Agenda

Introduction Cloud Technology

Some concepts of SAP Cloud Platform

Some implications for SAP Cloud ALM
Introduction

Cloud Technology
Cloud Benefits for Customers

- Unlimited scalability
- Always available
- Access from anywhere
- Always up to date
- Maximum efficiency
- No IT maintenance costs

Why Cloud?
Cloud Service Models

- **Software as a service (SaaS)**
  - Provide a complete application to end users. Customization and own user management possible.
  - ByDesign, SuccessFactors, FieldGlass, Cloud 4 Customer, Ariba, Hybris, Concur, S/4 Hana Cloud
  - SAP Cloud Application Management (CALM)

- **Platform as a service (PaaS)**
  - Development and lifecycle management platform including core components (e.g. DB) for building new apps.
  - SAP Cloud Platform (for customers extending SAP solutions, for custom developments, as a platform for SAP SaaS solutions)
  - Amazon Web Services, Google Cloud Platform, Microsoft Azure, Alibaba Cloud

- **Infrastructure as a service (IaaS)**
  - Provide processing, storage and network. OS and all on top is managed by the customer.
  - Amazon Elastic Compute Cloud, Google Compute Engine, Microsoft Azure Virtual Machines
Example: Data Centers for SAP Cloud Platform

- Alibaba planned for 2020
- Not all services are offered in each data center
  - Cloud Foundry isn’t offered in SAP data centers
  - CALM starting in AWS Frankfurt, expansion driven by demand
Example: Support & Deployment for SAP Cloud Platform

Sufficient monitoring and support to avoid unplanned down times

- SCP Cloud Cockpit
- SCP Job Scheduler
- SCP HANA Tools
- SCP Audit Log Viewer
- Performance Monitor (Dynatrace)
- Log Viewer (Kibana)

Temporary operation of as-is and to-be code level to avoid planned down times

1. Deployment
2. Log Out
3. Log In

True zero down time powered by Deploy with Confidence

Always available
Example: Monolith vs. Micro service based Architecture

Monolith Architecture

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Micro service based Architecture

<table>
<thead>
<tr>
<th>A</th>
<th>A</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>

Each micro service runs in **separate process / container**:

- Be deployed individually and frequently
- Scale independantly to varying loads (if possible automatically)
Example: Deploy with confidence (DwC)

“Develop features within distributed applications with ease and deliver them with confidence on a daily basis!”

- Optimized for cloud-native microservice applications
- Focus on fast and frequent delivery, powered by:
  - Dynamic Request Routing
  - Testing in production like environments
  - Feature Toggle Management
  - True zero-downtime for consumers
- Includes Best-practice templates for:
  - Onboarding
  - Business Configuration
  - Documentation/user assistance

© 2019 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC
Example: SAP Cloud Application Programming Model

complementing cloud-native platform…

- Large variety of open standards and SAP technologies
- How to combine between the technologies and solutions?
- Rather technical focus → no quick start
- Each project has to address platform integration and enterprise qualities individually
- High rates of boilerplate coding → slow time-to-market and hard to fix/evolve

with business domain focus

- Clear guidance to seamlessly integrated best practices…
- Common patterns and reuse models → interoperability
- Conceptual modeling → re-focus on business domains
- Capturing intent → what not how; single points to fix
- Automating tedious tasks → multi-tenancy, extensibility, authorization, audit logging, data privacy, localization, platform integration, …
Some concepts of SAP Cloud Platform
SAP Cloud Platform - High level architecture

Cloud Applications
- SAP Applications
- Customer-Specific Applications
- 3rd-Party Applications

Open Marketplace / Open APIs

Business Services

Technical Services

Data Foundation

Environments

Multi-cloud

2) planned innovations / future direction
SaaS (Software-as-a-Service) focuses on implementing valuable services in problem domain

PaaS (Platform-as-a-Service) manages (configures, deploys, scales, upgrades) highly available applications

IaaS (Infrastructure-as-a-Service) offers on-demand seemingly infinite resources and seamless failover
SAP Cloud Platform - Multi-Tenancy

One application instance can serve requests from multiple customers.

**Definition**

**Multitenancy** refers to a software architecture in which multiple customers (a.k.a. tenants) share the same technical resources while keeping Identity and Access Management (IAM) and data separated.
1. In SAP CP CF, each tenant has a separate **Identity Zone in SAP UAA**. This allows isolated User Management per tenant.

Each customer (SaaS tenant) and application provider (PaaS tenant) requires a **Subaccount** to attach his **IdPs** via his corresponding **Identity Zone**.
SAP Cloud Platform - Tenant Data Separation Options

**Discriminator Column**

One DB instance with one DB schema shared by all tenants.
Tenant separation by discriminator column in each table
Tenant id to be entered in each table access statement. Same DB user for all tenants.
Tenant move is difficult.
Resources (CPU, memory, disk) shared by all tenants.

**Schema Separation**

One DB instance with separate DB schema per tenant.
Separate DB users per schema with access to own schema only.
Resources (CPU, memory, disk) shared by all tenants.
Tenant move is relatively easy.

In SCP this type of separation is especially supported by HDI containers (HDI = Hana Deployment Infrastructure).
Each schema shown above is managed by one HDI container.

**Instance Separation**

One DB instance with separate DB schema per tenant.
Separate DB users per schema with access to own schema only.
Tenant move is very easy.
Private resources (CPU, memory, disk) per tenant.
Some implications for SAP Cloud ALM
SAP Cloud ALM – Selected development paradigm

Startup Developer

General Enterprise Cloud Developer

Open

- Other (React,...)
- Other (Spring,...)
- Other (Postgres,...)
- Integrate with
- Other (Kubernetes,...)

Integrate with

Other (React,...)

Other (Spring,...)

Other (Postgres,...)

Integrate with

Other (Kubernetes,...)

SAP Cloud ALM

– Selected development paradigm

Citizen Developer

ABAP developer

Opinionated

Build on

SAP Cloud Platform Foundation

Cloud Foundry

SAP Fiori/UIS

OData, REST

OData Libraries (Java, Node)

Core Data Services

Java/Node Frameworks

Cloud App. Programming Model

SQL

SAP HANA

DevOps + LCM

Extension Services
Events
Models
Replication
Interoperability
Reuse of Skills

Build on

SAP Fiori/UIS

OData, REST

OData Protocol (ABAP)

Core Data Services

ABAP Frameworks

ABAP Development

ABAP RESTful Programming Model

SQL

SAP HANA

Technical Environment chosen for SAP Cloud ALM

1) planned innovations / future direction
Every customer with appropriate Enterprise Support / Cloud Subscription contract will be entitled to use SAP Cloud ALM (limited by resource quota).

If customer decides to use SAP Cloud ALM appropriate CALM Global Customer account and CALM Customer sub account will be generated (independently from existing SCP accounts).

CALM Customer Sub Account subscribes to CALM application running in one specific CALM Provider Sub Account.

CALM Global Provider Account and associated CALM Provider Sub Accounts are owned by SAP.

CALM Provider Sub Accounts are specific for one SCP data center and represents the actual physical resources ➔ All other accounts have pure logical character.

Account structure chosen for SAP Cloud ALM
SAP Cloud ALM - Realization of Resilience on Persistency Level

**Without Persistency Resilience**
- All use cases and their microservices use the same schema (HDI container).
- If this schema (resp. the hosting DB instance) is not available no use case is working.

**With Persistency Resilience**
- For each use case and reuse services with own persistency use separate schemas (HDI containers) for their persistency.
- This enables resilience from development level (independent logical persistency stores). For achieving "physical" resilience those separate use case specific schemas have to be located within separate DB instances.
- The resource overhead of an HDI container is very low. Moving all customers data from one place to another (e.g. to another data center) will be more difficult with separate schemas, however automation is requested.

Persistency model chosen for SAP Cloud ALM
The SAP Cloud Application Programming Model is an integrated **framework of tools, languages, and libraries** to efficiently and rapidly build enterprise services and applications in a **full-stack development** approach.

It guides developers along a **golden path of best practices**, allowing them to **focus on their domain** while relieving them from tedious technical tasks.

**Programming model chosen for SAP Cloud ALM**
SAP Cloud ALM for Operations – Semantical Services

Cross use case services
- End-to-End Event & Alert Management Service
- End-to-End Operation Automation Service
- End-to-End Analytics & Intelligence Service
- Business Service Management Service

Use case services
- Integration Monitoring Service
- Business Process Monitoring Service
- Synthetic User Monitoring Service
- Data Consistency Management Service
- System Monitoring Service
- Real User Monitoring Service
- Business Process Improvement Service
- Job Monitoring Service
- Configuration Monitoring Service
- Root Cause Analysis Service

Re-use services
- Exception Monitoring Service
- Performance Monitoring Service
- Availability Monitoring Service

Infrastructure services
- Landscape Management Service
- Notification Management Service
- Data Collection Pull Service
- Data Collection Push Service
SAP Cloud ALM for Operations – High Level Architecture
Join our social media family

Follow us on Twitter (#solman and #CloudALM), get the latest news, learn about support offerings and events!

Search for “SAP Cloud ALM” on YouTube for product demos, expert interviews and event insights!

Subscribe to WhatsApp SAP Product Support channels and receive updates including KBAs, wiki's, guided answers, SAP Notes and “hot tips”.

Join the SAP Digital Business Services LinkedIn group for networking and updates!
Thank you.

Contact information:

Janko Budzisch
Chief Product Owner
janko.budzisch@sap.com

Christian Niedermayer
Cross Architect
christian.niedermayer@sap.com