Aus der Debian Appliance in die Deutsche Verwaltungscloud - behalten oder neu machen?

FrOSCon 2023

Ingo Steuwer
steuwer@univention.de

univention

Agenda

1. Der Rahmen
   a) Was macht Univention?
   b) Was ist die „Deutsche Verwaltungscloud-Strategie“, was braucht es sonst noch?
2. Wir haben das immer so gemacht, was machen wir jetzt?
   a) Was behalten wir, was machen wir anders
   b) Was soll dabei raus kommen
3. Wo stehen wir?

Disclaimer

- Thematisch wird es breit, nicht tief
- Es gibt Auslassungen / Vereinfachungen an allen möglichen Stellen

- Ich freue mich auf Fragen und Diskussion :-)

Der Rahmen: Wo stehen wir?

Univention GmbH

- Open Source Software Hersteller: 100% OSS
- „be open“
  - Open Source
  - Offene Menschen
  - Offene Unternehmenskultur
- Gegründet in Bremen, ~100 Menschen in Europa

  www.univention.de/ueber-uns/karriere/

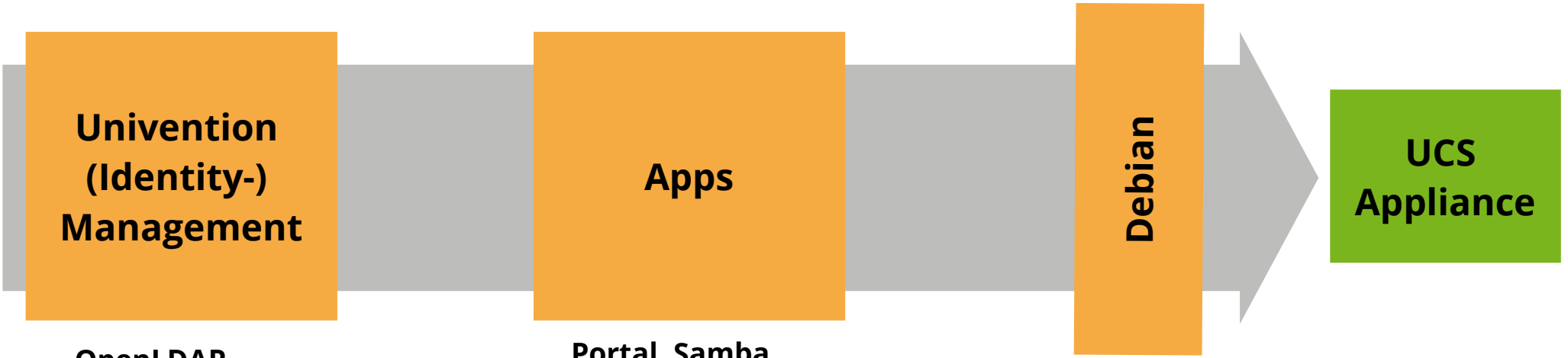
„Univention Corporate Server“

- Vereinheitlichtes Benutzer- und Berechtigungsmanagement („IAM“)
- Einfacher Zugang für Nutzende (Login, Portal, Self Service)
- Integrations-Schnittstellen: Single Sign-On, Provisionierung, Deployment, …
- Einfaches Deployment bei hoher Skalierbarkeit (Multi-Instanzumgebung)
- Standard-Integrationen mit vielen OpenSource Anwendungen
  - Nextcloud, ownCloud, OpenXchange, OpenProject, XWiki, Jitsi, …

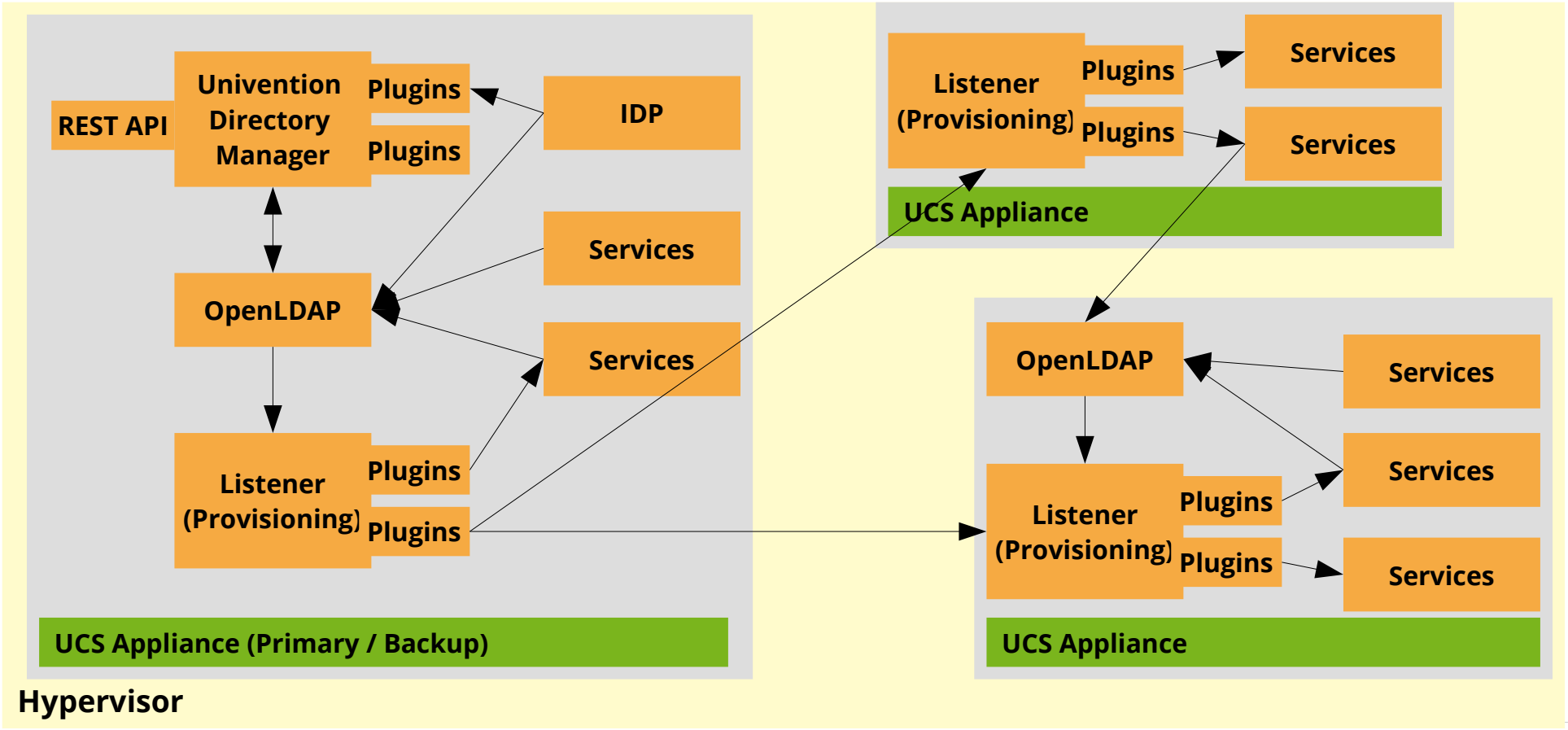
„Univention Corporate Server“

- Technisch:
  - Software Appliance basierend auf Debian
  - Viel eigene Paketierung (Debian, Container) für einfache Vorkonfiguration
  - Darauf aufbauende Eigenentwicklungen für Management von Usern, Infrastruktur, Services, Portal
  - Darauf aufbauendes „App Center“ für Integration und Deployment anderer Software
- Stichworte:
  - OpenLDAP, Samba, Keycloak, Apache, BIND, Cups, Python, ...
  - Kerberos, SAML, OpenID Connect, DNS, DHCP, ...

Bausteine UCS



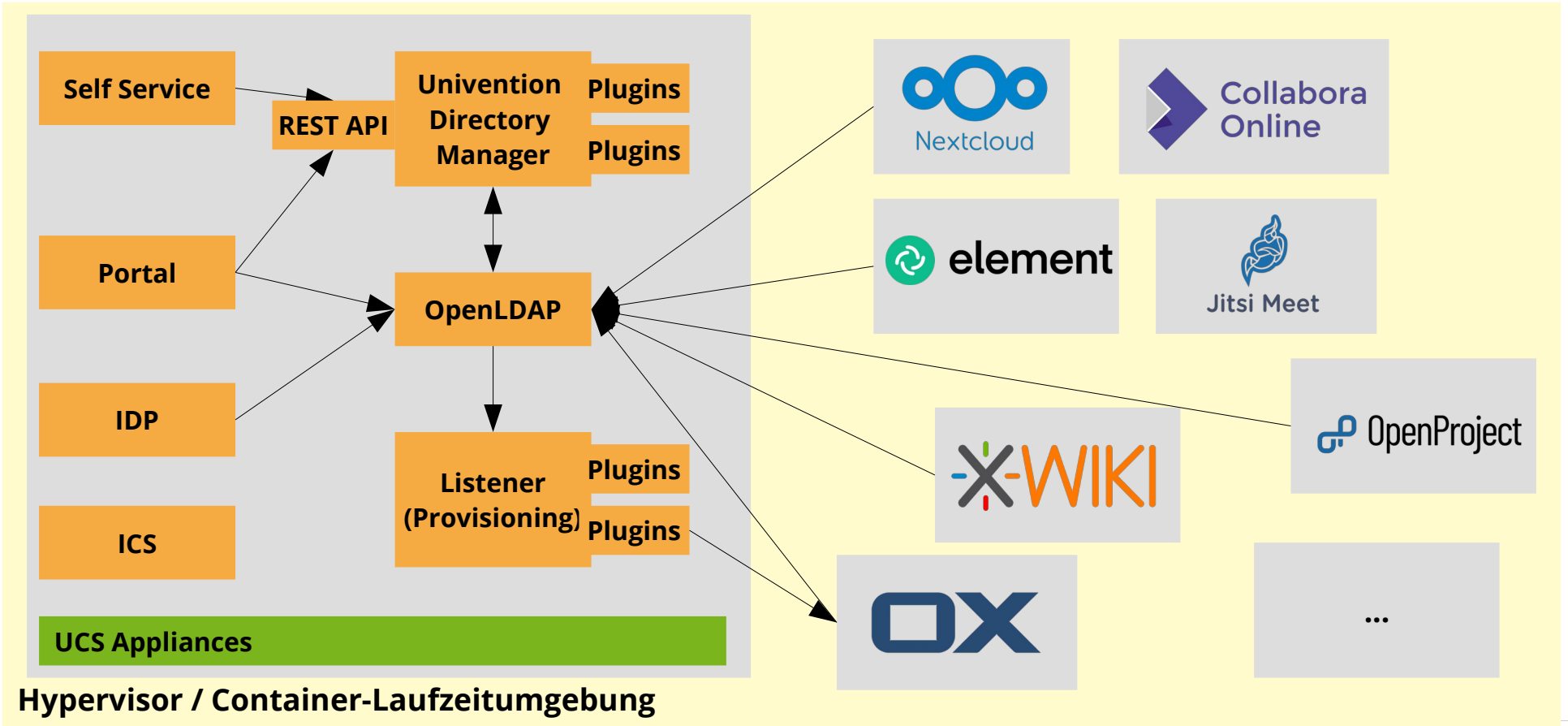
UCS – Module: Beispiel Skalierung



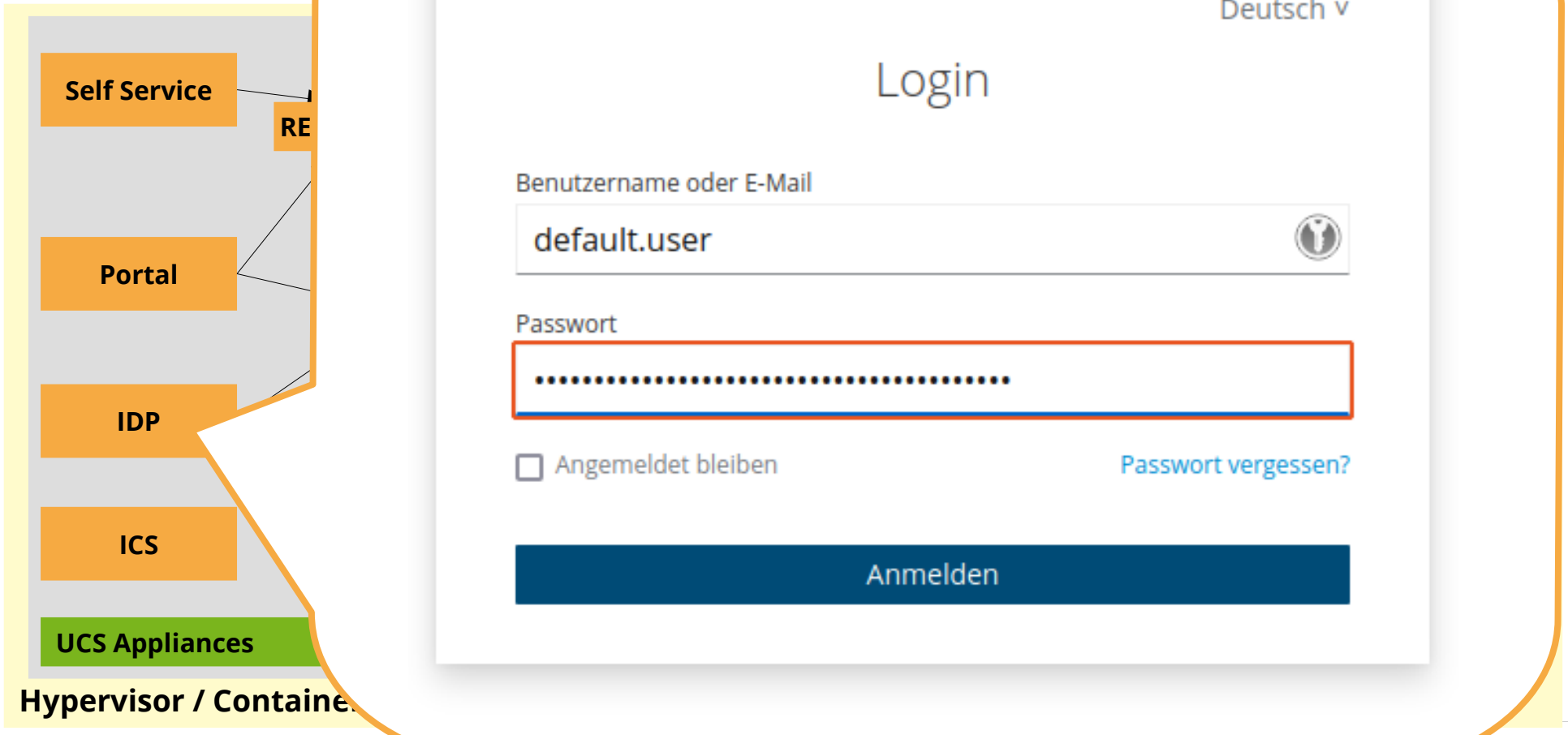
Projekt „Souveräner Arbeitsplatz“ / „OpenDesk“

- Projekt des BMI / ZenDIS, Umsetzung durch Dataport + OSS Unternehmen („Hersteller“)
- Ziel: Webbasierte Standardservices für Behördenarbeitsplatz, u.a.
  - Mail / Groupware
  - „Cloud“-Filestore + Online Office
  - Kommunikation Chat / Video
  - Organisation Projektmanagement / Dokumentation
- Aufgabe Univention als einer der OSS Hersteller:
  - Einheitliches, föderierbares Nutzermanagement + Single Sign-On
  - Dienste-Integration:
    - In Portal als Standardzugang für Nutzende
    - Im Backend für Nutzer-Provisionierung, SSO, Rechte/Rollen

UCS – Module: Ausschnitt Souveräner Arbeitsplatz



UCS – Module


Self Service
RE
Portal
IDP
ICS
UCS Appliances
Hypervisor / Containe
Deutsch
Login
Benutzername oder E-Mail
default.user
Passwort
Angemeldet bleiben
Passwort vergessen?
Anmelden

UCS – Mod

Self Service

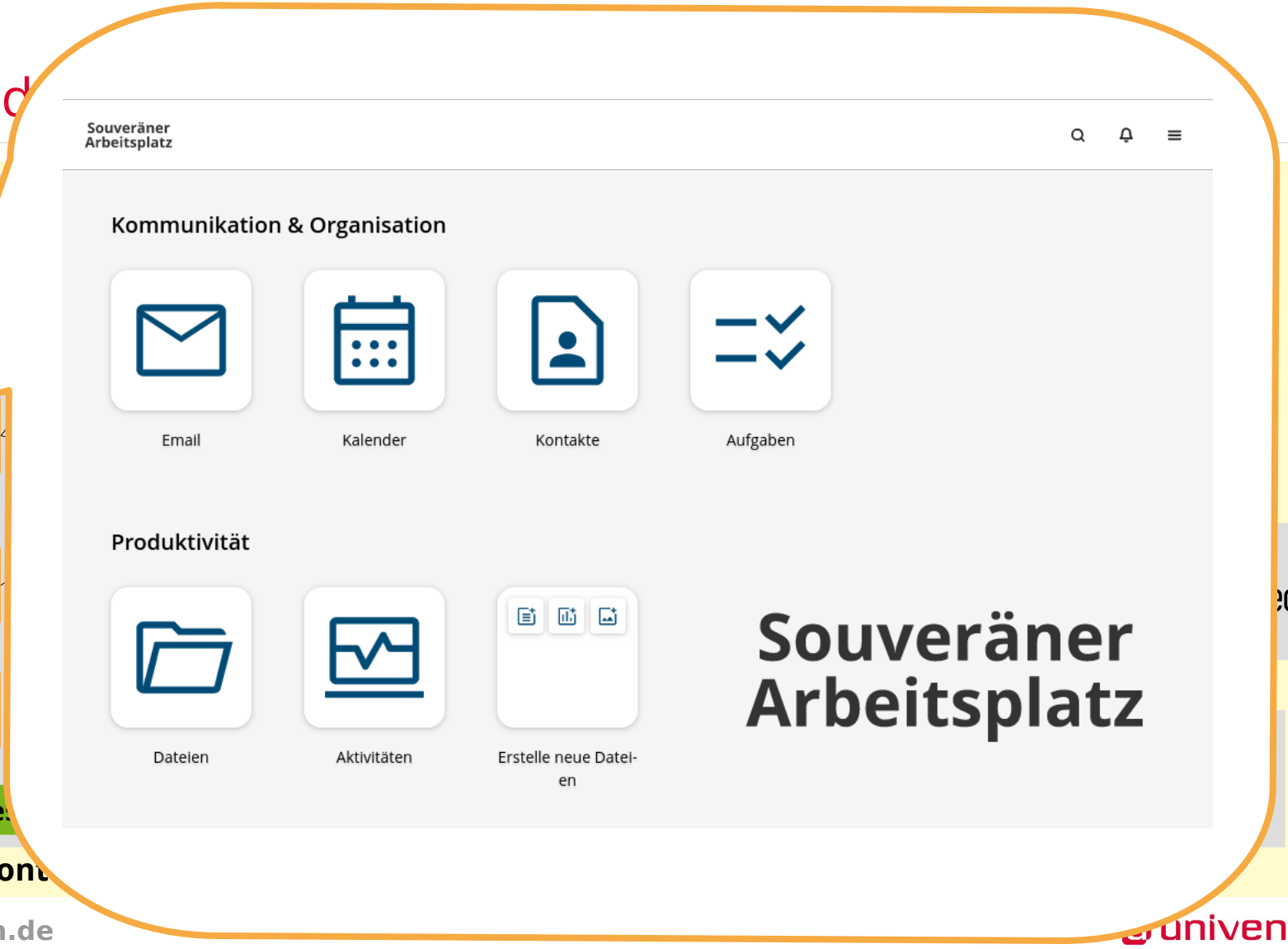
Portal

IDP

ICS

UCS Appliance

Hypervisor / Cont



UCS – Mo

Self Ser

Portal

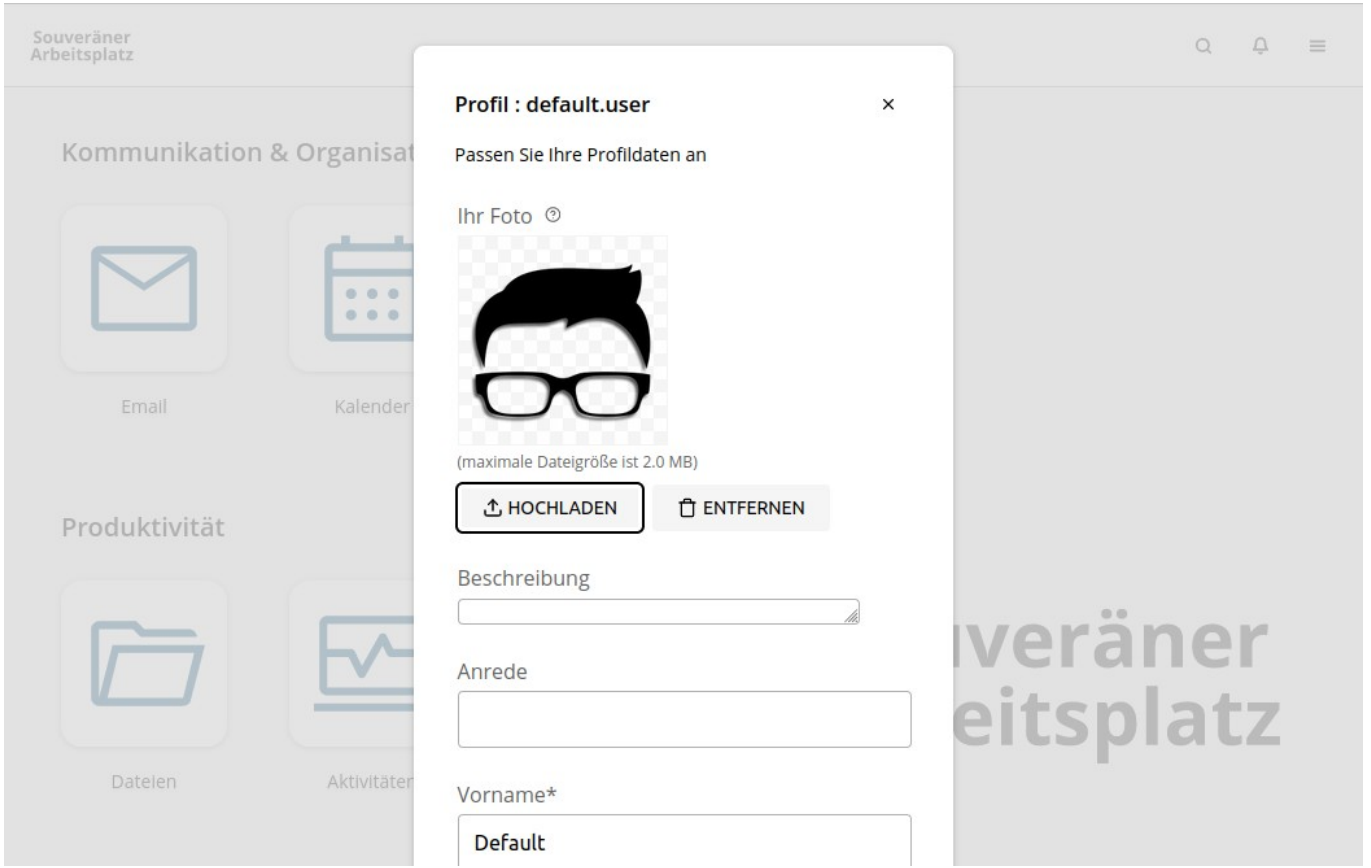
IDP

ICS

UCS Appliar

Hypervisor / C

roject



univention

Rahmenbedingungen Öffentliche Hand / Souv. Arbeitsplatz

- Funktionale Anforderungen
- Barrierefreiheit
- Deutsche Verwaltungscloud-Strategie (DVS)
- Souveräner Arbeitsplatz: OSS

[ja, ist sehr stark vereinfacht]

- **Im Weiteren Fokus auf DVS**

Deutsche Verwaltungscloud-Strategie - DVS

- Beschluss des IT Planungsrat – Arbeitsgruppe Cloud-Computing und Digitale Souveränität

  „Ziel des Dokumentes ist es, gemeinsame Standards für die föderale Cloud-Infrastruktur der ÖV und deren Standorte zu definieren“

- Salopp formuliert:

  Wie soll IT in der Verwaltung in Zukunft betrieben werden & unter welchen Rahmenbedingungen wird eine auszurollende Software arbeiten müssen?

- Beschreibt Anforderungen an Betreiber sowie an zu betreibende Software
- Im Folgenden: Fokus auf die Auswirkungen auf Bereitstellung von Software

Stand „Beschluss 2022/47“: https://www.it-planungsrat.de/beschluss/beschluss-2022-47

DVS - Kernforderungen

- Einhaltung existierender Vorgaben
  - *BSI Grundschutz*
  - *Kriterienkatalog Cloud Computing des BSI (C5)*
  - *Architekturrichtlinie für die IT des Bundes*
  - *Weitere Bundes- und länderspezifische sowie kommunale Architekturrichtlinien/-vorgaben bzw. Mindestanforderungen für die IT*
  - *Föderale Architekturrichtlinien für die IT*
  - *Anforderungen an Technologieanbieter und -lösungen zur Stärkung der Digitalen Souveränität*

DVS - Kernforderungen

- Explizite Eckpunkte
  - *Verteilter IT-Betrieb*
  - *Allgemeine Verfügbarkeit von Cloud-Services*
  - *Einsatz von OS-Software (OSS)*
  - *Zentrale Verwaltung von Services*
  - *Gemeinsame Weiterentwicklung*

DVS - Kernforderungen

- Festgelegte Softwarekomponenten
  - *Kubernetes*
  - *Helm*
  - *Container-Registry (z.B. Harbor)*

Wir haben das immer so gemacht, was machen wir jetzt?

Auswirkungen auf UCS (Ausschnitt)

- Software in Container-Images + Helm Charts bereitstellen

  → nicht als Virtuelle Maschine oder in .deb Paketen

- Container-Images folgen BSI Grundschutz / C5

  → u.a. Skalierung auf Service-Ebene, nicht auf Basis virtueller Maschinen

  → minimale Container, kein root, …

- Andere, aber weniger Teile des BSI Grundschutz

Vorgehen: Funktionalität identifizieren I

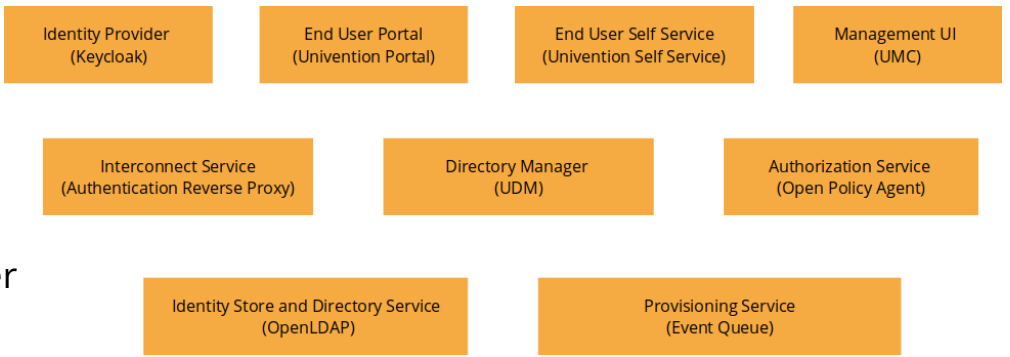
- Was aus UCS brauchen wir eigentlich?
- Fokus: „Einfacher Zugang zu IT-Anwendungen“
  - **Nutzerverwaltung** (Identitäten, Rollen, Rechte)
  - Über **Portal + Single Sign-On** für EndanwenderInnen
  - Über einfache **Integrationsmöglichkeiten** mit Anwendungen (APIs, Integrationspakete) und bestehenden IAMs

→ Die von Univention bereitgestellte Software verbindet verschiedene OSS „Module“ zu einer einheitlich Administrier- und Nutzbaren Anwendungslandschaft

→ Es entfällt viel: kein vollständiges Debian, kein Deployment von Anwendungen usw.

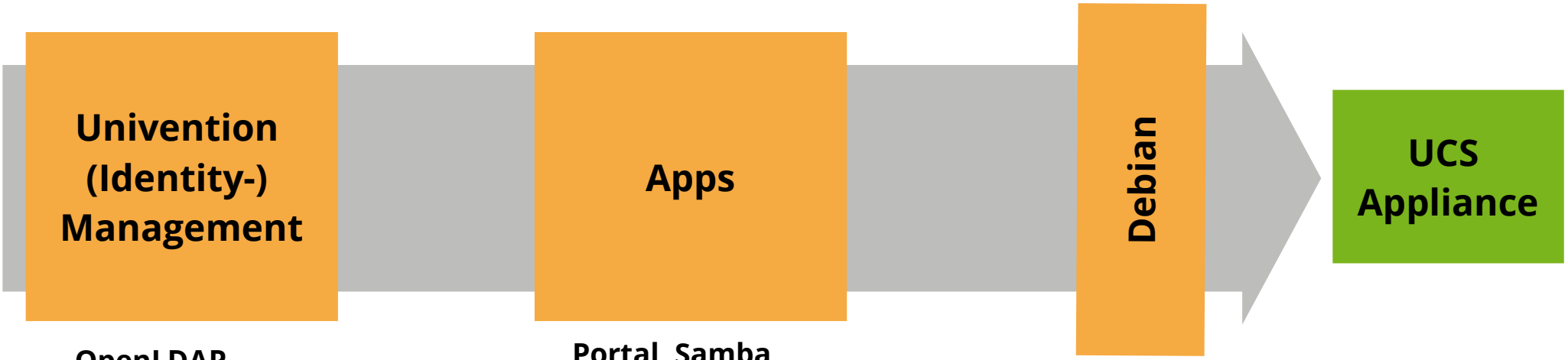
Vorgehen: Funktionalität identifizieren II

- Sinnvoll „clustern“ in Funktionsblöcke („separation of concern“)
- Ziele:
  - Funktionsblock wird individuell in Containern bereitgestellt
  - Funktionsblock teilt sich „upstream“ Code mit der Appliance (kein Fork)

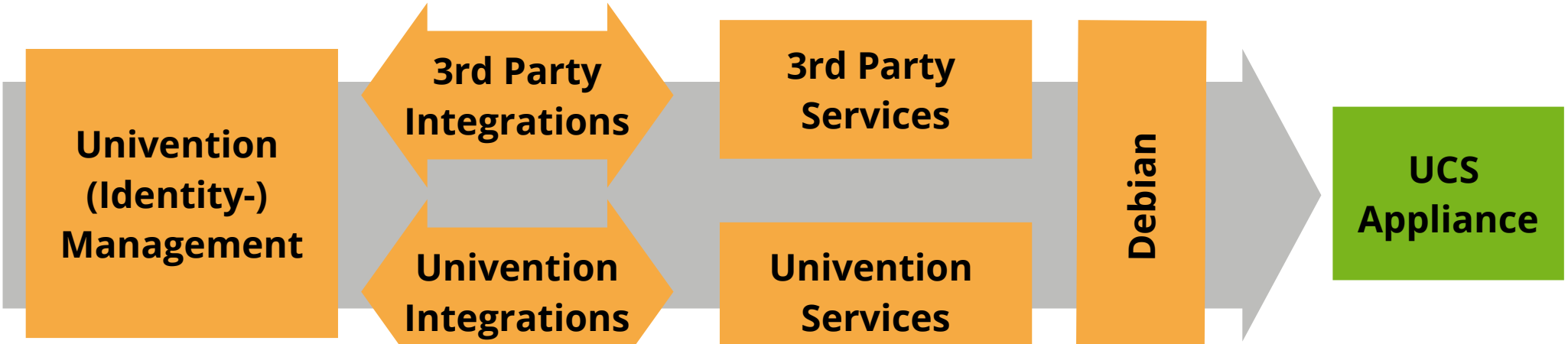


- Alles „Andere“ wird nicht angefasst/übernommen

Recap: Bausteine Univention Corporate Server



Bausteine – Integrationen & Service Deployment



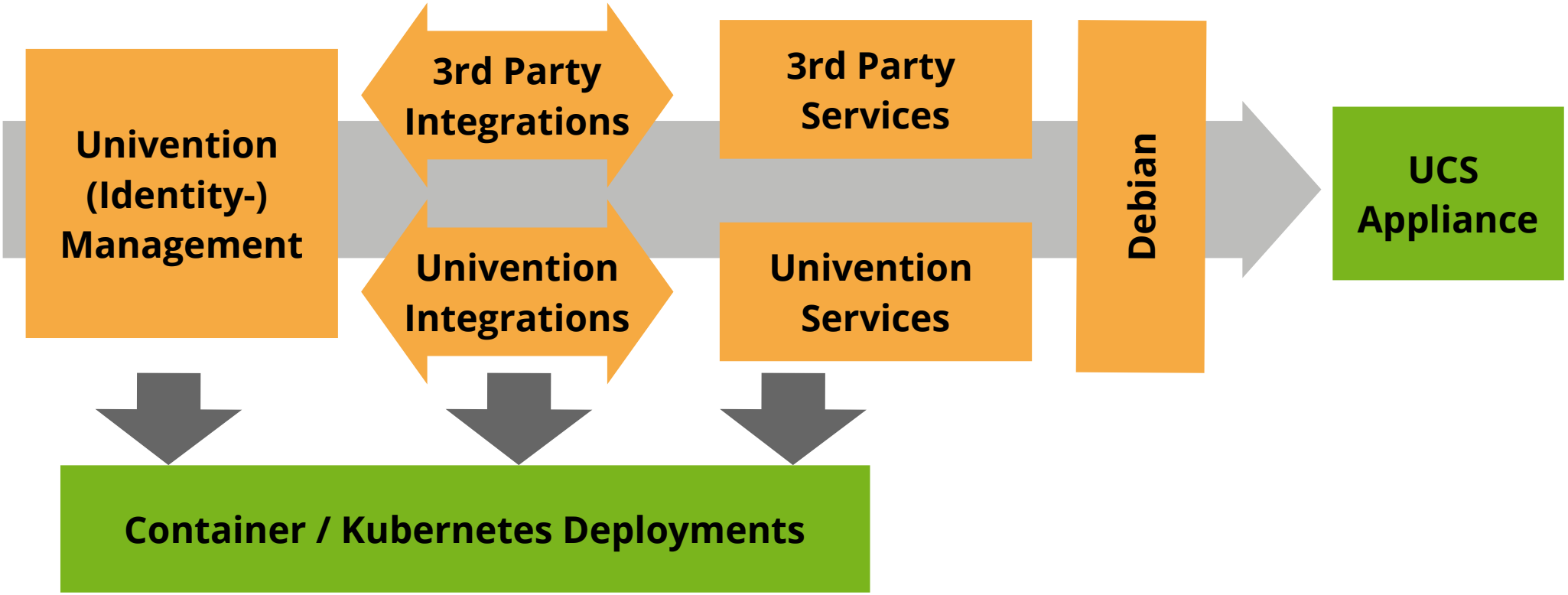
OpenLDAP, Univention Directory Manager, UMC, Keycloak, Self Service, ...

MS 365 Connector, G-Suite Connector, OX Connector, ...

Portal, Samba, File & Print, Groupware, File Sync & Share, Collaboration, DMS, ...

Enterprise Linux Distribution

Bausteine – Kubernetes & UCS Appliance



Funktionsblöcke – „Univention Management Stack“

Identity Provider
(Keycloak)

End User Portal
(Univention Portal)

End User Self Service
(Univention Self Service)

Management UI
(UMC)

Interconnect Service
(Authentication Reverse Proxy)

Directory Manager
(UDM)

Authorization Service
(Open Policy Agent)

Identity Store and Directory Service
(OpenLDAP)

Provisioning Service
(Event Queue)

Funktionsblöcke – Frontend / Nutzende

End User Portal
(Univention Portal)

End User Self Service
(Univention Self Service)

Identity Provider
(Keycloak)

Management UI
(UMC)

Zugang für Nutzende:

» **Portal**: Konfigurierbare Einstiegs-/Übersichtseite für personalisierten Zugang zu allen Webservices

» **Self Service**: Verwaltung des eigenen Nutzerkontos (Passwortwechsel, Kontaktdaten etc.)

» **Identity Provider**: Authentifikation für Nutzende per OpenID Connect / SAML, Föderierung mit externen IDPs

» **Management UI**: Web UI für Administrierende und ggf. Nutzende zur Verwaltung aller Identitäten und Berechtigungen

Funktionsblöcke – Backend / APIs

Directory Manager (UDM)

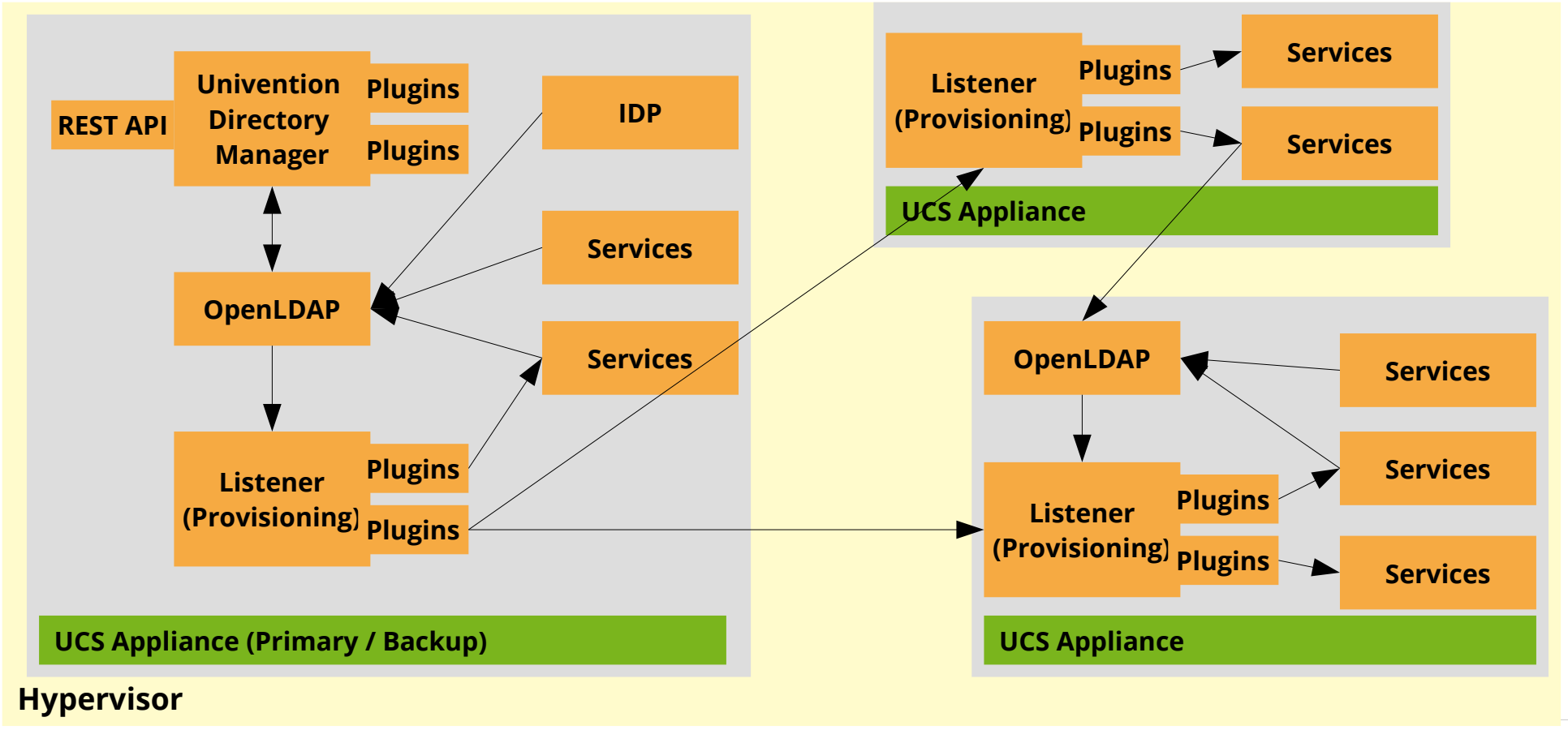
Authorization Service (Open Policy Agent / OPA)

Identity Store and Directory Service (OpenLDAP)

Provisioning Service (Event Queue)

- **UDM (REST API)**: Verwaltung von Identitäten / Konten, Gruppen, Berechtigungen und zugeordneten Ressourcen
- **Authorization Service**: Authorizations-API für Definition von Zugriffsrechten auf Services / APIs /UIs durch Nutzer- und Systemkonten
- **OpenLDAP**: (Identity-)Store zur RFC-konformen Speicherung von Identitäten, Gruppen etc. („Pull“ Zugriff durch Services)
- **Provisioning Service**: Verarbeitung von Events (mit Fokus IAM Events), z.B. zur „Push“-Integration von Services

Rückblick: bisherige UCS Module



Herausforderungen: „IAM Provisioning“

- Ziel: Events aus Änderungen an Objekten (z.B. Nutzerkonten) generieren und verarbeiten
  - Beispiel: Anlegen von Nutzerkonten in OpenXchange, Löschen von Daten beim Entfernen von Konten
- Bisher: Notifier/Listener
  - Herausforderungen u.a.:
    - Eine Queue mit X Plugins je virtuelle Maschine

      → skaliert nicht mit Services, sondern mit Instanzen
    - Datenaustausch an vielen Stellen dateibasiert, nicht API-basiert
- Entscheidung: Neukonzeption und -Entwicklung

Herausforderungen: Konfiguration

- Bisher „Univention Configuration Registry“
  - Konfigurationsdatenbank (key/value store) je virtueller Maschine
  - generiert Konfigurationsdateien, kann per Python Lib abgefragt werden
- Herausforderungen
  - Je Maschine, nicht je Service
  - Dateibasiert, nicht (remote-)API basiert
  - Vermischt Deployment und Betrieb
- Entscheidung: Ersetzen
  - Betriebskonfiguration wird eigene Datenhaltung ausgelagert („DCD“, basiert auf Redis + Python API)
  - Deploymentkonfiguration wird in Standardmechanismen (Helm etc.) verschoben

Herausforderungen: Container

- Minimale Container
  - Kein unnötiger Code / Binary im Container (auch keine Debugging-Tools…)
  - „read only“ images, idempotent, …
  - Ideal: Distroless / „single binary“

    → „Fleißarbeit“, ggf. Architekturänderung bei Aufteilen in mehr Container Images, Init Container usw.
- Skalierbare Container
  - Redundanz durch mehrfache Instanziierung
  - Skalierung durch mehrfache Instanziierung
  - Automatisierung (Kubernetes Health Checks & Autoscaler)

    → mehr als „Fleißarbeit“, ggf. grundlegende Architekturänderungen für Skalierung

Zwischenfazit

- Gute Nachricht:
  - Wesentliche Elemente der Grundstruktur bleiben erhalten
  - Anwendung von Patterns aus der Containerisierung verbessert auch die Appliance
- Herausforderung:
  - Viel zu tun…

Vorgehen / aktueller Stand (Juli 2023)

- Grundidee: Iterativ vorgehen, immer einen funktionierenden Gesamt-Stack haben
- I. Meilenstein: Herauslösen erster Services (erreicht Q1/23)
  - PoC: Portal – Aufgeteilt in mehrere Container, Betrieb parallel zur virtuellen Maschine
- II. Meilenstein: Dev-Env ohne virtuelle Maschine (erreicht Mitte 2023)
  - Alle notwendigen Funktionsblöcke als eigene Container – Start (mindestens) per Docker Compose

    https://gitlab.opencode.de/bmi/souveraener_arbeitsplatz/component-code/crossfunctional/ucs/wip-containerization

- ToDo:
  - Umsetzung Provisioning Stack
  - Standardisierung Build- und Release-Prozesse, Stabilisierung
  - Minimierung und Skalierung Container (iterativ…)
  - Ersatz UCR durch „DCD“

Wo kann ich das sehen?

- Univention GitHub Mirror:
  - Änderungen an den „Upstream“-Komponenten
  - Schrittweise auch Container

    https://github.com/univention/
- OpenCoDE
  - Gesamtergebnisse des Souveränen Arbeitsplatzes

    https://gitlab.opencode.de/bmi/souveraener_arbeitsplatz
  - Teilweise erst NACH Prüfung durch Projektpartner (daher nicht alle Entwicklungsstände)

Kontaktdaten

Ingo Steuwer
VP Platform and Technology
<steuwer@univention.de>
@IngoS@mastodon.social

Univention GmbH
Mary-Somerville-Str. 1
28359 Bremen

Univention GmbH Berlin
Mariannenstr. 9-10
10999 Berlin

Univention North America Inc.
7241 185th AVE NE #3206
Redmond, WA 98073-3206