

Open Banking, BBVA API_Market

librecon.io

Noviembre 2018

Índice

- 01** BBVA API_Market
- 02** Stack & Infraestructure
- 03** Microservices





01

BBVA API_Market



01. WHAT IS BBVA API_MARKET?

BBVA API_Market is the portal developed by BBVA to offer API products that will be consumed by third parties.

Our actual catalog consists of:

11 APIs in SPAIN

4 APIs in USA

2 APIs in MEXICO

USA

- Customers
- Accounts
- Cards
- Payments

MEXICO

- Accounts
- Loans Auto

SPAIN

- Customers
- Accounts
- Cards
- Payments
- Loans
- Notifications
- QRPay
- Business Accounts
- Business Notifications
- PayStats
- SEL





Why APIs?

Accessing internal BBVA services is not an easy task, as it is expected. Complications reside in the entry parameters for the service, security, language...



OP3N is born with the idea of opening these internal services to third parties so BBVA and our clients can benefit from their data. Through APIs these services are seamlessly encapsulated in a way that third parties only have to deal with a very easy to understand and integrate layer. Making the APIs system the best one for both BBVA and third parties.

Even so, creating an API that points to a BBVA internal service is not the hardest task. Managing clients, apps, security, make the model scalable, monitoring, access limit and payment management is much more than just creating the APIs.

How do we manage the APIs?

BBVA Spain

Retail client



Accounts



Customers



Cards



QRPay



Loans



Funds



Payments



Notifications

Business client



Business Accounts

Data



PayStats



SEL



Integrated console



A developer that is not logged in could launch any API call using test environment and it'll response hardcoded data, a useful way to know what data is available



The screenshot displays the API Market integrated console interface. The breadcrumb navigation at the top reads: `api_market > Products > PayStats > Documentation`. The main content area is titled "Overview" and shows the selected endpoint `/info/merchants_categories` with a "GET" button. The response pane on the right shows the following details:

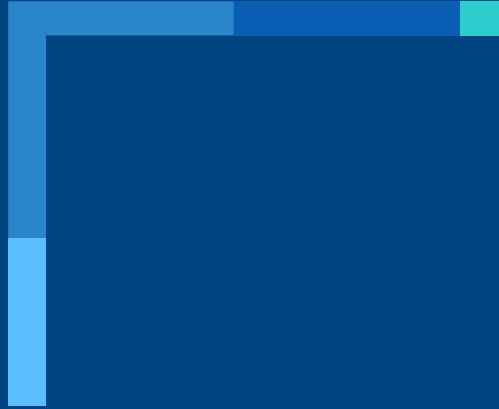
```

RESPONSE

STATUS
200

HEADERS
date Thu, 01 Mar 2018 18:38:51 GMT
content-length 1676
pragma no-cache, no-cache
content-type application/json; charset=utf-8

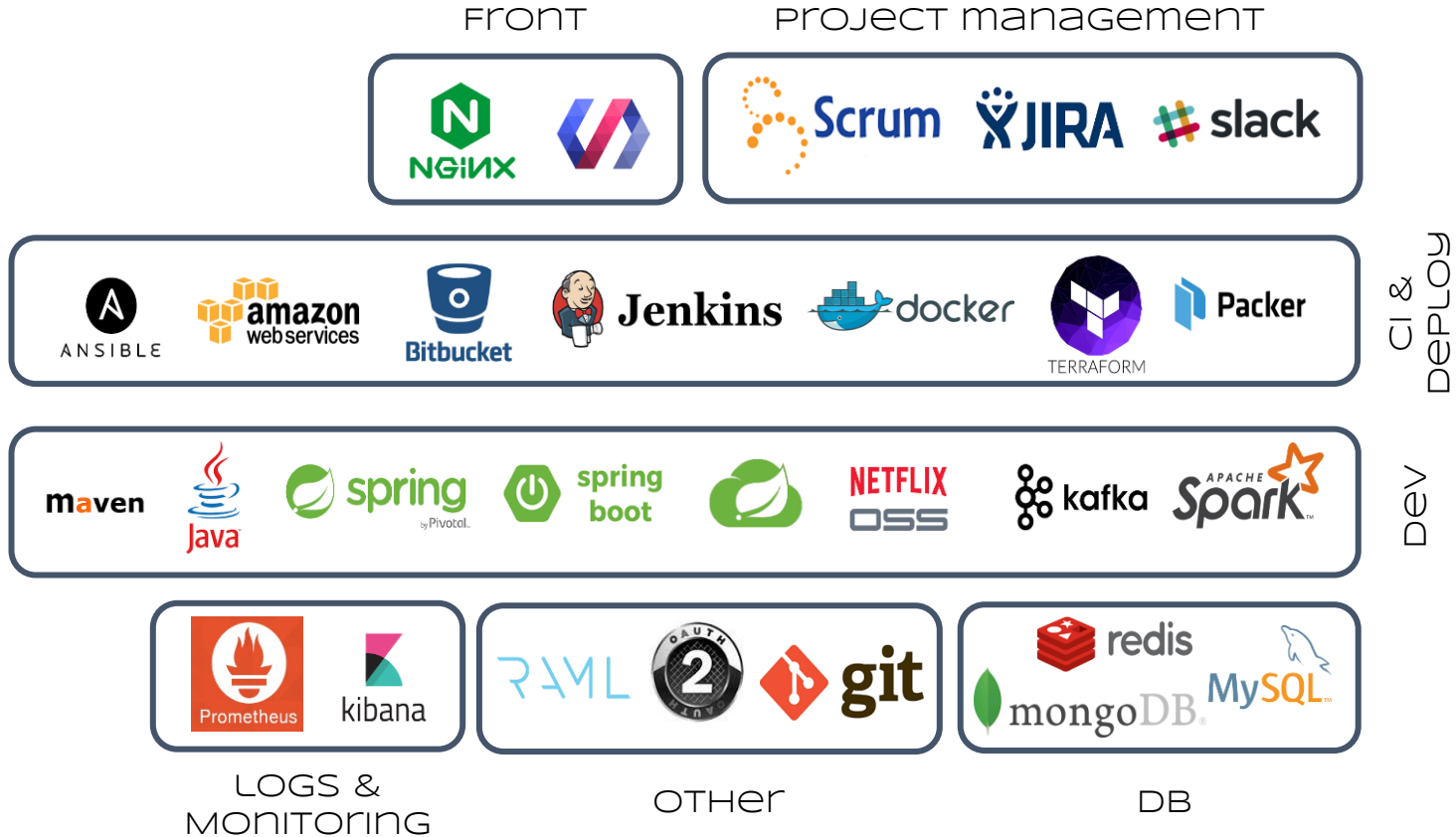
BODY
{
  "result": {
    "code": 200,
    "info": "OK"
  },
  "data": [
    {
      "size": "18",
      "categories": [
        {
          "code": "es.auto",
          "description": "Automoción",
          "subcategorias": [
            {
              "code": "es.car",
              "description": "Coche"
            }
          ]
        }
      ]
    }
  ]
}
    
```



02

BBVA API_Market Infrastructure

Stack tecnológico



Product development & architecture principles



Idea or Business opportunity

APIs as a product ↩

Code ↩

Microservice ↩

Container ↩



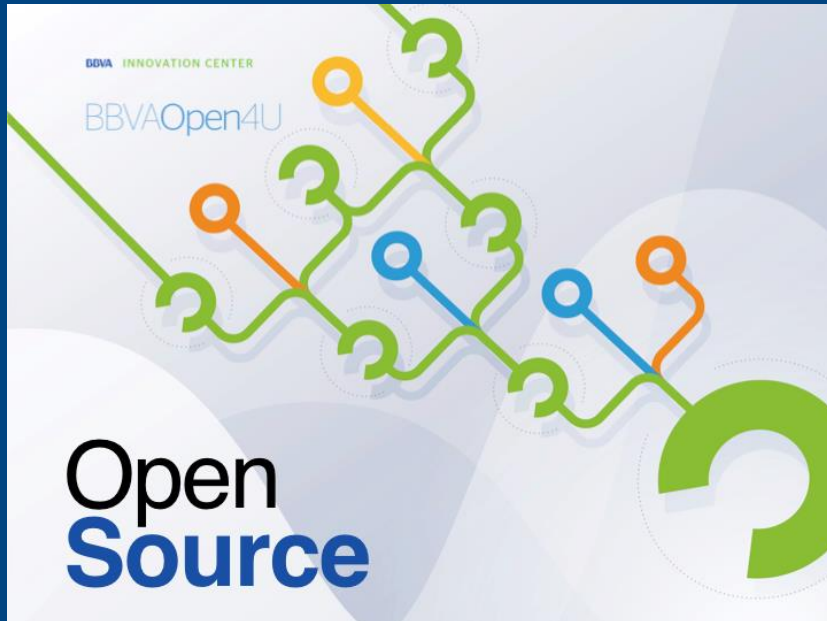
- ★ Microservices are built around business capabilities and independently deployable by fully automated deployment machinery.
- ★ Containers for reliable deployments.
- ★ PaaS-based architecture model by leveraging Open Source technologies and AWS components: EC2, **ECS**, RDS, ElasticCache, etc.

Switching to microservices architecture...

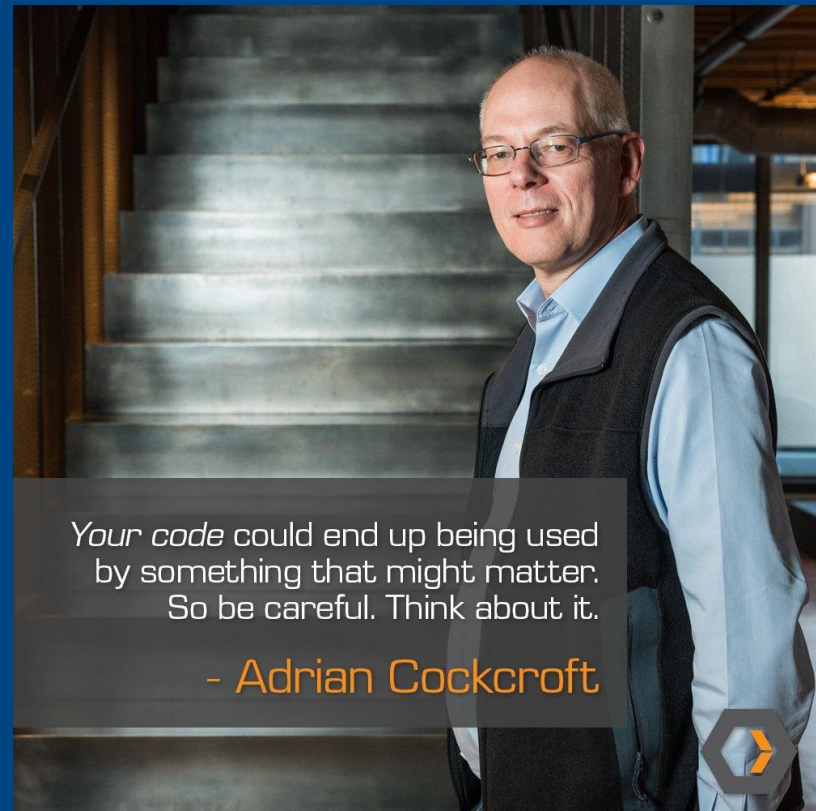


... without figuring out the platform and the dependencies





BBVAOPEN4U.com



Your code could end up being used by something that might matter. So be careful. Think about it.

- Adrian Cockcroft

<https://twitter.com/adrianco/status/705637183156125696>

Gross numbers: EC2 vs. ECS

$$\begin{array}{rclcl}
 80 & \times & 4 & + & 80 & = & 400 & \text{instances (minimum)} \\
 \text{services} & & \text{environments} & & \text{services for HA} & & & \\
 & & \text{(dev, testing,} & & \text{tolerance (AZ) \&} & & & \\
 & & \text{staging, production)} & & \text{scale-up on} & & & \\
 & & & & \text{demand} & & &
 \end{array}$$

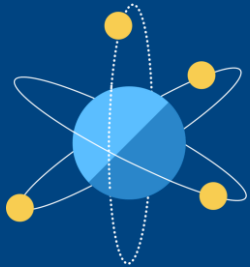
1:16

~ **400** EC2 small instances

vs.

4 ECS clusters with 25 EC2 large instances)

Why Amazon ECS (and not Kubernetes)

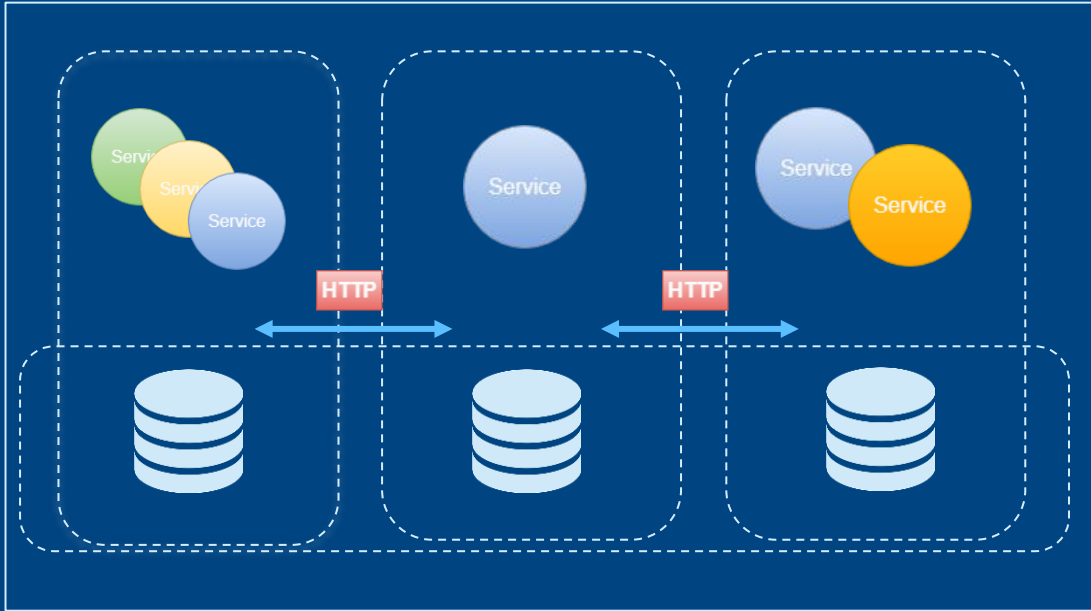


- ★ Amazon EC2 Container Service (ECS) is a fully managed container management service.
- ★ Service scales as microservice architecture grows.
- ★ Shared state, optimistic concurrency system that provides flexible scheduling capabilities for your tasks and containers.
- ★ Integration with Amazon CloudWatch for monitoring and logging.
- ★ Integration with CI/CD services: Amazon ECS lets you launch and stop container-based applications with simple API call.



Requirements for Microservices Implementation

GOVERNANCE (BUSINESS)

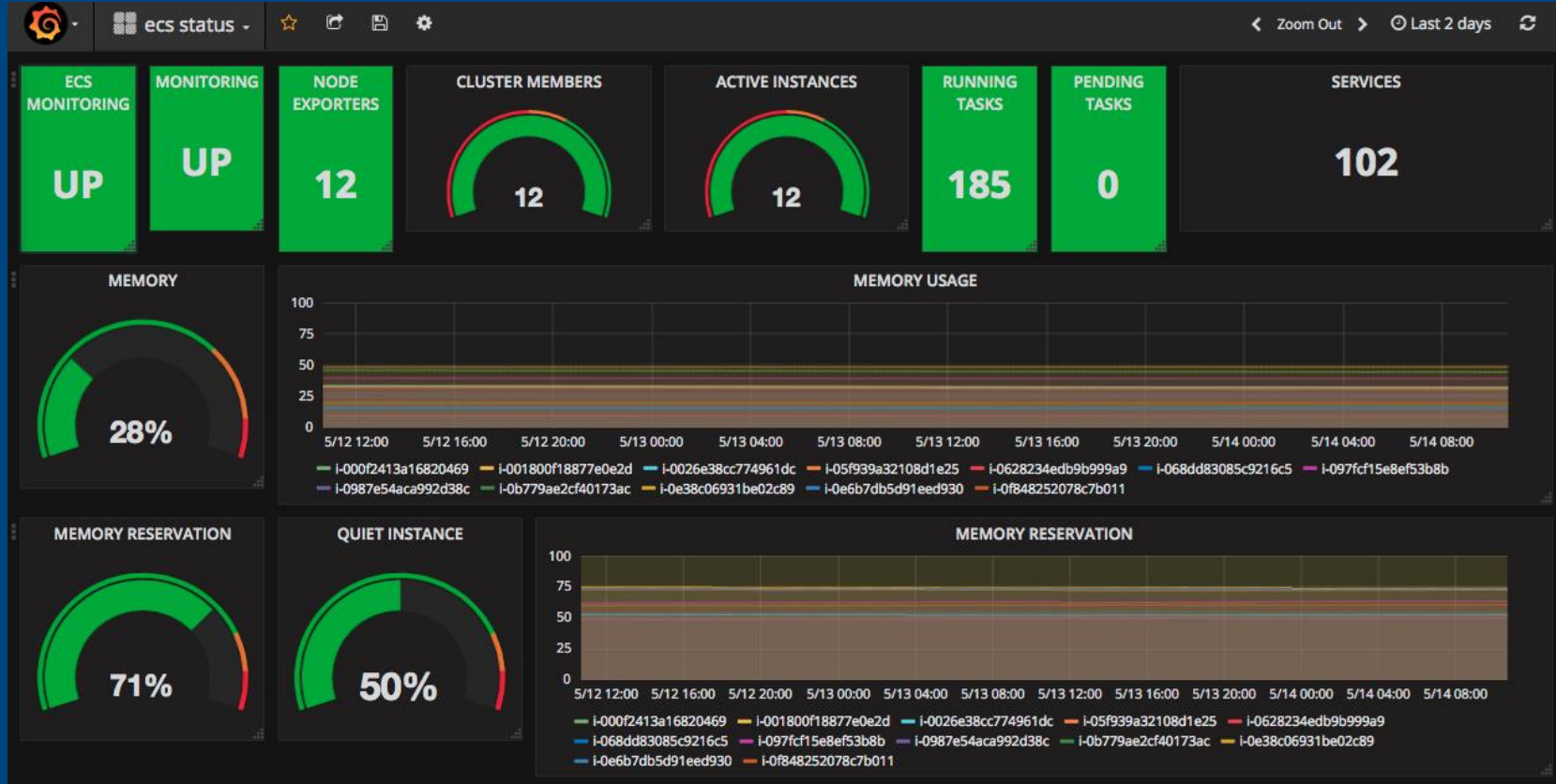


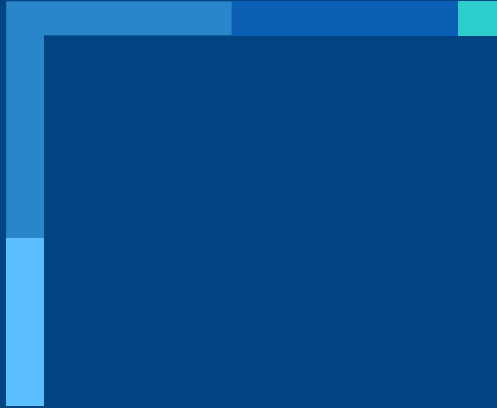
GOVERNANCE (IT)

P
L
A
T
F
O
R
M

Load Balancing	AWS ALB
HA Tolerance, Scaling	Task-placement strategies, Multi-AZ & CloudWatch Events
Provisioning	CI/CD Integration with ECS API
Service Discovery	ECS Service Discovery
Security	ECS Security Group, CloudTrail, VPC Flow Logs
Health	ECS Docker Health Checks + ALB Health
Monitoring	CloudWatch Events + Grafana
Logging	S3, CWL Logs, Lambda & AWS ElasticSearch

Monitoring - Third party tools out-of-the-box integration with CloudWatch API

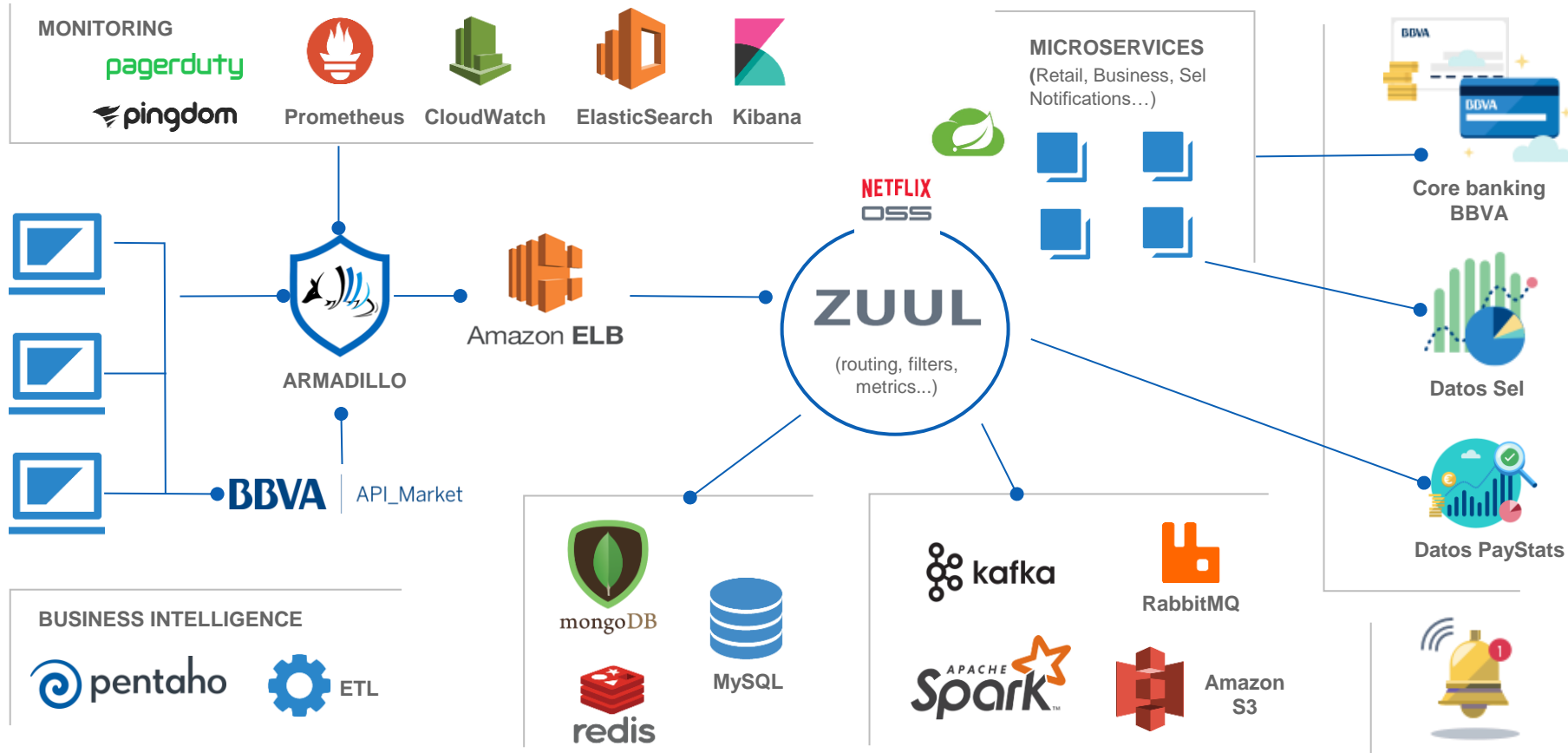




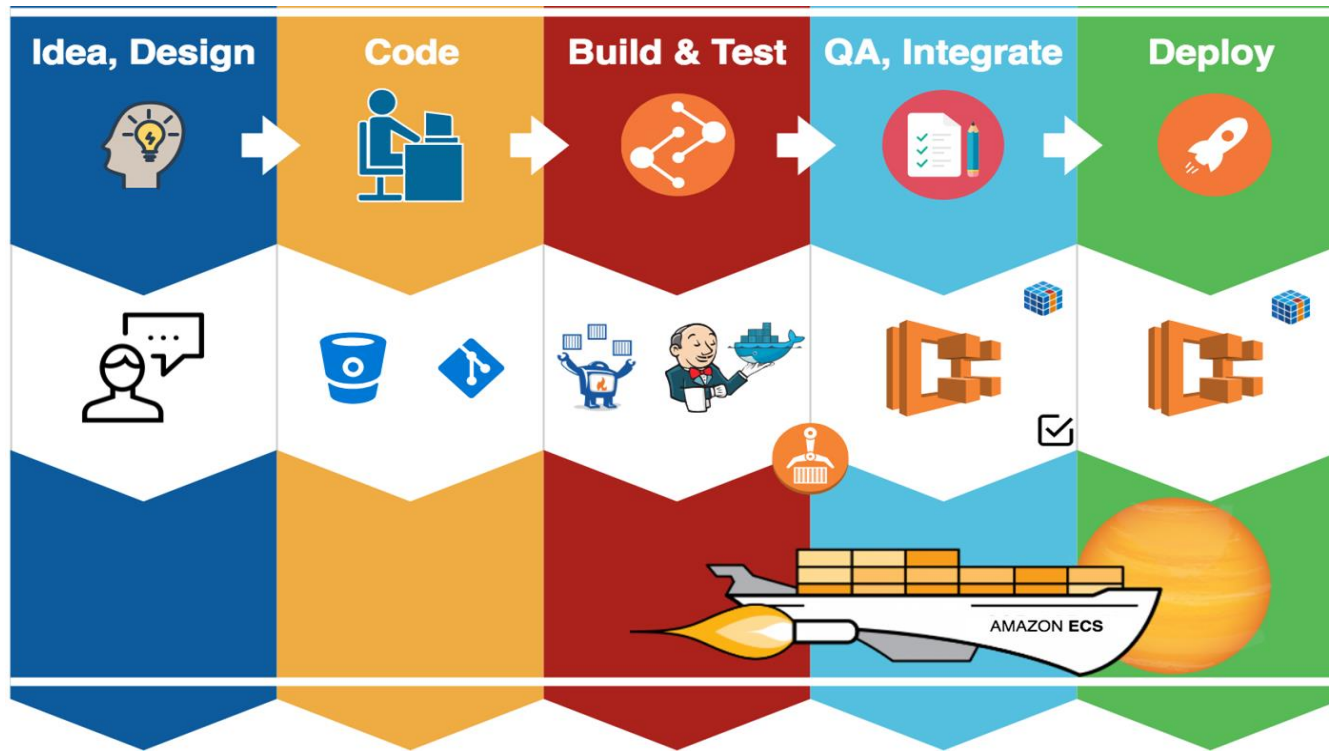
03

Microservices

BBVA OP3N Architecture



Continuous delivery pipeline



Conway's Law

In every organization there will
always be one person who
knows what is going on.

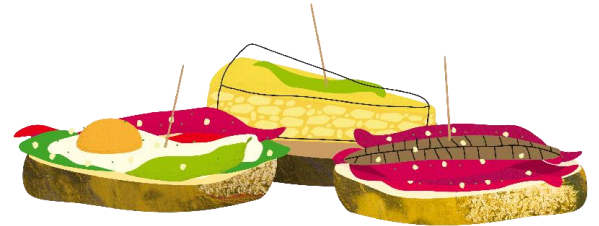
This person must be fired !!



Two Pizza Team

a.k.a

“Pintxo Team”

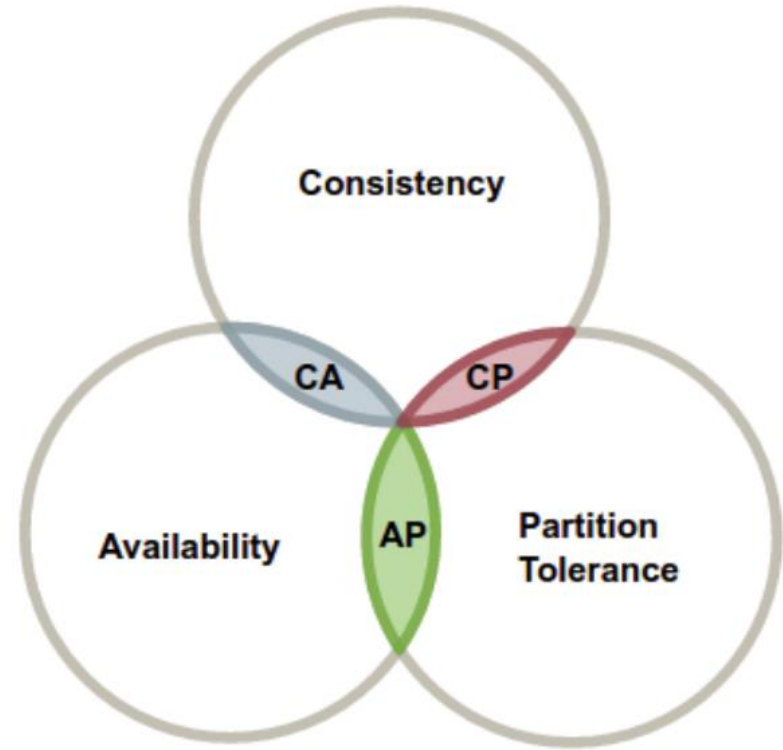




As small as possible

but as big as necessary...

CAP Theorem



Domain Driven Design

Testing

Monitoring

Circuit breaker

Saga pattern

Audit pattern

Event driven

Fully asynchronous

Deployment



Command Query Responsibility Segregation

¿Preguntas?

Francisco Froufe

 froufe

Thomas Person

 personthomas

api.market.support@bbva.com

www.bbvaapimarket.com