IT Architecture: From SOA To Microservices
• Why?
• Culture
• Technology
• Principles & Practices
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

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Complexity</th>
<th>Service Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;3 millions of accounts</td>
<td>&gt;350 locations in 3 countries</td>
<td>RTO 12h</td>
</tr>
<tr>
<td>&gt;2 millions of transactions/day</td>
<td>&gt;5.500 users</td>
<td>RPO 0</td>
</tr>
<tr>
<td>&gt;7 millions of API calls/day</td>
<td>&gt;€36bn of Assets</td>
<td>99.987% availability (2013-2018)</td>
</tr>
<tr>
<td>&gt;3,000 Tb of storage</td>
<td>&gt;1.700 server in private cloud</td>
<td></td>
</tr>
<tr>
<td>&gt;1.700 change request every year</td>
<td>&gt;8.000 devices managed</td>
<td></td>
</tr>
<tr>
<td>&gt;120,000 POS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BPO already manages important volumes

<table>
<thead>
<tr>
<th>Services</th>
<th>Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete back office operations and document management</td>
<td>&gt; 40Mln tickets/year (data entry – processing – digitazion– archiviation)</td>
</tr>
<tr>
<td>Credit Practices issued</td>
<td>&gt;28k/year</td>
</tr>
<tr>
<td>Inbound call, chat e email</td>
<td>&gt;1Mln contacts per year</td>
</tr>
<tr>
<td><strong>Robot Process Automation</strong></td>
<td>169 flows managed by Robots</td>
</tr>
</tbody>
</table>
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?
Why?

Hype customers trend (forecasting in yellow)
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**
Culture
Communication Skill

### TOGAF® Standard — Version 9.2

#### Communication Skill Level

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

#### Generic Skills

<table>
<thead>
<tr>
<th>Roles</th>
<th>Architecture Board Member</th>
<th>Architecture Sponsor</th>
<th>Enterprise Architecture Manager</th>
<th>Enterprise Architecture Technology</th>
<th>Enterprise Architecture Data</th>
<th>Enterprise Architecture Applications</th>
<th>Enterprise Architecture Business</th>
<th>Program/Project Manager</th>
<th>IT Designer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leadership</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Teamwork</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Inter-personal</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Oral Communications</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Written Communications</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Logical Analysis</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Stakeholder Management</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Risk Management</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap46.html
**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.
Two Pizzas Teams

Source: The Mythical Man-Month 1975 Fred Brooks
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
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.
Container VS Virtual Machine

**VIRTUAL MACHINES**

- VM
  - App
  - App
  - App
  - App
  - OS Dependencies
  - Kernel
  - Hypervisor
  - Hardware

VM virtualizes the hardware

**CONTAINERS**

- Container
  - Container
  - Container
  - Container
  - Container Host (Kernel)
  - Hardware

Container virtualizes the process

**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
Build & Deploy Containers

**BUILD APP**
(OpenShift)

**BUILD IMAGE**
(OpenShift)

**DEPLOY**
(OpenShift)
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.
[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.
SOA vs Microservices

SOA services based on single java ear monolith

Applications use different tables or schemas if not different instances

MicroServices

RESTful

Applications

DB

Application A

Application A

Application B

Application Z

H2O

Services

SOA

DB

Application

RESTful
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
Architecture AS-IS
Principles & Practices
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
APM 2/2

Automatically detects problems in a ecosystem of IT services
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.
Infrastructure as a code: Automation fear

Treat your servers like cattle, not pets.
Integrazione nel Sistema Informatico

**AI**
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

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
Bibliografía

Books:
• Beyond the Twelve-Factor App by Kevin Hoffman
• Microservices and Containers, First edition by Parminder Singh Kocher
• Architectural Patterns by Pethuru Raj; Anupama Raman; Harihara Subramanian
• Cloud Native Java by Kenny Bastani; Josh Long
• DevOps with OpenShift by Stefano Picozzi; Mike Hepburn; Noel O’Connor
• The Mythical Man-Month Fred Brooks
• Infrastructure as Code by Kief Morris

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