

Berufssakademie Sachsen

Staatliche Studienakademie Leipzig

Evaluierung der SAP Cloud Platform für die Entwicklung
und Anwendung (energiewirtschaftlicher) Funktionen (Apps)
am Beispiel einer selbst entwickelten Funktion

Bachelorarbeit

zur Erlangung des akademischen Grades eines

„Bachelor of Science“

in der Studienrichtung Informatik

Eingereicht von: Angela Stöckert

Herrmann-Meyer-Straße 11, 04207 Leipzig

Seminargruppe: CS15-1

Matrikelnr.: 5000559

Betreuer: M. Sc. André Kierzkowski

Arvato Systems Perdata GmbH

Martin-Luther-Ring 7-9

04109 Leipzig

Leipzig, 12.07.2018

Danksagung

An dieser Stelle möchte ich mich bei allen bedanken, die mich bei der Anfertigung dieser Arbeit unterstützt haben. Dies sind an erster Stelle Frau Prof. Dr. Susanne Schneider und Herr M. Sc. André Kierzkowski, welche die Bachelorarbeit betreut und begutachtet haben. Ihre Hinweise zur Erstellung und Gestaltung meiner Arbeit waren sehr hilfreich und wertvoll.

Ebenso dürfen sich alle weiteren Dozenten der Berufsakademie Leipzig im Studiengang Informatik angesprochen fühlen, die mit ihren Vorlesungen den Grundstein für meine Bachelorarbeit legten.

Meinem Arbeitgeber, der Arvato Systems Perdata GmbH, und im besonderen einigen Kollegen gebührt ebenfalls mein Dank. Jan Arensmeyer, Meikel Bode, Enrico Dittmann, Christoph Kümmel-Schulte, Max Marschall und Andreas Oczko standen mir für die Beantwortung fachlicher Fragen zur Verfügung. Von Mirko Finsterbusch und Enrico Wehner erhielt ich wichtige Literaturhinweise. Kent Dee Kleinstuber hat mir eine SAP-Nutzer-ID verschafft und damit den Zugriff auf besondere Hilfsmittel von SAP ermöglicht. Zur Qualität der vorliegenden Arbeit haben Daniel Lippmann, Lukas Reinhardt und André Tschirner durch Korrekturen maßgeblich beigetragen.

Besonderen Dank möchte ich Georg Andrásy aussprechen. Ohne seine Hinweise zum wissenschaftlichen Arbeiten, die mehrmalige umfassende Prüfung der Arbeit und die mentale Unterstützung hätte ich das vorliegende Niveau sicherlich nicht erreicht.

Inhaltsverzeichnis

1 Einführung in die Thematik	1
1.1 Vorwort.....	1
1.2 Motivation und Zielstellung	2
1.3 Thesen	4
2 Vorstellung der SAP Cloud Platform	5
2.1 Plattform.....	5
2.2 Cloud-Computing.....	5
2.3 SAP Cloud Platform.....	9
2.3.1 Zielgruppe	10
2.3.2 Umgebungen und Regionen.....	10
2.3.3 Dienste	13
2.3.4 Zugang zu Diensten auf der SAP Cloud Platform und Bezahlung.....	16
2.3.5 Anbindung an andere Systeme	18
2.3.6 Verfügbarkeit und Wartung.....	19
3 Vorüberlegungen zur Implementierung einer Webanwendung	21
3.1 Bestehende Anwendung mit Eigenentwicklung	21
3.2 Anforderungen an die Anwendung	22
3.3 Fiori	24
3.4 Verfügbarkeit und Erweiterung von Standardfunktionen	25
3.5 Architektur der eigenen Anwendung	26
3.6 Sicherheit und Datenschutz.....	28
3.7 Entwicklungsumgebung.....	33
3.8 Evaluierungskriterien	35
4 Evaluierung der SAP Cloud Platform	37
4.1 Umsetzung der Webanwendung.....	37
4.1.1 Verbindung zwischen SAP Backend und SAP Cloud Platform	38
4.1.2 Einrichten und Veröffentlichen eines OData-Dienstes	42
4.1.3 Erstellen und Anpassen der Oberfläche	47
4.1.4 Deployment der Anwendung	52
4.1.5 Erreichbarkeit der Anwendung für Nutzer	55
4.1.6 Verwendung der Anwendung mit einem Standard-OData-Dienst	56
4.2 Einschätzung der SAP Cloud Platform	59
4.2.1 Entwicklungsumgebung SAP Web IDE.....	59
4.2.2 Verbindungen zur Cloud Platform.....	60
4.2.3 SAP Cloud Platform als Ganzes	61

5 Fazit und Ausblick.....	64
5.1 Probleme	64
5.2 Ausblick	65
6 Literaturverzeichnis	IX
7 Anhangsverzeichnis	XX
8 Anhang.....	XXI
9 Selbstständigkeitserklärung	XLVIII

Abkürzungsverzeichnis

ABAP	Advanced Business Application Programming
API	Application Programming Interface
App	Applikation
ASE	Adaptive Server Enterprise
ASP	Arvato Systems Perdata GmbH
CI	Customizing Include
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DPC	Data Provider Class
DSGVO	Datenschutzgrundverordnung
ECC	ERP Central Component 6.0
ERP	Enterprise Resource Planning
EU	Europäische Union
GUI	Graphical User Interface
HANA	High Performance Analytic Appliance
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
I18N	Internationalisation
IaaS	Infrastructure-as-a-Service
IDE	Integrated Development Environment
IMAP	Internet Message Access Protocoll
IT	Informationstechnik
IW-BEP	Information Worker – Business Enablement Provisioning
JDK	Java Development Kit
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KDC	Key Distribution Center
LDAP	Lightweight Directory Access Protocol
MPC	Model Provider Class
MVC	Model-View-Controller
NIST	National Institute of Standards and Technology
OAuth	Open Authorization
OData	Open-Data-Protokoll

PaaS	Platform-as-a-Service
POP	Post Office Protocol
QR	Quick Response
RFC	Remote Function Call
S/4 HANA	SAP Business Suite 4 HANA
SaaS	Software-as-a-Service
SAML	Security Assertion Markup Language
SAP	Systeme, Anwendungen und Produkte in der Datenverarbeitung
SAP HANA XS	SAP HANA Extended Application Services
SAPUI5	SAP UI Development Toolkit for HTML5
SBE	SAP Backend
SE	Societas Europaea
SEGW	SAP Gateway Service Builder
SGW	SAP Gateway
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
SSO	Single Sign-On
TCP	Transmission Control Protocol
UI	User Interface
VW	Volkswagen
WZV	Web-Zugriffsverwaltungsprodukt
XML	Extensible Markup Language

Abbildungsverzeichnis

Abbildung 1: Dienstebenen in der Cloud	7
Abbildung 2: SAP HANA Cloud Platform.....	9
Abbildung 3: SAP Cloud Platform Cockpit der Neo-Umgebung.....	11
Abbildung 4: Regionen in der SAP Cloud Platform.....	13
Abbildung 5: Dienstkategorien auf der SAP Cloud Platform.....	14
Abbildung 6: Integrationsmöglichkeiten von SAP Cloud Platform.....	18
Abbildung 7: Wartungsfenster und Upgrade-Frequenz für Cloud-Dienste.....	20
Abbildung 8: Startbild der Transaktion <i>BP</i>	22
Abbildung 9: Ansicht der Transaktion <i>BP</i>	22
Abbildung 10: Anforderungen zur Umsetzung der Webanwendung	23
Abbildung 11: Bildschirmaufnahme der Filterung von SAP-Fiori-Anwendungen	25
Abbildung 12: Übersicht über die mögliche Architektur von Fiori-Applikationen	26
Abbildung 13: Anzeige der vom Client geforderten Berechtigungen bei OAuth 2.0.....	30
Abbildung 14: Ablauf einer Kerberos-Authentifizierung	30
Abbildung 15: Ablauf einer SAML-Authentifizierung	31
Abbildung 16: Übersicht der Umsetzung	37
Abbildung 17: Ansicht des SAP Cloud Connector	39
Abbildung 18: Ressourcen auf dem SGW-System	40
Abbildung 19: Datenmodell und Laufzeitklassen eines OData-Dienstes	42
Abbildung 20: Elemente im Datenmodell eines OData-Dienstes.....	43
Abbildung 21: Annotationen der Entitätsmengen <i>Accounts</i> und <i>ContractAccounts</i>	44
Abbildung 22: Elemente im Datenmodell des selbst definierten OData-Dienstes.....	45
Abbildung 23: Hauptseite des OData-Provisioning-Dienstes.....	46
Abbildung 24: Fenster zur Auswahl eines Dienstes.....	46
Abbildung 25: Konfiguration der SAPUI5-Version eines Projekts	47
Abbildung 26: Anzeige zur Auswahl einer Projektvorlage	48
Abbildung 27: Anzeige der Kundenart in der CRUD-Anwendung.....	50
Abbildung 28: Ausschnitt aus der englischsprachigen Ressourcen-Datei	51
Abbildung 29: Ansicht zum Erstellen einer Seite im Portal-Dienst.....	53
Abbildung 30: Zusammenspiel verschiedener Objekte im Fiori-Launchpad	54
Abbildung 31: Rollenvergabe in der SAP Cloud Platform.....	56
Abbildung 32: Anlegen eines neuen Erweiterungsprojektes.....	57
Abbildung 33: Anzeige der Erweiterungsmöglichkeiten	58

Tabellenverzeichnis

Tabelle 1: Parameter zur Konfiguration der Verbindung im SAP Cloud Connector	40
Tabelle 2: Parameter zur Konfiguration von Zielen.....	41
Tabelle 3: Anlegen einer Anwendung mit Hilfe einer Vorlage.....	49

1 Einführung in die Thematik

1.1 Vorwort

Nach einer kurzen Erläuterung der Motivation und der Festlegung von Arbeitsthesen im ersten Kapitel beleuchtet das zweite Kapitel zuerst den theoretischen Hintergrund und erklärt in diesem Zuge die Begriffe *Cloud*, *Plattform* und *Platform-as-a-Service*. Die nächsten Abschnitte stellen die Zielgruppe, die Umgebungen und Regionen sowie die Dienste der SAP Cloud Platform vor. Weiterhin wird beschrieben, wie ein Nutzer Zugang zur SAP Cloud Platform bekommen kann. Abschließend werden die Möglichkeiten zur Anbindung anderer Systeme und die Verfügbarkeit und Wartung der SAP Cloud Platform festgehalten.

Das dritte Kapitel beschäftigt sich mit den Vorüberlegungen zur Implementierung der Anwendung. Der erste Abschnitt stellt die bestehende Eigenentwicklung im SAP-ECC-System vor, der zweite leitet daraus die Anforderungen für die zu entwickelnde Anwendung ab. Anschließend werden Fiori-Anwendungen im Allgemeinen vorgestellt und geprüft, ob eine bestehende Anwendung bereits die Anforderungen erfüllt. Trifft dies nicht zu, wird anschließend die Architektur einer selbst zu entwickelnden Anwendung festgelegt und die Themen Sicherheit und Datenschutz evaluiert. Der für die Erstellung der Anwendung besonders wichtigen Entwicklungsumgebung wird der vorletzte Abschnitt im dritten Kapitel gewidmet, bevor es durch Auflistung der Evaluationskriterien für die SAP Cloud Platform abschließt.

Das vierte Kapitel beschreibt die einzelnen Schritte zur Umsetzung der Webanwendung: das Erstellen und Anpassen einer Oberfläche, ihre Anbindung an die Datenquelle, die Auslieferung der Anwendung in der Cloud und auf einem lokalen Applikationsserver, sowie die Einbindung der veröffentlichten Anwendung auf einem Fiori-Launchpad, das in der SAP Cloud Platform bzw. auf einem lokalen Anwendungsserver betrieben wird. Die Anwendung wurde zur Umsetzung einer vom Standard abweichenden Datengrundlage konzipiert. Es wird geprüft ob sie auch zur Anzeige einer im Standard vorliegenden Datengrundlage verwendet werden kann und ob dafür Änderungen an der Oberfläche nötig sind.

Das fünfte Kapitel fasst die Ergebnisse der Umsetzung zusammen und führt auf, welche Probleme im Laufe der Umsetzung aufgetreten sind. Anschließend gibt es einen Ausblick auf Themen, auf die ASP zukünftig seinen Fokus legen sollte.

Im Titel dieser Arbeit ist von Funktionen die Rede. In der Programmierung finden sich diese als Quelltextanweisungen, welche als ausgelagerte Einheiten existieren und daher wiederverwendbar sind [1]. Jedes Programm ist zugleich eine Funktion, da es von anderen Programmen aufgerufen und wiederverwendet werden kann. Unter Funktionen werden jedoch unter anderem auch Teifunktionalitäten von Programmen verstanden. Um komplett

Programme von Teilprogrammen abzugrenzen, werden erstere innerhalb dieser Arbeit vorwiegend als *Anwendung* oder *Applikation* (kurz *App*) bezeichnet.

Die in dieser Arbeit verwendeten Systemnamen SGW (kurz für SAP Gateway) und SBE (kurz für SAP Backend) wurden aus Datenschutzgründen verändert und existieren so nicht in der Realität. Die Gestaltung dieser Arbeit orientiert sich an den Gestaltungsrichtlinien der Berufsakademie Leipzig und der Universität Leipzig [2]. Daher fiel die Entscheidung auf eine Seitennummerierung mit römischen Zahlen zu Beginn und zu Ende der Arbeit sowie auf eine Kapitelnummerierung mit römischen Zahlen an den gleichen Stellen.

1.2 Motivation und Zielstellung

Arvato Systems Perdata GmbH (ASP) steht aktuell vor der Aufgabe, seinen Kunden den Umstieg auf moderne SAP-Systeme zu ermöglichen und dabei die Kosten so gering wie möglich zu halten. Im Rahmen dieser Arbeit gilt es zu evaluieren, ob die *SAP Cloud Platform* in der Lage ist, energiewirtschaftliche Individualprogrammierungen aus dem System *SAP ERP Central Component 6.0* (ECC), das zur Planung von Unternehmensressourcen dient, als *SAP-Fiori*¹-Anwendungen [3] abzubilden. Hierfür sei ein Beispiel auszuwählen und umzusetzen und daran zu prüfen, ob dessen Oberfläche nach einmaliger Implementierung auch für Energieversorger mit Standardausprägung, also ohne Eigenentwicklungen, gleichermaßen nutzbar ist. In so einem Fall erwartet das Unternehmen ASP zukünftig eine drastische Verringerung der Wartungskosten für die Eigenentwicklung. Untersucht wird auch, ob durch die Verwendung von Diensten in der SAP Cloud Platform ein Mehrwert für ASP entsteht.

Das Unternehmen ist Teil der *Arvato Systems GmbH* und als solche auch Teil des *Bertelsmann*-Konzerns mit Hauptsitz in Gütersloh. Sie konzentriert sich auf Dienstleistungen im Informationstechnik-Sektor (IT-Sektor). Die Geschäftseinheit ASP, unter anderem am Standort Leipzig, hat sich insbesondere auf Dienstleistungen für Unternehmen der Ver- und Entsorgungswirtschaft spezialisiert. Dabei entwickelt ASP mit rund 300 Mitarbeitern individuelle IT-Lösungen, die exakt auf die Bedürfnisse des Kunden abgestimmt sind. Die Kernkompetenz liegt besonders in der Beratung, aber auch der Planung und Bereitstellung sowie dem Betrieb von SAP-basierten Systemen.

Ein Kunde, der von ASP betreut wird, hat in der Regel bereits bestehende, historisch gewachsene Geschäftsprozesse, die das System zur Ressourcenplanung eines

¹ Vergleiche Abschnitt 3.3 - Fiori

Unternehmens (ERP²) der SAP SE³ (SAP) in der Standardversion oftmals nur unzureichend abbilden kann. Anpassungen der ausgelieferten Funktionalitäten an die spezifischen betriebswirtschaftlichen Anforderungen des Kunden (*Customizing*) ist im ERP durch Parametrisierung vorgesehen [4, S. 11]. „Der Anwender kann so den Funktionsumfang sowie den Programmablauf der Standardsoftware steuern und gemäß seinen Vorstellungen modifizieren [5].“ Für die Erweiterung von Tabellen, die auch in der Geschäftslogik beachtet werden, stehen z.B. *Customizing Includes* (CI) zur Verfügung.

Die Möglichkeit der Erweiterungsprogrammierung wird nur dann ergriffen, wenn der durch das Customizing vorgegebene Rahmen nicht ausreicht [6], da sie mit einer Einschränkung der Garantie für die Kompatibilität von Aktualisierungen durch den Hersteller einhergeht [7, S. 338]. Garantien betreffen die Standardsoftware und berücksichtigen nur jene Schnittstellen zu Individualprogrammierungen, die in den von SAP vorgesehenen Platzhaltern untergebracht sind. Diese sind durch vordefinierte Schnittstellen mit dem ERP verbunden. Individuelle Programmbestandteile außerhalb dieser Platzhalter müssen nach dem Einspielen einer neuen Programmversion manuell auf ihre Funktionsfähigkeit geprüft und bestehende Probleme behoben werden. Dies ist für ASP mit Zeit und für den Kunden mit Kosten verbunden. Dennoch gibt es Fälle, in denen die Individualisierungen des Kunden so wichtig sind, dass er ASP mit Erweiterungsprogrammierungen außerhalb von Platzhaltern beauftragt.

Aber nicht nur bei Versions-, sondern auch bei Produktwechseln kann es zu hohen Aufwänden in Bezug zu Erweiterungsprogrammierungen kommen. SAP SE gab im Oktober 2014 bekannt, dass die aktuelle ERP-Produktlinie ECC (oft ebenso als *Business Suite* bezeichnet) ab 2026 nicht mehr unterstützt wird [8]. Kunden von ASP benötigen daher in den nächsten Jahren ein Folgesystem, das ihre Geschäftsprozesse abbilden kann und ihren Anforderungen an Stabilität, Zukunftsfähigkeit und Sicherheit entspricht. ASP sieht in der *SAP Business Suite 4 HANA* (SAP S/4 HANA) eine solche Alternative. Das System unterscheidet sich vom vorherigen ERP in vielen Punkten. Während bei ECC die Wahl der Datenbank dem Kunden überlassen ist, benutzt S/4 HANA zwingend *SAP HANA*, eine *In-Memory-Plattform*, die Datenzugriffe beschleunigen und Auswertungen in Echtzeit ermöglichen soll. In-Memory-Datenbanken nutzen primär den Arbeitsspeicher eines Computers als Speicher und erreichen so einen stark beschleunigten Datenzugriff [9]. Weiterhin können Datentabellen spaltenweise persistiert werden.

² engl. Enterprise Resource Planning

³ Societas Europaea (SE) „ist eine Rechtsform für Aktiengesellschaften in der Europäischen Union und im Europäischen Wirtschaftsraum“ [111].

Zudem bietet S/4 HANA mit Fiori neue Benutzeroberflächen. Diese machen einzelne betriebswirtschaftliche Anforderungen unter Berücksichtigung der Rolle und der Berechtigungen eines Nutzers als Apps auf allen Endgeräten verfügbar [10]. Solche modernen Benutzeroberflächen können bereits für ECC-Systeme eingerichtet werden, um den Kunden einen fließenden Übergang zum neuen System zu ermöglichen. Zum Beispiel ist es denkbar, im ersten Schritt einzelne Anwendungen im ERP-System durch Webapplikationen zu ersetzen [11, S. 20], damit Nutzer sich an die neuen Oberflächen und Navigationswege gewöhnen können. Passende Werkzeuge dafür bietet SAP auf der *SAP Cloud Platform* (früher *SAP HANA Cloud Platform*) an [12]. Bei ihr handelt es sich um eine Entwicklungsplattform, mit deren Hilfsmitteln beispielsweise Fiori-Oberflächen entwickelt, erweitert und angepasst werden können [13].

1.3 Thesen

Folgende Thesen gilt es, in der Analyse näher zu untersuchen.

- Es ist mit der SAP Cloud Platform möglich, Fiori-Anwendungen zu entwickeln, die ihre Daten aus ECC-Systemen beziehen.
- Für die Anpassung und Erweiterung von Fiori-Benutzeroberflächen ist die SAP Cloud Platform unverzichtbar.
- Individualprogrammierungen für Kunden können mit den Werkzeugen der SAP Cloud Platform als Fiori-Apps umgesetzt werden.
- Fiori-Apps, die eine Eigenentwicklung eines Kunden abbilden, können ohne Änderungen des Quelltextes auch für andere Kunden mit Standardausprägung genutzt werden.
- Die SAP Cloud Platform stellt einen Mehrwert bei der Entwicklung von Webanwendungen dar.

2 Vorstellung der SAP Cloud Platform

In diesem Kapitel sollen die namensgebenden Begriffe Plattform und Cloud-Computing definiert und die SAP Cloud Platform vorgestellt werden.

2.1 Plattform

Der Begriff *Plattform* ist nicht IT-spezifisch. Hinter ihm verbirgt sich jedoch weit mehr als nur eine *erhöhte Ebene*, wie im allgemeinsprachlichen Sinn. Er existiert in verschiedenen Anwendungsdomänen, wie beispielsweise in der Automobilindustrie. Dort versteht man darunter eine technische Basis, die für verschiedene Fahrzeugmodelle gleich verwendet wird. So bauen zum Beispiel der *VW Polo*, der *Seat Ibiza* und der *Skoda Fabia* auf derselben Plattform, bestehend aus Bodenplatte, Tank, Auspuffanlage, Heizung und weiteren Teilen, auf [14].

In der Informatik finden sich unzählige Plattformen, die sich zuerst einmal in Hardware- und Software-Plattformen einteilen lassen. Erstere beschreiben die Rechnerarchitektur mit ihren Arbeits-, Befehls- und Prozessstrukturen. Beispiele für Minicomputer-Plattformen sind *Arduino* oder *Raspberry Pi*. Software-Plattformen können mehrschichtig existieren. Die untere Schicht, zumeist ein Betriebssystem, läuft auf einer Hardware-Plattform und kann wiederum Basis für weitere Software-Plattformen sein, wie etwa für das Internet. Das Internet selbst ist die Basis für verschiedene Arten von Plattformen. Darunter finden sich Suchmaschinen (*Google*), Vergleichsportale (*Check24*), Marktplätze (*Amazon*), Sharing-Economy-Plattformen (*AirBnB*, *Mitfahrzentrale*), Inhaltsdienste (*Youtube*) und soziale Netzwerke (*Facebook*) [15, S. 4].

Für Softwareentwickler besonders interessant sind so genannte Entwicklungsplattformen. Dies sind Plattformen, die Schnittstellen zur Infrastruktur und Hilfsmittel zur Entwicklung von Anwendungen zur Verfügung stellen [16, S. 8]. Die Schnittstellen sind in der Regel standardisiert [17, S. 95]. Plattformen ermöglichen dem Entwickler, eigene Dienste zu definieren, die kompatibel mit denen von Partnern, Konkurrenten und Kunden sind, welche die gleiche Plattform nutzen [18, S. 625].

2.2 Cloud-Computing

In der Literatur finden sich unterschiedlichste Versionen von Definitionen des Cloud-Computing-Begriffs. 2011 hat das *National Institute of Standards and Technology* (NIST) zur Entwirrung beigetragen und eine offizielle Definition von Cloud-Computing publiziert [19], die Charakteristika, Service-Modelle und Verteilungsmodelle festlegt. Da sich auch in der deutschsprachigen Literatur mehrere Bezüge auf diese Definition finden lassen [20, S. 5],

[17, S. 95], [21, S. 69], sei sie auch in dieser Arbeit verwendet. Das NIST definiert fünf wesentliche Merkmale für Cloud-Computing:

- Der Zugriff des Nutzers auf die ihm zur Verfügung gestellten Dienste erfolgt nach Bedarf selbstständig und ohne menschliche Interaktion.
- Die Dienste sind auf verschiedenen Endgeräten (PC, Laptop, Smartphones und Tablets) erreichbar.
- Die Dienste des Anbieters sind in einem Ressourcenvorrat gesammelt und werden von mehreren Nutzern zeitgleich unter Berücksichtigung derer Bedarfe genutzt.
- Die Dienste können schnell und in einigen Fällen sogar automatisch gebucht und freigegeben werden, um sich „elastisch“ dem Bedarf des Nutzers anzupassen. Der Nutzer hat dabei den Eindruck, dass ihm unbegrenzte Ressourcen zur Verfügung stehen.
- Die Dienste können automatisch kontrolliert und der Ressourcenverbrauch optimiert werden. Diese Messung der Dienste mit Hilfe geeigneter Ressourcen sorgt für Transparenz für den Anbieter und den Nutzer der Cloud.

Die in der Cloud angebotenen Dienste werden allgemein in drei Dienstebenen unterteilt: *Infrastructure-as-a-Service* (IaaS), *Platform-as-a-Service* (PaaS) und *Software-as-a-Service* (SaaS). Baun [20, S. 39] erwähnt eine vierte Ebene, *Humans-as-a-Service* (HuaaS). Wie in Abbildung 1 erkennbar, unterscheiden sich die Ebenen darin, welcher Teil der IT vom Dienstleister betreut wird und welches Fachwissen der Kunde selbst braucht, um Angebote auf dieser Ebene zu nutzen. Bei IaaS wird dem Nutzer, etwa der IT-Abteilung eines Unternehmens, eine abstrahierte Sicht auf die vom Cloud-Anbieter zur Verfügung gestellte Hardware geboten. Er kann diese nutzen, um Betriebssysteme zu installieren, Netzwerktopologien zu definieren oder um beanspruchte Kapazitäten zu skalieren [20, S. 32]. Große Anbieter in dieser Kategorie sind Amazon Web Services, Microsoft und Google. Die Berechnung der Kosten auf dieser Ebene gestaltet sich komplex, da Kosten für Rechenzeit und – leistung, für Datenbanken und -speicher, für Anfragen zwischen Client und Server und für übertragene Daten anfallen können [22, S. 47]. Der Begriff Client stammt aus der Systemarchitektur Client-Server-Architektur, die verteilte Anwendungssysteme in den Dienstnehmer (Client) und den Dienstgeber (Server) unterteilen. Dabei können nur Clients einen Server anfragen, eine bestimmte Aufgabe für sie zu erledigen, woraufhin der Server entsprechend antwortet [23].

Dienste der PaaS-Ebene richten sich typischerweise an Softwareentwickler oder IT-Architekten und beinhalten Entwicklungs- und Laufzeitumgebungen [20, S. 35], welche die Zeit bis zur Fertigstellung der Anwendung deutlich verkürzen können, da sich der Nutzer nicht zusätzlich um die zugrunde liegende Hardware kümmern muss [22, S. 44].

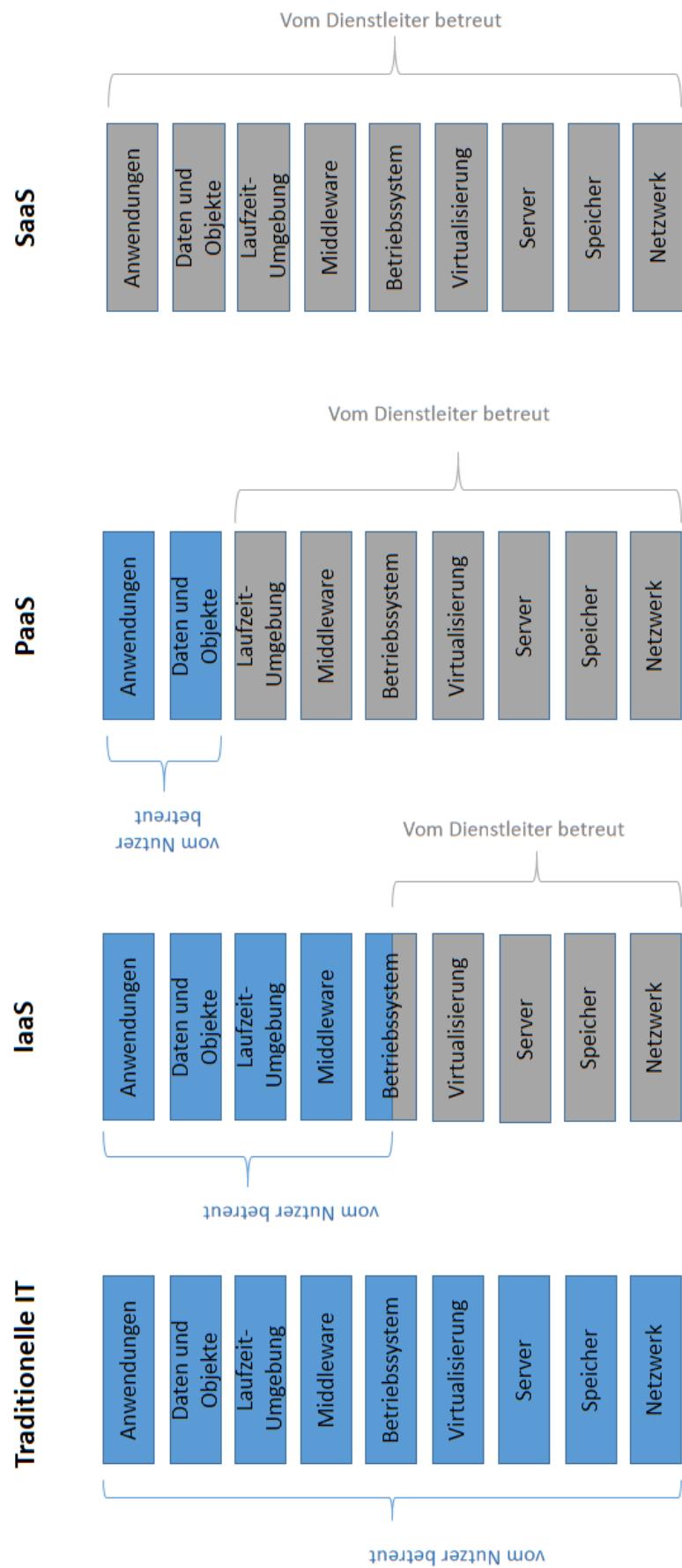


Abbildung 1: Dienstebenen in der Cloud
(Quelle: [24])

Auf dieser Ebene entstehen Entwicklungskosten, die nach Anzahl der Entwickler pro Monat berechnet werden und zusätzlich Entwicklerlizenzen enthalten können. Eventuell muss auch der Transfer von Daten bei der Berechnung beachtet werden [22, S. 47].

Auf der obersten Ebene können Endnutzer von SaaS profitieren. Sie nutzen eine Anwendung von jedem Cloud-fähigen Endgerät aus an jedem Ort und müssen nicht über die Ressourcen nachdenken, welche die Anwendung benötigt. Dafür müssen Sie jedoch auch in manchen Fällen Einschränkungen in der Konfigurierbarkeit der Software akzeptieren [19, S. 6]. Ein populäres Beispiel hierfür ist das Cloud-Gaming, das dem Nutzer ein Computerspiel bereitstellt, das trotz hoher Anforderungen an die Grafikkarte auf einem leistungsschwachen Computer oder mobilen Endgerät gespielt werden kann [20, S. 70]. Die Abrechnung der Dienste erfolgt meist pro Nutzer in Form einer monatlichen Gebühr [22, S. 47].

Die Cloud-Dienste können in unterschiedlichen Cloud-Typen angeboten werden. Eine öffentliche Cloud (*public cloud/external cloud*) wird von einem externen Dienstleister angeboten und betrieben [22, S. 45]. Der Nutzer kann meist mit Hilfe eines Selbstbedienungsportals⁴ selbstständig den benötigten Leistungsumfang angeben [20, S. 27-28].

Bei einer privaten Cloud (*private cloud*, teilweise auch *internal cloud*) gibt es verschiedene Definition. Laut Baun gehören der Anbieter und der Benutzer der Cloud zur selben organisatorischen Einheit, demselben Unternehmen [20]. In manchen Fällen können auch Kunden oder Partner des Nutzers zugreifen. Das NIST legt das Hauptaugenmerk der Kategorisierung darauf, dass die Cloud-Infrastruktur bei einer privaten Cloud für die ausschließliche Nutzung durch eine Organisation, eventuell mit mehreren Abteilungen, gedacht ist. Laut NIST spielt es dabei keine Rolle, wo die Infrastruktur sich befindet, wer sie besitzt und wer sie betreibt.

Barton [22, S. 46] beschreibt Unterformen der privaten Cloud, die von genau diesen Faktoren abhängen. Eine private Cloud, die der Nutzer selbst im eigenen Rechenzentrum betreibt, nennt er *insourced private cloud*. Wird der Betrieb an externe Dienstleister übergeben, hängt seine Definition davon ab, ob die IT-Infrastruktur dem externen Dienstleister gehört und sich physisch bei ihm befindet (*outsourced*) oder ob er auf die Infrastruktur des Kunden zurückgreift (*managed*). Der Nutzer bleibt bei allen Formen von privaten Clouds Besitzer seiner Daten und hat die volle Kontrolle über die Einhaltung von Datenschutzverordnungen in Bezug auf personenbezogene Daten [20, S. 27-28].

Werden Dienste aus privaten und öffentlichen Clouds zusammengesetzt, spricht man von einer hybriden Cloud. Bei Lastspitzen oder bestimmten, eigenständigen Funktionen kann es sinnvoll sein, diese in eine öffentliche Cloud auszulagern, während der Rest des Systems

⁴ Self-Service-Portal, siehe Abschnitt 2.3.3

in einer privaten Cloud verwaltet wird. Hierbei ist es wichtig, zu prüfen, ob die Auslagerung im Konflikt mit den Sicherheitsbestimmungen des Nutzers steht. Personenbezogene Daten sollten auch bei hybrider Cloudnutzung im privaten Systemteil verbleiben [20, S. 29]. Ebenfalls spricht man von einer hybriden Cloud, wenn eine Cloud mit einer traditionellen IT-Landschaft verbunden wird [22, S. 46]. Traditionell bedeutet in diesem Zusammenhang, dass die gesamte Infrastruktur im eigenen Rechenzentrum des Unternehmens untergebracht ist.

2.3 SAP Cloud Platform

Bei der SAP Cloud Platform handelt sich um ein Platform-as-a-Service-Angebot, das umfassende Dienste und Fähigkeiten der Applikationsentwicklung bereitstellt, die es dem Nutzer ermöglichen, Geschäftsanwendungen in der Cloud zu entwickeln, zu erweitern und mit jeder modernen SAP-Software [25, S. 126] sowie mit Software anderer Anbieter zu kombinieren [26, S. 8].

Ihren Beginn fand die SAP Cloud Platform im Oktober 2012 als *SAP HANA Cloud* mit den Hauptbestandteilen *SAP HANA AppServices* und *SAP HANA DBServices*, wie Abbildung 2 zeigt.



Abbildung 2: SAP HANA Cloud Platform
(Quelle: [27])

SAP HANA DBServices bezeichnet die SAP-HANA-Datenbank selbst und Hilfsmittel, mit denen die Daten durchsucht oder bearbeitet werden können, um anschließend als Grund-

lage für Vorhersagen zu dienen. Dabei werden sowohl spezielle Such- und Strukturerkennungsalgorithmen, als auch raumbezogene Analysen und Diagramme verwendet und eingesetzt. Strukturerkennungsalgorithmen werden genutzt um Muster und Zusammenhänge in Texten und Daten zu finden und Entscheidungen daraus abzuleiten [28].

Darauf aufbauend existieren zwei Arten von *SAP AppServices*: die *Enablement Services* und die *Application Services*. Erstere stellen Dienste zur Verwaltung von Applikationen und Systemen zur Verfügung, welche unter anderem Persistenz, Konnektivität, sowie Identitäts- und Dokumentenverwaltung ermöglichen. Letztere enthalten Werkzeuge für spezielle Szenarien, beispielsweise für mobile Applikationen und Portale, für die Zusammenarbeit innerhalb der Firma und mit Partnern sowie zur Integration von Systemen.

In den folgenden Jahren hat SAP die Plattform erweitert, ausgebaut und sie schließlich im Februar 2017 neu unter dem Namen *SAP Cloud Platform* veröffentlicht. Im April 2018 stehen dem Nutzer auf der SAP Cloud Platform Dienste in zwölf verschiedenen Kategorien zur Verfügung. Diese werden in Abschnitt 2.3.3 detailliert behandelt.

2.3.1 Zielgruppe

Nach Aussage von Matthias Steiner, einem Technologie-Experten von SAP, bietet die SAP Cloud Platform einen Mehrwert für neugegründete Firmen, die eine Applikation auf Basis von SAP HANA entwickeln wollen - entweder für Partner, die eine Erweiterung anbieten möchten oder für Kunden, denen eine speziell angepasste Lösung für die von ihnen genutzten SAP-Anwendungen fehlt [27]. Doch nicht nur neugegründete Firmen, sondern auch die Firmen *Bosch* und *Siemens*, die *Hamburger Hafenbehörde* und der Gabelstaplerhersteller *Still GmbH* haben bereits Projekte mit der SAP Cloud Platform umgesetzt. Siemens ist dadurch in der Lage, Vorhersagen über anfallende Wartungen zu treffen. Bosch ermöglicht LKW-Fahrern per App, einen sicheren Stellplatz für die Nacht zu finden. Im Hamburger Hafen werden in Echtzeit Containerschiffe koordiniert und bei Still GmbH Gabelstapler zu autonomem Handeln befähigt [29].

Partner von SAP nutzen die Fähigkeiten der SAP Cloud Platform zur Erstellung von Erweiterungen bestehender SAP-Systeme und verkaufen diese dann im *SAP App Center*. Bestehende SAP-Kunden suchen dort nach einer passenden Lösung für ihr Problem. Wenn sie nicht fündig werden oder nicht bereit sind, einer anderen Firma den gewünschten Betrag für eine Anwendung zu zahlen, können sie die SAP Cloud Platform nutzen, um SAP-Systeme nach ihren Wünschen anzupassen und zu erweitern.

2.3.2 Umgebungen und Regionen

Meldet sich ein Nutzer an der Testversion der SAP Cloud Platform an, empfängt ihn eine Startseite, die bereits einige Navigationsziele (siehe Abbildung 3) enthält. Links oben befindet sich ein Menü. Links unten findet sich eine Verknüpfung zu für den Anwender nützlichen Internetadressen (*Useful Links*). Darunter ist z.B. die Dokumentation der Entwicklungsplattform. Rechts oben erreicht der Nutzer Informationen zu seinem Benutzerkonto. Dort kann er sich für eine der angebotenen Sprachen Englisch, vereinfachtes Chinesisch, Koreanisch oder Japanisch entscheiden.

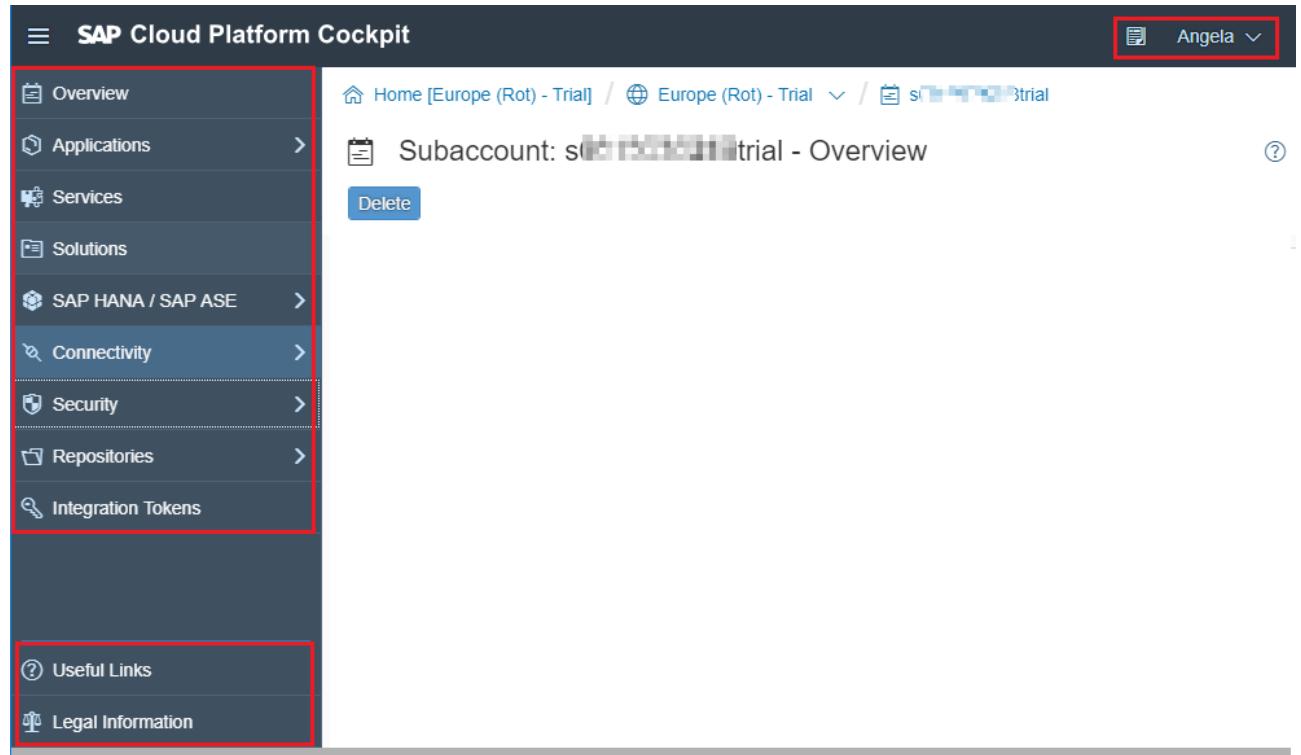


Abbildung 3: SAP Cloud Platform Cockpit der Neo-Umgebung
(Quelle: eigener Screenshot)

Die SAP Cloud Platform bietet Entwicklern zwei Umgebungen für ihre Entwicklungen an. Die *Cloud-Foundry*- und die *Neo*-Umgebung. Cloud Foundry ist ein quelloffenes PaaS zur Entwicklung von Anwendungen, das sich durch die unternehmensfähige Architektur, Portabilität, Multi-Cloud-Funktionen und offenen Standardprotokollen von den Konkurrenten abhebt [30]. Unter *Multi-Cloud* ist zu verstehen, dass die Plattform auf Infrastrukturdiensten von unterschiedlichen Anbietern betrieben wird [31].

SAP empfiehlt die Cloud Foundry Umgebung für folgende Entwicklungen [26, S. 11]:

- **12-Faktor-basierte Anwendungen:** „Das 12 Factor App Manifest gibt einige Empfehlungen dafür, wie Web-Apps für die Cloud am Besten entwickelt werden können“. Darin stehen 12 Faktoren, die bei einer SaaS-Anwendung für bessere Skalierbarkeit, und Portierbarkeit sorgen [32].

- **Mikrodienst-basierte Anwendungen:** Jeder Mikrodienst erfüllt nur eine Aufgabe. Durch die Verbindung mehrerer Mikrodienste entsteht eine komplexe Anwendung [33].
- **Native SAP HANA Anwendungen:** Wenn eine Anwendung speziell für ein bestimmtes Betriebssystem entwickelt wird, gilt sie als nativ. „Dadurch ist sichergestellt, dass alle Schnittstellen zu Hardware einheitlich funktionieren und die Ressourcen des Geräts optimal genutzt werden“ [34].
- **Anwendungen mit Bezug auf das Internet der Dinge:** „Das Internet der Dinge beschreibt eine globale Netzwerkinfrastruktur, an die Maschinen und Geräte angegeschlossen werden“ können [35, S. 6]
- **für maschinelles Lernen**

Aufgrund der offenen Standardprotokolle ist eine Applikation, die in SAP Cloud Foundry entwickelt wurde, auch bei anderen Anbietern von Cloud Foundry ausführbar. Dies wird durch die Zertifizierung der SAP Cloud Platform durch die *Cloud Foundry Stiftung* garantiert [36]. Im Standard werden Programmiersprachen wie *Java* und *NodeJS* angeboten [26, S. 11]. „Mit der Programmiersprache Java lassen sich plattformunabhängige Anwendungen entwickeln. Nur ein Texteditor, das Java SE Development Kit, kurz JDK, mit Java-Laufzeitumgebung und ein wenig Entwicklungsgeschick werden dafür benötigt“ [37]. *Node.js* ist „eine – wenn auch Server-seitiges – JavaScript-Laufzeitumgebung“ [38]. Cloud Foundry bietet dem Nutzer zusätzlich die Möglichkeit, weitere Programmiersprachen einzubinden [39].

Die zweite, *Neo* genannte, Umgebung ist die Weiterentwicklung der SAP NetWeaver Cloud, eines frühen PaaS von SAP, das seit 2011 existiert [40] und im März 2013 in die SAP Cloud Platform integriert wurde [41]. Mit ihr kann ein Kunde *Java*- [42], *SAP-HANA-XS*- und *HTML5*-Anwendungen in der Cloud entwickeln und betreiben [26, S. 11]. Die Hauptidee von SAP HANA XS ist es, einen voll funktionsfähigen Applikationsserver, einen Web-Server und eine Entwicklungsumgebung in die SAP-HANA-Datenbank einzubetten [43]. „Das Kürzel *HTML5* steht für Hypertext Markup Language und bezeichnet die fünfte Version der Internet-Auszeichnungssprache“ [44].

Wie Abbildung 4 zeigt, bietet die SAP Cloud Platform für jede Umgebung verschiedene Rechenzentrumsstandorte (Regionen) an, die der Nutzer wählen kann, um die Leistungsfähigkeit seiner Anwendungen in Bezug auf die Antwortzeit des Servers zu optimieren [26, S. 7]. In der Neo-Umgebung basieren alle Regionen auf der Infrastruktur von SAP. In der Cloud-Foundry-Umgebung greift SAP für den weltweiten Betrieb auf der Infrastruktur (IaaS) unterschiedlicher Cloud-Anbieter wie *Amazon Web Services*, *Microsoft*

und Google zurück. So wird etwa die Region *US East* mit Hilfe von Amazon Web Services betrieben. Einige Regionen sind noch im *Beta*-Stadium. Das heißt, dass sie sich in einer Testphase befinden und noch nicht für den produktiven Einsatz geeignet sind.

The screenshot shows the SAP Cloud Platform interface for managing regions. It is organized into three main sections:

- Cloud Foundry Environment:** Contains four regions:
 - Europe (Frankfurt) - Infrastructure: AWS
 - Europe (Netherlands) - Infrastructure: Azure
 - US Central (IA) Beta - Infrastructure: GCP (BETA)
 - US East (VA) - Infrastructure: AWS
- Neo Environment:** Contains nine regions:
 - Australia (Sydney) - Infrastructure: SAP
 - Brazil (São Paulo) - Infrastructure: SAP
 - Canada (Toronto) - Infrastructure: SAP
 - Europe (Amsterdam) - Infrastructure: SAP
 - Europe (Rot) - Infrastructure: SAP
 - Japan (Tokyo) - Infrastructure: SAP
 - KSA (Riyadh) - Infrastructure: SAP
 - Russia (Moscow) - Infrastructure: SAP
 - UAE (Dubai) - Infrastructure: SAP
 - US East (Ashburn) - Infrastructure: SAP
 - US East (Sterling) - Infrastructure: SAP
 - US West (Chandler) - Infrastructure: SAP
- Trial:** Contains one region:
 - Europe (Rot) - Trial - Infrastructure: SAP

Abbildung 4: Regionen in der SAP Cloud Platform
(Quelle: eigener Screenshot)

2.3.3 Dienste

Mit der Navigation zum Menüpunkt *Services* (siehe Abbildung 3) erreicht der Nutzer die in der aktuell gewählten Umgebung vorhandenen Dienste. Wie bereits in Abschnitt 2.3 erwähnt, bietet die SAP Cloud Platform diese in zwölf Kategorien an, die in Abbildung 5 aufgelistet sind. Die grünen Ziffern geben die Anzahl der zum Zeitpunkt dieser Arbeit verfügbaren Angebote in dieser Kategorie an. Eine detaillierte Erläuterung aller Dienste findet sich in der Dokumentation der SAP Cloud Platform [26] oder auf der Internetseite [45].

Im Bereich *Data & Storage* bietet SAP mittlerweile neben zwei HANA-Diensten auch Unterstützung für eigene (*SAP ASE*) und fremde (*PostgreSQL*) relationale Datenbanken, für die *NoSQL*-Datenbanken *MongoDB* und *Redis*, einen Dokumenten- und ein Objektspeicherdienst und Dienste zum Umgang mit großen, komplexen Datenmengen an.

Weil der Fokus der Plattform nicht mehr auf HANA liegt, wurde dieser Begriff auch aus dem Namen der Cloud Platform gestrichen.



Abbildung 5: Dienstkategorien auf der SAP Cloud Platform
(Quelle: [45])

Für Anwendungsentwickler besonders interessant ist die Kategorie *Dev Ops*. Diese enthält unter anderem eine integrierte Entwicklungsumgebung (IDE), einen Übersetzungsdienst und ein Hilfsmittel zur Quelltextverwaltung. Unter Dev Ops versteht man einen Ansatz zur Optimierung des Entwicklungsprozesses [46]. Mit der *SAP Web IDE Full-Stack* (Web IDE) können alle Ebenen einer Anwendung entwickelt werden, wie z.B. ein SAP-HANA-basiertes Modell, Geschäftslogik (in Java) und eine Fiori- bzw. HTML5-Oberfläche. Auf der SAP Cloud Platform findet sich eine zweite Version, die *SAP Web IDE* (ohne Zusatz). Seit dem 14. Mai 2018 rät SAP dazu, alle Projekte aus der einfachen Web IDE in die Full-Stack-Version zu migrieren und diese auch ausschließlich für Neuentwicklungen zu nutzen. Die dritte Version, die *SAP Web IDE for SAP HANA*, steht nicht auf der SAP Cloud Platform zur Verfügung, sondern wird zusammen mit SAP HANA ausgeliefert und kann für die Entwicklung nativer HANA-Apps genutzt werden [47].

Die SAP Web IDE Full Stack wird alle 14 Tage aktualisiert. Für Offline-Entwicklungen steht die *Web IDE Personal Edition* als lokale Installation zur Verfügung [48]. Diese ist als Ergänzung gedacht und kann nur zur Entwicklung von Benutzeroberflächen für Webbrowser und mobile Endgeräte verwendet werden. Sie unterliegt weiteren Einschränkungen [49, S. 424]. Die Auslieferung (engl. *Deployment*) von Anwendungen auf die SAP Cloud Platform ist nicht möglich. Unter Deployment werden alle Prozesse verstanden, die notwendig sind, um neue Soft- oder Hardware in einer Umgebung bereitzustellen. Dazu gehören die Installation, Konfiguration, das Ausführen und Testen sowie die Umsetzung nötiger Änderungen [50]. Zudem ist sie nur Nutzern vorbehalten, die auch Zugang zum Web-IDE-Dienst auf der SAP Cloud Platform haben.

Oberflächen von Anwendungen können mit Hilfe von Ressourcendateien in verschiedenen Sprachen angezeigt werden. Zur Erstellung von Übersetzungen bietet SAP den

Translation Hub an. In diesem werden Übersetzungsprojekte angelegt, welche die Quelldatei und die Zielsprachen definieren. Die Übersetzungen erstellt SAP mit Hilfe von mehrsprachigen Datenbanken [51].

Der *Git*-Dienst unterstützt Entwickler bei der Verwaltung von Quelldateien. Deren Versionen können in Git miteinander verglichen, zusammengefügt oder zurückgesetzt werden. Linus Torvalds hat die Software im Jahr 2005 als freie Software zur verteilten Verwaltung von Dateiversionen initiiert [52].

In der Kategorie Integration werden in dieser Arbeit der *OData Provisioning Service* und der Verbindungsdiensst (*Connectivity*) verwendet. Letzterer und der damit in Zusammenhang stehende *SAP Cloud Connector* werden in Abschnitt 2.3.5 behandelt.

Der OData-Provisioning-Dienst ermöglicht das Bereitstellen von OData-Diensten. Das *Open-Data-Protokoll* (OData) dient zur offenen Datenübertragung und basiert auf dem Programmierparadigma *REST*. Die Abkürzung steht für *Representational State Transfer* und „*ist eine Abstraktion der Struktur und des Verhaltens des World Wide Web*“ [53]. *REST* wird durch folgende sechs Kriterien definiert [54, S. 66]:

- Die Architektur muss aus Client und Server bestehen.
- Client und Server müssen durch eine einheitliche Schnittstelle verbunden werden, Ressourcen werden nach einheitlichen Verfahren identifiziert, abgefragt und manipuliert.
- Jede Anfrage des Clients muss alle Daten beinhalten, die der Server zur Verarbeitung braucht, da dieser keine Daten speichert.
- Anfragen sollen gepuffert werden um die Verarbeitungsgeschwindigkeit zu erhöhen. Die Pufferdauer muss aber spezifiziert werden, um zu vermeiden, dass Clients mit veralteten Daten arbeiten.
- Für den Client spielt es keine Rolle, ob er direkt oder über einen Mittler mit dem Server verbunden ist, der die Daten für die Antwort auf seine Anfrage liefert.
- Optional können Server ausführbaren Programquelltext an Clients ausliefern, um deren Funktionsumfang zu erweitern. Ein Beispiel hierfür ist die Übermittlung von JavaScript-Befehlen zur Anzeige einer HTML-Seite.

OData kann auf unterschiedliche Quellen zugreifen und Daten mischen und zusammenführen, ist über alle Versionen vollständig abwärtskompatibel und der Grad der Implementierung ist stets dem tatsächlichen Bedarf angepasst. Eine nachträgliche Erweiterung ist möglich.

Die REST-Kommunikation erfolgt mit folgenden Befehlen (Operationen) [54, S. 67]:

- **GET:** Ermitteln eines einzelnen Eintrags oder einer Liste von Einträgen

- **POST:** Erzeugen eines neuen Eintrags
- **PUT:** Verändern eines existierenden Eintrags
- **DELETE:** Löschen eines existierenden Eintrags
- **PATCH:** Aktualisieren einzelner Attribute eines existierenden Eintrags.

Ein Vorteil der Verwendung von OData-Diensten für SAP-Webanwendungen ist, dass der Webentwickler keine SAP-Spezifika verstehen muss. Es genügt, wenn er sich mit OData auskennt [54, S. 45]. SAP nutzte OData-Dienste bereits, bevor es die SAP Cloud Platform gab. ECC-Systeme sind in der Lage, OData-Dienste zu definieren, die dann über einen Server zum Zugang (*Gateway*) veröffentlicht und in Webanwendungen angesprochen werden können. Mit Nutzung des OData-Provisioning-Dienstes verlagert sich die Veröffentlichung des OData-Dienstes in die Cloud. In Abschnitt 4.1.2 wird das Einrichten eines OData-Dienstes und dessen Veröffentlichung erläutert.

In der Kategorie *User Experience* stehen Hilfsdienste zur Verbesserung des Nutzererlebnisses zur Verfügung. Mit dem *UI Theme Designer* können Anwendungen an das Erscheinungsbild⁵ der Firma (engl. *Corporate Identity*) angepasst werden. Mit dem *Portal-Dienst* können Selbstbedienungsportale erstellt werden, in denen sich Kunden Tag und Nacht einen Überblick über ihre Verträge und Profile verschaffen und einfache Transaktionen selbstständig im Internet abwickeln können. Diese Verlagerung von Serviceangeboten ins Internet entlastet Personal und spart Kosten [55]. Der Portal-Dienst ermöglicht die Erstellung und Veröffentlichung einer ansprechenden Benutzeroberfläche sowie die Zuordnung von Anwendungen zu berechtigten Nutzern. Dabei können von SAP entwickelte Apps zusammen mit eigenen Anwendungen angezeigt werden. Die genaue Vorgehensweise zur Veröffentlichung einer Fiori-Selbstbedienungsseite wird in Abschnitt 4.1.4 beschrieben.

2.3.4 Zugang zu Diensten auf der SAP Cloud Platform und Bezahlung

Es gibt zwei Ansätze zur Nutzung und Abrechnung der SAP Cloud Platform, ein abonnementbasiertes und ein nutzungsbasiertes Modell [56]. Die Entscheidung für das eine oder andere hängt von den Anforderungen und Wünschen des Nutzers ab.

Bei der abonnementbasierten Variante kann der Kunde im Voraus definieren, welche Services er nutzen möchte und wie diese skaliert sein müssen. Während der Abrechnungsperiode kann er diese Einstellungen nicht verändern. Er zahlt den vereinbarten Preis unabhängig davon, ob er die vereinbarten Dienste tatsächlich nutzt oder nicht. Für dieses Modell gibt es Pakete mit einem festgelegten monatlichen Preis und einzelne Services, die der

⁵ Engl. Corporate Identity

Nutzer zusätzlich zu diesen Paketen buchen kann [57]. Die Bezahlung erfolgt am Beginn der Abrechnungsperiode.

Beim nutzungsbasierten Modell profitiert der Nutzer von der Modularität der SAP Cloud Platform, die es ihm ermöglicht die einzelnen Dienste unabhängig voneinander zu nutzen. Einzelne Module lassen sich dann buchen, wenn sie benötigt, und abmelden, sobald sie überflüssig werden. Der Kunde hat Zugang zu einem Portal in dem er die Module verwalten und sich über ihre Nutzung und Auslastung informieren kann. Dies ermöglicht ihm eine flexible Skalierung seiner Dienste. Die Bezahlung der Services erfolgt nutzungsbasiert und mit *Cloud Credits*, die einem festgelegten Geldwert entsprechen und zum Vertragsbeginn auf ein Konto eingezahlt werden. Wenn das Guthaben erschöpft ist, muss der Kunde weitere Cloud Credits kaufen, um die SAP Cloud Platform weiterhin nutzen zu können.

Beide Bezahlmodelle gelten für eine auf Infrastruktur von SAP betriebene SAP Cloud Platform. Es ist trotzdem möglich eine SAP Cloud Platform im eigenen Rechenzentrum zu installieren, wenn man die geeignete Infrastruktur bereitstellt. Dies hat den Vorteil, dass die Kontrolle über die Daten beim Nutzer selbst verbleibt, jedoch muss er dann zusätzlichen finanziellen und zeitlichen Aufwand für die Beschaffung bzw. Verwaltung der Infrastruktur aufbringen (siehe Anhang A). Das Bezahlmodell konnte im Rahmen dieser Arbeit trotz mehrfacher Nachfrage bei verantwortlichen Personen nicht ermittelt werden.

Für Erstnutzer steht eine kostenlose Testversion der SAP Cloud Platform zur Verfügung, die es ermöglicht, einige Services auszuprobieren und erste Applikationen in HTML5 und Java zu entwickeln. SAP bietet kostenlose Online-Schulungen an, die zu großen Teilen mit der Testvariante der Plattform verfolgbar sind. In der Testversion kann der Kunde sich bereits einen Überblick darüber verschaffen, welche Dienste er braucht. Der Zugriff zur SAP Cloud Platform erfolgt über den Link: <http://hanatrial.ondemand.com>. Dort kann sich ein Benutzer mit einem bestehenden SAP-Konto anmelden oder ein neues Konto registrieren. Die Testlizenz ist zeitlich unbefristet. Eine Tabelle zur Übersicht über die Einschränkungen in den Testkonten von Cloud-Foundry- und Neo-Umgebung befindet sich im Anhang B. Beispielsweise ist es in der Neo-Umgebung nicht möglich, weitere Personen zum Testkonto hinzuzufügen und es können nur zwei lokal installierte Systeme mit der SAP-Cloud Platform verbunden werden. *Lokale* Systeme befinden sich im eigenen Rechenzentrum des Unternehmens (On-Premise), lokale Anwendungen sind auf einem Rechner installiert, der dem Nutzer physisch vorliegt.

2.3.5 Anbindung an andere Systeme

Die SAP Cloud Platform bietet für die Vernetzung von Systemen einen großen Vorteil: Anstatt je zwei Systeme durch Punkt-zu-Punkt-Verbindungen zu verknüpfen und damit eine unübersichtliche Menge von zu verwaltenden Verbindungen zu schaffen, stellt die SAP Cloud Platform einen zentralen Verbindungsknoten dar. Jedes Endgerät wird einmal daran angebunden und steht damit automatisch für jedes andere, mit der Platform verbundene, Endgerät zur Verfügung. Wie Abbildung 6 zeigt, ist die SAP Cloud Platform in der Lage, mobile, lokale und Cloud-basierte Anwendungen von Geschäftspartnern mit öffentlichen Behörden sowie sozialen Netzwerken zu verknüpfen.

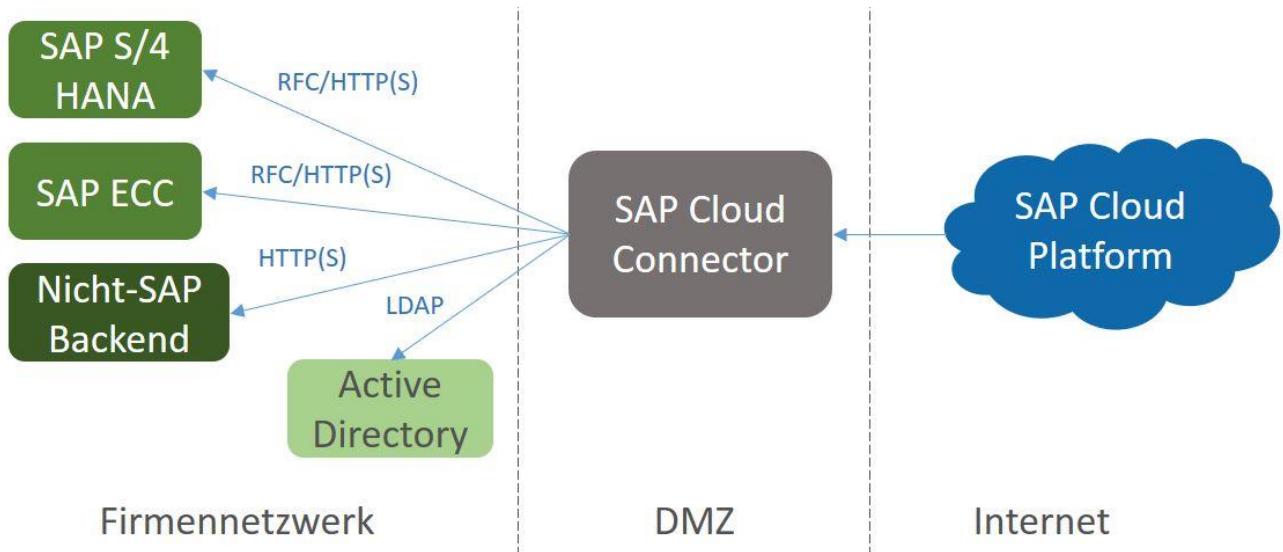


Abbildung 6: Integrationsmöglichkeiten von SAP Cloud Platform
(Quelle: Eigene Darstellung nach [58])

Die SAP Cloud Platform unterstützt dazu in der Neo-Umgebung folgende Protokolle⁶: *HTTP, SMTP, IMAP, POP3, LDAP, TCP* und *RFC*. In der Cloud-Foundry-Umgebung kann nur *HTTP* genutzt werden [26, S. 28f]. Die Protokolle *IMAP*, *POP3* und *SMTP* dienen zum Senden und Empfangen von E-Mails. Über *LDAP* können zum Beispiel Nutzerverwaltungssysteme wie das *Active Directory* von *Microsoft* angeschlossen werden. Die Verbindung zu On-Premise-Systemen von SAP oder anderen Anbietern kann über *HTTP*, *TCP* und *RFC* erfolgen. *HTTP* ermöglicht zusätzlich die Verbindung zu Internetdiensten. *RFC* kann genutzt werden, um Funktionsbausteine in SAP-Systemen aufzurufen.

In der Dokumentation der SAP Cloud Platform lassen sich Hilfsmittel zur Unterstützung von der Integration in und Verbindungen zu Drittsystemen finden. Eines davon ist von besonderer Bedeutung für diese Arbeit: der Cloud Connector Dienst.

⁶ Siehe Abkürzungsverzeichnis in Abschnitt 0.

Dabei handelt es sich um einen Hilfsdienst der SAP, mit dem eine sichere Verbindung zwischen lokalen Systemen und der SAP Cloud Platform erzeugt wird. Potenziell schädlichen Angriffen, z.B. Man-In-The-Middle-Angriffen, kann so vorgebeugt werden. Ein Angreifer schaltet sich zwischen zwei Parteien und gibt sich als eine von ihnen aus, um an die Daten der anderen Partei zu gelangen [59].

Um den Konnektor nutzen zu können, ist eine kostenlose Software von SAP notwendig, die als installierbare und portable Version zur Verfügung steht. Der Sicherheitsaspekt des Konnektors wird in Abschnitt 3.6 näher beleuchtet. Mit der Einrichtung beschäftigt sich der Abschnitt 4.1.1.

2.3.6 Verfügbarkeit und Wartung

Bei der Entscheidung über die Verwendung eines Cloud-Dienstes für unternehmerische Zwecke spielt die Frage nach der Verfügbarkeit und der Wartung eine große Rolle. Mit Abgabe der Verantwortung für die Einrichtung und den Betrieb der Cloud-Infrastruktur verliert der Nutzer auch die Kontrolle darüber, wann die Anwendung eine Wartung erhält, ob er darüber vorher informiert wird und wie schnell der Ausfall behoben ist. Daher haben sich so genannte Dienststufenvereinbarungen (*Service Level Agreement*, kurz SLA) eingebürgert. Diese legen genau fest, in welchem Rhythmus ein Dienst gewartet wird und wie lange er dafür ausfallen darf. Ebenso verpflichtet sich der Anbieter zu einer prozentual festgelegten Verfügbarkeit des Dienstes und zu einem Zeitfenster, das ihm für die Behebung von Ausfällen zur Verfügung steht. Die Dienststufen hängen von der Wichtigkeit des Dienstes ab.

Für die SAP Cloud Platform gelten die von SAP definierten SLA für Cloud-Dienste und eine spezielle Ergänzung. Beide Dokumente lassen sich im Internet finden [60]. Von den Vereinbarungen ausgeschlossen werden Hilfsprogramme, die der Nutzer herunterlädt und im eigenen Umfeld installiert. Dazu gehören unter anderem die SAP Web IDE Personal Edition und der SAP Cloud Connector. Ebenfalls ausgenommen werden *Beta*-Dienste, da diese sich noch in einer Testphase befinden.

Bei Nichterfüllung der Versprechungen durch den Anbieter hat der Nutzer ein Anrecht auf eine Gutschrift. Diese muss er jedoch rechtzeitig beantragen. Außerdem muss der Ausfallgrund im Verantwortungsbereich von SAP liegen und darf nicht mindestens 5 Tage vorher angekündigt worden sein. Über die prozentuale monatliche Systemverfügbarkeit erfolgt ein monatlicher Bericht durch den Anbieter.

Die generelle Systemverfügbarkeit von Clouddiensten legt SAP auf 99,5% fest. Im ergänzenden Dokument für die Cloud Platform werden sogar 99,9% genannt und einzelne Services, wie der Dokumentendienst ausgeschlossen. Außerdem gibt es eine Hochverfügbarkeitsoption (*High Availability*) für HANA- und ASE-Dienste, bei der mehrere Instanzen

des Dienstes gemeinsam genutzt werden und dadurch eine höhere Verfügbarkeit (99,95%) möglich ist. Von den Verfügbarkeitsversprechen ausgeschlossen sind regelmäßige Wartungsfenster und Zeiten für wichtige Upgrades (siehe Abbildung 7). Auch die gesamte Testversion der SAP Cloud Platform unterliegt nicht den Dienststufenvereinbarungen. Den aktuellen Status wichtiger Komponenten jeder Region veröffentlicht SAP im Internet unter <https://sapcp.statuspage.io/>.

CLOUD SERVICE	Regelmäßige Wartungsfenster Anfangszeit in UTC nach Region	Dauer	Wichtige Upgrades Zeitraum in UTC nach Region
SAP Cloud Platform	Keine Ausfallzeit für Wartung erforderlich	-	bis zu viermal im Jahr APJ: FR 14:00–18:00 Uhr, UTC Europa: FR 22:00 Uhr bis SA 02:00 Uhr, UTC Amerika: SA 04:00–08:00 Uhr, UTC
SAP Cloud Platform API Management	APJ: SA 15:00 Uhr, UTC Europa: SA 22:00 Uhr, UTC Amerika: SO 04:00 Uhr, UTC	2 Stunden	bis zu viermal im Jahr APJ: FR 14:00 bis SA 14:00 Uhr, UTC Europa: FR 22:00 bis SA 22:00 Uhr, UTC Amerika: SA 04:00 bis SO 04:00 Uhr, UTC
SAP Cloud Platform Gamification	alle zwei (2) Wochen: APJ: SA 15:00 Uhr, UTC Europa: SA 22:00 Uhr, UTC Amerika: SO 04:00 Uhr, UTC	1 Stunde	bis zu viermal im Jahr APJ: FR 14:00–18:00 Uhr, UTC Europa: FR 22:00 Uhr bis SA 02:00 Uhr, UTC Amerika: SA 04:00–08:00 Uhr, UTC

Abbildung 7: Wartungsfenster und Upgrade-Frequenz für Cloud-Dienste
(Quelle: eigener Screenshot)

3 Vorüberlegungen zur Implementierung einer Webanwendung

Aufbauend auf den vorher erarbeiteten theoretischen Grundlagen werden nun die Anforderungen für die zu entwickelnde Anwendung und die Herangehensweise an die Evaluierung der SAP Cloud Platform festgelegt und beschrieben.

3.1 Bestehende Anwendung mit Eigenentwicklung

Die Wahl der Eigenentwicklung, die mit Hilfe der SAP Cloud Platform umzusetzen ist, fällt auf das Szenario *Geschäftspartner anzeigen*. Im ECC-System liegen Daten über alle Kunden eines Energieversorgers vor. Diese heißen bei SAP Geschäftspartner und besitzen Stammdaten, wie Adressen oder ihre Bankverbindungen. Durch Zuordnung bestimmter Unterscheidungsmerkmale wird festgelegt, welche Felder ihnen angezeigt werden und welche Applikationen sie ausführen können. SAP bietet die Unterscheidungskriterien Organisation, Person und Gruppe an.

Im energiewirtschaftlichen Umfeld werden Partner unter anderem in Vertriebspartner und Lieferant unterteilt. Einige Kunden von ASP benötigen zusätzlich die Unterscheidung in Einzel- und Bündelkunden. Einzelkunden sind Endverbraucher, die im Regelfall eine einzige Adresse haben, für die sie Energie vom Versorger beziehen. Als Bündelkunden bezeichnet man eine „Einkaufsgemeinschaft, die ihren Strombedarf für mehrere Abnahmestellen von einem einzelnen Lieferanten decken lässt (z. B. bundesweite Energieversorgung von Filialen eines Unternehmens durch einen einzigen Versorger)“ [61]. Im ECC-System ist diese Unterscheidung nach Kundenart nicht standardmäßig vorgesehen und musste daher in einer Eigenentwicklung umgesetzt werden.

Im ECC-System erfolgt die Anzeige des Geschäftspartners beispielsweise mit der Transaktion *BP*. Eine Transaktion ist eine Zeichenfolge, mit der ein Programm schnell aufgerufen werden kann. Abbildung 8 zeigt den Startbildschirm dieses Programms in der SAP Standardbenutzeroberfläche *SAP GUI*⁷. Links unter der zweiten Menüleiste befindet sich eine graue Box mit zwei Reitern. Der Reiter *Suche* dient dazu, eine Auswahl von Geschäftspartnern als Liste auszugeben. Als Suchkriterium kann hier der Name, die Nummer oder die Adresse des Kunden gewählt werden. Führt man einen Doppelklick auf einen der Treffer durch, werden die Stammdaten im rechten Teil des Bildschirms in acht Reitern angezeigt (siehe Abbildung 9). Der vierte Reiter, mit dem Namen *Steuerung*, beinhaltet die Information über die Kundenart bzw. ob der Geschäftspartner ein Einzel- oder Bündelkunde ist.

⁷ Vgl. Abkürzungsverzeichnis.

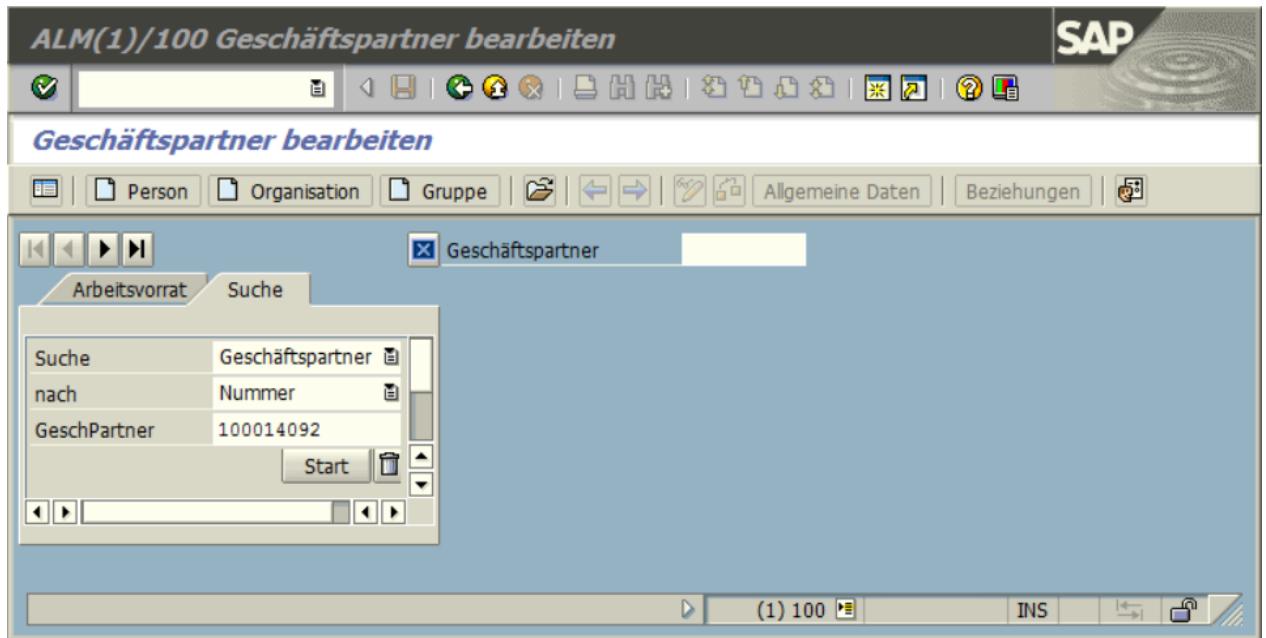


Abbildung 8: Startbild der Transaktion BP
(Quelle: eigener Screenshot)



Abbildung 9: Ansicht der Transaktion BP
(Quelle: eigener Screenshot)

Die zu entwickelnde Webanwendung soll in vereinfachter Weise die Funktion der Transaktion *BP* abbilden. Ein Geschäftspartner soll anhand seiner Nummer auffindbar sein und mit den Stammdaten (Name, Vorname, Kundenart, Vertragskonten) angezeigt werden.

3.2 Anforderungen an die Anwendung

Vor der Umsetzung der Webanwendung müssen einige Anforderungen an die Anwendung, wie ihre Funktionalität, Zielgruppe und die Laufzeitumgebung definiert werden.

Bei der Funktionalität ist zwischen Muss-, Soll- und Kann-Kriterien zu unterscheiden. Erstere sind zwingend zu erfüllen, damit die Applikation einen Nutzen bringt, während es sich bei Soll-Kriterien um die Beschreibung weiterer, geplanter Ausbaustufen handelt. Kann-Kriterien geben einen Ausblick auf weitere Funktionalitäten, die in der Anwendung eingebaut werden können, jedoch nicht Teil des aktuellen Auftrags sind [62]. Der Kriterienkatalog wird aus Abbildung 10 erkenntlich. Im Rahmen dieser Arbeit werden nur die Musskriterien

umgesetzt, da diese genügen um die Machbarkeit der gestellten Aufgabe zu belegen. Weitere Anforderungen können von ASP im Anschluss umgesetzt werden, etwa um die SAP Cloud Platform weitergehend zu analysieren.

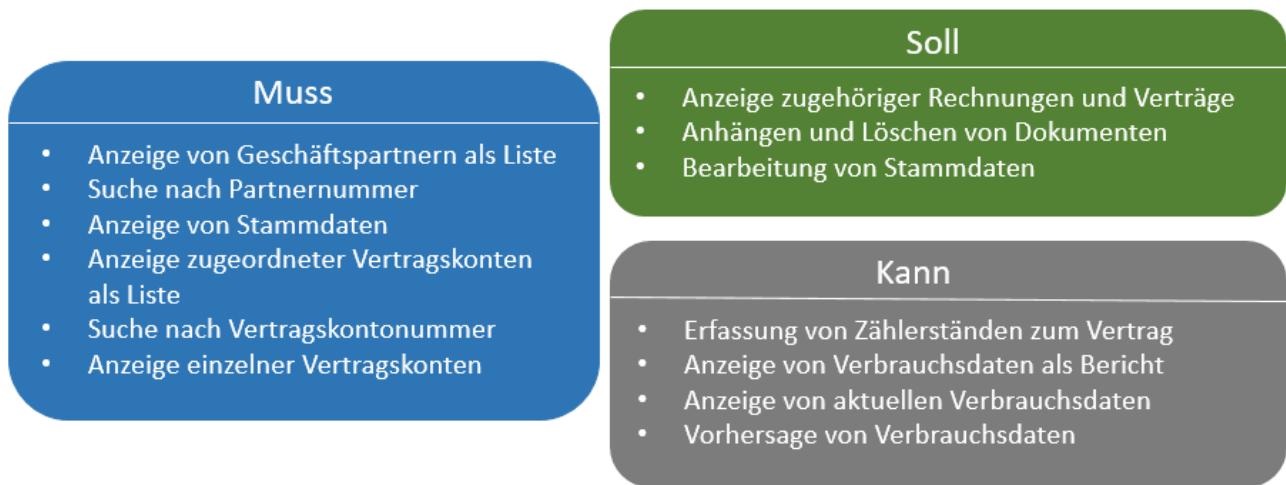


Abbildung 10: Anforderungen zur Umsetzung der Webanwendung
(Quelle: eigene Darstellung)

Zwingend notwendig für eine sinnvolle Anwendung ist die Anzeige von Geschäftspartnern als Liste. Diese ist nach der Partnernummer durchsuchbar. Nach der Auswahl eines Geschäftspartners erscheint eine Detailansicht mit den grundlegenden Stammdaten Nachname, Vorname, Kundenart und Partnernummer. Zudem werden die dem Partner zugeordneten Vertragskonten angezeigt. Bei Markierung erfolgt eine Navigation zu den einzelnen Vertragskonten.

Weiterführend kann die Anzeige von Rechnungen und Verträgen, das Anhängen und Löschen von Dokumenten und die Bearbeitung von Stammdaten implementiert werden. Gleiches gilt für die Erfassung von Zählerständen zum passenden Vertrag und die Anzeige von Verbrauchsdaten als Bericht, als aktuelle Auskunft und als Vorhersage. Zielgruppe der Anwendung zur Anzeige von Geschäftspartnern sind Mitarbeiter des Energieversorgers, welche die Verträge oder Stammdaten von Kunden ansehen, verwalten oder ändern müssen. Mit Vergabe passender Berechtigungen ist dafür zu sorgen, dass Mitarbeiter nur die Aktionen ausführen können, die ihnen im Rahmen ihrer Tätigkeit erlaubt sind.

Die Anwendung soll in Webbrowsern und auf verschiedenen Endgeräten lauffähig sein. Dies ist bei Fiori-Apps der Fall. Dennoch gibt es in der Anzeige Unterschiede, die im Programmquelltext berücksichtigt werden müssen. In dieser Arbeit soll die Anwendung zur Anzeige im Webbrowser von Desktop-PCs und Laptops, insbesondere in Google Chrome optimiert werden. Der Aufruf der Anwendung erfolgt von der Fiori-Startseite, die auch als Launchpad bezeichnet wird.

3.3 Fiori

Hinter dem Namen Fiori verbirgt sich laut Aussage von SAP „kein Produkt, sondern ein Paradigmenwechsel in der Anwendungsentwicklung!“ [63, S. 18]. Es handelt sich um eine „Art, wie Anwendungen bei SAP implementiert werden“ [63, S. 25], um sie besser als die bisherige Standardoberfläche SAP GUI an die heutigen Bedürfnisse von Softwarenutzern anzupassen. Eine Fiori-Anwendung zeigt dem Nutzer rollenbasiert nur die Daten und Funktionen an, die er benötigt. Die Applikationen sind benutzerfreundlich und bilden nur je eine Funktion ab. Alle Fiori-Apps sind in einem einheitlichen *Look-and-Feel* gehalten und können auf jedem vom Nutzer gewählten Endgerät (Laptop, Smartphone, Tablet, Smartwatch) aufgerufen werden [63, S. 26, 29]. *Look-and-Feel*, bezeichnet die „Beschreibung des optischen Erscheinungsbildes einer Benutzeroberfläche in Kombination mit ihren Bedienungseigenschaften“ [64].

Fiori-Anwendungen bestehen aus einer Oberfläche (*Frontend*) und einem Datenlieferanten (*Backend*). Für die Oberflächenentwicklung kommt das *SAP UI Development Toolkit for HTML5 (SAPUI5)* zum Einsatz. Dieses wurde von SAP entwickelt, um die Implementation von browserbasierten Anwendungen trotz der Vielfalt an mobilen Plattformen zu ermöglichen [63]. Es basiert auf HTML5 und verwendet CSS3 zur visuellen Gestaltung. SAPUI5 richtet sich nach dem Architekturmuster *Model-View-Controller(MVC)* [63, S. 360]. Darunter versteht man, dass die Anwendung in drei Module unterteilt ist: Modell (*Model*), Ansicht (*View*) und Steuerung (*Controller*). Das Model enthält Daten, die View zeigt Daten an und ermöglicht Nutzerinteraktionen, wie das Betätigen einer Schaltfläche. Die Oberfläche einer Anwendung kann aus einer oder mehreren Ansichten bestehen. Der Controller dient als Vermittler zwischen beiden und reagiert auf Nutzerinteraktionen [65, S. 29]. Neben SAPUI5 gibt es noch *OpenUI5*, eine quelloffene Lösung, die in den Kernkomponenten exakt mit SAPUI5 übereinstimmt. Da diese jedoch nicht alle Teile der Bibliothek enthält, empfiehlt sich im direkten SAP-Umfeld die Nutzung der proprietären Version [66].

Die Oberfläche einer Fiori-Anwendung unterliegt einer Richtlinie, der *SAP Fiori Design Guideline* [67]. Ihre Anzeige erfolgt üblicherweise im Fiori-Launchpad. Damit wird eine Startseite im Browser bezeichnet, die alle Anwendungen anzeigt, für die ein Nutzer berechtigt ist. Die Verbindung zwischen dem Frontend und dem Datenlieferanten schaffen hauptsächlich OData-Dienste, die im Gateway veröffentlicht werden, nachdem sie beim Datenlieferanten modelliert wurden [68]. Jede Anwendung, die diesen Kriterien entspricht, darf *Fioriähnlich* genannt werden [10].

SAP unterscheidet bei den durch das Unternehmen selbst zur Verfügung gestellten Fiori-Anwendungen zwischen *analytischen Anwendungen*, *Infoblättern* und *transaktionalen*

Anwendungen. Analytische Applikationen „sammeln Kennzahlen und zeigen sie im Browser an“ [10]. Infoblätter enthalten Stammdaten von zentralen Objekten und zeigen diese kontextbezogen an. Sie ermöglichen eine Navigation zu anderen, im Zusammenhang stehenden Infoblättern. Sowohl analytische Anwendungen als auch Infoblätter sind an die Nutzung einer SAP-HANA-Datenbank gebunden. Unter transaktional ist zu verstehen, dass der Nutzer in der Anwendung eine Aufgabe ausführt, beispielsweise Urlaubsanträge seiner Mitarbeiter genehmigt. Diese Fiori-Applikationen können mit jeder darunterliegenden Datenbank genutzt werden.

Wie bereits in Abschnitt 1.2 erwähnt, können Fiori-Benutzeroberflächen auch für ECC-Systeme implementiert werden. Dies trifft jedoch speziell nur auf transaktionale Anwendungen zu. Die inhaltlichen Anforderungen der umzusetzenden Anwendung definieren, dass Stammdaten zu Kunden von Energieversorgern angezeigt werden sollen. Dementsprechend ist die Anwendung als Infoblatt einzuordnen und nur mit Hilfe einer SAP-HANA-Datenbank betreibbar. Um eine transktionale Anwendung zu erhalten, muss eine spezielle Aufgabe in Bezug zum Objekt *Geschäftspartner* gewählt werden, z.B. die Adressänderung oder die Änderung der Kundenart. Im nächsten Abschnitt wurde geprüft, ob eine transktionale Anwendung existiert, die im Zusammenhang mit einer speziellen Aufgabe ausgewählte Stammdaten anzeigt.

3.4 Verfügbarkeit und Erweiterung von Standardfunktionen

Verfügbare Fiori-Anwendungen konnten in der *Fiori Apps Reference Library* unter <https://fioriappslibrary.hana.ondemand.com> gesucht werden. Die Filterbedingungen waren für diesen Fall wie folgt festzulegen:

- **Product Suite:**
 - SAP Business Suite (ECC)
- **Required Back-End Product:**
 - SAP ERP
 - SAP ERP, min. EHP 7
 - SAP ERP, min. EHP 8
- **Database:**
 - Any DB

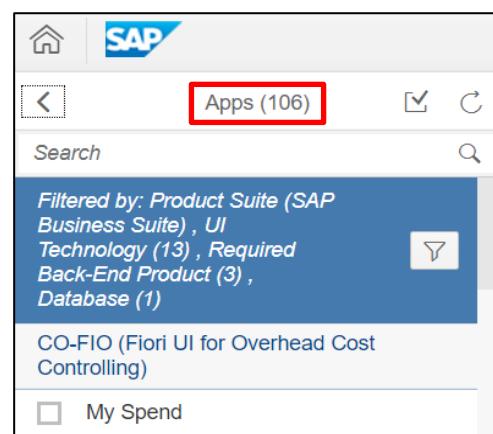


Abbildung 11: Bildschirmaufnahme der Filterung von SAP-Fiori-Anwendungen
(Quelle: eigener Screenshot)

Im Fall der Datenbank ließ sich nur nach *HANA DB* oder *Any DB* filtern. Auch bei *Any DB* war die SAP-HANA-Datenbank nicht ausgeschlossen. Die Auswahl garantierte aber, dass die angezeigten Anwendungen auch auf ECC-Systemen (*Business Suite*) ohne SAP-

HANA-Datenbank lauffähig waren. Als Ergebnis wurden nun 106 Anwendungen angezeigt, die mit Hilfe von Suchbegriffen weiter eingeschränkt werden konnten (siehe Abbildung 11). Eine Suche nach *business*, *partner*, *bp*, *customer* oder *address* lieferte kein passendes Ergebnis. Nach Entfernung aller Filter und der Suche nach *business partner*, ergaben sich 137 Anwendungen als Ergebnis, darunter die Applikation *Display Business Partner (BUP3)* [69], welche jedoch nur für S/4-HANA-Systeme mit HANA-Datenbank geeignet war.

Weil keine passende Standardanwendung gefunden werden konnte, war im Rahmen dieser Arbeit eine eigene energiewirtschaftliche Anwendung zu entwickeln. Die Einbindung und Anpassung von durch SAP erstellten Fiori-Anwendungen wurde nicht gezeigt.

3.5 Architektur der eigenen Anwendung

Für die Eigenentwicklung der Anwendung standen verschiedene Technologien und mögliche Software-Architekturen zur Verfügung. Die App sollte Fiori-ähnlich sein. Das heißt ihre Oberfläche baut auf dem SAPUI5-Framework auf und hält sich an die Fiori Design Guideline. Das Frontend bezog seine Daten mit Hilfe eines OData-Dienstes aus dem Backend-System. Zudem war die Anwendung in der SAP Cloud Platform zu entwickeln und anschließend auf einem Fiori-Launchpad anzusehen.

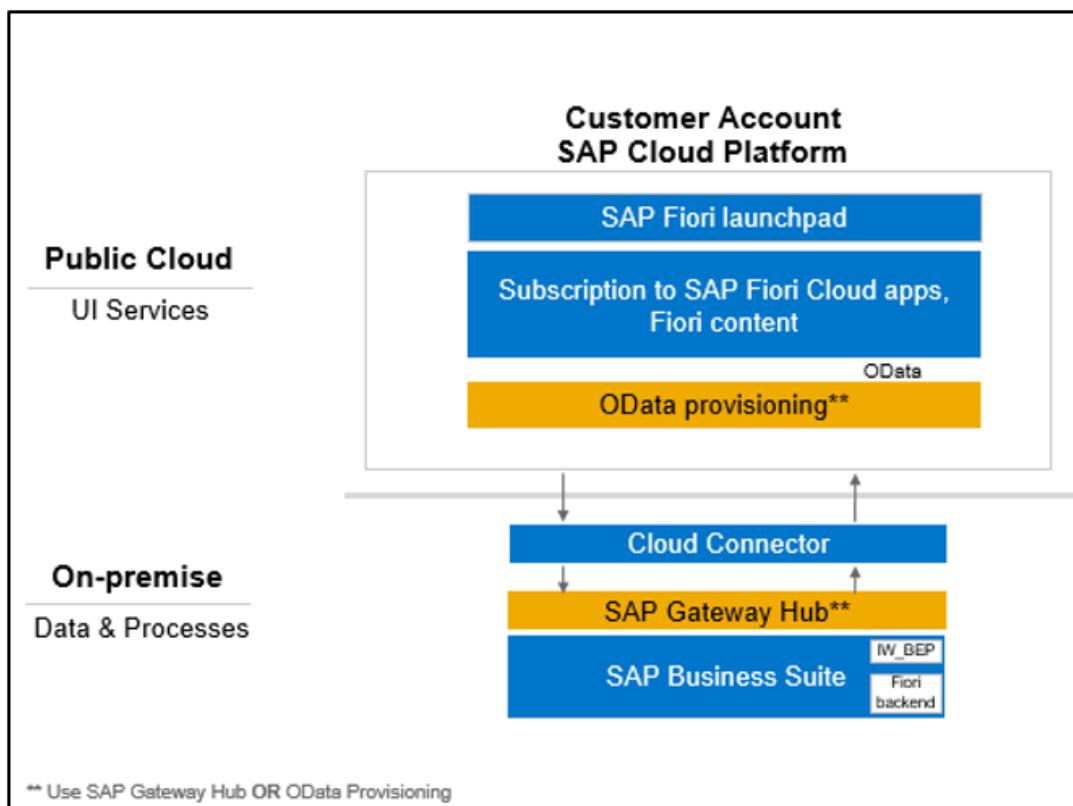


Abbildung 12: Übersicht über die mögliche Architektur von Fiori-Applikationen
(Quelle: [70])

Abbildung 12 zeigt zwei Varianten für die Verbindung mit dem OData-Dienst. Wie bereits in Abschnitt 2.3.5 dargestellt, wird ein SAP Cloud Connector benötigt, um eine Verbindung

zwischen der SAP Cloud Platform und lokalen Systemen, wie dem Gateway Hub, herzustellen. Dieser übernimmt dann die Bereitstellung des OData-Dienstes. Sollte kein separater Gateway-Server zur Verfügung stehen, ist es auch möglich zur Bereitstellung des OData-Dienstes den OData-Provisioning-Dienst der SAP Cloud Platform zu nutzen. Dafür wird im Backend die Software-Komponente *Business Enablement Provisioning* (IW-BEP) [71] benötigt. Beide Varianten wurden in dieser Arbeit zur Prüfung ihrer Machbarkeit nachvollzogen.

Ebenso ist es denkbar, an Stelle eines separaten Gateway-Servers ein direkt auf dem Backend-Server installiertes Gateway zu verwenden (*embedded deployment*) [72]. Dies wird jedoch von SAP als kritisch angesehen, weil dadurch bei einer Aktualisierung der Gateway- oder SAPUI5-Version ein höherer Aufwand entsteht [63, S. 35-36]. Zudem kann ein Angreifer, der Zugang zum SAP Gateway Hub erlangt, auf alle Daten des Backend-Systems zugreifen.

Als separates Gateway wurde der Frontend-Server SGW verwendet, der eine SAP-Gateway-Komponente beinhaltete und per RFC-Verbindung mit dem SAP-ECC-System SBE verbunden war. *Remote Function Call* (RFC) bezeichnet von SAP entworfene „Verfahren, mit denen Funktionen in einem entfernten System aufgerufen werden“ [73]. Das SBE wurde als Backend verwendet und war die Kopie eines bei einem Energieversorger eingesetzten Produktivsystems. Daher waren im System auch Kopien von Kundendaten enthalten. Im Rahmen dieser Arbeit sollte aus Gründen des Datenschutzes (siehe Abschnitt 3.6) mit anonymisierten oder fiktiven Kundendaten gearbeitet werden.

Über das Gateway im SGW wurde bereits ein OData-Dienst exponiert. Er enthielt alle Daten, die für diese Anwendung benötigt werden: Kundennamen, Vertragskonten und die Kundenart. Der Service hieß `/ASPSP/ERP_UTILITIES_UMC`.

ECC-Systeme enthalten im Standard keine OData-Dienste für Versorgungsunternehmen. Sie können durch zusätzliche Produkte ergänzt werden. Eine solche Erweiterung ist u.a. *SAP Multichannel Foundation for Utilities and Public Sector* (MCF). Die Anwendung gibt Versorgungsunternehmen und öffentlichen Behörden die Möglichkeit, Selbstbedienungsportale für ihre Kunden zur Verfügung zu stellen. Kunden eines Energieversorgers können in einer solchen Webanwendung unter anderem Ihre Stammdaten einsehen und ändern oder ihren selbstabgelesenen Zählerstand eintragen. Seit es die so genannten intelligenten Stromzähler (*Smart Meter*) gibt, kann dem Kunden in einem Selbstbedienungsportal auch der aktuelle Verbrauch und eine entsprechende Verbrauchsvorhersage angezeigt werden. Bei dem Selbstbedienungsportal handelt es sich in der Regel um eine Webanwendung, die ihre Daten mit Hilfe von OData-Diensten aus dem Backend bezieht. MCF bringt daher eine

eigene OData-Bibliothek mit, die für energiewirtschaftliche Anwendungen genutzt werden kann. Es war zu prüfen, ob der erwähnte Dienst /ASPSP/ERP_UTILITIES_UMC⁸ für die Implementierung in dieser Arbeit verwendet werden konnte oder ein neuer OData-Dienst exponiert werden musste.

Nach Abschluss der Entwicklung stehen zwei Varianten zur Auslieferung und zur Anzeige der Anwendung zur Verfügung. Bei der Cloud-Variante wird die Anwendung in der SAP Cloud Platform veröffentlicht. Dort kann im Portal-Dienst ein Fiori-Launchpad erzeugt werden, dem die Anwendung nach erfolgreicher Auslieferung in der Cloud zuordenbar ist. Diese Variante wird in Abbildung 12 dargestellt. Bei der lokalen Alternative wird ein Fiori-Launchpad auf dem Applikationsserver eingerichtet, dem die Anwendung nach Durchführung des Deployment dorthin zugeordnet werden kann. Der Server SGW übernahm in dieser Arbeit eine Doppelrolle als Gateway- und Applikationsserver und stellte bereits ein Fiori-Launchpad zur Verfügung, so dass die Anwendung ohne weitere Hilfsmittel angezeigt werden konnte. Eine Anleitung zur Einrichtung des Fiori-Launchpads findet sich im Internet [74] oder bei *Englbrecht* [63, S. 253-259]. Um die verschiedenen Optionen vergleichen und ihre Machbarkeit bewerten zu können, wurden beide Varianten im Rahmen dieser Arbeit umgesetzt.

3.6 Sicherheit und Datenschutz

Personenbezogene Daten unterliegen besonderem Schutz durch den Gesetzgeber und müssen daher entsprechend vor Missbrauch durch Dritte gesichert werden. Für die in dieser Arbeit entwickelte Anwendung ist es daher wichtig, dass personenbezogene Daten nur befugten Personen angezeigt werden und dass sie bei der Übertragung vom Backend zum Frontend so gut wie möglich geschützt sind. Dies trifft für die Entwicklungsdauer als auch für die produktive Anwendung zu.

Zusätzlich zum deutschen Bundesdatenschutzgesetz existiert seit dem 25. Mai 2018 auch die europäische Datenschutzgrundverordnung (EU-DSGVO). Diese gilt für Unternehmen mit Niederlassung in der *Europäischen Union* (EU), die personenbezogene Daten verarbeiten, unabhängig davon, ob die Verarbeitung in der EU stattfindet. Zudem gilt sie aber auch für Unternehmen mit Sitz außerhalb der EU, wenn diese einer Person in der EU Waren oder Dienstleistungen anbieten oder deren Verhalten beobachten [27]. ASP verarbeitet die personenbezogenen Daten, die ihre Kunden über deren Kunden speichern müssen, um die Energieversorgung zu ermöglichen.

⁸ Dieser ist eine Anpassung des Dienstes ERP_UTILITIES_UMC aus der OData-Bibliothek in MCF.

Als deutsches Unternehmen unterliegt ASP in vollem Maße der Datenschutzgrundverordnung und muss sicherstellen, dass Anwendungen, die dem Kunden zur Verfügung gestellt werden, den Datenschutzrichtlinien entsprechen. Für die Authentifizierung eines Nutzers an SAP-Backend-Systemen stehen verschiedene Authentifizierungsoptionen zur Verfügung, die teilweise auch *Single Sign-On (SSO)* ermöglichen. SSO bedeutet, dass sich ein Nutzer während einer gültigen Sitzung nur einmal anmelden muss und alle gekoppelten Systeme und Anwendungen ohne weitere Anmeldung nutzen kann [75]. „Eine Sitzung (englisch session) bezeichnet in der EDV eine stehende Verbindung eines Clients mit einem Server“ [76].

- **Benutzer-ID und Kennwort (Basic Authentication):** Dies ist das von ASP für die Backend-Systeme genutzte Szenario. Ein Nutzer wird im System eingerichtet und bekommt ein Initialkennwort zugewiesen, das er beim ersten Login ändern muss. Eine Umsetzung als SSO ist hiermit nicht möglich. Wenn ein OData-Dienst als privat deklariert wurde, ist er ebenfalls nur mit den Zugangsdaten des entsprechenden Systems erreichbar.
- **Anmeldetickets:** Ein in der Systemlandschaft befindliches Portal stellt einem Nutzer, der sich dort mit Passwort und Benutzername anmeldet, ein Anmeldeticket aus. Das wird als *Cookie* im Web-Browser des Benutzers abgelegt und meldet ihn automatisch bei allen Systemen, die Anmeldezertifikate unterstützen, an. „Cookies sind Dateien, die von Webseiten auf dem lokalen Rechner gespeichert und beim erneuten Besuch der Seite abgerufen werden“ [77].
- **Header-Variablen:** Ein Nutzer meldet sich an einem *Web-Zugriffsverwaltungsprodukt* (WZV) eines Drittanbieters an und erhält eine authentifizierte Benutzerkennung als Teil des *HTTP-Headers* zurück. Damit kann er sich anschließend bei Anwendungen eines SAP-Anwendungsservers anmelden. HTTP-Nachrichten bestehen aus zwei Teilen, dem Kopf (Header) und dem Rumpf (Body). Der Kopf „enthält Informationen über den Nachrichtenrumpf wie etwa verwendete Kodierungen oder den Inhaltstyp, damit dieser vom Empfänger korrekt interpretiert werden kann“ [78].
- **Open Authorization (OAuth) 2.0:** Es handelt sich dabei um einen Standard zur sicheren API-Authentifizierung. So kann ein Nutzer, in der Rolle als *Resource Owner*,

einer Anwendung, dem *Client*, erlauben, in seinem Namen auf geschützte Ressourcen bzw. Daten eines Servers (*Resource Server*) zuzugreifen. Dafür muss er seine Anmeldedaten nicht bekannt geben. Dies nutzen Entwickler einer Anwendung um eine Authentifizierung zu implementieren, ohne Anmeldedaten von Nutzern zu speichern.



Stattdessen delegieren Sie die Authentifizierung der Nutzer an einen OAuth-Anbieter wie *Google*, *Facebook* oder *Twitter*. OAuth beinhaltet eine detaillierte Einschränkung der Zugriffsrechte. Dem Nutzer wird nach Eingabe der Anmeldedaten angezeigt, welche Berechtigungen der Client fordert (siehe Abbildung 13). Stimmt der Nutzer zu, wird er zurück zur Client-Anwendung geleitet. Hierbei handelt es sich nicht um eine SSO-Lösung. OAuth kann aber mit anderen Diensten gekoppelt werden, um SSO zu ermöglichen.

- **Kerberos-Authentifizierung:** Eine zentrale Schlüsselvergabe (*Key Distribution Center*, KDC) dient im Netzwerk als vertrauenswürdige dritte Partei bei der Vermittlung zwischen zwei Systemen. Sie besteht aus einem Authentifizierungsserver und einem Ticketvergabeserver. Ein Nutzer meldet sich an der SAP Cloud Platform an und möchte ein Backend-System erreichen (siehe Abbildung 14). Dafür sendet die SAP Cloud Platform eine verschlüsselte Nachricht mit der Identität des Benutzers an den SAP Cloud Connector, der die Kerberos-Bibliothek enthält.

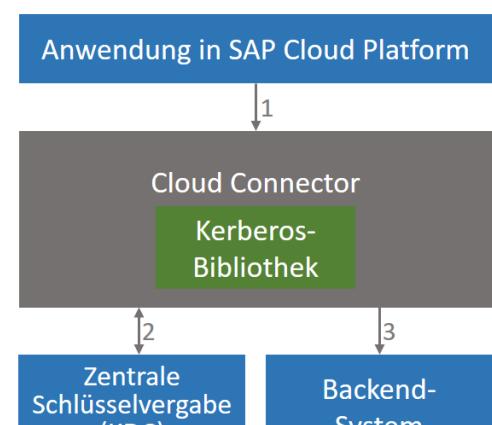


Abbildung 14: Ablauf einer Kerberos-Authentifizierung
(Quelle: [26, S. 305])

Zuerst wird die Nachricht entschlüsselt und der Nutzer bei der KDC abgefragt. Es findet eine Kommunikation zwischen dem Authentifizierungsserver und dem Ticketvergabeserver statt. Ist diese Abfrage erfolgreich, erhält der SAP Cloud Connector ein Ticket, das die Identität des Nutzers bestätigt. Dieses sendet er in einer Anfrage an

das Backend-System. Dort wird darauf vertraut, dass es sich tatsächlich um den Nutzer handelt, der im Ticket aufgeführt wird und eine Anmeldung erfolgt.

Einige der Anmeldeoptionen von SAP-Systemen können auch im Zusammenhang mit der SAP Cloud Platform verwendet werden, um eine sichere und nutzerfreundliche Authentifizierung mit SSO zu erreichen.

- **X.509 Client-Zertifikate:** Hat ein Nutzer in seinem Browser ein gültiges X.509-Zertifikat wird er für den Zugriff zu SAP-Systemen authentifiziert, ohne seine Benutzerkennung eingeben zu müssen, sofern das Zertifikat von einer vertrauenswürdigen Stelle ausgestellt wurde und die Systeme für die Verwendung von *Secure Sockets Layer* (SSL) konfiguriert sind.
- **SAML-Browser-Artefakte / SAML 2.0:** SAML ist die Abkürzung für *Security Assertion Markup Language* und bezeichnet einen offenen Standard für die Umsetzung von SSO. Dafür werden ein Dienstanbieter (*Service Provider*) und ein Identitätsanbieter (*Identity Provider*) benötigt.

Möchte sich ein Nutzer am Dienstanbieter anmelden, leitet dieser die Authentifizierungsanfrage an einen Identitätsanbieter weiter, der Identitätsinformationen von Nutzern, Rollen und Gruppen (zusammengefasst *Principals*) speichert und prüft, ob diese für den Dienst, von dem die Authentifizierungsanfrage kommt, berechtigt sind (siehe Abbildung 15). Falls der Nutzer sich in der aktuellen Sitzung bereits beim Identitätsanbieter angemeldet hat, entfällt eine weitere Abfrage der Anmeldedaten. Stattdessen erhält der Dienstanbieter eine Nachricht mit Informationen über den Nutzer und der Authentifizierungsantwort.

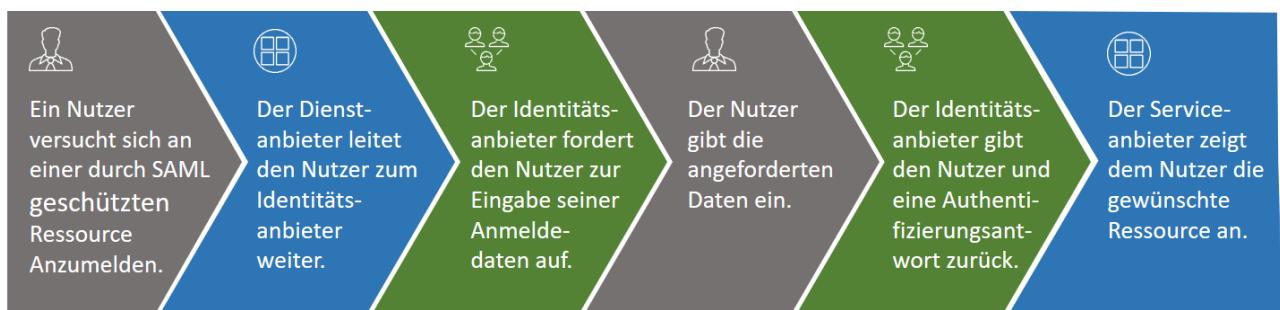


Abbildung 15: Ablauf einer SAML-Authentifizierung
(Quelle: [79])

SAP stellt auf der SAP Cloud Platform selbst einen Identitätsanbieter zur Verfügung. Dieser ermöglicht es dem Nutzer, auf alle Dienste der SAP Cloud Platform mit einmaliger Anmeldung zuzugreifen. Diese Art der Authentifizierung kann ebenfalls genutzt

werden, um einen bereits bei der Cloud Platform authentifizierten Anwender am Backend-System anzumelden, ohne nach den entsprechenden Anmeldedaten zu fragen.

ASP nutzt zur Anmeldung *Basic Authentication*. Durch Verwendung einer verschlüsselten HTTP-Verbindung (HTTPS⁹) können Benutzername und Passwort nicht mehr so leicht bei der Übertragung abgefangen werden. Jedoch wird das Passwort im Browser entweder kurzzeitig - um zu verhindern, dass der Nutzer für jede HTTP-Anfrage sein Passwort eingeben muss - oder langfristig - wenn der Nutzer diese Option im Browser wählt - gespeichert. Perspektivisch wäre es daher sinnvoll, eine andere Authentifizierungsmethode zu wählen oder die Basic Authentication um einen weiteren Faktor zu verstärken. Ein solcher Faktor kann unter anderem das Smartphone des Nutzers sein, an das ein Code für die Anmeldung gesendet wird. Die Kombination aus etwas, das der Nutzer wissen und etwas, das er haben muss, nennt sich *2-Faktor-Authentifizierung* [80]. Möglicherweise kann es sogar für den Nutzerkomfort sinnvoll sein, über die Einrichtung von SSO nachzudenken. Dies ist jedoch nicht Teil dieser Arbeit.

Werden Dritte, wie SAP, in die Dienstleistung involviert, haftet Arvato Systems Perdata dafür, dass auch diese die Datenschutzrichtlinien einhalten [28 S. 41]. Da es sich bei SAP ebenfalls um ein deutsches Unternehmen handelt, kann davon ausgegangen werden, dass SAP selbst ein großes Interesse daran hat, für eine Einhaltung des Datenschutzes zu sorgen. Die SAP Cloud Platform ist nur mit einem Konto und passendem Passwort zugänglich. Im Rahmen der SAP Cloud Platform involviert SAP jedoch wiederum Dritte und es ist zu prüfen, ob diese ebenfalls datenschutzkonform arbeiten, da ASP für alle in der Wertschöpfungskette folgenden Dienstleister haftet, falls es zu einem Datenschutzvorfall kommt.

Daher ist bei der Wahl der Region (siehe Abschnitt 2.3.2), die man in der SAP Cloud Platform benutzen möchte, Vorsicht geboten. Alle Daten der Testzugänge werden in der Region *Europe (Rot)-Trial* verwaltet und liegen demzufolge in einem SAP eigenen Rechenzentrum in St. Leon-Rot. Bei einem bezahlten Zugang zur SAP Cloud Platform ist dies abhängig von der gewählten Region. Es stellt sich die Frage, ob amerikanische Großkonzerne wie *Google*, *Amazon* und *Microsoft* ihre Dienstleistungen tatsächlich an die Anforderungen der europäischen Datenschutzverordnung angepasst haben. Klar ist, dass Sie dieser unterliegen, sobald sich die Personen, deren Daten verwaltet werden, in der EU aufhalten [81]. Um so sicher wie möglich zu sein, sind grundsätzlich die SAP-eigenen, europäischen Regionen in der SAP Cloud Platform vorzuziehen. Selbst dann ist es jedoch möglich, dass SAP

⁹ Vgl. Abkürzungsverzeichnis

die Daten zur Verarbeitung in Länder außerhalb des Europäischen Wirtschaftsraums übermittelt. Dies wird in der Datenschutzerklärung festgehalten. In einem solchen Fall werden jedoch Verträge mit den zusätzlichen Verarbeitern abgeschlossen, die dafür sorgen, dass dennoch das europäische Datenschutzniveau eingehalten wird [82].

Auch in Deutschland können die personenbezogenen Daten durch SAP im Rahmen einer Auskunftsanfrage weitergegeben werden. Dafür muss jedoch entweder die Einwilligung der betroffenen Person oder eine ausreichende rechtliche Grundlage vorliegen. In den Nutzungsbedingungen der SAP Cloud Platform verankert SAP die Einwilligung zur Weitergabe personenbezogener Daten, falls dies rechtlich notwendig ist [83].

Einen Angriffspunkt stellt auch der SAP Cloud Connector dar, da er eine verschlüsselte Verbindung zwischen dem Internet und dem Gateway bzw. dem Backend öffnet. SAP empfiehlt daher, den Zugriff auf den Cloud Connector auf die unbedingt notwendigen Personen zu beschränken und einen eigenen Server nur für den SAP Cloud Connector zu benutzen [71, S. 266]. Zudem prüft die Software den Sicherheitsstatus in Echtzeit und gibt Ratschläge zur Verbesserung.

Da die Datenschutzbestimmungen und ihre möglichen Umsetzungen in Webanwendungen äußerst komplex sind, werden die Sicherheitsmaßnahmen dieser Anwendung darauf beschränkt, den Zugang zur Anwendung und die Berechtigungen zu begrenzen sowie die Übertragungswege der personenbezogenen Daten zu sichern.

3.7 Entwicklungsumgebung

Zur Entwicklung von Fiori-Anwendungen gibt es mehrere mögliche Entwicklungsumgebungen. Die einfachste Basis dafür ist das SAPUI5-Framework, das von SAP im Internet zur Verfügung gestellt wird [48]. In der SAP Cloud Platform werden Fiori-Anwendungen in der Neo-Umgebung entwickelt. SAP empfiehlt die Nutzung der dort vorhandenen SAP Web IDE Full-Stack (siehe Abschnitt 2.3.3). Sie bietet folgende Funktionalitäten:

- **Quelltext-Bearbeitung:** Mit der SAP Web IDE Full-Stack können freie Quelltexte verfasst werden. Vervollständigung, Validierung und Vorlagen (*Templates*) werden für XML, JavaScript, Ressourcen-Dateien und SAPUI5 angeboten. XML¹⁰ ist eine Metasprache für die Definition von anwendungsspezifischen Auszeichnungssprachen [84]. JavaScript „(oft auch JS abgekürzt) ist eine leichte, interpretierte, objektorientierte Sprache [...] und ist bekannt als Skriptsprache für Webseiten“ [85]. Die Ressourcen-Dateien ermöglichen durch „Auslagerung aller sprach- oder kulturräumsspezifischer Daten“ die Übersetzung von Anwendungen [86]. Die Texte

¹⁰ Vgl. Abkürzungsverzeichnis

sind vollständig durchsuchbar und können automatisch formatiert werden (JavaScript, JSON, XML und CSS). *JavaScript Object Notation* (JSON) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen (Analysieren von Datenstrukturen) und zu generieren ist. Es basiert auf einer Untergruppe der JavaScript Programmiersprache, Standard ECMA-262 dritte Edition - Dezember 1999“ [87]. Probleme mit dem Quelltext können in einer eigenen Ansicht angezeigt werden.

- **Testen und Fehlersuche:** Anwendungen können im Browser in verschiedenen Bildschirm-Auflösungen und Sprachen ausgeführt werden. Dafür stehen in der Web IDE Testdaten zur Verfügung. Um die Anwendung auf einem mobilen Endgerät zu öffnen, kann in der Web IDE ein Quick Response (QR)-Code erzeugt werden, der dann mit dem Endgerät gescannt werden muss, um die manuelle Eingabe der Anwendungsadresse zu vermeiden.
- **Unterstützung mehrerer Geräte:** Nicht nur die darin entwickelten SAPUI5-Anwendungen, sondern auch die Web IDE selbst kann auf stationären und tragbaren Rechnern, Tablets und Mobiltelefonen ausgeführt werden.
- **Anwendungsvorlagen:** SAPUI5-Anwendungen können mit Hilfe von Vorlagen und Assistenten unter Berücksichtigung der Fiori-Richtlinien erstellt werden.
- **Graphische Oberflächen-Editoren:** SAP Web IDE bietet einen graphischen Layout-Editor und einen *UI Adaption* Editor. Ersterer dient dazu, ein Layout zu entwerfen und Daten an bestimmte Oberflächenelemente zu binden. Der Quelltext wird dabei automatisch im Hintergrund generiert. *UI Adaption* ermöglicht die Anpassung von unterstützten Fiori-Oberflächen zur Laufzeit.
- **Erweiterungen von Fiori-Anwendungen:** Von SAP zur Verfügung gestellte Fiori-Anwendungen können mit Hilfe eines visuellen Erweiterungsseditors bearbeitet werden.
- **Auslieferung von Anwendungen:** Entwicklungsprodukte können aus der Web IDE in die Cloud Platform ausgeliefert und in einem Fiori-Launchpad auf der Plattform registriert werden. Ebenso lassen sich Anwendungen auf lokale SAP-Server ausliefern.

Bevor die SAP Web IDE zur Verfügung stand, baute SAP auf die Entwicklung in einer *Eclipse IDE*. Dort lässt sich das SAPUI5-Framework mit Hilfe einer Erweiterung (*Plugin*) von SAP integrieren. Das Eclipse-Plugin wird jedoch nur noch aktualisiert um mit neuen SAPUI5-Versionen lauffähig zu bleiben. Neue Funktionen der Bibliotheken werden nicht in Eclipse verfügbar sein [63, S. 349]. Theoretisch kann das SAPUI5-Framework in der quelloffenen OpenUI5-Version in jede integrierte Entwicklungsumgebung importiert werden. Die

Entwicklung und Anpassung ist daher nicht von der SAP Cloud Platform abhängig. Holger Schäfer nennt neben Eclipse und der SAP Web IDE auch das *SAP HANA Studio in der Developer Edition* und die *WebStorm IDE* der Firma *JetBrains* [88]. Ersteres ist vor allem bei Entwicklungen für Anwendungen auf SAP-HANA-Basis sinnvoll. Letztere wird von SAP nicht durch ein Plugin unterstützt und die initiale Konfiguration ist daher aufwändiger als bei Eclipse oder SAP Web IDE. Es war daher sinnvoll, der Empfehlung von SAP zu folgen und im Rahmen dieser Arbeit die SAP Web IDE Full-Stack zu verwenden.

Bei Nutzung einer Entwicklungsumgebung in der Cloud ist eine ständige Internetverbindung zwingend nötig. Um ohne Internet weiterentwickeln zu können, bedarf es der Web IDE Personal Edition, welche lokal installiert wird. Beide Entwicklungsumgebungen werden mit Hilfe des *Git-Dienstes*¹¹ verbunden. Eine ausführliche Anleitung hierzu findet sich im Internet [89]. Der Entwickler speichert eine Quelltextversion im Git-Verzeichnis und wechselt zur lokalen Entwicklungsumgebung. Darin ruft er die Version ab und kann an der Stelle weiterarbeiten, wo er auf der SAP Cloud Platform aufgehört hat. Im Fall eines Ausfalls der SAP Cloud Platform Web IDE steht nur die zuletzt in Git gespeicherte Version zur Verfügung. Es ist daher empfehlenswert, den Quelltext so oft wie möglich in Git zu sichern.

Für die Verwendung der SAP Web IDE Cloud Version musste der Dienst in der SAP Cloud Platform geöffnet werden. Die Internetadresse konnte als Lesezeichen gespeichert und vor dort wieder aufgerufen werden. Dadurch erübrigte sich der Umweg über das SAP Cloud Platform Cockpit. Die Personal Edition der Web IDE wurde aus dem Internet [48] heruntergeladen und das Verzeichnis in den gewünschten Ordner verpackt. Im Unterordner *eclipse* fand sich eine Datei namens *orion.exe*. Ihre Öffnung startete die Laufzeitumgebung. Anschließend war die Web IDE Personal Edition über den Link <http://localhost:8080/webide/index.html> erreichbar. Beide Entwicklungsumgebungen waren sofort einsatzbereit und benötigten keine Installation von Erweiterungen.

3.8 Evaluierungskriterien

Im Rahmen der Umsetzung, soll die SAP Cloud Platform evaluiert werden. Dieser Abschnitt stellt die Fragen vor, mit deren Hilfe die SAP Cloud Platform letztendlich untersucht wird. Die Beantwortung der Fragen erfolgt in Abschnitt 4.2. Der zugehörige Fragenkatalog sieht wie folgt aus:

- Ist es möglich ein SAP ECC-System mit der Cloud Plattform zu verbinden?
- Kann darin eine Fiori-App mit Daten aus einem ECC-System entwickelt werden?

¹¹ Vgl. Abschnitt 2.3.3

- Kann die entwickelte Anwendung für Energieversorger genutzt werden, die keine Unterscheidung zwischen Einzel- und Bündelkunde benötigen?
- Ist dies ohne Änderungen des Quelltextes möglich?

Darüber hinaus ergeben sich weitere Fragen zur Evaluation der SAP Cloud Platform:

- Welchen Mehrwert stellt die SAP Cloud Platform für die Entwicklung von Fiori-Apps dar?
- Ist es denkbar, die Cloud Platform auch zum Ausliefern von Anwendungen für den Produktivbetrieb zu nutzen?

4 Evaluierung der SAP Cloud Platform

In diesem Kapitel soll die SAP Cloud Platform evaluiert werden. Dazu wird die Anwendung, die im vorherigen Abschnitt definiert wurde, umgesetzt und die einzelnen Schritte dokumentiert. Der Zweck dieser Umsetzung ist, Erfahrungen mit den einzelnen Diensten in der SAP Cloud Platform zu sammeln und deren Benutzerfreundlich- und Bedienbarkeit zu bewerten. Anschließend schätzt das Kapitel die Plattform ein. Ihr Mehrwert für unterschiedliche Zielgruppen wird bestimmt und die positiven und negativen Aspekte der genutzten Dienste erläutert.

4.1 Umsetzung der Webanwendung

Bei der Umsetzung (siehe Abbildung 16) wurden zuerst das Backend und das Gateway im SAP Cloud Connector angelegt, anschließend der OData-Service im Backend ausgeprägt (1) und über das Gateway bzw. den OData-Provisioning-Dienst exponiert (2). Der SAP Cloud Platform wurden die Systeme als *Destinations* hinzugefügt. Das sind Textdateien, welche die Verbindungsinformationen zu einem System enthalten. Anschließend wurde die Oberfläche in der Web IDE aus einer Vorlage erstellt und der veröffentlichte OData-Dienst eingebunden. Zum Schluss musste die Anwendung angepasst werden (3).

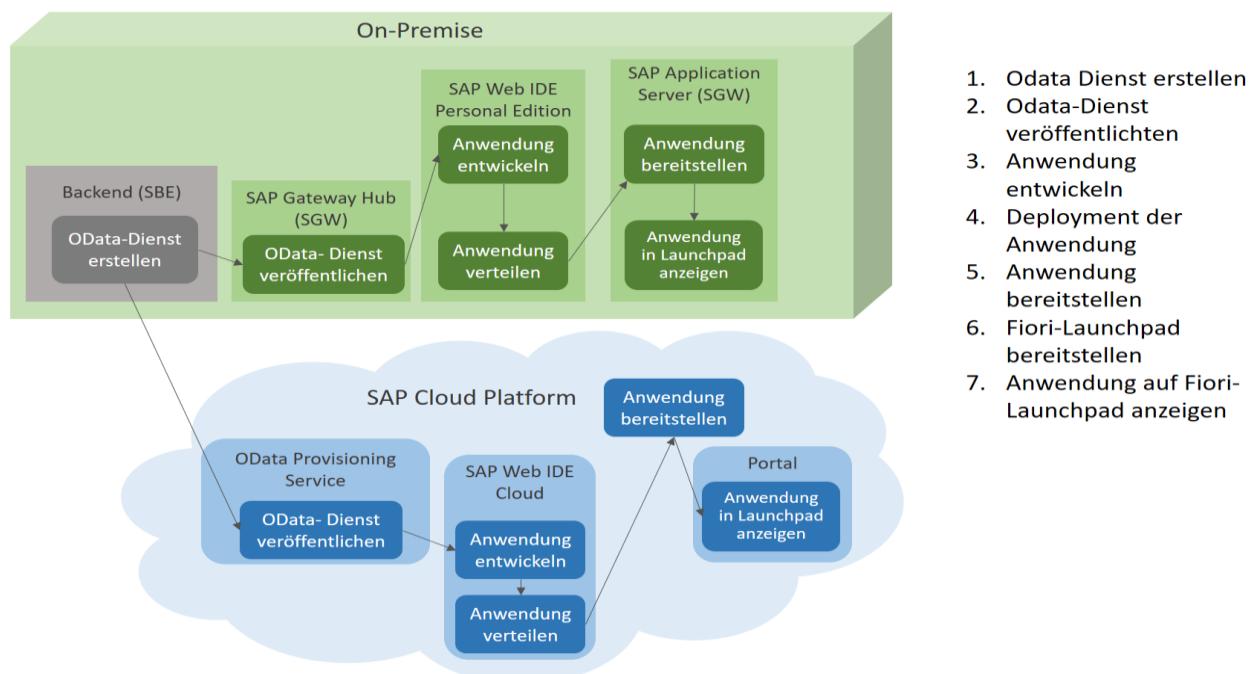


Abbildung 16: Übersicht der Umsetzung
(Quelle: eigene Darstellung)

Das Deployment der fertigen Anwendung erfolgte auf die SAP Cloud Platform sowie auf den lokalen Applikationsserver (4). Zuletzt wurde die Anwendung in einem Fiori-Launchpad in der Cloud bzw. lokal eingebunden und angezeigt (7). Bei der Dokumentation wurden nicht

alle, sondern nur besonders wichtige, bzw. Cloud-Platform-spezifische Schritte, erfasst. Ausführliche Anleitungen zu unterschiedlichen Szenarien finden sich unter: <https://blogs.sap.com/>. Für die weiteren Schritte wurde ein Konto auf der SAP Cloud Platform und ein Unterkonto in der Neo-Umgebung benötigt.

4.1.1 Verbindung zwischen SAP Backend und SAP Cloud Platform

Wie bereits in Abschnitt 2.3.5 erwähnt, erfolgt die Anbindung von SAP-Systemen an die SAP Cloud Platform mit Hilfe des Konnektor-Dienstes [90]. Dieser konnte bei den Entwicklungswerkzeugen von SAP im Internet [48] heruntergeladen werden. Für die portable Version wurde das Verzeichnis in einen gewünschten Ordner entpackt und die darin befindliche Datei „go.bat“ geöffnet. Die installierbare Version wurde mit der Dateiendung .msi ausgeliefert und öffnete einen Assistenten zur Installation. Der SAP Cloud Connector war nun unter der Internetadresse <https://localhost:8443/> mit der Anschlussnummer (*Port*) 8443 erreichbar. Für beide Versionen wurde mindestens die *Java Development Kit* (JDK) Version 7 benötigt. SAP bot dies als *Java Virtual Machine* (JVM) ebenfalls bei den Entwicklungswerkzeugen an. Bei der installierten Version konnte der Pfad zum JDK im Assistenten angegeben werden. Für die portable Version waren die Umgebungsvariablen des Computers *JAVA_HOME* und *PATH* auf das Verzeichnis des JDK zu setzen.

Um den SAP Cloud Connector gegen Ausfälle zu sichern, können zu einem Konto auf der SAP Cloud Platform zwei unabhängige Konnektoren zugewiesen werden, eine Master- und eine Shadow-Instanz. Der Master überträgt alle Konfigurationen zum Shadow, der wiederum dem Master in regelmäßigen Abständen eine Abfrage sendet und die Master-Rolle übernimmt, falls er für eine in der Dokumentation nicht näher definierte Zeitspanne keine Antwort bekommt [26, S. 320-321]. Im Rahmen dieser Arbeit wurden beide Instanzen als portable Version auf demselben Rechner installiert. Dazu musste der Port einer Instanz geändert werden. In der Praxis sollten sie auf unterschiedlichen Servern liegen, da sonst keine erhöhte Ausfallsicherheit gegeben ist.

Das Layout des Konnektors (siehe Abbildung 17) ist an die SAP Cloud Platform angepasst. In der Master-Instanz wurde zuerst der Hochverfügbarkeitsmodus (*High-Availability*) aktiviert. Anschließend wurden in der Shadow-Instanz die Verbindungsdaten zum Master eingegeben und beide miteinander verbunden. Alle weiteren Konfigurationen wurden in der Master-Instanz des Konnektors durchgeführt. Abbildungen der einzelnen Schritte befinden sich im Anhang D. Die Verbindung zu einem SAP-Cloud-Platform-Konto, in diesem Fall dem Konto aus der Neo-Umgebung, musste im SAP Cloud Connector

angelegt werden. Dafür befand sich auf der Startseite ein Knopf mit der Aufschrift Unterkonto hinzufügen (*Add Subaccount*).

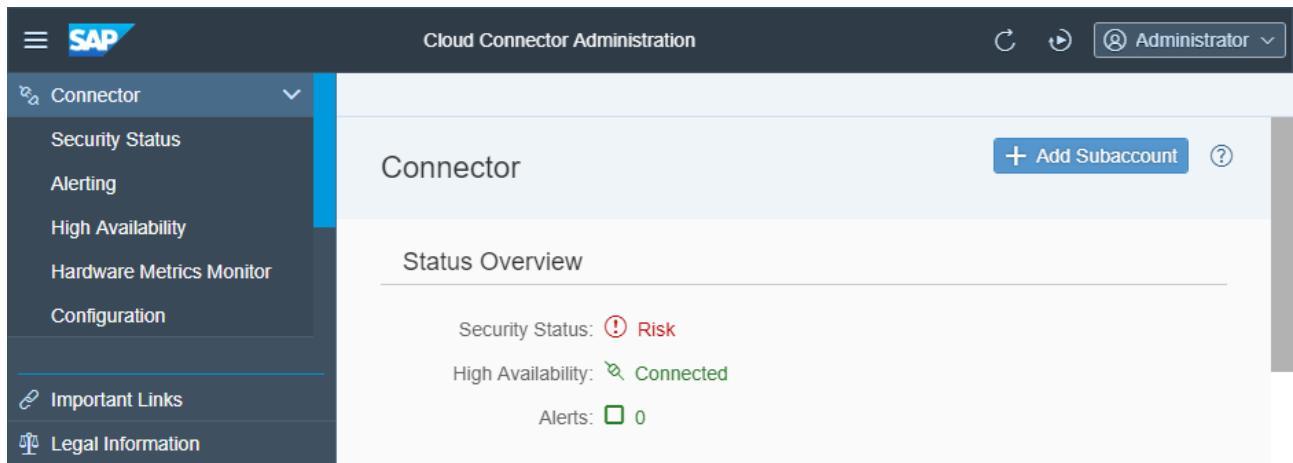


Abbildung 17: Ansicht des SAP Cloud Connector
(Quelle: eigener Screenshot)

Im Konto wurde eine *Cloud To On-Premise*-Verbindung hergestellt. Abbildungen hierzu befinden sich im Anhang E. Dies ist immer der Fall, wenn aus einer Anwendung in der SAP Cloud Platform auf Daten in einem lokalen System zugegriffen werden soll. Im umgekehrten Fall können die Daten auch in einer HANA-Datenbank in der SAP Cloud Platform liegen und von einer lokalen Applikation aufgerufen werden.

In der Registerkarte Zugangskontrolle (*Access Control*) musste nun das lokale System konkret zugeordnet werden. Einige Parameter hatten hier mehrere Auswahlmöglichkeiten. Die Dokumentationen von SAP enthielten keine vollständige Liste aller Optionen. Stattdessen sind darin detaillierte Anleitungen für die meisten Nutzungsszenarien vorhanden.

Tabelle 1 zeigt die Einstellungen für den vorliegenden Fall. Um eine ausreichende Sicherheit zu gewährleisten, wurden beide Systeme mit Hilfe von HTTPS und den entsprechenden Adressen (*Internal Host*) sowie Ports an den SAP Cloud Connector angebunden. *Virtual Host* und *Virtual Port* sind die Bezeichnungen und Ports, unter denen die Systeme in der SAP Cloud Platform erreichbar sind. Dies dient ebenfalls der Sicherheit. Die tatsächlichen Pfade bleiben für Außenstehende unsichtbar. Der Parameter *Principal Type* definiert, welcher Benutzer (*Principal*, siehe Anhang C) zur Authentifizierung bei der Einrichtung von Destinations in der SAP Cloud Platform genutzt wird [26, S. 149]. Diese Einstellung ist nur für die Nutzung von Single Sign-On relevant und erhielt daher hier den Wert *Keiner (None)*.

Parameter-name	Wert	Erläuterung
Backend Type	SAP Gateway	Typ des lokalen Systems Einstellung für das SGW-System
	ABAP System	Einstellung für das SBE-System
Protocol	HTTPS	Verbindungsprotokoll
Internal Host	xxxx.perdom.intern	Internetadresse des jeweiligen Systems.
Internal Port	8001	Anschlussnummer des lokalen Systems Port des SGW-Systems
	5200	Port des SBE-Systems
Virtual Host	sgw/sbe	Frei wählbare Kennung unter der das System aus der SAP Cloud Platform erreicht werden kann
Virtual Port	8001/5200	Anschlussnummer unter der das System aus der SAP Cloud Platform erreicht werden kann
Principal Type	None	Wenn das verbundene System keine Authentifizierung mit Zertifikaten unterstützt.
Description	Beschreibung der Verbindung	

Tabelle 1: Parameter zur Konfiguration der Verbindung im SAP Cloud Connector
(Quelle: eigene Darstellung)

Abschließend wurde im System noch eingetragen, welche Ressourcen erreichbar sein sollten. Diese Option ließ sich auf einen oder mehrere Teilstücke beschränken. Die einfachste Lösung ist die Freigabe aller Ressourcen durch Eintragung von /sap/. Dies ist jedoch keine sichere Lösung. Daher wurden die Ressourcen auf die nötigen Pfade [91] beschränkt, wie in Abbildung 18 erkennbar.

Resources Accessible On sgw:8001			
Enabled	Status	URL Path	Access Policy
<input checked="" type="checkbox"/>	<input type="checkbox"/>	/sap/bc/adt	Path and all sub-paths
<input checked="" type="checkbox"/>	<input type="checkbox"/>	/sap/opu/odata/sap/	Path and all sub-paths

Abbildung 18: Ressourcen auf dem SGW-System
(Quelle: eigener Screenshot)

Die erste Ressource führte zum Speicherort von SAPUI5-Anwendungen, die zweite zum OData-Dienst. Beim SBE-System griff der OData-Provisioning-Dienst auf die Backend-

Komponente *Information Worker - Business Enablement Provisioning* (IW-BEP) [92] zu, um die Liste der OData-Dienste zu erhalten. Der exponierte Pfad lautete daher `/sap/iwbep/`. Im SAP Cloud Platform Cockpit waren weitere Einstellungen notwendig. Im Menüpunkt Konnektivität (*Connectivity*) konnten der Status des verbundenen Konnektors und die expo-nierten Backend-Systeme überprüft werden. Im gleichen Untermenü war eine Destination für das SGW-System anzulegen. Tabelle 2 zeigt die Parameter zur Konfiguration. Beide Verbindungen wurden über das HTTPS-Protokoll an den SAP Cloud Connector gebunden. Dieser baut eine verschlüsselte Verbindung zur SAP Cloud Platform auf. Da eine doppelte Verschlüsselung unnötig ist, wird HTTPS bei der Zielkonfiguration nicht als Option angebo-tten. Die *Location ID* dient zur Unterscheidung von SAP-Cloud-Connector-Systemen, welche mit demselben Konto auf der SAP Cloud Platform verbunden wurden. Da die Konnektor-Instanzen Master- und Shadow nur als ein System gewertet wurden, war eine Angabe der Location ID in dieser Umsetzung nicht nötig.

Parametername	Wert	Beschreibung
Name	SGW-destination	Frei wählbarer Name der Des- tination
	SBE-destination	
Type	HTTP	Verbindungsprotokoll
Description		Beschreibung
Location ID		Kennnummer des Cloud Konnektors
URL	http://sgw:8001 http://sbe:5200/sap/iwbep/	Internetadresse der Destina-tion
Proxy Type	On-Premise	
Authentication	Basic Authentication	Art der Authentifizierung
Additional Properties		Zusätzliche Einstellungen
Jco.client.client	010	Für SGW
WebIDEEnabled	True	
WebIDEUsage	odata_abap,ui5_exe- cute_abap,dev_abap,dev_gen,bsp_exe- cute_abap	
Sap-client	100	Für SBE

Tabelle 2: Parameter zur Konfiguration von Zielen
(Quelle: eigene Darstellung)

Die Destination für das SBE-System wurde in der Konfiguration des OData-Provisioning-Dienstes angelegt. Zuvor musste dieser aktiviert werden. Dafür wurde im Unterkonto zu Services navigiert, dort der OData-Provisioning-Dienst gesucht und auf Einschalten (*Enable*) geklickt. Danach konnte der Dienst konfiguriert oder geöffnet werden. Die Konfiguration bot die Unterseiten *Destinations* und Rollen (*Roles*). In *Destinations* wurde das SBE-System hinterlegt. Bei *Roles* fanden sich zwei Rollen, die dem Nutzer zugeordnet werden mussten, der den OData-Dienst aus dem System mit dem Cloud-Platform-Dienst veröffentlichen will. Auch im Web-IDE-Dienst konnten Destinations angelegt werden. Anders als beim OData-Provisioning-Dienst genügte dies jedoch nicht. Die Destination musste übergeordnet im Cloud Cockpit vorliegen. Bei der Web IDE Personal Edition wurde eine Zielverknüpfung als Datei im Unterordner *Eclipse>config_master>service.destinations>destinations* angelegt. Diese enthielt ebenfalls die Parameter aus Tabelle 2.

4.1.2 Einrichten und Veröffentlichen eines OData-Dienstes

Bevor der OData-Dienst in der SAP Cloud Platform veröffentlicht und genutzt werden konnte, musste er im Backend-System erstellt werden. Dies erfolgte im *SAP Gateway Service Builder (SEGW)*. Dort konnte ein bestehendes Projekt kopiert oder ein neues Projekt angelegt werden. In Abschnitt 3.5 wurde der bereits vorhandene OData-Dienst */ASPSP/ERP_UTILITIES_UMC* erwähnt. Um diesen vor versehentlichen Änderungen zu schützen, wurde er kopiert und als *YSA_ERP_UTILITIES_UMC* in dieser Arbeit behandelt.

Im OData-Programmiermodell besteht ein OData-Dienst aus einer Modellanbieter-Klasse (*Model Provider Class*, kurz MPC) und einer Datenanbieter-Klasse (*Data Provider Class*, kurz DPC) sowie aus deren Basisklassen (siehe Abbildung 19). Die Basisklassen haben die Endung _MPC bzw. _DPC. Die MPC endet auf _MPC_EXT, die DPC auf _DPC_EXT. Die MPC enthält alle Informationen über das Datenmodell des OData-Dienstes. Die DPC beschreibt, wie der Service auf Anfragen antwortet. Sie enthält die Implementierungen der REST-Befehle. Das Datenmodell wird im SEGW konfiguriert und in die Klassen generiert. Anpassungen am Antwortverhalten des Dienstes müssen direkt in der DPC-Quelltextdatei vorgenommen werden. Dabei ist jedoch Vorsicht geboten, da DPC und MPC leicht mit den ähnlich benannten Basisklassen verwechselt werden können. Bei jeder Änderung im Datenmodell werden die Basisklassen überschrieben und manuelle Änderungen im Quelltext gehen verloren. Die DPC und die MPC bleiben jedoch unverändert erhalten.

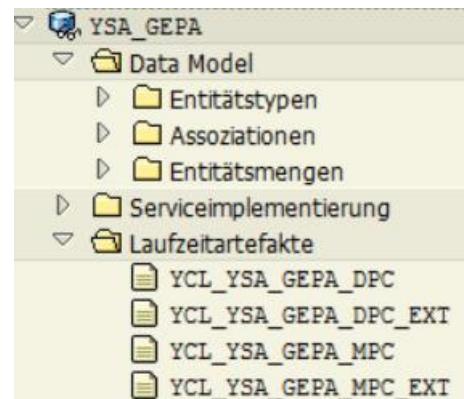


Abbildung 19: Datenmodell und Laufzeitklassen eines OData-Dienstes
(Quelle: eigener Screenshot)

Im Datenmodell (siehe Abbildung 20) finden sich zuerst so genannte Entitäten. Diese sind vergleichbar mit Instanzen einer Klasse in der Objektorientierung. Sie haben Attribute und werden von Entitätstypen abgeleitet, die selbst mit Klassen vergleichbar sind. Zwischen Entitätstypen bestehen Assoziationen, welche mit einer Kardinalität versehen sind. Die Kardinalität definiert, wie viele Entitäten von Typ A mit einer Entität von Typ B in Zusammenhang stehen.

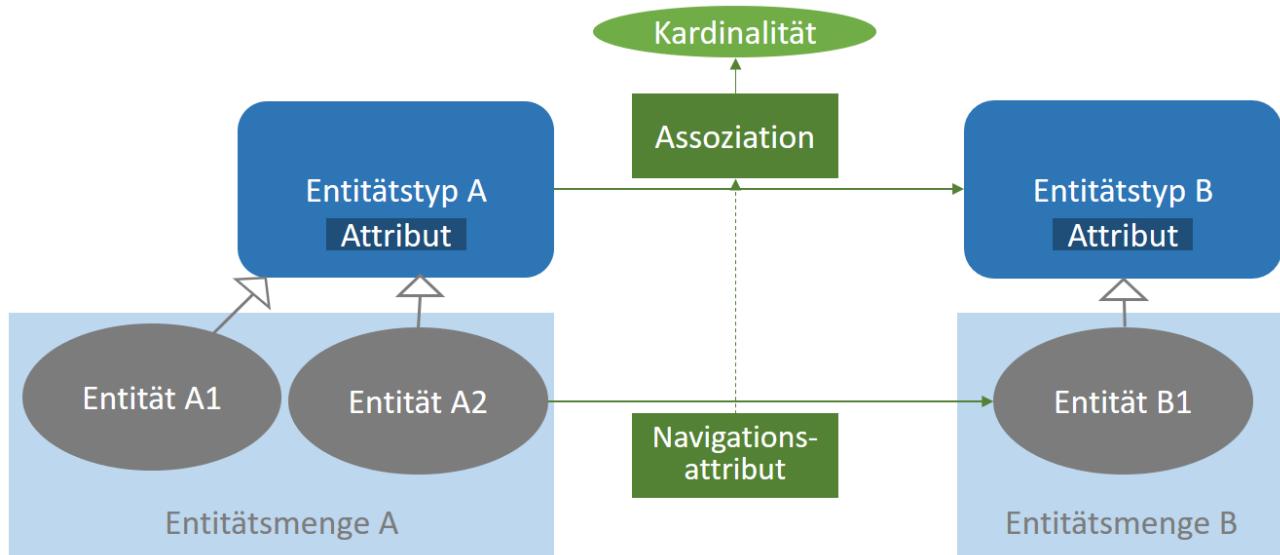


Abbildung 20: Elemente im Datenmodell eines OData-Dienstes
(Quelle: Eigene Darstellung)

Die Navigation von einer Entität zur anderen erfolgt mit Hilfe eines Navigationsattributs, das auf die im Entitätstyp definierte Assoziation verweist. Eine Entitätsmenge repräsentiert die zur Laufzeit vorhandene Menge von Entitäten. Von einem Entitätstyp können mehrere Entitätsmengen abgeleitet werden – beispielsweise kann ein Entitätstyp *Person* als Grundlage für die Entitätsmengen *MitarbeiterSet*, *AbteilungsleiterSet* und *ManagerSet* verwendet werden.

Der Dienst YSA_ERP_UTILITIES_UMC besitzt 110 Entitätstypen. Für diese Anwendung waren jedoch nur der Typ Geschäftspartner (*Account*) und Vertragskonto (*ContractAccount*) relevant. Wenn die Eigenschaften, die ein Entitätstyp haben soll, mit der Beschreibung einer Tabelle aus dem SAP-System übereinstimmen, können sie in den SEGW importiert werden. Alternativ ist eine manuelle Anlage auch möglich.

Zu jedem Entitätstyp wird auf Wunsch eine Entitätsmenge angelegt. Um zu beschreiben, welche Operationen der Konsument eines OData-Dienstes ausführen darf, hat SAP Annotations eingeführt. Entitätsmengen können unter anderem anlegbar, aktualisierbar, löschbar und adressierbar sein. Eine nicht adressierbare Entitätsmenge ist nur über Navi-

gationsattribute erreichbar. Dies gilt jedoch nur innerhalb von Clients, welche die Adressierungsregeln beachten [93]. Ruft man einen OData-Dienst in der Adresszeile des Internetbrowsers auf, wird diese Einstellung nicht berücksichtigt.

Die SAP Web IDE respektiert die Einstellungen zur Adressierbarkeit eines OData-Dienstes [94]. Die Entitätsmengen für den Geschäftspartner und das Vertragskonto waren nicht adressierbar (siehe Abbildung 21) und eine Änderung der Einstellung war nicht möglich. Daher schloss sich eine Nutzung des erwähnten Dienstes in der SAP Web IDE aus. Stattdessen wurde ein eigener Dienst mit einer kleineren Menge an Eigenschaften für die Entitätstypen definiert. Zudem wurden eigene Tabellen mit Testdaten erstellt.

Entitätsmengen						
	Name	Name des Entitätstyps	Anlegbar	Aktualisierbar	Löschbar	Adressierbar
	Accounts	Account	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	AddressTypes	AddressType				
	Attachments	Attachment				
	BankAccounts	BankAccount	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
	Banks	Bank				
	ConsumptionPeriodTypes	ConsumptionPeriodType				
	ContactActions	ContactAction			<input checked="" type="checkbox"/>	
	ContactTypes	ContactType			<input checked="" type="checkbox"/>	
	ContractAccountBalances	ContractAccountBalance				
	ContractAccounts	ContractAccount	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abbildung 21: Annotationen der Entitätsmengen *Accounts* und *ContractAccounts*
(Quelle: eigener Screenshot)

Aus den Anforderungen an die Anwendung ergab sich, dass die sie Informationen über Geschäftspartner und Vertragskonten benötigt. Im SBE-System lagen diese Informationen in den Tabellen YSA_GP (Geschäftspartner) und YSA_VKGP (Vertragskonten) in der Datenbank. Um Probleme beim Aufruf des OData-Dienstes im Frontend durch führende Nullen zu vermeiden, wurde bei der Anlage der Tabellen darauf geachtet, dass die eingetragenen Schlüsselwerte der tatsächlichen Länge des Datenfeldes entsprechen.

Der Entitätstyp für den Geschäftspartner wurde *Person* genannt. Abbildung 22 zeigt das Datenmodell des Dienstes. Die Entitätsmengen hießen *PersonSet* und *VertragskontoSet*. Die Assoziation hieß *VertragskontoZuPerson* und nannte als Kardinalität den Wert *1:m*. Das bedeutet, dass jeder Person kein, ein oder mehrere Vertragskonten zugeordnet sein können. Der Primärschlüssel einer Person war das *Partner*-Attribut. Dieses tauchte auch als Fremdschlüssel in den Vertragskonten auf.

Beide Entitätsmengen wurden als adressierbar annotiert. Die einzelnen Eigenschaften eines Entitätstyps konnten ebenfalls Annotationen erhalten, etwa um Nullwerte auszuschließen. Dies war für alle Eigenschaften sinnvoll, die Primär- oder Fremdschlüssel sind. Anschließend wurden die ausführbaren Objekte (*Laufzeitartefakte*) des Dienstes im SEGW generiert.

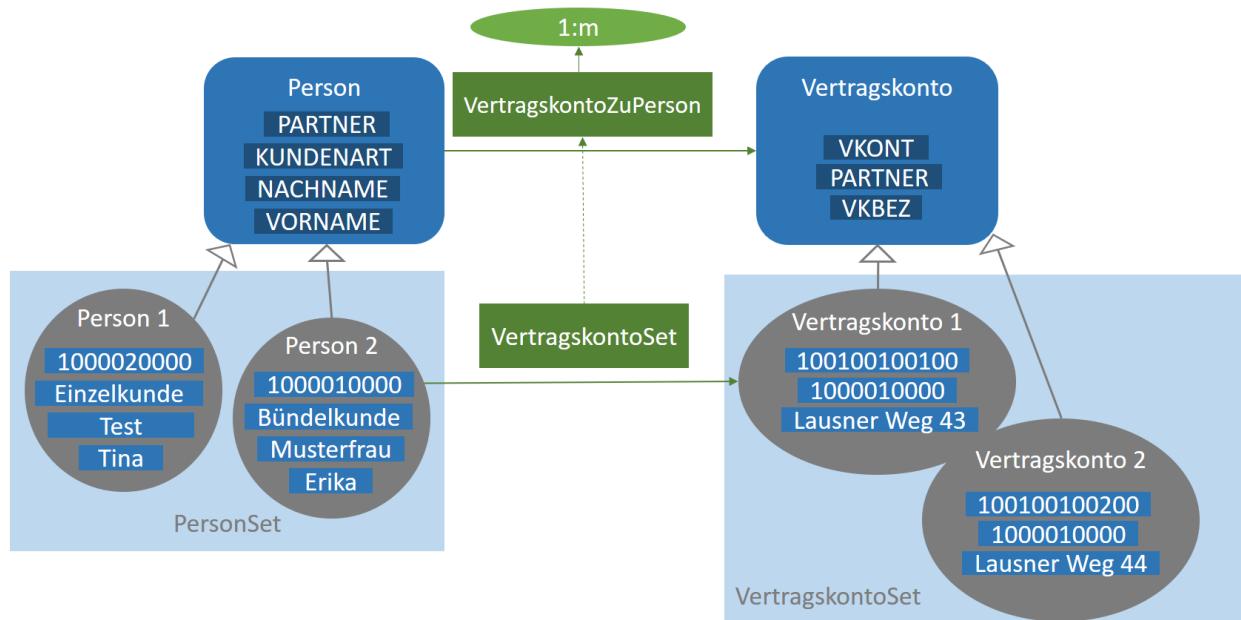


Abbildung 22: Elemente im Datenmodell des selbst definierten OData-Dienstes
 (Quelle: eigene Darstellung)

In jeder Entitätsmenge im SEGW sind automatisch die so genannten CRUD-Operationen angelegt: Erstellen (*Create*), Lesen einer Entität (*Read*, *GetEntity*), Lesen mehrerer Entitäten (*Query*, *GetEntitySet*), Ändern (*Update*) und Löschen (*Delete*). Diese Befehle lassen sich in gleicher Reihenfolge auf die bereits in Abschnitt 2.3.3 beschriebenen REST-Befehle POST, GET (für Read und Query), PUT und DELETE übertragen. Für diese Arbeit besonders relevant waren die beiden GET-Methoden, da die Daten zu den Geschäftspartnern vorerst nur angezeigt werden sollen. Folgende Abfragen wurden benötigt:

- Abfrage aller Geschäftspartner
- Filterung aller Geschäftspartner nach dem Primärschlüssel *Partner*
- Abfrage eines einzelnen Geschäftspartners
- Abfrage aller Vertragskonten mit einer bestimmten Partner-Nummer
- Abfrage eines einzelnen Vertragskontos

Diese wurden in der *ABAP Workbench* des SBE-Systems geändert. Der Quelltext befindet sich im Anhang F. Der Dienst konnte nun auf zwei Arten veröffentlicht werden: über ein Gateway oder mit Hilfe des OData-Provisioning-Dienstes der SAP Cloud Platform. Für die erste Option wurde das SBE-System mit dem SGW-System verknüpft und der Dienst dort registriert. Dadurch wurde ihm die Adresse zugewiesen, unter dem er zukünftig in einem Browser erreichbar ist:

https://servername:8001/sap/opu/odata/sap/YSA_GEPA_SRV.

Ob es sich um einen privaten oder um einen öffentlichen Dienst handelt, konnte im Gateway-System bestimmt werden. Für einen öffentlichen Dienst wird meist ein eigener Benutzer eingerichtet, der dann fest im Dienst hinterlegt wird. Bei einem privaten Dienst muss sich ein Nutzer mit seinen eigenen Benutzerdaten anmelden. Der Dienst in dieser Arbeit wurde als privater Dienst deklariert, um die durch ihn gelieferten Daten zu schützen.

Für die zweite Option zur Veröffentlichung durch den OData-Provisioning-Dienst wurde der OData-Dienst zunächst in der zugehörigen Administrationsansicht registriert (siehe Abbildung 23). In einem sich öffnenden Fenster (siehe Abbildung 24) wurde die für das SBE-System angelegte Destination ausgewählt und über die Schaltfläche mit dem Lupensymbol nach OData-Diensten auf diesem System gesucht.

Registered Services								
Actions		Service Name		Description	Version	Activation...	Default Destination	Service Document
Register	Activate/Deactivate	YSA_GEPA_SRV	SAP	Anzeigen Geschäftspartner...	1	Active	SBE-destination	Open Service Document

Abbildung 23: Hauptseite des OData-Provisioning-Dienstes
(Quelle: eigener Screenshot)

Register Service

Select Destination	SBE-destination			
Search Services	<input type="text"/>			
Available Services				
Service Name	Service Description	Version	Service Namespace	Model ID
YSA_GEPA_SRV	Anzeigen Geschäftspartner...	1		YSA_GEPA_MD...
YSA_SFLIGHT_PROJECT_...	Angelas Flugservice	1		YSA_SFIGHT_...

Abbildung 24: Fenster zur Auswahl eines Dienstes
(Quelle: eigener Screenshot)

Der passende Dienst wurde durch Klick in das quadratische Feld am Zeilenbeginn (*Markierungsfeld*) ausgewählt und aktivierte sich mit der Registrierung automatisch. Der Dienst wurde dann durch ein Gateway in der Cloud (Gateway-as-a-Service) veröffentlicht und konnte mit *Dienstdokument öffnen (Open Service Document)* angezeigt werden. Der Link lautete:

https://gwaas-s0019333218trial.hanatrial.onemand.com/odata/SAP/YSA_GEPA_SRV.

4.1.3 Erstellen und Anpassen der Oberfläche

Der OData-Dienst stand nun für Anfragen von Webanwendungen zur Verfügung. Für die Entwicklung der Oberfläche war die Art der Veröffentlichung von geringer Bedeutung, solange auf die korrekte Adresse des OData-Dienstes verwiesen wurde.

Bereits bei der Entwicklung der Anwendung spielte die Version des SAPUI5-Frameworks eine große Rolle. Das System, das die Funktion anschließend bereitstellen sollte, musste mit der Anwendung kompatibel sein. Wird die Anwendung auf die SAP Cloud Platform ausgeliefert, kann von Kompatibilität ausgegangen werden. Im SGW-System konnte die Version der SAPUI5-Bibliotheken unter folgender Adresse abgerufen werden:

https://servername:8001/sap/public/bc/ui5_ui5/index.html.

Dort zeigte sich, dass das SGW-System SAPUI5 bis zur Version 1.28 unterstützte (siehe Anhang G).

In der SAP Web IDE konnte ein Projekt von der SAP Cloud Platform bzw. von einem Applikationsserver importiert oder eine neue Anwendung erstellt werden. Projekte waren mit einer Vorlage (*Template*), aus einer Beispielanwendung oder ohne Vorlage erstellbar. Um eine möglichst Fiori-ähnliche Anwendung zu erreichen, bot sich die Nutzung einer Vorlage an. Dabei musste auf die SAPUI5-Version geachtet werden. Die Spanne der möglichen Werte reichte von Version 1.54 bis zur Version 1.38 zurück. Keine der vorhandenen Vorlagen waren daher mit der Bibliothek des lokalen Applikationsservers kompatibel. In den Einstellungen eines Projekts war es zwar möglich, die SAPUI5-Version zu verringern. Dies führte jedoch zu einer Fehlermeldung (siehe Abbildung 25).

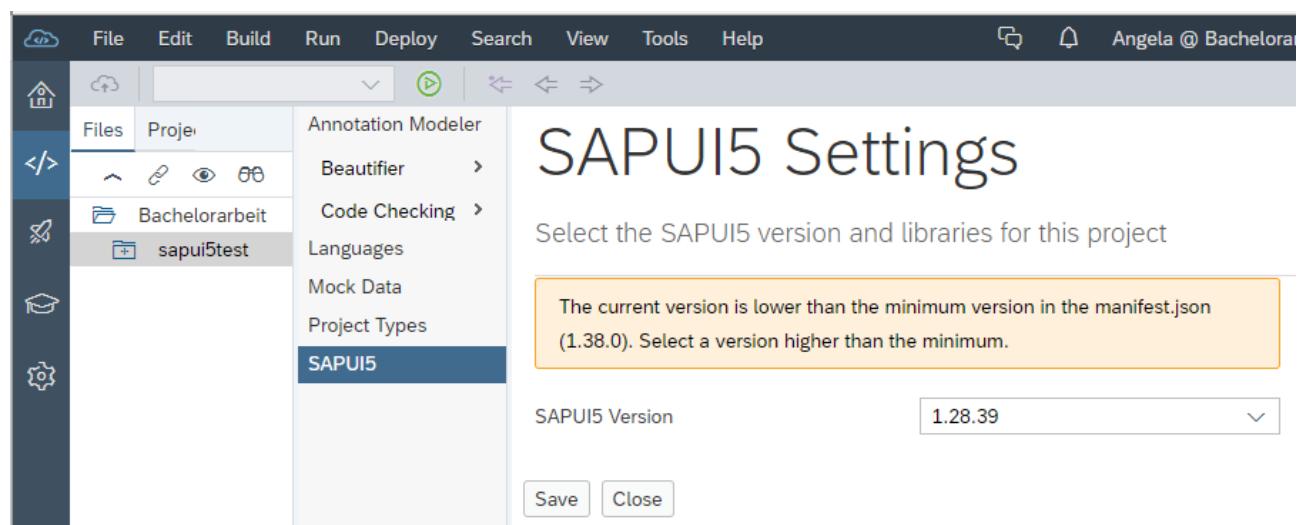


Abbildung 25: Konfiguration der SAPUI5-Version eines Projekts
(Quelle: eigener Screenshot)

Anders stellte sich der Sachverhalt in der SAP Web IDE Personal Edition dar. Dort konnten Versionen zwischen 1.44 und 1.28 gewählt werden. Die Entwicklung einer mit dem SGW-System kompatiblen Anwendung war damit möglich. Eine zweite Version der Anwendung

wurde mit der neuesten Version des SAPUI5-Frameworks in der Cloudversion der SAP Web IDE entwickelt und dokumentiert, um den aktuellen Stand der Fiori-Oberflächen zu zeigen. In der Cloudversion der SAP Web IDE standen für die SAPUI5-Version 1.54 vierzehn Vorlagen in fünf Kategorien zur Verfügung. Die Kategorie *Fiori Elements* enthielt Teilobjekte, welche die Einhaltung der Fiori-Gestaltungsrichtlinien erleichtern sollten. Es war möglich, eigene Vorlagen zu entwickeln und diese einigen oder allen Nutzern der SAP Web IDE zur Verfügung zu stellen.

Vier der gegebenen Vorlagen erzeugen mit wenigen Konfigurationsschritten grundlegende Fiori-Anwendungen (siehe Abbildung 26):

- **SAP Fiori Master-Detail Application:** Die Anzeige wird in den Master- und den Detailbereich unterteilt. Der Masterbereich zeigt eine Liste von Objekten an, von denen ein ausgewähltes im Detailbereich darstellbar ist.
- **CRUD Master-Detail Application:** Diese Vorlage bietet zusätzlich zu den Funktionen der Master-Detail-Anwendung die Möglichkeit, Objekte als Datensätze anzulegen, zu bearbeiten und zu löschen.

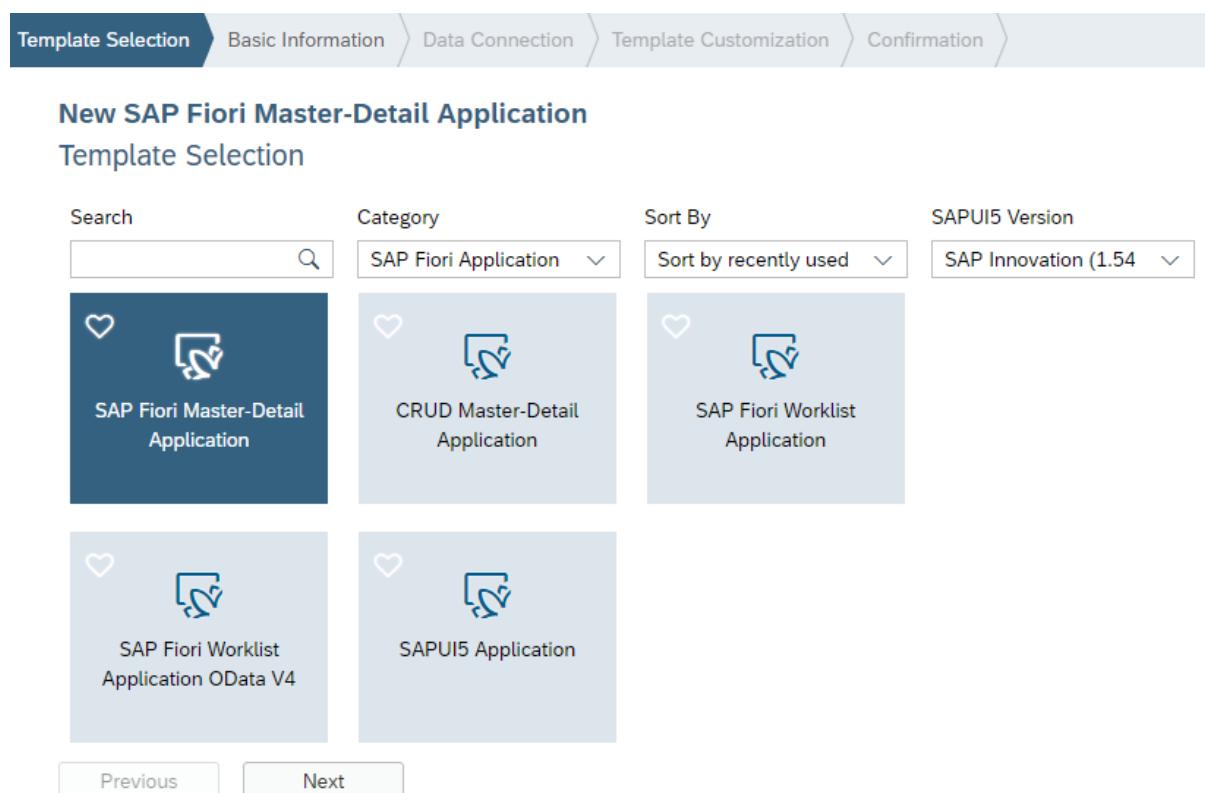


Abbildung 26: Anzeige zur Auswahl einer Projektvorlage
(Quelle: eigener Screenshot)

- **SAP Fiori Worklist Application:** Eine tabellenartige Liste von Objekten wird mit ausgewählten Eigenschaften angezeigt.

- **SAP Fiori Worklist Application OData V4:** Diese Anwendung benötigt einen OData-Dienst ab Version 4 um eine Objektliste anzuzeigen.

In der Kategorie *SAP Fiori Application* wurde zudem eine Vorlage für SAPUI5-Anwendungen angeboten. Diese erstellte ein Projekt mit der grundlegenden Struktur einer SAPUI5-Anwendung und einer optionalen Ansicht. Weitere Ansichten konnten manuell in der Anwendung erstellt werden.

Entwicklungsumgebung	SAP Web IDE in der SAP Cloud Platform	SAP Web IDE Personal Edition
Pfad	File>New>Project from Template	
SAPUI5-Version	1.54	1.28
Gewählte Vorlage	CRUD Master-Detail Application	SAP Fiori Master Master-Detail Application
Grundlegende Informationen	Projektname: bp_crud Titel: Partner anzeigen Namensraum: angela	Projektname: bp_mmd
Datenverbindung	Dienst-URL>GWaaS Link: /odata/SAP/YS_A_GEPA_SRV Bzw. /odata/SAP/YS_A_GEPA_ OHNE_BUENDELKUNDE_SRV	Dienstkatalog>SGW-destination
Detaillierte Anpassung der Vorlage	Siehe Abbildung im Anhang H.1	Siehe Abbildung im Anhang H.2
Ausführen	Run>Run as>Web Application flpSandbox.html	Keine weitere Auswahl nötig

Tabelle 3: Anlegen einer Anwendung mit Hilfe einer Vorlage
(Quelle: eigene Darstellung)

Zu den in Abschnitt 3.2 definierten Anforderungen an die Anwendung passte die Vorlage für eine *CRUD-Master-Detail*-Anwendung. Die Master-Detail-Ansicht war optisch ansprechender als eine Worklist und die erweiterten Optionen im Vergleich zur reinen Master-Detail-Anwendung vereinfachten die zukünftige Umsetzung der Bearbeitung von Stammdaten. In der lokalen Web IDE mit niedrigerer SAPUI5-Version stand diese Option nicht zur Verfügung und es wurde stattdessen eine *Master-Master-Detail*-Anwendung umgesetzt. Diese unterschied sich von der modernen Master-Detail-Anwendung dadurch, dass bei Auswahl eines Objekts aus der Masteransicht, eine weitere Masterliste mit den Unterobjekten erschien.

Der Detailbereich zeigte dann die Details des Unterobjekts an. Tabelle 3 enthält eine Übersicht der Einzelschritte zur Erzeugung der Anwendung.

In beiden Entwicklungsumgebungen öffnete sich bei der ersten Benutzung ein leerer Arbeitsbereich (*Workspace*). Der Menüpfad *File>New>Project from Template* führte zu dem in Abbildung 26 gezeigten Assistenten. Dieser verlange die Entscheidung für eine Vorlage, die Eingabe von Projektinformationen und die Verknüpfung von Felder mit den passenden Eigenschaften des zuvor gewählten OData-Dienstes. Bei der Anwendung in der SAP Cloud Platform stammte dieser aus dem SBE- und in der Personal Edition aus dem SGW-System. Diese Zuordnung war willkürlich.

Die Anwendungen waren nach abgeschlossener Konfiguration der Vorlage als Webanwendung ausführbar und zeigten die Daten aus dem OData-Dienst an. Da der Projekttyp mit *SAP Fiori* spezifiziert wurde, konnte die Anwendung auch in einer *SAP Fiori Launchpad Sandbox* getestet werden. Wurde die Ausführungsoption *With Frame* markiert, konnte zur Laufzeit zwischen allen Sprachen der Anwendungen und zwischen verschiedenen Auflösungen und Geräten gewechselt werden. Dies gab dem Entwickler die Sicherheit, dass die Anwendung tatsächlich geräteunabhängig ist. Zur Anzeige auf einem Smartphone konnte in der Vorschau ein QR-Code angezeigt werden, den das Telefon dann scannte, um dem Entwickler die Eingabe langer Internetadressen zu ersparen. Für die Anwendungen konnten Tests des Quelltextes und der Oberfläche durchgeführt werden. Einige Testszenarien waren bereits vorkonfiguriert.

Die Quelltextdateien waren nach dem Softwarearchitekturmuster MVC (siehe Abschnitt 3.3) strukturiert. Der Ordner *webapp* enthielt unter anderem die Unterordner *model*, *view* und *controller*. Die View-Dateien liegen im XML-Format vor, bei den Controllern sind es JavaScript-Dateien. An beiden Oberflächen waren noch kleine Änderungen im Quelltext nötig, um die Anwendung auslieferbereit zu machen. Die Anzeige der Kundenart erfolgte in der CRUD-Anwendung im Detailbereich (siehe Abbildung 27). Dort befand sich ein aus zwei Piktogrammen bestehendes Menü mit einem Informations- und einem Dokumentenreiter. Die Kundende-

Details for Business
Partner 1000010000



First Name:
Erika

Last Name:
Musterfrau

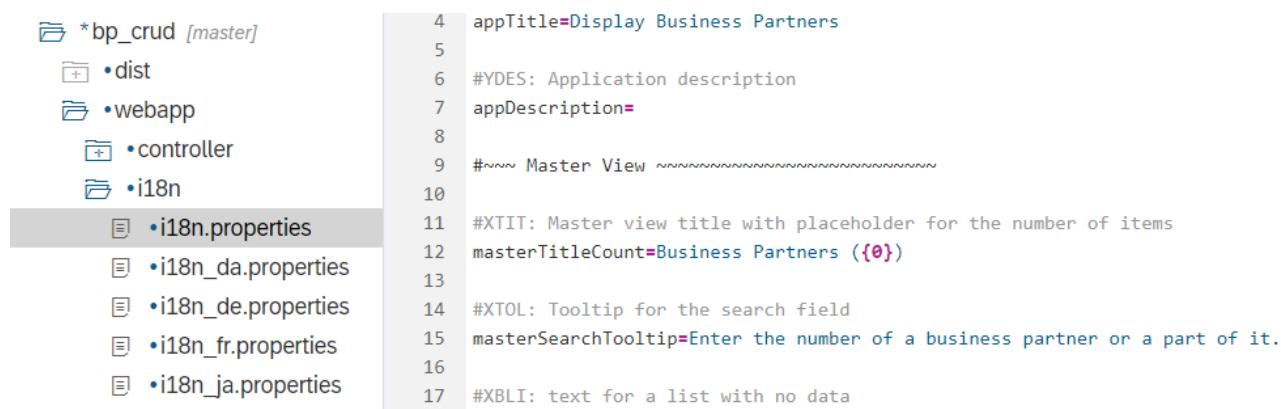
Partner Type:
BÜNDELKUNDE

Abbildung 27: Anzeige der Kundenart in der CRUD-Anwendung
(Quelle: eigener Screenshot)

tails Vorname, Name und Kundenart wurden auf dem Informationsreiter in Form von Textfeldern mit Beschriftungsfeld (*Label*) hinterlegt. Dem Textfeld wurde der Wert der Eigenschaft in der View-Datei dynamisch mit der Anweisung `text={<Eigenschaft>}`, zum Beispiel `text={Kundenart}`, zugewiesen.

In der CRUD-Anwendung mussten außerdem einige Texte angepasst werden. Die Überschrift des Masterbereichs und die der Vertragskontoliste im Detailbereich enthielten die Namen der Entitätsmengen (PersonSet anstatt von Geschäftspartner und Vertragskontoset anstatt von Vertragskonten, siehe Anhang I.1). Die Beschriftungstexte der Partnerdetails waren für die Übersetzung vorzubereiten. In der Master-Master-Detail-Anwendung waren die Bezeichnungen der Kontodetails (ebenfalls Label) noch nicht verständlich genug für Endkunden.

Damit Oberflächen von Anwendungen schnell in mehreren Sprachen angezeigt werden konnten, bot die Fiori-Vorlage eine Ressourcen-Datei namens „i18n.properties“ (siehe Abbildung 28). Die Abkürzung i18n steht für Internationalisierung (*Internationalisation*). Die Datei enthielt Schlüssel-Wert-Paare. In der Oberfläche wurde an Stelle des Textes das Schlüsselwort eingetragen und die Anwendung ermittelte zur Laufzeit die anzugezeigte Sprache und die entsprechenden tatsächlichen Texte. Die verfügbaren Sprachen mussten dazu auch in den Projekteinstellungen berücksichtigt werden. Pro Sprache musste eine Ressourcendatei vorliegen.



```
4 appTitle=Display Business Partners
5
6 #YDES: Application description
7 appDescription=
8
9 #~~~ Master View ~~~~~
10
11 #XTIT: Master view title with placeholder for the number of items
12 masterTitleCount=Business Partners ({0})
13
14 #XTOL: Tooltip for the search field
15 masterSearchTooltip=Enter the number of a business partner or a part of it.
16
17 #XBLI: text for a list with no data
```

Abbildung 28: Ausschnitt aus der englischsprachigen Ressourcen-Datei
(Quelle: eigener Screenshot)

Bei der Anpassung der Vorlage im Assistenten wurden bereits einige Übersetzungen gepflegt, andere waren durch SAP vorbelegt. Es war daher sinnvoll die gegebene Datei als englische Ressourcendatei zu verstehen und weitere Texte darin ebenfalls auf Englisch zu pflegen. Die Überschrift des Masterbereichs war mit dem Schlüssel `masterTitleCount` und dem Wert `PersonSet ({0})` hinterlegt. Die Angabe in Klammern steht für einen Platzhalter. Mithilfe einer JavaScript-Funktion wird dem Text bei Anzeige der Anwendung die tatsächliche Nummer der Geschäftspartner hinzugefügt. Dieser Wert wurde zu *Business Partners*

{0} geändert und alle anderen Texte der Datei waren zu überprüfen und ggf. anzupassen. Für die Beschriftungstexte der Geschäftspartnerdetails wurden drei weitere Schlüsselwörter benötigt. Diese waren frei wählbar, solange sie im Quelltext der anzeigenenden Oberfläche vorkommen. Um die Les- und Wartbarkeit der Anwendung durch andere Entwickler zu gewährleisten, boten sich sprechende Titel an, wie z.B. *partnerNameLast* für die Beschriftung des Nachnamens. Nach diesen Änderungen wurde die Anwendung in englischer Sprache wie gewünscht angezeigt.

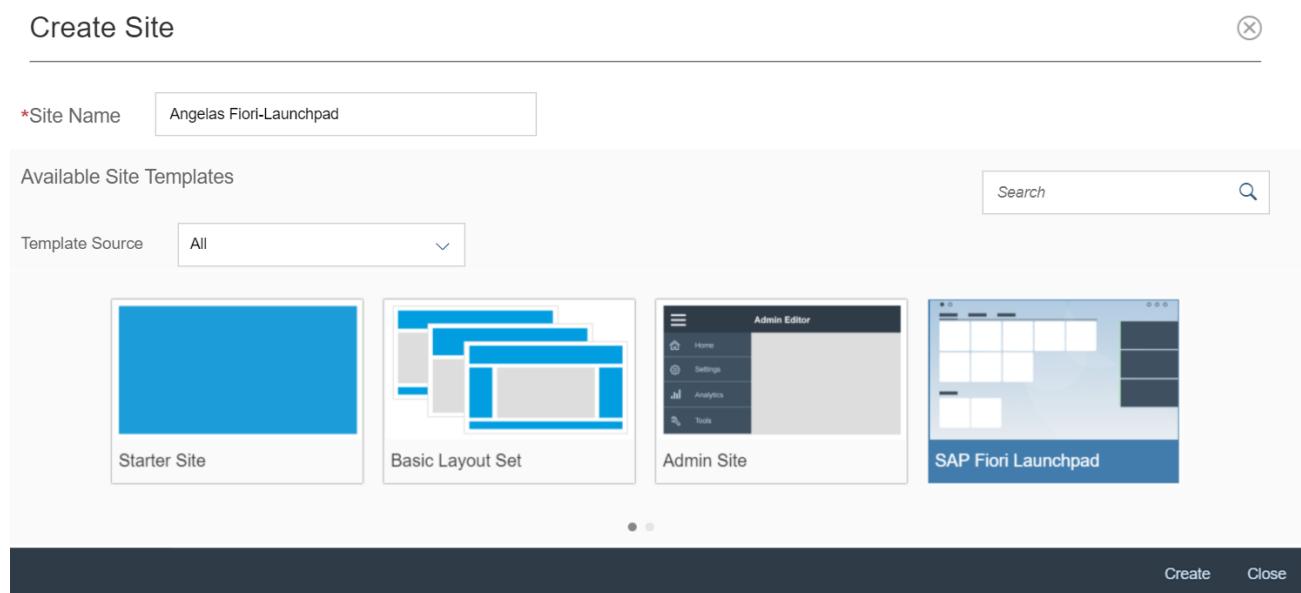
Die Übersetzung der Oberflächen in andere Sprachen erfolgte in der SAP Cloud Platform komfortabel aus der SAP Web IDE. Dazu musste der Dienst *SAP Translation Hub* im SAP Cloud Platform Cockpit aktiviert und das Projekt in einem Git-Verzeichnis gespeichert werden. Im Kontextmenü der Ressourcendatei fand sich die Option *Übersetzungsdateien erzeugen (Generate Translation Files)*. Diese erstellte ein Projekt im SAP Translation Hub und startete die Generierung der Übersetzungsdateien. Nach Ende des Prozesses erschien ein Hinweis, dass die Dateien erstellt wurden und im Git-Verzeichnis abgefragt werden konnten. Die Benennung der Sprachdateien erfolgte durch Erweiterung des Namens der ausgehenden Datei - aus *i18n.properties* wurde etwa *i18n_de.properties*. Die Ursprungsdatei behielt ihren Namen.

Der Übersetzungsdiensst stand in der SAP Web IDE Personal Edition nicht zur Verfügung. Da beide Entwicklungsumgebungen aber durch ein Git-Verzeichnis verknüpft waren (siehe Abschnitt 3.7), konnte das Projekt zur Übersetzung in die SAP Cloud Platform importiert und anschließend im SAP Translation Hub übersetzt werden. Nachdem alle Texte überprüft und gegebenenfalls angepasst worden waren, konnte die Anwendung ausgeliefert werden.

4.1.4 Deployment der Anwendung

Wie in Abschnitt 3.5 beschrieben, erfolgte das Deployment der Anwendung auf zwei verschiedene Arten: über die SAP Cloud Platform in ein Fiori-Launchpad in der Cloud und über einen lokalen Applikationsserver in ein lokales Fiori-Launchpad. Das Cloud-Deployment wurde mit Hilfe der Cloud-Version der SAP Web IDE gezeigt, die zweite Variante mit der lokalen Version der Web IDE. Während die Cloudversion beide Optionen abdecken konnte, war der Deployment-Prozess einer Anwendung aus der lokalen Entwicklungsumgebung in die Cloud nicht direkt möglich. Stattdessen war es notwendig, das mit der Cloud verknüpfte Git-Verzeichnis zu aktualisieren und die Auslieferung dann von dort aus durchzuführen. Im Kontextmenü des Projekts in der Web IDE wählte der Nutzer *Deploy>Deploy to SAP Cloud Platform*. Ein sich öffnendes Fenster (siehe Anhang J) erlaubte die Auslieferung einer neuen oder die Aktualisierung einer bereits bestehenden Anwendung. Das Konto und der

Projektname wurden aus der Web IDE, der Anwendungsname sowie die Versionsnummer aus der manuellen Eingabe übernommen. Mit Klick auf *Ausliefern* („Deploy“) startete der Prozess. Nach Abschluss erschien eine Nachricht mit der Option, die Anwendung auf einem Fiori-Launchpad zu registrieren. Hierfür musste zuerst in der SAP Cloud Platform der Portal-Dienst geöffnet und dort ein Fiori-Launchpad eingerichtet werden. Auf der Startseite des Portal-Dienstes erschien die Schaltfläche, mit der eine neue Seite erstellt werden konnte. Dieser wurde ein Name (*Angelas Fiori-Launchpad*) und eine Vorlage (*SAP Fiori Launchpad*) zugewiesen (siehe Abbildung 29).



**Abbildung 29: Ansicht zum Erstellen einer Seite im Portal-Dienst
(Quelle: eigener Screenshot)**

Im Anschluss musste im Fiori-Launchpad ein Anwendungskatalog erstellt werden. Dieser erhielt einen Namen und eine Beschreibung. Anschließend galt es, mindestens noch eine Rolle zuzuordnen, damit der Benutzer die Berechtigung erhielt, Apps aus dem Katalog zu sehen. Anwendungen bestanden in diesem Kontext aus Kacheln (der Oberfläche, die als Verknüpfung zur Anwendung diente) und den Zielzuordnungen (beinhalteten den Pfad).

Anschließend wurde eine Kachelgruppe angelegt. Diese diente der übersichtlichen Organisation von Anwendungsverknüpfungen auf der Startseite des Nutzers. Sie kann Kacheln von Anwendungen aus verschiedenen Katalogen enthalten und musste ebenfalls mindestens eine Rolle zugeordnet bekommen [95]. Abbildung 30 veranschaulicht den Zusammenhang zwischen Nutzer, Rolle, Katalog, Gruppe und der Anwendung. Abbildungen zu den Einzelschritten der Umsetzung finden sich im Anhang K.



Abbildung 30: Zusammenspiel verschiedener Objekte im Fiori-Launchpad
(Quelle: [96])

Sobald die Anwendung in der SAP Cloud Platform veröffentlicht war, konnte sie entweder über die SAP Web IDE oder den Portal-Dienst auf einem Launchpad registriert werden. Dabei mussten ein Name und eine Zielverknüpfung für das Objekt eingegeben werden. Anschließend wurde eine Kachel zur Navigation erstellt und die Anwendung einem Katalog und einer Gruppe zugeordnet.

Anhang L dokumentiert die nötigen Schritte in der SAP Web IDE. Bei der Anpassung der Kachel trat ein Fehler im Assistenten auf. Die Schaltfläche für die Navigation nach vorne war nicht auswählbar und musste durch manuelles Ändern des Kacheltyps aktiviert werden. Sobald alles konfiguriert war, konnte das Launchpad im Portal-Dienst der SAP Cloud Platform veröffentlicht werden. Dies war an dieser Stelle zu Testzwecken auch zurücknehmbar.

Die Auslieferung der Anwendung auf den lokalen Applikationsserver des SGW-Systems erfolgte aus der SAP Web IDE Personal Edition. Dort führte der Weg im Kontextmenü über *Deploy>Deploy to SAPUI5 ABAP Repository*. In dem sich öffnenden Fenster (siehe Anhang M) entschied sich der Nutzer zuerst für ein System. Dieses musste als Destination in der lokalen Web IDE vorliegen und der Pfad `/sap/bc/adt` im SAP Cloud Connector freigegeben sein [97]. Der Assistent bestätigte, dass die Anwendung und der Applikationsserver in Bezug auf die SAPUI5-Version kompatibel waren. Mit einer Auswahl

legte der Nutzer fest, ob die Anwendung neu erstellt oder eine bestehende aktualisiert werden soll. Nach erfolgreicher Auslieferung erschien ein Hinweis zur Bestätigung. Im Applikationsserver konnte die Anwendung nun aktiviert und als alleinstehende Applikation (engl. *Stand-Alone* [98]) getestet werden.

Das SGW-System enthielt alle nötigen Komponenten, um Fiori-Launchpads zu erstellen und anzuzeigen. Zuerst erstellte der Administrator ein neues Launchpad und fügte die zuvor ausgelieferte Anwendung hinzu. Anschließend wurde ein so genanntes semantisches Objekt erstellt, das auf die Anwendung verwies. Im Launchpad-Designer, einer Webanwendung zur Konfiguration, wurden anschließend analog zur Cloud-Version der Katalog und die Rolle erstellt. Die Anwendung musste manuell zu beiden hinzugefügt werden. Abschließend musste im SGW-System noch eine Rolle definiert und dem Katalog, der Gruppe und dem Nutzer zugeordnet werden. Abbildungen der einzelnen Teilschritte finden sich im Anhang N. Es kam vor, dass sich das Fiori Launchpad bei Änderungen nicht aktualisierte. In diesem Fall mussten die Zwischenspeicher (*Cache*) im Applikationsserver und dem Internetbrowser gelöscht werden [99]. Um das Launchpad in mehreren Sprachen zu testen, konnte der URL-Parameter *sap-language=<zweistelliges Länderkürzel>* verwendet werden. Für die deutsche Anzeige ergab sich somit *sap-language=de*.

4.1.5 Erreichbarkeit der Anwendung für Nutzer

Aufgrund der Vielzahl an Rollen, die Mitarbeiter in einem Unternehmen einnehmen können, bot das Fiori-Launchpad die Möglichkeit, verschiedene Rollen für u.a. Sachbearbeiter oder SAP-Berater zu hinterlegen. Damit konnte eine Steuerung der Ansicht bezüglich der Aufgabenschwerpunkte erreicht werden.

Das lokale Fiori-Launchpad, das auf dem SGW-System veröffentlicht wurde, bezog seine Rollen und Benutzerdaten aus dem Applikationsserver. Ein berechtigter Nutzer erreichte die Startseite über eine feste Zielverknüpfung: <https://servername:8001/fiori>. Dort konnte er sich mit seinen SGW-Benutzerdaten anmelden und sah die ihm zugeordneten Anwendungsgruppen.

Alle Anwendungen der SAP Cloud Platform arbeiten mit einem so genannten Identitätsanbieter, dem SAP ID Service. Dieser verwaltet Nutzer und Rollen für SAP-Anwendungen und SAP-Internetseiten. Um die Anwendung im Portal auf der SAP Cloud Platform inklusive der integrierten Daten ansehen zu können, benötigt der Nutzer die passende Berechtigungsrolle für den Portal- und den OData-Provisioning-Dienst. Wie bereits in Abschnitt 4.1.4 erwähnt, müssen dieser Rolle sowohl der Katalog als auch die Gruppe zugeordnet werden, welche die Anwendung enthalten.

Beispielhaft sollte der Nutzer mit der ID *P2000134653* Zugriff zur entwickelten Anwendung im veröffentlichten Fiori-Launchpad erhalten. Für den OData-Dienst erhielt er die vordefinierte Rolle *GW_User* (siehe Abbildung 31). Im Portal-Dienst bot sich die Rolle *Portal_User* an. Um die Zuweisungen nicht einzeln in der Konfiguration des Dienstes vorzunehmen, konnte dies auch im SAP Cloud Platform Cockpit unter dem Menüpunkt *Security>Authorizations* passieren. An Stelle der vordefinierten Rollen konnten auch selbst definierte Rollen genutzt werden. Diese mussten in der Konfiguration der einzelnen Dienste erstellt werden.

Name	Type	Actions
GW_Admin	Predefined	
GW_User	Predefined	
....		

GW_User Predefined: Provisioned by the application

Individual Users			Assign	Unassign All	Groups			Assign	Unassign All
User ID	Actions				Group:	Actions			
p2000134653			Unassign						

Abbildung 31: Rollenvergabe in der SAP Cloud Platform
(Quelle: eigener Screenshot)

4.1.6 Verwendung der Anwendung mit einem Standard-OData-Dienst

Die in den vorherigen Abschnitten entwickelte Anwendung war in der Lage, eine vom Standard abweichende Datengrundlage in einem Webbrowser anzuzeigen. Weitergehend musste geprüft werden, ob dieselbe Oberfläche dazu geeignet war, auch die Standardversion der Datengrundlage anzuzeigen und wie groß der Änderungsaufwand ist. Dafür wurde ein zweiter OData-Dienst erstellt, bei dem das selbstentwickelte Datenfeld *Bündelkunde* nicht vorkam. In allen anderen Punkten fanden keine Änderungen statt. Der Dienst wurde über den OData-Provisioning-Dienst veröffentlicht.

In der SAP Web IDE gab es mehrere Möglichkeiten. Die Anwendung konnte mit dem geänderten Dienst neu erstellt werden. Dann mussten anschließend alle Dateien aus dem

bestehenden Projekt, die von der Vorlage abweichen, kopiert werden. Das waren die *Detail.view.xml* und die Ressourcendateien zur Übersetzung. Dabei musste auf die Anpassung aller Erwähnungen des Projektnamens im Quelltext geachtet werden.

Die zweite Möglichkeit war die Erweiterung (*Extension*) des Projekts. Diese konnte erst nach dem Deployment einer Anwendung vorgenommen werden. In der SAP Web IDE öffnete sich unter dem Pfad *File>New>Extension Project* ein Fenster (siehe Abbildung 32), in dem eine Anwendungsquelle gewählt wurde. Zur Auswahl standen die SAP Cloud Platform oder ein SAPUI5 ABAP Repository auf einem Applikationsserver. In der Quelle fiel die Wahl auf die gewünschte Anwendung und ein Name wurde vergeben. Im Arbeitsbereich lag nun ein neues Projekt vor. Dieses bot nur wenige Quelltextdateien, wie die Dateien *index.html*, *manifest.json*, *neo-app.json* und *Component.js*.

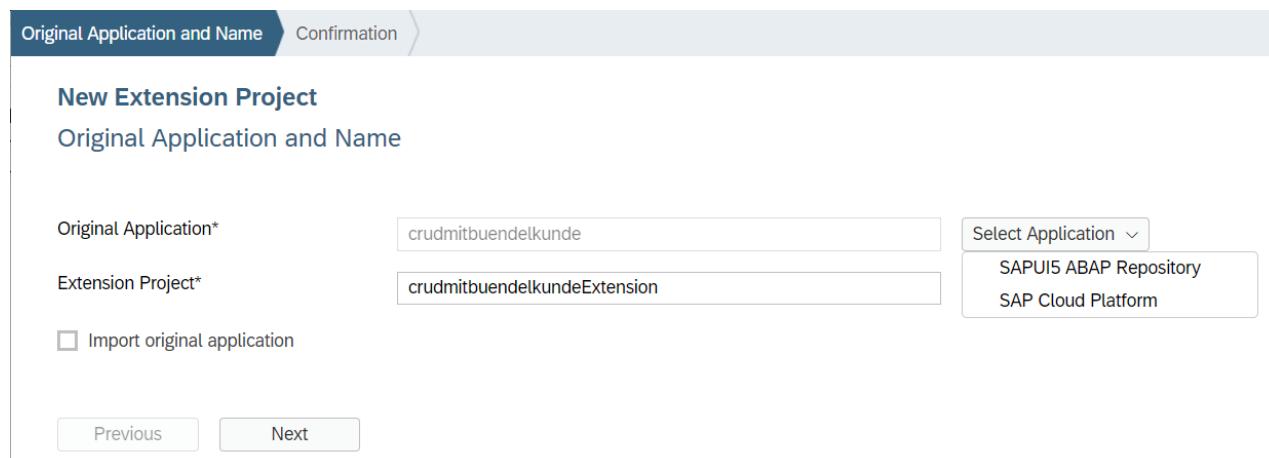


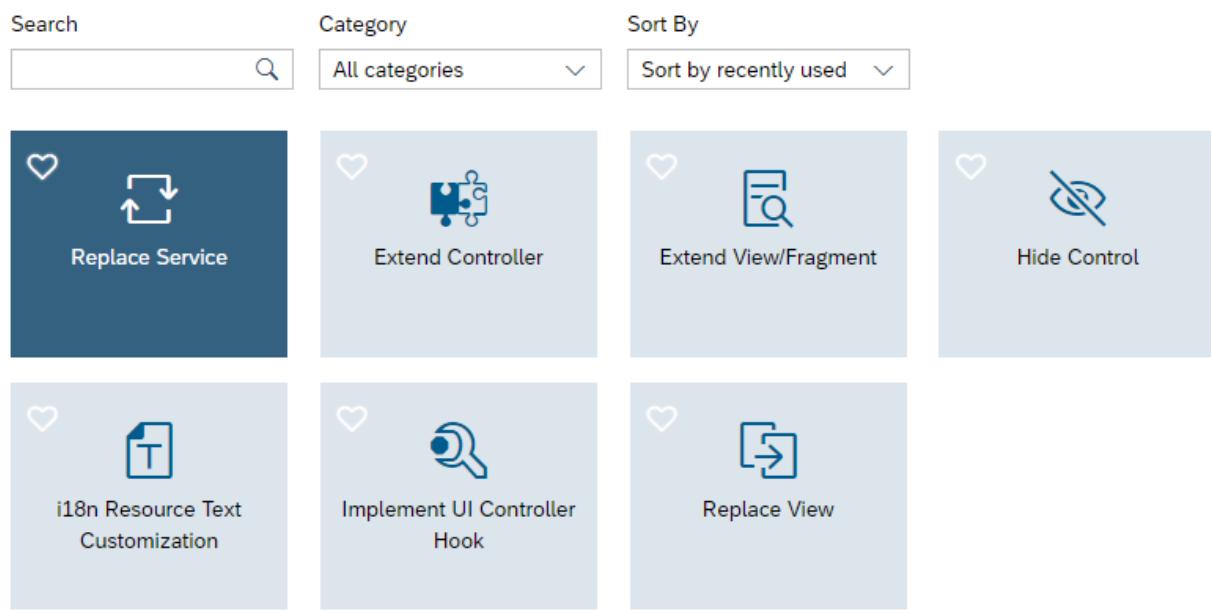
Abbildung 32: Anlegen eines neuen Erweiterungsprojektes
(Quelle: eigener Screenshot)

Eine Erweiterung konnte anschließend im Kontextmenü *New>Extension* angelegt und dort bestätigt werden, dass das vorbelegte Erweiterungsprojektes korrekt ist. Danach zeigte der Assistent sieben mögliche Erweiterungsarten an (siehe Abbildung 33):

- **Replace Service:** In dieser Option kann der von der Anwendung benutzte OData-Dienst getauscht werden.
- **Hide Control:** Versteckt ein ausgewähltes Oberflächenelement.
- **I18n Resource Text Customization:** Macht die i18n-Dateien im Erweiterungsprojekt verfügbar. Sie können anschließend durch den Entwickler angepasst werden.
- **Replace View:** Eine der in der Anwendung vorhandenen Ansichten kann bearbeitet werden. Als Startpunkt dient hierfür entweder eine leere Quelldatei oder eine Kopie der bestehenden Ansicht.

Extension Selection

Select the extension you want to add.



Extension to replace the OData service that is currently in use with a new one.

[Previous](#)

[Next](#)

Abbildung 33: Anzeige der Erweiterungsmöglichkeiten
(Quelle: eigener Screenshot)

- **Extend View/Fragment:** Erweitert Ansichten und *Fragmente*, in denen *Extension Point* existiert. Fragmente sind Oberflächenteile, die keine eigene Logik besitzen. Sie sind wiederverwendbar und werden unter anderem für Dialoge oder Nachrichten verwendet, die sich in eigenen kleinen Fenstern öffnen [100]. Extension Points dienen als Ankerpunkte für Erweiterungen in Ansichten [101]. Um eine Erweiterung an einer vorhergesehenen Stelle einzufügen, kann ein Fragment mit einem Extension Point verbunden werden [102]. In Controllern heißen diese *UI Controller Hooks* [103].
- **Extend Controller:** Einer der in der Anwendung verwendeten Controller kann bearbeitet werden. Als Startpunkt dient hierfür entweder eine leere Quelltextdatei oder eine Kopie des bestehenden Controllers.
- **Implement UI Controller Hook:** Anpassung eines Controllers an einer durch einen Platzhalter gekennzeichneten Stelle.

Die Option *Replace Service* bot die gesuchte Funktion – den Austausch von OData-Diensten. Es öffnete sich der gleiche Dialog, wie er auch bei Erstellung der Anwendung zur Auswahl des OData-Dienstes gezeigt wurde. Nach erfolgter Auswahl und Bestätigung konnte die Anwendung mit dem neuen OData-Dienst ausgeführt und ausgeliefert werden.

4.2 Einschätzung der SAP Cloud Platform

Im Rahmen der Arbeit konnte gezeigt werden, dass ein SAP-ECC-System mit Hilfe des Konnektor-Dienstes mit der SAP Cloud Platform verbunden werden kann. Wenn die Daten als OData-Dienst veröffentlicht werden, kann eine Fiori-ähnliche Anwendung darauf zugreifen und diese anzeigen. Bei der Änderung des OData-Dienstes ist in diesem Fall keine Änderung an der entwickelten Oberfläche nötig. Die Machbarkeit der gestellten Aufgabe konnte somit bestätigt werden.

Bei der Recherche und Umsetzung sind einige Punkte positiv und andere negativ aufgefallen. Da die Benutzung der Entwicklungsumgebung und die Anbindung von Systemen einen großen Teil der Arbeit ausmachten, werden diese auch gesondert eingeschätzt. Zudem ist zu beachten, dass sich der Eindruck in dieser Arbeit auf die Testversion der SAP Cloud Platform bezieht. Ein Test der produktiven Version war aus finanziellen Gründen nicht möglich. Von den zwei vorhandenen Umgebungen wurde nur die Neo-Umgebung evaluiert, da sie für die Umsetzung von Webanwendungen empfohlen wurde. Eine Analyse der Cloud-Foundry-Umgebung hätte einer weiteren Entwicklungsaufgabe bedurft, welche jedoch den Rahmen dieser Arbeit überschritten hätte.

4.2.1 Entwicklungsumgebung SAP Web IDE

Die SAP Web IDE bietet den schnellstmöglichen Start in die SAPUI5-Entwicklung. Die von SAP unterstützte Entwicklungsumgebung Eclipse kann durch Installation einer von SAP angebotenen Erweiterung vorbereitet werden. Bei WebStorm muss die Bibliothek manuell eingebunden und ein lokaler Webserver eingerichtet werden. Für Entwickler, die einen ersten Einblick in die SAPUI5-Entwicklung gewinnen wollen, ist die Web IDE daher eine gute Lösung.

Die Web IDE konnte die Entwicklung der Anwendung in diesem Fall abdecken und sich mit einigen Merkmalen positiv hervorheben. Aufgrund der vorhandenen Anwendungsvorlagen können Oberflächen besonders schnell erstellt werden. Mit Hilfe des graphischen Layout Editors ist die Änderung von Ansichten sogar mit sehr wenigen Programmierkenntnissen möglich. Die Anbindung des SAP Translation Hub vereinfacht die Übersetzung von Anwendungen in hohem Maße, da Ressourcendateien direkt aus dem Kontextmenü übersetzt werden können.

Zum Testen der Oberfläche können zur Laufzeit Sprachen und Auflösungen verändert werden. Ein QR-Code-Generator ermöglicht die schnelle Anzeige auf einem mobilen Endgerät. Die Funktionalität des Quelltextes sichern Modultests und Oberflächentests, die durch SAP vorkonfiguriert sind.

Die in dieser Arbeit gefundenen Nachteile der SAP Web IDE sind gegenüber den Vorteilen eher gering. Projekte im Arbeitsbereich können nicht geschlossen, also aus dem Arbeitsbereich entfernt, oder umbenannt werden. Zudem sind die Namenskonventionen für Anwendungen auf der SAP Cloud Platform unpraktisch, da sie nur Kleinbuchstaben und Zahlen zulassen. Unterstriche und Großbuchstaben, die einen positiven Einfluss auf die Lesbarkeit von langen Namen haben, sind verboten.

Im Rahmen dieser Arbeit fiel negativ auf, dass die Anwendungsvorlagen nicht für beliebig alte SAPUI5-Versionen verfügbar waren. Mit einer niedrigeren Version entwickelte Anwendungen sind aber immerhin lauffähig. In einem produktiven Szenario sollte ohnehin auf eine aktuellere Version des SAPUI5-Frameworks zurück gegriffen werden, da diese mehr Funktionalität bietet und Fehler aus vorherigen Versionen behebt.

Die Versionsverwaltung mit Git bietet in der SAP Web IDE alle von Git bereitgestellten Standardfunktionen. Die Bedienbarkeit war jedoch nicht immer intuitiv, einige Funktionen konnten sogar erst nach längerer Recherche in der Dokumentation der Web IDE gefunden werden.

4.2.2 Verbindungen zur Cloud Platform

Die Verbindung zwischen der SAP Cloud Platform und SAP-Backend-Systemen, die lokal installiert sind, ist dank des SAP-Cloud-Connector-Dienstes einfach herzustellen. Im Falle einer RFC-Verbindung können aus der SAP Cloud Platform Funktionsbausteine aufgerufen werden. Dies war jedoch nicht Bestandteil dieser Arbeit. Ebenso ist eine HTTP(S)-Verbindung möglich. Diese kann auch Backend-Systeme anderer Anbieter mit der Plattform verknüpfen. Mit dem Protokoll LDAP kann das Nutzerverzeichnis *Active Directory* von Microsoft mit der SAP Cloud Platform verbunden werden, um dort als Identitätsanbieter zu dienen.

Der SAP Cloud Connector ist kostenlos und baut einen mit SSL verschlüsselten Tunnel zwischen beiden Umgebungen auf. Seine Oberfläche ist an die SAP Cloud Platform angepasst. Das trägt zur schnelleren Orientierung des Nutzers bei.

Die Anwendung steht im Internet als portable und fest-installierbare Version zur Verfügung. Letztere ist im Rahmen dieser Arbeit negativ aufgefallen, da sie sich nach Entfernung einer Java-Installation nicht mehr vollständig vom Rechner deinstallieren ließ. Aufgrund verbleibender Programmteile war anschließend eine Neuinstallation nicht mehr möglich. Dennoch bevorzugt ASP für produktive Szenarien eine Installation des Konnektors.

Um einen besonders ausfallsicheren Tunnel zu erzeugen können zwei Instanzen des Konnektor-Dienstes verbunden werden. Dies nennt sich dann Hochverfügbarkeitsmodus (*High-Availability*). Die führende Instanz (*Master*) leitet alle Konfigurationen an die Hilfsinstanz (*Shadow*) weiter. Dieser prüft regelmäßig, ob der Master noch erreichbar ist und übernimmt bei einem Ausfall dessen Funktion. Die Anbindung von Diensten aus dem Internet erfolgt direkt in der SAP Cloud Platform durch Anlage einer Zielverknüpfung (*Destination*).

4.2.3 SAP Cloud Platform als Ganzes

Ein positiver Aspekt der SAP Cloud Platform ist der große Umfang der Testversion. In der Neo-Umgebung stehen vierzig Dienste zur Verfügung. Dabei ist vor allem die zeitlich unbegrenzte und kostenfreie Nutzung der Testversion als besonders positiv zu bewerten. Die Dienste sind verzahnt und bilden vollständige Entwicklungsszenarien ab.

Die Verfügbarkeit der SAP Cloud Platform ist in der Testversion nicht garantiert. Dennoch kam es während der Erstellung dieser Arbeit nur zu einem Ausfall, der auf der Statusseite vermerkt und in kurzer Zeit behoben wurde. Falls die SAP Cloud Platform während einer produktiven Nutzung ausfällt und sie ausschließlich zur Entwicklung von Anwendungen genutzt wird, kann mit der lokalen Version der Web IDE gearbeitet werden. Dies funktioniert natürlich nur unter der Bedingung, dass die Entwicklungsprojekte mit einem Git-Verzeichnis verknüpft und regelmäßig gespeichert wurden.

Werden die Dienste der SAP Cloud Platform zur Auslieferung von Anwendungen an Kunden genutzt, besteht ein Vertrag zwischen Arvato Systems Perdata und ihnen. Dieser schließt immer eine Dienststufenvereinbarung mit ein. Die darin garantierte Systemverfügbarkeit darf die von SAP nicht überschreiten. Ebenso dürfen die *Pönalen* (alternativ Vertragsstrafen) für Verstöße gegen Vertragsbestandteile bzw. geltendes Recht nicht zu weit von denen abweichen, die SAP zahlt. Wenn dies gegeben ist, entstehen für Arvato Systems Perdata bei einem Ausfall keine finanziellen Nachteile, da der Betrieb der Infrastruktur bei SAP liegt. Die Wartungsfenster für einzelne Dienste werden durch SAP frühzeitig angekündigt und können den Kunden rechtzeitig bekannt gegeben werden.

Zur Zielgruppe der SAP Cloud Platform gehören neben bestehenden SAP-Kunden auch Entwickler aus einem SAP-fremden Umfeld, die von der In-Memory-Technologie der HANA-Datenbank oder vom SAPUI5-Framework zur Entwicklung von modernen, geräteunabhängigen Oberflächen profitieren wollen. Diese benötigen bei Verwendung der SAP Cloud Platform keine SAP-Kenntnisse und keine weiteren SAP-Produkte. Sie können laut SAP sogar ihre eigene Programmiersprache und andere Werkzeuge in die SAP Cloud Platform integrieren. Dies ist jedoch nicht Bestandteil dieser Arbeit.

Den hohen Ansprüchen an die Ausführungsgeschwindigkeit von Anwendungen begnett SAP durch das Angebot verschiedener Serverstandorte (Regionen). Die Latenzzeiten bei der Kommunikation mit Servern kann dadurch maßgeblich verringert werden. Zudem bieten die Regionen aber auch Sicherheit beim Datenschutz. Der Entwickler kann sicherstellen, dass personenbezogene Daten seiner Anwendungen, falls diese in der SAP Cloud Platform gespeichert werden, innerhalb der EU, sogar in Deutschland abgelegt werden. SAP behält sich vor, Daten in andere Länder und Regionen zu übermitteln, garantiert in diesem Fall jedoch die Einhaltung des EU-Datenschutzniveaus. Zur Sicherung der Anwendung bietet die SAP Cloud Platform moderne Technologien wie OAuth 2.0 und SAML, eigene Identitätsanbieter und eine umfangreiche Nutzer- und Rollenverwaltung.

Einige der Sicherungstechnologien ermöglichen gleichzeitig auch eine anwendungsübergreifende Anmeldung (Single Sign-On). Dies trägt zu einem positiven Nutzererlebnis bei. Innerhalb der SAP Cloud Platform und der Dienste wurde auf eine einheitliche Oberfläche geachtet, die übersichtlich sortiert ist. Die Dokumentation für alle Dienste und die Plattform an sich sind im Internet frei verfügbar. Zusätzlich gibt es zahlreiche *Blogs* und *Tutorials* zu einzelnen Themen der SAP Cloud Platform. Blogs sind Internetseiten auf denen regelmäßig tagebuchartige Einträge zu einem oder mehreren Themen veröffentlicht werden [104]. Der Begriff Tutorial bezeichnet eine „Dokumentation für Endbenutzer eines Softwaresystems, in der im Sinne einer Einführung die wichtigsten Systemfunktionen erläutert werden“ [105]. Allerdings sind viele dieser Informationen nicht in deutscher Sprache verfügbar.

Das Bezahlmodell der SAP Cloud Platform ist detailreich und lässt sich genau auf den Nutzer abstimmen. Genau deshalb wirkt es auch unübersichtlich und kompliziert. Ein Taschenrechner auf der Internetseite ermöglicht einen besseren Überblick. Eine Unstimmigkeit ist im Rahmen dieser Arbeit aufgefallen. Der OData-Provisioning-Dienst wurde verwendet, obwohl er laut der offiziellen Preisinformation nicht Teil der kostenlosen Testversion ist. Im Falle einer produktiven Nutzung wird ein Vertrag zwischen SAP und dem Nutzer geschlossen, der eindeutig die vereinbarten Leistungen und Preise darstellt und ähnliche Unstimmigkeiten daher ausschließt.

Insgesamt betrachtet hält die SAP Cloud Platform ihre Versprechen. Sie ermöglicht einem Entwickler eine schnelle Einarbeitung in die SAPUI5-Entwicklung. Erfahrene Entwickler sehen möglicherweise eher Nachteile in einer Cloud-basierten Entwicklungsumgebung und der Nutzung eines Frameworks. So beeinflusst beispielsweise SAPUI5 durch Nutzung des MVC-Architekturmusters „auch die Struktur der individuellen Anwendung“ [106]. Nutzer

mit wenig Erfahrung in der Webentwicklung können aber von der SAP Web IDE mit SAPUI5 profitieren [107].

5 Fazit und Ausblick

Im Rahmen dieser Arbeit sollte überprüft werden, ob die SAP Cloud Platform in der Lage ist, energiewirtschaftliche Individualprogrammierungen aus dem System zur Planung von Unternehmensressourcen *ECC* als *SAP-Fiori*-Anwendungen abzubilden. Hierfür wurde ein Beispiel ausgewählt und umgesetzt. Anschließend wurde evaluiert, ob dessen Oberfläche nach einmaliger Implementierung für einen Energieversorger mit Standardausprägung gleichermaßen nutzbar ist.

Die Machbarkeit der gestellten Aufgabe konnte im Rahmen der Arbeit gezeigt werden. Der Mehrwert der SAP Cloud Platform hängt vom genauen Nutzungsszenario und den Eigenschaften sowie Anforderungen des Nutzers ab. Sie eignet sich gut für bestehende SAP-Kunden, die neue Oberflächen mit Daten aus bestehenden Backend-Systemen entwickeln wollen. Diese Anwendungen können helfen, den Kunden an neue Oberflächen zu gewöhnen und so den Umstieg vom auslaufenden SAP-ERP-ECC-6.0- auf ein S/4-HANA-System erleichtern.

5.1 Probleme

Im Rahmen dieser Arbeit traten einige Probleme auf. Zu Beginn waren diese vor allem durch die ungewohnten Begrifflichkeiten bei SAP geschuldet. Aufgrund sich ähnelnder Produktnamen konnte die Angebotslandschaft von SAP nur mühsam erfasst werden. So gibt es zum Beispiel *SAP HANA*, *SAP Business Suite 4 HANA* (*SAP S/4 HANA*), *SAP Business Suite on HANA*, *S/4 HANA Cloud* und die *SAP Cloud Platform*, die zuvor *SAP HANA Cloud Platform* hieß.

Die SAP Cloud Platform wurde durch ASP im Vorfeld mit SAP S/4 HANA verwechselt und als nachfolgendes ERP-System verstanden. Erst im Zuge der Recherche konnte dieser Irrtum aufgeklärt und die Arbeit umstrukturiert werden. Dabei wurde zuerst erwägt, den Titel der Arbeit zu ändern und S/4 HANA zu evaluieren. Dies scheiterte jedoch daran, dass kein S/4-HANA-System für die Erstellung dieser Arbeit zur Verfügung stand. Zudem war unsicher, ob die kostenlose Testversion des S/4-HANA-Systems in der Cloud-Version die benötigten Berechtigungen aufweisen würde, um alle Aufgaben umzusetzen. Im Gespräch mit dem zuständigen Mitarbeiter bei ASP wurde daher der Arbeitsinhalt so definiert, wie er in dieser Fassung vorliegt.

Eine Unsicherheit entstand durch den permanenten Wandel der SAP Cloud Platform und aller aktuellen SAP Produkte. Diese werden ständig modular weiterentwickelt und so können Erkenntnisse, die zu Beginn der Bearbeitung gewonnen wurden, bei Abschluss der

Arbeit bereits veraltet sein. Dies ist auch ein generelles Problem für die Entwicklung und Anwendung der Produkte.

5.2 Ausblick

Die vorliegende Arbeit kann ASP als Grundlage zur Klärung weiterer Fragen dienen. In Bezug auf die Anwendung sollten zuerst das in Abschnitt 4.1.4 aufgetretene Problem analysiert und, wenn möglich, behoben werden. Es wurde festgestellt, dass die Kacheln, die in Fiori-Launchpads zur Darstellung der Anwendungsverknüpfungen verwendet werden, nicht dynamisch beschriftet und daher nicht übersetzt werden. Gerade auf der Übersichtsseite benötigt ein Anwender jedoch einen guten Überblick und für ihn verständliche Beschriftungen von Anwendungen. Eine Weiterentwicklung mit Umsetzung der in Abschnitt 3.2 definierten Soll- und Kann-Kriterien ist möglich. Weiterführend ist zu klären, ob die entwickelte Anwendung auch Daten aus einem S/4-HANA-Backend anzeigen kann und wie dort die Erstellung und Veröffentlichung von OData-Diensten funktioniert.

In Bezug auf das SAPUI5-Framework und die SAP Web IDE konnte in dieser Arbeit nicht untersucht werden, welcher Mehrwert für erfahrene Webentwickler besteht. Auch aus Kostengründen ist zu prüfen, ob bei der Entwicklung von Anwendungen für SAP-Systeme auf die SAP Cloud Platform verzichtet werden kann. Die integrierte Entwicklungsumgebung WebStorm der Firma JetBrains wurde bereits in Abschnitt 3.7 erwähnt und stellt eine kostengünstige Alternative zur SAP Web IDE dar. Die Schritte zur Einbindung [108] des quelloffenen OpenUI5-Frameworks und passender Werkzeugen [109] können im Internet nachvollzogen werden. SAPUI5 selbst basiert auf anderen Oberflächen-Frameworks wie *Bootstrap* und *jQuery*. Es stellt sich daher die Frage, ob Fiori-ähnliche Anwendungen nicht auch ohne SAPUI5 mit anderen Frameworks erstellt werden können. Während die grundlegende Entwicklung damit möglicherweise mehr Zeit in Anspruch nimmt, sind die Ergebnisse anschließend immer genau auf die Anforderungen zugeschnitten und bleiben anpassbar [107].

In Bezug auf die SAP Cloud Platform ist weitergehend zu evaluieren, ob diese zur produktiven Auslieferung von Anwendungen genutzt werden kann. Der technische Aspekt wurde im Rahmen dieser Arbeit nur oberflächlich und nicht vollständig betrachtet. Das gleiche trifft auf die finanzielle Komponente zu. Auch der Mehrwert der SAP Cloud Platform für Entwicklungen außerhalb des SAP-Umfeldes ist zukünftig für ASP von großem Interesse. Für dessen Feststellung spielt die Betrachtung von Konkurrenzprodukten eine wichtige Rolle. Unabhängig von der SAP Cloud Platform ist in Zukunft eine bessere Absicherung der SAP-Backend-Systeme durch *2-Faktor-Authentifizierung* oder die Implementierung von SSO-Methoden empfehlenswert.

6 Literaturverzeichnis

- [1]. **Wikipedia.** Funktion (Programmierung). [Online] 29. Mai 2018. [Zitat vom: 02. Juli 2018.]
[https://de.wikipedia.org/wiki/Funktion_\(Programmierung\)](https://de.wikipedia.org/wiki/Funktion_(Programmierung)).
- [2]. **Universität Leipzig, Wirtschaftswissenschaftliche Fakultät.** Richtlinien zur Gestaltung wissenschaftlicher Arbeiten . [Online] 05. Mai 2011. [Zitat vom: 10. Juli 2018.] Universität Leipzig .
- [3]. **SAP SE.** SAP S/4 HANA. [Online] [Zitat vom: 08. Mai 2017.]
<https://www.sap.com/germany/products/s4hana-erp.html>.
- [4]. **SAP AG.** *Teilnehmerhandbuch CRM Customizing - Grundlagen*. 2010.
- [5]. **Lanninger, Volker und Wendt, Oliver.** Enzyklopädie der Wirtschaftsinformatik. *Parametrisierung von Standardsoftware*. [Online] 25. Oktober 2012. [Zitat vom: 03. April 2018.]
<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Einsatz-von-Standardanwendungssoftware/Customizing-von-Standardsoftware/Parametrisierung-von-Standardsoftware>.
- [6]. **Lanninger, Volker und Wendt, Oliver.** Enzyklopädie der Wirtschaftsinformatik. *Erweiterungsprogrammierung*. [Online] 25. Oktober 2012. [Zitat vom: 03. April 2018.]
<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Einsatz-von-Standardanwendungssoftware/Customizing-von-Standardsoftware/Erweiterungsprogrammierung>.
- [7]. **Schuh, Günther, [Hrsg.].** *Produktionsplanung und -steuerung - Grundlagen, Gestaltung und Konzepte*. 3. Auflage. Berlin Heidelberg : Springer, 2006.
- [8]. **SAP SE.** SAP Support Portal. *SAP Support Strategy*. [Online] [Zitat vom: 05. April 2018.]
<https://support.sap.com/en/offerings-programs/strategy.html>.
- [9]. **Prof. Dr. Plattner, Hasso.** Enzyklopädie der Wirtschaftsinformatik. *In-Memory Data Management*. [Online] 05. September 2017. [Zitat vom: 02. Mai 2018.]
<http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/datenwissen/Datenmanagement/Datenbanksystem/In-Memory-Data-Management/index.html/?searchterm=in-memory>.

- [10]. **Walch, Christian.** Computerwoche. *SAP Fiori - Schneller Einstieg in die mobile IT-Strategie.* [Online] 22. Oktober 2014. [Zitat vom: 04. Mai 2018.]
<https://www.computerwoche.de/a/schneller-einstieg-in-die-mobile-it-strategie,3069372,3>.
- [11]. **SAP SE.** SAP Fiori UX. [Online] März 2015. [Zitat vom: 08. Mai 2018.]
<https://www.sap.com/documents/2014/06/f6a44520-5a7c-0010-82c7-eda71af511fa.html>.
- [12]. **Wikipedia.** SAP Cloud Platform. [Online] 19. Januar 2018. [Zitat vom: 08. Mai 2018.]
https://de.wikipedia.org/wiki/SAP_Cloud_Platform.
- [13]. **SAP SE.** Geschichte. [Online] [Zitat vom: 07. April 2018.]
<https://www.sap.com/corporate/de/company/history/>.
- [14]. **Autobild.** Drei auf einer Plattform. [Online] 05. April 2002. [Zitat vom: 11. April 2018.]
<http://www.autobild.de/artikel/seat-ibiza-gegen-skoda-fabia-und-vw-polo-36151.html>.
- [15]. **Dr. Arnold, René, et al.** *Internet-basierte Plattformen und ihre Bedeutung in Deutschland.* Bad Honnef : Wissenschaftliches Institut für Infrastruktur und Kommunikationsdienste, 2016.
- [16]. **Repschläger, Jonas, Pannicke, Danny und Zarnekow, Rüdiger.** Cloud Computing: Definitionen, Geschäftsmodelle und Entwicklungspotenziale. *HMD Praxis der Wirtschaftsinformatik.* 2010, Bd. 47, 5, S. 6–15.
- [17]. **Alpar, Paul, et al.** *Anwendungsorientierte Wirtschaftsinformatik - Strategische Planung, Entwicklung und Nutzung von Informationssystemen.* Wiesbaden : Springer Vieweg, 2016.
- [18]. **Klees, Frank und Moehlmann, Thore.** Die Cloud in der digitalen Revolution und ihre Bedeutung für das SAP-Angebot. *HMD Praxis der Wirtschaftsinformatik.* 2016, Bd. 53, 5, S. 619–634.
- [19]. **Mell, Peter M. und Grance, Timothy.** *The NIST definition of cloud computing.* Gaithersburg, MD : National Institute of Standards and Technology, 2011. S. 7.
- [20]. **Baun, Christian, et al.** Cloud Computing - Web-basierte dynamische IT-Services. 2. Auflage, 2011.
- [21]. **Wagner, Klaus-P., Hüttl, Thomas und Backin, Dieter.** *Einführung Wirtschaftsinformatik - IT-Grundwissen für Studium und Praxis.* [Hrsg.] Iris Vieweg und Christian Werner. Wiesbaden : Gabler Verlag | Springer Fachmedien, 2012.

- [22]. **Barton, Thomas.** *E-Business mit Cloud Computing - Grundlagen, Praktische Anwendungen, verständliche Lösungsansätze*. Wiesbaden : Springer Vieweg, 2014.
- [23]. **Fettke, Peter.** Enzyklopädie der Wirtschaftsinformatik. *Client-Server-Architektur*. [Online] 23. September 2016. [Zitat vom: 26. Juni 2018.] <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Softwarearchitektur/Architekturparadigmen/Client-Server-Architektur>.
- [24]. **Gilbert, Salina.** Cloudikon. *Wolkige Aussichten - Die Zeit des Cloud Computings*. [Online] 23. November 2015. [Zitat vom: 12. April 2018.] <https://cloudikon.de/wolkige-aussichten-die-zeit-des-cloud-computings/>.
- [25]. **Densborn, Frank, et al.** *Migration nach SAP S/4 HANA*. Bonn : Rheinwerk Verlag GmbH, 2017.
- [26]. **SAP SE.** SAP Help Portal. *Dokumentation der SAP Cloud Platform*. [Online] [Zitat vom: 16. April 2018.] https://help.sap.com/doc/bd6250c40c9c4c5391e3009a6f26dc3b/Cloud/en-US/SAP_Cloud_Platform.pdf.
- [27]. **Steiner, Matthias.** SAP HANA Blog. *The SAP HANA & Cloud Symbiosis*. [Online] 14. Oktober 2013. [Zitat vom: 19. April 2018.] <https://blogs.saphana.com/2013/10/14/the-sap-hana-cloud-platform-symbiosis/>.
- [28]. **Geierhos, Michaela und Bäumer, Frederik S.** Enzyklopädie der Wirtschaftsinformatik. *Text Mining*. [Online] 04. Februar 2018. [Zitat vom: 02. Juli 2018.] <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/KI-und-Softcomputing/text-mining>.
- [29]. **SAP SE.** SAP Cloud Platform. *Success Stories*. [Online] [Zitat vom: 20. April 2018.] https://cloudplatform.sap.com/content/skywalker/website/en_us/success.html.
- [30]. **Geall, Marc.** SAP News Center. *SAP Cloud Platform and Cloud Foundry: What Does It Mean for Partners?* [Online] 11. Juli 2017. <https://news.sap.com/sap-leonardo-live-sap-cloud-platform-and-cloud-foundry-what-does-it-mean-for-partners/>.
- [31]. **Shanmugham, Murali.** GROM. *SAP Cloud Platform Goes Multi-Cloud*. [Online] 23. Juni 2017. [Zitat vom: 26. April 2018.] <http://www.grom.com/sap-cloud-platform-goes-multi-cloud/>.

- [32]. **Klose, Ann-Cathrin.** entwickler.de. *Mit dem 12 Factor App Manifest zur guten Cloud-App.* [Online] 13. November 2015. [Zitat vom: 26. Juni 2018.] <https://entwickler.de/online/development/12-factor-app-manifest-cloud-189818.html>.
- [33]. **il1411.** Dev Insider. *Was sind Microservices?* [Online] 25. August 2017. [Zitat vom: 26. Juni 2018.] <https://www.dev-insider.de/was-sind-microservices-a-634583/>.
- [34]. **App Entwickler Verzeichnis.** Native Apps vs. Web Apps - Unterschiede und Vorteile. [Online] 28. April 2017. [Zitat vom: 26. Juni 2018.] <https://app-entwickler-verzeichnis.de/faq-app-entwicklung/11-definitionen/586-unterschiede-und-vergleich-native-apps-vs-web-apps-2>.
- [35]. **Kaufmann, Timothy.** *Geschäftsmodelle in Industrie 4.0 und dem Internet der Dinge.* Wiesbaden : Springer Fachmedien, 2015.
- [36]. **Neumann, Alexander.** Heise Developer. *PaaS: Zertifizierungsprogramm für Cloud Foundry gestartet.* [Online] 17. Dezember 2015. [Zitat vom: 23. April 2018.] <https://www.heise.de/developer/meldung/PaaS-Zertifizierungsprogramm-fuer-Cloud-Foundry-gestartet-3045722.html>.
- [37]. **Rentrop, Christian.** Dev Insider. *Was ist Java-Programmierung?* [Online] 06. Januar 2017. [Zitat vom: 29. Juni 2018.] <https://www.dev-insider.de/was-ist-java-programmierung-a-569186/>.
- [38]. **Drilling, Thomas.** Dev Insider. *Die JavaScript-Laufzeitumgebung Node.js.* [Online] 02. April 2018. [Zitat vom: 29. Juni 2018.] <https://www.dev-insider.de/die-javascript-laufzeitumgebung-nodejs-a-695744/>.
- [39]. **Ramamoorthy, Hariprasauth.** SAP Community. *Getting started with Python development – Bring Your Own Language.* [Online] 17. Mai 2017. [Zitat vom: 23. April 2018.] <https://blogs.sap.com/2017/05/17/getting-started-with-python-development-bring-your-own-language>.
- [40]. **Neumann, Alexander.** Heise Developer. *NetWeaver Cloud – Platform as a Service à la SAP.* [Online] 21. Juni 2012. [Zitat vom: 23. April 2018.] <https://www.heise.de/developer/meldung/NetWeaver-Cloud-Platform-as-a-Service-a-la-SAP-1622874.html>.

- [41]. **Kazi, Aiaz.** SAP HANA Blog. *Evolution of the SAP HANA Cloud Platform*. [Online] 10. Mai 2013. [Zitat vom: 23. April 2018.] <https://blogs.saphana.com/2013/05/10/evolution-of-the-sap-hana-cloud-platform>.
- [42]. **Masiero, Manuel.** Chip. *Die Plattform: SAP NetWeaver Cloud*. [Online] 08. Oktober 2012. [Zitat vom: 23. April 2018.] http://www.chip.de/artikel/Artikelserie-Cloud-Computing-SAP-4_57786088.html.
- [43]. **SapHanaTutorial.com.** Introduction to SAP HANA XS. [Online] [Zitat vom: 29. Juni 2018.] <http://saphanatutorial.com/sap-hana-xs-sap-hana-extended-application-services/>.
- [44]. **tutanch.** Dev Insider. *Was ist HTML5?* [Online] 10. März 2017. [Zitat vom: 29. Juni 2018.] <https://www.dev-insider.de/was-ist-html5-a-584719/>.
- [45]. **SAP SE.** SAP Cloud Platform. *Capabilities*. [Online] [Zitat vom: 20. April 2018.] <https://cloudplatform.sap.com/capabilities.html>.
- [46]. **Wikipedia.** DevOps. [Online] 15. Juni 2018. [Zitat vom: 15. Juni 2018.] <https://de.wikipedia.org/wiki/DevOps>.
- [47]. **Keidar, Michal.** SAP Blog. *SAP Web IDE Versions... Which do I use...?* [Online] 19. September 2017. [Zitat vom: 28. Mai 2018.] <https://blogs.sap.com/2017/09/19/sap-web-ide-versions...-which-do-i-use.../>.
- [48]. **SAP SE.** SAP Development Tools. [Online] [Zitat vom: 28. Mai 2018.] <https://tools.hana.ondemand.com>.
- [49]. **SAP SE.** SAP Help Portal. *SAP Web IDE Full-Stack*. [Online] 23. Mai 2018. [Zitat vom: 28. Mai 2018.] https://help.sap.com/doc/97abfdee97f744ed8987154fe3a78c0f/CF/en-US/SAP_Web_IDE_en.pdf.
- [50]. **TechTarget.** deploy. [Online] 2008. [Zitat vom: 28. Juni 2018.] <https://whatis.techtarget.com/definition/deploy>.
- [51]. **SAP SE.** SAP Help Portal. *SAP Translation Hub*. [Online] [Zitat vom: 15. Juni 2018.] <https://help.sap.com/viewer/ed6ce7a29bdd42169f5f0d7868bce6eb/Cloud/en-US/1b15cf69580449c0bd8525696c97b90d.html>.
- [52]. **Wikipedia.** Git. [Online] 05. Juni 2018. [Zitat vom: 26. Juni 2018.] <https://de.wikipedia.org/wiki/Git>.

[53]. **Wikipedia.** Representational State Transfer. [Online] 23. Juni 2018. [Zitat vom: 27. Juni 2018.] https://de.wikipedia.org/wiki/Representational_State_Transfer.

[54]. **Bönnen, Carsten, et al.** *OData und SAP Gateway*. Bonn : Galileo Press, 2014.

[55]. **Plessner, Christoph.** Computerwoche. *Self-Service-Portale in CRM integrieren*. [Online] 09. Juli 2013. [Zitat vom: 07. Mai 2018.] <https://www.computerwoche.de/a/self-service-portale-in-crm-integrieren,2541839>.

[56]. **SAP SE.** SAP Cloud Platform. *Pricing*. [Online] [Zitat vom: 20. April 2017.] <https://cloudplatform.sap.com/pricing.html>.

[57]. **SAP SE.** SAP Cloud Platform. *Pricing and Packages*. [Online] 2017. [Zitat vom: 20. April 2018.] https://cloudplatform.sap.com/content/dam/website/skywalker/en_us/PDFs/SAP_CP_Pricing_PDF_2_21.pdf.

[58]. **Esch, Sebastian.** [Online] IBsolution GmbH, 27. OOktober 2016. [Zitat vom: 17. Mai 2018.] <https://www.slideshare.net/IBSolutionGmbH/integration-von-cloud-und-onpremise-mit-der-sap-hana-cloud-platform>.

[59]. **Malenkovich, Serge.** Kaspersky. *Was ist eine Man-in-the-Middle-Attacke?* [Online] 10. April 2013. [Zitat vom: 26. Juni 2018.] <https://www.kaspersky.de/blog/was-ist-eine-man-in-the-middle-attacke/905/>.

[60]. **SAP SE.** SAP Cloud Trust Center. *Cloud Services*. [Online] [Zitat vom: 15. Juni 2018.] <https://www.sap.com/about/cloud-trust-center/cloud-service-level-agreements/cloud-services.html>.

[61]. **RheinEnergie AG.** Energielexikon. [Online] [Zitat vom: 06. Mai 2018.] https://www.rheinenergie.com/de/privatkundenportal/service_1/energielexikon/index.php?letter=B.

[62]. **Prof. Knabe, Ch.** Beuth Hochschule für Technik Berlin. *Pflichtenheft / Lastenheft*. [Online] 25. 09 2006. [Zitat vom: 25. 05 2018.] <http://public.beuth-hochschule.de/~knabe/fach/swp-i/Definition-PH.pdf>.

[63]. **Englbrecht, Michael und Wegelin, Michael.** *SAP Fiori - Implementierung und Entwicklung*. 2., aktualisierte und erweiterte Auflage. Bonn : Rheinwerk Verlag GmbH, 2017.

- [64]. **Duden.** *Look-and-feel, Look-and-Feel, das.* [Online] [Zitat vom: 27. Juni 2018.] https://www.duden.de/rechtschreibung/Look_and_feel.
- [65]. **Gorshkova, E.A., et al.** A UML-Based Modeling of Web Application Controller. *Programming and Computer Software*. 2005, Bd. Vol. 31, Nr. 1, S. 29-33.
- [66]. **Kunz, Andreas.** SAP Blog. *What is OpenUI5 / SAPUI5 ?* [Online] 11. Dezember 2013. [Zitat vom: 15. Juni 2018.] <https://blogs.sap.com/2013/12/11/what-is-openui5-sapui5>.
- [67]. **SAP SE.** Fiori Design Guidelines. [Online] [Zitat vom: 12. Juni 2018.] <https://experience.sap.com/fiori-design/>.
- [68]. **Taté Tsiledze, Jean-Francois.** Conet Technologie-Blog. *Der Weg zur ersten Fiori App.* [Online] 01. März 2016. [Zitat vom: 08. Mai 2018.] <https://www.conet.de/blog/technologien/sap/2016-03-01/weg-zur-ersten-fiori-app-3422>.
- [69]. **SAP SE.** Fiori App Reference Library. *Display Business Partners.* [Online] [Zitat vom: 07. Mai 2018.] [https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/#/detail/Apps\('BUP3'\)/S100P](https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/#/detail/Apps('BUP3')/S100P).
- [70]. **Shanmugham, Murali.** SAP Blog. *Fiori Cloud and supported landscape scenarios.* [Online] 06. Juli 2017. [Zitat vom: 03. Mai 2018.] <https://blogs.sap.com/2017/07/06/fiori-cloud-and-supported-landscape-scenarios/>.
- [71]. **SAP SE.** SAP Documentation. *Business Enablement Provisioning (IW_BEP).* [Online] 15. Februar 2018. [Zitat vom: 22. Mai 2018.] https://help.sap.com/saphelp_gateway20sp12/helpdata/en/a2/15f0b42f2948f6bb9e51f98e8c39e9/frameset.htm.
- [72]. **Fischer, Andre.** SAP Blog. *SAP Gateway deployment options in a nutshell.* [Online] 27. Mai 2013. [Zitat vom: 08. Mai 2018.] <https://blogs.sap.com/2013/05/27/sap-netweaver-gateway-deployment-options-in-a-nutshell/>.
- [73]. **Wikipedia.** Remote Function Call. [Online] 23. April 2018. [Zitat vom: 28. Juni 2018.] https://de.wikipedia.org/wiki/Remote_Function_Call.
- [74]. **SAP SE.** SAP Help Portal. *SAP Fiori: Einrichtung und Konfiguration.* [Online] [Zitat vom: 01. Juni 2018.]

<https://help.sap.com/viewer/d71464d9f3204ea8be1144d62acd9ac3/7.52/de-DE/270dd0bc79044592ab22545227f2435b.html>.

[75]. **SAP SE.** SAP Dokumentation. *Single Sign-On für Web-basierten Zugriff*. [Online] [Zitat vom: 03. Juni 2018.]

https://help.sap.com/saphelp_nw73ehp1/helpdata/de/4f/991d85b10c16c7e10000000a42189d/frameset.htm.

[76]. **Wikipedia.** Sitzung (Informatik). [Online] 03. April 2018. [Zitat vom: 28. Juni 2018.]
[https://de.wikipedia.org/wiki/Sitzung_\(Informatik\)](https://de.wikipedia.org/wiki/Sitzung_(Informatik)).

[77]. **Rieschel, Sebastian.** GS Lexikon. *Cookie*. [Online] [Zitat vom: 28. Juni 2018.]
<https://www.gruenderszene.de/lexikon/begriffe/cookie>.

[78]. **Wikipedia.** Hypertext Transfer Protocol. [Online] 22. Mai 2018. [Zitat vom: 29. Juni 2018.] https://de.wikipedia.org/wiki/Hypertext_Transfer_Protocol.

[79]. **SAP SE.** SAP Help Portal. *Identity Provider for Single Sign-On and SAP Identity Management*. [Online] 15. Mai 2017. [Zitat vom: 04. Juni 2018.]
https://help.sap.com/doc/339459818c4e4cb881c353e04a037a97/2.15/en-US/IdentityProviderForSAPSsingleSign-OnAndSAPIidentityManagement_uacp.pdf.

[80]. **Zosel, Ralf.** Zwei-Faktor-Authentifizierung. [Online] [Zitat vom: 29. Juni 2018.]
<https://ralfzosel.de/lexikon/zwei-faktor-authentifizierung/>.

[81]. **Bager, Jo.** Hinter die Mauern - Warum Sie im Jahr 2017 Ihre Daten in der EU behalten sollten. c't. 2017, 09/2007, S. 70.

[82]. **SAP SE.** SAP-Datenschutzerklärung. [Online] 22. Januar 2018. [Zitat vom: 31. Mai 2018.] <https://www.sap.com/germany/about/legal/privacy.html>.

[83]. **SAP SE.** Terms of Use Agreement for the SAP Cloud Platform Site. [Online] [Zitat vom: 31. Mai 2018.] https://help.hana.ondemand.com/terms_of_use.html.

[84]. **Prof. Dr. Neumann, Gustaf.** Enzyklopädie der Wirtschaftsinformatik. *XML*. [Online] 26. September 2014. [Zitat vom: 29. Juni 2018.] <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/technologien-methoden/Sprache/Auszeichnungssprache/XML/index.html>.

- [85]. **Mozilla Foundation.** MDN web docs. *Über JavaScript*. [Online] 27. Mai 2018. [Zitat vom: 29. Juni 2018.]
https://developer.mozilla.org/de/docs/Web/JavaScript/About_JavaScript.
- [86]. **Dr. Schwichtenberg, Holger.** IT-Visions. *Erklärung des Begriffs: Ressourcendatei*. [Online] [Zitat vom: 29. Juni 2018.] <https://www.it-visions.de/glossar/alle/4133/Ressourcendatei.aspx>.
- [87]. **o.V.** Einführung in JSON. [Online] [Zitat vom: 29. Juni 2018.] <https://www.json.org/json-de.html>.
- [88]. **Schäfer, Holger.** OpenUI Developer. *The Swiss Knife for UI5 Developers*. [Online] 2014. [Zitat vom: 15. Juni 2018.] <https://openui5.blogspot.com/p/ide.html>.
- [89]. **o.V.** SAP Blog. *SAP Web IDE – Online and Offline development with Web IDE on HCP and Web IDE Personal Edition*. [Online] 05. Oktober 2016. [Zitat vom: 01. Juni 2018.] <https://blogs.sap.com/2016/10/05/how-to-use-the-hana-cloud-platform-web-ide-personal-edition/>.
- [90]. **o.V.** SAP Blog. *Connect ABAP Backend to HCP via HCP OData provisioning*. [Online] 14. Oktober 2016. [Zitat vom: 10. Juli 2018.] <https://blogs.sap.com/2016/10/14/connect-abap-backend-hcp-via-hci-odata-provisioning/>.
- [91]. **SAP SE.** SAP Help Portal. *Connect to ABAP Systems*. [Online] [Zitat vom: 21. Juni 2018.] <https://help.sap.com/viewer/825270ffffe74d9f988a0f0066ad59f0/Cloud/en-US/5c3debce758a470e8342161457fd6f70.html>.
- [92]. **o.V.** SAP Archive. *IWFND and IWBEPE*. [Online] [Zitat vom: 05. Juni 2018.] <https://archive.sap.com/discussions/thread/3570063>.
- [93]. **Froehlich, Steffen.** SAP Blog. *NW Gateway: Tips & Tricks*. [Online] 18. Juli 2013. [Zitat vom: 29. Juni 2018.] <https://blogs.sap.com/2013/07/18/nw-gateway-tips-tricks/>.
- [94]. **SAP SE.** SAP Community WIKI. *SAP Web IDE - Technical FAQ*. [Online] 04. April 2018. [Zitat vom: 06. Juni 2018.] <https://wiki.scn.sap.com/wiki/display/SWI/SAP+Web+IDE+-+Technical+FAQ>.
- [95]. **SAP SE.** SAP Help Portal. *SAP Cloud Platform Portal*. [Online] [Zitat vom: 12. Juni 2018.] <https://help.sap.com/viewer/3ca6847da92847d79b27753d690ac5d5/Cloud/en-US/18b3fc15e6fe487baa99be07c1da82c3.html>.

- [96]. **SAP SE.** SAP HANA Cloud Documentation. *About SAP Fiori Launchpad Objects*. [Online] [Zitat vom: 14. Juni 2018.]
https://help.hana.ondemand.com/cloud_portal_flp/frameset.htm?18b3fc15e6fe487baa99be07c1da82c3.html.
- [97]. **Keidar, Michal.** SAP Blog. *Troubleshooting Deployment to an ABAP System from SAP Web IDE*. [Online] 25. Februar 2016. [Zitat vom: 13. Juni 2018.]
<https://blogs.sap.com/2016/02/25/troubleshooting-deployment-to-abap-system-from-sap-web-ide/>.
- [98]. **Wikipedia.** Stand-Alone. [Online] 26. September 2017. [Zitat vom: 06. Juli 2018.]
<https://de.wikipedia.org/wiki/Stand-Alone>.
- [99]. **o.V.** SAP Blog. *Cache Maintenance in Fiori*. [Online] 02. März 2016. [Zitat vom: 13. Juni 2018.] <https://blogs.sap.com/2016/03/02/cache-maintenance-in-fiori>.
- [100]. **o.V.** SAP Blog. *Use of Fragments in SAPUI5 (Reusability)*. [Online] 21. Mai 2018. [Zitat vom: 01. Juli 2018.] <https://blogs.sap.com/2015/05/21/use-of-fragments-in-sapui5-reusability/>.
- [101]. **SAP SE.** SAP Help Portal. *Extension Points*. [Online] [Zitat vom: 01. Juli 2018.]
https://help.sap.com/saphelp_ssbe/helpdata/en/84/1edf5209ca4b1de10000000a423f68/frameset.htm.
- [102]. **SAP AG.** How to extend Fiori Wave 2 Applications. [Online] 2014. [Zitat vom: 01. Juli 2018.] <https://archive.sap.com/kmuuid2/20dbf542-f3e7-3110-a6b8-ebe96b57c4e8/End-to-End%20SAP%20Fiori%20Extensibility%20Use%20Case.pdf>.
- [103]. **SAP SE.** SAP Help Portal. *UI Controller Hooks*. [Online] [Zitat vom: 01. Juli 2018.]
https://help.sap.com/saphelp_ssbe/helpdata/en/20/f0df5215eb5c3fe10000000a423f68/frameset.htm.
- [104]. **Kaczmarek, Joel.** GS Lexikon. *Blog*. [Online] [Zitat vom: 01. Juli 2018.]
<https://www.gruenderszene.de/lexikon/begriffe/blog>.
- [105]. **Prof. Dr. Lackes, Richard und Dr. Siepermann, Markus.** Gabler Wirtschaftslexikon. *Tutorial*. [Online] 19. Februar 2018. [Zitat vom: 01. Juli 2018.]
<https://wirtschaftslexikon.gabler.de/definition/tutorial-47052>.

[106]. **Wikipedia.** Framework. [Online] 27. März 2018. [Zitat vom: 09. Juli 2018.] <https://de.wikipedia.org/wiki/Framework>.

[107]. **Hirschmann, Dirk und Mistler, Elias.** HTML5 vs. SAPUI5. [Online] [Zitat vom: 19. Juni 2018.] https://www.pikon.com/fileadmin/Dateien/PIKON_DE/Artikel-pdfs/HTML5_vs_SAPUI5_E3.pdf.

[108]. **van het Hof, Robin.** Sap Blog. *Configuring JetBrains WebStorm for UI5 development.* [Online] 22. September 2014. [Zitat vom: 19. Juni 2018.] <https://blogs.sap.com/2014/09/22/configuring-jetbrains-webstorm-for-ui5-development/>.

[109]. **Schuff, Christian.** SAP Blog. *UI5ers Buzz #18: Useful Little Helpers—Faster SAPUI5 Development in WebStorm.* [Online] 20. November 2017. [Zitat vom: 19. Juni 2018.] <https://blogs.sap.com/2017/11/20/ui5ers-buzz-18-useful-little-helpers%E2%80%8A-%E2%80%8Afaster-sapui5-development-in-webstorm/>.

[110]. **predic8 GmbH.** OAuth2 Tutorial: Google as Authentication Service for Web Applications. [Online] [Zitat vom: 05. Juni 2018.] <https://www.membrane-soa.org/service-proxy-doc/4.4/oauth2-google.htm>.

[111]. **Wikipedia.** Europäische Gesellschaft. [Online] 17. April 2018. [Zitat vom: 08. Mai 2018.] https://de.wikipedia.org/wiki/Europ%C3%A4ische_Gesellschaft.

7 Anhangsverzeichnis

A	SAP Support-Chat	XXI
B	Überblick über Testkonten in der SAP Cloud Platform	XXII
C	Definition des Wortes <i>Principal</i> in Transaktion SAPTERM	XXIII
D	Verbindung zwischen SAP Cloud Connector und der SAP Cloud Platform	XXIV
E	Anlage eines SAP-Systems im SAP Cloud Connector	XXVII
F	Quelltexte zur Implementierung des OData-Dienstes.....	XXXI
G	Informationen zur Version des SAPUI5-Frameworks im SGW-System	XXXIII
H	Anpassen der Anwendungsvorlage	XXXIV
I	Ansicht der Anwendung nach der Erstellung	XXXVI
J	Ausliefern einer Anwendung in die SAP Cloud Platform.....	XXXVIII
K	Anlegen eines Fiori-Launchpad im Portal-Dienst der SAP Cloud Platform.....	XXXIX
L	Registrierung einer Anwendung auf einem Launchpad in der Cloud	XLI
M	Ausliefern einer Anwendung auf einen lokalen Applikationsserver.....	XLIII
N	Registrieren einer Anwendung auf einem lokalen Fiori-Launchpad.....	XLV

8 Anhang

A SAP Support-Chat

Info um 11:29, Apr 23:

Sie möchten mehr über unsere Lösungen und Services erfahren? Bitte warten Sie, während wir den nächsten verfügbaren SAP-Mitarbeiter kontaktieren. Alle Angaben, die Sie hier machen, werden ausschließlich für dieses Gespräch genutzt. Näheres entnehmen Sie bitte unserer [Datenschutzerklärung](#).

Sie chatten jetzt mit Herr Krämer.

Angela um 11:29, Apr 23:

Hallo Herr Krämer,

Herr Krämer um 11:29, Apr 23:

Hallo, wie kann ich Ihnen helfen?

Angela um 11:29, Apr 23:

ist es möglich, die SAP Cloud Platform auf einem eigenen Server zu hosten?

Herr Krämer um 11:31, Apr 23:

Grundsätzlich ist dies möglich. Sie benötigen aber die entsprechende Infrastruktur.

Herr Krämer um 11:32, Apr 23:

Hier finden Sie Infos zur Cloud Platfom: <https://cloudplatform.sap.com/index.html>

Angela um 11:33, Apr 23:

was wäre die entsprechende Infrastruktur? Und sehen Sie darin überhaupt einen Vorteil?

Angela um 11:33, Apr 23:

vielleicht reicht es, wenn sie die zweite Frage beantworten :-D

Angela um 11:33, Apr 23:

die erste war etwas zu allgemein.

Herr Krämer um 11:35, Apr 23:

Wenn Sie Ihre Daten nicht der SAP anvertrauen wollen, ist die private Cloud die Möglichkeit. Im Prinzip ist die allgemeine Cloud sinnvoller, da Sie sich nicht um die IT kümmern müssen sondern nur user sind.

Herr Krämer um 11:37, Apr 23:

Wir sprechen bei der Cloud Platform von PaaS, Platform as a Service.

Angela um 11:37, Apr 23:

wie funktioniert das Bezahlmodell, wenn wir die Cloud Platform selbst hosten würden? gibt es ein subscription-based package für diesen Fall?

Herr Krämer um 11:39, Apr 23:

Ich bedauere, Kosten und Lizenzmodelle werden ausschließlich durch unseren Vertrieb kommuniziert.

Angela um 11:39, Apr 23:

schade. trotzdem vielen Dank für Ihre Bemühungen, wie kann ich jemanden erreichen, der mir diese Frage beantworten könnte?

Herr Krämer um 11:40, Apr 23:

Setzen Sie berets SAP Lösungen ein?

Angela um 11:41, Apr 23:

ja, wir sind als IT-Outsourcing Dienstleister auf SAP-Produkte spezialisiert.

Herr Krämer um 11:42, Apr 23:

Sind Sie SAP Partner?

Angela um 11:42, Apr 23:

ja.

Herr Krämer um 11:42, Apr 23:

Dann haben Sie einen Partner Manager. Er wäre der richtige Ansprechpartner.

Angela um 11:43, Apr 23:

Vielen Dank! Ich wünsche Ihnen eine schöne Woche.

Quelle:<https://www.sap.com/germany/index.html>

B Überblick über Testkonten in der SAP Cloud Platform

Trial Accounts

	Cloud Foundry Environment	Neo Environment
Use case	A trial account enables you to explore the basic functionality of the Cloud Foundry environment for 90 days. Access is open to everyone.	A trial account enables you to explore the basic functionality of the Neo environment for a non-committal and unlimited period. Access is open to everyone.
Benefits	<ul style="list-style-type: none"> • Free of charge • Self-service registration • Usage possible for 30 days. You can extend the trial period to a maximum of 90 days, after which your account is automatically deleted. <p>The cockpit shows the time left in a free trial. Once a trial is suspended, you can still log on to it, but you won't be able to use applications or services. Between 30 and 90 days after the creation of the trial, you can renew the trial by using the <i>Extend Free Trial</i> button in the cockpit.</p>	<ul style="list-style-type: none"> • Free of charge • Self-service registration • Usage possible for an unlimited period of time • A trial tenant database on a shared HANA MDC system that you can use for 12 hours. <div style="background-color: #FFFACD; padding: 10px; border-radius: 5px;"> <p>⚠ Restriction</p> <p>After 12 hours, it is shut down automatically to free resources (see Restrictions [page 610]).</p> <p>If you do not use the tenant database for 7 days, it is automatically deleted.</p> </div> <ul style="list-style-type: none"> • Multiple deployed Java applications
Services available	Productive and beta services.	Productive and beta services.

Quelle [26, S. 15]

C Definition des Wortes *Principal* in Transaktion SAPTERM

The screenshot shows the SAPTERM application interface. At the top, there is a navigation bar with the SAP logo and the word "term". Below it, a search bar has "Quellsprache Englisch" (Source language English), "Zielsprache Deutsch" (Target language German), and "Suchbegriff Principal" (Search term Principal) set. A "Suchen" (Search) button is also present.

Result 1: principal (BC-SEC)

- Komponente:** BC-SEC
- Englisch:** principal
- Deutsch:** Principal
- Definition:** Security (BC-SEC) / Sicherheit (BC-SEC)
- Beschreibung:** Object in SAP User Management Engine, for example, user, user account, group, or role. / Objekt der SAP User Management Engine, z.B. ein Benutzer, ein Benutzerkonto, eine Gruppe oder eine Rolle.

Result 2: principal (STM0000004)

- Komponente:** STM0000004
- Englisch:** system principal
- Deutsch:** Systembenutzer
- Definition:** Portal Knowledge Management (EP-KM) / Portal Knowledge Management (EP-KM)
- Beschreibung:** A user who has been assigned system-wide permissions in the KM configuration. / Ein Benutzer, dem systemweite Berechtigungen in der KM-Konfiguration zugewiesen wurden.
- Details:** A system principal is defined in the portal user management and may be
 - An individual user
 - A group
 - A role
 A system principal is typically employed by system services (such as, the index management service) that need certain permissions for all resources.

Quelle: eigener Screenshot

D Verbindung zwischen SAP Cloud Connector und der SAP Cloud Platform

D.1 Eingabe der Verbindungsdaten

Add Subaccount

*Region Host:	hanatrial.ondemand.com	▼
*Subaccount:	s*****trial	
Display Name:		
*Subaccount User:	s*****	
*Password:	
Location ID:	<i>Enter location ID to overwrite default</i>	
Description:		

Save Cancel

Quelle: eigener Screenshot

D.2 Detailanzeige des verbundenen Kontos

The screenshot shows the Cloud Connector Administration interface. At the top, it displays "Cloud Connector Administration" and the subaccount "s...trial". A message indicates a connection to "hanatrial.ondemand.com" since 02.07.2018 12:39:50. Below this, the "Connector State" section shows various configuration details like Region Host, Subaccount, and System Certificate. The "Disaster Recovery Subaccount" section notes that features are not available for trial subaccounts. The "Tunnel Information" section provides details about the current tunnel connection, including Status, Remote Name, Tunnel ID, Connector ID, Local Name, and Local IP. The "Cloud Connections" section shows a table with columns for Application, Connections, Connected Since, Peer Labels, and Actions, but it displays "No data".

Application	Connections	Connected Since	Peer Labels	Actions
No data				

Quelle: eigener Screenshot

D.3 Anzeige von Konnektoren in der SAP Cloud Platform

The screenshot shows the SAP Cloud Platform Cockpit interface. At the top, there are navigation links: Home [Europe (Rot) - Trial], Europe (Rot) - Trial, Subaccount: s..., and Cloud Connectors. A user profile icon for 'Angela' is also present.

Master Instance (Connected)

Connector ID: 6D9D1B80667311EBB21DDEA60AD49FA
Connected since: 02.07.2018 12:29:13
Initiated by: s...
Version: 2.10.2
Java Version: 1.7.0_161 (SAP AG)
High Availability: inactive

Exposed Back-End Systems

Host	Protocol	Back-End Type	Resources
sgw:8001	HTTP	SAP Gateway	Available
sgw:8002	HTTP	ABAP System	Available

Master Instance (Leipzig)

Connector ID: 28D250007BE31E8B363D25B0A0F010B
Connected since: 02.07.2018 12:13:55
Initiated by: s...
Version: 2.10.2
Java Version: 1.7.0_161 (SAP AG)
High Availability: inactive

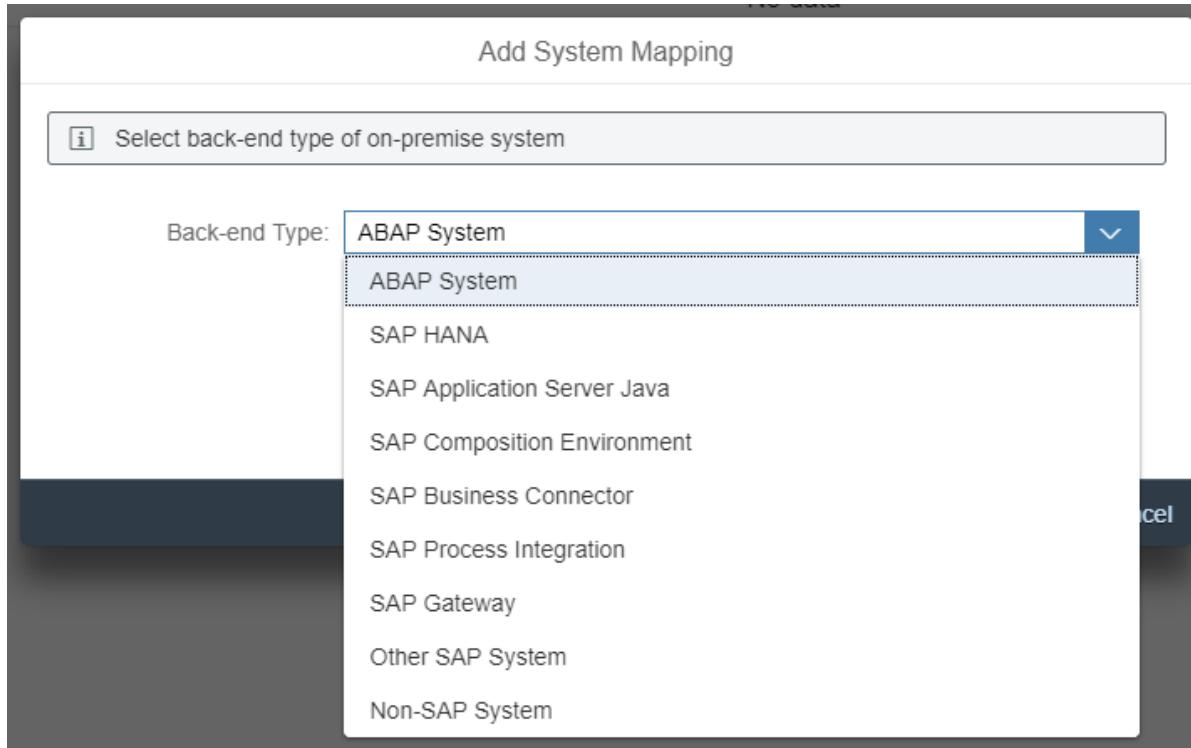
Exposed Back-End Systems

Host	Protocol	Back-End Type	Resources
sgw:8001	HTTP	SAP Gateway	Available

Quelle: eigener Screenshot

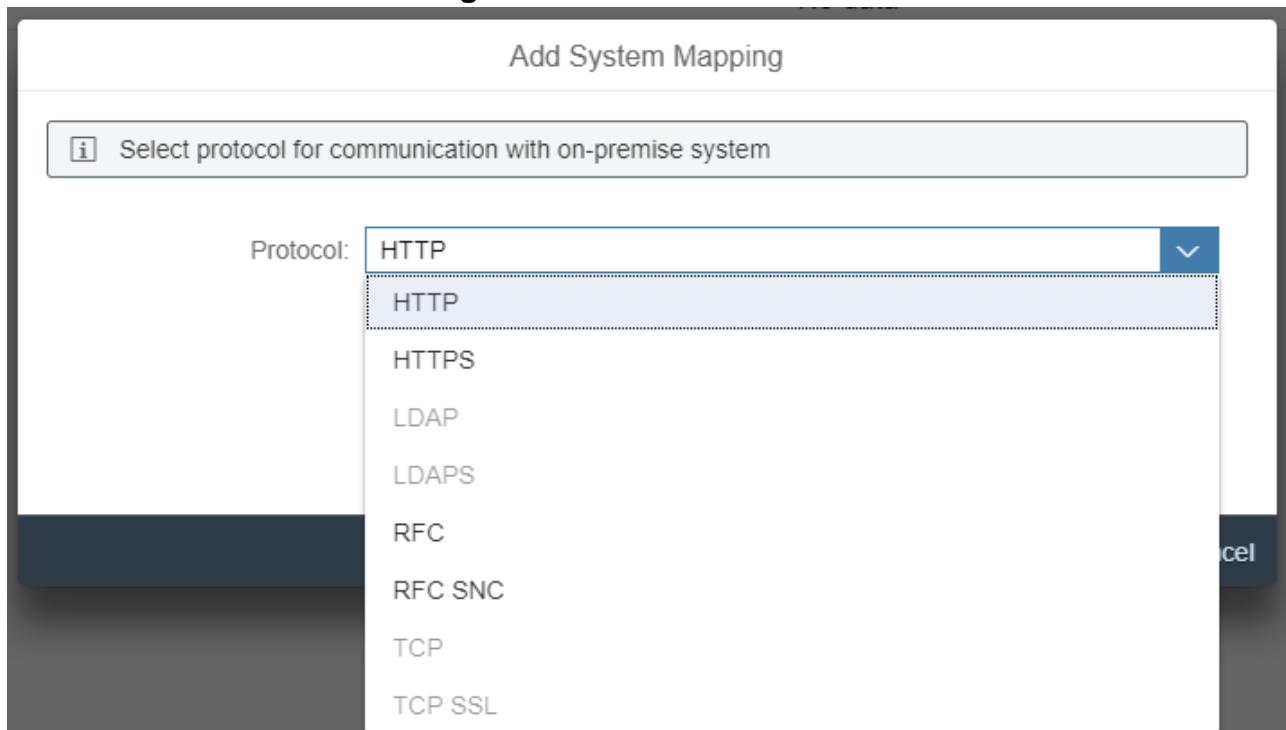
E Anlage eines SAP-Systems im SAP Cloud Connector

E.1 Auswahl der Systemart



Quelle: eigener Screenshot

E.2 Auswahl der Verbindungsart



Quelle: eigener Screenshot

E.3 Eingabe der Verbindungsdaten

Add System Mapping

i Enter internal (on-premise) host and port

*Internal Host:

*Internal Port:

[Previous](#) [Next](#) [Cancel](#)

Quelle: eigener Screenshot

E.4 Auswahl des Benutzertypen

Add System Mapping

i Select principal type

Principal Type: ▼

- None
- X.509 Certificate
- Kerberos

[Previous](#) [Next](#) [Cancel](#)

Quelle: eigener Screenshot

E.5 Möglichkeit zur Eingabe einer Beschreibung

The screenshot shows a user interface titled 'Add System Mapping'. At the top, there is a note: 'Optionally enter a description'. Below this is a large text input field labeled 'Description:' with a placeholder '|'. At the bottom of the screen, there are navigation buttons: 'Previous', 'Next', and 'Cancel'.

Quelle: eigener Screenshot

E.6 Zusammenfassung der Systemdaten

The screenshot shows a user interface titled 'Add System Mapping'. It displays a summary section with the following details:

- Protocol: HTTPS (None)
- Internal: [REDACTED].perdom.intern:8001
- Virtual: sgw:8001
- Check Internal Host:

At the bottom, there are navigation buttons: 'Previous', 'Finish', and 'Cancel'.

Quelle: eigener Screenshot

E.7 Übersicht über angelegte Systeme

The screenshot shows the SAP Cloud Connector Administration interface. The left sidebar is a navigation tree with the following structure:

- Connector
 - Cloud To On-Premise
 - On-Premise to Cloud
 - Monitor
 - Audits
 - Log And Trace Files

The main area is titled 'Cloud To On-Premise' and contains several tabs: ACCESS CONTROL, COOKIE DOMAINS, TRUSTED APPLICATIONS, and PRINCIPAL PROPAGATION. Under the ACCESS CONTROL tab, there is a table titled 'Mapping Virtual To Internal System' with the following data:

Status	Virtual Host	Internal Host	Check Result	Protocol	Back-end Type	Actions
◊	sgw:8001	[REDACTED].perdom.int...	<input checked="" type="checkbox"/> Reachable	HTTPS	SAP Gateway	

Below the table, there is a section titled 'Resources Accessible On sgw:8001' with a table showing 'No data'.

Quelle: eigener Screenshot

E.8 Freigabe von Ressourcen des angelegten Systems

The screenshot shows a dialog box titled "Add Resource". It contains the following fields:

- *URL Path:
- Enabled:
- Access Policy:
 - Path only (sub-paths are excluded)
 - Path and all sub-paths
- Description:

At the bottom right of the dialog box are two buttons: "Save" and "Cancel".

Quelle: eigener Screenshot

F Quelltexte zur Implementierung des OData-Dienstes

F.1 Entität Person

Methode PERSONSET_GET_ENTITYSET aktiv

```
1 method PERSONSET_GET_ENTITYSET.  
2     DATA:ls_filter_select_options TYPE /iwbp/s_mgw_select_option,  
3           ls_select_option      TYPE /iwbp/s_cod_select_option.  
4     DATA:ls_selectparams TYPE selopt,  
5           lt_selectparams TYPE STANDARD TABLE OF selopt.  
6     LOOP AT it_filter_select_options INTO ls_filter_select_options.  
7         IF ls_filter_select_options-property EQ 'Partner'.  
8             LOOP AT ls_filter_select_options-select_options INTO ls_select_option.  
9                 ls_selectparams-sign = ls_select_option-sign.  
10                ls_selectparams-option = ls_select_option-option.  
11                ls_selectparams-low = ls_select_option-low.  
12                ls_selectparams-high = ls_select_option-high.  
13                APPEND ls_selectparams TO lt_selectparams.  
14            ENDLOOP.  
15        ENDIF.  
16    ENDLOOP.  
17    SELECT * FROM ysa_gp INTO CORRESPONDING FIELDS OF TABLE ET_ENTITYSET  
18      WHERE partner in lt_selectparams.  
19  
20  
21 endmethod.
```

Quelle: eigener Screenshot

Methode PERSONSET_GET_ENTITY aktiv

```
1 method PERSONSET_GET_ENTITY.  
2     data: ls_key like line of it_key_tab.  
3  
4     read table it_key_tab into ls_key with key name = 'Partner'.  
5     if sy-subrc eq 0.  
6         select * from ysa_gp into corresponding fields of er_entity  
7             where partner = ls_key-value.  
8         endselect.  
9     else.  
10    endif.  
11 endmethod.
```

Quelle: eigener Screenshot

F.2 Entität „Vertragskonto“

Methode	VERTRAGSKONTOSET_GET_ENTITYSET	aktiv
1	<code>method VERTRAGSKONTOSET_GET_ENTITYSET.</code>	
2		
3	<code>DATA:ls_filter_select_options TYPE /iwbp/s_mgw_select_option,</code>	
4	<code> ls_select_option TYPE /iwbp/s_cod_select_option.</code>	
5	<code>DATA:ls_selectparams TYPE selopt,</code>	
6	<code> lt_selectparams TYPE STANDARD TABLE OF selopt.</code>	
7	<code>data: ls_key like line of it_key_tab.</code>	
8		
9	<code>read table it_key_tab into ls_key with key name = 'Partner'.</code>	
10	<code>if sy-subrc eq 0.</code>	
11	<code> select * from ysa_vkgp into corresponding fields of table et_entityset</code>	
12	<code> WHERE partner = ls_key-value.</code>	
13	<code>* If there is no carrier specified, read all the flights</code>	
14	<code>else.</code>	
15	<code> select * from ysa_vkgp into corresponding fields of table et_entityset.</code>	
16	<code>endif.</code>	
17		
18	<code>LOOP AT it_filter_select_options INTO ls_filter_select_options.</code>	
19	<code> IF ls_filter_select_options-property EQ 'Vkont'.</code>	
20	<code> LOOP AT ls_filter_select_options-select_options INTO ls_select_option.</code>	
21	<code> move-corresponding ls_select_option to ls_selectparams.</code>	
22	<code> APPEND ls_selectparams TO lt_selectparams.</code>	
23	<code> ENDLOOP.</code>	
24		
25	<code> ENDIF.</code>	
26		
27	<code>ENDLOOP.</code>	
28		
29	<code>DELETE et_entityset where vkont not IN lt_selectparams.</code>	
30		
31	<code>endmethod.</code>	

Quelle: eigener Screenshot

Methode	VERTRAGSKONTOSET_GET_ENTITY	aktiv
1	<code>method VERTRAGSKONTOSET_GET_ENTITY.</code>	
2	<code>data: ls_key like line of it_key_tab.</code>	
3		
4	<code>read table it_key_tab into ls_key with key name = 'Vkont'.</code>	
5	<code>if sy-subrc eq 0.</code>	
6	<code> select * from ysa_vkgp into corresponding fields of er_entity</code>	
7	<code> where vkont = ls_key-value.</code>	
8	<code> endselect.</code>	
9	<code>else.</code>	
10		
11	<code>endif.</code>	
12	<code>endmethod.</code>	

Quelle: eigener Screenshot

G Informationen zur Version des SAPUI5-Frameworks im SGW-System

SAP UI development toolkit for HTML5 from ABAP server

This Application provides a ready-to-consume version of SAPUI5.
The SAPUI5 sap-ui-core.js bootstrap file is available at: [/sap/public/bc/ui5_u15/1/resources/sap-ui-core.js](https://sap/public/bc/ui5_u15/1/resources/sap-ui-core.js)

The following libraries are available:

Library	Version: none	Version: 1.22 (1.22.14)	Version: 1.24 (1.24.8)	Version: 1.26 (1.26.18)	Version: 1.28 (1.28.39)
com.sap.apf.apf-lib	1.22.2	1.24.3	1.26.7	1.28.16	
com.sap.ca.scfld.md	1.22.4	1.24.7	1.26.5	1.28.13	
com.sap.ca.ui	1.22.4	1.24.7	1.26.5	1.28.13	
com.sap.collaboration.collaboration-comm	1.22.3	1.24.3	1.26.5	1.28.10	
com.sap.ip.bi.zen.rt.components.crosstab	15.1.0.201509240501				
com.sap.ip.bi.zen.rt.dsh.phx	15.1.0.201605012045				
com.sap.ip.bi.zen.rt.themelib.phx	15.1.0.201509240501				
com.sap.lm.uiext.landvisz.sap.landvisz		1.24.0	1.26.1	1.28.0	
com.sap.portal.ui5.portal-ui5-ext	1.22.2	1.24.0	1.26.0	1.28.0	
com.sap.smartbusiness.suite.smartbusines	1.22.3	1.24.3	1.26.5	1.28.26	
com.sap.suite.suite-ui-commons	1.22.3	1.24.3	1.26.6	1.28.25	
com.sap.ui5.common	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.comp	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.core	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.dist.fiori-lib	1.22.14	1.24.8	1.26.18	1.28.39	
com.sap.ui5.fl			1.26.16	1.28.38	
com.sap.ui5.inbox	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.layout	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.makit	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.mobile	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.mobile-ext	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.ndc				1.28.38	
com.sap.ui5.richtexteditor	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.suite	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.table	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.themelib_sap_bluecrystal	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.themelib_sap_goldreflection	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.themelib_sap_platinum	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.unified	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.ux3	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ui5.vbm	1.22.6	1.24.2	1.26.6	1.28.8	
com.sap.ui5.viz	1.22.13	1.24.7	1.26.16	1.28.38	
com.sap.ushell.ushell-lib	1.22.9	1.24.7	1.26.10	1.28.35	
com.sap.ushell.ushell_abap	1.28.34				
com.sap.uxap.uxap-uilib			1.26.6	1.28.10	
sap.epm.ui.controls	1.22.2				
sap.epm.ui.controls.mobile	1.16.2				

For more information and documentation look at the SAPUI5 Demokit

[Demokit](#)

[Demokit](#)

[Demokit](#)

[Demokit](#)

You can use SAPUI5 in any HTML file with the following script tag:

```
<script id="sap-ui-bootstrap"
type="text/javascript"
src="/sap/public/bc/ui5_u15/1/resources/sap-ui-cachebuster/sap-ui-core.js"
data-sap-ui-theme="sap_goldreflection"
data-sap-ui-libs="sap.ui.commons">
</script>
```

Quelle: eigener Screenshot

H Anpassen der Anwendungsvorlage

H.1 SAP Web IDE in SAP Cloud Platform – CRUD Master-Detail Application

New CRUD Master-Detail Application

Template Customization

Data Binding - Object	Object Collection*	PersonSet
	Object Collection ID*	Partner
	Object Title*	Partner
	Object Numeric Attribute	
	Object Unit of Measure	
	<input checked="" type="checkbox"/> Display only required fields when creating a new entity.	
Data Binding - Line Item	Line Item Collection	VertragskontoSet
	Line Item Collection ID	Vkont
	Line Item Title	Vkbez
	Line Item Numeric Attribute	
	Line Item Unit of Measure	

[Previous](#) [Next](#) [Finish](#)

Quelle: eigener Screenshot

H.2 SAP Web IDE Personal Edition – Master Master-Detail Application

New SAP Fiori Master Master-Detail Application Template Customization

Master 1 Section	Title	Partner anzeigen
	OData Collection	PersonSet
	Search Placeholder	Geschäftspartner
	Search Tooltip	Bitte geben Sie die Nummer eines Geschäftspartners ein.
	Search Field	Partner
Master 1 Data Fields	Item Title	Partner
	Attribute 1	Attribute from the OData collection
	Attribute 2	Attribute from the OData collection
Master 2 Section	Title	Vertragskonten
	OData Collection Navigation	VertragskontoSet
	Search Placeholder	Vertragskontonummer
	Search Tooltip	Bitte geben Sie die Nummer eines Vertragskontos ein.
	Search Field	Vkont
Master 2 Data Fields	Item Title	Vkont
	Numeric Attribute	Numeric attribute from the target collection
	Unit of Measure	Unit of Measure from the target collection
	Status Attribute	Status attribute from the target collection
	Attribute	Additional attribute from the target collection
Detail Section	Title	Informationen zum Vertragskonto
	Attribute 1	Vkbez
	Attribute 2	Vkont
	Attribute 3	Partner

[Previous](#)[Next](#)[Finish](#)

Quelle: eigener Screenshot

I Ansicht der Anwendung nach der Erstellung

I.1 SAP Web IDE in SAP Cloud Platform – CRUD Master-Detail Application

The screenshot shows a SAP Web IDE interface for a CRUD Master-Detail application. The left sidebar lists four entries under 'PersonSet (4)': '1000010000', '1000020000', '1000030000', and '1000040000'. The main panel displays the details for the selected entry, '1000010000'. The top right of the main panel shows 'PersonSet' and a search bar. Below the search bar are two blue circular icons: one with an 'i' for information and one with a clipboard for copy. The main content area shows a nested 'VertragskontoSet (3)' section with three entries: 'Lausner Weg 43', 'Lausner Weg 44', and 'Lausner Weg 45', each with its corresponding ID below it. At the bottom of the main panel are buttons for 'Edit', 'Delete', and a refresh icon.

Quelle: eigener Screenshot

I.2 SAP Web IDE Personal Edition – Master Master-Detail Application

The screenshot shows a search interface for partners. At the top, a header reads "Partner anzeigen". Below it is a search bar with the placeholder "Geschäftspartner" and a magnifying glass icon. A list of partner numbers is displayed below the search bar:

- 1000010000
- 1000020000
- 1000030000
- 1000040000

Quelle: eigener Screenshot

The screenshot shows a master list of contract accounts on the left and detailed information on the right. The left pane is titled "Vertragskonten" and contains a search bar for "Vertragskontonummer" with a magnifying glass and refresh icon. A list of contract account numbers is shown:

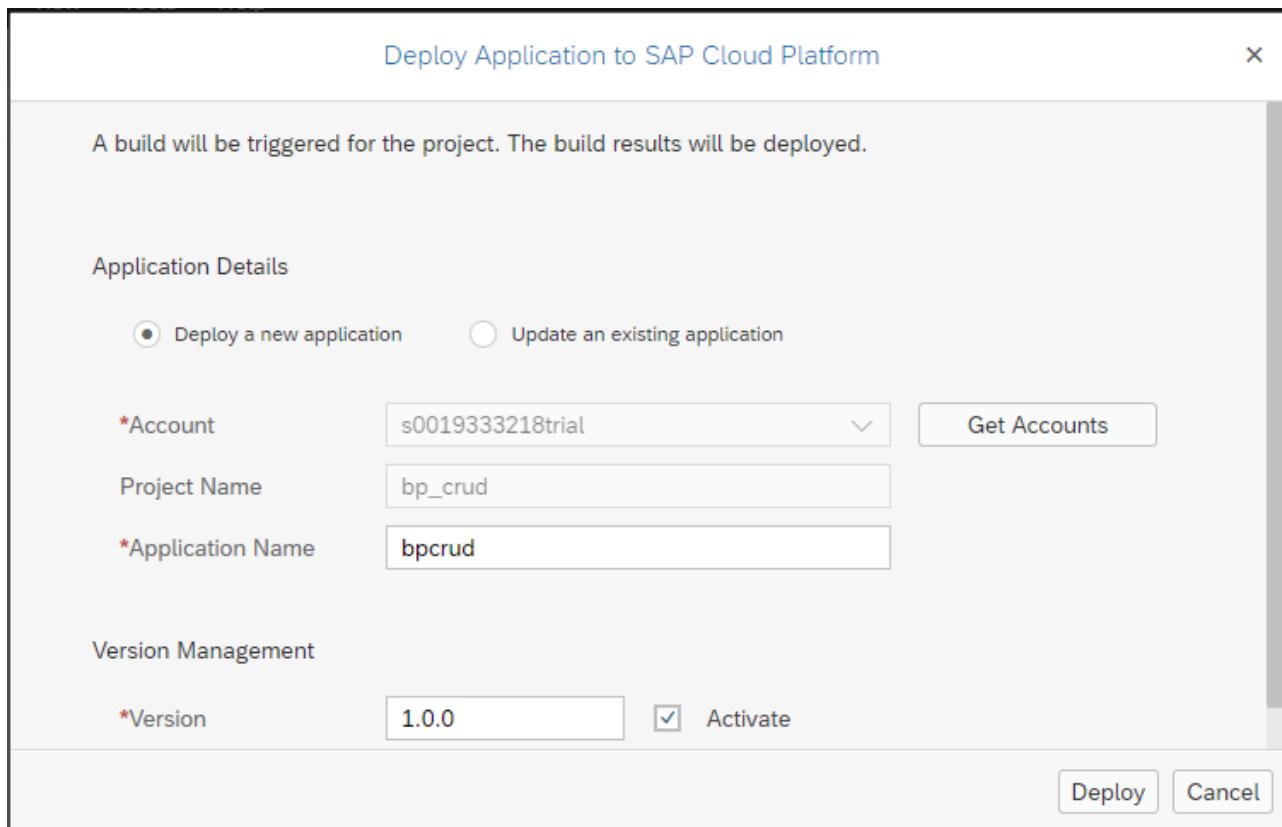
- 100100100100
- 100100100200
- 100100100300

The right pane is titled "Informationen zum Vertragskonto" and displays details for the selected contract account (100100100100). It includes:

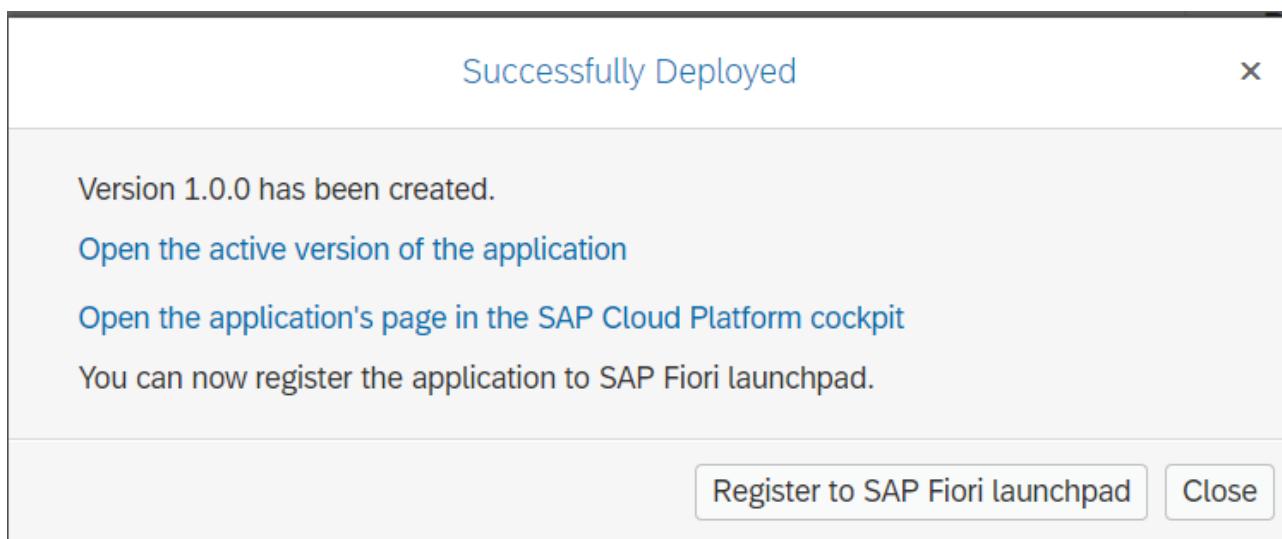
- Vkbez: Lausner Weg 43
- Vkont: 100100100100
- Partner: 1000010000

Quelle: eigener Screenshot

J Ausliefern einer Anwendung in die SAP Cloud Platform



Quelle: eigener Screenshot



Quelle: eigener Screenshot

K Anlegen eines Fiori-Launchpad im Portal-Dienst der SAP Cloud Platform

K.1 Anlage des Katalogs

The screenshot shows the Fiori Configuration Cockpit interface. On the left, a sidebar menu includes 'Startseite', 'Einstellungen', 'Inhaltsverwaltung' (selected), 'Apps', 'Kataloge' (selected), 'Gruppen', 'Rollen', 'Services und Tools', 'App-Ressourcen', and 'Verwendungsanalyse'. The main area displays 'Kataloge (3)' with a search bar. A catalog named 'Angelas Katalog' is selected, showing its details: 'Katalog für Bachelorarbeit'. Below it are 'Sample Catalog' and 'Workflow Catalog'. On the right, the 'Allgemein' tab is active, showing the catalog's name, ID, and description. Under 'Zusatzinformationen', it lists creation and modification details. At the bottom right are 'Bearbeiten' and 'Löschen' buttons.

Quelle: eigener Screenshot

This screenshot shows the 'Angelas Katalog' details page. It features three buttons: 'Eigenschaften' (Properties), 'Apps', and 'Rollen' (Roles). The 'Rollen' button is highlighted with a blue border. Below the buttons, the heading 'Zugeordnete Rollen (1)' is displayed, followed by a table with one row. The table has two columns: 'Rollenname' (Role Name) and 'Rollen-ID' (Role ID). The single entry is 'Portal_User'.

Rollenname	Rollen-ID
Portal_User	Portal_User

Quelle: eigener Screenshot

K.2 Anlage einer Gruppe

Angelas Gruppe

 Eigenschaften  Apps  Rollen

Allgemein

Gruppenname: Angelas Gruppe

Gruppen-ID: 4e70ebd8-b221-40c6-8226-363628ddf37f-1528903015105

Gesperre Gruppe

Zusatzinformationen

Angelegt: 17:19 13.6.2018

Angelegt von: s0019333218

Zuletzt geändert: 17:21 13.6.2018

Zuletzt geändert von: s0019333218

Quelle: eigener Screenshot

Angelas Gruppe

 Eigenschaften  Apps  Rollen

Zugeordnete Rollen (1)

Rollenname	Rollen-ID
Portal_User	Portal_User

Quelle: eigener Screenshot

L Registrierung einer Anwendung auf einem Launchpad in der Cloud

L.1 Vergabe der Anwendungsbezeichnung

The screenshot shows the 'General Information' step of a registration process titled 'Register to SAP Fiori Launchpad'. The form includes fields for Provider Account (selected: trial (flpportal)), Application Name (angela.bp_crud), Description (Geschäftspartner anzeigen), and Intent (two entries: Semantic Object and bp_crud both checked). Navigation buttons 'Previous' and 'Next' are at the bottom.

Intent	Semantic Object	Action
<input checked="" type="checkbox"/>	Semantic Object	Action
<input checked="" type="checkbox"/>	bp_crud	Display

Quelle: eigener Screenshot

L.2 Konfiguration der Anwendungskachel

The screenshot shows the 'Tile Configuration' step of the registration process. It includes fields for Type (Dynamic), Title (Geschäftspartner anzeigen), Subtitle (empty), Icon (sap-icon://person-placeholder), and Collection (PersonSet). To the right, a preview window shows a tile with the title 'Geschäftspartner anzeigen' and a small icon, with the number '4' indicating the count of items in the collection.

Quelle: eigener Screenshot

L.3 Auswahl von Launchpad, Katalog und Gruppe

The screenshot shows a four-step navigation bar at the top: General Information, Tile Configuration, Assignment (highlighted in blue), and Confirmation. Below the bar, the title "Register to SAP Fiori Launchpad" and the section "Assignment" are displayed. Three dropdown menus are shown: *Site (Angelas Launchpad), *Catalog (Angelas Katalog), and *Group (Angelas Gruppe). At the bottom are buttons for Previous, Next, and Finish.

General Information Tile Configuration **Assignment** Confirmation

Register to SAP Fiori Launchpad

Assignment

*Site: Angelas Launchpad

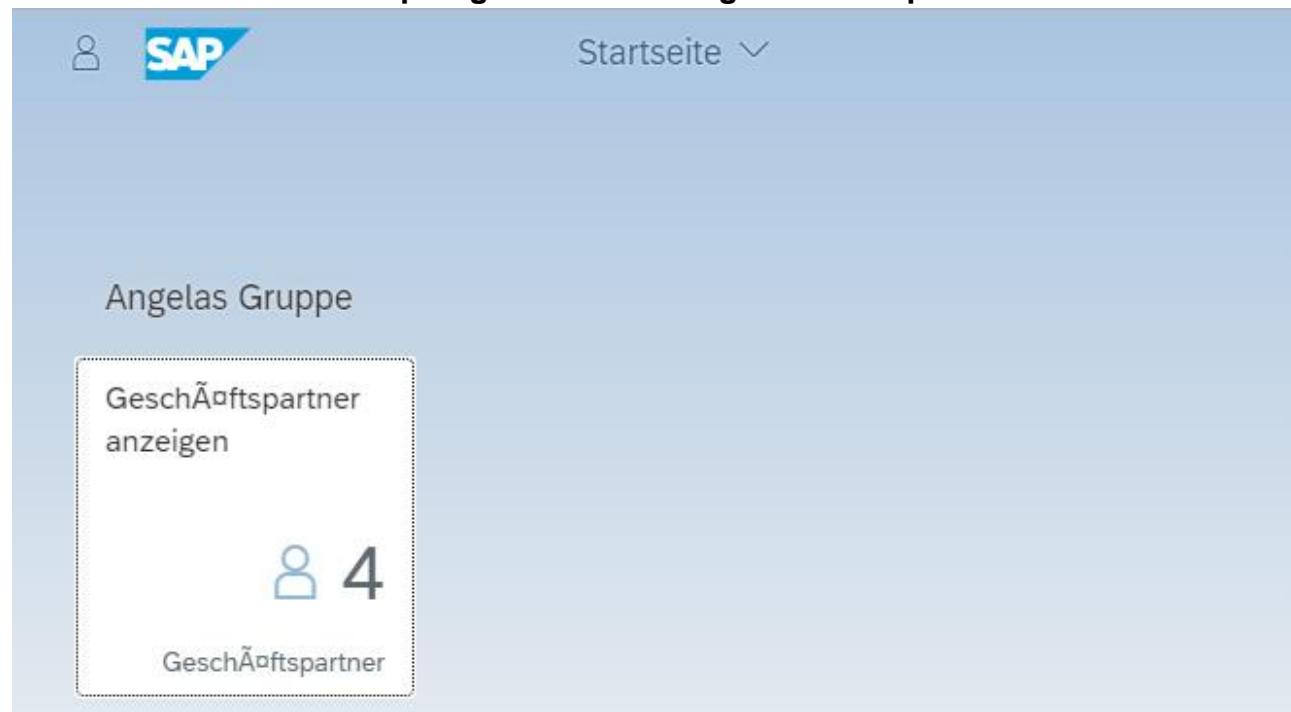
*Catalog: Angelas Katalog

*Group: Angelas Gruppe

Previous Next Finish

Quelle: eigener Screenshot

L.4 Ansicht der Verknüpfung zur Anwendung im Launchpad



Quelle: eigener Screenshot

M Ausliefern einer Anwendung auf einen lokalen Applikationsserver

Deployment Options Deploy a New Application Select a Transport Request Confirmation

Deploy to SAPUI5 ABAP Repository

Deployment Options

System * SGW-destination

Deploy a new application
 Update an existing application

i The SAPUI5 versions of your application and the selected SAP system are compatible.
Application:1.28.40
SAP system:1.28.38

Previous Next

Quelle: eigener Screenshot

Deployment Options Deploy a New Application Select a Transport Request Confirmation

Deploy to SAPUI5 ABAP Repository

Deploy a New Application

Name * ② ysa_bp_mmd

Description * Geschäftspartner anzeigen

Package * ② YSA_ABAP_SAPUI5_REPOSITORY Browse

Previous Next

Quelle: eigener Screenshot

Deployment Options Deploy a New Application **Select a Transport Request** Confirmation

Deploy to SAPUI5 ABAP Repository

Select a Transport Request

Choose from requests in which I am involved

Request Number	SGWK900181
----------------	------------

Transport Request	User	Target	Description
SGWK900181	STOECKAN	/ZSGW/	Auftrag Bachelorarbeit Angela

Create a new request

Request Description

The Request ID is automatically generated

Enter a request number

Request Number

[Previous](#) [Next](#)

Quelle: eigener Screenshot

N Registrieren einer Anwendung auf einem lokalen Fiori-Launchpad

N.1 Ansicht Anwendungsspeicher auf Applikationsserver

The screenshot shows the SAP Fiori Launchpad interface. At the top, a dropdown menu is set to 'Paket' and displays 'YSA_ABAP_SAPUI5_REPOSITORY'. Below the menu is a toolbar with various icons. The main area is a table titled 'Objektname' (Object Name) and 'Beschreibung' (Description). The table lists the following entries:

Objektname	Beschreibung
YSA_ABAP_SAPUI5_REPOSITORY	eigenes Paket als ABAP repository
BSP-Bibliothek	
BSP-Applikationen	
YSA_BP	Geschäftspartner anzeigen Angela
YSA_BP_MMD	Geschäftspartner anzeigen

Quelle: eigener Screenshot

N.2 Semantisches Objekt anlegen

Sicht "Semantische Objekte - Kundenansicht" anzeigen: Übersicht

The screenshot shows the SAP Fiori Launchpad interface. At the top, there is a toolbar with icons. Below it is a section titled 'Semantische Objekte - Kundenansicht'. A table lists semantic objects:

Semantisches Objekt	Name semantisches Objekt	Beschreibung semantisches Objekt
SEMOBJ_YSA_BP	SEMOBJ_YSA_BP	Semantic Object for YSA_BP
SEMOBJ_YSA_BP_MMD	SEMOBJ_YSA_BP_MMD	Semantic Object for YSA_BP_MMD

Quelle: eigener Screenshot

N.3 Benutzerrolle mit Zuordnung von Katalog, Gruppe und Nutzer

The screenshot shows the SAP Fiori Launchpad interface. At the top, a form is filled with the following data:

Rolle	YSAROLE	<input type="checkbox"/> Veraltet
Kurzbeschreibung	Angelas Rolle	
Zielsystem		<input checked="" type="checkbox"/> Keine Destination

Below the form is a navigation bar with tabs: 'Beschreibung', 'Menü', 'Berechtigungen', 'Benutzer', and 'Personalisierung'. The 'Berechtigungen' tab is selected. On the left, a 'Hierarchie' tree shows 'Menü der Rolle' with 'Angelas Katalog' and 'Angelas Gruppe'. On the right, 'Knotendetails' are listed:

Type	SAP Fiori Kachelkatalog
Objekt	X-SAP-UI2-CATALOGPAGE:YSA_KATALOG?AUTH_DEFAULTS=
Text	Angelas Katalog

Quelle: eigener Screenshot

The screenshot shows the SAP Fiori Launchpad interface. At the top, a table titled 'Benutzerzuordnungen' is displayed:

Benutzerkennung	Benutzername	von	bis	I...
STOECKAN	Angela Stöckert	23.05.2018	31.12.9999	

Quelle: eigener Screenshot

N.4 Konfiguration des Launchpad im Applikationsserver

Launchpad ändern - Rolle: YSALPD, Instanz: TRANSACTIONAL (DE)

The screenshot shows the SAP Fiori Launchpad configuration interface. On the left, there's a tree view under 'Kunden-Launchpad' with 'Angelas Launchpad' expanded, showing 'Inaktive Anwendungen' and 'Geschäftspartner anzeigen'. The main area contains several tabs for configuration:

- Link-Informationen:** Link-Text: Geschäftspartner anzeigen; Beschreibung: (empty)
- Anwendungstyp:** Anwendungstyp: URL (with icon)
- Anwendungsparameter:** URL: /sap/bc/ui5_ui5/sap/ysa_bp_mmd
- System:** Systemalias: (empty)
- Erweiterte (optionale) Parameter ausblenden**
- Erweiterte Parameter (optional):** Anwendung - Deaktivierung durch Benutzer: Anwendung kann nicht aus LPD entfernt werden (checkbox)
- Anwendungsspezifische Parameter:** Anwendungsalias: geschaeftpartner_anzeigen (with icon); Value: geschaeftpartner_anzeigen
- Zeilanwendungsparam.**: (empty)
- Aussetzen/fortsetzen**: (checkbox)
- Proxy-Klasse**: (empty)
- Zusatzinformationen**: SAPUI5.Component=angela
- FPM-Ereignis-ID**: (empty)
- Portalparameter:** Navigationsmodus: EXT_HEAD Portal-Fenster ohne Ko... (with icon); Fenstereigenschaften: (with icon); Historiemonodus: 1 Navigationseintrag kann in Histori... (with icon); Parameterweiterleitung: G Parameter holen (with icon)

Quelle: eigener Screenshot

N.5 Kachelkatalog mit Zielverknüpfungen im Launchpad Designer

The screenshot shows the SAP Launchpad Designer interface. On the left, there's a sidebar with 'Catalogs' and 'Groups' tabs, and a search bar for catalogs. Below it is a list of catalog entries:

- Angelas Katalog (YSA_KATALOG) - 6 items
- Außendienstmitarbeiter (SD) - 11 items (SAP_SD_BC_FIELDSALESREP_X1)
- Buyer (MM) - Content - 4 items

The main area displays the details for 'Angelas Katalog'. It shows the ID: X-SAP-UI2-CATALOGPAGE:YSA_KATALOG and a summary: 'Tiles 4', 'Tiles 4', and 'Target Mapp... 2'. Below this is a table with columns: Icon, Title, Semantic Object, Action, Parameters, Target URL, Reference, and Outd... . The table contains four rows, each representing a tile in the catalog.

Icon	Title	Semantic Object	Action	Parameters	Target URL	Reference	Outd...
	Geschäftspartner anzeigen	SEMOBJ_YSA_BP_MMD	display		#SEMOBJ_YSA_BP_MMD-display		
	Geschäftspartner anzeigen	SEMOBJ_YSA_BP	display		#SEMOBJ_YSA_BP-display		
	Geschäftspartner anzeigen	SEMOBJ_YSA_BP	display		#SEMOBJ_YSA_BP-display		
	Geschäftspartner anzeigen	SEMOBJ_YSA_BP_MMD	display		#SEMOBJ_YSA_BP_MMD-display		

At the bottom, there are buttons for Create, Configure, Create Reference, Delete, and Where-Used.

Quelle: eigener Screenshot

N.6 Gruppe mit Anwendungskacheln im Launchpad Designer

The screenshot shows the SAP Launchpad Designer interface. On the left, there's a sidebar with 'Catalogs' and 'Groups' tabs, and a search bar for groups. Below it is a list of group entries:

- Angelas Gruppe (YSA_GROUP) - 4 items
- arvato Systems Applikationen für Z_GROUP_ISU_APPS - 3 items
- arvato Systems Vertriebsmitarbeiter TEST_VERTREIB_01 - 5 items
- Buyer (MM) (SAP_MM_BCG_BUYER) - 3 items
- Buyer (MM) (SAP_MM_BCG_Buyer_X1) - 5 items
- Cost Manager (SAP_FIN_BCG_CostManager_X1) - 1 item

The main area displays the details for 'Angelas Gruppe'. It shows the ID: YSA_GROUP and a summary: 'Geschäftspartner anzeigen' (4 items). Below this are four application tiles arranged in a 2x2 grid:

Geschäftspartner anzeigen	Geschäftspartner anzeigen
1,23K Geschäftspartner	1,23K Geschäftspartner
Geschäftspartner anzeigen alte Version	Geschäftspartner anzeigen alte Version
1,23K Geschäftspartner	1,23K Geschäftspartner

At the bottom, there's a large '+' button.

Quelle: eigener Screenshot

9 Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form weder veröffentlicht, noch einer anderen Prüfungsbehörde vorgelegt.

Leipzig, 12.07.2018

Ort, Datum

Unterschrift