IT Architecture:
From SOA To Microservices

</*>

Stefano Priola CTO

Roma ,December 17, 2019

centr/co :

Agenda

- Why?
- Culture
- Technology
- Principles & Practices

centr/co:

We enable banks and fintechs to innovate

thanks to our proven open banking system,

that is easy to integrate and customise,

and minimises time to market

Who is Centrico?

- Operational and IT backbone of Sella Group since 1968
- Autonomous company since June 26th 2018 (Net equity €23m)
- «Società strumentale» authorized by Bankit since December 30th 2018, in operations since March 1st 2019
- IT and BPO full outsourcing supported by 500 HC in Italy, 400 HC in Romania and 300 HC in India (800 IT and 400 BPO)
- 2 global in house centers (GICs) that offer direct access of useful, specialized and talented resources in India and Romania

IT Core banking LDS and Metrix

Metrics	Complexity	Service Levels
>3 milions of accounts	>350 locations in 3 countries	RTO 12h
>2 milions of transactions/ day	>5.500 users	RPO 0
>7 milions of API calls/ day		
>3,000 Tb of storage	>€36bn of Assets	99.987% availability (2013-2018)
>1.700 change request every year	>1.700 server in private cloud	
>120.000 POS	>8.000 devices managed	

BPO already manages important volumes

Services	Metrics
Complete back office operations and document management	> 40Mln tickets/year (data entry – processing – digitazion– archiviation)
Credit Practices issued	>28k/year
Inbound call, chat e email	>1Mln contacts per year
Robot Process Automation	169 flows managed by Robots

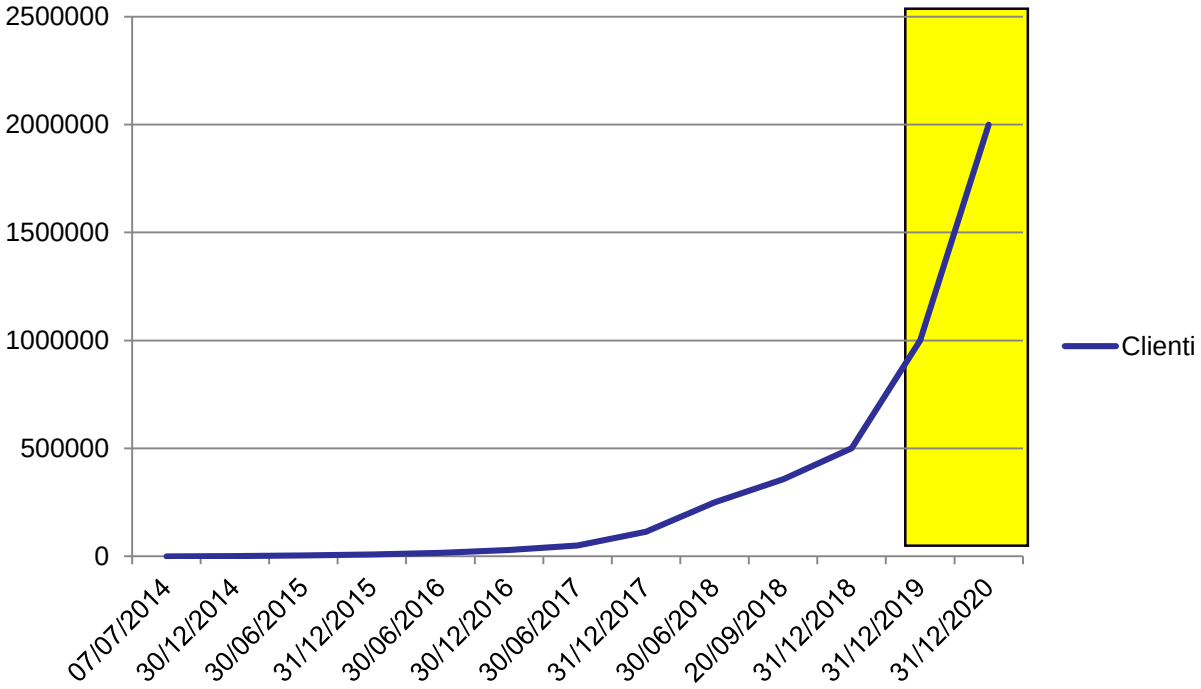
Centrico IT System reliable and flexible

- 80% of the modules developed internally, mainly in java (java ee) with verticalisation in .NET
- Cobit, Itil, Togaf, Agile Scrum
- Native, 24 x 7, continuous Integration, continuous Delivery
- API exposure to services, container ready
- 2 proprietary data centers, infrastructure automation ready
- virtual team components CERTfin Italy

Why Architectural Change?

centrico:

Why?



Hype customers trend (forecasting in yellow)

centrico

Why?

- **Speed and simplicity**. time to have a server, who I have to ask what?,..
- **Capacity to scale**
- **Minimization of risk**. Fewer failures, Faster recovery.
- **Increased efficiency in software development.** Less time doing rework. More time doing new work.
- **Cost savings**. more servers managed per person
- **Improve security**. Prove compliance.
- …
- **More employee recommendations as a great place to work**

centr/co:

Culture

{centrico:

Communication Skill

The TOGAF® Standard — Version 9.2

Level	Achievement	Description
1	Background	Not a required skill, though should be able to define and manage skill if required.
2	Awareness	Understands the background, issues, and implications sufficiently to be able to understand how to proceed further and advise client accordingly.
3	Knowledge	Detailed knowledge of subject area and capable of providing professional advice and guidance. Ability to integrate capability into architecture design.
4	Expert	Extensive and substantial practical experience and applied knowledge on the subject.

Roles	Architecture Board Member	Architecture Sponsor	Enterprise Architecture Manager	Enterprise Architecture Technology	Enterprise Architecture Data	Enterprise Architecture Applications	Enterprise Architecture Business	Program/ Project Manager	IT Designer
Generic Skills									
Leadership	4	4	4	3	3	3	3	4	1
Teamwork	3	3	4	4	4	4	4	4	2
Inter-personal	4	4	4	4	4	4	4	4	2
Oral Communications	3	3	4	4	4	4	4	4	2
Written Communications	3	3	4	4	4	4	4	3	3
Logical Analysis	2	2	4	4	4	4	4	3	3
Stakeholder Management	4	3	4	3	3	3	3	4	2
Risk Management	3	3	4	3	3	3	3	4	1

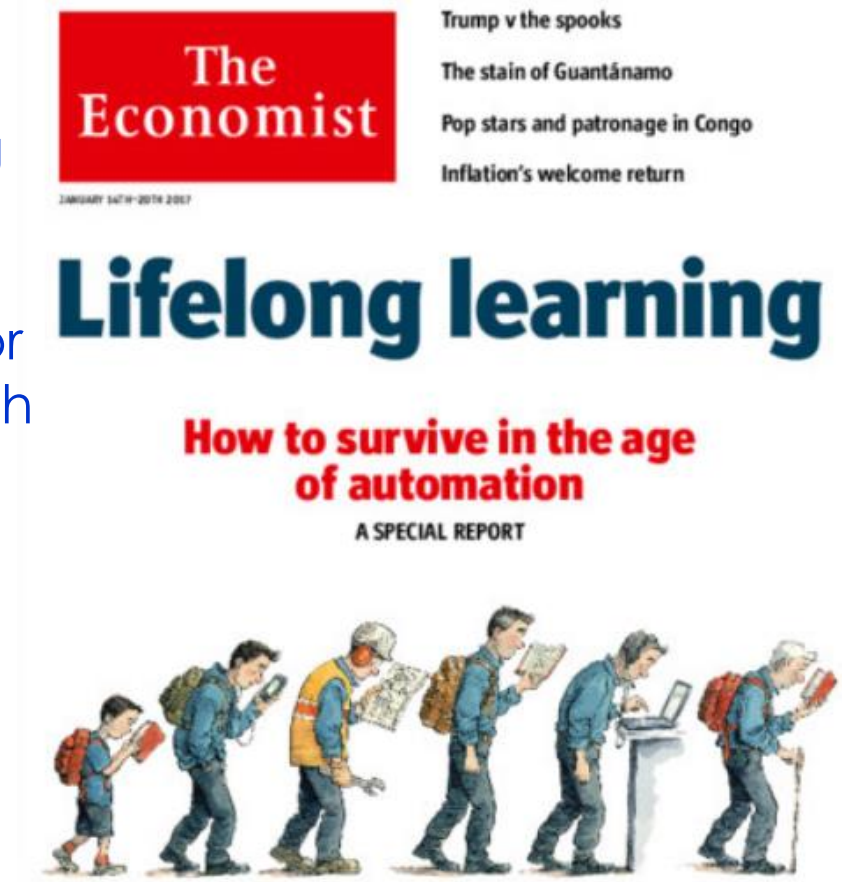
http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap46.html

centrico

Formazione Continua e Apertura mentale

Lifelong learning n. a form of or approach to education which promotes the **continuation of learning throughout adult life**, esp. by making educational material and instruction available through libraries, colleges, or information technology [Oxford English Dictionary]

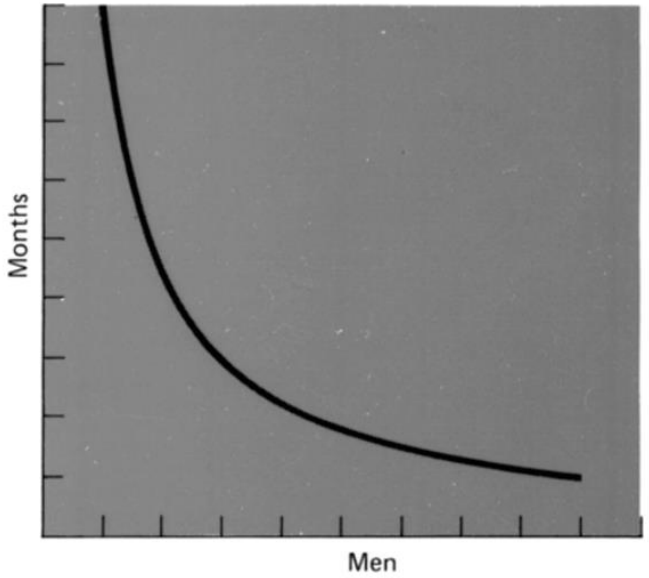
Continues learning is your **self-motivated persistence in acquiring knowledge and competencies** in order to expand your skill set and develop future opportunities. It forms part of your personal and professional development in an effort to avoid stagnation and reach your full potential.



coursera edX UDACITY kadenze centr/co:

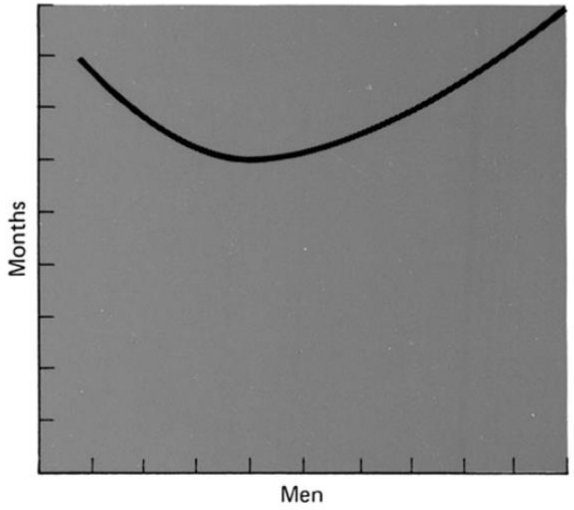
Two Pizzas Teams

Figure 2.1. Time versus number of workers—perfectly partitionable task



Source:
The Mythical Man-Month
1975 Fred Brooks

Figure 2.4. Time versus number of workers—task with complex interrelationships



centrico:

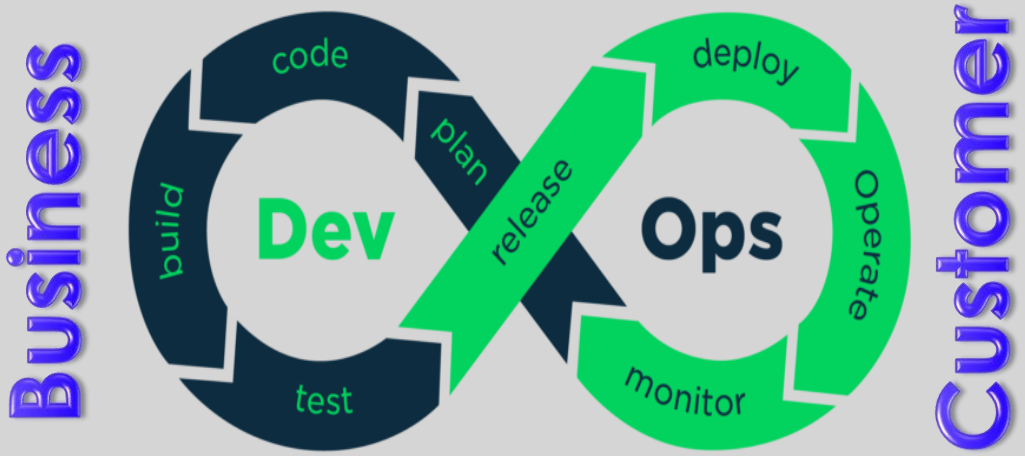
DEVOPS

DevOps is the **combination of cultural philosophies, practices, and tools** that increases an organization's ability to deliver applications and services at high velocity

Under a DevOps model, development and operations teams are no longer "siloed." Sometimes, these two teams are merged into a single team (agile team).

DevOps emphasizes behavioral- or cultural-related changes such as those which encourage teaming, inclusion, feedback, and experimentation.

Developers come from a mindset where change is what they're paid to accomplish. The **business** depends on them to **respond to changing** needs



Operations is naturally motivated to resist change, because it undermines **stability and reliability.** **They are the most close to Customer**

Technology

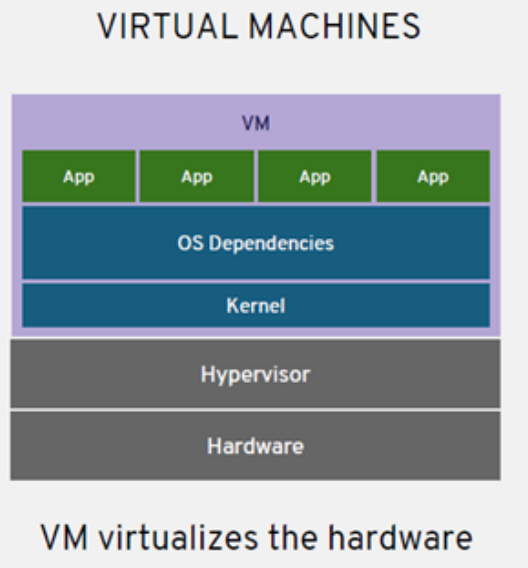
Container

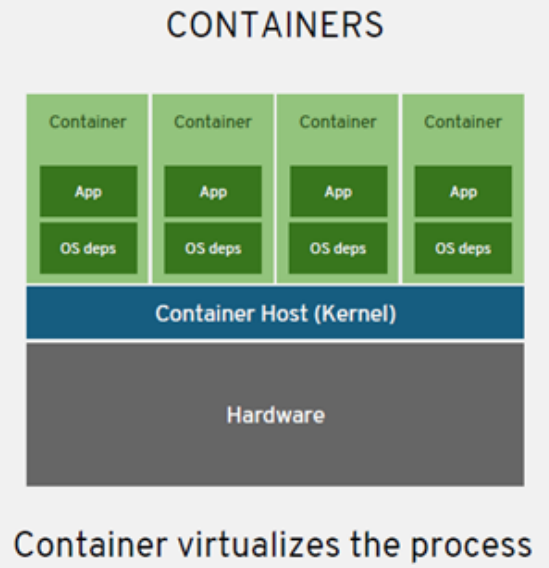
Containers are the **runtime representation of a packaging format** based on a lightweight, immutable **image**.
Runtime dependencies are resolved within the image which facilitates portability
An important corollary is that launching a new workload does not incur the cost of provisioning new compute infrastructure.

Container orchestration involves **the lifecycle management of container workloads,** including functions such as to schedule, stop, start, and replicate across a cluster of machines. Compute resources for running workloads are abstracted, allowing the host infrastructure to be treated as a single logical deployment target. Kubernetes is an open source community project addressing container orchestration.

centrico

Container VS Virtual Machine



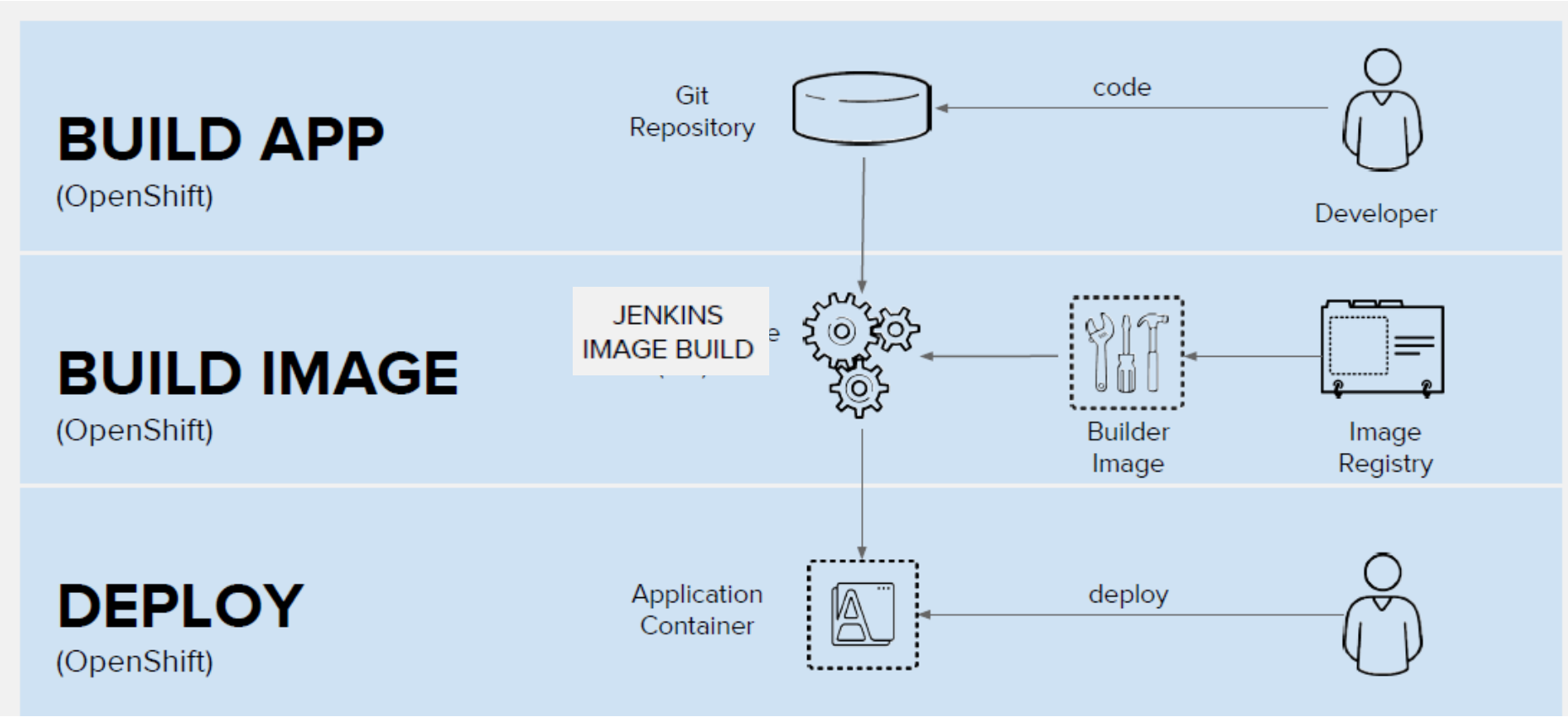


- ✚ VM Isolation
- ➖ Complete OS
- ➖ Static Compute
- ➖ Static Memory
- ➖ High Resource Usage

- ✚ Container Isolation
- ✚ Shared Kernel
- ✚ Burstable Compute
- ✚ Burstable Memory
- ✚ Low Resource Usage
- ✚ Container Portability
- ✚ Automatic handling

centr/co:

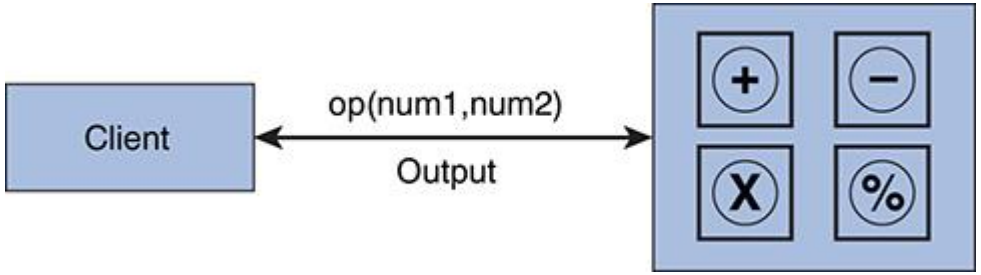
Build & Deploy Containers



centrico:

Service Oriented Architecture

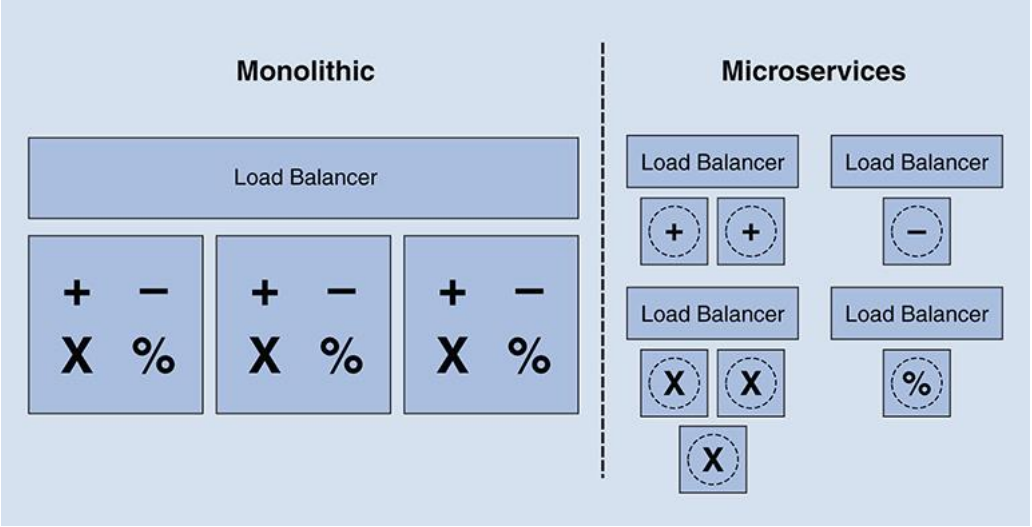
Service Oriented Architecture (SOA): is an approach to service design, of application software solutions, oriented to respond effectively and efficiently to business demands.
Software packages and libraries are being developed as a collection of services
The business should be able to understand easily the SOA contract of a service..



centrico

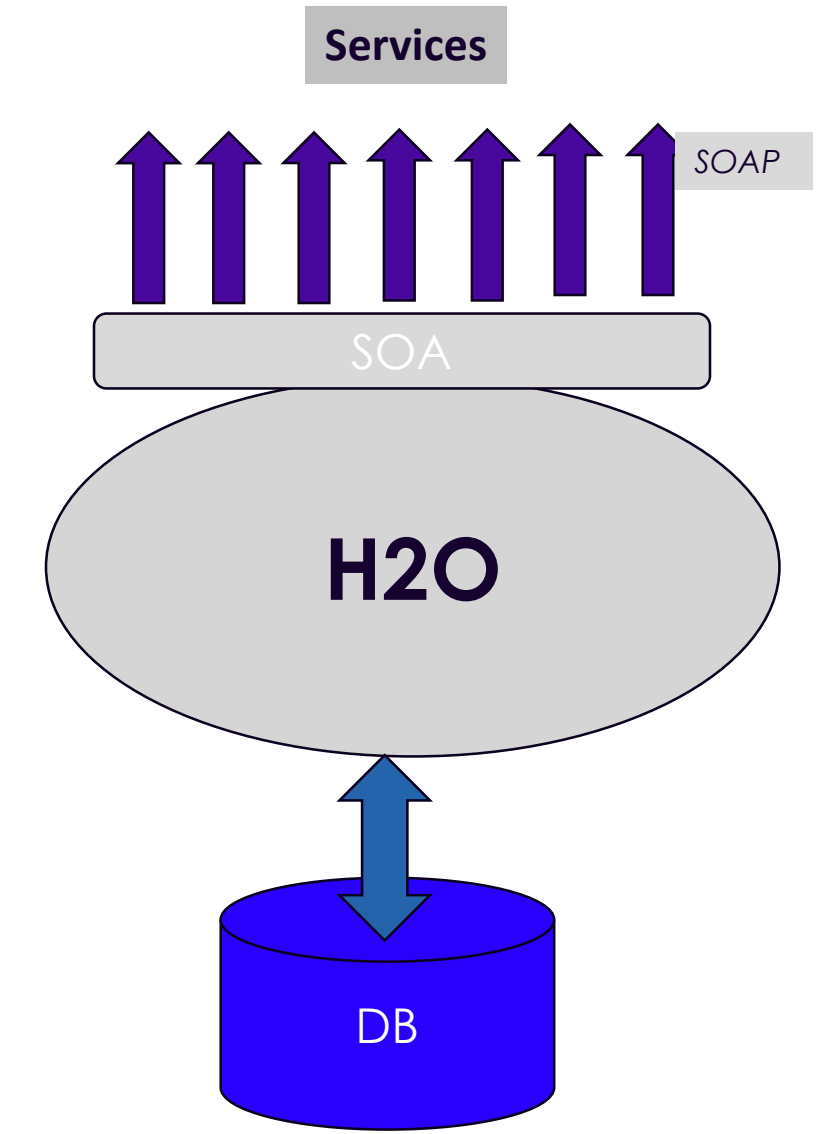
Microservices

- [Lewis/Fowler] the microservice architectural style is an **approach to developing a single application as a suite of small services**, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API.
- These services are built around business capabilities and independently deployable by fully **automated deployment** machinery.
- A microservice is (generally) a single process focused on one aspect of the application, operating in **isolation** as much as possible.

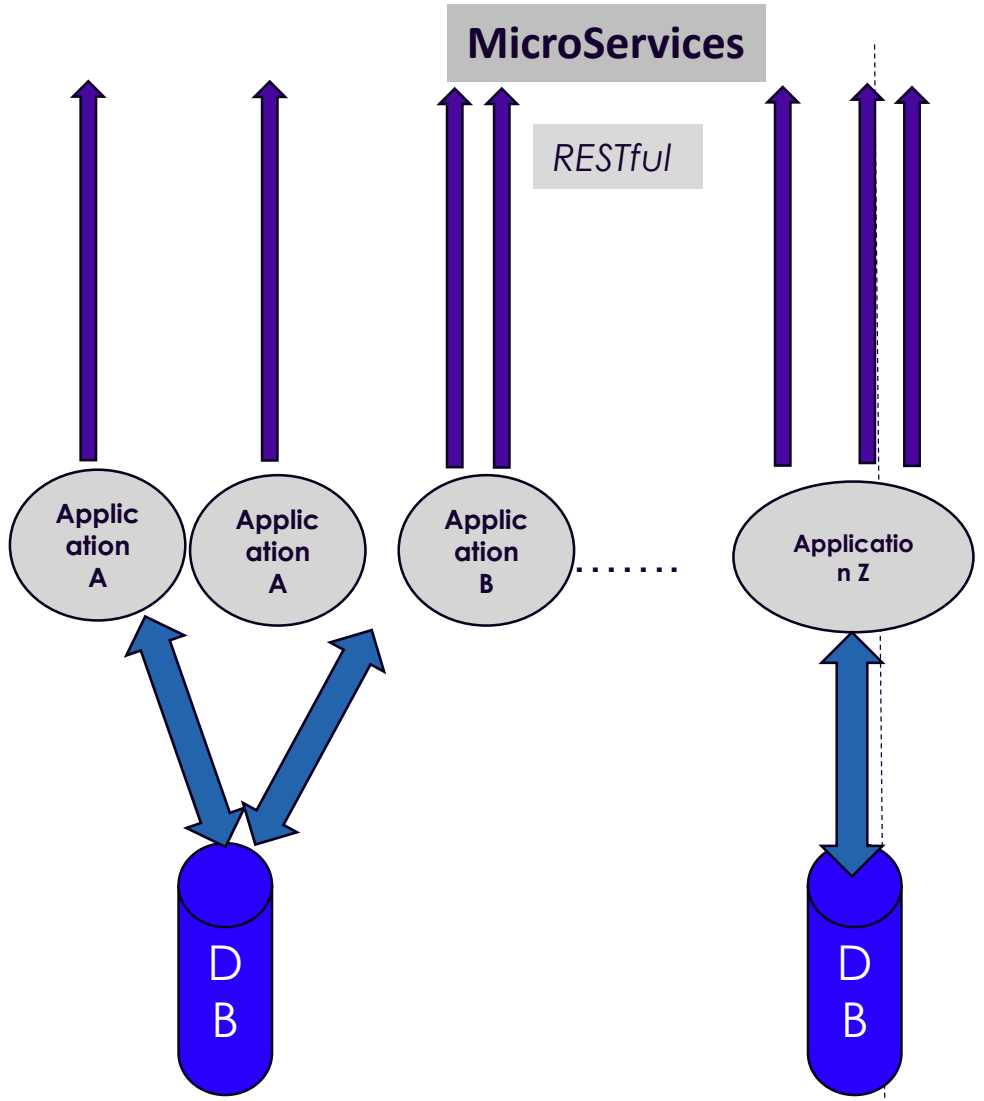


centr/co:

SOA vs Microservices



SOA services based on single java ear monolith



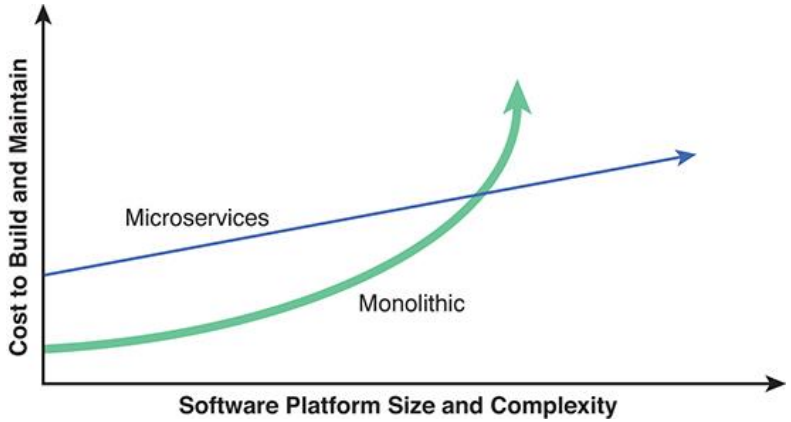
Applications use different tables or schemas if not different istances

centrico

Microservices

Advantages:

- Simplicity: Each **microservice performs only one distinct** and well-defined function, so there is less code to take care of, less cohesion and dependency within the code, and a lower probability of bugs.
- **Scalability**
- Continuous delivery
- More freedom and fewer dependencies
- **Fault isolation**
- **Data segregation and decentralization**

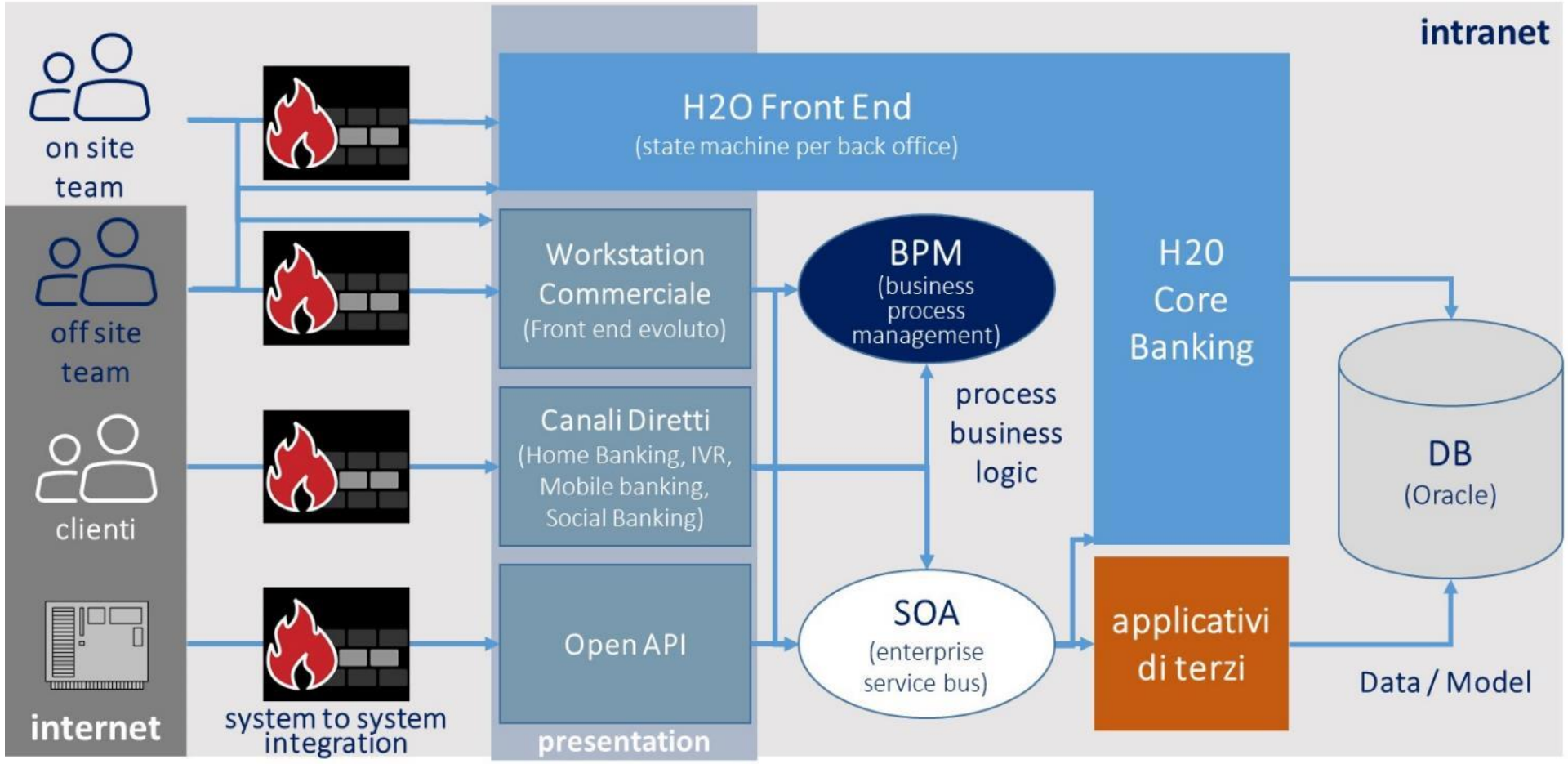


Disadvantages

- Troubleshooting complexity
  - If an end user reports a problem such as slow performance or timeouts, where do I start my troubleshooting?
- Increased latency Intraprocess communication (like the kind used in monolithic applications) is much faster than the interprocess communication used by microservices.
- Operational complexity several hundreds to thousands of microservices in a real-world application
- Version control

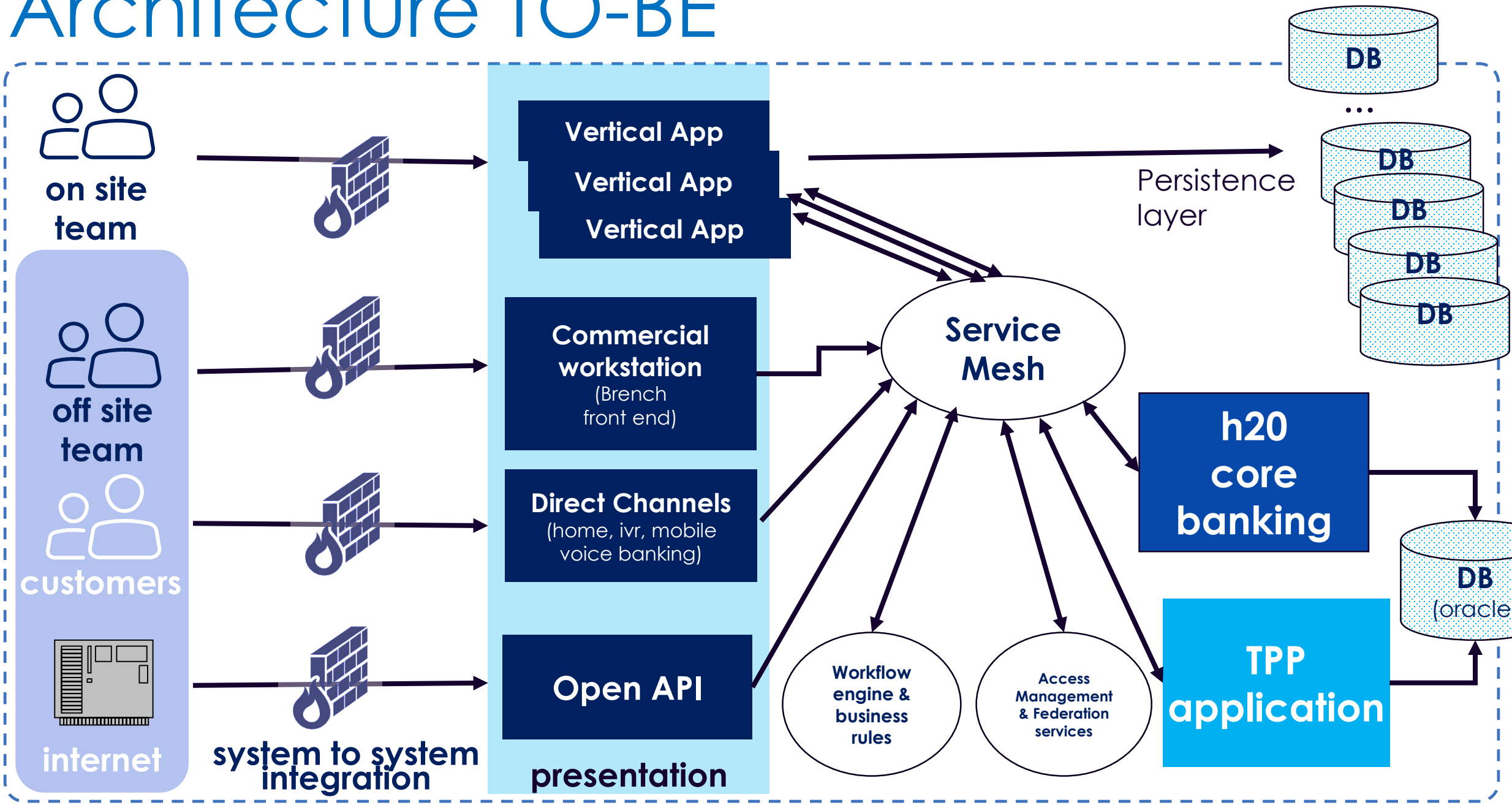
centrico

Architecture AS-IS



centrico

Architecture TO-BE



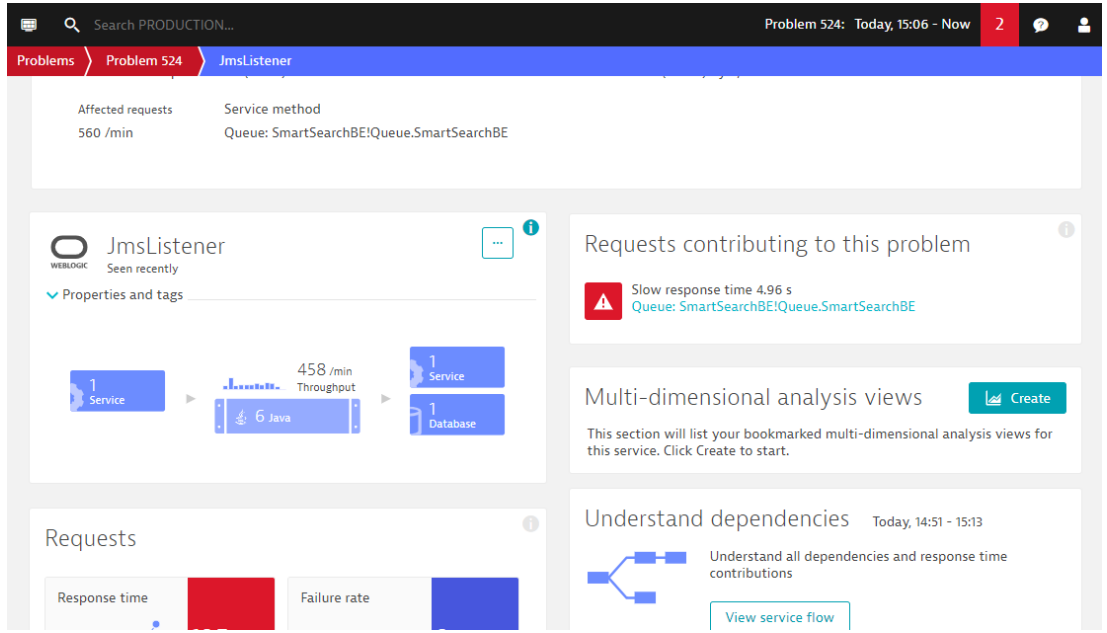
Principles & Practices

centr/co:

APM1/2

TREAT APPS LIKE SPACE PROBES

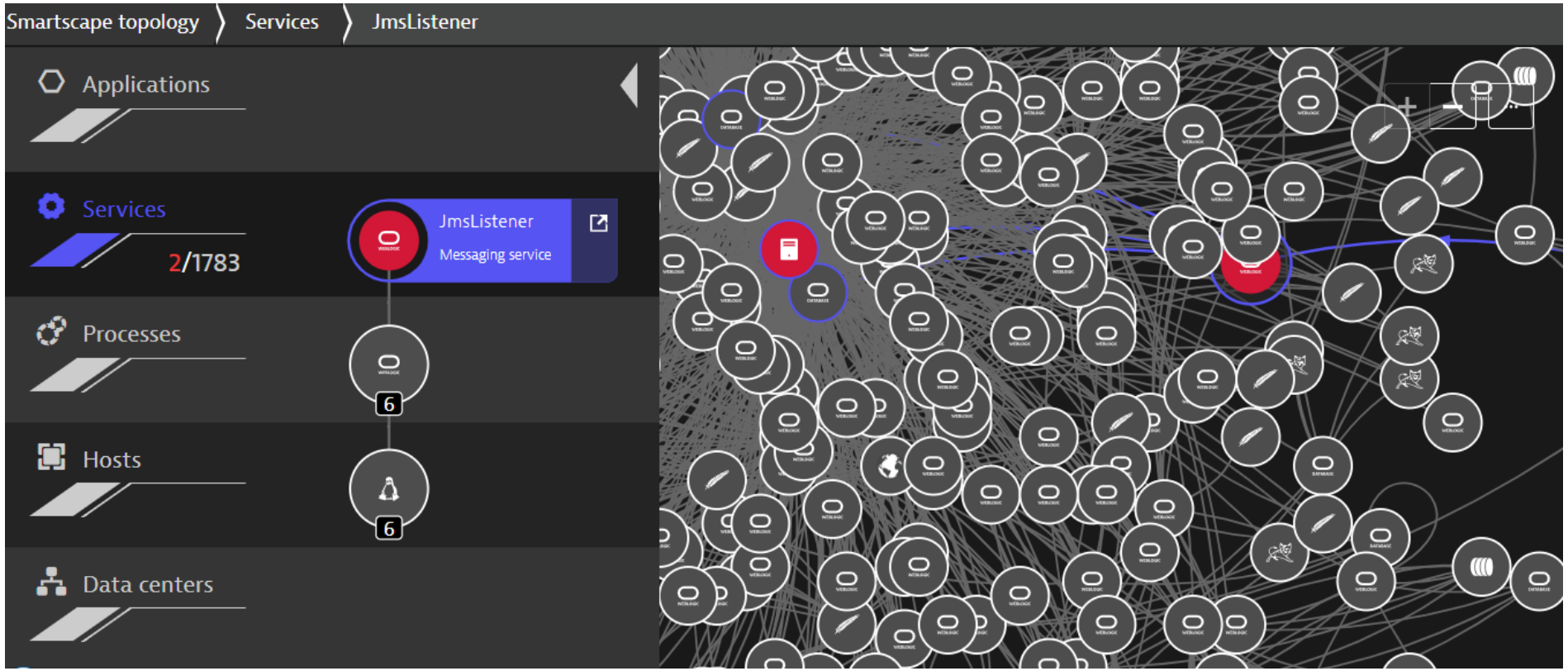
APM, consists of a stream of events that can be used by tools outside servers to keep tabs on how well your application is performing.



It's a team inside architecture office, supported by a tool

centr/co :

APM 2/2

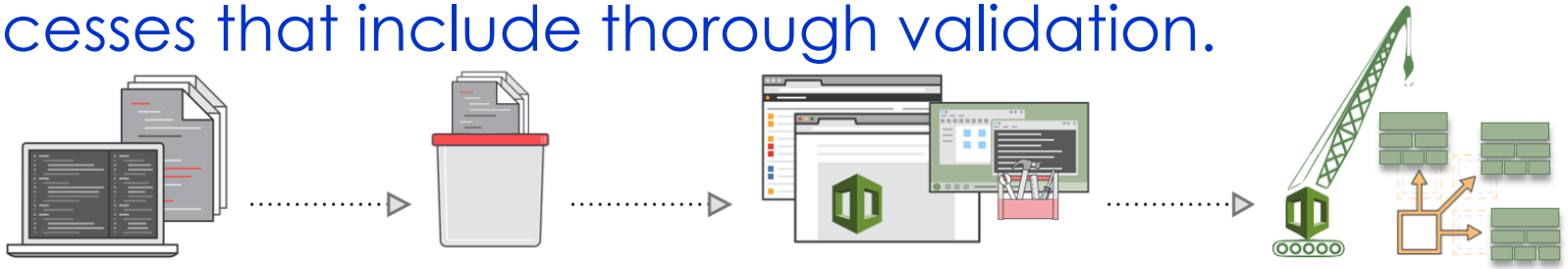


Automatically detects problems in a ecosystem of IT services

centrico

Infrastructure as a code

Infrastructure as code is an approach to infrastructure automation based on practices from software development.
It emphasizes consistent, repeatable routines for provisioning and changing systems and their configuration.
Changes are made to definitions and then rolled out to systems through unattended processes that include thorough validation.



Code your infrastructure from scratch with the CloudFormation template language, in either YAML or JSON format, or start from many available sample templates

Check out your template code locally, or upload it into an S3 bucket

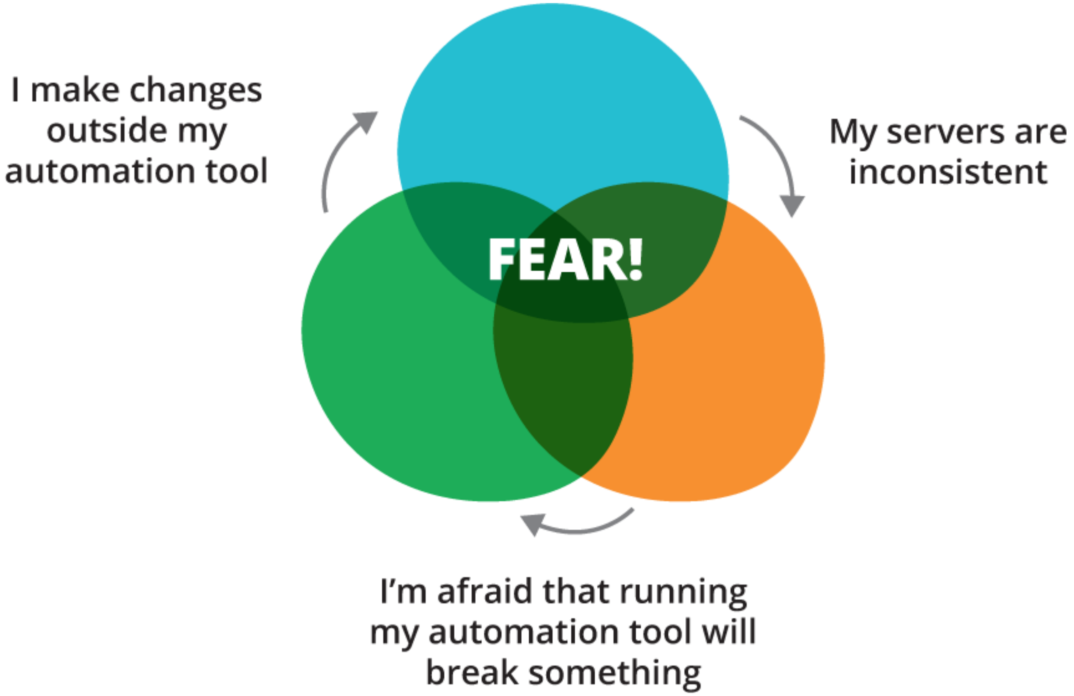
Use AWS CloudFormation via the browser console, command line tools or APIs to create a stack based on your template code

AWS CloudFormation provisions and configures the stacks and resources you specified on your template

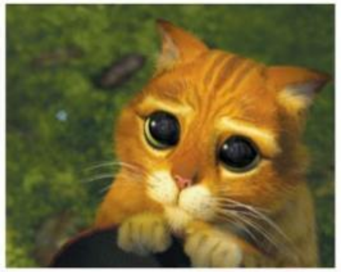
Infrastructure as a code Example

centrico:

Infrastructure as a code: Automation fear

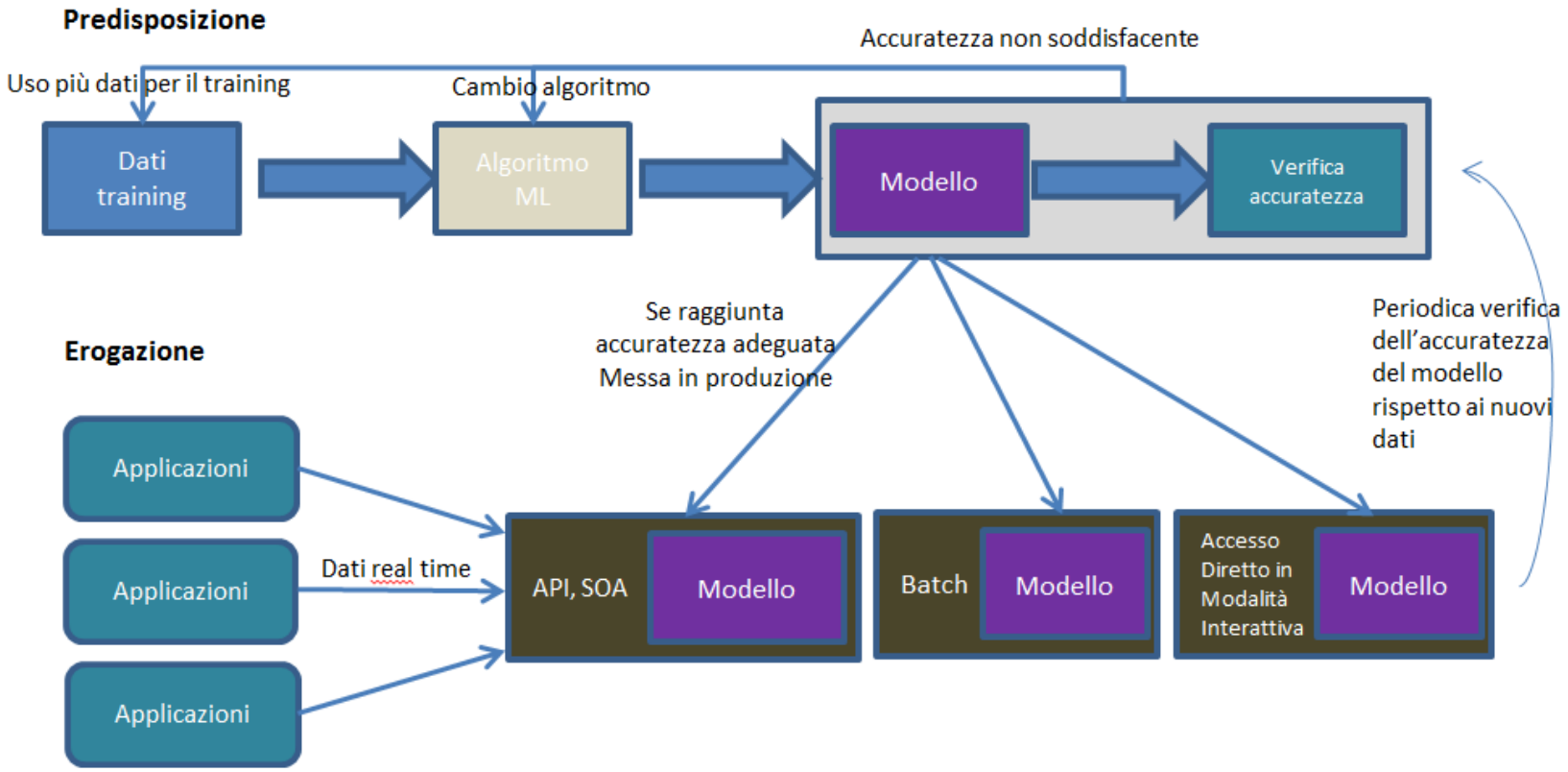


Treat your servers like cattle, not pets.



AI

Integrazione nel Sistema Informatico



centrico

12 Factor App

What is 12FA:

- Methodology for building SaaS apps
- Apps has clean contract with underlying operating system.
- Enable continuous deployment with maximum agility, significant scale up capability
- And Independent of programming languages and back end services.

Why:

- Maximum portability between environments.
- Suitable for deployment on modern cloud platforms.
- Scale up
- Minimize time and costs with automation.
- Continuous deployment.
- Complexity increases if you have diverse programming environments.
- Applying certain constraints will help adequately track, use and share the applications easily.
- Its build to exploit the modern cloud platforms principles with agility in mind.

Twelve Factor & Beyond

Heroku 2012

I. Codebase
One codebase tracked in revision control, many deploys
II. Dependencies
Explicitly declare and isolate dependencies
III. Config
Store config in the environment
IV. Backing services
Treat backing services as attached resources
V. Build, release, run
Strictly separate build and run stages
VI. Processes
Execute the app as one or more stateless processes
VII. Port binding
Export services via port binding
VIII. Concurrency
Scale out via the process model
IX. Disposability
Maximize robustness with fast startup and graceful shutdown
X. Dev/prod parity
Keep development, staging, and production as similar as possible
XI. Logs
Treat logs as event streams
XII. Admin processes
Run admin/management tasks as one-off processes

Bibliogafia

Books:

- Beyond the Twelve-Factor Appby Kevin Hoffman
- Microservices and Containers, First editionby Parminder Singh Kocher
- Architectural Patternsby Pethuru Raj; Anupama Raman; Harihara Subramanian
- Cloud Native Javaby Kenny Bastani; Josh Long
- DevOps with OpenShiftby Stefano Picozzi; Mike Hepburn; Noel O'Connor
- *The Mythical Man-Month* Fred Brooks
- Infrastructure as Codeby Kief Morris

Website:

https://martinfowler.com/
https://aws.amazon.com/blogs/aws/
https://developers.redhat.com/

centrico:

centrico:

Centrico S.p.A
sede legale in Piazza Gaudenzio Sella n. 1, 13900 Biella (BI)
Iscritta al Registro di Biella e Vercelli, codice fiscale 02668670025
Società appartenente al Gruppo IVA Maurizio Sella S.A.A. con partita IVA 02675650028
Soggetta all'attività di direzione e coordinamento di Banca Sella Holding S.p.A.
PEC: centrico@pec.centrico.tech MAIL: info@centrico.tech

centrico