



# DEVOPS AND MICROSERVICE APIS

2nd Vienna Software Seminar (VSS)

## Opportunities and Challenges when Applying DevOps Principles and Designing Architectures

Stepan Seycek - BOC

# BOC Group – Facts & Figures



CUSTOMERS

**1.000+**

in 50+ Countries

PARTNERS

**90+**

in 25+ Countries

PROJECTS

**5.000+**

in 20+ Years

WE ARE 200+ Employees

Product Development  
Business & IT Consultants  
Solution Engineers  
Marketing and Sales



0%  
Outsourcing

100%  
Independent

USER COMMUNITY

**90.000+**

in 120+ Countries

INSTALLATIONS

**35.000+**

in 80+ Countries



Offices and Partners  
Worldwide



20+ Years of Experience



0% Outsourcing  
100% BOC Group



Everything under one roof  
Product, Consultancy, Training

# BOC Cloud Services

What does DevOps mean for us and how do we apply it



## What we do

- Delivery of ADONIS, ADOIT and GRC as **SaaS**
- Manage **360 installations** of our software

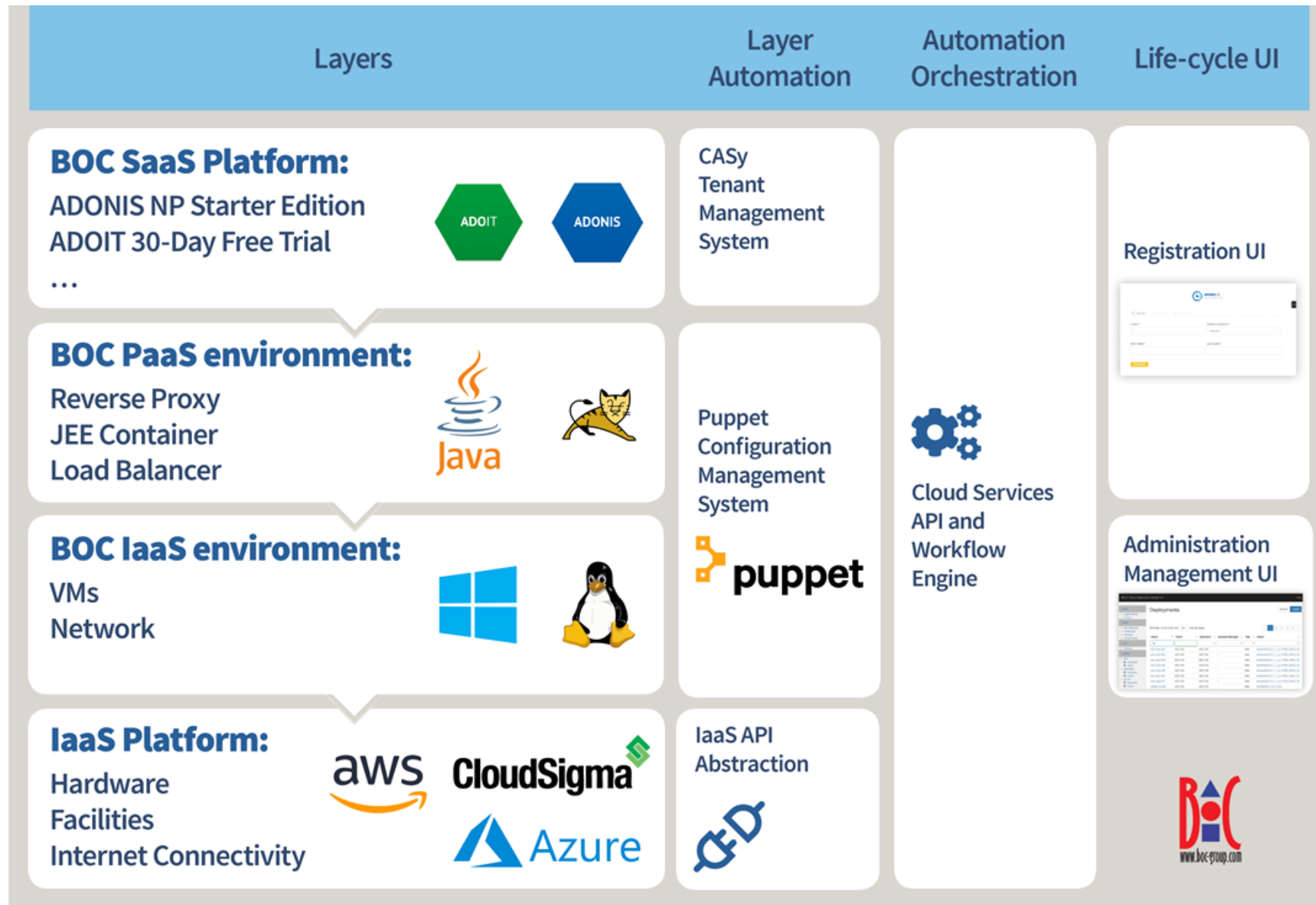
## What we don't do

- Deal with **data centre** facilities
- Deal with **hardware** and virtualization

## How we embrace DevOps

- **Maintain** the **systems** on top of IaaS
- Use **version control** for everything
- Develop and operate tools for **automation**
  - of **deployment**
  - for **managing accounts**

# BOC Cloud Services Platform



# The challenge of **ADONIS** Community Edition



**Mission:** *“Replace the existing ADONIS Community Edition (download version) with a cloud service.”*

**Challenge:** Existing ADONIS community members: 90,000+

**Approach:** *“Provide the service to those who really use it.”*

## Establish an **account life-cycle**

- Provision the account upon registration
- Track the account usage
- Warn the user when the account is about to expire
- Save user’s content upon expiration
- Decommission unused accounts
- Provide the saved content to the user

# ADONIS CE SaaS – designing the architecture



Identify all **use cases** to be covered

- 6 user centric
- 2 admin centric

Identify **bounded contexts**

- 6 identified

Evaluate **API technologies**

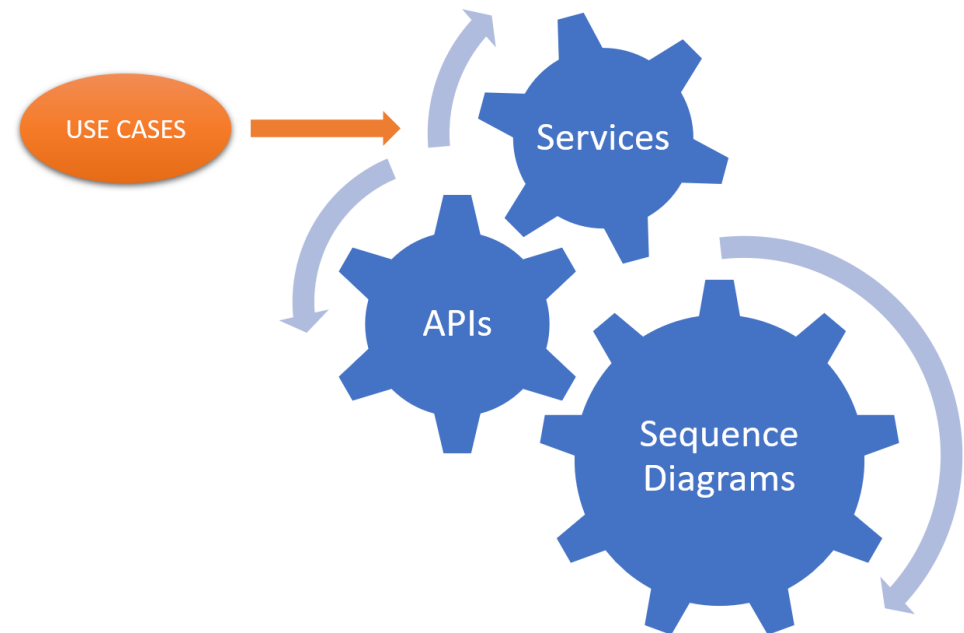
- REST
- AMQP

Design **APIs and data models**

- CRUD where possible

Document **workflows**

- Sequence diagrams



# ADONIS CE SaaS – the **positive take-aways**



## **Good granularity** found

- ✓ Clear **separation of concerns**
- ✓ Use of **best matching technology**
- ✓ **Distribution** among teams

## **API design**

- ✓ **Consistent**
- ✓ Only **minimal adaptations** required during implementation

## **AMQP** messaging

- ✓ Helps with **load balancing**
- ✓ Helps with **queuing**
- ✓ Well suited for **workflow communication**



## Deployment complexity

- Every single microservice instance configured in **Puppet**
- Explicit **load balancer configuration** in Puppet
- **Credentials** spread across multiple micro service configurations
- No automatic **service discovery**

- **Containerise** the micro services
- Orchestrate with **Kubernetes** in order to
  - **reduce complexity** (service discovery, built in load balancer, secrets manager, ...)
  - benefit from built in **cluster mechanisms**

