Dienst ICT Uitvoering
Ministerie van Economische Zaken

Do's & don'ts @DICTU

Program Case Management

OpenShift, PEGA, EDB and Alfresco

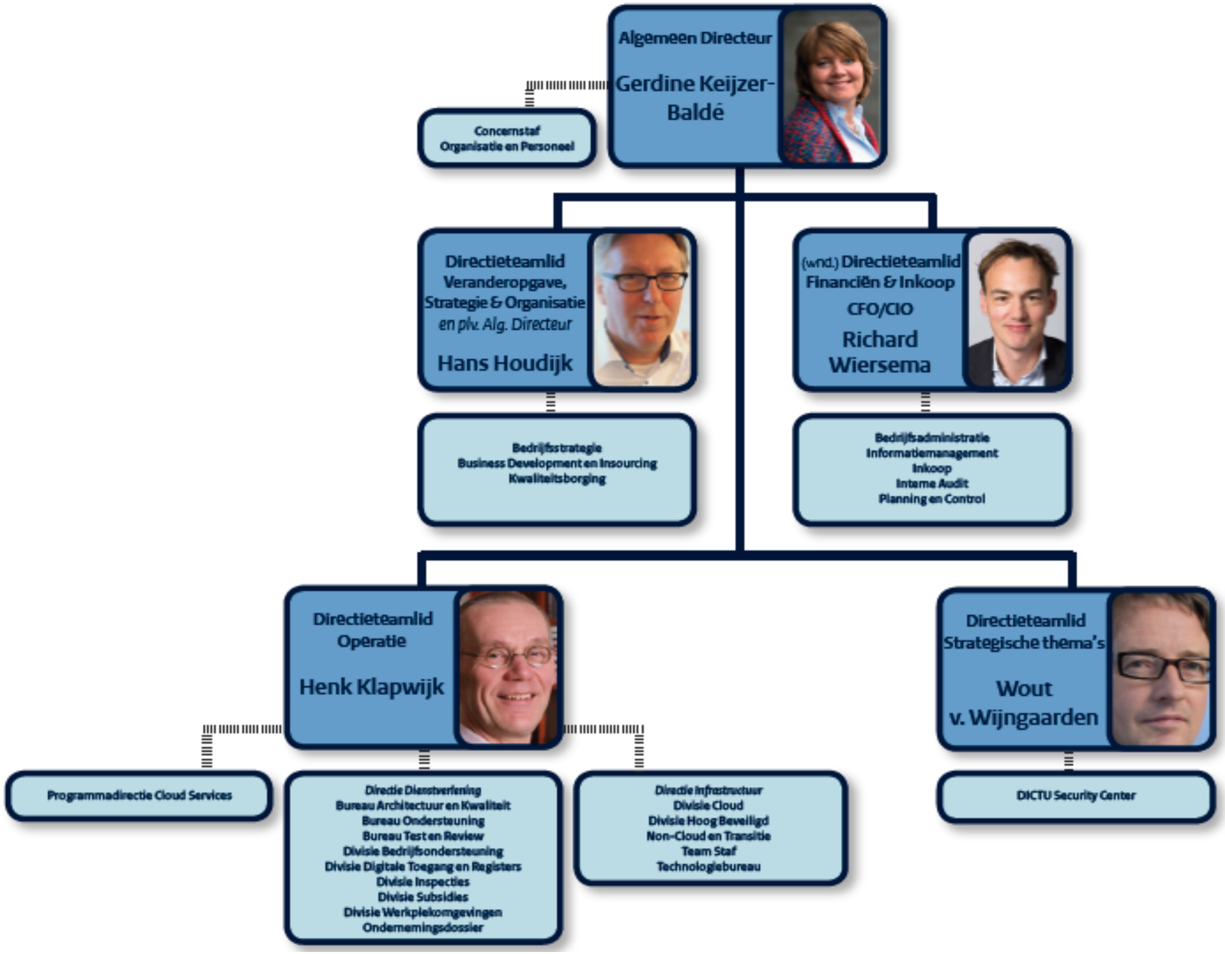
Contents

1. Introduction Dutch Government
2. Introduction Program Case Management
3. Development product “ZGW-Basisdienst” Platform as a Service
4. Demonstration
5. Do's & don’ts @DICTU
6. Questions?

1. Introduction Dutch Government

- Cloud Computing (digital video)
- DICTU (Ministry of Economic Affairs)
- Data Center North (digital video)

Introduction: WWW.DICTU.NL
- organization -



Algemeen Directeur
Gerdine Keijzer-Baldé

Concernstaf
Organisatie en Personeel

Directieteamlid
Veranderopgave, Strategie & Organisatie
en plv. Alg. Directeur
Hans Houdijk

Bedrijfsstrategie
Business Development en Insourcing
Kwaliteitsborging

(wnd.) Directieteamlid
Financiën & Inkoop
CFO/CIO
Richard Wiersema

Bedrijfsadministratie
Informatiemanagement
Inkoop
Interne Audit
Planning en Control

Directieteamlid
Operatie
Henk Klapwijk

Programmadirectie Cloud Services

Directie Dienstverlening
Bureau Architectuur en Kwaliteit
Bureau Ondersteuning
Bureau Test en Review
Divisie Bedrijfsondersteuning
Divisie Digitale Toegang en Registers
Divisie Inspecties
Divisie Subsidies
Divisie Werkplekomgevingen
Ondernemingsdossier

Directie Infrastructuur
Divisie Cloud
Divisie Hoog Beveiligd
Non-Cloud en Transitie
Team Staf
Technologiebureau

Directieteamlid
Strategische thema's
Wout v. Wijngaarden

DICTU Security Center

Introduction: WWW.DICTU.NL
- facts and numbers -

- DICTU is IT Shared service organisation
- Part of the Ministry of Economic Affairs
- 1500 applications, of which 830 bespoke business applications;
- Data connections in the Netherlands to about 210 locations;
- Twin data centres in Groningen;
- Over 3300 servers;
- 14,000 workplaces and 13,000 mobile devices;
- As income tax service, DICTU has a turnover of 200 million euros;
- DICTU employs over 1200 people.

Introduction: WWW.DICTU.NL
- strategy for the next 3 years -

- **Who** - DICTU works only for the Dutch government
- **What** - DICTU provides adequate service
- **How** - DICTU is reliable and agile

DICT wants to deliver added value through:

- *customer focus*
- *reliable, cost-effective and innovative services*

2. Introduction Program Case Management

- Opportunities
- Standard Government solution
- Why using PegaSystems & Alfresco?
- Why using the OpenShift Container Platform?
- People who can

Opportunities for small Government organizations

Ministry of Justice

Council for Criminal Justice and Youth Protection

Dienst Justitiële Inrichtingen
Ministerie van Veiligheid en Justitie

Raad voor Strafrechtstoepassing
en Jeugdbescherming

Ministry of Economic Affairs

Inspection telecom frequencies

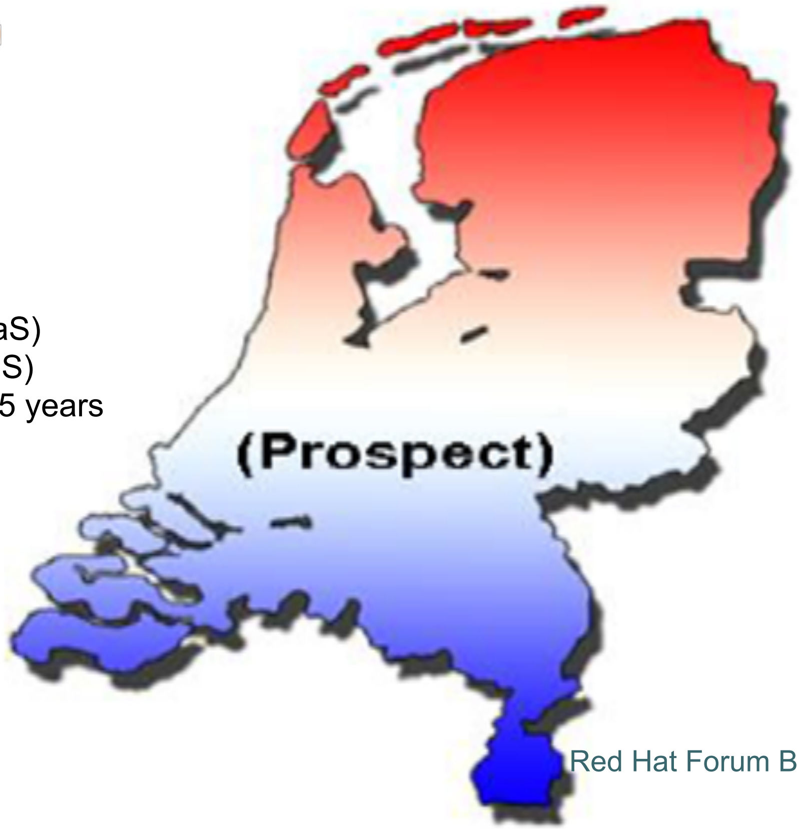
Agentschap Telecom
Ministerie van Economische Zaken

Ministry of Public Health, Welfare and Sport

Health Inspection

Inspectie voor de Gezondheidszorg
Ministerie van Volksgezondheid,
Welzijn en Sport

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Maintenance contracts for 5 years
- Pay per User model

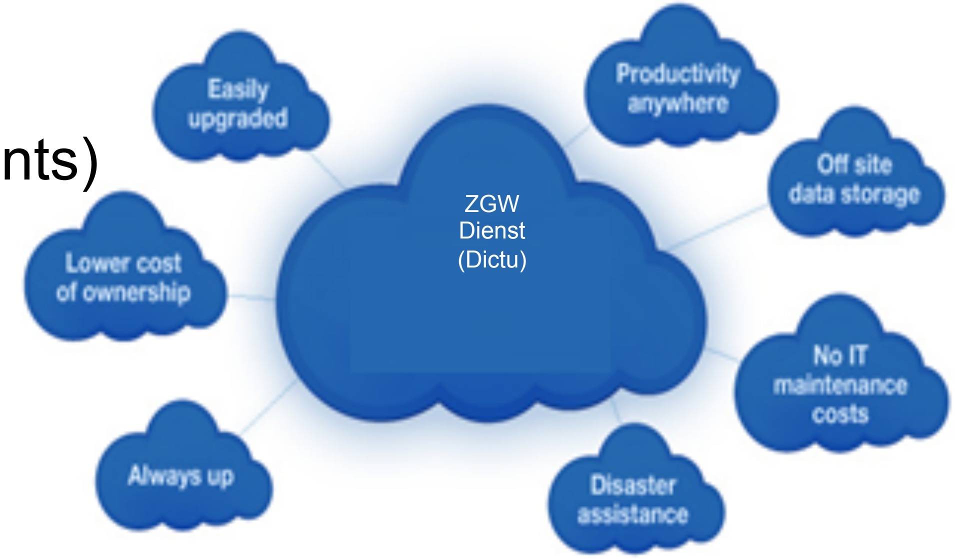


- 230 Government Inspections
- 100 and 1000 Employees
- 150.000 Government Users
- Revenue 300 million euro’s - maintenance
- Revenue 300 million euro’s - Digital Innovation

Our Standard Government Solution

The USPs for Government organizations are

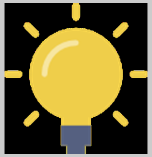
- Business in the lead
- Reusable business processes
- Security (government requirements)
- Affordable (pay per user)
- Agile (approach)
- Scalable (Horizontal & Vertical)
- Simple (no tender needed)
- Fast (short time to market)
- Any time, any place, any where and any device
- Not for Profit



The DICTU Approach for new Customers

Consists of four phases

Phase I Startup



- Kick-off
- Requirements
- Processes (IST/SOLL)
- Architecture IST/SOLL
- Employees and Organization
- Readiness check PAAS solution
- Business case
- Go/ No Go

- Project plan Phase II

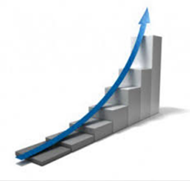
4 - 8 weeks

Phase II Initiaton

- Proof of Value (POV)
- Pricing (phase III,IV)
- Project plan(phase II, IV)
- Go/No Go Phase III Realization

4 weeks

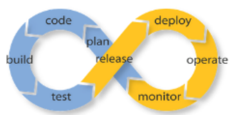
Phase III Realization



- Agile
- DCO
- Development
- Test
- Implementation

X months

Phase IV Maintenance

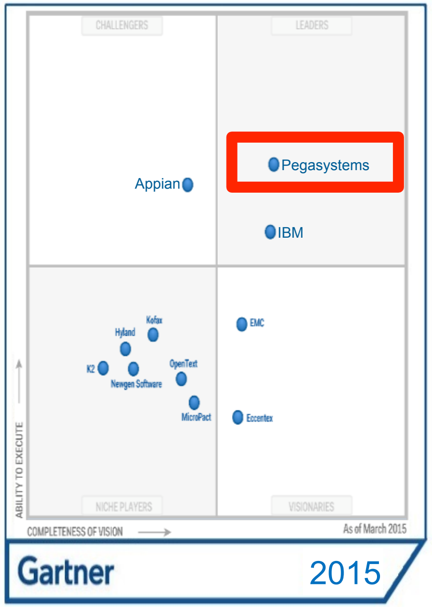


- Changes
- Maintenance
- Service management
- Account management

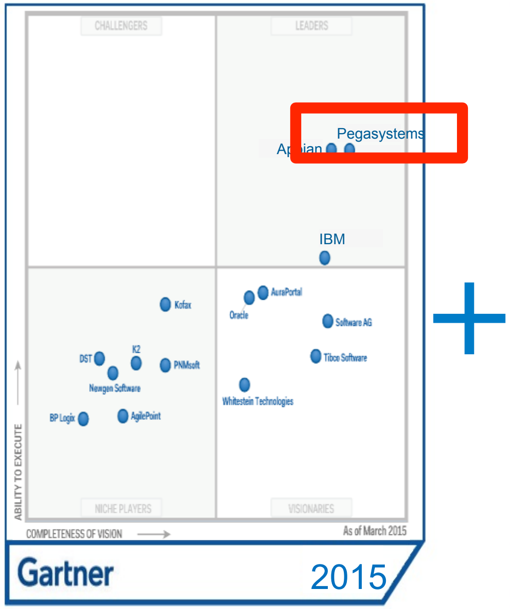
5 years

Why using PEGA & Alfresco ?

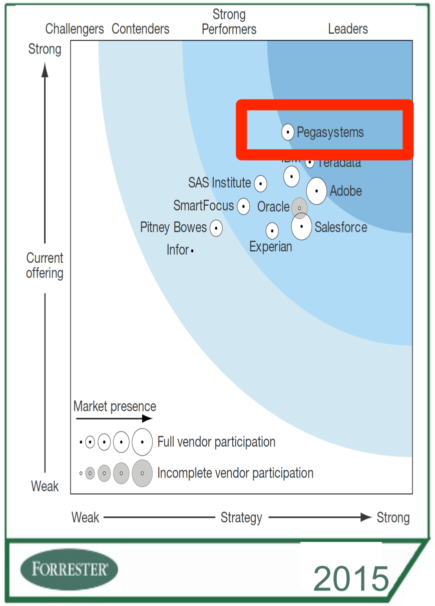
Casemanagement, BPM, Management information en CRM solution



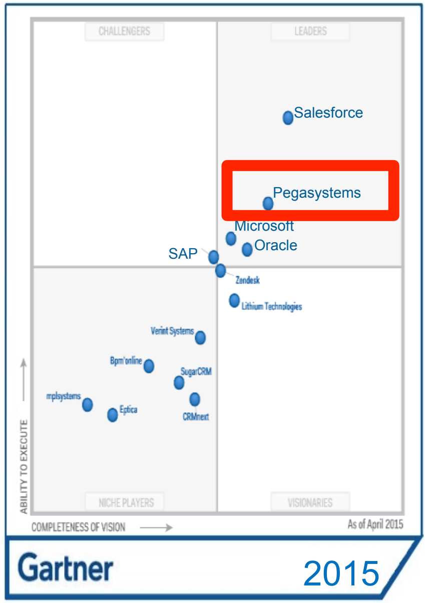
CASE MANAGEMENT #1



BPM #1



REAL-TIME DECISIONS #1



LEADER IN CRM

“Pegasystems has the best ability to model and predict a customer's behavior, and to deliver the next action to the agent.”

Gartner*

*Magic Quadrant for the CRM Customer Engagement Center, 2015

PEGA references in the public sector

TCEQ

HM Revenue & Customs

USDA

energysafe VICTORIA

Texas Workforce Commission

Australian Government
Department of Immigration and Border Protection

DMDC

AHVLA
Animal Health and Veterinary Laboratories Agency

Texas Department of Transportation

PEGA Nederland

Voornamelijk actief in Private sector en voor EZ in publieke sector

nationale nederlanden

achmea

PHILIPS

Rabobank

Dienst ICT Uitvoering
Ministerie van Economische Zaken

Why using Red Hat Open Shift Container Platform ?

Requirements	Components
Scalability (vertical en horizontal)	Virtualisation infrastructure Containerisation and redundant configuration of Pega, Alfresco en PostgreSQL
Performance	Continuous Delivery: performance testing Monitoring: End user monitoring and resource monitoring Containerisation: deployment application- and database nodes. Efficient resource usage
Availability7x24	Containerisation with redundant configuration of Pega, Alfresco en PostgreSQL
Continuous Integration (incl. automated testing)	Pega: Pega 7.3 (when available) Continuous Delivery: Tosca/Selenium, Jenkins, GitLab, Nexus, …
Capable to run multiple parallel test processes	Containerisation and maintenance: automated deployment of environments Continuous Deployment: : automated deployment of application instance and testing environment
Continuous Deployment	Continuous Deployment: Automated deployment
Configuratiemanagement	Continuous Delivery: GitLab, Nexus. Management of containers, application functionality, configuration files, testing functionality.
Multi-tenant PaaS model	Containerisation: management of a big number of servers and components, extensive resource pooling, division of tenants via routing
Data integrity in case of incidents and recovery	ProgreSQL: standard quadruple installation Infrastructure: storage with geo distribution

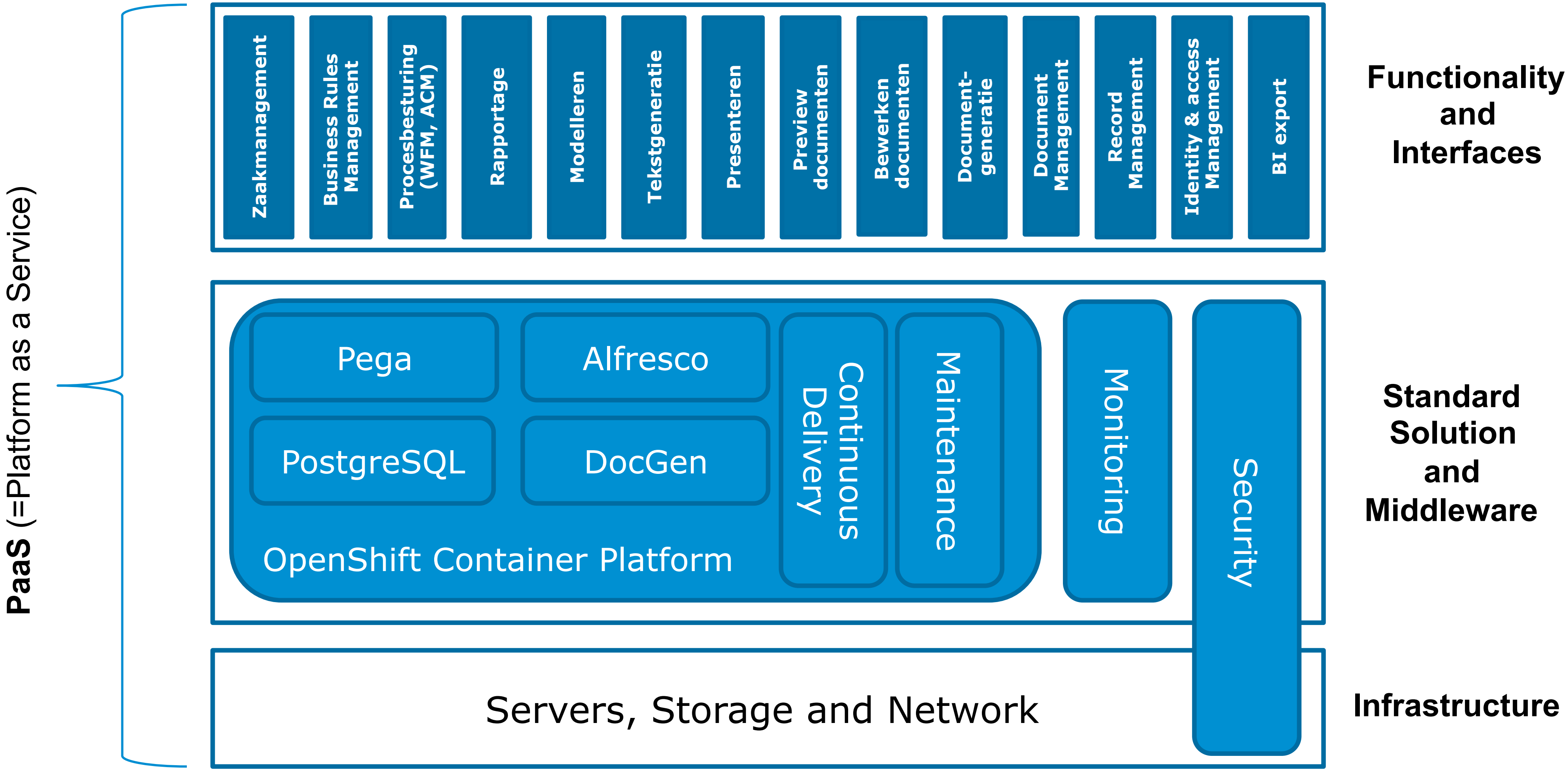
We kunnen mailen -)
RSG live!
Super gedaan mensen

3. Development product “ZGW-Basisdienst”

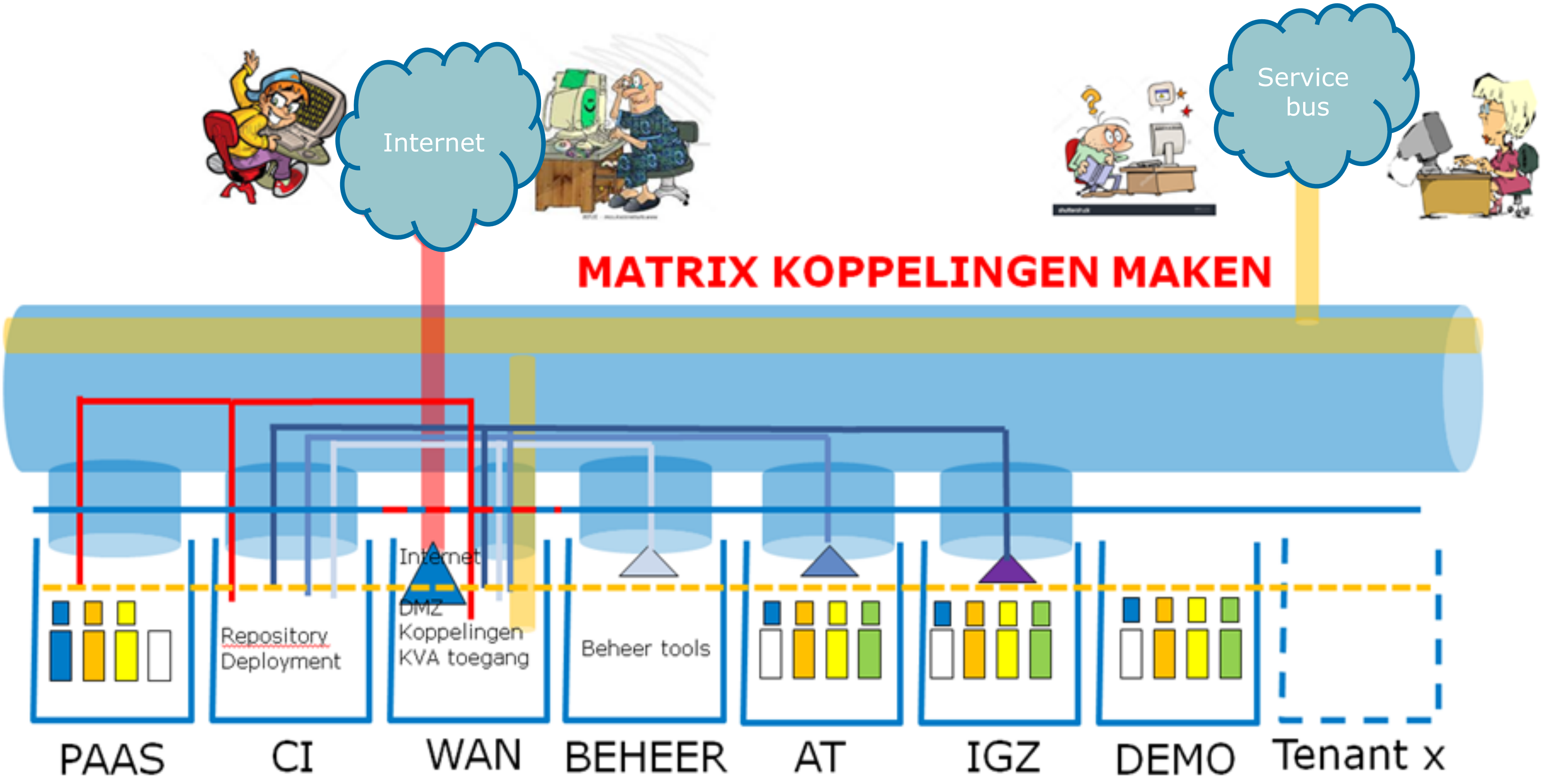
- Multi tenant platform
- IaaS and PaaS
- Continuous Delivery

Multitenant platform

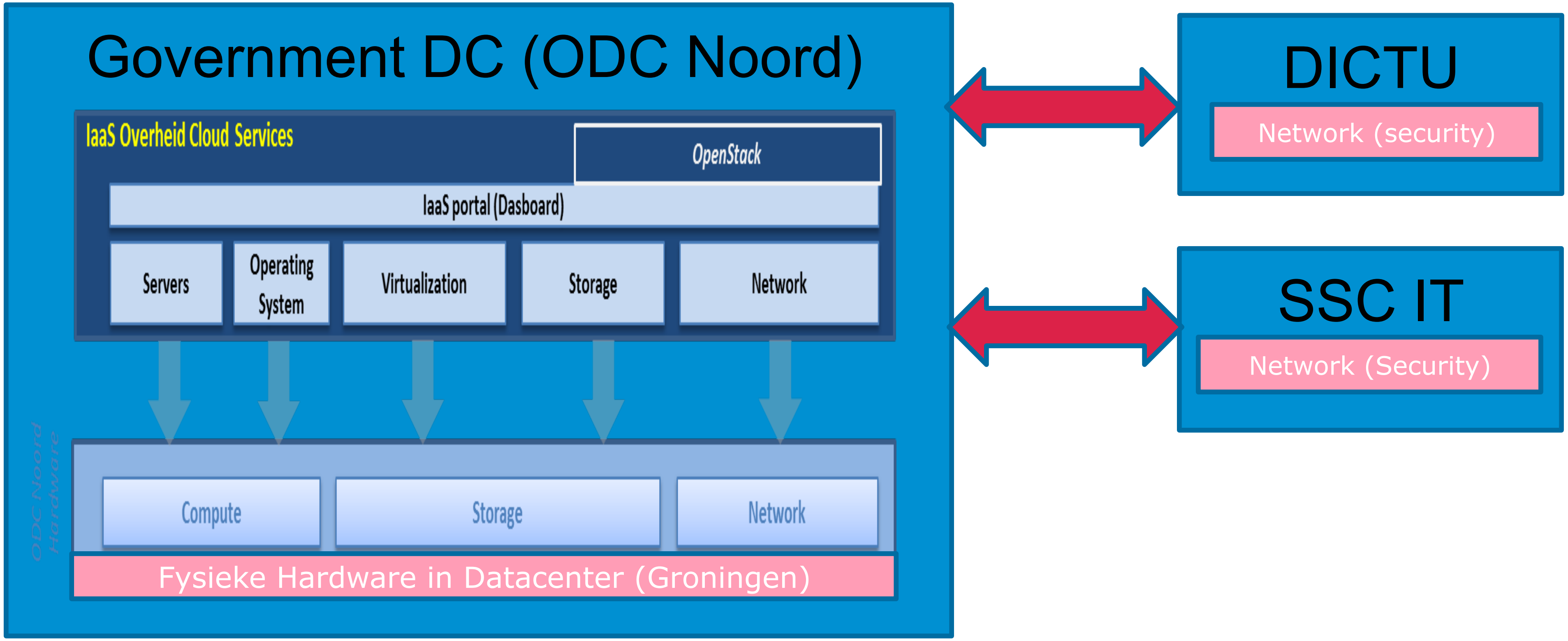
(Multidisciplinary team with product knowledge)



Multi-tenant (Low-level customer separation)



Infrastructuur as a Service



Government DC (ODC Noord)

IaaS Overheid Cloud Services

OpenStack

IaaS portal (Dasboard)

Servers

Operating System

Virtualization

Storage

Network

ODC Noord Hardware

Compute

Storage

Network

Fysieke Hardware in Datacenter (Groningen)

DICTU

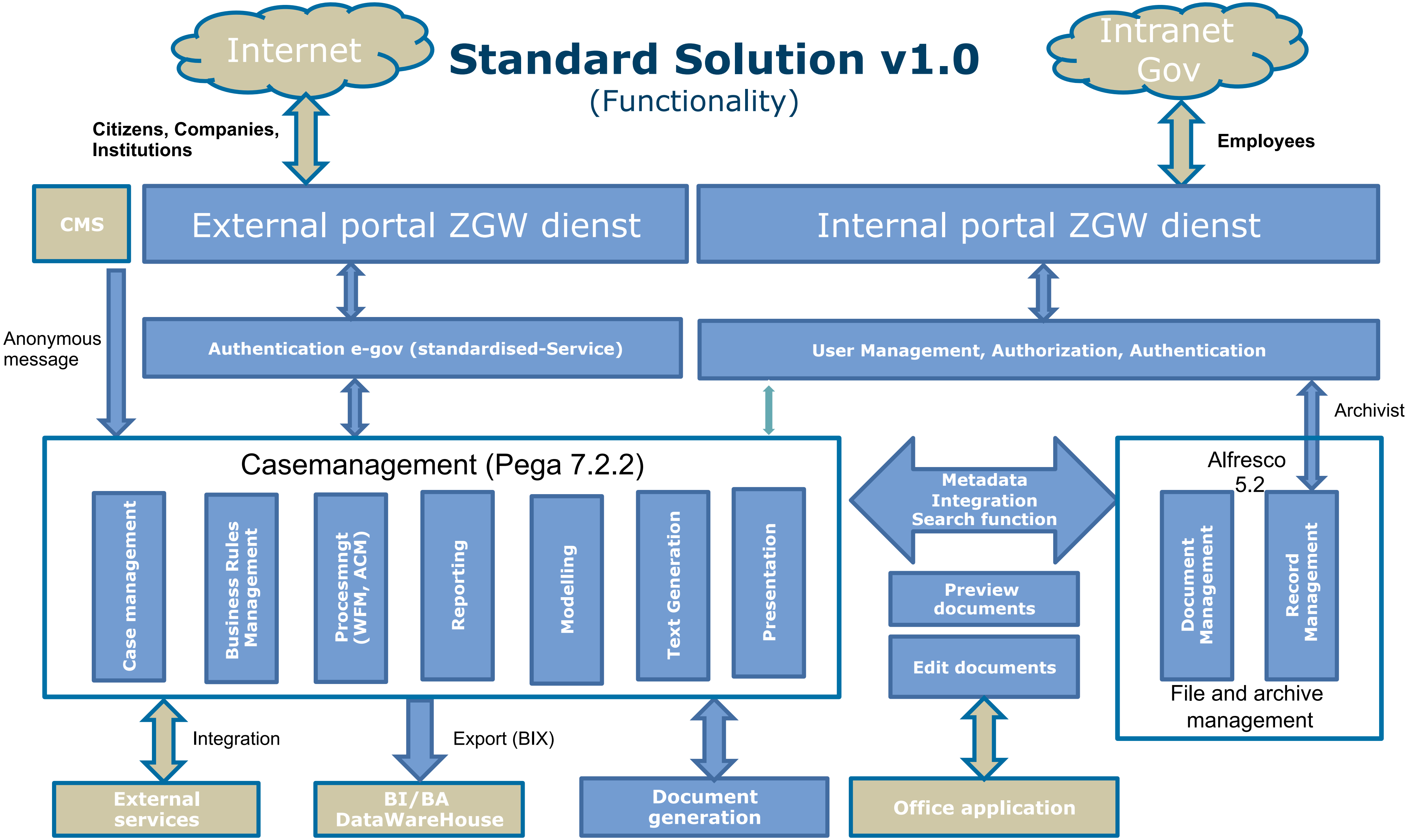
Network (security)

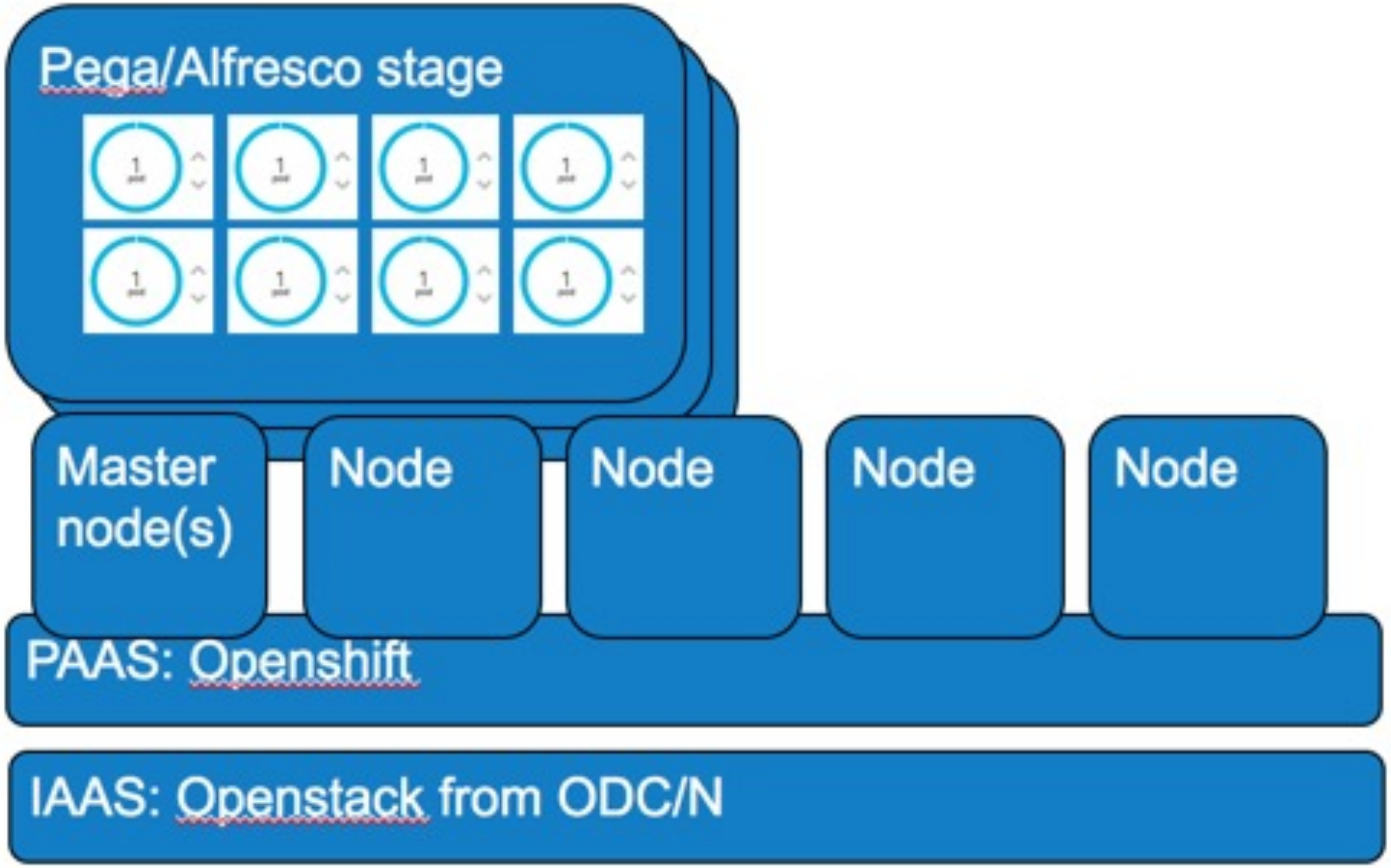
SSC IT

Network (Security)

Standard Solution v1.0

(Functionality)





Pega/Alfresco stage

Master node(s)

Node

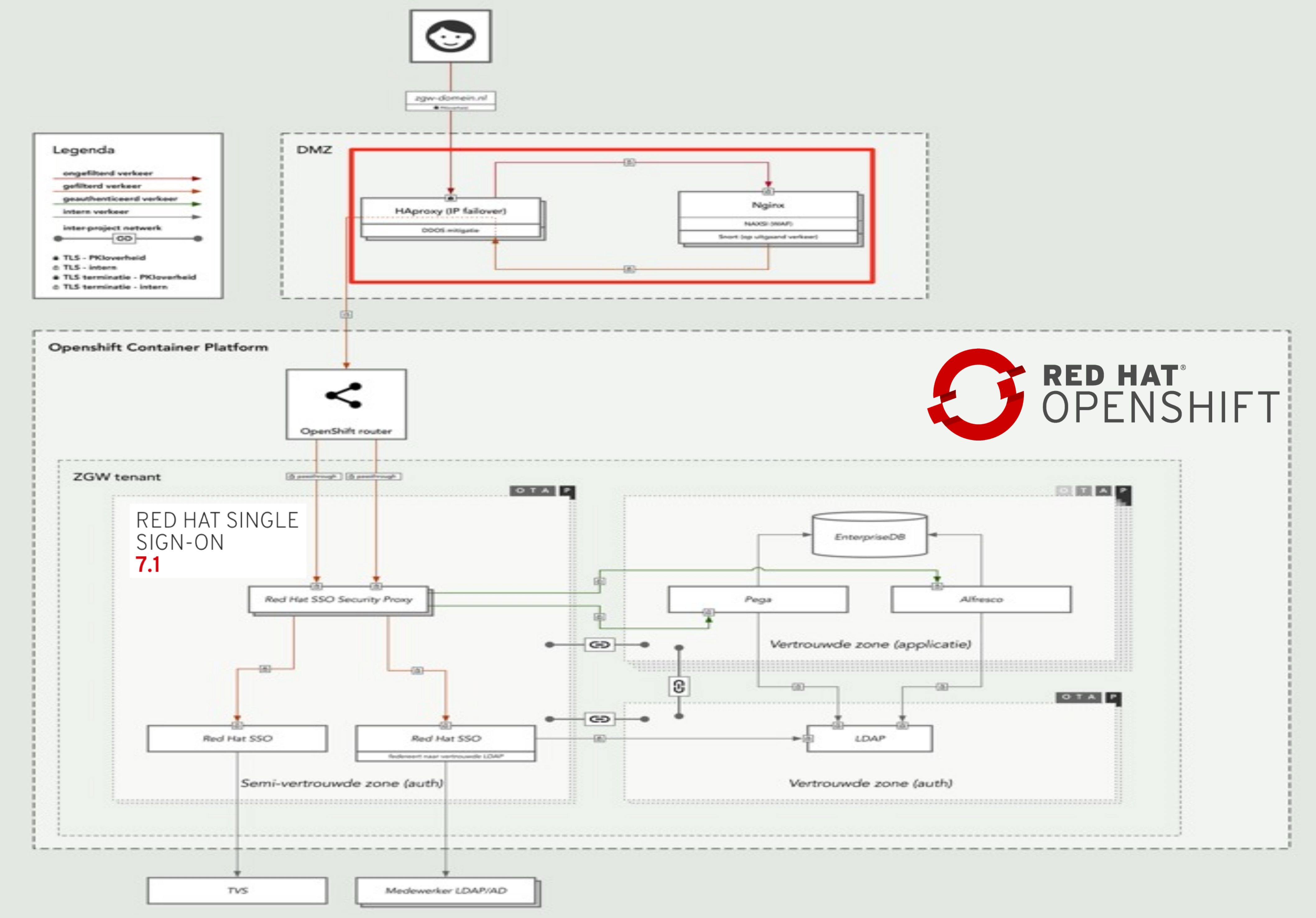
Node

Node

Node

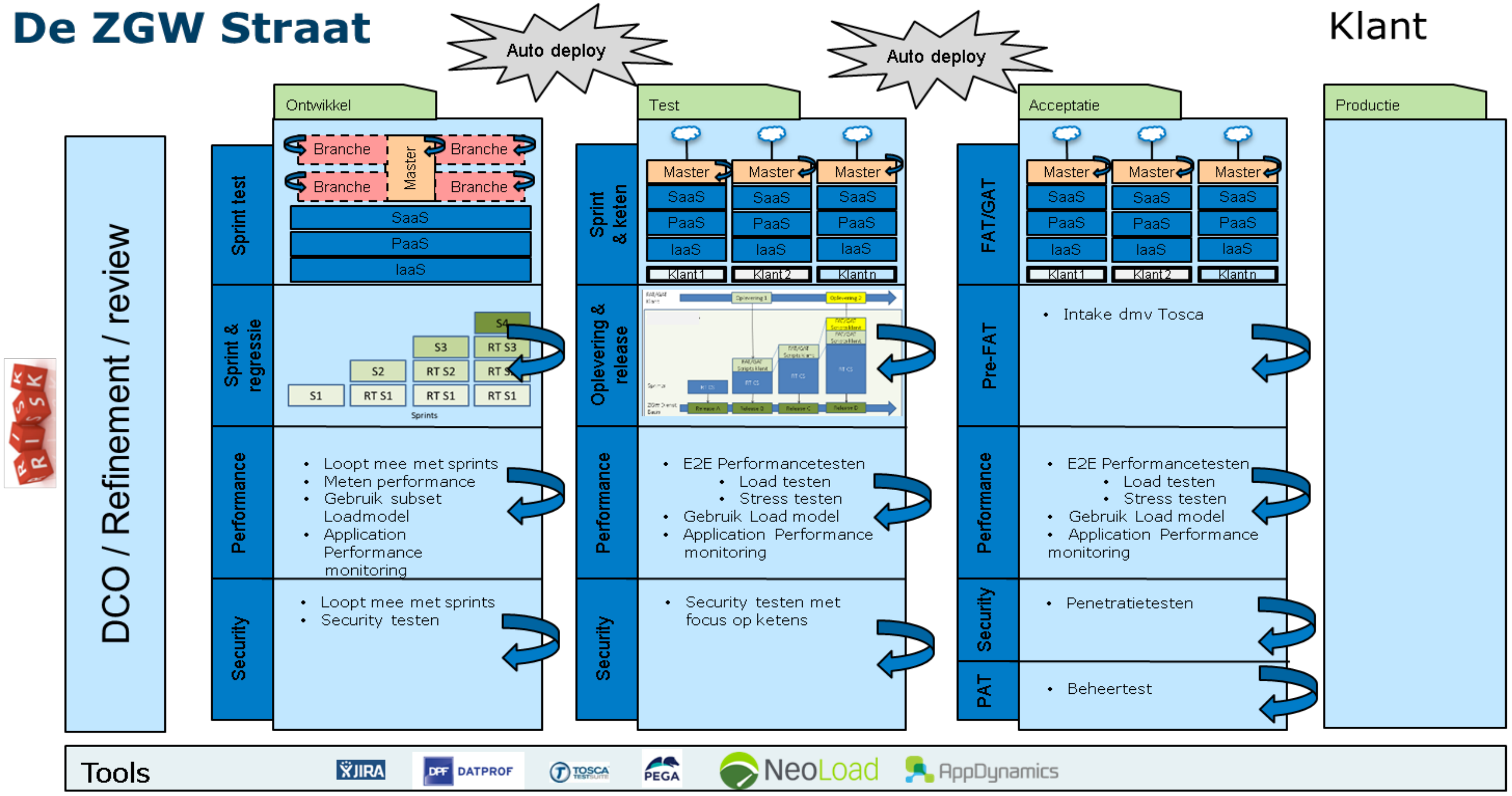
PAAS: Openshift

IAAS: Openstack from ODC/N



RED HAT SINGLE SIGN-ON **7.1**

De ZGW Straat



Auto deploy

Auto deploy

Klant

DCO / Refinement / review

Ontwikkel

- Sprint test: Branche, Master, Branche, Branche, Branche; SaaS; PaaS; IaaS
- Sprint & regressie: S1, S2, S3, S4; RT S1, RT S2, RT S3; Sprints
- Performance:
  - Loopt mee met sprints
  - Meten performance
  - Gebruik subset Loadmodel
  - Application Performance monitoring
- Security:
  - Loopt mee met sprints
  - Security testen

Test

- Sprint & keten: Master, SaaS, PaaS, IaaS; Klant1, Klant2, Klant n
- Oplevering & release
- Performance:
  - E2E Performancetesten
    - Load testen
    - Stress testen
  - Gebruik Load model
  - Application Performance monitoring
- Security:
  - Security testen met focus op ketens

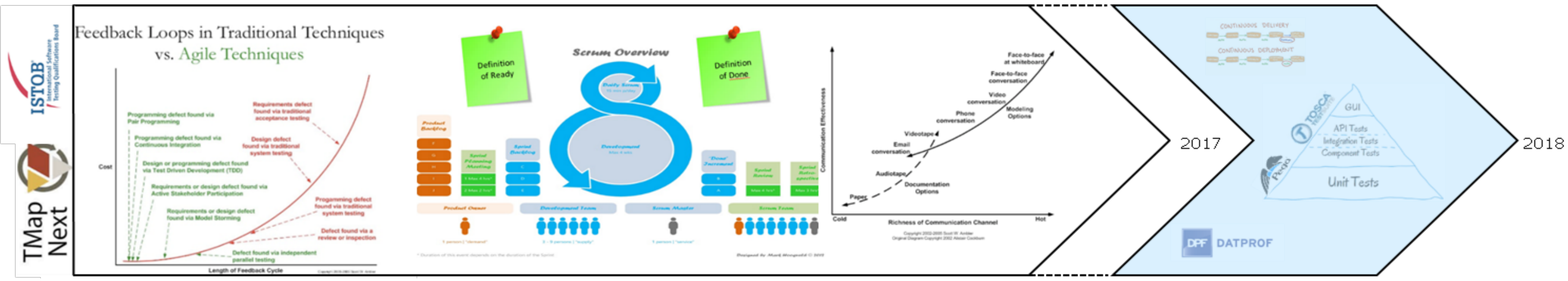
Acceptatie

- FAT/GAT: Master, SaaS, PaaS, IaaS; Klant1, Klant2, Klant n
- Pre-FAT:
  - Intake dmv Tosca
- Performance:
  - E2E Performancetesten
    - Load testen
    - Stress testen
  - Gebruik Load model
  - Application Performance monitoring
- Security:
  - Penetratietesten
- PAT:
  - Beheertest

Productie

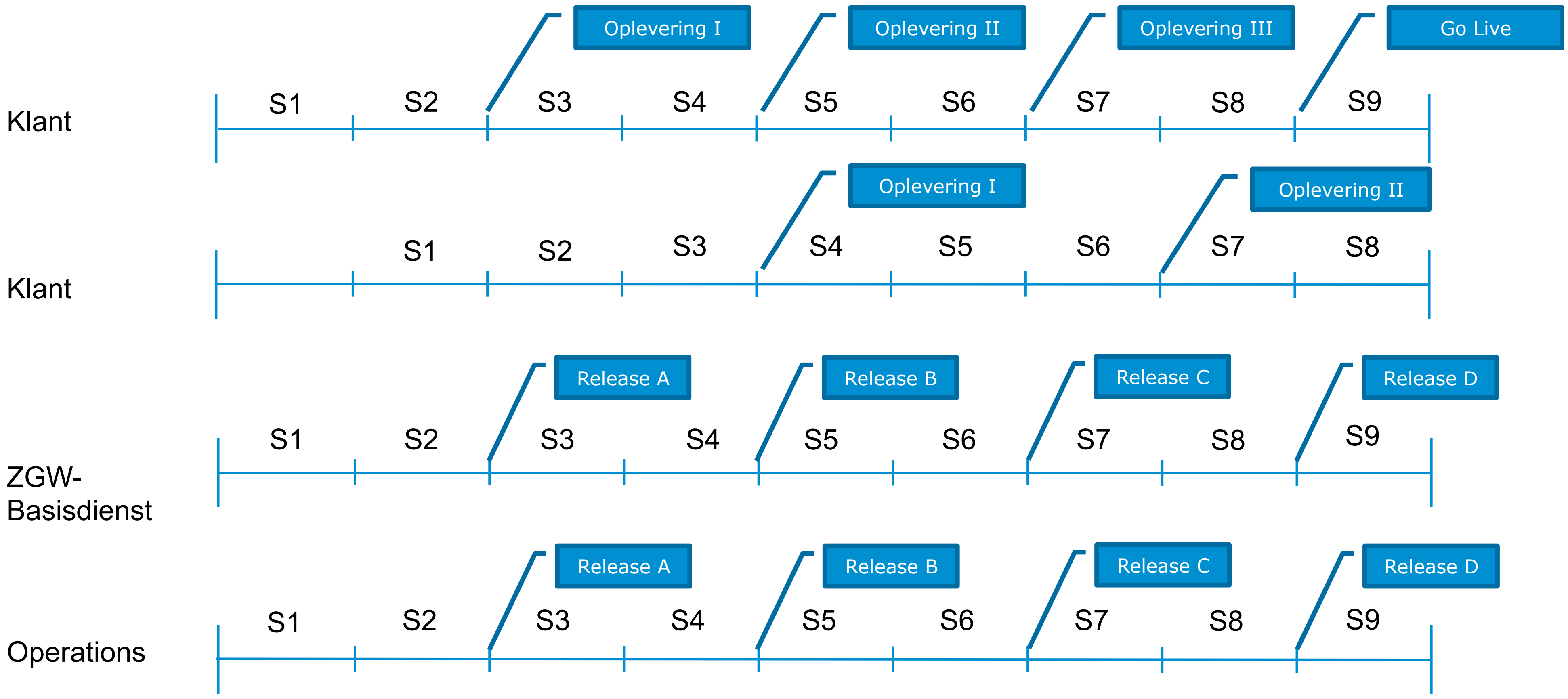
Tools

JIRA
DATPROF
TOSCA TESTSUITE
PEGA
NeoLoad
AppDynamics



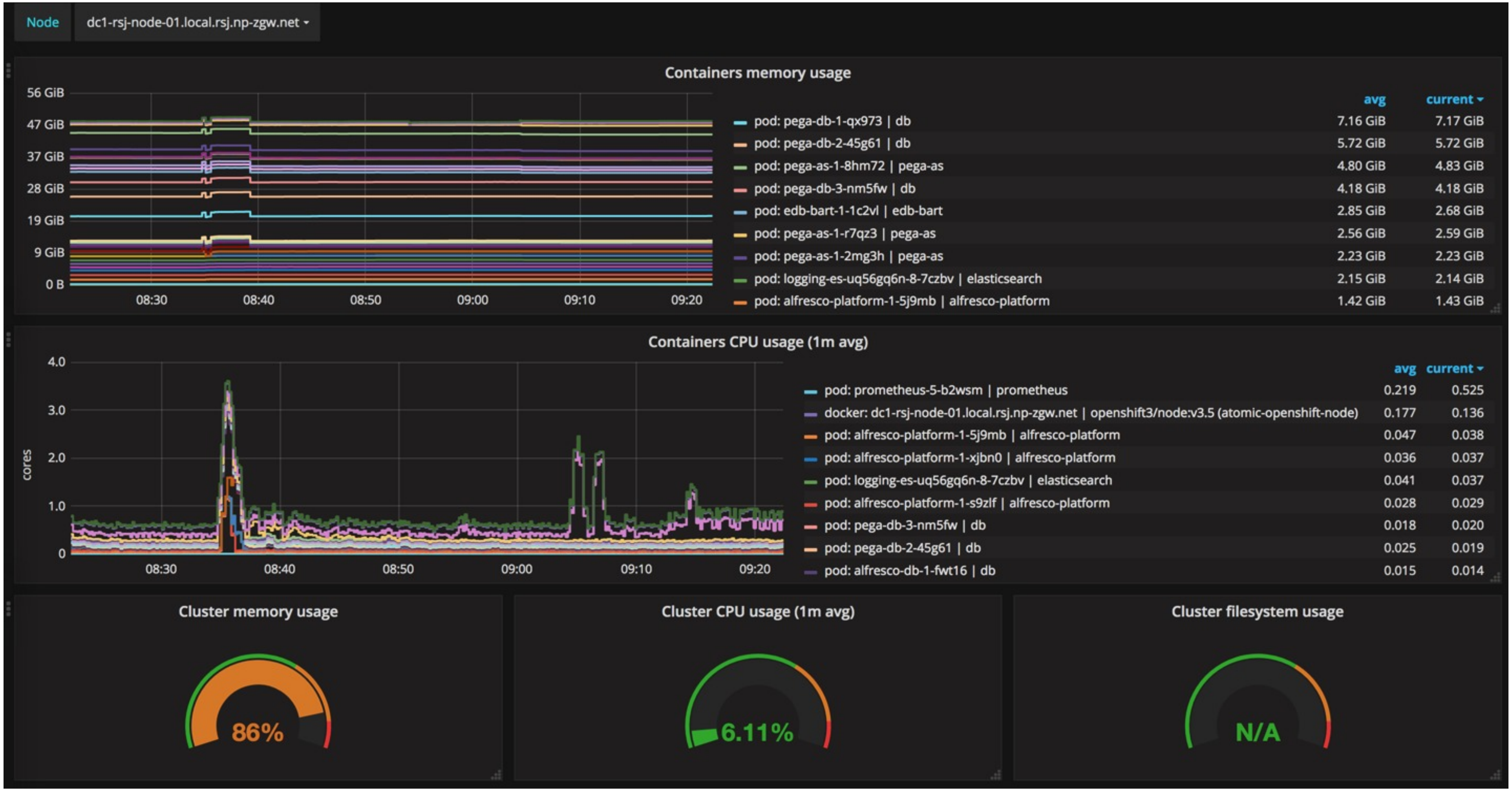
ISTQB International Software Testing Qualifications Board
TMap Next
Feedback Loops in Traditional Techniques vs. Agile Techniques
Cost
Length of Feedback Cycle
Programming defect found via Pair Programming
Programming defect found via Continuous Integration
Design or programming defect found via Test Driven Development (TDD)
Requirements or design defect found via Active Stakeholder Participation
Requirements or design defect found via Model Storming
Requirements defect found via traditional acceptance testing
Design defect found via traditional system testing
Programming defect found via traditional system testing
Defect found via a review or inspection
Defect found via independent parallel testing
Definition of Ready
Scrum Overview
Definition of Done
Product Backlog
Sprint Planning Meeting
Sprint Backlog
Daily Scrum
Development
"Done" Increment
Sprint Review
Sprint Retrospective
Product Owner
Development Team
Scrum Master
Scrum Team
Communication Effectiveness
Richness of Communication Channel
Cold
Hot
Paper
Audiotape
Documentation Options
Email conversation
Videotape
Phone conversation
Video conversation
Modeling Options
Face-to-face conversation
Face-to-face at whiteboard
2017
CONTINUOUS DELIVERY
CONTINUOUS DEPLOYMENT
TOSCA TESTSUITE
Pega
GUI
API Tests
Integration Tests
Component Tests
Unit Tests
DATPROF
2018

Releases



- Oplevering: Pull vanuit Klant team
- Release: Push vanuit Basisdienst ZGW

4. Demonstration



Node: dc1-rsj-node-01.local.rsj.np-zgw.net

Containers memory usage

	avg	current
pod: pega-db-1-qx973 | db	7.16 GiB	7.17 GiB
pod: pega-db-2-45g61 | db	5.72 GiB	5.72 GiB
pod: pega-as-1-8hm72 | pega-as	4.80 GiB	4.83 GiB
pod: pega-db-3-nm5fw | db	4.18 GiB	4.18 GiB
pod: edb-bart-1-1c2vl | edb-bart	2.85 GiB	2.68 GiB
pod: pega-as-1-r7qz3 | pega-as	2.56 GiB	2.59 GiB
pod: pega-as-1-2mg3h | pega-as	2.23 GiB	2.23 GiB
pod: logging-es-uq56gq6n-8-7czbv | elasticsearch	2.15 GiB	2.14 GiB
pod: alfresco-platform-1-5j9mb | alfresco-platform	1.42 GiB	1.43 GiB

Containers CPU usage (1m avg)

	avg	current
pod: prometheus-5-b2wsm | prometheus	0.219	0.525
docker: dc1-rsj-node-01.local.rsj.np-zgw.net | openshift3/node:v3.5 (atomic-openshift-node)	0.177	0.136
pod: alfresco-platform-1-5j9mb | alfresco-platform	0.047	0.038
pod: alfresco-platform-1-xjbn0 | alfresco-platform	0.036	0.037
pod: logging-es-uq56gq6n-8-7czbv | elasticsearch	0.041	0.037
pod: alfresco-platform-1-s9zlf | alfresco-platform	0.028	0.029
pod: pega-db-3-nm5fw | db	0.018	0.020
pod: pega-db-2-45g61 | db	0.025	0.019
pod: alfresco-db-1-fwt16 | db	0.015	0.014

Cluster memory usage

86%

Cluster CPU usage (1m avg)

6.11%

Cluster filesystem usage

N/A

5. Lessons Learned (Management/process)

- Start small, step by step (Agile way)
- Facilitating leadership
- Focus on People Management
- Use native language within program for project staff
- Coaching on the job a top level DevOps team as a mixed team off professionals with different individual qualities (infra, network, Linux, DBA, OpenShift container platform, application management PegaSystems, Alfresco, CI/CD)
- Use experiences other project (for example Quattro at ODC/N)
- Pay close attention to documenting and sharing of knowledge
- Working together with implementation partners (Trivento, Red Hat)

5. Lessons Learned (Technical/Products)

- Every 2th same manual activity will be scripted in OpenShift
- Start small, take it step by step
  1. Step 1: Create single node cluster 64GB/8 cores expand with additional nodes, no shared storage, so every pod attach to a particular host (advantage is less overhead to additional repositories, and monitoring dashboards)
  2. Step 2: Development step by step Multi node cluster with Cloud Native Storage (CNS)
- Backups to other data centre
- Principle ‘Eating your own dog food’
- Use correct OpenShift license (final goal is also pay per use)
- Use SSO components (RH SSO (=Keycloak), Ldap) rolled out as part of the ZGW template
- Use native clustering with EDB-Enterprise Server Edition
- Use the complete OpenShift subscription (for example RHEL/SSO)
- Use quotas memory, storage, CPU (for example Java 8 processes)

6. Questions