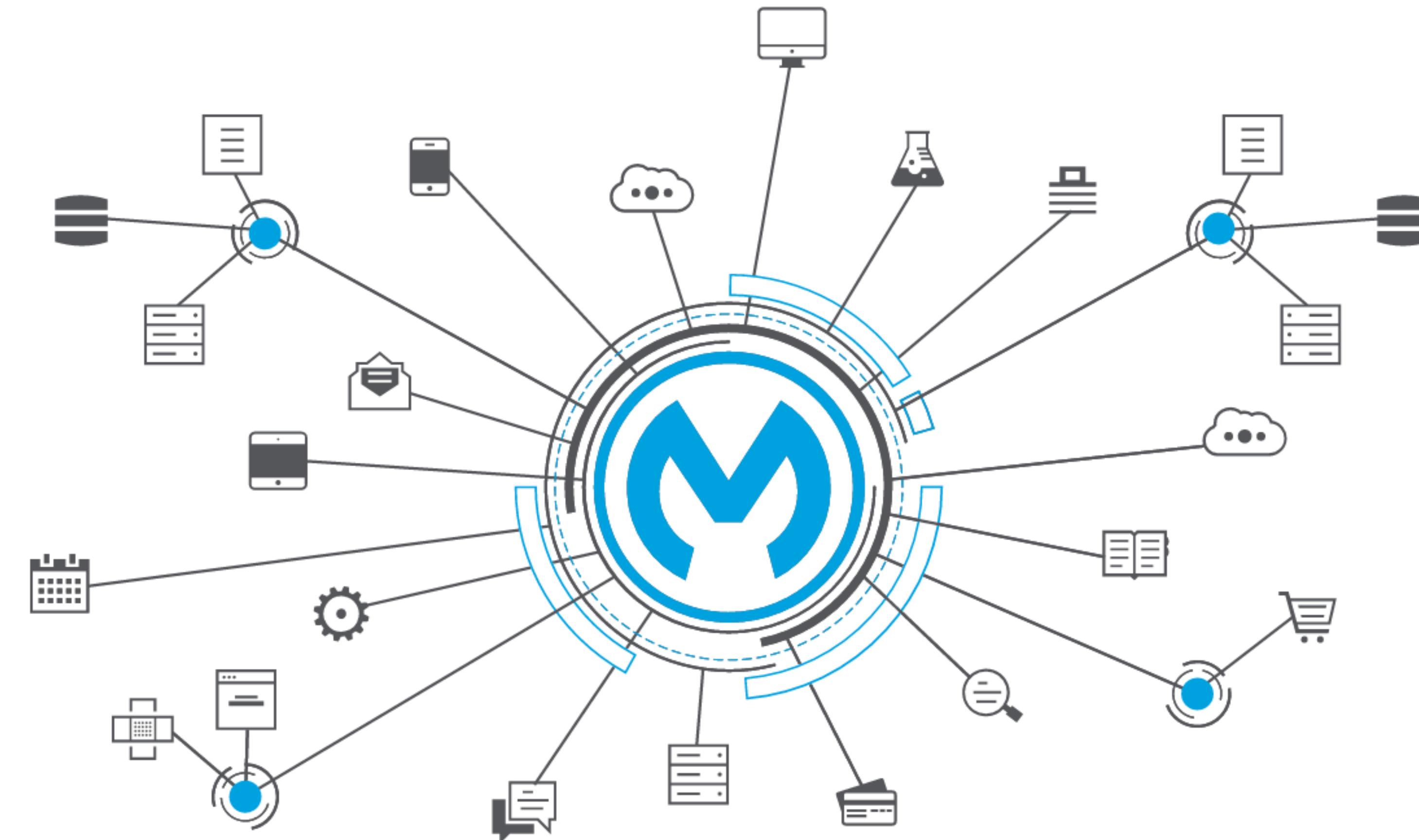


# Mulesoft for Java Developers

WJAX 2021/Anand Shankar Uday Shankar/Tony Horst

/



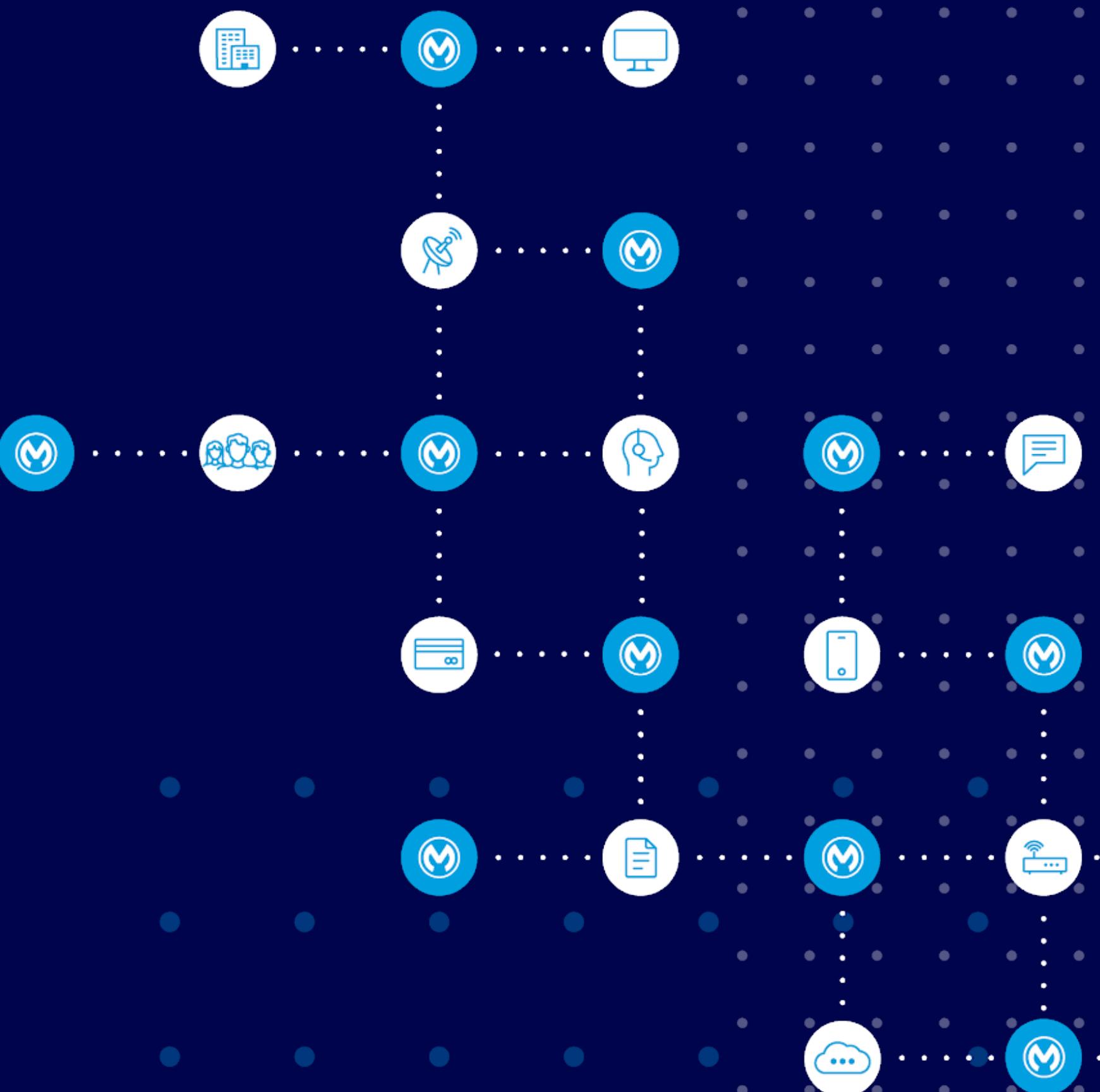
# Agenda

- Was sind die Probleme die Firmen heutzutage in ihrer IT Landschaft haben?
- Was ist MuleSoft überhaupt?
  - Wie ist es aufgebaut?
  - Was macht es?
  - Wieso ist es für mich als Java Entwickler so interessant?
- API Led - der etwas andere Architektur Ansatz
- Live Demo
  - Software Development Lifecycle
    - API Design
    - Exchange Publisher
  - Anypoint Studio - die (fast) perfekte Eclipse Erweiterung
  - Deployment
  - Monitoring
  - API Manager Policy



MuleSoft

# Probleme

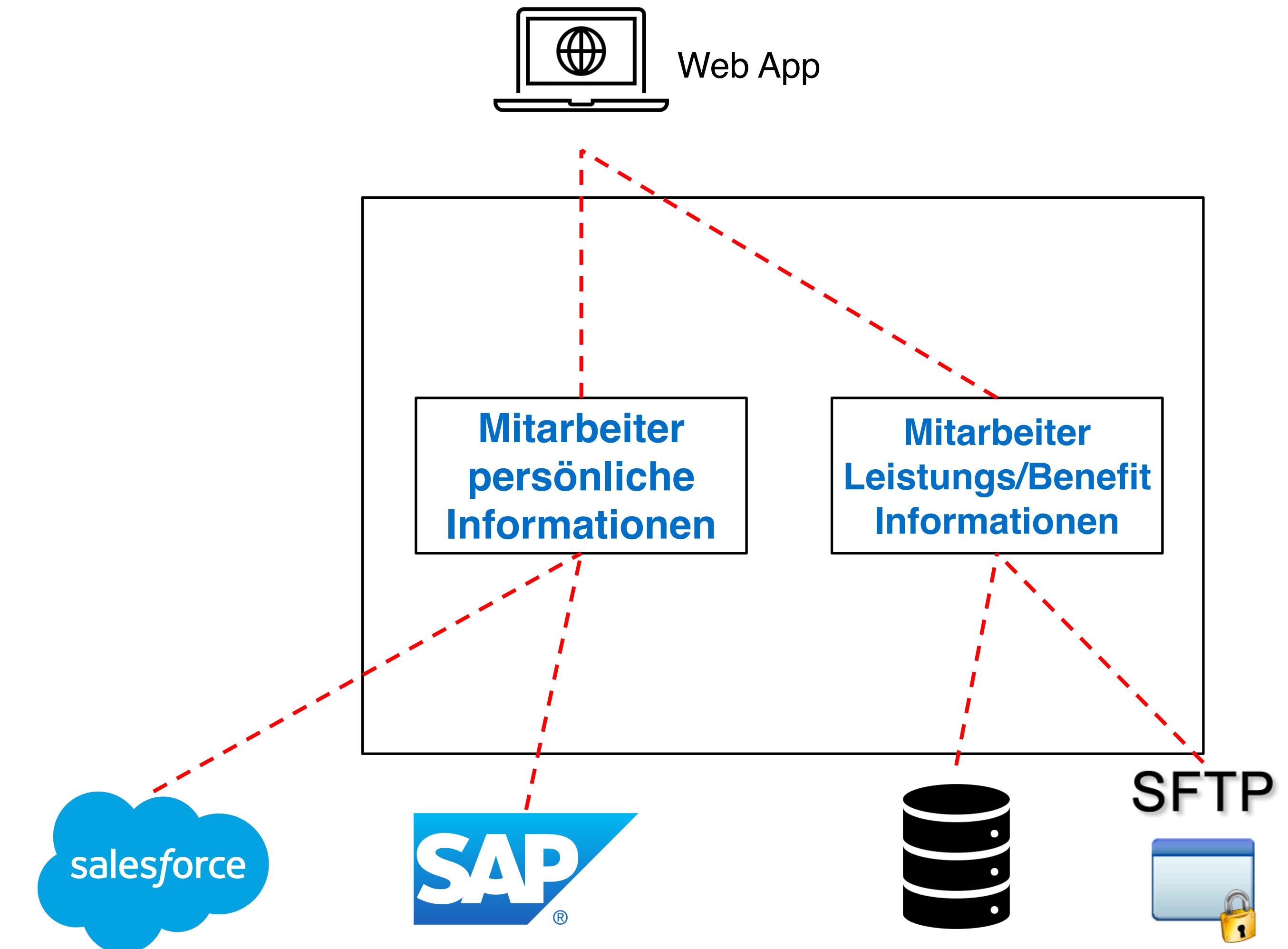


# Nachteile existierender Point-to-Point Integrationslösungen



## Projekt Übersicht: Web Application die Mitarbeiterinformationen anzeigen soll

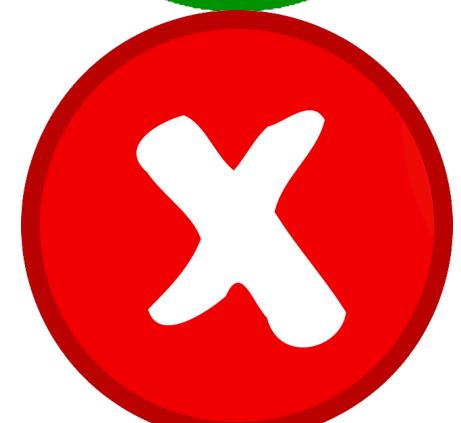
- EMPSYS ==> Daten Mitarbeiter
- Salesforce System ==> Urlaub & Gehalt
- Datenbank (😢) ==> Werbeaktionen & Benefits
- SFTP Server ==> Leistungsanalysen



# Vorteile und Nachteile existierender Point-to-Point Integrationslösungen



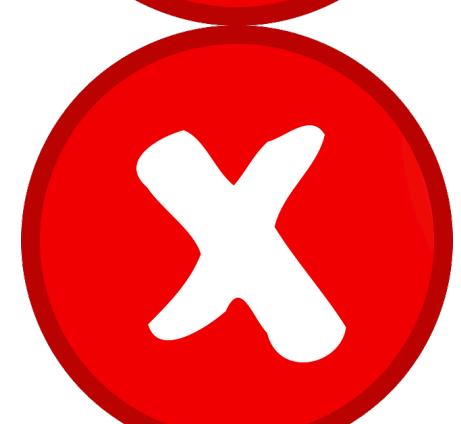
**Pünktlich und im Budget**



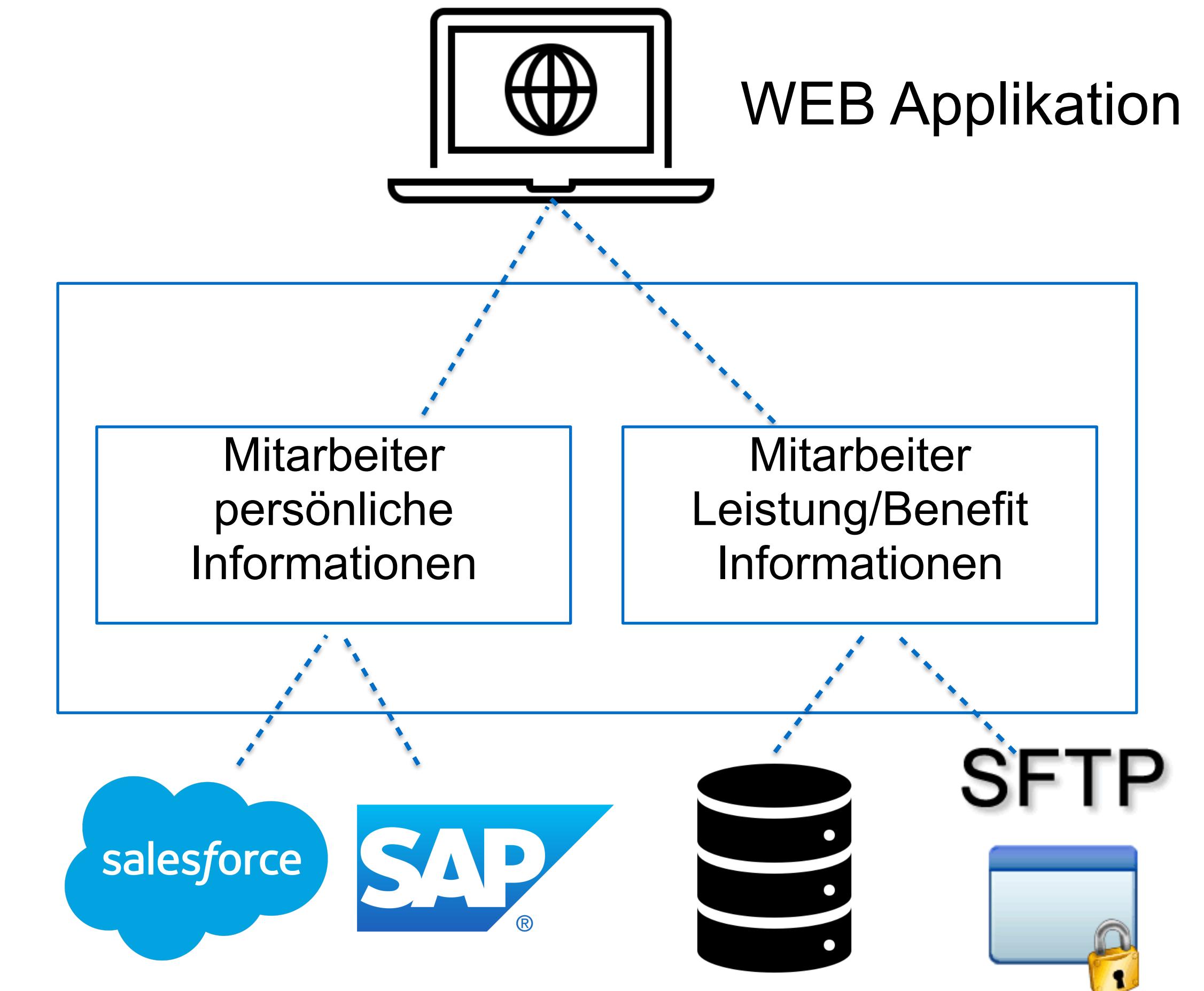
**Begrenzte Wiederverwendbarkeit**



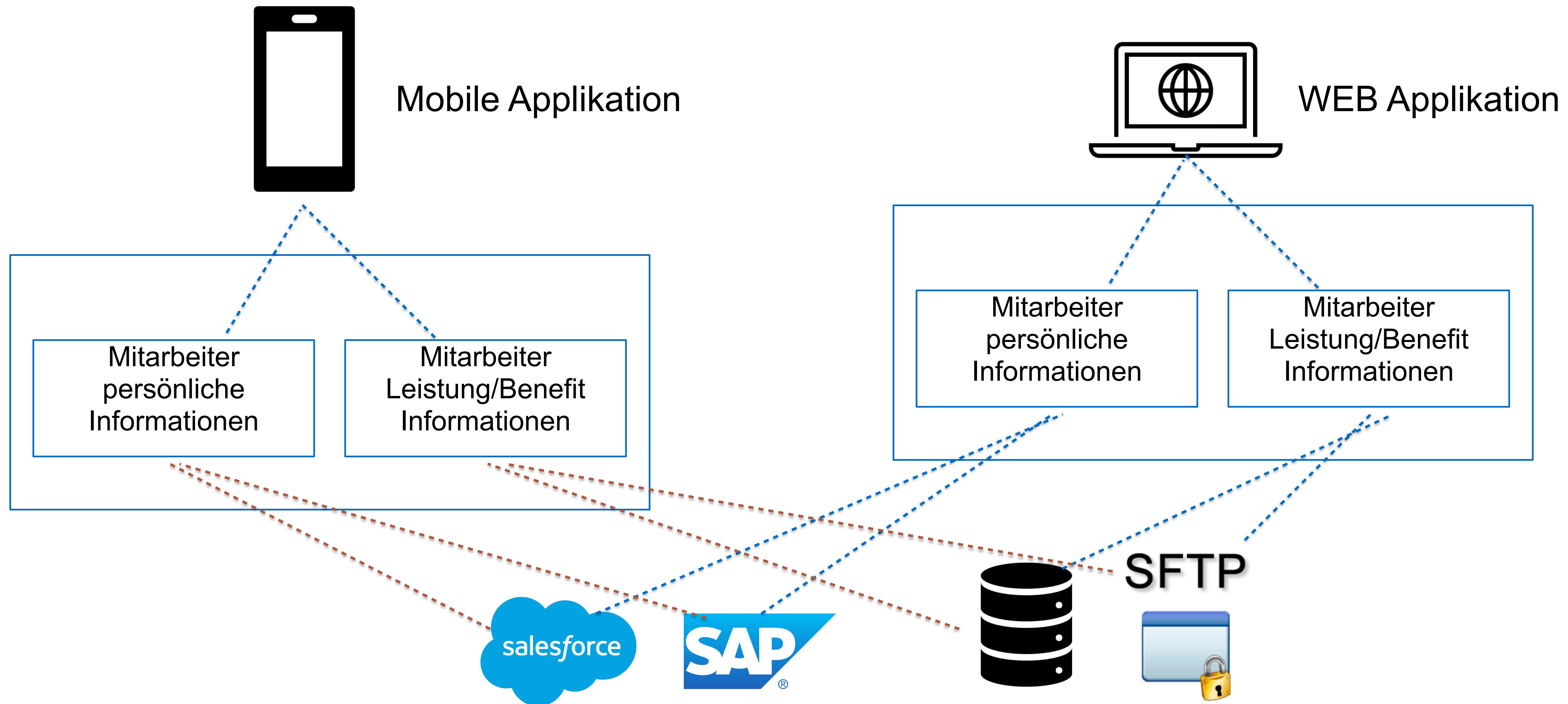
**Eine feste Kopplung der Komponenten**



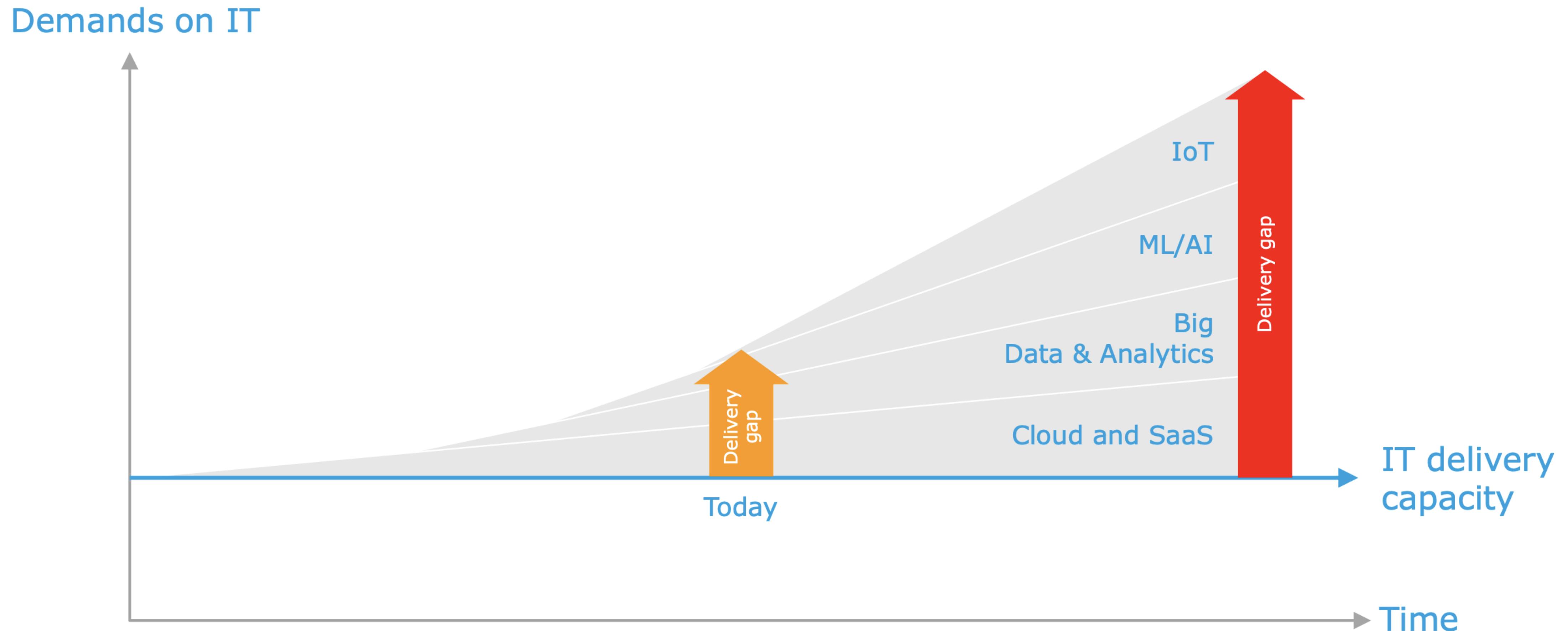
**Schwierig auf Änderungen erweiterbar**



# 6 Monate später



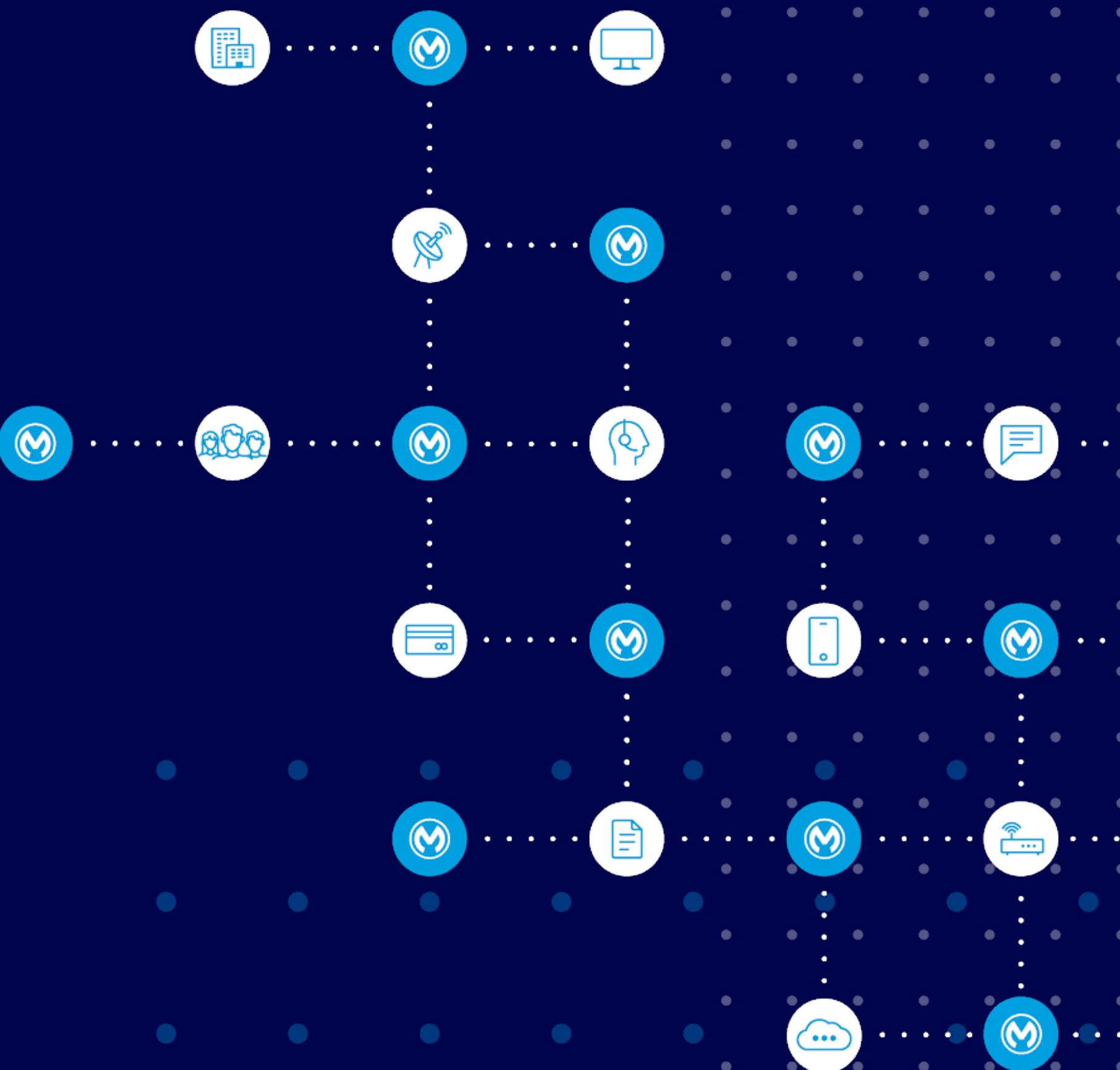
# Der aktuelle Trend





MuleSoft

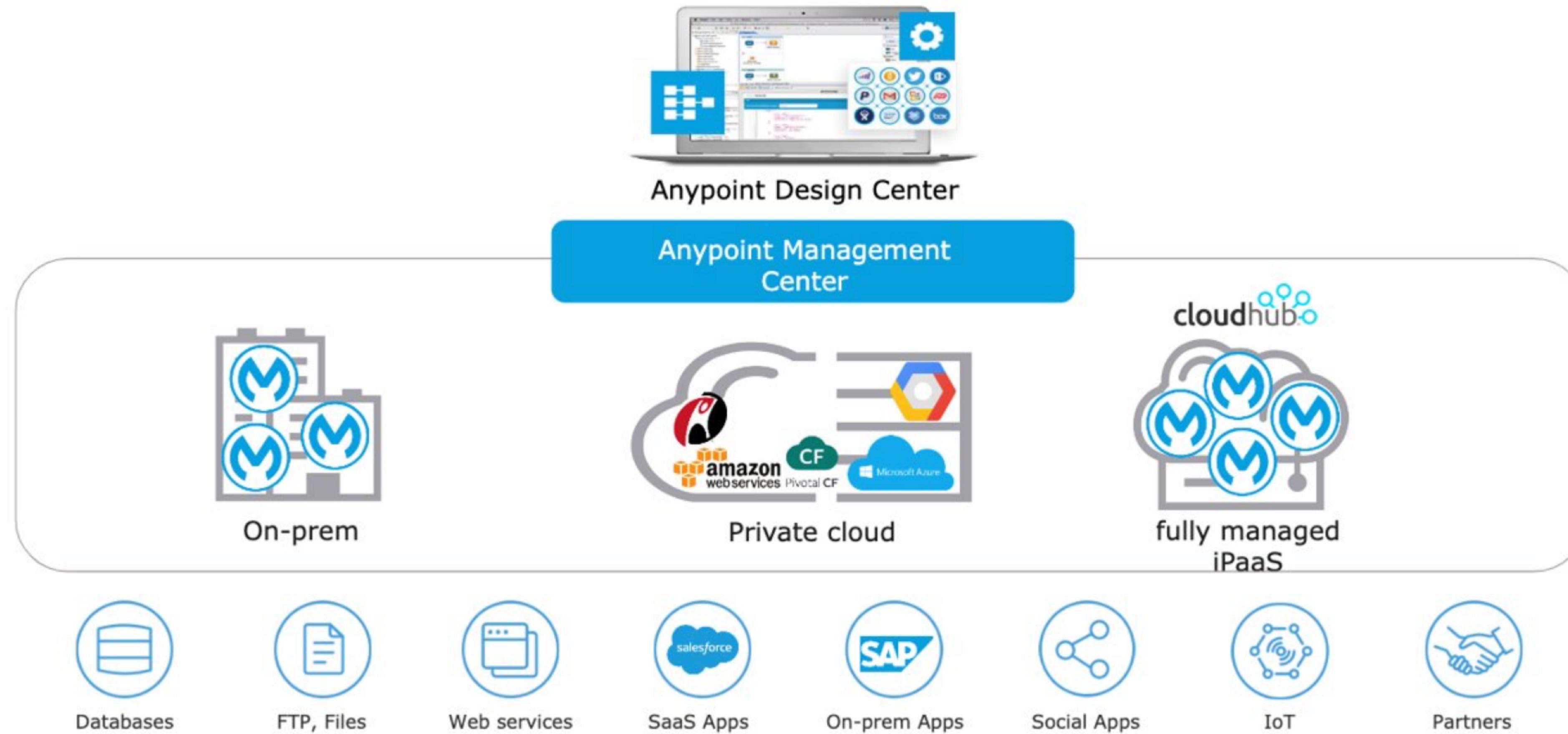
Lösung ==> MuleSoft



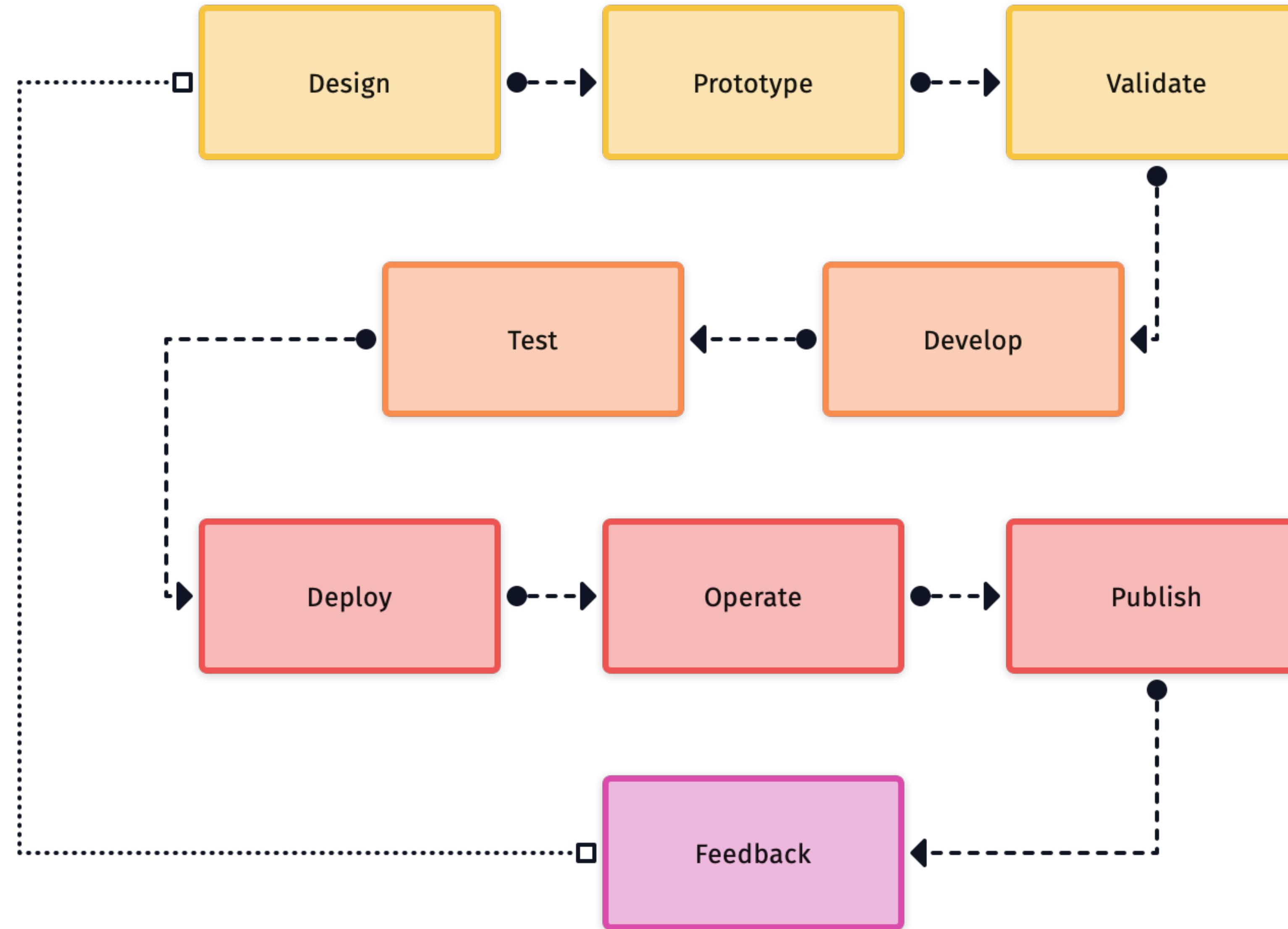
## Was ist Mulesoft überhaupt?

- Integrationslösung von Salesforce, mit deren Hilfe effizient und schnell APIs geschrieben werden (nach dem WYSIWYG Prinzip)
- Die unterschiedlichsten Systeme und Protokolle können genutzt und angebunden werden ([eine kleine Übersicht](#)) - als Beispiele
  - **Protokolle:** REST, SOAP
  - **Datenbanken:** MongoDB, IBM, Microsoft SQL
  - **Programme/Tools/Cloud:** Salesforce, SAP, Office365, Amazon AWS
  - **Queues:** IBM Queue, Websphere, JMS Queues
- Im Kern ==> Systeme, die sonst nicht miteinander sprechen würden, können über spezielle Transformationen und Anbindungen miteinander arbeiten (über Transformationen) => [Dataweave Language](#)

# Wie ist Mulesoft aufgebaut?



# Wie ist Mulesoft aufgebaut? (Software Development Lifecycle)



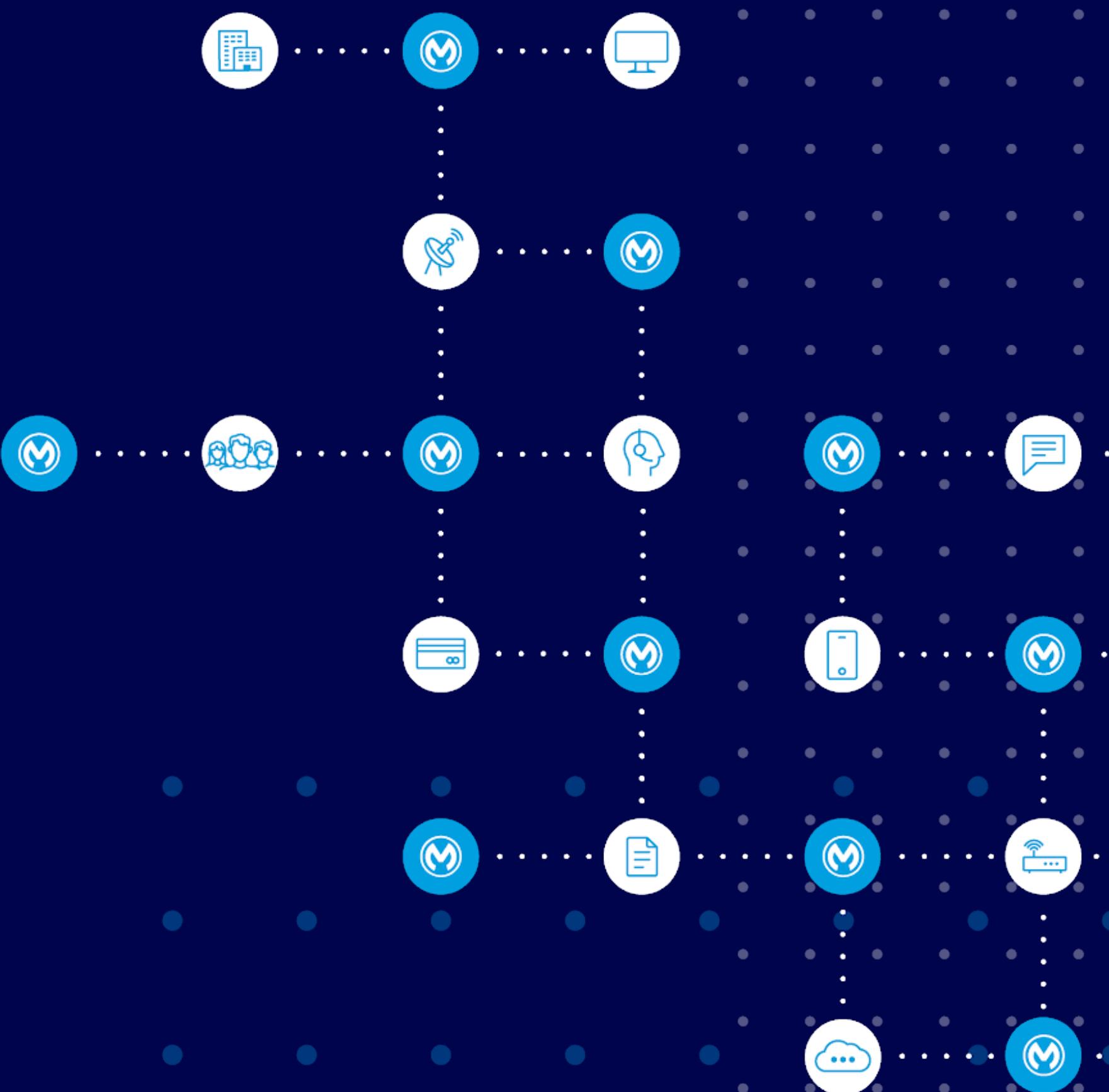
## Warum ist es für mich als Java Entwickler interessant?

- Basiert auf Java - viele Dinge die man dort kennen und „lieben“ gelernt hat gibt es auch dort
  - Maven
  - Spring Integration (ESB)
  - UnitTests die als Java Anwendung geschrieben werden können
  - Stage/Common Properties
  - Keystore/TrustStore Anbindung und Security Verwaltung
  - Eine Entwicklungsumgebung die quasi eine Erweiterung von Eclipse ist (nein, es gibt leider keine brauchbare IntelliJ Erweiterung 😢)

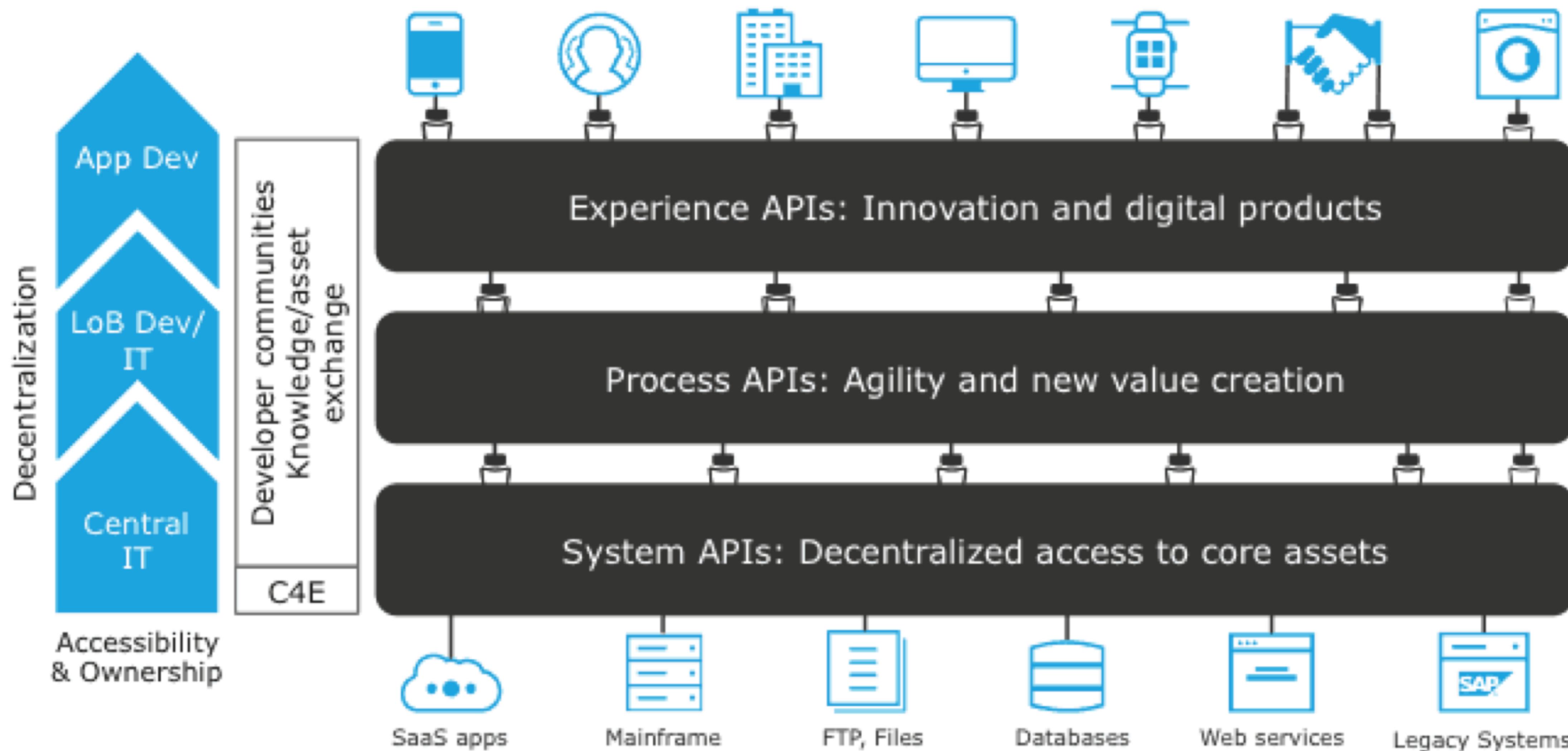


MuleSoft

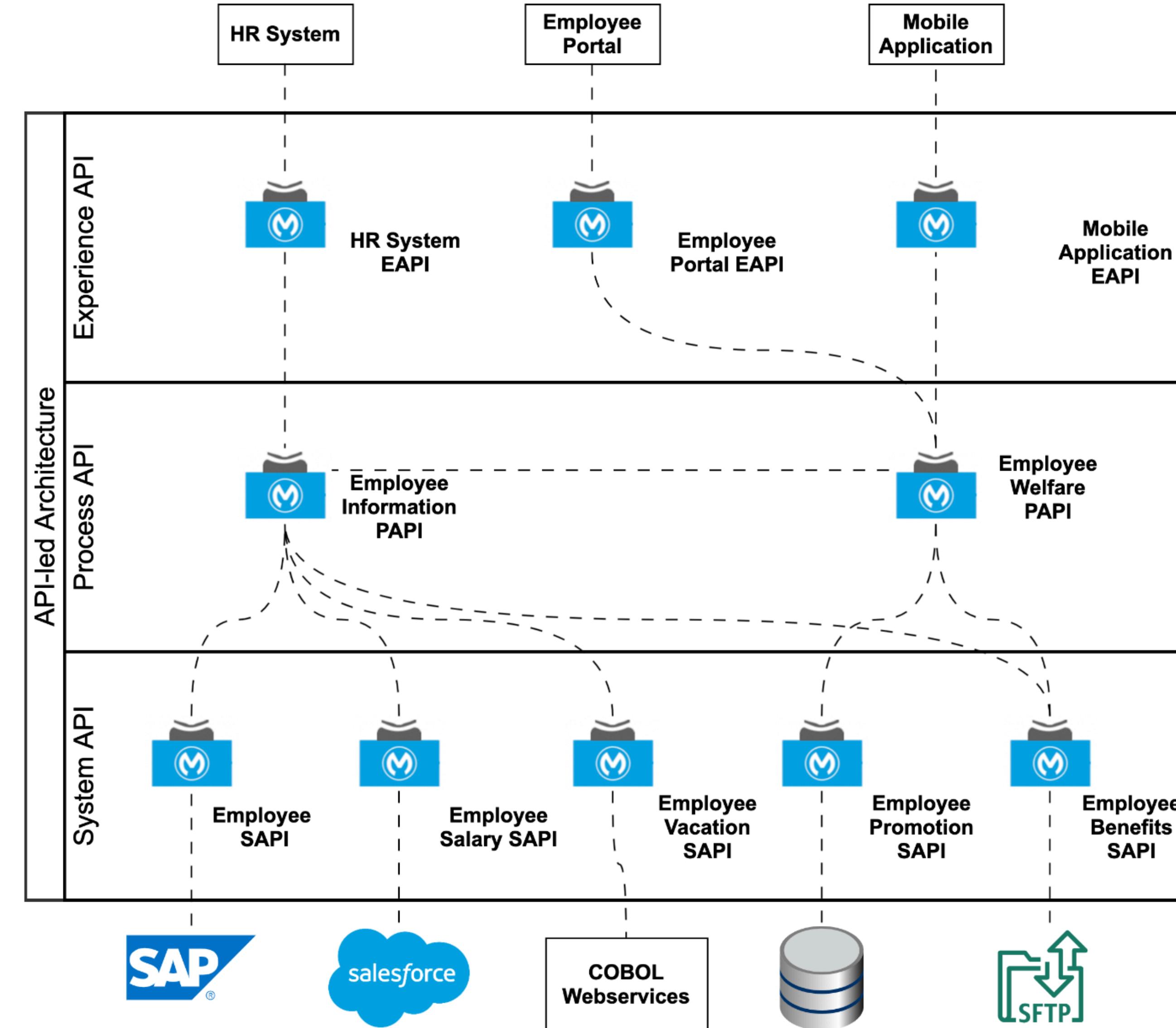
# API-led Architektur



# Was ist die API-LED Architektur



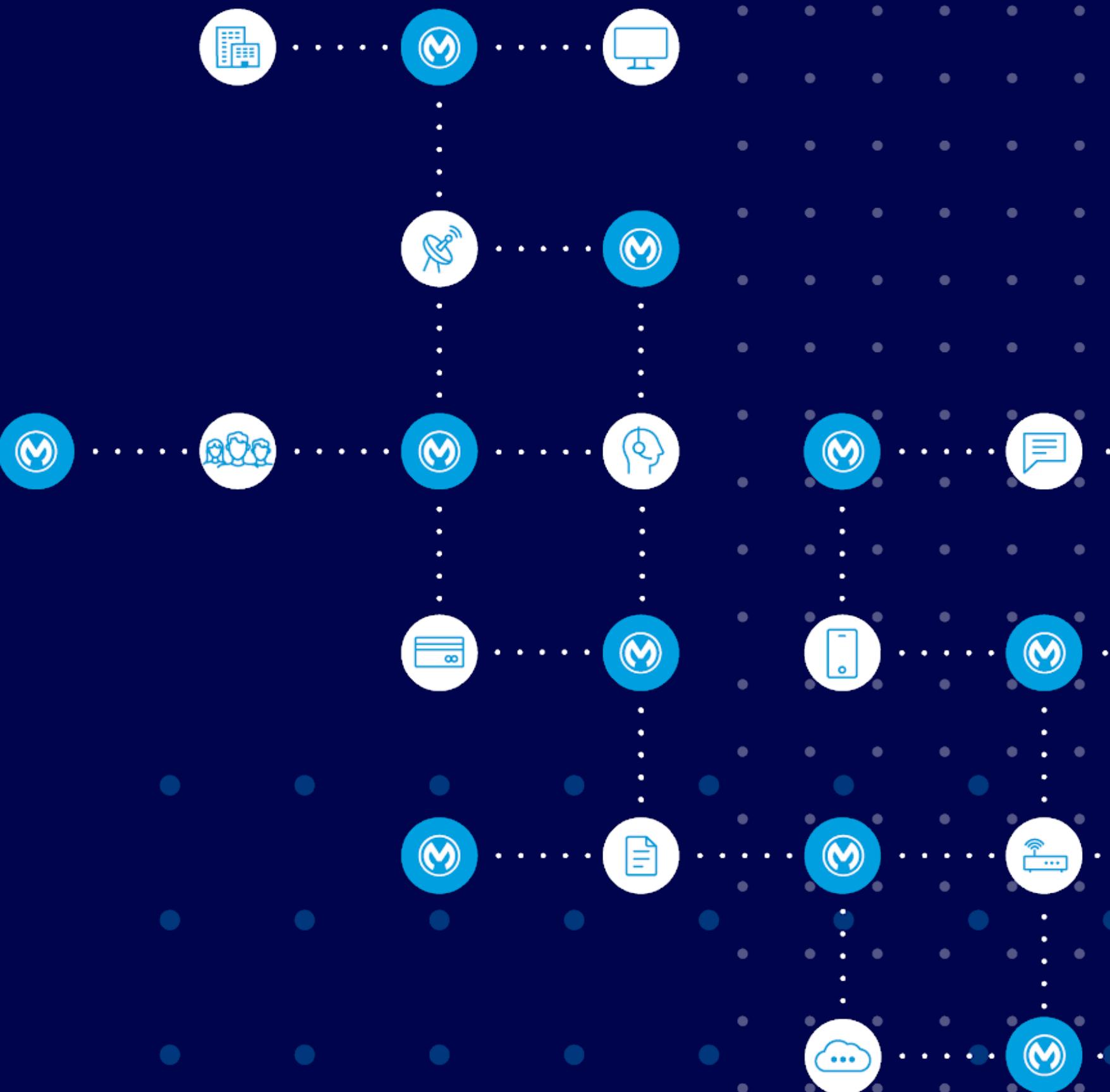
# Das Problem mit API-LED Architektur Ansatz lösen



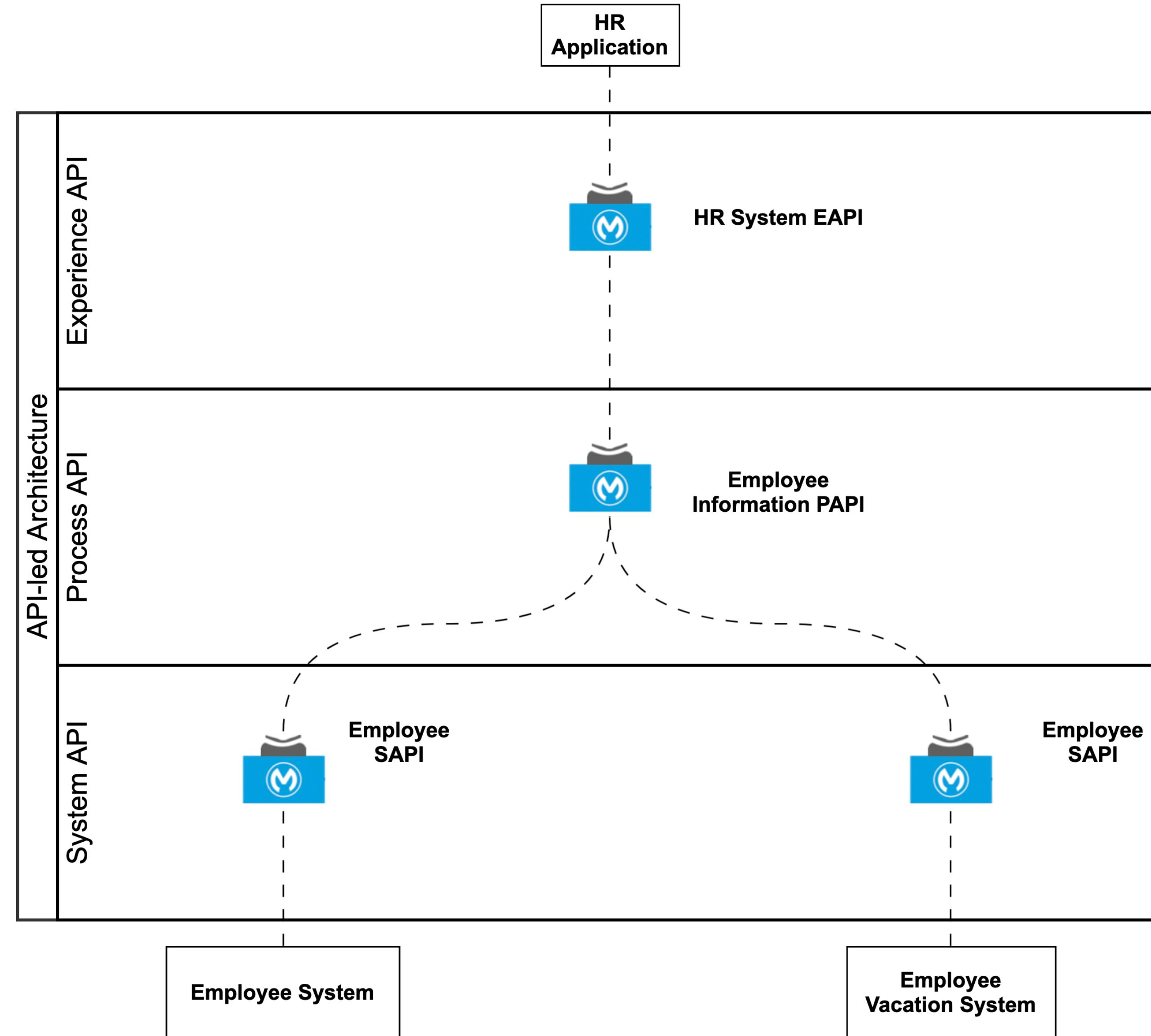


MuleSoft

# Live Demo :-)



# Kurzbeschreibung



# Unterlagen und Details zu den Sprechern



Anand Shankar Uday Shankar



Tony Horst

Anand ist ein zertifizierter Mulesoft-Entwickler und Mulesoft-Architekt.

Tony ist ein zertifizierter Mulesoft-Entwickler.

**Uday Shankar, Anand Shankar <[Anand-Shankar.Uday-Shankar@adesso.de](mailto:Anand-Shankar.Uday-Shankar@adesso.de)>**

**Horst, Tony <[tony.Horst@adesso.de](mailto:tony.Horst@adesso.de)>**