

# OpenShift Evolution bei Porsche Informatik

Keynote zum Openshift Anwendertreffen 25.02.2020

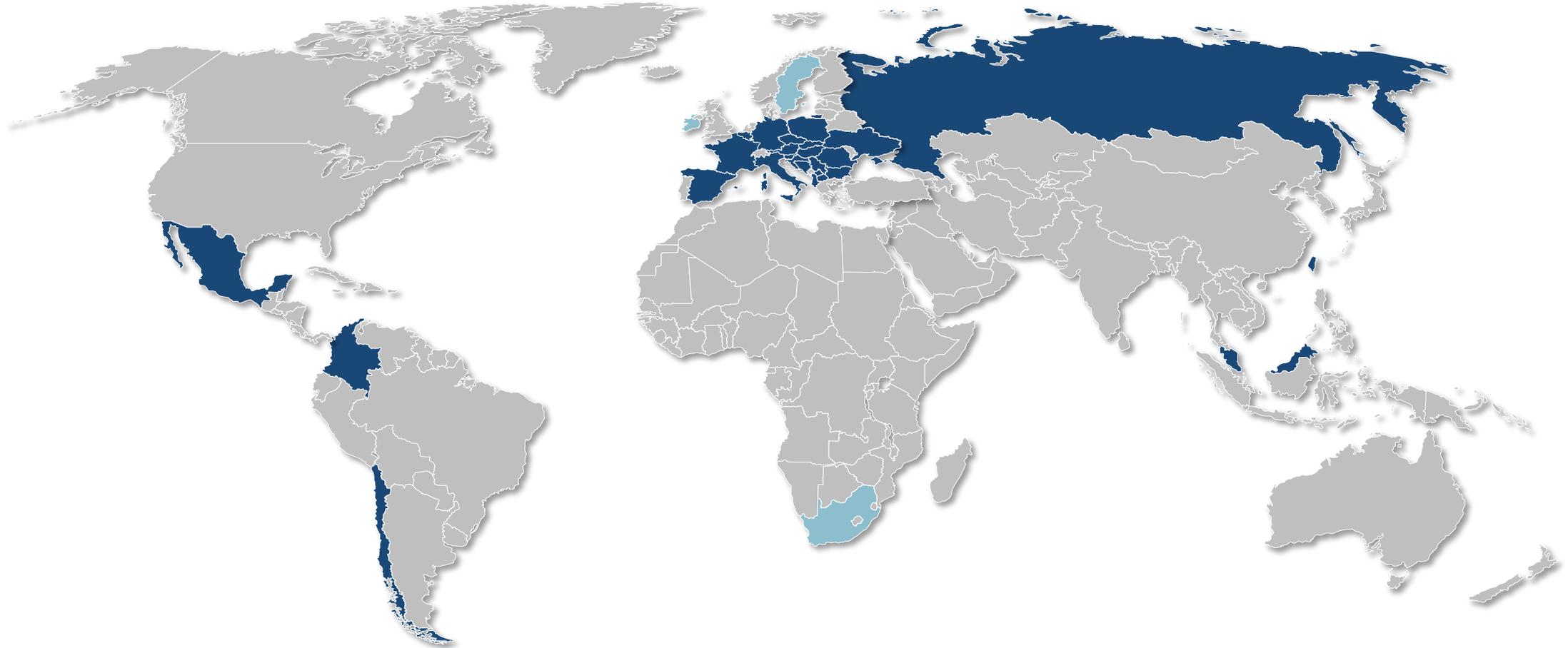


- Johannes Grumböck
- Seit 2001 bei Porsche Informatik
  - AIX Systems Engineer / Webadministration
  - Linux Systems Engineer / Webadministration / DNS
  - „Hans Dampf in allen Gassen“
  - Infrastruktur Architekt
  - 2015 erster Kontakt mit Openshift/Kubernetes
  - Seit 2017 Betrieb von Openshift
- Twitter: @jgrumboe
- LinkedIn: <https://www.linkedin.com/in/jgrumboe/>
- Credits go to: Günter Schulmeister (@Schulmeisterrr) für viele, viele Folien! Danke!



# Porsche-Informatik-Systeme in 30 Ländern auf 4 Kontinenten

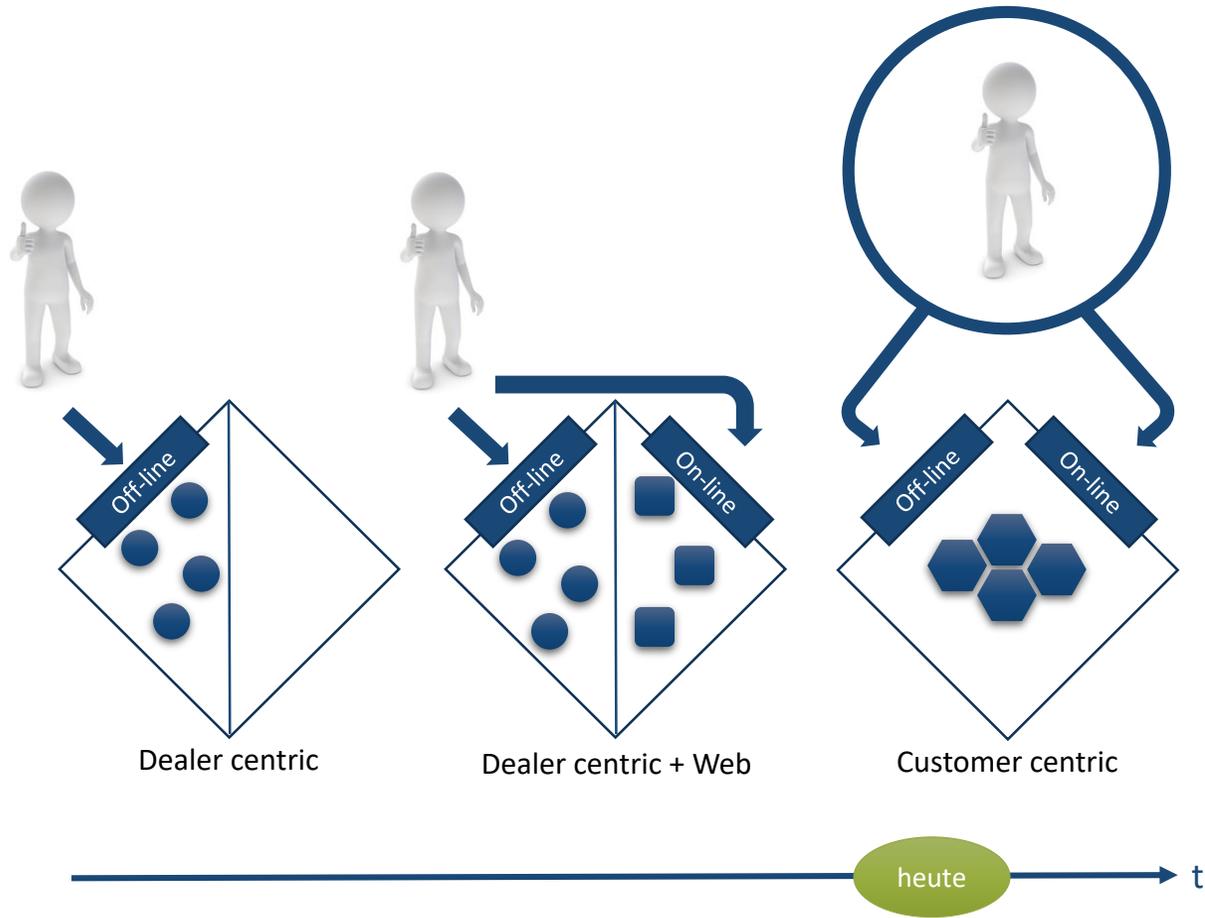
Laufende Projekte in drei weitere Ländern



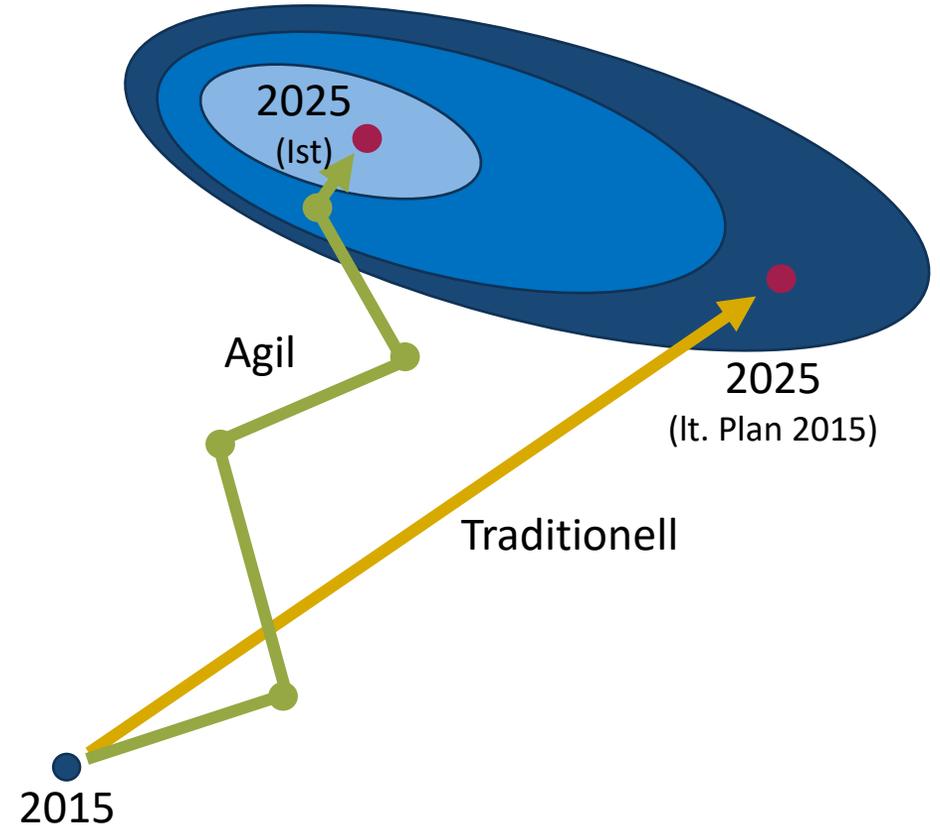
Porsche-Informatik-Systeme in  
Produktiveinsatz; laufende Projekte

Laufende Projekte

■ Von *Dealer centric* zu *Customer centric*



■ Agile Strategie



# “POI Cloud Journey” startet



- Strategy
- Private Cloud
- Public Cloud

Für uns ist

# CLOUD eine Plattform

zur Entwicklung und Betrieb

## horizontal skalierbarer Applikationen

Darüber hinaus sprechen wir von „Cloud Services“ wie z.B. Cloud Datenbanken,  
Cloud Storage (z.B. S3) oder Artificial Intelligence Services

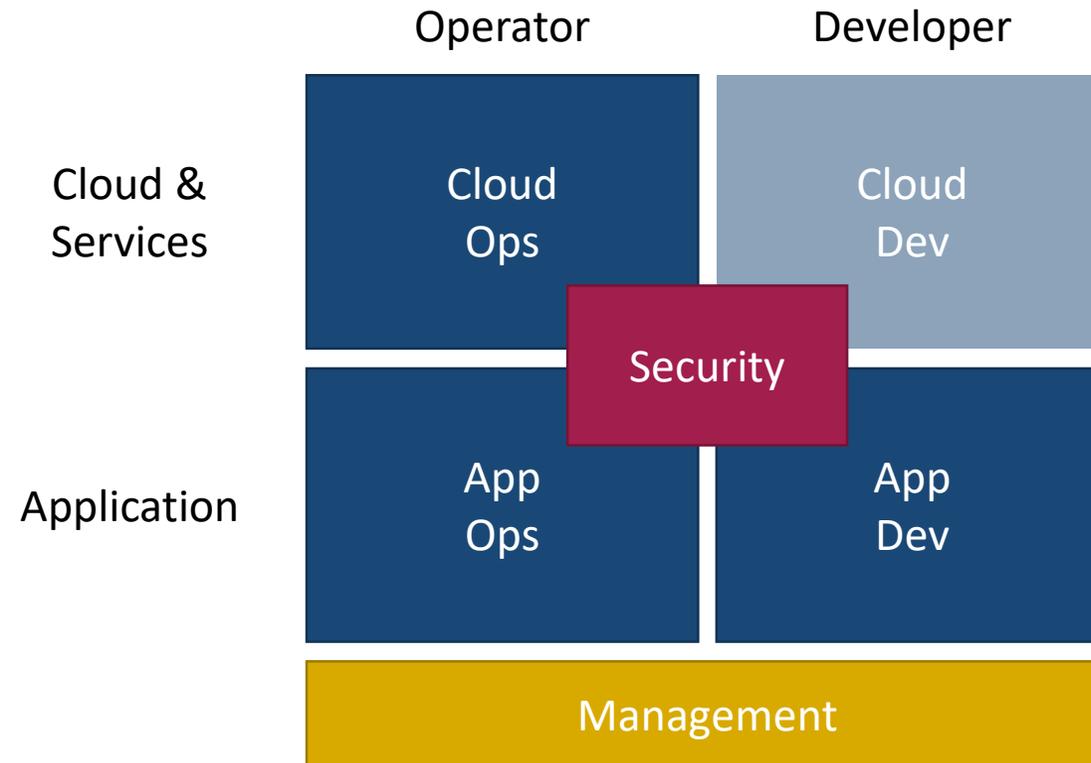
- Es müssen traditionelle Konzepte in Frage gestellt werden

- Applikationsentwicklung
- Betriebsführung
- Sicherheit
- Governance
- Geschäftsmodell

- Die Cloud betrifft alle

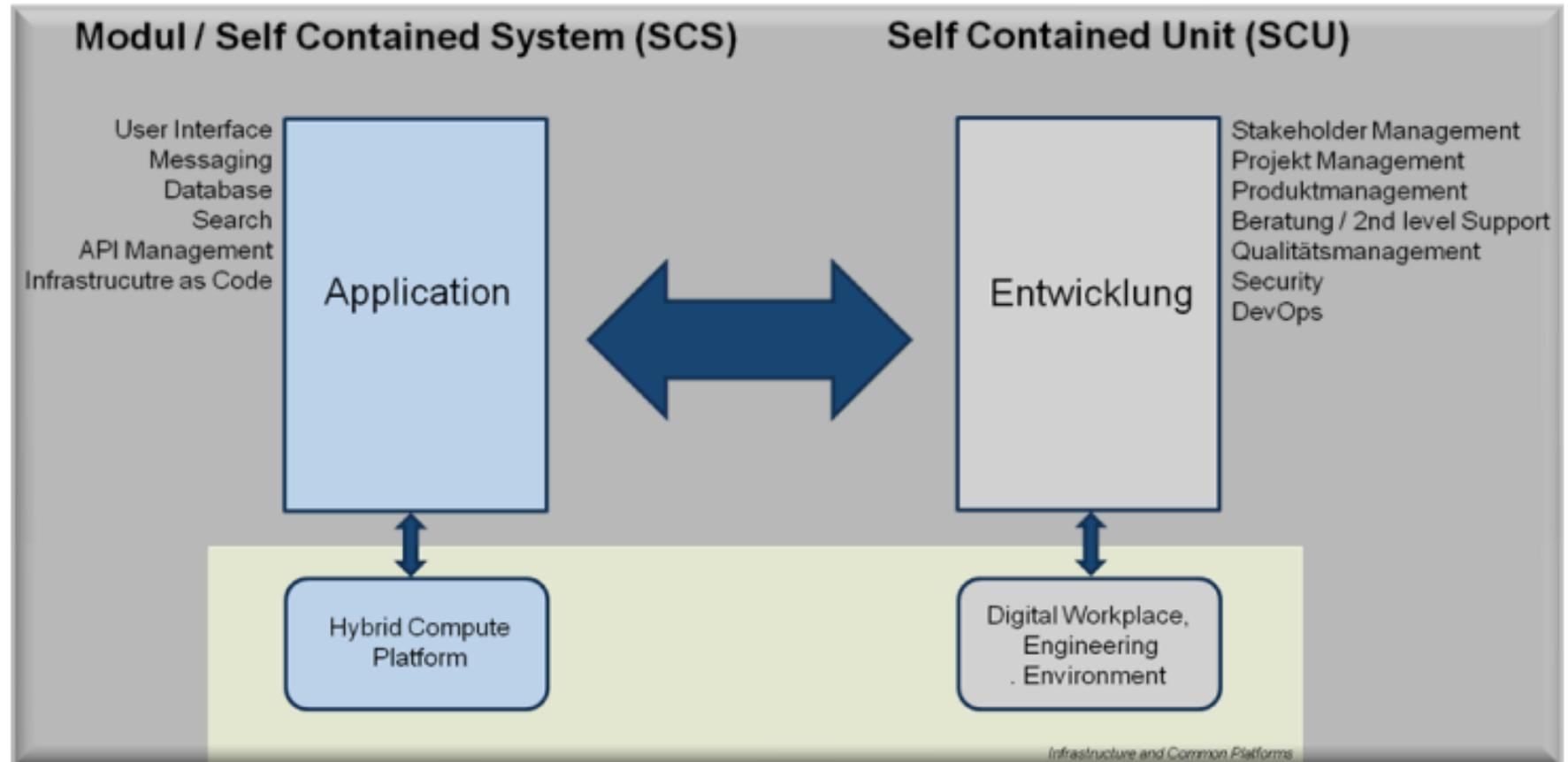
- CloudOps
- AppOps
- AppDev
- Security
- Management

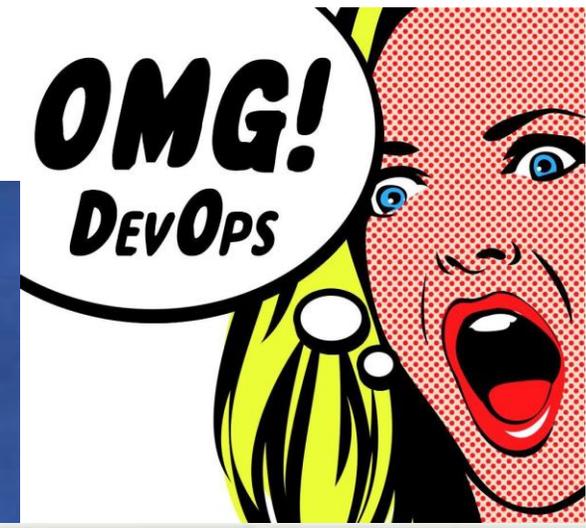
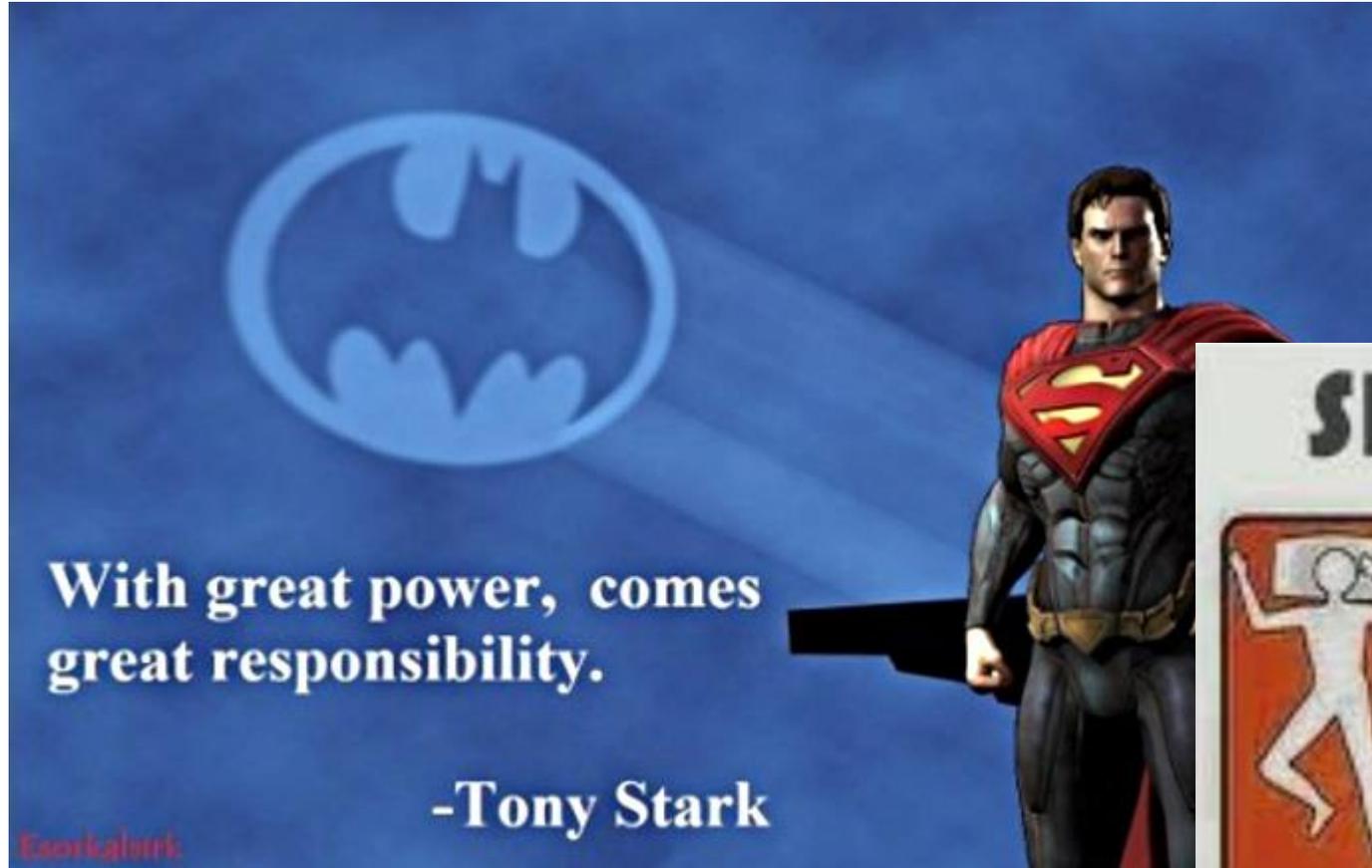
- Ein Cloudprojekt wird erfolgreich sein, wenn alle gleichberechtigt darin mitarbeiten

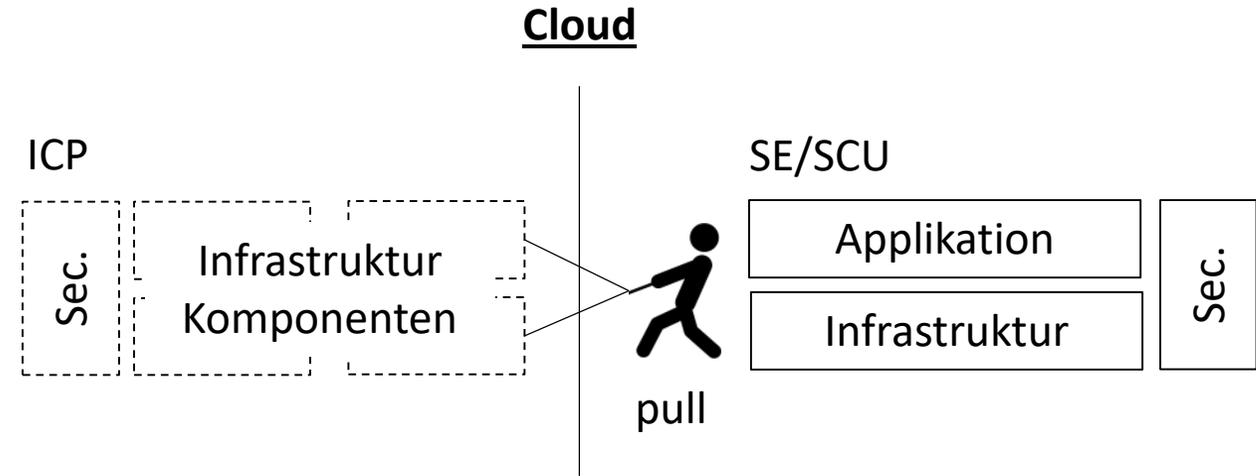
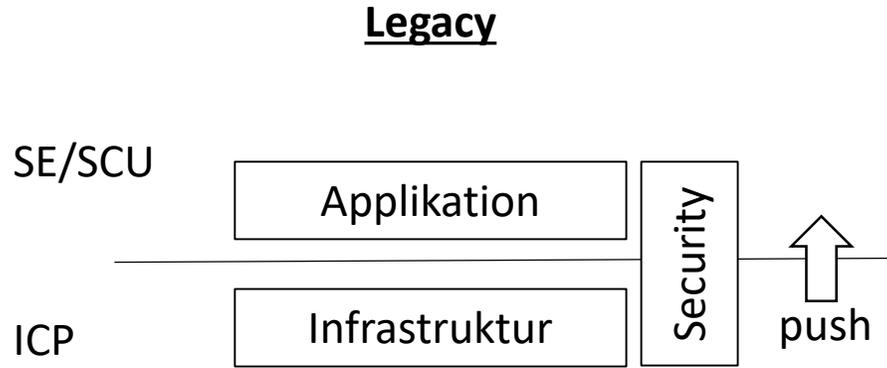


Teamstruktur reflektiert modulare Architektur

- Applikationsteams werden zu Self Contained Units (SCU)
- Sie betreuen Module / Self Contained Systems über deren ges. Lebenszyklus







## Betriebsmodell: Full Service

- auf Anforderung
- Individuallösung (für jede Applikation)
- Manueller Infrastrukturaufbau
- Langlebige, auf Applikation fixierte Infrastruktur

Beispiele:

Betrieb patched alle Server monatlich

Betrieb erstellt täglich Backups aller Daten(banken)

Betrieb stellt sicher, dass keine Daten ungewollt öffentlich zugänglich sind

## Betriebsmodell: Self Service

- Per API
- Standardisierte Komponenten
- Automatisierter Infrastrukturaufbau (IaC)
- Jedes Deploy generiert neue Infrastruktur

SCU deployed min. 1x im Monate mit aktuellen Images – es wird auditert (extern/intern), ob das auch passiert

SCU definiert, wovon ein Backup erstellt werden soll – ICP führt das Backup durch

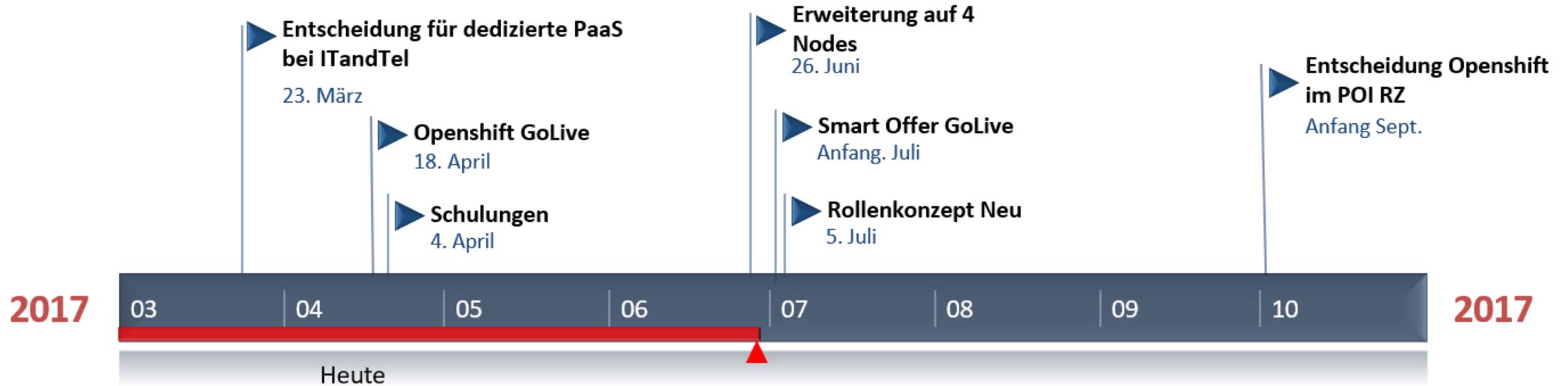
SCU definiert, welche Daten öffentlich zugänglich sind – diese werden regelmäßig auditert (extern/intern)

shift



shift





## Anbieter- vs. Systemauswahl

## Überführung in geregelten Betrieb

## Business Openshift im POI RZ

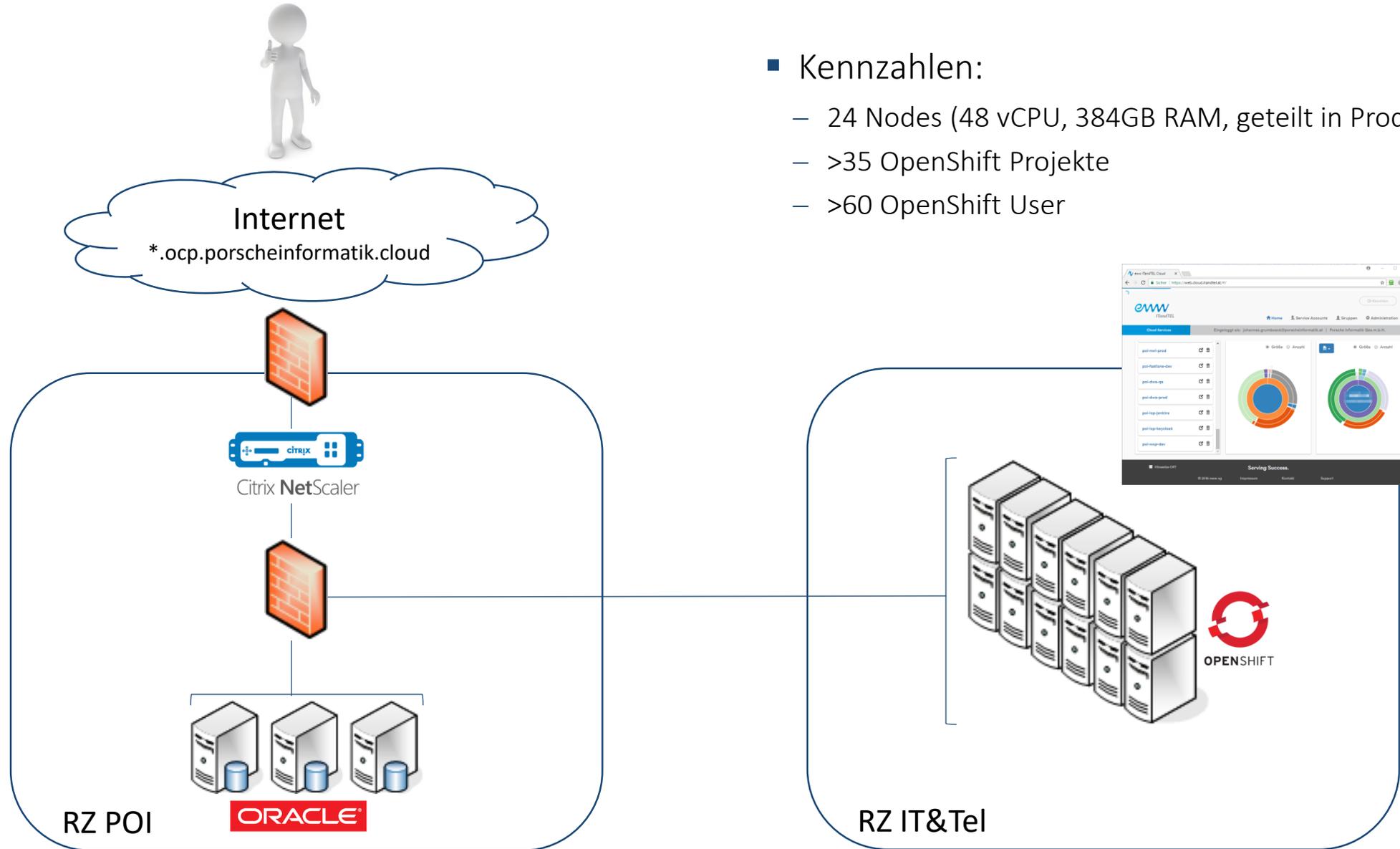
## Projektplanung Phase 2

### Driver:

- Applikationen mit Cloudtechnologie
- Openshift Know-How bei Provider
- Optimierung des Betriebs der Tomcat Umgebung

### Blocker:

- Fehlende Ressourcen
- Verbrauchsorientiertes vs. politisches Verrechnungsmodell
- Kosten CaaS/PaaS im POI RZ



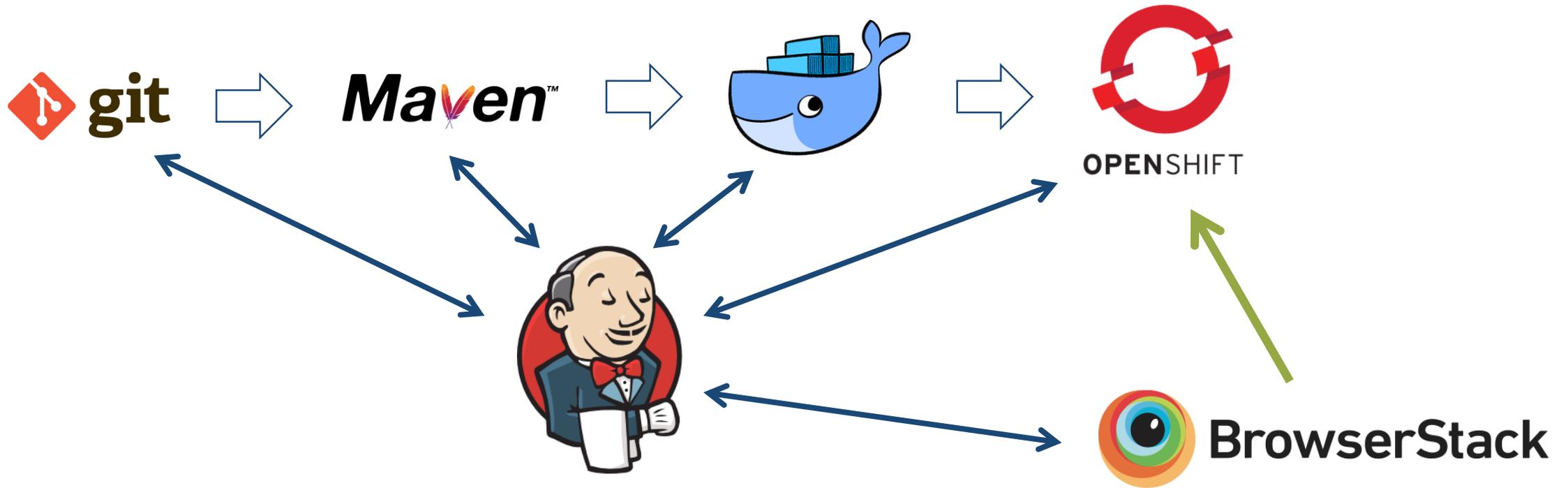
## ■ Kennzahlen:

- 24 Nodes (48 vCPU, 384GB RAM, geteilt in Prod/Non-Prod)
- >35 OpenShift Projekte
- >60 OpenShift User

- Einfaches Zero-Downtime Deployment - Continuous Delivery
- Neue Technologien können leichter angeboten werden
  - NodeJS, Ruby, Python
  - Andere Datenbanken (PostgreSQL, MongoDB, Cassandra, ...)
  - Caches (Redis, ...)
- Entwickler können komplett eigenständig Applikationen / Middleware aufsetzen ohne auf die darunterliegende Infrastruktur eingehen zu müssen

# OpenShift – Use Case 2

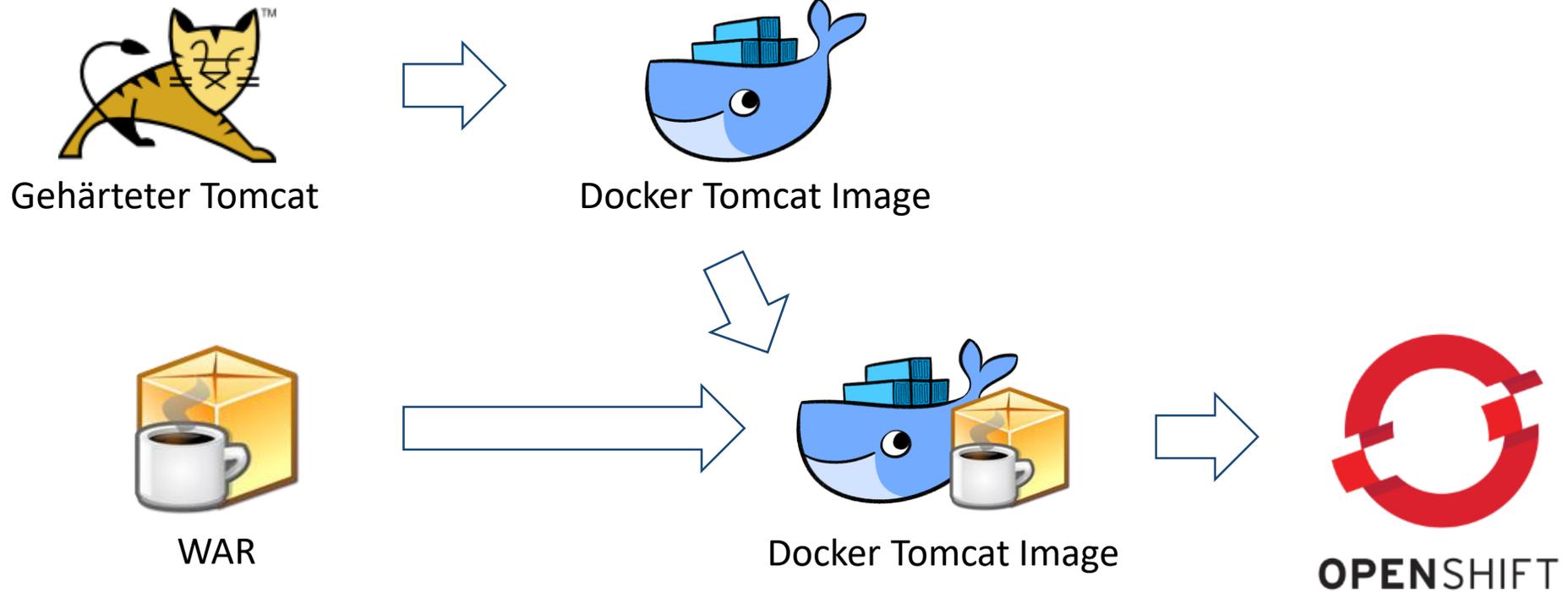
Testumgebungen / Automatisierte Tests



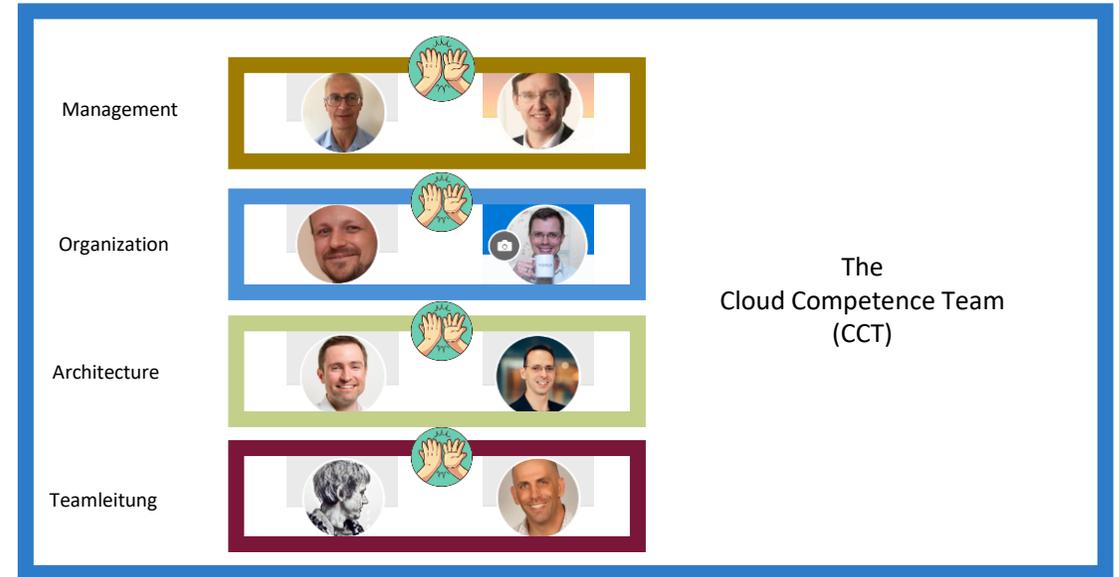
- Plan: Testumgebung je Change/Pull-Request

# OpenShift – Use Case 3

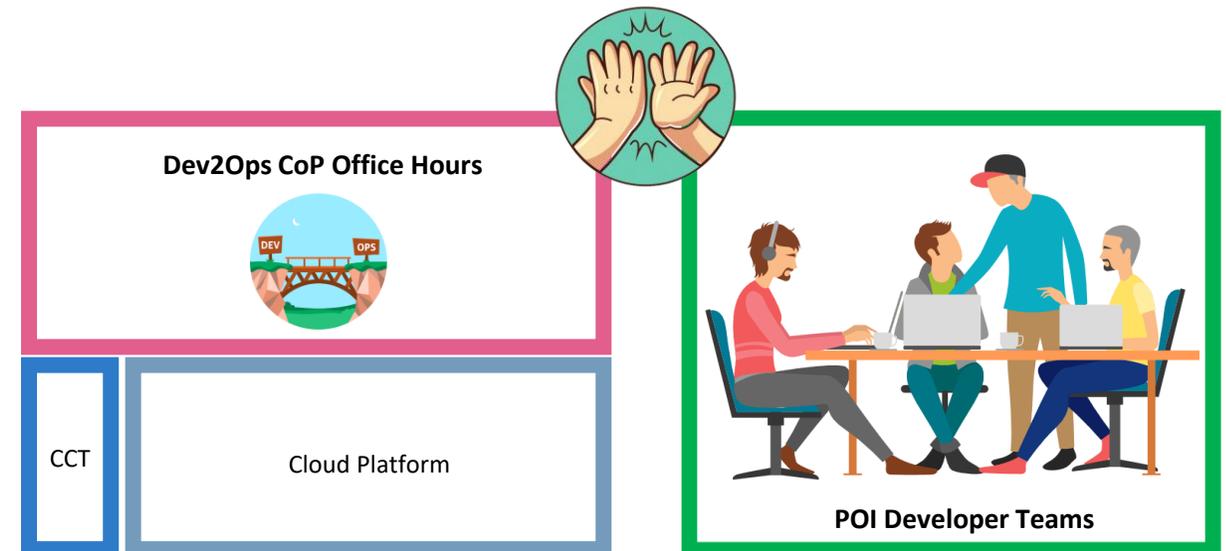
## Legacy Tomcat Applikationen



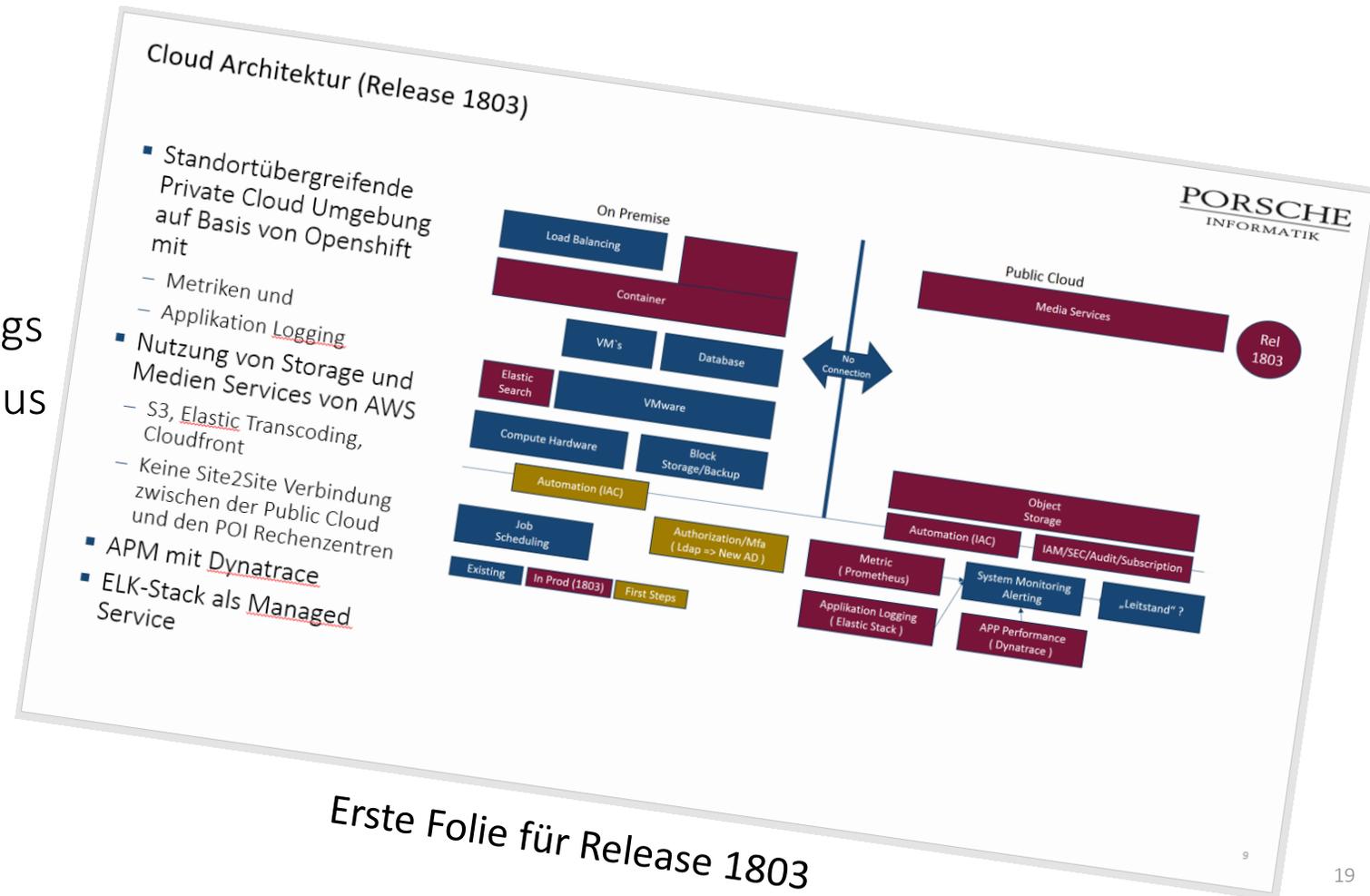
- Zusammenarbeit auf allen Ebenen
  - Cloud Competence Team
  - „Buddies“ im Sinne von DevOps



- Community of Practise „Dev2Ops“
  - Offen für Alle
  - Monatliche „Office Hour“
    - Online-Meeting mit vorher gemeinsam bestimmten Themenschwerpunkt
    - Video-Aufzeichnung für späteres Nachhören und Sammlung in Confluence



- Ist-Stand und Wünsche erheben
- „Was ist in den nächsten 6 Monaten wirklich wichtig?“
- Schrittweise Verbesserung der Cloud Platform durch „Releases“
- Releasebenennung analog zu Microsoft
  - Erstes Release „1803“
- 2-wöchentliche Team-Statusmeetings
- Monatliches Review des Projektstatus
- Backlog-Scrubbing



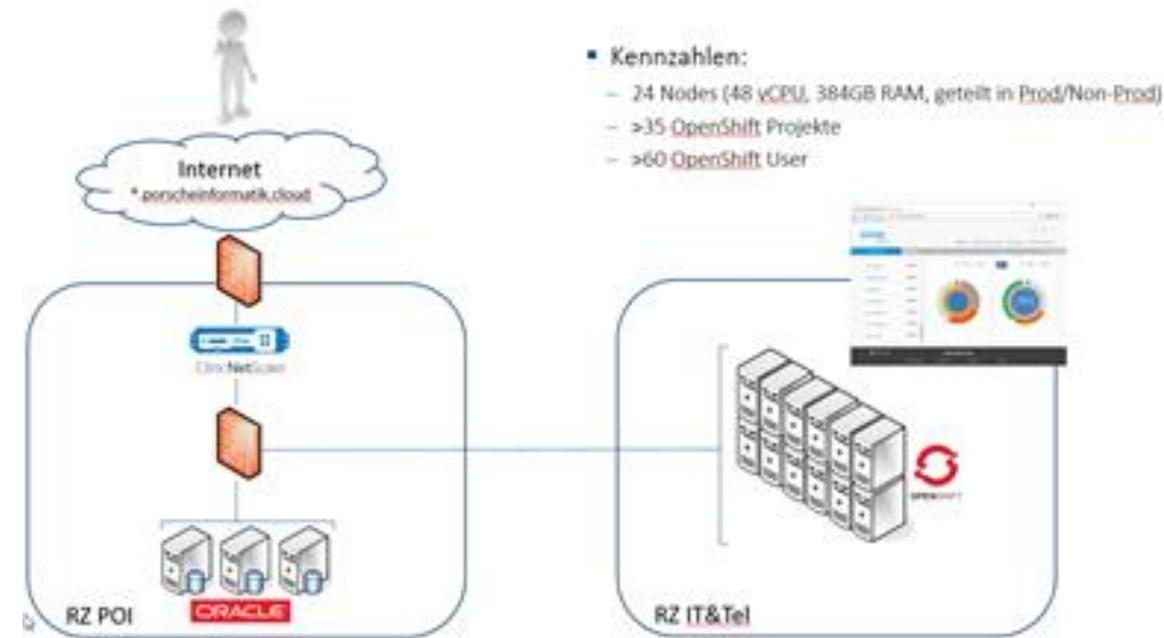
## Hosting und Betrieb bei ITandTEL

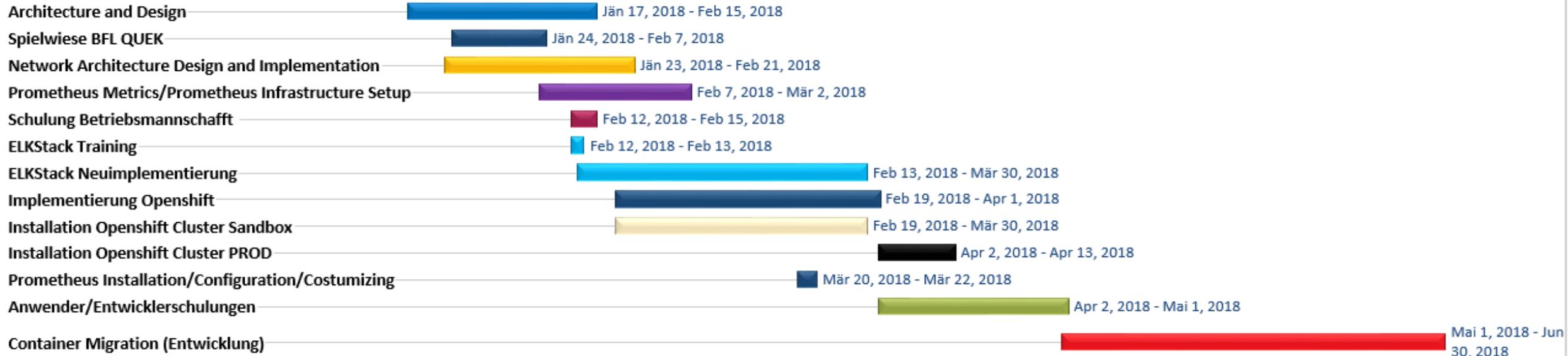
### ■ Vorteile:

- Geringe Komplexität für SCU
- Betriebsressourcen bei ITandTEL

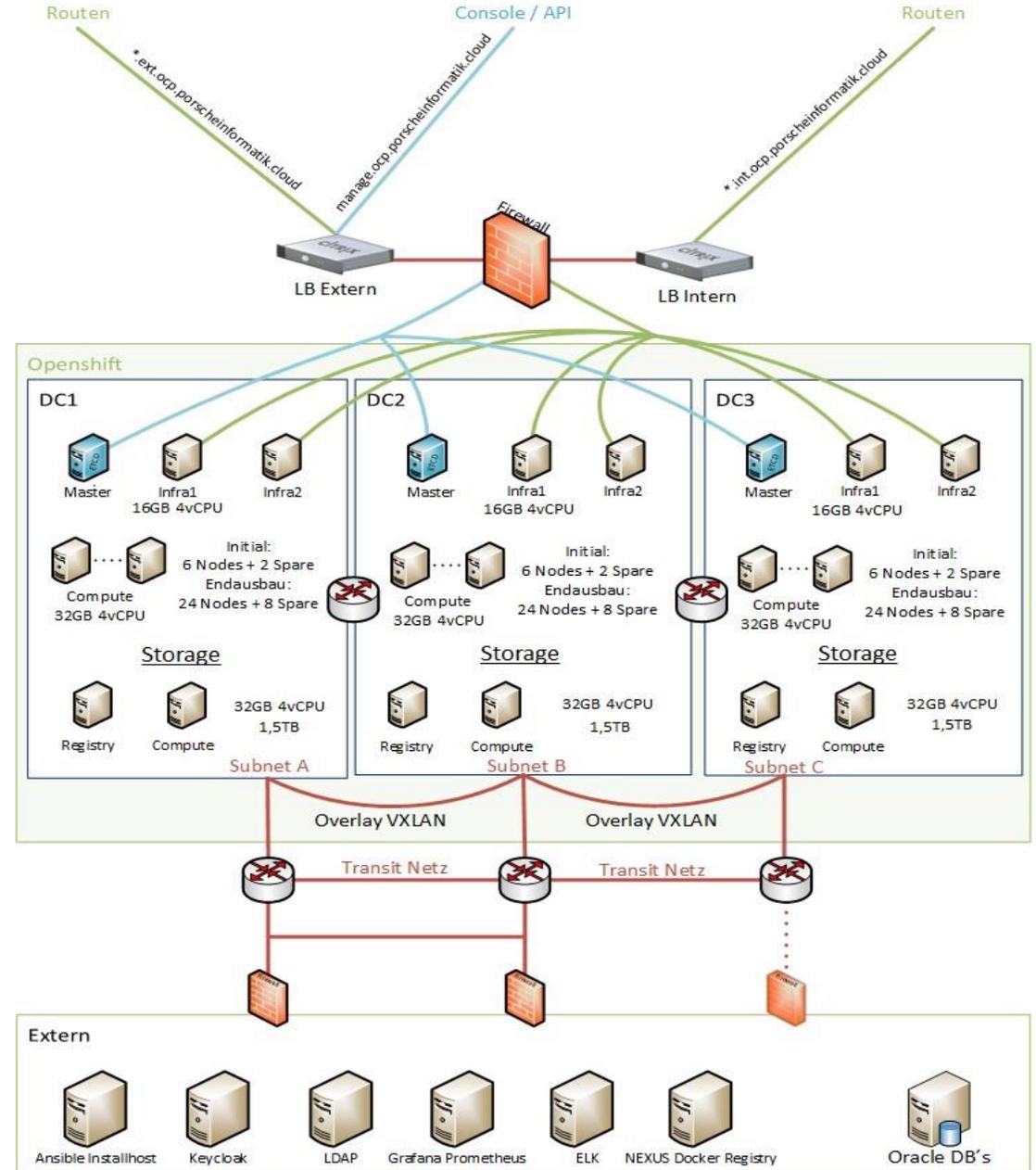
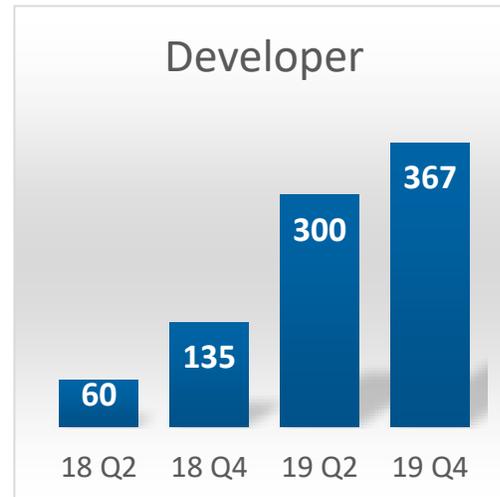
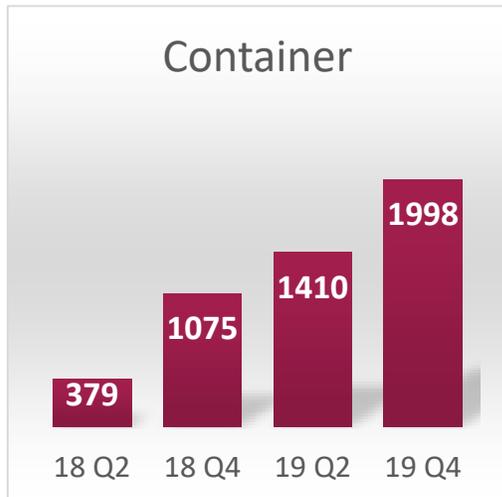
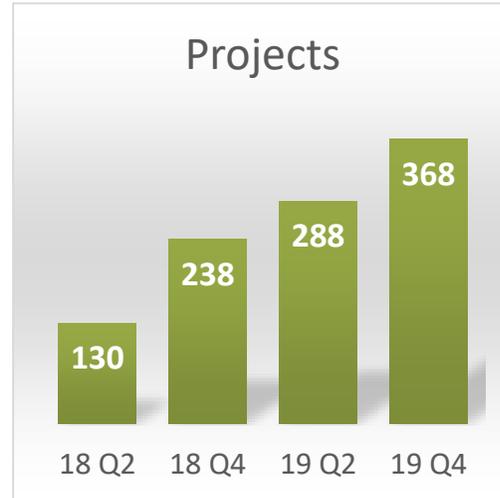
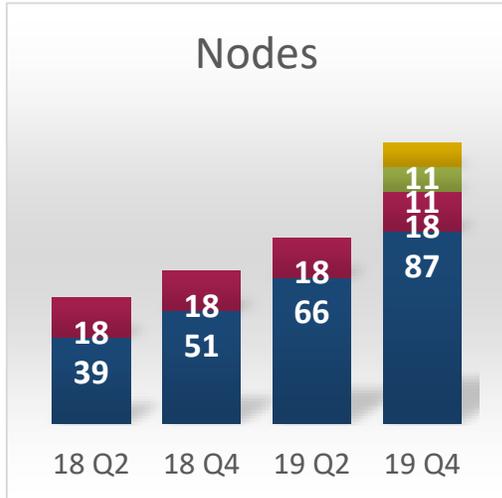
### ■ Nachteile:

- 6 ms Latenz (Roundtrip) zwischen Private Cloud und RZ Conova
  - Performanceeinschränkung bei Abfrage unserer Oracle Datenbanken
  - Eine Umgebung für alle Stages (Dev, QA, Prod)
  - Einige Komponenten werden mit anderen Kunden geteilt
    - Abhängigkeiten zu anderen Kunden
      - Manche Experimente und PoC sind nur schwer zu realisieren
- Keine „saubere“ Private Cloud





# Aktuelles Setup und Wachstum



# “POI Cloud Journey” zusammengefasst



Openshift POC  
Jän 4 - Mär 31

POI 4.0  
Nov 2 - Feb 28

Openshift hosted @ ITandTel

Mär 3 - Apr 18

AWS Jumpstart  
Sep 1 - Dez 31



Standardizing AWS Storage and Mediaservices

Jän 8 - Mai 31

Standardizing Azure for Compute

Feb 4 - Mai 31

Rampup Openshift OnPrem

Jän 8 - Mai 31

Container Security

Feb 4 - Mai 31

Migrate and Trainings

Mai 2 - Jun 30

"Day two operations" for AWS Storage and Mediaservices

Aug 1 - Nov 30

Continious Deployments

Aug 1 - Nov 30

Strategy

Private Cloud

Public Cloud

- Jährliches Kickoff Mitte März
- „Change Award“ fixes Element der neuen Strategie
  - Würdigt Leistungen, die die Strategie unterstützen und vorwärtsbringen
  - 5 Kategorien: Culture, Innovation, Quality, Time-to-Market, Manage-the-Growth
  - Einreichung
  - Nominierung der 3 Finalisten
  - Pitches vor Jury
  - „Oscar“-ähnliche Verleihung
- Gewinner der Kategorie „Time-to-Market“



## Plattform

- Single-Cluster für alle Stages nicht optimal. Mindestens Trennung zwischen NonProd und Prod!
- GlusterFS war/ist bis jetzt unsere „Achillesferse“
- Einige Komponenten der Applikationsinfrastruktur sind (noch) schwer zu managen.
  - Z.B. PostgreSQL, MongoDB, Redis, ...
- Teilen sich mehrere Teams den Cluster hat man bei der Nutzung der Ressourcen ein Optimierungsproblem.
  - Z.B. bei Vergabe von Request und Limit, Patchzeitpunkt,
- Eine Plattform bedeutet man hat mehr Zentrale Systeme.
  - HA wird wichtiger, da der Ausfall eines dieser Systeme einen großen Teil der Applikationen betreffen kann

## Applikation

- Einige traditionellen Frameworks und Komponenten nicht 100% für Container und Microservices geeignet. Z.B.
  - Java aufgrund des hohen Ressourcenbedarfs beim Start
  - ActiveMQ aufgrund geringen Durchsatz bei Persistenz
  - Tapestry aufgrund des hohen Ressourcenbedarfs
- Es braucht ein Konzept für Image Streams. Ansonsten schlägt ein Redeployment fehl oder es wird die falsche Version eingesetzt
- Bis die Plattform reif ist, muss sich die Entwicklung um mehr kümmern
  - Updates/Patches (von Images), Backup/Restore von Daten, ...
  - Verantwortung verschiebt sich ("Shift left")

Die technologische Basis ist geschaffen, aber „Developer Enablement“ noch (lange) nicht abgeschlossen

- Training der Applikationsteams

- neue Funktionen und neue Verantwortlichkeiten



- Nutzung der neuen Möglichkeiten von Infrastructure as Code

- Jedes Deploy generiert neue Infrastruktur



- Bessere Unterstützung für Applikationsteams

- Ausbau der Plattform: Container Security Scanning, Managed PostgreSQL, Secret Vault, Metriken und Dashboards, Cluster Update Konzept, ..
- Ausbau Plattform Teams (Cloud & Automation, Engineering Environment)



- Entwicklung resilienter cloudfähiger Applikationen

- Containerisierung ist erst der Anfang



DIE REISE GEHT WEITER...

