channel
- APIs can handle orders, payments and logistics enabling e-commerce solutions based on APIs



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

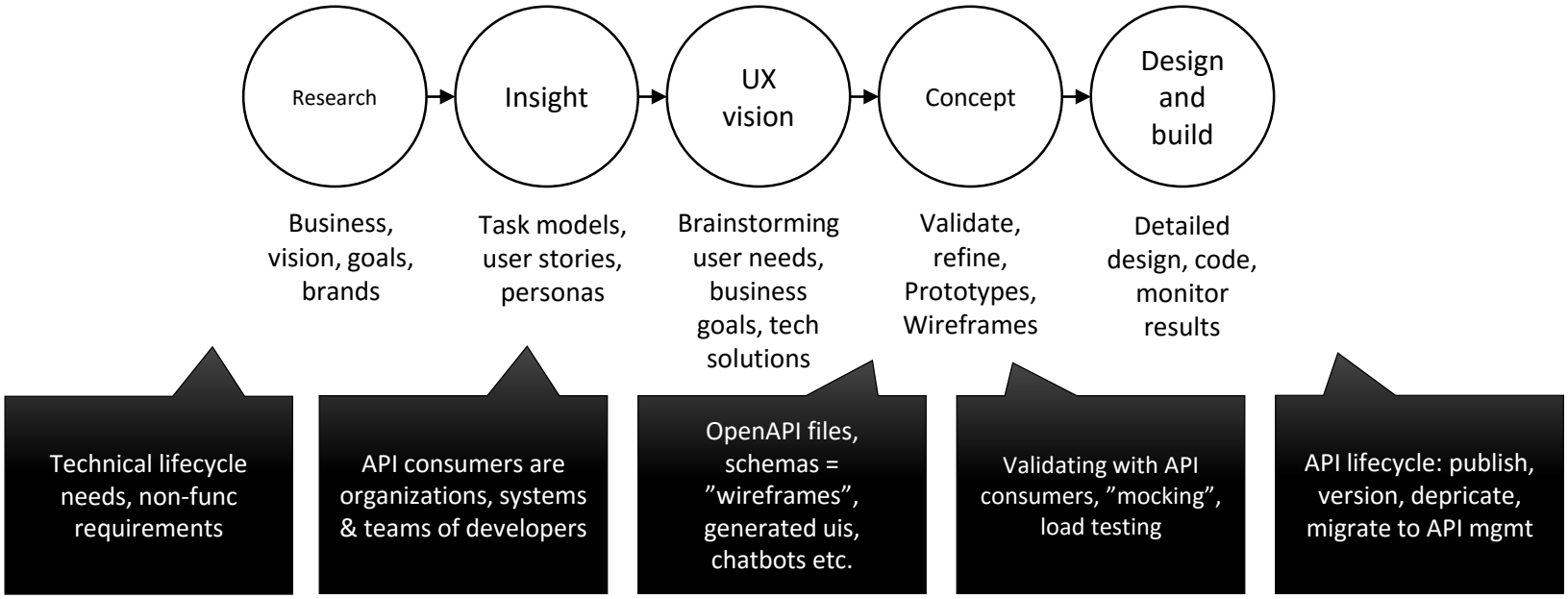
API Business Models in a nutshell

	Internal cost optimization	Indirect value	Additional sales	API consumer pays	API consumer is paid
Maturity	level 1 - 2	level 2-3	level 3	level 4	level 4
Revenue model	Market share ROI	Restricted free use "Freemium" model or for specific partner uses	APIs are included in the premium product/service packages or offered free for key customers	Pay-per-use or according to added value	Revenue sharing, %-based partnership
Opportunities	• Reacting faster to market changes • Faster technology adoption • Shorter development time: distributed simultaneous development, promoting reuse • As part of cloud adoption strategy: microservices, SaaS adoption, IoT, AI, BigData etc.	• Brand knowledge, marketing • Developer commitment • New channel, leads	• Alternative to customer-specific customized integrations • Increasing customer commitment and engagement • APIs add value of traditional products and services for customers	• APIs as "Commodity" • Traditional customer relationship, not a partnership • Payments received are used to cover costs of capacity and API and platform product development	• Growth in customer volumes, getting in to partners' customer networks • Additional value to existing customers • Sharing data and functionality with partners
Typical API consumers = Developer community	Internal consumers: Analytics platforms, CRM/ERP and other business applications, Digital services	Marketing partners For example startups or educational organizations can belong to this category before becoming actual reselling or innovation partners	Innovation partners, customers, partners	Customers to whom we offer APIs as our main products	Innovation partners, resellers, supply chain partners

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

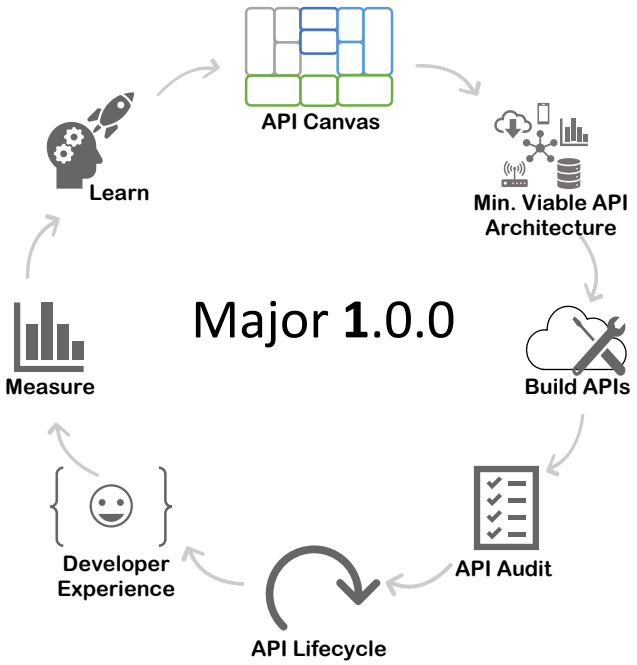
UX for UI design vs. UX for API design



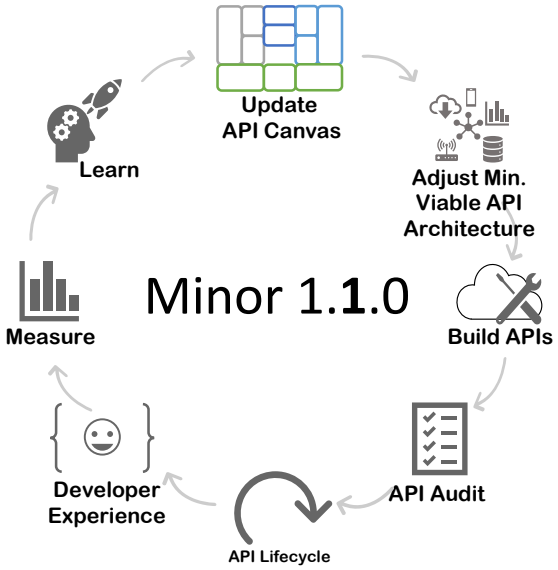
More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

APIOPS Cycle for all changes

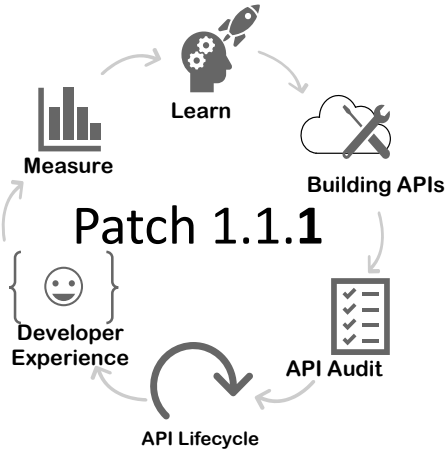
Breaking changes



New features
Depricating endpoints



Fixes to existing requirements



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

API Value Proposition Canvas

API PROVIDER

D **GAIN ENABLING features**

GAINS – benefits for using API B

API CONSUMER

F

API Products and services

API PRODUCT – MARKET FIT

A

TASKS Step by step tasks API consumer needs to achieve

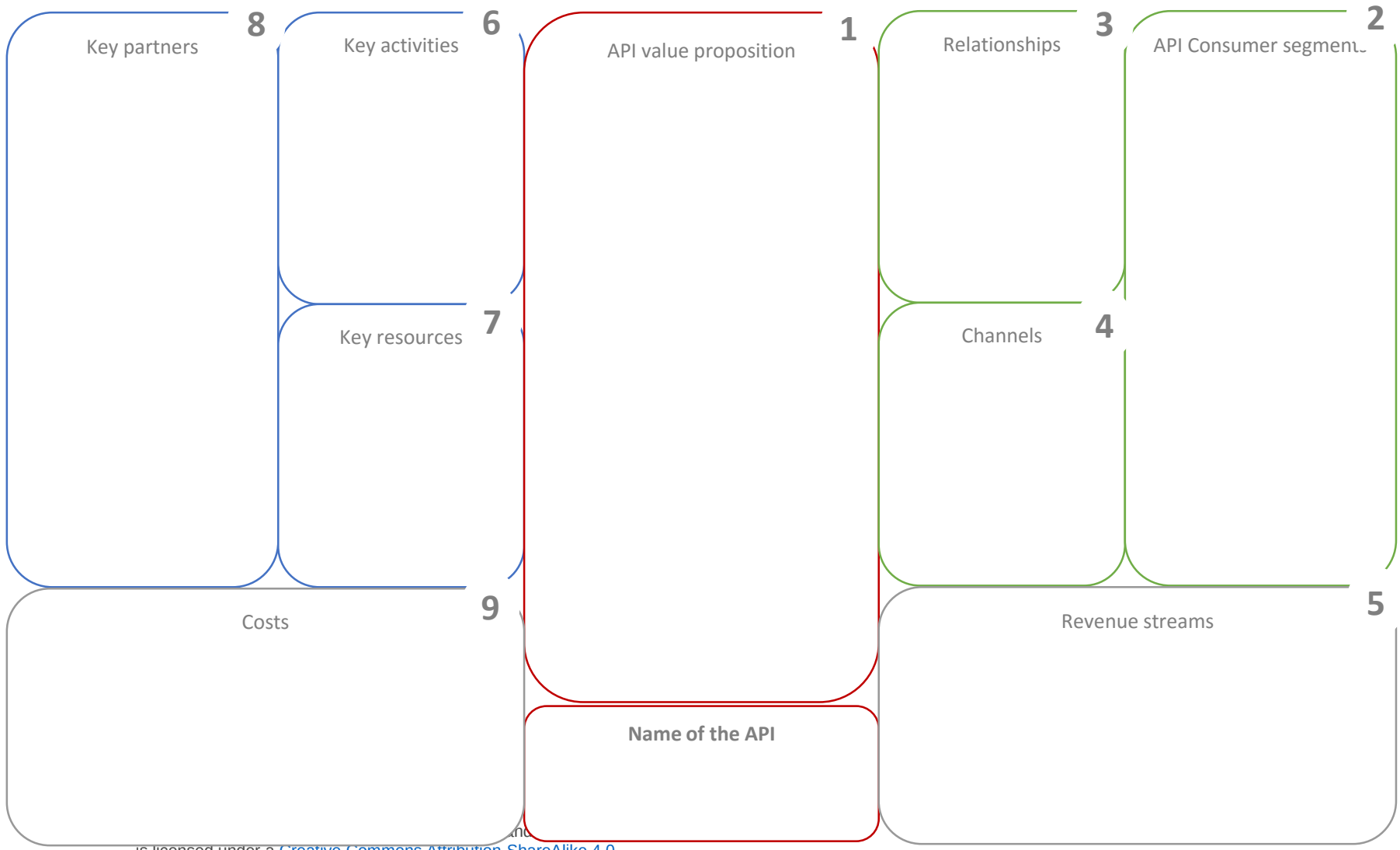
E **PAIN RELEAVER features**

PAINS – problems using API C

More information, feedback and contributions: https://github.com/APIOps/APIOps-cycles

apiops

API Canvas



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

Business impact (risk) mitigating activities

If API becomes unavailable:
impact if the API is not available for 1 minute? 1 hour? 1 day?

ARCHITECTURE RISKS

If API security fails:
3rd party gets access or legitimate API user gets too much access? 3rd 3rd party knows API even exists?

SECURITY RISKS

If API works incorrectly:
if the data is incorrect, missing, too old or too recent or partly working?

QUALITY RISKS

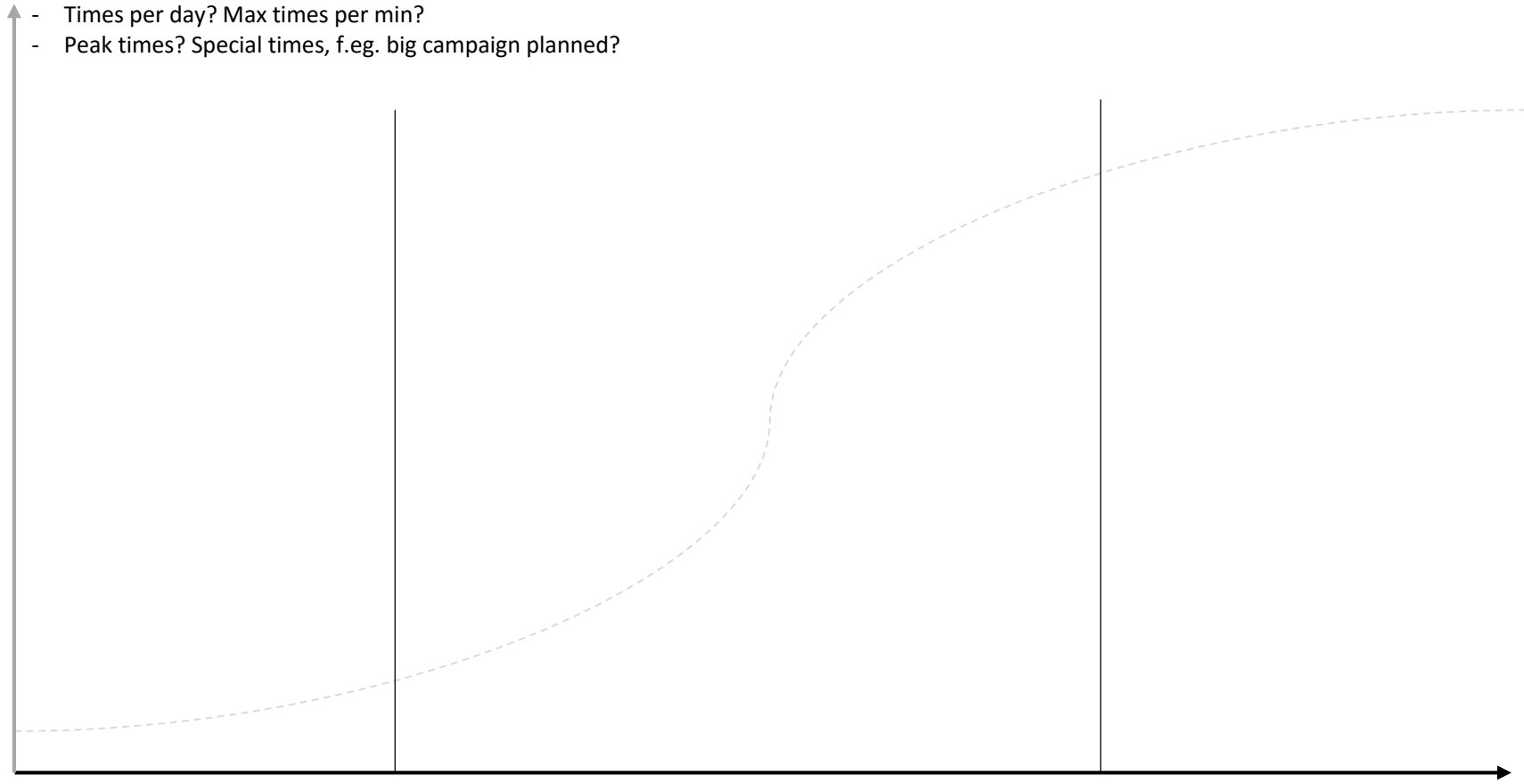
More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

Capacity

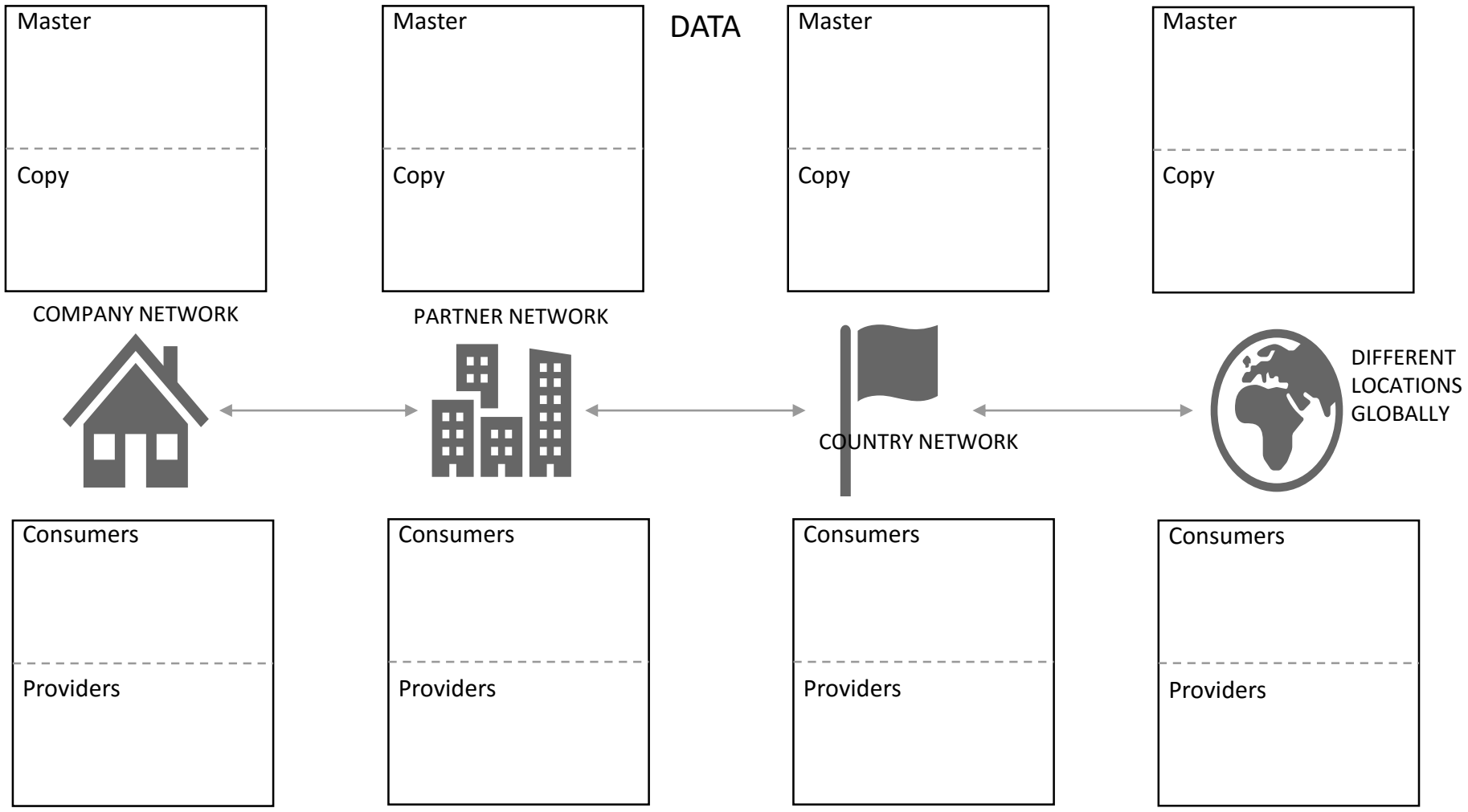
How many business events per day (f.eg. Postal packages sent, orders handled)

- Times per day? Max times per min?
- Peak times? Special times, f.eg. big campaign planned?



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

Locations of data and systems



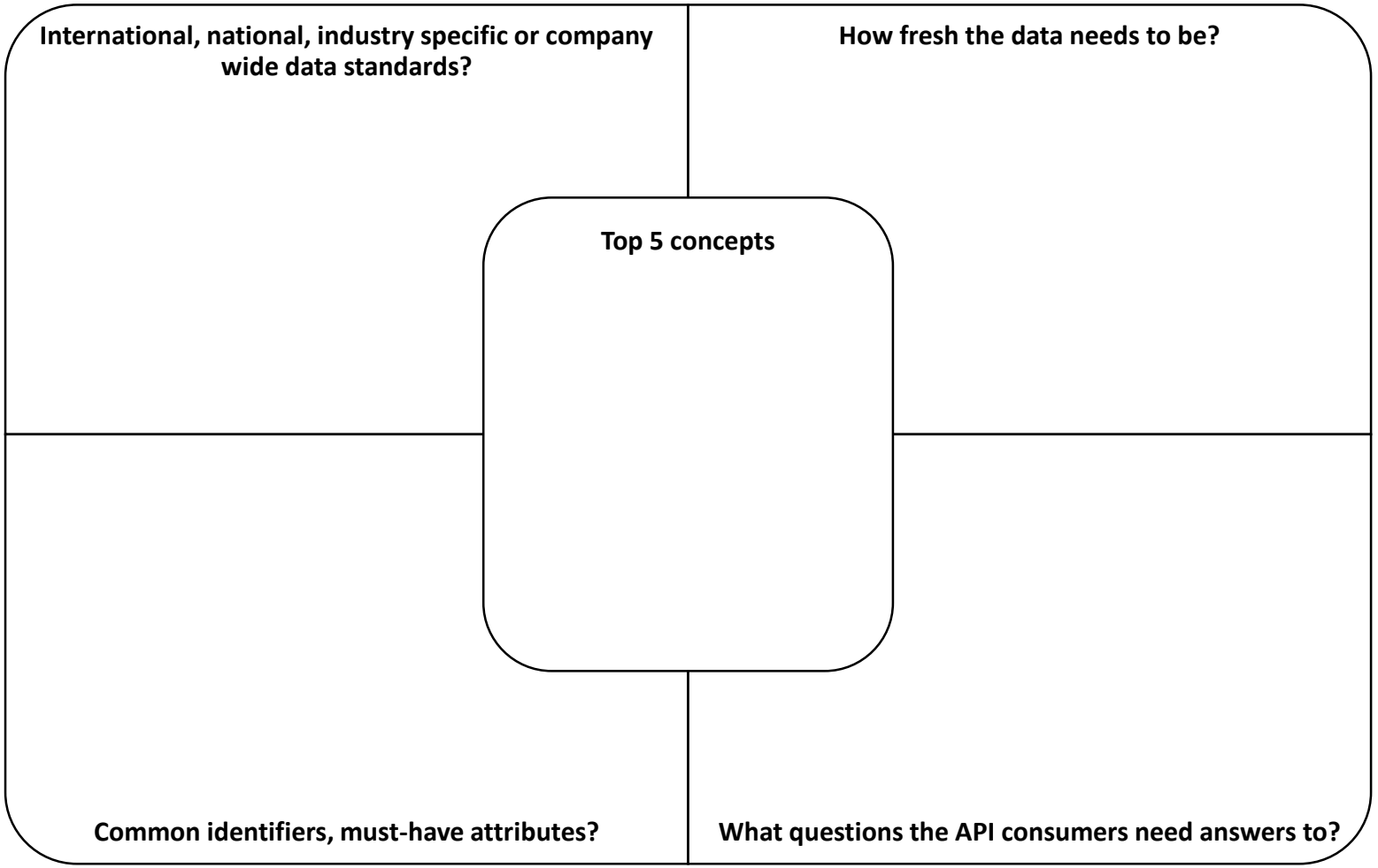
More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

API Consumer interview

Topic	Questions	Answers
Response times	What is the maximum amount of time the API consumer can wait for a response to any request? What is the expected response time for API so they can keep their customers using their system?	
Identity, authentication, authorization	Is there a need to identify users? What are the common identifiers between the API Consumer and the API (email, customer number, social security number)? How are the API consumer's end-users authenticated?	
Data formats	Which data formats the API consumers prefer and can easily process?	
Making requests	Does the API consumer have some technical limitations when calling the API? For example supported HTTP-verbs, headers?	
Handling responses	What kind of responses the API Consumer is able to handle from the API? (Which HTTP response codes supported? Special requirements for errors+)	
Localization and standards	Are there any specific requirements about language, currencies, codes, API specific error codes, error messages and time stamps?	
Encryption	Check if API consumers will be able to handle secure connections with TLS (i.e. HTTPS)? If API handles sensitive encrypted data, verify what encryption algorithms consumers can use?	

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

Data requirements



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

API Design with requests and responses

Scenario name:

API CONSUMER

API PROVIDER

Request

Perform request

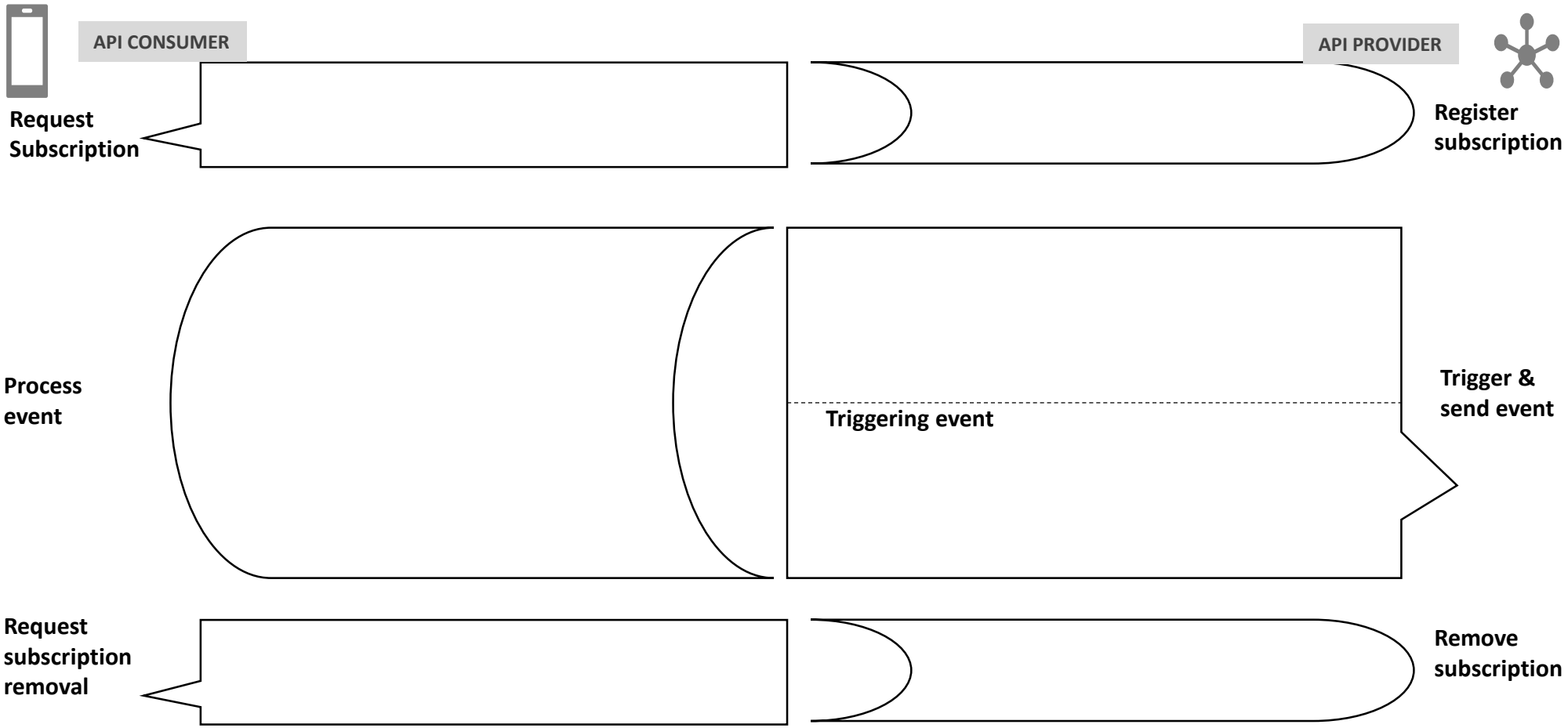
Response value

Send response

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

API Design with events (push)



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

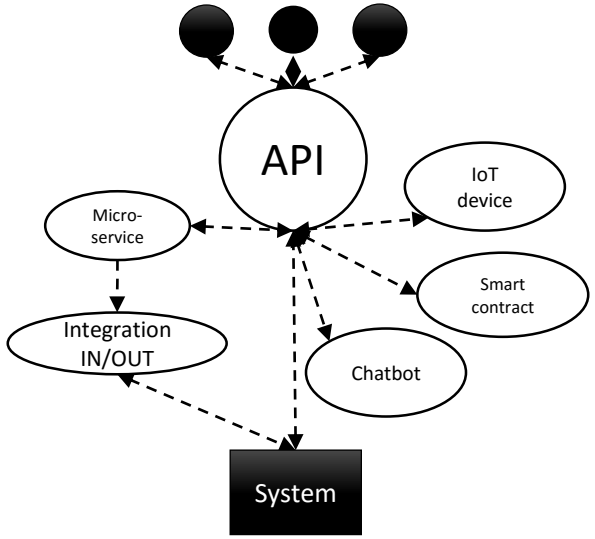
APIs <> integrations

0-N API Consumers
1-N backend services
0-N data storage, can just perform algorithm

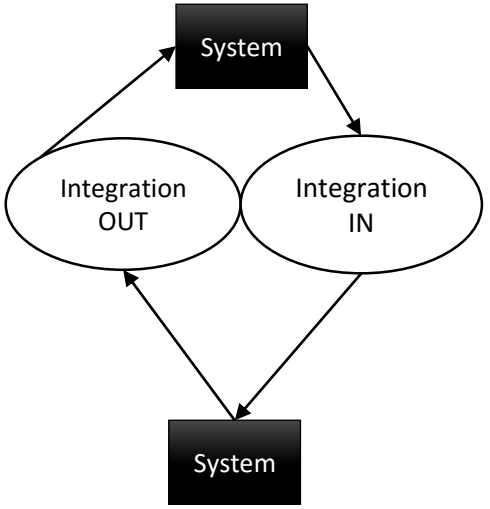
Public internet, HTTP(S)

Mostly insecure consumers, backends and network

Small payloads of data often



1-few integration parties
1 backend
Typically 1+1 data storages



Private networks, TCP etc. protocols

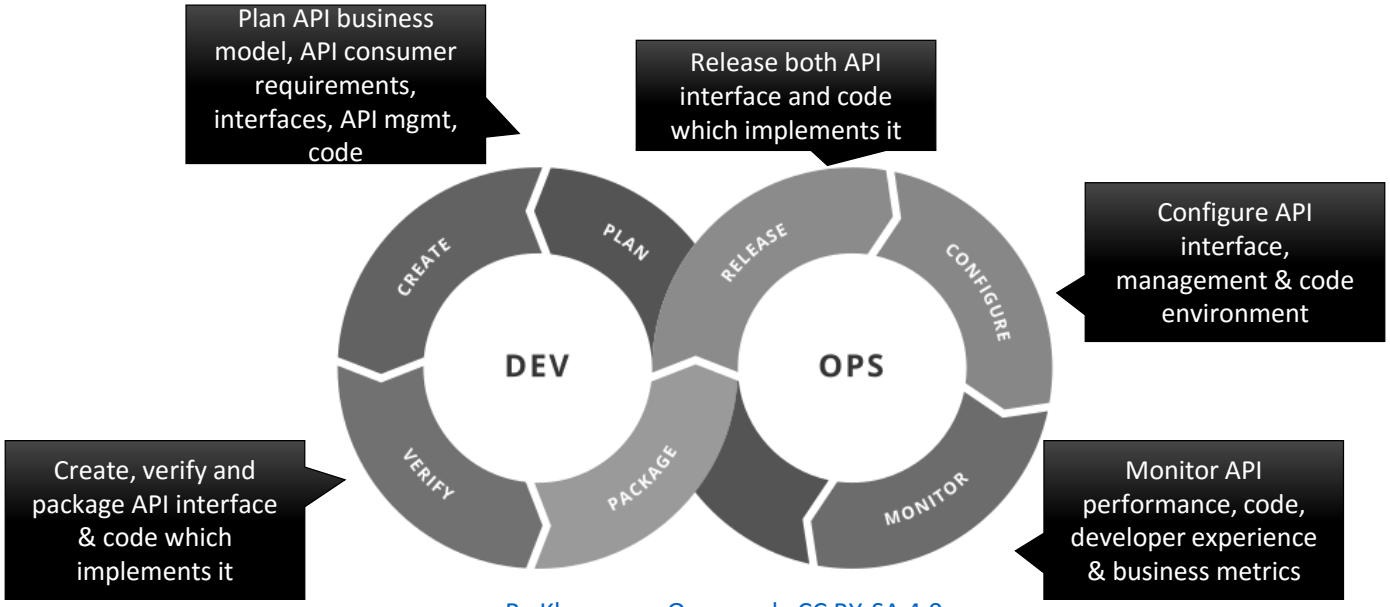
Mostly secure consumers, backends and network

Huge payloads of data at intervals

More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops

APIs need DevOps x2: Interface + Implementation



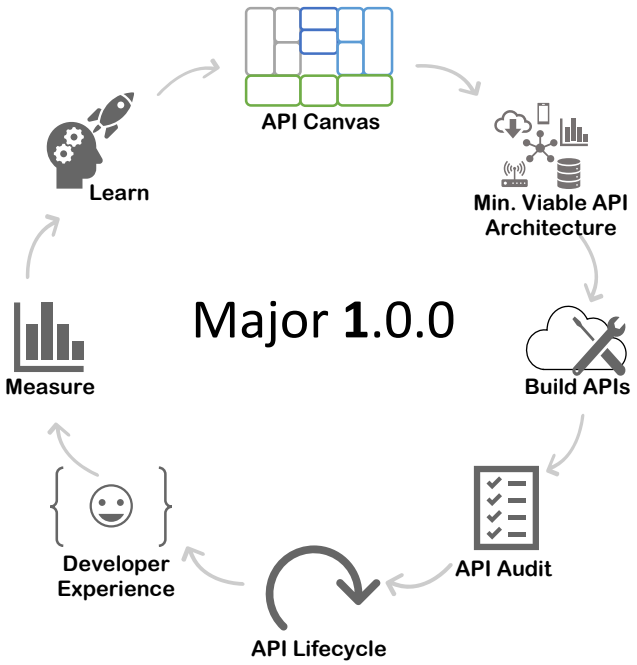
By Kharnagy - Own work, CC BY-SA 4.0

More information, feedback and contributions: https://github.com/APIOps/APIOps-cycles

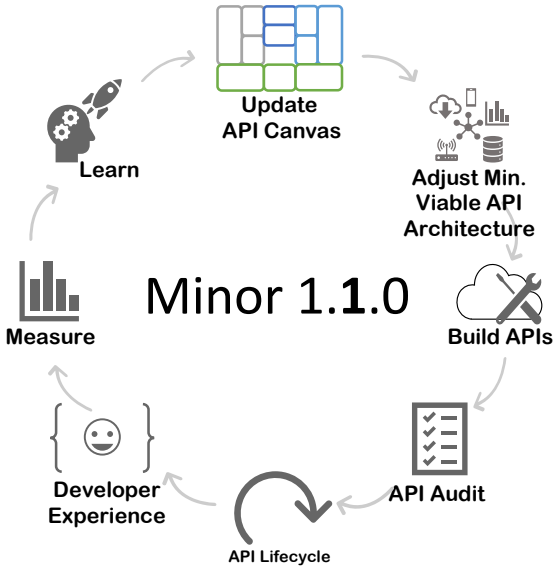
apiops

APIOPS Cycle for all changes

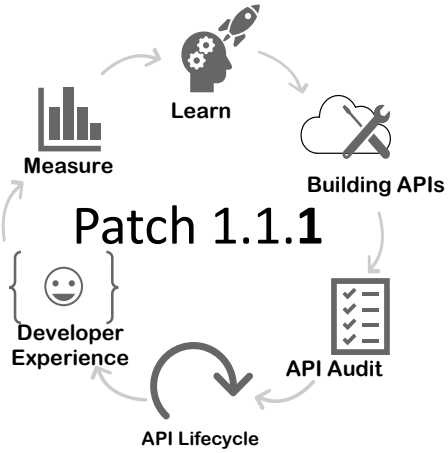
Breaking changes



New features
Depricating endpoints



Fixes to existing requirements



More information, feedback and contributions:
https://github.com/APIOps/APIOps-cycles

apiops