

Schlußbericht Einzelvorhaben SEGNO

Februar 2023

1.1 Förderzeichen

- 03SX486B – SEGNO Industrie Automation GmbH

1.2 Förderprogramm

- Maritimes Forschungsprogramm
- MARITIME.green - Umweltschonende maritime Technologien
- MARITIME.smart - Maritime Digitalisierung und smarte Technologien

1.3 Verbundvorhaben

OSCAR : Online Modeling, Simulation and Remote Control System for environmental technologies on-board (Cruise) Ships

1.4 Versionen

Revisionsstand	Änderung / Erweiterung	Datum / Zeichen
Version 1.0	Dokumentation	2021-07-28 / RBN
Version 1.1	Dokumentation	2021-10-18 / ALE
Version 1.2	Dokumentation	2022-10-12 / CNS
Version 1.3	Dokumentation	2022-11-23 / ALE
Version 1.4	Dokumentation	2022-12-06 / CNS
Version 1.5	Dokumentation	2023-01-05 / ALE
Schlussbericht	Dokumentation	2023-02-07 / CNS

1.5 Verfasser

SEGNO Industrie Automation GmbH	Christian Niclas / CNS
SEGNO Industrie Automation GmbH	Andreas Lüdtke / ALE
SEGNO Industrie Automation GmbH	Silja Mendes / SMS
SEGNO Industrie Automation GmbH	Ralf Baxmann / RBN
SEGNO Industrie Automation GmbH	Thorsten Arendt / TAT
SEGNO Industrie Automation GmbH	Jan Schütte / JSE

1.6 Inhaltsverzeichnis

1.1	Förderzeichen.....	1
1.2	Förderprogramm	1
1.3	Verbundvorhaben	1
1.4	Versionen.....	1
1.5	Verfasser.....	1
1.6	Inhaltsverzeichnis.....	2
2	Kurzdarstellung des Einzelvorhabens.....	8
2.1	System-Konzept OSCAR – Kommunikationswege.....	9
2.2	OSCAR Arbeitsplan und Arbeitspakete.....	9
2.3	OSCAR – SEGNO Ziele im Verbundprojekt	10
2.3.1	Ziel 1: Entwicklung IoT-Connector mit Übertragbarkeiten	10
2.3.2	Ziel 2: Entwicklung Connector digitaler Zwilling	11
2.3.3	Ziel 3: Industrielle Cloud-Anwendungen und Know-how Aufbau	11
3	Anforderungen an Cloud-Systeme	13
3.1	Software as a Service (SaaS).....	13
3.2	Platform as a Service (PaaS)	13
3.3	Infrastructure as a Service (IaaS).....	13
3.4	IT Security	13
4	OSCAR-IoT-Connector	14
4.1	Funktionsanforderungen.....	14
4.2	Konzept.....	14
4.3	Umsetzung.....	15
4.3.1	Entwicklungsumgebung	15
4.3.2	Verwendete Frameworks	15
4.3.3	Verwendete Bibliotheken.....	16
4.3.4	Datenbank	16
4.3.5	Funktionsmodule.....	16
4.3.6	Entwicklung und Tests.....	18
4.3.7	Veröffentlichung.....	20
4.4	Einrichtung	21
4.4.1	Einrichtung Betriebssystem.....	21
4.4.2	Zugriff auf den Unipi.....	22
4.4.3	System-Update	22
4.4.4	Einrichtung des Systems.....	23
4.4.5	Einrichtung via raspi-config	23
4.4.6	Einrichtung IoT-Connector	25
4.4.7	Autostart.....	26

4.5 Konfiguration IoT-Connector	26
4.5.1 Konfiguration der Konnektoren.....	27
4.5.2 Konfiguration Messwerte	28
4.5.3 Konfiguration Mindsphere-Konnektor	30
4.5.4 Konfiguration Sollwert-Übernahme	30
4.6 Entwicklung, Test und Übertragbarkeiten	31
4.6.1 Test der Übertragbarkeit auf verschiedene Kommunikationshardware	32
4.6.2 Unipi Neuron S103-G.....	32
4.6.3 Phoenix Contact BL2 BPC 1501E-64-W10	33
4.6.4 WAGO 752-9401 (Edge Computer)	33
4.6.5 Siemens IOT2050.....	34
5 Konfiguration der Siemens Mindsphere	35
5.1 Asset-Konfiguration.....	35
5.2 Asset-Types.....	36
5.3 Aspects	37
5.4 Agenten	37
5.5 Fehleranalyse / Logging.....	41
6 OSCAR Connect digitaler Zwilling	44
6.1 Funktionsanforderungen.....	44
6.2 ifakFAST	44
6.3 Übertragbarkeit in eine Clouddumgebung.....	44
7 OSCAR BigData	46
7.1 Funktionsanforderungen.....	46
7.2 Big Data	46
7.3 Konzept.....	46
7.4 Umsetzung.....	46
8 OSCAR Analysis.....	48
8.1 Funktionsanforderungen.....	48
8.2 Konzept.....	48
8.3 Aufbau	48
8.3.1 Layout	49
8.3.2 Mindsphere OS-Bar	49
8.3.3 Asset-Auswahl / Flotten-Management	49
8.3.4 Asset-Ansicht	49
8.3.5 Sollwerte.....	53
8.4 Umsetzung.....	54
8.5 Lokale Entwicklung	55
8.6 Erstellung des OSCAR-Analysis Projekt	55

8.6.1	Framework - Angular.....	55
8.6.2	Erstellung des Projekts	56
8.6.3	Applikation lokal starten	57
8.6.4	Implementierung von Mindsphere Web Komponenten.....	58
8.7	Zugriffsrechte	59
8.7.1	Scopes & Roles	60
8.8	Benötigte Rechte	60
8.8.1	Application Credentials	61
8.9	Bereitstellen einer Applikation via Cloud-Foundry	62
8.9.1	Installation	63
8.9.2	Login	63
8.9.3	Benutzer / Benutzerrechte	65
8.9.4	Deployen einer Applikation in die Mindsphere.....	66
8.9.5	Upload der Applikation in die Siemens Mindsphere.....	66
8.9.6	Builden der Applikation.....	67
8.9.7	Upload in die Mindsphere.....	67
8.10	Konfiguration der Applikation innerhalb der Mindsphere.....	68
8.10.1	Zuweisung der Applikation für andere Nutzer	71
8.11	Richtlinien für Mindsphere-Applikationen.....	72
8.12	Veröffentlichung der Applikation im Operator-Tenant.....	73
8.13	Untersuchung geplanter Cloud-Funktionalitäten	75
8.13.1	Implementierung eines dynamischen Prozessbildes	75
8.13.2	Melde- und Benachrichtigungssystem per E-Mail.....	75
9	Untersuchung OSCAR-Connect für andere Cloud-Systeme	76
9.1	Untersuchung Übertragbarkeit IOT-Connector.....	76
9.2	Untersuchung Übertragbarkeit OSCAR-Analysis.....	77
9.3	Untersuchung Übertragbarkeit des digitalen Zwillings in verschiedene Cloudumgebungen	78
10	Literaturangaben	79

Übersicht Abbildungen

Abbildung 1: System-Konzept OSCAR	8
Abbildung 2: System-Konzept OSCAR - SEGNO Anteile	9
Abbildung 3: OSCAR Arbeitsplan	9
Abbildung 4: IoT-Connector Funktionskonzept.....	15
Abbildung 5: IoT-Connector im Einsatz innerhalb der Laboranlage.....	19
Abbildung 6: Quelle: Darstellung eines Unipi Neuron S103-G	21
Abbildung 7: Konfiguration - raspi-config Hauptmenü	24
Abbildung 8: Konfiguration raspi-config - hostname	24
Abbildung 9: Konfiguration raspi-config SSID	25
Abbildung 10: Konfiguration raspi-config - passphrase	25
Abbildung 11: Syntax für nodeS7 Adressen	29
Abbildung 12: Darstellung eines Unipi Neuron S103-G	32
Abbildung 13: Darstellung eines Phoenix Contact BL2 BPC	33
Abbildung 14: Darstellung eines WAGO 752-9401	33
Abbildung 15: Darstellung eines Siemens IOT2050	34
Abbildung 16: Übersichtsseite Mindsphere	35
Abbildung 17: Mindsphere AssetManager - Asset Konfiguration.....	36
Abbildung 18: Mindsphere AssetManager - Aspect Konfiguration.....	37
Abbildung 19: Mindsphere AssetManager- Hauptansicht	38
Abbildung 20: Mindsphere Netzwerk-Verwaltung.....	38
Abbildung 21: Mindsphere MindConnectLib Hauptansicht	39
Abbildung 22: Mindsphere MindConnectLib Konfiguration	40
Abbildung 23: Mindsphere Hauptansicht Agent Diagnostic	41
Abbildung 24: Mindsphere Agent Diagnostic.....	42
Abbildung 25: Mindsphere Agent Diagnostic Log	43
Abbildung 26:Mindsphere Service-Übersicht	47
Abbildung 27: Mindsphere Operator - IoT Value Plan	48
Abbildung 28: OSCAR-Analysis - Layout	49
Abbildung 29: OSCAR-Analysis - TimeSeries	50
Abbildung 30: OSCAR Analysis - Events.....	51
Abbildung 31: Anlegen eines neuen Angular-Projekts.....	56
Abbildung 32: Grundstruktur Angular Projekt	56
Abbildung 33: Mindsphere OS-Bar	57
Abbildung 34: Einbinden der OS Bar in eine Webseite	57
Abbildung 35: Ansicht eines blanken Angular Projekts.....	58
Abbildung 36: OSCAR Analysis auf der Mindsphere Übersichtsseite.....	59
Abbildung 37: Benötigte Rechte für OSCAR Analysis	60
Abbildung 38: Mindsphere – Generieren der Application Credentials	62
Abbildung 39: Mindsphere - Auswahl der Zugriffsrechte einer Applikation.....	62
Abbildung 40: Mindsphere - Application Credentials	62
Abbildung 41: CloudFoundry Login per single-sign-on	63
Abbildung 42: Mindsphere Login Token Abfrage	64
Abbildung 43: CloudFoundry - Temporary Authentication Code.....	64
Abbildung 44: CloudFoundry - Organisationauswahl nach Login	64
Abbildung 45: CloudFoundry - Anlegen eines Space.....	65
Abbildung 46: CloudFoundry - Festlegen eines Target	65
Abbildung 47: CloudFoundry - Anzeigen der Nutzer einer Organisation.....	65
Abbildung 48: CloudFoundry - Vergabe von Benutzerrechten innerhalb einer Organisation	66
Abbildung 49: CloudFoundry - Setzen von Benutzerrechten innerhalb einer Organisation.....	66

Abbildung 50: CloudFoundry - Dokumentation von Benutzerrechten innerhalb eines Space	66
Abbildung 51: CloudFoundry - Setzen von Benutzerrechten innerhalb eines Space.....	66
Abbildung 52: Manifestdatei innerhalb eines Angular Projekts	67
Abbildung 53: Upload einer Applikation in CloudFoundry.....	68
Abbildung 54: Mindsphere - Anlegen einer neuen Applikation.....	68
Abbildung 55: Mindsphere - Konfiguration einer Applikation.....	69
Abbildung 56: Mindsphere - Eintragen der CloudFoundry Direct URL	69
Abbildung 57: Mindsphere - Endpoint hinzufügen	70
Abbildung 58: Mindsphere - Konfiguration der Rollen	70
Abbildung 59: Mindsphere - OSCAR Analysis auf der Übersichtsseite	71
Abbildung 60: Vergabe von Rollen einer Applikation	71
Abbildung 61: Mindsphere - Rollenzuweisung.....	72
Abbildung 62: Mindsphere - Veröffentlichen einer Applikation im Operator Tenant.....	73
Abbildung 63: Mindsphere - eingetragene Applikation	74
Abbildung 64: CloudFoundry - Anzeige der Information einer Applikation.....	74

Übersicht Tabellen

Tabelle 1: Big Endian	30
Tabelle 2: Little Endian	30
Tabelle 3: Mid-Big Endian.....	30
Tabelle 4: Mid-Little Endian	30

2 Kurzdarstellung des Einzelvorhabens

Mit der Konzipierung und Entwicklung einer standardisierten Daten-Schnittstelle (OSCAR-Connect / IoT) für Umweltanlagen auf (Kreuzfahrt)-Schiffen legen wir mit diesem Vorhaben eine Grundlage für umwelttechnischen Anlagen auf (Kreuzfahrt)-Schiffen sowie für die Übertragbarkeit auf andere Betriebsanlagen (Biogasanlagen, Abgasreinigungsanlagen, unterschiedliche verfahrenstechnische Abwasserreinigungsanlagen, industrielle Produktionsanlagen, etc..).

Die technologische Kombination der Datenerfassung, der sicheren und langfristigen Datenzwischenspeicherung sowie des gesicherten und robusten Transfers der Daten über das Internet in verschiedene Cloud-Umgebungen in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe ist uns in dieser Form unbekannt am Markt.

Mit der Einarbeitung und Umsetzung von speziellen umweltbezogenen Cloud-Lösungen in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe werden wir das Know-how sowie das Portfolio der SEGNO erweitern. Moderne WEB-Entwicklungen und Cloud-Apps von speziellen branchenbezogenen Umwelt-Daten (OSCAR-Analysis) in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe sind der Anfang einer ganzen Produktgruppe von Umwelt-Apps in Verbindung mit Cloud-Lösungen.

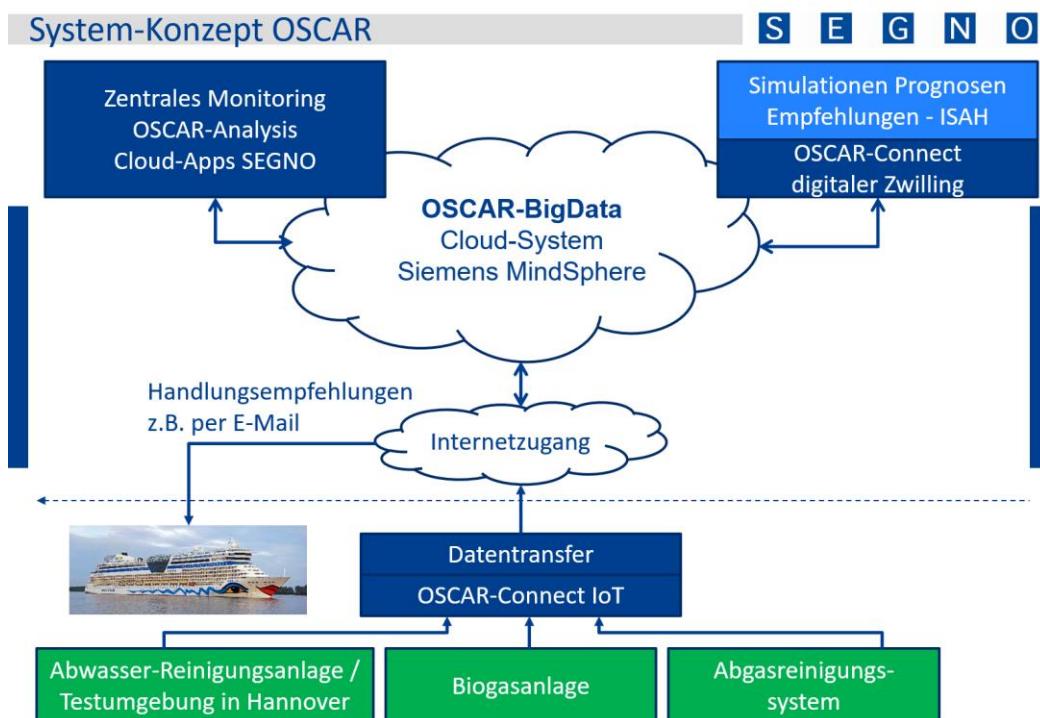


Abbildung 1: System-Konzept OSCAR

Für Simulationen, verfahrenstechnische Prognosen und daraus resultierende Empfehlungen für die Optimierung von Prozessparametern oder Anweisungen für Fahrbefehle von Umweltanlagen wird eine Kommunikationsschnittstelle (OSCAR-Connect / digitaler Zwilling) entwickelt für einen digitalen Zwilling, der außerhalb der Cloud-Umgebung operiert. Konzeptionell soll untersucht werden, ob der digitale Zwilling in einer standardisierten Form übertragbar als Softwaredienst in der Cloud-Umgebung lauffähig ist. Dazu soll in Kombination eine App für den digitalen Zwilling entwickelt werden. Für die Sicherstellung des reibungslosen Datenverkehrs ist eine Verbindungsüberwachung der Kommunikationswege, die Messwertüberwachung der Prozesswerte sowie eine Fernalarmierung für Handlungsempfehlungen vorgesehen.

2.1 System-Konzept OSCAR – Kommunikationswege

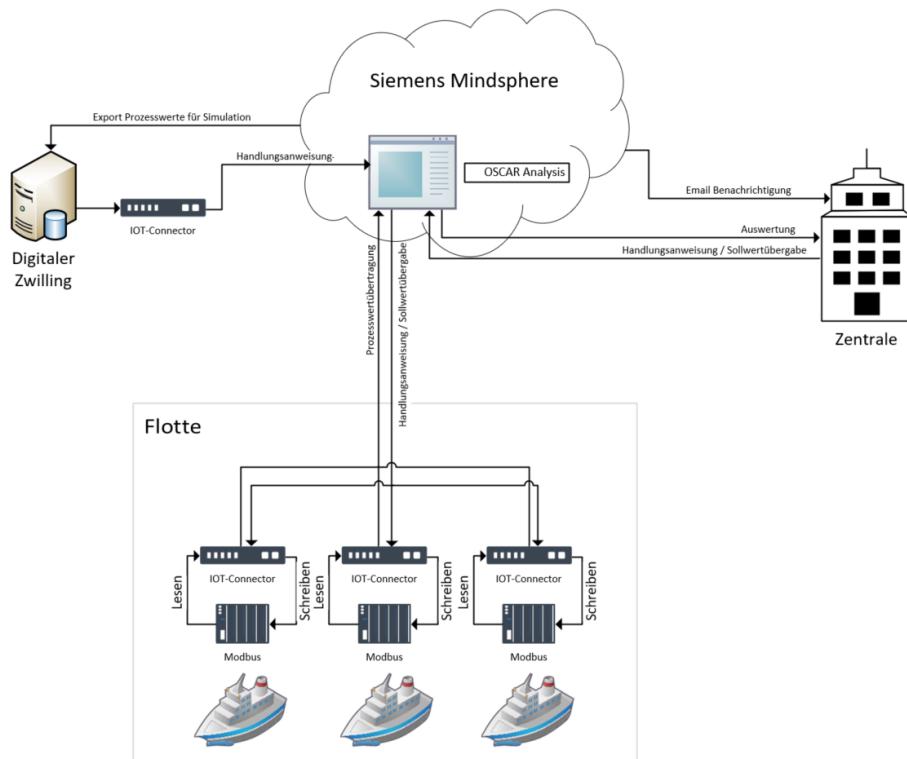


Abbildung 2: System-Konzept OSCAR - SEGNO Anteile

2.2 OSCAR Arbeitsplan und Arbeitspakete

Arbeitspakete	Quartale	2019				2020				2021				2022			
		3	4	1	2	3	4	1	2	3	4	1	2	3	4		
1 Erfassung der Ist-Situation an Bord von Kreuzfahrtschiffen (CMG)																	
1.1 Datenaufnahme und Infrastruktur (CMG)																	
1.2 Identification of material flows and biological processes (LUH)																	
2 Einrichtung PLS-Infrastruktur und Entwicklung des Prozesseitsystems (SEGNO)																	
2.1 Spezifikations- und Entwicklungsphase (OSCAR-Connect)																	
2.2 Datenaufbereitung und Vorhaltung im industriellen Cloud-System (OSCAR-Big Data)																	
2.3 Monitoring und Visualisierung der Umweltdaten in Cloud-App's (OSCAR-Analysis)																	
3 Aufbau und Entwicklung Digitaler Zwilling (LUH)																	
3.1 Aufbau/Inbetriebnahme Schiffskläranlage und Entwurf des zugehörigen digitalen Zwillings																	
3.2 Kalibrierung und Validierung des digitalen Zwillings anhand der Schiffskläranlage																	
3.3 Versuchphase zur Identifizierung geeigneter Überwachungsparameter																	
4 Entwicklung verfahrensspezifische Kontroll- und Betriebsstrategien (LUH)																	
4.1 Identifizierung von Leistungsindikatoren zur Prozesssteuerung																	
4.2 Entwicklung von (standard) Betriebsstrategien																	
4.3 Konzeptionelle Entwicklung eines 4-Stufen-Assistenzsystems zur Anlagensteuerung																	
5 Erprobung des SMARTEN Prozesseitsystems (CMG, SEGNO, LUH)																	
Meilensteine																	
1.	Es konnte eine Schnittstelle zum Abruf der umwelttechnischen Daten sowie eine rudimentäre Datenbank zur optimierten Datenarchivierung eingerichtet werden. Weiterhin liegen Daten in ausreichender Menge und Qualität, die erlauben spezifische Lastfälle für die digitalen Zwillinge aufzustellen.																
2.	Die Versuchphase der Schiffskläranlage an Land hat es ermöglicht einen digitalen Zwilling der Abwasserreinigung zu erstellen und konnte erfolgreich mit dem digitalen Biogas-Zwilling zu einem Umweltsystem verbunden werden. Ferner wurden erste optimierte Betriebs- und Steuerungskonzepte erarbeitet und im Rahmen von Simulationsdurchgängen erprobt, um an Bord im Normalbetrieb getestet zu werden.																
3.	3. und 4. Quartal 2022 kostenneutrale Projektverlängerung bedingt durch Projektverzögerungen seitens der Corona Pandemie																

Abbildung 3: OSCAR Arbeitsplan

Die einzelnen Arbeitspakete aus dem Arbeitsplan wurden durch SEGNO gemeinsam mit den Partnern durchgeführt. Die Vorhabenziele konnten, auch wegen der kostenneutralen Projektverlängerung im Jahre 2022 (Quartale 3 und 4), eingehalten und erfolgreich umgesetzt werden.

2.3 OSCAR – SEGNO Ziele im Verbundprojekt

Die im folgenden aufgeführten Ziele auch SMARTe Ziele genannt wurde im Laufe des Projekts zu berücksichtigt bzw. erfüllt.

SMART bedeutet:

- Specific (genau beschrieben)
- Measurable (messbar)
- Attainable (erreichbar)
- Relevant (wichtig)
- Timed (zeitlich bestimmt)

2.3.1 Ziel 1: Entwicklung IoT-Connector mit Übertragbarkeiten

Industrielle Cloud-Lösungen gelten als Türöffner für das Thema Industrie 4.0. Mit der geeigneten Infrastruktur können digitale Produktionen in Verbindung mit dem Internet der Dinge zukunftsfähig werden. Die Migration industrieller Daten in Cloud-Systeme soll zu keinen Datenverlusten und zu keinen längeren Produktions-Unterbrechungen führen. Die Auswahl der geeigneten Cloudumgebung im Vorfeld ist daher von entscheidender Bedeutung.

Konzeptionell haben wir eine Standard IoT-Connector-Lösung für die Siemens Mindsphere Cloudumgebung entwickelt, die neben der sicheren und langfristigen Datenspeicherung, Datenvorverarbeitung von umwelttechnischen Informationen sowie der Datenübertragung auch für andere Anwendungsfälle, wie z.B.: Biogasanlagen, Abgasreinigungsanlagen, Produktionsanlagen, etc., einsetzbar ist.

Zusätzlich ist die IoT-Connector-Software auf verschiedene Kommunikationshardware von marktführenden Unternehmen (Siemens, PhoenixContact und Wago) getestet und erprobt worden. Des Weiteren sollen die IoT-Connector-Lösungen mit verschiedenen Cloud-Umgebungen, z.B.: Siemens Mindsphere, Microsoft AZURE oder GE Predix, getestet werden. Diese übertragbare Vielfalt und Kombination einer IoT-Connector-Software ist so nicht am Markt erhältlich und wird sich positiv auf die Markteinführung auswirken.

Als SMARTe-Ziele wurde definiert:

SMARTes Vorhabenziele 1 Projektpartner SEGNO	<p>OSCAR Connect / IoT</p> <ul style="list-style-type: none"> - Entwicklung eines IoT Connectors zur sicheren Datenübertragung in ein industrielles Cloudsystem - Entwicklung der Datenkommunikation mit direkter Anbindung von Messtechnik zur Datenaufbereitung für den digitalen Zwilling - Entwicklung der lokalen Datenzwischenspeicherung und intelligenten Datenvorverarbeitung (Edge, Plausibilisierung) - Entwicklung einer Möglichkeit zur Übernahme von Daten aus einer industriellen Cloud-Umgebung - Entwicklung von Konzepten zur Übertragbarkeit von weiteren Cloud-Systemen und weiteren IoT Hardware-Systemen
---	--

	Eine erste Lösung wird geplant bis zum Start des Demonstrators. Verbesserungen und Anpassungen werden bis zum Vorhabensende durchgeführt.
--	---

2.3.2 Ziel 2: Entwicklung Connector digitaler Zwilling

Ein weiteres Ziel im Teilvorhaben SEGNO ist die Entwicklung einer Connector-Lösung für eine digitalen Zwilling im Umfeld Umwelttechnik. Dazu gehören sowohl die Datenübermittlung prozessrelevanter Umweltdaten von der industriellen Cloud in den digitalen Zwilling als auch umgekehrt die Rückführung von Prozessergebnissen für Handlungsempfehlungen, z.B. per E-Mail. Geplant und konzeptionell angedacht ist eine Konfigurations-Umgebung, welche die Parametrierung der Connector-Schnittstelle in Richtung Standardisierung und Übertragbarkeit auf andere Anwendungsfälle ermöglichen soll.

Im Vorhaben soll untersucht werden, ob der digitale Zwilling in einer standardisierten Form übertragbar als Softwaredienst in der Cloud-Umgebung lauffähig ist. Dazu soll in Kombination eine App für den digitalen Zwilling entwickelt werden. Ebenso soll getestet werden, inwieweit die entwickelte Connector-Lösung für einen digitalen Zwilling auch in verschiedenen Cloud-Umgebungen einsetzbar und transferierbar ist.

Als SMARTes-Ziele definiert:

SMARTes Vorhabenziele 2 Projektpartner SEGNO	<p>OSCAR Connect / digitaler Zwilling</p> <ul style="list-style-type: none"> - Entwicklung eines IoT Connectors zur Datenübertragung aus seiner industriellen Cloud-Umgebung in ein industrielles Simulationssystem / digitaler Zwilling - Entwicklung einer Möglichkeit zur Übernahme von Daten aus dem digitalen Zwilling in ein industrielles Cloudsystem <p>Eine erste Lösung wird geplant bis zum Start des Demonstrators. Verbesserungen und Anpassungen werden bis zum Vorhabensende durchgeführt.</p>
---	---

2.3.3 Ziel 3: Industrielle Cloud-Anwendungen und Know-how Aufbau

Neben dem Aufbau von generellem Know-how für Cloud-Umgebungen durch Training in Verbindung mit praktischen Übungen sollen spezielle WEB-Anwendungen (Cloud-Apps) für umweltbezogene Anlagen auf (Kreuzfahrt)-Schiffen in Abstimmung mit den Partnern im Vorhaben ausgearbeitet bzw. entwickelt werden.

Dazu gehören flottenbasierende Darstellungen mit leichter Bedienbarkeit und verständlicher Datenpräsentation (OSCAR-Analysis). Das erhöht die Akzeptanz bei zukünftigen Anwendern. Die App-Nutzung über Webbrowser ermöglicht eine erhöhte Flexibilität der Standort unabhängigen Datenbereitstellung für alle Projektpartner im Vorhaben. Hierzu müssen in der webbasierten Plattform personalisierte Darstellung von Inhalten und individuelle Rollen- und Rechte im Cloud-Management eingerichtet und ausgetestet werden.

Eine vollautomatische und hoch performante Datenakquisition vieler Messpunkte würde ein hohes Investment für die Cloud-Lösung nach sich ziehen und so an wirtschaftliche Grenzen und an einen

betriebswirtschaftlichen Nutzen stoßen. Konsequent zugeschnittene Datenaufzeichnungen sowie detaillierte Festlegung der Nutzdaten für die Wirtschaftlichkeit der Cloud-Lösung müssen daher gemeinsam im Vorhaben mit den Partnern ausgelotet werden (OSCAR-BigData).

Als SMARTe-Ziele definiert:

SMARTes Vorhabenziel 3 Projektpartner SEGNO	<p>OSCAR BigData</p> <ul style="list-style-type: none">- Datenaufbereitung und Datenvorhaltung in einem industriellen Cloud-System für die Weiterverarbeitung in einem Simulationssystem / digitaler Zwilling- Langzeit-Datenarchivierung für zukünftige Langzeitprognosen und für eine zentrale Nachweisbarkeit von Umweltdaten aus Kreuzfahrtschiffen <p>Die Umsetzung des Ziels wird geplant bis zum Start des Demonstrators. Verbesserungen und Anpassungen werden bis zum Vorhabensende durchgeführt.</p>
--	--

3 Anforderungen an Cloud-Systeme

Die drei Anforderungen an Cloud-Systeme sind die flexible Bereitstellung, Betriebskosten-einsparungen und Skalierbarkeit.

Zu den wichtigsten Vorteilen von Cloud Computing zählen:

- Schnelle Einführung
- Geringe oder keine Investitionskosten
- Sofortige Skalierbarkeit
- Keine Wartung
- Standortunabhängiger Zugriff
- Mehr Sicherheit

Cloud-Computing enthält drei verschiedene Servicemodelle:

3.1 Software as a Service (SaaS)

Clouds bieten Nutzungszugang von Software-Sammlungen und Anwendungsprogrammen. SaaS-Diensteanbieter offerieren spezielle Auswahlen von Software, die auf ihrer Infrastruktur läuft. SaaS wird auch als Software on demand (Software bei Bedarf) bezeichnet.

3.2 Platform as a Service (PaaS)

Clouds bieten Nutzungszugang von Programmierungs- oder Laufzeitumgebungen mit flexiblen, dynamisch anpassbaren Rechen- und Datenkapazitäten. Mit PaaS entwickeln Nutzer ihre eigenen Software-Anwendungen oder lassen diese hier ausführen, innerhalb einer Softwareumgebung, die vom Dienstanbieter (Service-Provider) bereitgestellt und unterhalten wird.

3.3 Infrastructure as a Service (IaaS)

Clouds bieten Nutzungszugang von virtualisierten Computerhardware-Ressourcen wie Rechnern, Netzen und Speicher. Mit IaaS gestalten sich Nutzer frei ihre eigenen virtuellen Computer-Cluster und sind daher für die Auswahl, die Installation, den Betrieb und das Funktionieren ihrer Software selbst verantwortlich.

3.4 IT Security

Durch die Unterstützung Cloud-Foundry innerhalb einer Cloud werden zudem weitere Sicherheiten gegeben. CloudFoundry ist ein offenes Plattform-as-a-Service (PaaS)-Framework, das Unternehmen dabei unterstützt, cloudbasierte Anwendungen schneller zu entwickeln und zu betreiben.

CloudFoundry bietet eine Vielzahl von Sicherheitsfunktionen, um die Sicherheit von Anwendungen und Daten in Cloud-Systemen zu gewährleisten.

Einige Beispiele für Sicherheitsfunktionen in CloudFoundry sind:

- Authentifizierung und Autorisierung: CloudFoundry verwendet Single Sign-On (SSO) und OAuth-basierte Authentifizierung, um sicherzustellen, dass nur autorisierte Benutzer auf die Anwendungen und Daten zugreifen können.
- Verschlüsselung von Netzwerkverkehr und Daten: CloudFoundry bietet automatische Verschlüsselung von Netzwerkverkehr und Daten, um sicherzustellen, dass die Daten sicher übertragen und gespeichert werden.

- Isolation von Anwendungen und Ressourcen: CloudFoundry bietet Container-basierte Isolation von Anwendungen und Ressourcen, um sicherzustellen, dass sich Anwendungen nicht gegenseitig beeinträchtigen und nur autorisierter Zugriff auf die Ressourcen erlaubt ist.
- Überwachung und Ereignisprotokollierung: CloudFoundry bietet Funktionen zur Überwachung und Ereignisprotokollierung, um Sicherheitsverletzungen und andere potenziell schädliche Aktivitäten zu erkennen und zu verfolgen.

Insgesamt bietet CloudFoundry eine Vielzahl von Sicherheitsfunktionen, die es Unternehmen ermöglichen, ihre Anwendungen und Daten in der Cloud sicher zu betreiben.

4 OSCAR-IoT-Connector

Als eine der Kernaufgaben des Projekts steht die Erfassung und Überwachung der Prozessdaten.

Diese sollen zentral abgelegt werden und von überall zugänglich sein. Dazu wurde entschieden die Daten innerhalb einer Cloud abzulegen. Nach den vorausgegangenen Recherchen wurde sich für die Siemens Mindsphere entschieden. Es handelt sich dabei um eine Cloud zur Erfassung von Zeitreihendaten (Timeseries) und der Verwaltung von Events und einem erweiterten Rechte-Management für einzelne Benutzer, welche ebenfalls CloudFoundry Container für die Applikationen bietet, und dadurch eine zusätzlichen Sicherheitsfaktor schafft.

Der IoT-Connector soll dabei die Übertragung der Prozesswerte aus dem Steuerungsnetz in die Cloud ermöglichen, und zusätzlich die dort hinterlegten Handlungsempfehlungen in Form von Sollwerten wieder zurück in die Steuerungen schreiben können.

In diesem Kapitel werden die Anforderungen, sowie die Umsetzung beschrieben.

4.1 Funktionsanforderungen

Der OSCAR IoT-Connector erfüllt folgende Funktionsanforderungen:

- Daten aus SPS-Systemen erfassen über S7-Verbindungen und/oder ModbusTCP
- Komprimieren von Datenpunkten mit der Möglichkeit der Bildung eines Mittelwertes
- Übertragung der Konfiguration in einen Mindsphere Agent via MindConnect-Lib
- Zeitreihenfolge-Daten an Cloudsystem senden (Timeseries)
- Generierung von Events innerhalb eines Mindsphere-Assets
- Auslesen von Sollwerten für Datenpunkte aus der Mindsphere
- Schreiben von Sollwerten in die SPS-Systeme.
- Hochladen von Zeitreihenfolge-Daten durch eine CSV-Datei (Import)
- Zwischenspeicherung der Daten bei nicht vorhandener Internetverbindung
- Daten aus anderem Cloudsystem erfassen und in eigenes Cloud-System übertragen

4.2 Konzept

Die folgende Grafik beschreibt das geplante Funktionskonzept

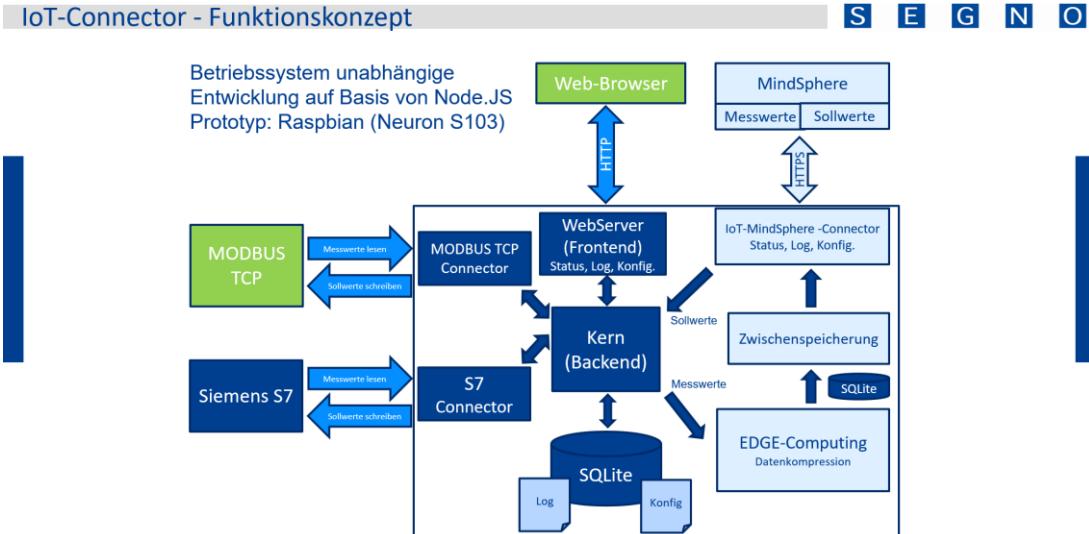


Abbildung 4: IoT-Connector Funktionskonzept

4.3 Umsetzung

Die Umsetzung vom IoT-Connector wurde auf Basis der ermittelten Anforderungen aus den Recherchen und Schulungen gewonnenen Kenntnissen durchgeführt.

4.3.1 Entwicklungsumgebung

Nach längerer Recherche bzgl. des Einsatzes einer Entwicklungsumgebung für ein IoT-Projekt, welches möglichst plattformunabhängig sein soll, wurde die Umsetzung der IoT-Connector auf Basis einer Node.JS Umgebung realisiert.

Node.JS ist eine Plattform, die serverseitig zum Betrieb von Netzwerkanwendungen konzipiert wurde. Es bietet die Möglichkeit, Web-Server Anwendung entwickeln und bereitzustellen, sowie Applikation in JavaScript laufen zu lassen. Ein großer Vorteil von Node.JS ist, dass dieses Plattformunabhängig ist, und ein Betrieb somit auf Windows sowie unter Linux möglich ist.

4.3.2 Verwendete Frameworks

Innerhalb der Node.JS Community werden viele verschiedene Frameworks und Bibliotheken angeboten, welche jede für sich einzelne Vor- und Nachteile für den jeweiligen Anwendungsfall bieten. Quellcode-Bibliotheken und Frameworks für Node.JS können über den Paketmanager „npm“ (node package manager) einfach in Softwareprojekte eingebunden werden. Innerhalb des Projektes wurden keine kostenpflichtigen Frameworks verwendet.

Alle verwendeten Frameworks unterliegen der MIT-Lizenz und sind somit frei verfügbar.

- **Angular**
Angular ist ein Framework für Webapplikationen und basiert auf Typescript. Die Konfiguration des IoT-Connector kann somit über eine Webseite erfolgen, welche direkt auf dem Gerät läuft.
- **Typescript**
Bei Typescript handelt es sich um typisiertes JavaScript, welches die Möglichkeit bietet, Klassen und Interfaces zu verwenden, sowie den Code zu kapseln und in Module aufzuteilen

- **NestJS**
Bei NestJS handelt es sich um ein serverseitiges HTTP Server Framework und ermöglicht die Verarbeitung der Daten die aus dem Frontend.
- **TypeORM**
TypeORM erlaubt eine objektrelationale Abbildung (ORM – object-relational mapping) für Typescript und ermöglicht die Abbildung von Objekten in objektorientierten Datenbanken.

4.3.3 Verwendete Bibliotheken

Genauso wie die verwendeten Frameworks, unterstehen ebenfalls alle verwendeten Node.js Bibliotheken der MIT-Lizenz und sind frei verfügbar. Sie lassen sich ebenfalls über den Node.js Paketmanager NPM installieren.

- **@mindconnect/mindconnect-nodejs**
Offizielle Siemens Bibliothek für die Mindsphere. Ermöglicht Login, Verwaltung des Agenten, Verarbeitung der Sollwerte, sowie dem Upload von Timeseries-Daten und Events
- **modbus-stream**
Bibliothek für die Kommunikation mit Modbus-Steuerungen
 - **nodes7**
Bibliothek für die Kommunikation mit S7 Steuerungen
 - **sqlite3**
Bibliothek für den Zugriff auf SQLite Datenbanken.

4.3.4 Datenbank

Um die Konfigurations- und Laufzeitendaten lokal auf dem IoT-Connector zu verwalten und zu speichern, ist der Einsatz einer Datenbank notwendig. Diese Datenbank sollte ebenfalls möglichst plattformunabhängig und ressourcenschonend sein.

Als Datenbank für solche Anforderungen bietet sich SQLite an. Bei SQLite handelt es sich um eine dateibasierte Datenbank, die ohne Server-Software zum Einsatz kommt und somit Ressourcen spart.

4.3.5 Funktionsmodule

Im Folgenden werden die einzelnen Funktionsmodule auf Basis des Funktionskonzeptes mit ihren einzelnen Anforderungen beschrieben.

Unterteilt sind die Funktionsmodule in das Frontend, welches als Visuelles Element in Form einer Webseite vorhanden ist, und dem Backend, in welchem die Logik eingebunden ist und die Daten verarbeitet.

4.3.5.1 *Frontend*

Als Frontend wird die Präsentationsfläche bezeichnet. Im Fall vom IoT-Connector handelt es sich um eine Webseite, die eine grafische Oberfläche zur Konfiguration und zur Prozessüberwachung bereitstellt. Das Frontend besteht aus einem Webserver sowie einer Webseite.

- **Web-Server**
Das Web-Server-Modul startet einen Webserver auf einem frei wählbaren Port des Systems. Er ermöglicht die Laufzeit einer Webseite die als Oberfläche für den Benutzer dient. Es ermöglicht die Konfiguration und Überwachung des IoT-Connector sowie die Statusübersicht.

4.3.5.2 **Backend**

Im Backend befindet sich die Logik der Applikation. Konfigurationen, die über das Frontend getätigten werden, werden an das Backend verschickt und durch dieses in der Datenbank abgelegt. Ebenfalls findet hier die Kommunikation mit den angebundenen Steuerungen sowie die Übertragung der Daten zur Mindsphere im Backend zusammen.

- **Kern-Modul (Backend)**

Im Kern-Modul wird die Verbindung zu den Steuerungen aufgebaut. Außerdem werden die parametrisierten Items dort zyklisch im angegebenen Zeitfenster ausgelesen und entsprechend der Konfiguration komprimiert. Anschliessend werden diese Daten in die Mindsphere hochgeladen. Ist ein Upload nicht möglich, werden die Werte lokale zwischengespeichert und bei der nächsten Verbindung mit der Mindsphere hochgeladen. Das Lesen von Sollwerten aus der Mindsphere, sowie dem Schreiben in die Steuerung ist

- **S7-Konnektor**

Mit dem S7-Konnektor besteht die Möglichkeit, Werte aus einer Siemens S7-SPS auszulesen und entsprechend der Konfiguration weiter bereitzustellen

- **Modbus/TCP-Konnektor**

Mit dem Modbus/TCP -Konnektor besteht die Möglichkeit, Werte aus beliebigen Geräten, die ModbusTCP unterstützen, auszulesen und entsprechend der Konfiguration weiter bereitzustellen

- **„Import-Only“ Konnektor (CSV)**

Es gibt einen Konnektor vom Typ „Import-Only“, der es ermöglicht Daten aus einer Steuerung in der Mindsphere zu hinterlegen, die keine Anbindung an den IoT-Connector hat. Über diesen ist es möglich, Daten per CSV-Datei in den IoT-Connector zu importieren

- **SQLite-Konfig-/Log-Datenbank**

In der SQLite-Datenbank werden alle notwendigen Konfigurationsdaten gespeichert. Außerdem werden dort die Programm-Logmeldungen abgelegt. SQLite ist Betriebssystem-unabhängig

- **EDGE-Computing**

Im Modul EDGE-Computing werden die Erfassten Rohwerte aus den Konnektoren zwischengespeichert und zu vorkonfigurierten Kompressionswerten zusammengefasst. anschließend erfolgt eine Zwischenspeicherung der Ergebnisse innerhalb der Applikation, bis diese an die Mindsphere verschickt wurden. Ist keine Internetverbindung vorhanden, so werden diese auf dem System zwischengespeichert.

- **Zwischenspeicherung**

Die Anbindung des IoT-Connectors an die Mindsphere erfolgt über das Internet. Besteht zu einem Zeitpunkt keine Internetverbindung, können Messwerte und Events nicht in die Cloud übertragen werden. In diesem Fall werden die Daten lokal auf dem Gerät als JSON-Datei zwischengespeichert. Sobald die Verbindung zur Mindsphere wiederhergestellt werden konnte, werden aufgelaufene Messwerte und Events nachträglich übertragen.

- **IoT-MindSphere-Connector**

Die Kommunikation mit der Mindsphere erfolgt über einen MindSphere-Connector. Dieser enthält die Credentials des MindConnect-Lib Agents der Mindsphere.

Mit dem IoT-MindSphere Connector werden aufgezeichneten Daten in der angegebenen Komprimierung übermittelt. Eine Intervalzeit für das Versenden der Daten ist hierbei möglich.

- **OPCUA-Client-Konnektor**

Bevor das Protokoll für die Kommunikation festgelegt wurde, wurden mehrere Protokolle in Betracht gezogen. Aufgrund der Projektspezifikationen ist eine Kommunikation zu OPCUA nicht erforderlich gewesen. Die Struktur wurde dennoch so gewählt, dass eine nachträgliche Anbindung an einen OPCUA-Server möglich ist

Neben den bereits erwähnten Paketen wurden noch einige kleinere Node.js Pakete verwendet, die keine wesentlichen Funktionen bieten, sondern bei denen es sich lediglich vereinfachte Implementierungen von Basisfunktionen bieten. Daher werden diese nicht explizit aufgeführt.

4.3.6 Entwicklung und Tests

Während der Entwicklung musste die Kommunikation mit Modbus-TCP implementiert und getestet werden. Da keine Modbus-TCP Steuerung vorhanden war musste hierfür eine Lösung gefunden werden. Zum Einsatz kam daher eine Modbus-TCP Simulation auf Basis von Node-RED.

4.3.6.1 **Node-RED**

Node-RED ist ein grafisches Entwicklungswerkzeug, welches über ein Baukastenprinzip innerhalb des Browsers konfiguriert werden kann. Die Installation von Node-RED kann wahlweise auf einem Windows-System, als auch unter Linux stattfinden.

Innerhalb des Entwicklungssystems wurde Node-RED dabei auf einen weiteren Unipi Neuron S103-G installiert.

Durch die Modbus-TCP Erweiterung „node-red-contrib-modbus“ innerhalb von Node-RED konnte dabei eine Steuerung simuliert werden, auf die wie eine Hardware-basierte Steuerung zugegriffen werden kann.

4.3.6.2 **Einrichtung Datenpunkte**

Nachdem der simulierte Modbus-TCP Server auf dem Node-RED System eingerichtet wurde, mussten die benötigten Datenpunkte für den Kommunikationstest angelegt werden.

In Absprache mit der LUH wurde im Vorfeld die Datenpunktliste der Testanlage ausgetauscht. So konnten die Datenpunkte innerhalb der Node-RED Testumgebung angelegt, und auf die korrekten Adressen gelegt und getestet werden.

Innerhalb von Node-RED wurde daher eine Simulierung jedes Datenpunkts innerhalb eines gültigen Wertebereiches angelegt.

Dies hatte den Vorteil, dass der IoT-Connector nach Fertigstellung direkt an die LUH verschickt werden konnte, und lediglich die Adresse der Steuerung angepasst werden musste. Die Konfiguration aller Datenpunkte war somit bereits abgeschlossen und die Einrichtung des IoT-Connector im LUH-Testsystem konnte ohne viel Aufwand in Betrieb genommen werden konnte.

4.3.6.3 **Testen der Pilotanlage**

Nachdem die Grundfunktionalitäten des IoT-Connector erfolgreich mit der simulierten Steuerung in der Entwicklungsumgebung innerhalb der SEGNO getestet wurde, wurde ein IoT-Connector an die LUH versandt und konnte erfolgreich für die Pilotanlage in Betrieb genommen.

Der im Rahmen von OSCAR im Hause SEGNO entwickelte IoT Connector wurde an die OSCAR-Versuchsanlage (ISAH Laborkläranlage) angeschlossen und sorgt seit Dezember 2020 für eine zuverlässige Datenerfassung und -übertragung in die OSCAR-Cloud. In der Cloud werden die Daten gespeichert, visualisiert und für die weitere Nutzung zugänglich gemacht. Auf diese Weise kann der Anlagenzustand aus der Ferne überwacht und die Prozessdaten weiter analysiert werden.

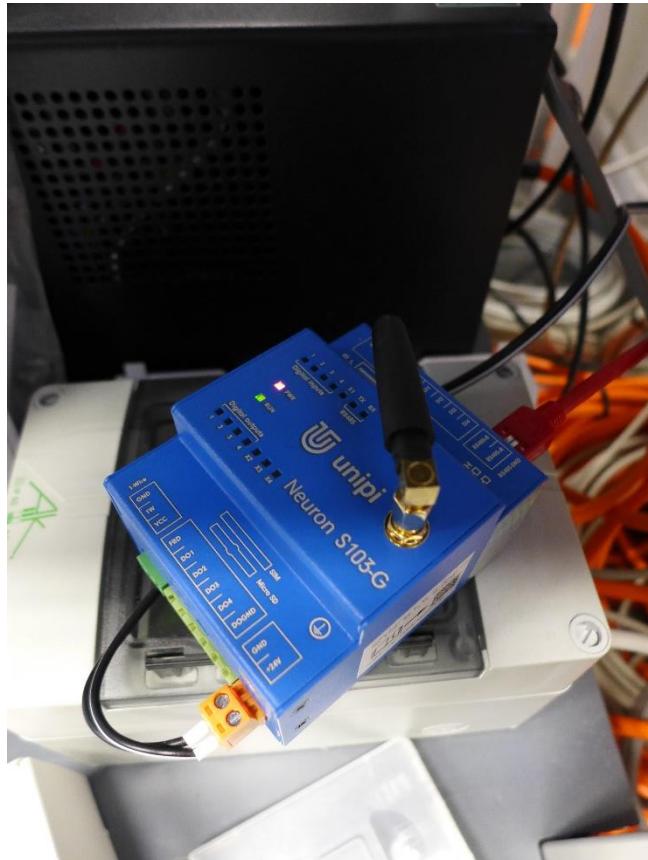


Abbildung 5: IoT-Connector im Einsatz innerhalb der Laboranlage

Anfang Oktober 2021 ist die Schiffskläranlage im Technikum des ISAH auf der Kläranlage Hannover-Herrenhausen in Betrieb gegangen. Die extra für OSCAR mit umfangreicher Messtechnik ausgestattete Schiffskläranlage wird genutzt um den digitalen Zwilling der Abwasserreinigung an Bord, die Datenfernübertragung mit dem IoT Connector und das OSCAR-Betriebsassistenzsystem zu entwickeln und zu erproben. Bei der Anlage handelt es sich um eine kleine Version (Gesamtvolumen ca. 4,5 m³) des Anlagentyps CS-MBR der RWO GmbH, mit der sowohl CSB als auch Nährstoffe (Stickstoff- und Phosphorverbindungen) entfernt werden können und die damit für den Einsatz in besonders sensiblen Seegebieten wie der Ostsee geeignet ist. Im Rahmen von OSCAR war die Anlage bis zum Sommer 2022 durchgehend in Betrieb.

4.3.6.4 Importieren von Handeingaben

Nachdem die Grundfunktionalität für das Erfassen aus einem Steuerungsnetz und dem Versenden der Daten in die Mindsphere umgesetzt wurde, musste ebenfalls eine Möglichkeit geschaffen werden damit Prozessdaten in die Mindsphere übertragen werden können, die aus anderen Quellen kommen, wie zum Beispiel Handeingaben. Dazu wurde zusammen mit der LUH ein Format festgelegt, mit dem Prozessdaten über eine CSV-Datei importiert werden können.
Die so gewonnenen Prozessdaten können über folgendes Format in einer CSV-Datei hinterlegt werden, und über die Webseite vom IoT-Connector in die Mindsphere hochgeladen werden.

Die Syntax innerhalb der CSV-Datei wurde dabei in Absprache mit der LUH wie folgt definiert:

- {Zeitstempel};{Konnektor-Name};{Datenpunkt-Name};{Prozesswert};{optionaler Kommentar}

Dabei stehen die Einträge für folgende Werte:

- **Zeitstempel**
Der Zeitstempel für den Datenpunkt muss im ISO8601 Format angegeben werden.
YYYY-MM-ddTHH:mm:ss
- **Konnektor-Name**
Der Konnektor Name muss im IoT-Connector hinterlegt sein, ebenso muss er in der Mindsphere als Aspect im Asset vorhanden sein.
- **Datenpunkt-Name**
Der Datenpunkt-Name entspricht dem Namen des Messwerts, und muss dem Konnektor im IoT-Connector zugehörig sein. Ebenfalls muss dieser als Variable im Aspect vorhanden sein.
- **Prozesswert**
Der Prozesswert entspricht dabei dem Wert, der in der Mindsphere hinterlegt werden soll. Er kann wahlweise als Boolean, Integer oder Double angegeben werden.
- **Kommentar**
Der Kommentar ist optional. Wird ein Kommentar gesetzt, so wird ebenfalls ein Event innerhalb der Mindsphere unter dem angegebenen Zeitstempel generiert.

Es ist dabei möglich, mehrere CSV-Dateien über die Webseite des IoT-Connector gleichzeitig zu importieren.

Gibt es Fehler innerhalb der CSV-Datei, so wird kein Eintrag der Datei in die Mindsphere übernommen.

4.3.7 Veröffentlichung

Bevor das Frontend und das Backend vom IoT-Connector auf das Zielsystem übertragen werden kann, müssen diese jeweils als lauffähige Release-Version erstellt werden. Dieser Schritt wird in der Entwicklungsumgebung Visual Studio Code ausgeführt. Nach der Veröffentlichung eines Projektes, enthält das Release alle Informationen über die benötigten Pakete, die auf dem Zielsystem vorhanden sein müssen. Abhängigkeiten zu benötigten Paketen werden dazu in der Datei „package.json“ hinterlegt.

Da es sich bei Node.js um eine platformunabhängige Programmiersprache handelt, können die verwendeten Pakete und Frameworks nicht mit im Release enthalten sein, da das Zielsystem auf einer anderen Architektur aufbauen kann und diese somit nicht lauffähig sind. Pakete aus dem „node package manager“ können verschiedene Implementierungen je unterliegendem System aufweisen.

Bei der Installation der benötigten Pakete auf dem Zielsystem wird dann die Architektur des Systems berücksichtigt, und die entsprechende Implementierung heruntergeladen.

4.3.7.1 Frontend

Bei dem Frontend handelt es sich um eine Applikation, welche mit dem Framework Angular umgesetzt wurde. Angular-Applikation lassen sich mit Angular-Kommandozeilenbefehlen veröffentlichen.

Eine Veröffentlichung ist mit folgendem Befehl möglich:

- „ng build –prod“

Das Befehlskürzel „ng“ steht dabei für „Angular“ und signalisiert, dass es sich bei dem folgenden Befehl um einen Angular-Befehl handelt. Gefolgt vom Angular-Befehl „build“, welcher die Applikation erstellt. Als Parameter wird noch --prod angegeben. Dies gibt dem Buildvorgang an, dass es sich um einen Build für eine Produktionsumgebung handelt. Bei einem Build für eine Produktionsumgebung

werden die Quellcode-Dateien des Projektes zu einer Datei zusammengefasst, und der Code darin minimiert. Zusätzlich wird der Quellcode mithilfe von „Uglification“ schwer lesbar gemacht, um so den Quellcode und das damit verbundene Know-How zu schützen.

Im Hintergrund führt der Buildvorgang von Angular den Buildvorgang vom „node package manager“ aus. Statt dem oben angeführten Befehl ist daher auch ein Erstellen des Release-Build über folgenden Befehl möglich:

- npm run build --prod

Das Befehlskürzel „npm“ steht dabei für „node package manager“ und signalisiert, dass es sich bei dem folgenden Befehl um einen Node.js-Befehl handelt. Gefolgt vom Befehl „run build“ sowie dem Parameter --prod für ein Build für die Prokutionsumgebung.

4.3.7.2 **Backend**

Bei dem Backend handelt es sich um ein Node.js Applikation, welche mit der Standard-Buildvorgang durch den „node package manager“ erstellt werden kann. Nach Veröffentlichung enthält dieser Release ebenfalls eine Datei mit allen benötigten Paketen für das Zielsystem. Ein Release ist daher ebenfalls wie für das Frontend über den Befehl „npm run build --prod“ möglich.

4.4 Einrichtung

Nachdem die Spezifikationen des IoT-Connectors festgelegt wurden, fiel die anfängliche Wahl für die Ersterprobung auf die Hardware des Unipi Neuron S103-G. Es handelt sich hierbei um ein kostengünstiges Einplatinen-System auf Basis eines Raspberry-PI der mit folgenden Anschluss-Möglichkeiten ausgestattet ist:

- WLAN
- Ethernet (RJ-45)
- 4x USB 2.0 Anschlüsse
- HDMI
- Mikro-SD Slot

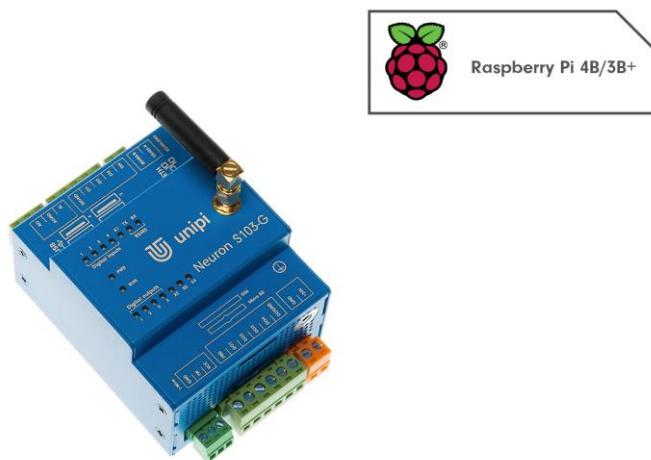


Abbildung 6: Quelle: Darstellung eines Unipi Neuron S103-G

4.4.1 Einrichtung Betriebssystem

Der Unipi Neuron S103-G wird werkseitig ohne Betriebssystem ausgeliefert. Als Betriebssystem bietet sich das „OpenSource OS“ an. Es handelt sich hierbei um ein frei verfügbares Debian Linux, 004.100 Schlussbericht 03SX486B OSCAR 2023-02-07 CNS.docx

welches direkt von der Webseite von Unipi heruntergeladen werden kann. Das Betriebssystem wird bei diesem Einplatinen-System auf eine Mikro SD-Karte gespielt, welche in einen Slot des Unipi Neuron S-103G gesteckt wird.

Mit Hilfe des Programms „Win32 Disk Imager“ kann die heruntergeladene Image-Datei auf die Mikro-SD Karte gespielt werden. Nachdem das Image auf die Karte geladen wurde, kann diese in den Unipi gesteckt werden und das System kann gestartet werden.

4.4.2 Zugriff auf den Unipi

Um einen Unipi einzurichten, gibt es zwei Möglichkeiten.

- Einrichtung am Gerät via HDMI
- Remote-Einrichtung via SSH

4.4.2.1 *Einrichtung via HDMI*

Die Einrichtung via HDMI bietet hierbei den Vorteil, dass die Konfiguration direkt auf dem Gerät erledigt werden kann, ohne dieses vorher im Netzwerk identifizieren zu müssen. Die HDMI-Schnittstelle kann allerdings erst verwendet werden, wenn ein Monitor vor dem Starten des Gerätes angeschlossen wird. Wurde ein Monitor nachträglich via HDMI angebunden, so sind die Treiber nicht geladen und es erfolgt keine Bildausgabe.

4.4.2.2 *Einrichtung via Remote*

Ist eine Einrichtung per Remote erforderlich, so muss die Netzwerkadresse des Gerätes bekannt sein.

Bei einem frisch aufgesetzten System ist die Konfiguration der Netzwerkkarte auf DHCP gestellt und das Gerät muss im Netzwerk erst identifiziert werden. Dieses kann sich in großen Netzwerken etwas schwierig erweisen. Selbst bei kleineren Netzwerken oder Direkt-Verbindung per Crossover muss die IP-Adresse erst ermittelt werden.

Ist die Netzwerkadresse bekannt, kann man sich per SSH auf das Gerät verbinden.

SSH ist standardmäßig unter OpenSource OS aktiviert. Wird ein anderes Betriebssystem gewählt, kann es sei das SSH erst aktiviert werden muss. Dazu muss das Gerät direkt via HDMI angeschlossen werden.

4.4.3 System-Update

Sofern das Gerät über DHCP schon eine Internetanbindung hat, sollte das System über Updates auf den neuesten Stand gebracht werden.

Dieses ist möglich über folgendene Befehle:

- „sudo apt-get update“
- „sudo apt-get upgrade“

Beide Befehle müssen mit dem sudo-Befehl ausgeführt werden. Befehle per sudo ermöglichen in der Linux Umgebung das Starten von Prozessen mit erhöhten Rechten. Sudo steht hierbei für den Begriff „Superuser do“. Das Upgraden / Updateen der Pakete erfordert hierbei erhöhte Rechte, da auf Systemdateien zugegriffen werden, die ein normaler Benutzer nicht schreiben darf.

Der erste Befehl „apt-get update“ lädt hier lediglich die Informationen der neuesten Version über die verwendeten Pakete herunter und hinterlegt diese im System. Erst durch den Befehl „apt-get upgrade“ werden die neuesten Pakete auch heruntergeladen und installiert.

4.4.4 Einrichtung des Systems

Für die weitere Einrichtung wurde ein neuer Benutzer im System angelegt. Damit dieser Benutzer alle nötigen Rechte besitzt, um die System-Einstellungen anzupassen, müssen diese dem Benutzer nach der Erstellung erst zugewiesen werden. Dies ist möglich über den Unix-Befehl sudo, der es erlaubt Befehle mit erhöhten Rechten zu Starten. Sofern nicht anders angegeben, werden diese über den Superuser „root“ ausgeführt. Alle systemrelevanten Einstellungen können daher nicht ohne den Befehl sudo erfolgen.

Statt jeden einzelnen Befehl mit einem vorausgegangen sudo zu starten, kann über folgenden Befehl ein dauerhafter Wechsel der Identität erfolgen.

- „sudo -i“

4.4.4.1 Weiteren Benutzer anlegen

Damit die Verwaltung des Systems nicht über den Root-Zugang erfolgt, können dem System weitere Benutzer hinzugefügt werden. Dies hat den Vorteil das verschiedenen Benutzern je nach Anforderungen verschiedene Rechte zugeteilt werden.

Das Anlegen eines weiteren Benutzers erfolgt über folgenden Befehl.

- „useradd -m segno -p {password}“

Damit ein neu angelegter Benutzer ebenfalls die benötigten Rechte erhält, müssen zudem folgende Befehle ausgeführt werden.

- „usermod -g users segno“
- „gpasswd -a segno ssh“
- „gpasswd -a segno sudo“

Über den ersten Befehl wird der neue Benutzer der Gruppe Users hinzugefügt. Weiterhin wird dieser zur Gruppe SSH und sudo hinzugefügt. Dies ermöglicht eine Anmeldung für diesen Benutzer über SSH sowie das ausführen von Befehlen mit erhöhten Rechten.

4.4.5 Einrichtung via raspi-config

Das verwendete OpenSource OS kommt ohne grafische Oberfläche. Konfigurationen müssen daher über die Konsole getätigkt werden, was mit einem erhöhten Aufwand einhergeht, da die Befehle einzeln eingetippt werden müssen. Um den Vorgang der Einrichtung etwas zu beschleunigen, bietet sich das Kommandozeilen-Tool „raspi-config“ an. Hier sind zentrale Einstellungen des Systems über eine Listenauswahl leicht erreichbar, ohne die genauen Befehle per Konsole nutzen zu müssen.

Das Tool kann über folgenden Befehl nachträglich installiert werden:

- „sudo apt-get install raspi-config“

Nach der Installation kann raspi-config über folgenden Befehl bestartet werden:

- „sudo raspi-config“

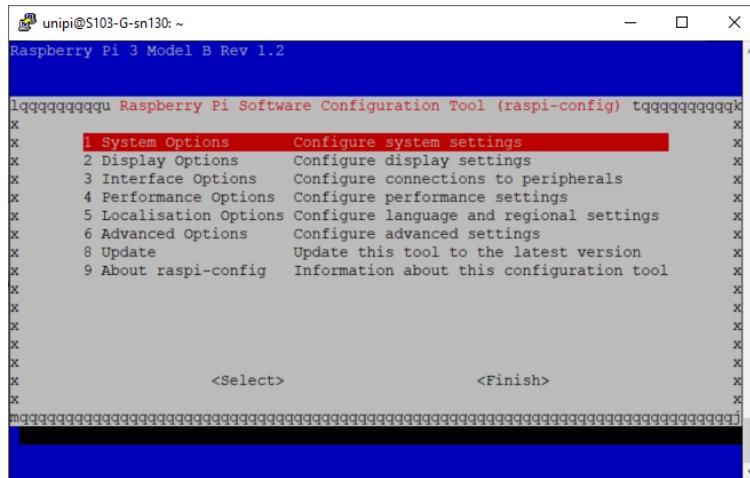


Abbildung 7: Konfiguration - raspi-config Hauptmenü

4.4.5.1 Ändern des Hostnamen

Für eine bessere Erkennung des Gerätes im Netzwerk, sollte der Hostname geändert werden. Dieser wurde im Zuge des Projektes auf „IoT-Connector“ geändert.

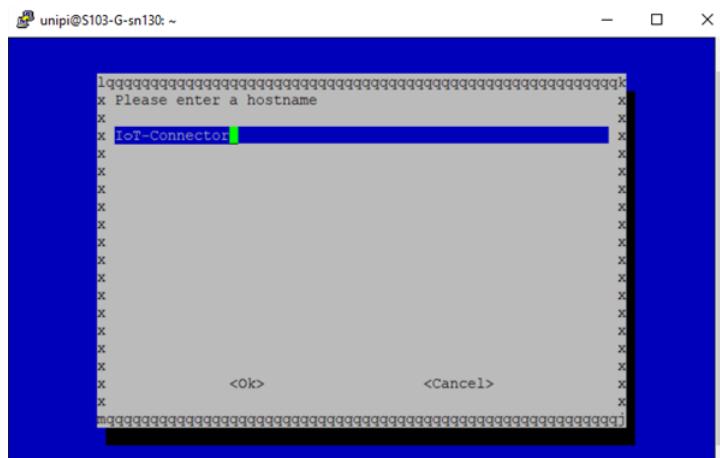


Abbildung 8: Konfiguration raspi-config - hostname

4.4.5.2 Einrichtung WLAN

Zusätzlich zur Ethernet-Schnittstelle, ist der Unipi Neuron S-103G mit einer WLAN-Schnittstelle ausgestattet. Durch die Ethernet Schnittstelle ist der Neuron an das PLS-Netz angebunden. Aus Sicherheitsgründen befindet sich diese Netze meist nicht im Internet. Damit die Daten dennoch per Internet an die Cloud geschickt werden kann, wird die WLAN Schnittstelle genutzt. Benötigt werden daher die SSID, sowie das dazugehörige Kennwort (Passphrase).

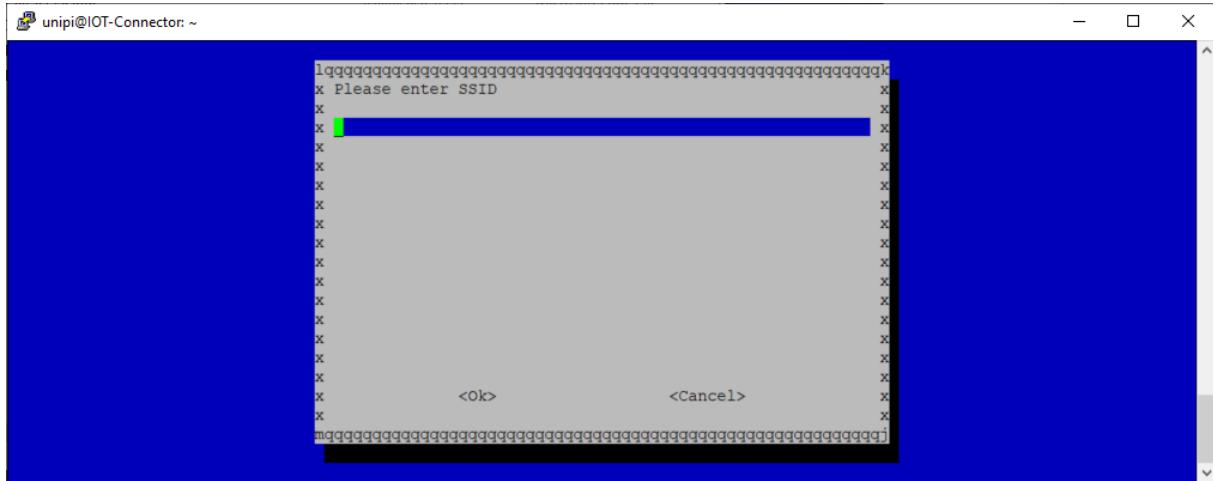


Abbildung 9: Konfiguration raspi-config SSID



Abbildung 10: Konfiguration raspi-config - passphrase

4.4.5.3 Einrichtung Zeitzone / Uhrzeit

Sofern nicht automatisch erkannt, muss die Zeitzone angegeben werden, in dem sich das Gerät befindet. Es handelt sich hierbei um die Systemzeit, welche zum Beispiel für alle Logmeldungen innerhalb des IOT-Connector genutzt wird. Ebenfalls wird dieser Zeitstempel beim Auslesen der Ist-Werte aus den Steuerungen genutzt, die in die Cloud übertragen werden.

4.4.6 Einrichtung IoT-Connector

Nachdem das Zielsystem konfiguriert wurde, kann die IoT-Connector – Software auf das Gerät kopiert und eingerichtet werden. Der Kopiervorgang und die Installation der Pakete können über das Netzwerk stattfinden. Über das frei verfügbare Programm WinSCP können die Daten direkt auf das System kopiert werden.

WinSCP nutzt dazu eine verschlüsselte Verbindung via SSH-Protokoll und kommt mit einer grafischen Oberfläche. Das Kopieren der Daten kann so über einen Datei-Explorer stattfinden. Befinden sich die Daten auf dem Zielsystem, so ist es über eine SSH-Verbindung zu dem System möglich, die Einrichtung der IoT-Connector Software per Kommandozeile auszuführen. Dazu bietet sich das frei verfügbare Programm „Putty“ an.

Verbindet man sich über SSH auf das Gerät, so kann die Installation der npm-Pakete erfolgen. Dies ist gesondert für das Frontend und das Backend auszuführen. Eine Installation der Pakete kann über folgenden Kommandozeilenbefehl direkt im Ordner erfolgen, in dem auch die „package.json“ liegt.

- sudo npm install

Dieser Befehl muss ebenfalls per sudo mit erhöhten Rechten ausgeführt werden. Der „node package manager“ sucht daraufhin in dem Verzeichnis aus dem der Befehl aufgerufen wird nach der „package.json“ und installiert alle benötigten Pakete. Nach Abschluss der Installation kann der IoT-Connector verwendet werden.

4.4.7 Autostart

Nachdem die Software auf dem System eingerichtet wurde, kann diese an den Autostart des Systems angehängt werden. Verliert der IoT-Connector die Stromzufuhr, so kann auf diese Weise sichergestellt werden, dass das Frontend und Backend bei jedem Systemstart automatisch gestartet wird. Die Einrichtung des Autostarts hängt vom jeweiligen System ab. Beim verwendeten Unipi Neuron S103-G sowie dem verwendeten Betriebssystem „OpenSource OS“ gibt es die Möglichkeit ein Shell-Skript im Systemordner „/etc/init.d“ abzulegen. Der Skriptname ist frei wählbar.

Anschliessend müssen die Rechte des Skripts noch mit folgendem Befehl festgelegt werden:

- sudo chmod 644 /etc/init.d/{skriptname}

Der Befehl „chmod“ ändert dabei die Zugriffsberechtigungen für das Skript-Shell und benötigt dafür erhöhte Rechte. Die darauffolgenden Zahlen geben die Berechtigungen an und unterliegen den Spezifikationen der Unix-Rechtevergabe. Diese werden mit „644“ angegeben und schlüsseln sich folgendermaßen auf.

Die erste Zahl steht dabei für die Berechtigungen des Besitzers, die zweite für die Gruppenmitglieder und die letzte Zahl gibt an, welche Rechte alle anderen User besitzen.

Mit der „6“ wird angegeben, dass der Besitzer Lese-/Schreibzugriff besitzt. Eine „4“ ermöglicht dabei das Ausführen der Datei, sowie einen Lesezugriff. Es handelt sich hierbei um Dateiberechtigungen

Damit das Skript beim Systemstart ausgeführt wird, muss es noch als Dienst aktiviert werden. Da es sich um Systemeinstellungen handelt, muss der folgende Befehl ebenfalls mit erhöhten Rechten ausgeführt werden:

- sudo update-rc.d omc defaults

Der Befehl „update-rc.d“ fügt eine Verknüpfung zum Autostart hinzu. Der IoT-Connector ist nun für den Autostart konfiguriert.

4.5 Konfiguration IoT-Connector

Um die geplanten Anforderungen zu realisieren, bietet der IoT-Connector verschiedene Konfigurationsmöglichkeiten. Für die Einrichtung müssen daher die Anbindungen an die Steuerungen, die auszulesenden Tags, sowie die Zugangsdaten eines „MindConnect-Lib“ Agenten zur Kommunikation mit der Mindsphere konfiguriert werden. Zusätzlich muss innerhalb der Mindsphere

ein äquivalent zur IoT-Connector-Konfiguration in Form eines Assets vorhanden sein, in dem die hochgeladenen Daten abgelegt werden können.

Nach abgeschlossener Konfiguration lädt der IoT-Connector zyklisch die Daten aus den Steuerungen in die TimeSeries der zuvor konfigurierten Assets in der Mindsphere.

4.5.1 Konfiguration der Konnektoren

Ein Konnektor übernimmt die Anbindung an verschiedene Steuerungen und ermöglicht das Auslesen der Prozesswerte. Je nach Steuerung und Kommunikationsprotokoll werden verschiedene Parameter benötigt, um eine Kommunikation aufzubauen.

Aktuelle stehen die folgenden drei Konnektoren zur Auswahl:

- S7
- ModbusTCP
- CSV

Neben dem S7- und dem ModbusTCP-Konnektor gibt es den CSV-Konnektor, der keine direkt Anbindung an eine Steuerung besitzt. Hierüber können Handeingaben über eine CSV-Datei mit einem fest definierten Format erfolgende.

Abgesehen vom CSV-Konnektor gibt es für den S7 und den Modbus-TCP Konnektor ein Lesezyklus-Intervall, über den angegeben werden kann, in welchem Intervall die Messwerte einer Steuerung ausgelesen werden. Die Messwerte werden im angegebenen Intervall ausgelesen und im Backend des IoT-Connector für die weitere Verarbeitung bereitgestellt.

Ein globaler Wert pro Steuerung hat den Vorteil, dass bei Protokollen auf Stream-basierten Daten wie zum Beispiel Modbus keine Überschneidung von Datenpaketen auftreten, wenn Messwerte parallel abgefragt werden. Dies könnte zu Problemen beim Auslesen führen.

Jeder Konnektor benötigt zusätzlich einen Namen, um bei mehreren Konnektoren eines Protokolls eine genaue Identifizierung bei der Konfiguration der Messwerte zu ermöglichen.

4.5.1.1 Konfiguration S7-Konnektor

Mit einem S7-Konnektor kann eine Verbindung zu einer Siemens S7 Steuerung aufgebaut werden. Hierfür werden folgende Parameter benötigt:

- Name
- IP-Adresse
- Port
- Lesezyklus
- Rack
- Slot

4.5.1.2 Konfiguration ModbusTCP-Konnektor

Mit einem ModbusTCP-Konnektor kann eine Verbindung zu einer ModbusTCP Steuerung aufgebaut werden.

Hierfür werden folgende Parameter benötigt:

- Name
- IP-Adresse
- Port
- Lesezyklus

4.5.1.3 Konfiguration CSV-Konnektor

Mit einem CSV-Konnektor wird keine aktive Verbindung zu einer Steuerung aufgebaut. Daher benötigt dieser Konnektor lediglich einen Namen.

4.5.2 Konfiguration Messwerte

Ein Messwert definiert einen bestimmten Datenpunkt innerhalb einer Steuerung. Je nach Steuerung und Protokoll unterscheiden sich die benötigten Informationen, die zum Anlegen eines Messwertes benötigt werden. Jeder Messwert muss einem Konnektor zugewiesen werden und benötigt eine Einheit und einen Datentyp. Beide Informationen müssen der Einheit und dem Datentyp innerhalb der Mindsphere entsprechen. Folgende Datentypen werden vom IoT-Connector unterstützt:

- Boolean
- Integer
- Double

Zusätzlich gibt es für jeden Messwert die Möglichkeit einen Offset, sowie einen Faktor anzugeben. Der übermittelte Wert an die Mindsphere setzt sich daher aus folgender Formel zusammen.

- Wert = (Rohwert * Faktor) + Offset

Für jeden neuen Messwert sind die Standardwerte für den Faktor mit 1 und für den Offset mit 0 angegeben. Optional kann jedem Messwert noch eine Beschreibung gegeben werden.

Zusätzlich gibt es die Möglichkeit, jedem Messwert einer Kompression zu unterziehen. Dafür steht aktuell die Kompressionsart „Mittelwert“ zur Verfügung. Hierüber kann zum Beispiel bei einem Messwert, der alle 5 Sekunden ausgelesen wird, ein Mittelwert über 60 Sekunden gebildet werden. Die Berechnung des Mittelwertes unterliegt daher folgender Formel:

$$\text{- } \bar{x} = \frac{a_1 + a_2 + \dots + a_n}{n}$$

4.5.2.1 Konfiguration Messwerte S7-Konnektor

Wird ein Messwert für eine S7-Steuerung angelegt, so werden folgende Informationen benötigt:

- Name
- Item (S7-Adresse)
- Einheit
- Datentyp
- Offset
- Faktor
- Lesezyklus
- Kompressionsart
- Kompressionszeitraum

Damit ein Wert aus einer S7-Steuerung ausgelesen werden kann, wird die Adresse innerhalb der Steuerung benötigt. Das Item eines S7-Messwerts muss daher der Syntax entsprechen, die der nodeS7- Bibliothek vorgibt. Nachfolgend daher ein Auszug der benötigten Syntax für das Definieren eines Datenpunkts in der S7 Steuerung aus der offiziellen Dokumentation der nodeS7-Bibliothek.

```
'MR4',           // Memory real at MD4
'M32.2',         // Bit at M32.2
'M20.0',         // Bit at M20.0
'DB1,REAL0..20', // Array of 20 values in DB1
'DB1,REAL4',     // Single real value
'DB1,REAL8',     // Another single real value
'DB1,INT12..2', // Two integer value array
'DB1,LREAL4',   // Single 8-byte real value
'DB1,X14.0',    // Single bit in a data block
'DB1,X14.0..8' // Array of 8 bits in a data block
```

Abbildung 11: Syntax für nodeS7 Adressen

4.5.2.2 Konfiguration Messwerte Modbus-TCP

Wird ein Messwert für eine ModbusTCP-Steuerung angelegt, so werden folgenden Informationen benötigt.

- Name
- Item (ModbusTCP Adresse mit Vorgegebener Syntax)
- Einheit
- Datentyp
- Offset
- Faktor
- Register-Typ
- Word-Order
- Byte-Order
- Offset
- Faktor

Wichtig beim Auslesen von Werten aus einer Modbus-Steuerung ist die Word- sowie Byte-Reihenfolge. Aufgrund der Architektur einer Modbus-Steuerung, findet der Datenaustausch über 16-Bit Register statt. Dabei handelt es sich um je Zwei 8-Bit Register, die je nach Steuerungstyp und Implementation in anderer Wertigkeit definiert werden können. Unterschieden wird dabei zwischen Big-Endian sowie Little-Endian.

- Big-Endian bedeutet, dass der höchstwertige Teil an der niedrigsten Speicheradresse steht.
- Little-Endian bedeutet, dass der niedrigwertigste Teil an der niedrigsten Speicheradresse steht.

Neben der Word-Order muss zusätzlich die Byte-Order berücksichtigt werden, da auch hier eine Unterscheidung zwischen Big- und Little-Endian möglich ist. Daraus resultiert neben den Big-/Little Endian ebenfalls noch Mid-Big Endian sowie Mit-Little Endian.

Es ergeben sich die folgenden möglichen Konstellationen.

Tabelle 1: Big Endian

A	B	C	D
11110000 00001111 11001100 10101010			

Tabelle 2: Little Endian

D	C	B	A
10101010 11001100 00001111 11110000			

Tabelle 3: Mid-BigEndian

B	A	D	C
00001111 11110000 10101010 11001100			

Tabelle 4: Mid-LittleEndian

C	D	A	B
11001100 10101010 11110000 00001111			

4.5.3 Konfiguration Mindsphere-Konnektor

Nachdem die Konnektoren sowie deren Messwerte parametriert wurden, kann eine Verbindung zur Siemens Mindsphere Cloudumgebung über den Mindsphere-Konnektor aufgebaut werden. Um Zeitreihenfolge-Daten in die Mindsphere zu übertragen wird ein Agent vom Typ Mindconnect-Lib benötigt. Es handelt sich hierbei um spezielles Mindsphere-Netzwerk Objekt welche einen Verbindungsschlüssel in Form eines JSON Web Tokens generiert. Über diesen ist es möglich, eine Verbindung zur Mindsphere aufzubauen. Dieser Token ist für maximal ein Jahr gültig und muss danach wieder neu generiert werden, und auf jedem IoT-Connector neu hinterlegt werden.

Mit diesem Token kann ein Onboarding durch das node.JS Paket „@mindconnect/mindconnect-nodejs“ durchgeführt werden, welches das System, auf dem das Onboarding stattgefunden hat, zur Kommunikation mit der Mindsphere autorisiert. Nachdem das Onboarding auf dem IoT-Connector ausgeführt wurde, muss die Konfiguration in das Mindconnect-Lib Objekt übertragen werden.

Dazu gibt es innerhalb des IoT-Connector unter dem Punkt „Messwerte“ die Möglichkeit die Konfiguration in die Mindsphere zu übertragen. Nach übertragener Konfiguration sind die Datenpunkte im Agenten innerhalb der Mindsphere bekannt.

Übertragene Daten, welche an die Mindsphere gesendet werden, müssen daraufhin noch mit einem Asset verknüpft werden, da diese sonst verworfen werden. Hierfür gibt es im Agenten in der Mindsphere die Möglichkeit, eingehende Daten mit beliebigen Assets und deren Datenpunkten zu verknüpfen. Allerdings gibt es die Restriktion, dass die Datenpunkte aus dem Agenten die gleiche Einheit und den gleichen Datentyp des Datenpunktes innerhalb des Assets besitzen müssen.

4.5.4 Konfiguration Sollwert-Übernahme

Eines der Projektziele bestand darin, Sollwerte zurück in die Steuerung zu schreiben. Allerdings ist die Übernahme der Sollwerte optional und kann entsprechend aktiviert, oder deaktiviert werden. Hierzu wurde innerhalb der OSCAR-Analysis Applikation in Mindsphere eine Möglichkeit geschaffen,

Sollwerte für einzelne Messwerte zu hinterlegen. Der IoT-Connector kann dazu bei bestehender Verbindung zur Cloud die Messwerte innerhalb der Cloud auslesen, und diese daraufhin in die Steuerung schreiben. Nach erfolgreichem Schreiben des Wertes wird ein Event innerhalb der Mindsphere registriert, und der zu setzende Sollwert wird in der Mindsphere zurückgesetzt.

Dazu werden folgende Informationen benötigt:

- Host-Tenant
- Tenant-Name
- Gateway
- Mindsphere-Applikation
- Mindsphere-Applikation Versionsnummer
- Client-ID
- Client-Secret
- Asset-ID

Anders als beim Hochladen der Messwerte in die Mindsphere, erfolgt ein Zugriff auf die Sollwerte innerhalb der Cloud nicht über den Agenten vom Typ „MindConnectLib“, sondern über Zugriffsrechte der Applikation, über welche die Sollwerte hinterlegt wurden. Eine Anmeldung an der Mindsphere über die Applikation wird durch Application-Credentials ermöglicht. Diese müssen in der Mindsphere explizit vergeben werden.

Beim Anlegen der Zugriffsberechtigungen werden von der Mindsphere die Client-ID, sowie das Client-Secret generiert. Zusätzlich wird für den Zugriff über das SDK noch das Gateway (z.B: eu1.mindsphere.io), der Applikationsname sowie dessen Versionsnummer benötigt. Des Weiteren wird noch der Host-Tenant sowie der Tenant benötigt, über den der Zugriff erfolgt. Als Host-Tenant wird dabei der Mindsphere Operator-Tenant bezeichnet, der die Applikation bereitstellt. Der Tenant dient zur eigenen Identifizierung und muss dem Tenant entsprechen, der die Mindsphere-Applikation nutzt. Sollwerte innerhalb der Mindsphere werden innerhalb einer JSON-Konfigurationsdatei innerhalb eines Assets definiert. Für den Zugriff wird daher ebenfalls die Asset-ID benötigt.

4.6 Entwicklung, Test und Übertragbarkeiten

Im Rahmen des Projektumfangs sollte eine Portierung und Übertragbarkeit auf verschiedene Kommunikationshardware untersucht werden. Bevor mit der Entwicklung des IOT-Connectors begonnen wurde, wurden daher die Rahmenbedingungen geprüft, die nötig sind, um die Software auf unterschiedlichen Geräten sowie Betriebssystemen lauffähig zu machen.

Im Vorfeld wurde der IOT-Connector daher wie bereits beschrieben in das Backend (Logik / Datenbank), sowie das Frontend (Benutzeroberfläche) unterteilt. Beide Projekte mussten daher in einer Programmiersprache entwickelt werden, die plattformunabhängig sind. Dabei ist die Wahl auf eine Entwicklung in Node.js / Angular gefallen, da diese Plattformunabhängig einsetzbar sind.

Entwickelt wurde der IOT-Connector innerhalb einer virtuellen Maschine mit Windows 10. Durch die Architektur von Node.js und Angular sind die benötigten Pakete und Laufzeitumgebungen nur im Projekt definiert, und nicht hinterlegt. Wird das Projekt auf ein Zielsystem kopiert, müssen die in den Projekten hinterlegte Pakete für das jeweilige System installiert werden. Dies geschieht über den „node package manager“. Durch den Aufruf von Konsolen-Befehl „npm install“ im Projektverzeichnis ermittelt der „node package manager“ anhand des Betriebssystems, welche Implementierung eines

Pakete heruntergeladen werden muss, und legt diese im Projektverzeichnis im Unterordner „node_modules“ ab. Der Ordner „node_modules“ kann somit nicht mit auf ein anderes Zielsystem mit unterschiedlicher Architektur kopiert werden.

Es musste lediglich darauf geachtet werden, dass Pfadangaben für Dateizugriffe (wie zum Beispiel den Pfad für die lokale Zwischenspeicherung von Prozesswerten bei nicht vorhandener Internetverbindung) mit korrekter Groß-/Kleinschreibung hinterlegt wurden. Während Pfadangaben in Windows-Betriebssystemen die Groß-/Kleinschreibung ignorieren, ist dies unter Linux nicht der Fall.

Zum Beispiel wird der relative Pfad „./data_storage“ unter Linux nicht gefunden, wenn das tatsächliche Verzeichnis „./Data_storage“ benannt wurde.

4.6.1 Test der Übertragbarkeit auf verschiedene Kommunikationshardware

Während der Entwicklung wurde nach größeren Funktionsupdates die Funktionalität auf allen Geräten mit unterschiedlicher Architektur getestet und erprobt. Eine Übertragbarkeit des OSCAR IOT Connectors ist gegeben und wurde erfolgreich auf den nachfolgenden aufgeführten Kommunikationshardware durchgeführt.

Ausblick:

Im Rahmen eines möglichen Anschlussvorhabens könnte eine genauere Untersuchung der oben beschriebenen Funktionalitäten auf weitere Kommunikationshardware weitere industrieller Automatisierungshersteller ausgetestet und erprobt werden.

4.6.2 Unipi Neuron S103-G



Abbildung 12: Darstellung eines Unipi Neuron S103-G

Parallel zur Entwicklung wurde nach der Fertigstellung einzelner Funktionen der aktuelle Entwicklungsstand zuerst auf einen Unipi Neuron S103-G installiert und die Funktionalität überprüft. Während der Entwicklung gab es keine Probleme beim Portieren von Projektzwischenständen auf den Unipi Neuron S103-G.

Der Unipi Neuron S103-G ist eine speicherprogrammierbare Steuerung für Automatisierung, Steuerung, Regelung und Überwachung. Die universelle Architektur des S103-G umfasst digitale und analoge I/Os und eignet sich somit für einfache Installationen. Als Besonderheit ist das S103-G zudem

mit einer integrierten GSM/GPRS-Funktionalität für den Internetzugang oder das Senden/Empfangen von SMS-Nachrichten ausgestattet.

Als Basissoftware für die einfache Implementierung der Benutzeranwendung wird das Betriebssystem Linux OpenSource OS und die Kommunikationsschnittstelle Modbus auf TCP-Protokoll genutzt.

Während der Entwicklung wurde nach größeren Funktionsupdates ebenfalls die Funktionalität auf den anderen Geräten mit unterschiedlicher Architektur getestet.

4.6.3 Phoenix Contact BL2 BPC 1501E-64-W10



Abbildung 13: Darstellung eines Phoenix Contact BL2 BPC

Der BL2 BPC 1501E-64-W10 ist ein konfigurierbarer Box-PC (BPC) mit geringem Platzbedarf. Der geringe Platzbedarf ermöglicht auch den Einsatz in Anwendungen, für die größere PCs zu groß oder anderweitig ungeeignet sind. Es handelt sich dabei um einen lüfterlosen Industrie-PC mit IP30 Schutzart. Das System wurde mit dem Betriebssystem Windows 10 IOT Enterprise ausgeliefert.

4.6.4 WAGO 752-9401 (Edge Computer)



Abbildung 14: Darstellung eines WAGO 752-9401

Der WAGO 752-9401 Edge Computer ist ein kabel- und lüfterloser Automatisierungscomputer, der Steuerungs-, Überwachungs- und Kommunikationsaufgaben erledigen kann. Das Produkt ist für eine DIN-/Tragschienen-Montage geeignet und zeichnet sich durch verschiedene Schnittstellen aus. Das Produkt kann für Anwendungen im Maschinen- und Anlagenbau sowie in der Prozessindustrie und der Gebäudetechnik eingesetzt werden. Es handelt sich dabei um einen lüfterlosen Industrie-PC mit IP40 Schutzart. Das System wurde mit Debian Linux 10.5 ausgeliefert.

4.6.5 Siemens IOT2050



Abbildung 15: Darstellung eines Siemens IOT2050

Der Siemens IOT2050 ist ein vielseitiges IoT-Edge-Gerät, mit dem Sie SPSEN z.B.: in einer Fabrikhalle oder auf einem Schiff an die Cloud anbinden können. Ein Dual-Core- oder Quad-Core-Prozessor bietet genug Leistung, um Analysen und Entscheidungen am Edge durchzuführen, bevor die Daten an die Cloud gesendet werden.

Das Gateway Siemens IOT2050 basiert hardwareseitig auf robuster, zuverlässiger und langlebiger Industrietechnologie. Das für die Wand- und Hutschienenmontage geeignete Gerät ist ausgestattet mit stromsparendem Texas Instruments ARM AM 6548, zwei Gigabyte DDR4 RAM und zahlreichen Schnittstellen, darunter zwei Gbit LAN, zwei USB, eine serielle und eine Arduino-Schnittstelle. Das Betriebssystem Simatic Industrial OS ist bereits vorinstalliert. Simatic IOT2050 ist mit Arduino Shields und miniPCIe Cards einfach für maßgeschneiderte Lösungen erweiterbar und unterstützt dazu Linux based on Debian. Vielfältige Möglichkeiten zur Programmierung in Hochsprachen, stehen zur Verfügung.

5 Konfiguration der Siemens Mindsphere

Um die Prozessdaten vom IoT-Connector an die Zeitreihendaten der Siemens Mindsphere zu übermitteln, muss innerhalb der Mindsphere noch die Struktur für die eingehenden Daten erstellt, und die Möglichkeit einer Datenübertragung aus einer Quelle außerhalb der Mindsphere geschafft werden. Daten, die innerhalb der Mindsphere auflaufen, werden in sogenannten Assets hinterlegt. Für jeden angebundenen IoT-Connector muss daher ein Asset generiert werden, welches dieselbe Struktur wie die Quelle aufweist.

Zusätzlich muss innerhalb der Mindsphere Netzwerk-Objekt geschaffen werden, durch das eine Verbindung zum Mindsphere und dem dazugehörigen Asset generiert werden kann. Dieses Netzwerkobjekt wird als Agent bezeichnet. Die Erstellung und Bearbeitung der Objekte wird über die Mindsphere Applikation „Asset-Manager“ ermöglicht.

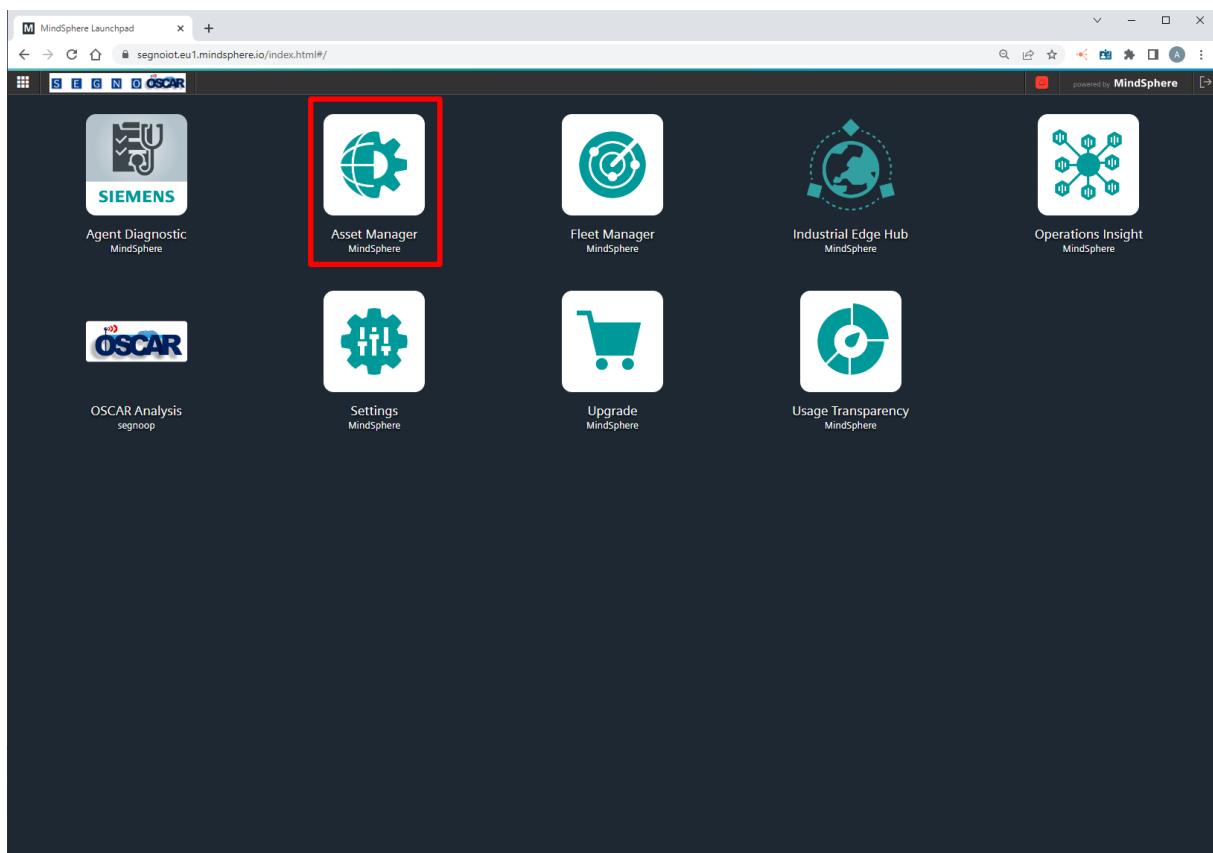


Abbildung 16: Übersichtsseite Mindsphere

5.1 Asset-Konfiguration

Assets innerhalb der Mindsphere definieren eine Datenstruktur für Prozesswerte, und können dabei aus mehreren Aspects bestehen, sowie einzelnen Datenpunkten bestehen. Aspects wiederum definieren weitere Unterstrukturen aus mehreren Datenpunkten. Dadurch lässt sich ein IoT-Connector als Asset ansehen, und die angebundenen Konnektoren / Steuerungen entsprechen den Aspects.

Geht man von einer baugleichen Weise einer Steuerung und den damit verbundenen Datenpunkten aus, lässt sich so der Einrichtungsaufwand für jeden weiteren IoT-Connector deutlich minimieren, da die Datenpunkte aus den jeweils angebundenen Steuerungen bereits über die Aspects bekannt sind. Es kann daher ein Asset mit der benötigten Struktur aus den Asset-Vorlagen generiert werden, und muss nur noch mit den eingehenden Daten eines Agenten verknüpft werden.

Innerhalb der Mindsphere gibt es bereits vorgefertigte Assets, es besteht aber auch die Möglichkeit eine eigene Vorlage anzulegen. Die Vorlagen entsprechen dabei einem Asset-Type. Über einen Asset-Type ist es möglich, eine neue Instanz eines Assets zu generieren.

5.2 Asset-Types

Ein Asset wird immer durch seinen Typ definiert. Durch den Asset-Manager ist es möglich, einen eigenen Asset-Type für Assets mit denselben Datenpunkten zu definieren. Asset-Types können mehrere Datenpunkte, oder Unterstrukturen vom Typ Aspects definieren. Fasst man mehrere Aspects in einem Asset-Type zusammen, lässt sich das daraus resultierende Asset ohne weiteren Aufwand anlegen.

Name	Einheit	Datentyp	Max. Länge	Default-Wert
Schiffname	-	STRING	60	-

Name	Aspect	Kategorie	Definiert
> BlueBox	segnoiot.BlueBox	Dynamisch	<input checked="" type="checkbox"/>
> FP4IC	segnoiot.FP4IC	Dynamisch	<input checked="" type="checkbox"/>
> Labor	segnoiot.Labor	Dynamisch	<input checked="" type="checkbox"/>
> SC1000	segnoiot.SC1000	Dynamisch	<input checked="" type="checkbox"/>

Abbildung 17: Mindsphere AssetManager - Asset Konfiguration

5.3 Aspects

Aspects stellen die kleinste Unterstruktur dar. Sie können als äquivalent einer Steuerung angesehen werden. Wird also eine baugleiche Steuerung mehrfach verwendet, bietet es sich an daraus ein eigenes Aspect zu generieren.

Ein Datenpunkt muss dabei immer aus einem Namen, dem Datentyp sowie einer Einheit bestehen. Die Kombination aus den drei Eigenschaften muss zudem eindeutig sein.

Name	Einheit	Datentyp	Max. Länge	Default-Wert
deni_Cl	mg/L	DOUBLE	-	-
deni_cond	mS/cm	DOUBLE	-	-
deni_DO	mg/L	DOUBLE	-	-
deni_K	mg/L	DOUBLE	-	-
deni_NH4_N	mg N/L	DOUBLE	-	-
deni_NO3_N	mg N/L	DOUBLE	-	-
deni_redox	mV	DOUBLE	-	-
deni_T_cond	°C	DOUBLE	-	-
deni_T_DO	°C	DOUBLE	-	-
deni_T_ISE	°C	DOUBLE	-	-
deni_T_redox	°C	DOUBLE	-	-
deni_TSS	g/L	DOUBLE	-	-
influent_K	mg/L	DOUBLE	-	-
influent_NH4_N	mg N/L	DOUBLE	-	-
influent_T_ISE	°C	DOUBLE	-	-

Abbildung 18: Mindsphere AssetManager - Aspect Konfiguration

5.4 Agenten

Als Agent wird ein Netzwerk-Objekt innerhalb der Mindsphere bezeichnet, der die Kommunikation von Applikationen außerhalb der Mindsphere ermöglicht. Agenten können ebenfalls im „Asset-Manager“ unter dem Punkt „Netzwerk“ angelegt werden, und sind bereits im vorgefertigten Typ „MindConnectLib“ (core.mclib) vorhanden.

The screenshot shows the Asset Manager interface. At the top, it displays 'Tenant Details' with a name 'segnoiot', timezone 'Europe/Berlin', and a lock icon indicating changes are allowed. On the right, a sidebar titled 'Ressourcen-Verbrauch' shows statistics: 8 of 50 assets, 3 of 5 types, and 1 of 10 agents. Below this, the 'Asset Model' section includes 'Assets' (description), 'Types' (description), and 'Aspects' (description). The 'Features' section contains 'Freigaben' (description) with a button to 'Netzwerkelemente anzeigen' (highlighted with a red box) and 'Netzwerk' (description) with a button to 'Netzwerkelemente anzeigen' (also highlighted with a red box). A navigation bar at the bottom left shows 'Konnektivität / Netzwerk-Verwaltung'.

Abbildung 19: Mindsphere AssetManager- Hauptansicht

The screenshot shows the 'Netzwerk-Verwaltung' (Network Management) interface. It displays a table of assets under 'Konnektivität / Netzwerk-Verwaltung'. The table has columns: Name, Type, Online Status, Firmware Version, Update Status, and Massenupdate. One row is shown: 'Agent' with Type 'core.mclib', Online Status 'Online' (green circle), Firmware Version '2022-06-15 14:49:41', and Update Status '-'. A blue edit icon is visible next to the row. A navigation bar at the bottom left shows 'Konnektivität / Netzwerk-Verwaltung'.

Abbildung 20: Mindsphere Netzwerk-Verwaltung

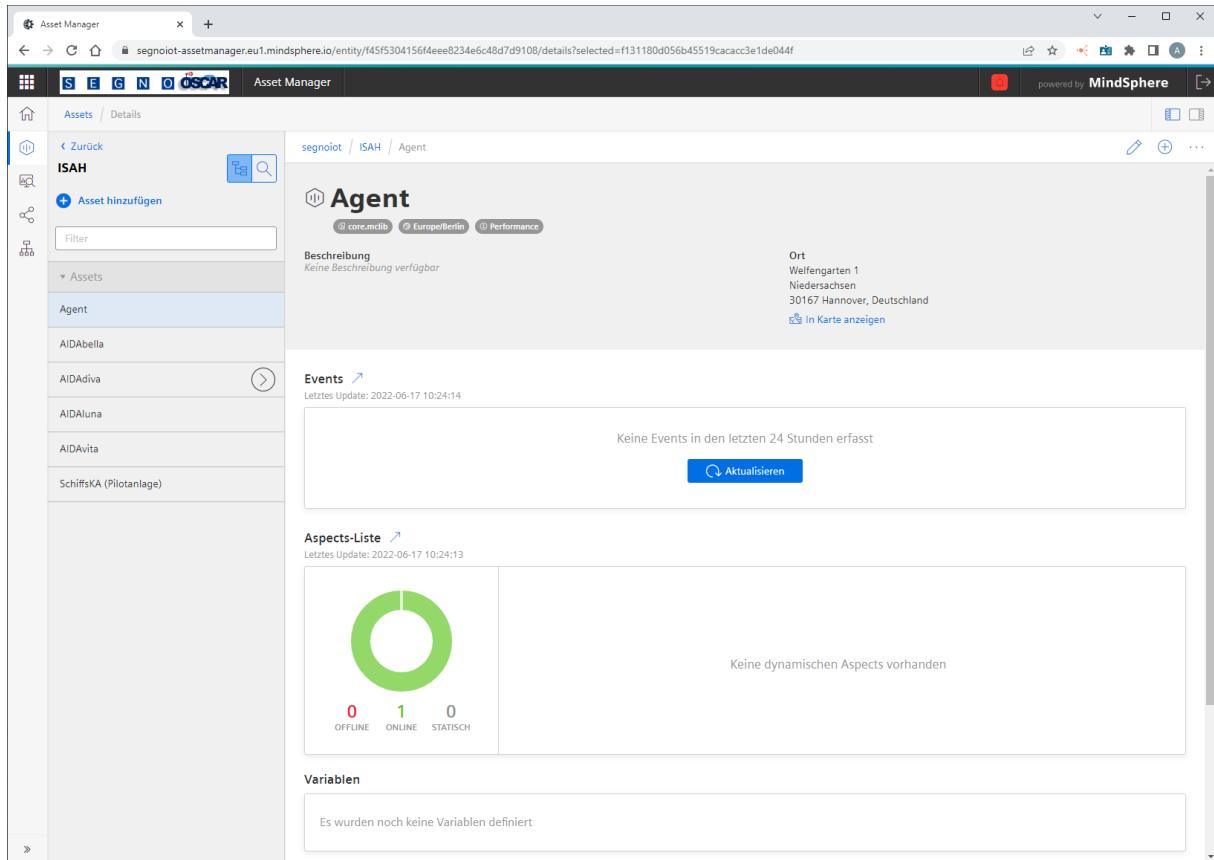


Abbildung 21: Mindsphere MindConnectLib Hauptansicht

Ein Agent muss nach dem Anlegen konfiguriert werden. Die Konfiguration wird dabei durch den IoT-Connector hochgeladen und kann anschließend mit dem gewünschten Asset verknüpft werden.

Um die Konfiguration eines IoT-Connectors anzulegen, müssen die Zugangsdaten (Client-ID, Client-Secret) des Agenten im IoT-Connector hinterlegt werden. Nach erfolgreicher Verbindung prüft der IoT-Connector die lokalen Datenpunkte mit denen des Agenten ab. Stimmen die Datenpunkte nicht überein, oder es existieren keine, so kann die Konfiguration in den Agenten der Mindsphere übertragen werden.

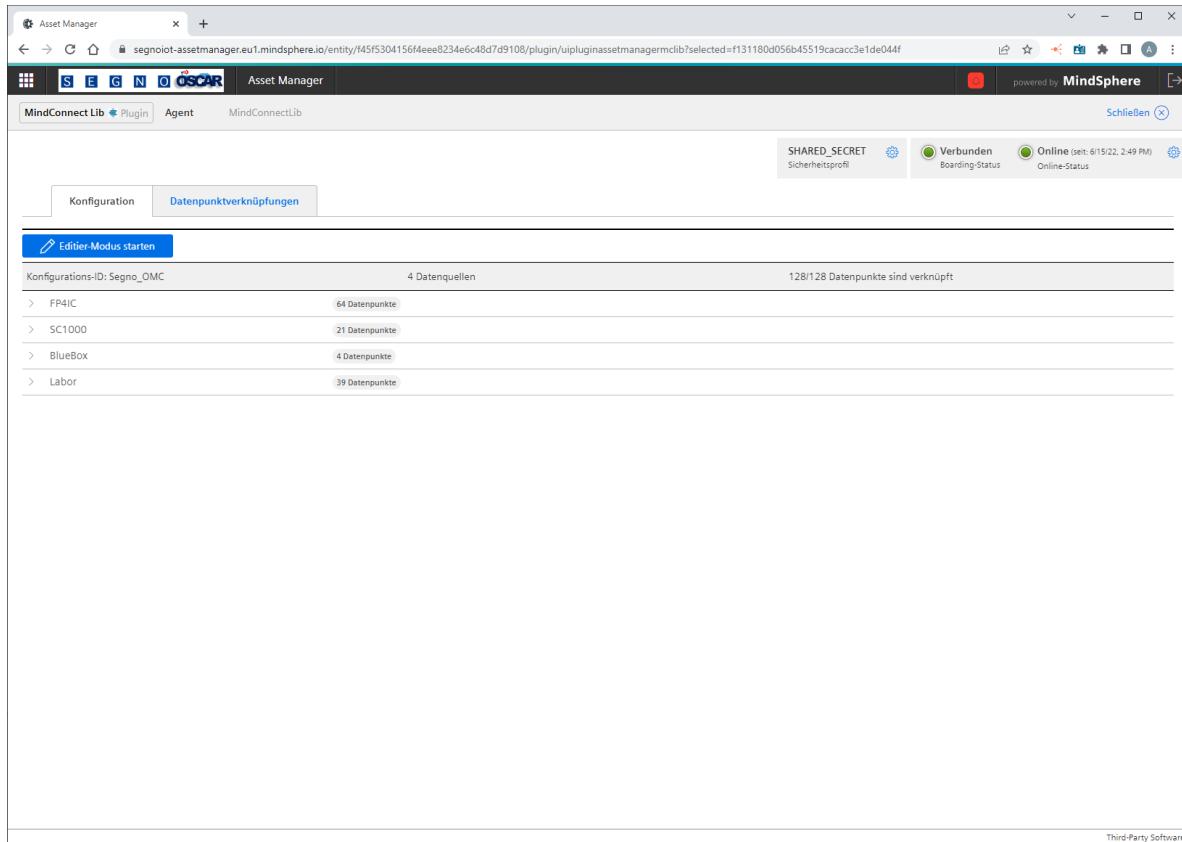
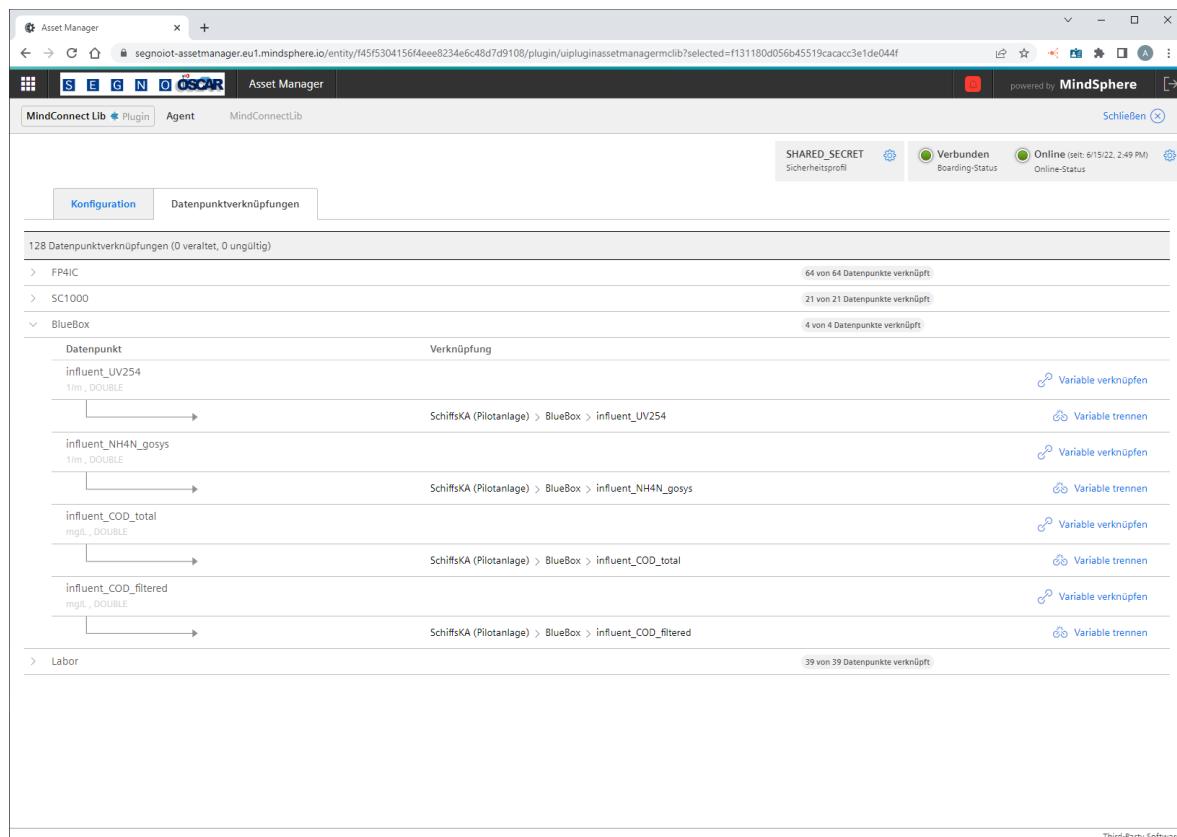


Abbildung 22: Mindsphere MindConnectLib Konfiguration

Nachdem die Konfiguration des IoT-Connector im Agenten verfügbar ist, müssen die Datenpunkte noch an das gewünschte Asset geknüpft werden. Dies geschieht über den Reiter „Datenpunktverknüpfungen“.



5.5 Fehleranalyse / Logging

Nachdem die Datenpunkten eines Agenten mit den Datenpunkten des jeweiligen Assets verknüpft worden sind, kann der IoT-Connector die Daten an den Mindsphere Agenten schicken, der diese wiederum im Asset ablegt. Sollten die Daten nicht im Asset ankommen, bietet die Mindsphere über die Applikation Agent Diagnostic die Möglichkeit, Agenten zu überwachen und die Vorgänge auszugeben.

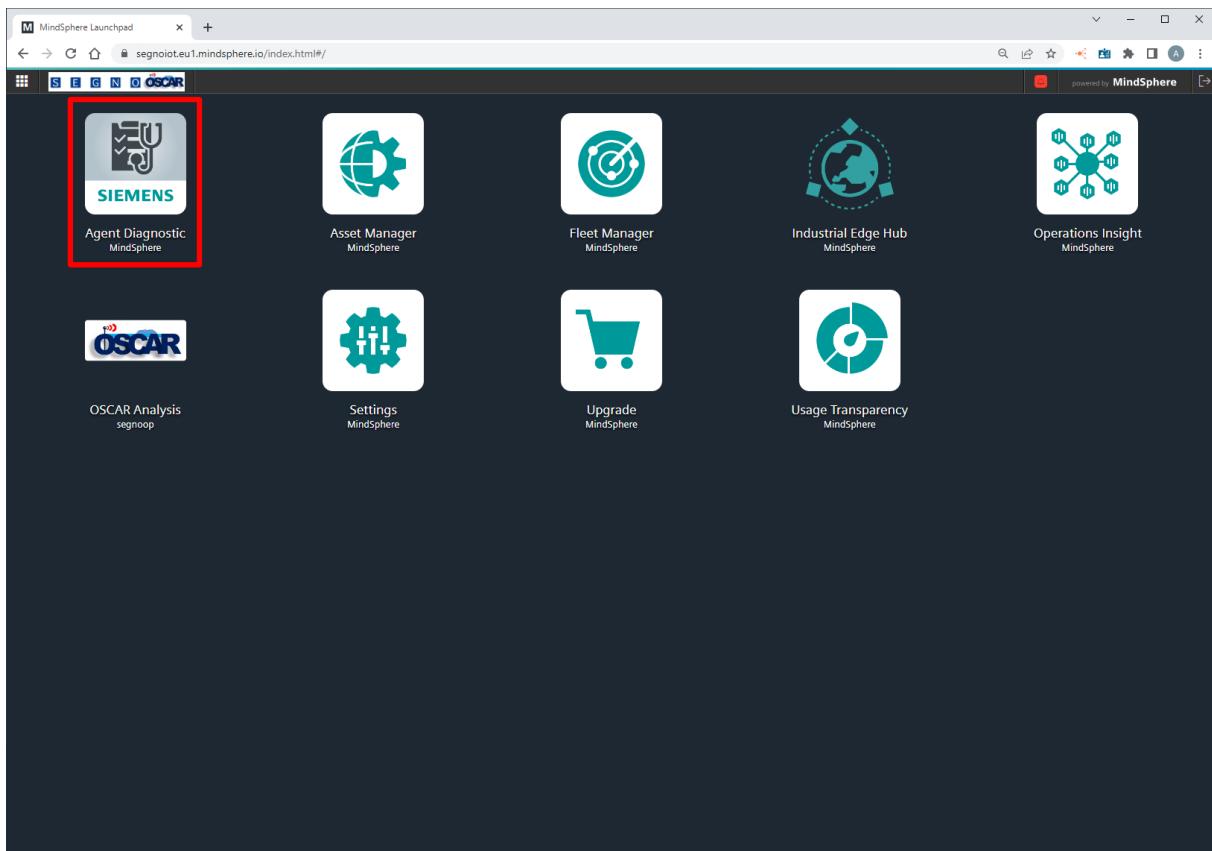


Abbildung 23: Mindsphere Hauptansicht Agent Diagnostic

Innerhalb der „Agent Diagnostic“ wird eine Liste alle Agenten aufgelistet. Agenten werden nicht dauerhaft überwacht, sondern müssen aktiviert werden. Nach Aktivierung werden diese über einen bestimmten Zeitraum aufgezeichnet.

The screenshot shows a web browser window titled "Agent Diagnostic" with the URL "segnoiot-agentdiagnostic.eu1.mindsphere.io/activations". The page has a header with the "S E G N O OSCAR" logo and "Agent Diagnostic". A sidebar on the left has "Setup" selected, with "Monitor" also available. The main content area is titled "Setup" and says "Here you can see the list of all the agents you have setup to monitor". It features a "Create new agent monitor" button and a search/filter section with "None" dropdown and "Search" input. Below is a table with columns: Name, Id, Status, Monitor, and Actions. One row is shown: "Agent Deutschland Niedersachsen Hannover" with Id "f131180d056b45519cacacc3e1de044f", Status "Off", Monitor "View Logs" (button highlighted with a red box), and Actions (checkbox and delete icon). The table includes pagination with "1 items found" and "Last Updated 10:07:30 AM GMT+02:00".

Abbildung 24: Mindsphere Agent Diagnostic

Mithilfe des Buttons „View Logs“ lassen sich nach Aufzeichnung der Logs alle Aktivitäten anzeigen. Innerhalb der Loganzeige kann über die Spalte „Severity“ nach Warnungen oder Fehlern gesucht werden.

Daten aus dem IoT-Connector werden nach gesammelter Zeit in einem Paket übertragen. Gibt es einen Fehler für einen einzigen Datenpunkt, werden alle anderen Datensätze verworfen. Nach Einrichtung der Kommunikation zwischen IoT-Connector und dem Agenten bietet es sich daher an, die Logmeldungen auf Probleme zu überprüfen.

The screenshot shows the 'Agent Diagnostic' interface from the 'Mindsphere' platform. The left sidebar has navigation links: 'Setup', 'Monitor' (which is selected), and 'OSCAR'. The main area is titled 'Monitor' and displays the logs of all active agents. A filter bar at the top allows filtering by 'None' or 'Search' and includes date range selection ('17.06.2022 - 17.06.2022'). The log table has columns: 'Agent', 'Source', 'State', 'Message', 'Severity', and 'Time'. The data shows various agent activities, such as TimeSeries upload completion and Exchange requests. At the bottom, there is a pagination control with buttons for 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, and a link to 'Last Updated 10:07:24 AM GMT+02:00'.

Agent	Source	State	Message	Severity	Time
Agent Deutschland Niedersachsen Hannover	TIMESERIES	FINISHED	[Finished] TimeSeries upload completed. <6> samplings and <24> properties in <1> requests. <0> values are dropped.	INFO	2022-06-16T11:00:18.929Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	PROCESSING	[Processing] TimeSeries Data arrived. Processing Started.	INFO	2022-06-16T11:00:18.829Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	FINISHED	[Finished] TimeSeries upload completed. <8> samplings and <382> properties in <1> requests. <0> values are dropped.	INFO	2022-06-16T11:00:18.823Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	FINISHED	[Finished] TimeSeries upload completed. <7> samplings and <125> properties in <1> requests. <0> values are dropped.	INFO	2022-06-16T11:00:18.822Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	PROCESSING	[Processing] TimeSeries Data arrived. Processing Started.	INFO	2022-06-16T11:00:18.746Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	PROCESSING	[Processing] TimeSeries Data arrived. Processing Started.	INFO	2022-06-16T11:00:18.745Z
Agent Deutschland Niedersachsen Hannover	EXCHANGE	ACCEPTED	[Accepted] Exchange request arrived.	INFO	2022-06-16T11:00:18.438Z
Agent Deutschland Niedersachsen Hannover	EXCHANGE	ACCEPTED	[Accepted] Exchange request arrived.	INFO	2022-06-16T11:00:18.375Z
Agent Deutschland Niedersachsen Hannover	EXCHANGE	ACCEPTED	[Accepted] Exchange request arrived.	INFO	2022-06-16T11:00:18.350Z
Agent Deutschland Niedersachsen Hannover	TIMESERIES	FINISHED	[Finished] TimeSeries upload completed. <6> samplings and <124> properties in <1> requests. <0> values are dropped.	INFO	2022-06-16T10:59:49.339Z

Abbildung 25: Mindsphere Agent Diagnostic Log

6 OSCAR Connect digitaler Zwilling

Konzeptionell sollte untersucht werden, ob der digitale Zwilling in einer standardisierten Form übertragbar als Softwaredienst in der Cloud-Umgebung lauffähig ist.

Ein digitaler Zwilling ist eine digitale Repräsentation Systems, die mithilfe von Simulationstechniken und -algorithmen erstellt wird. Der digitale Zwilling kann dazu verwendet werden, verschiedene Szenarien zu simulieren und zu testen, um die Leistung des Systems zu verbessern oder zu optimieren.

Der digitale Zwilling kann auch dazu verwendet werden, die Verfügbarkeit und Effizienz von Anlagen und Maschinen zu verbessern, indem mögliche Ausfälle vorhergesagt und verhindert werden. Er kann auch dazu beitragen, die Qualität von Produkten und Dienstleistungen zu verbessern, indem mögliche Fehlerquellen identifiziert und beseitigt werden.

Der digitale Zwilling wird häufig in der Fertigungsindustrie, im Maschinenbau und in der Energieerzeugung eingesetzt, aber auch in anderen Branchen, in denen die Leistung von Anlagen und Maschinen von großer Bedeutung ist.

Als digitaler Zwilling soll dabei die Simulationstechnologie ifakFAST (Framework für integrierte Automation und Simulationstechnologie) zum Einsatz kommen.

6.1 Funktionsanforderungen

- Daten aus dem Cloudsystem einlesen
- Simulationsanweisung

6.2 ifakFAST

ifakFAST ist eine offene Plattform und soll die Realisierung von Lösungen unterstützen, die auf die Überwachung und Betriebsoptimierung von technischen Prozessen konzipiert ist. Der Anwendungsschwerpunkt liegt hierbei im Bereich wasserwirtschaftlicher Anlagen und Prozessen, insbesondere Kläranlagen. Die ifakFAST-Plattform steht seit 2019 als Open Source zur Verfügung (<https://fast.ifak.eu>) und eignet sich insbesondere für die praktische Realisierung von modellbasierten Überwachungs- und Steuerungskonzepten mit SIMBA[#] („digitaler Zwilling“).

Für die Umsetzung stellt ifakFAST einen „Mediator Core“ zur Verfügung. Der Mediator-Kern ist für die Überwachung und Integration der Module verantwortlich und bietet eine Zeitreihendatenverwaltung und eine rollenbasierte Rechteverwaltung. Übergeordnete Funktionalität muss von Modulen bereitgestellt werden. Ein Modul ist eine Softwarekomponente mit einem spezifischen Konfigurationsmodell (typischerweise in Form einer XML-Datei), das einen Satz von Variablen definiert. Eine Variable stellt einen sich zur Laufzeit ändernden Wert mit Zeitstempel und Qualität dar, z. eine Messung oder ein Sollwert. Ein Modul kann Variablen und die Konfiguration anderer Module lesen und schreiben und kann bestimmte Dienste zur Verwendung durch andere Module bereitstellen.

6.3 Übertragbarkeit in eine Cloudumgebung

Während der Testphase sollte eine Möglichkeit geschaffen werden, Prozessdaten aus der Mindsphere zu exportieren, damit diese in die lokale Simulationssoftware importiert werden kann.

Für diesen Zweck stellt die Mindsphere eine Möglichkeit des Exports aller Prozessdaten eines Assets als CSV-Datei bereit.

Die exportierten Daten müssen daraufhin in das richtige Format gebracht, und in das Simulationssystem eingepflegt werden. Während der Testphase existierte dies nur auf einem lokalen System.

Die Simulationsergebnisse müssen auf diese Weise allerdings wieder manuell zurück in die Mindsphere importiert werden, damit mögliche Handlungsanweisungen über OSCAR-Analysis zurück in die jeweilige Steuerung transportiert werden können.

Dieser Prozess könnte dadurch automatisiert werden, indem die Simulation direkt in der Cloudumgebung stattfindet, und ein direkter Austausch mit OSCAR-Analysis die optimierten Verfahrensanweisungen automatisiert für die jeweiligen Steuerungen im Asset bereitstellt.

Da bisher keine Erfahrungen mit ifastFAST / SIMBA gemacht wurden, ist eine Implementierung in eine Cloudumgebung nur auf dessen Machbarkeit überprüft worden.

Der iFakFAST Mediator unterstützt die Programmiersprache .Net Core 3.1. Eine Umsetzung der Simulation könnte somit genauso wie die Applikation OSCAR-Analysis in einem CloudFoundry Container realisiert werden, und mit den Zeitreihendaten aus der Mindsphere verwendet werden. CloudFoundry, sowie die Mindsphere unterstützen neben der bisher genutzten Programmiersprache Node.JS ebenfalls .NET Core Applikation.

Es ist somit denkbar, die Simulation in einer neuen Applikation in einem CloudFoundry Container zu hinterlegen, und diese direkt an OSCAR-Analysis zu koppeln.

Innerhalb einer neuen Applikation kann die Simulierung anhand der gesammelten Zeitreihendaten aus den einzelnen IoT-Connectoren verwendet werden, und die Berechnungen und den daraus resultieren Optimierungen wieder zurück in die jeweiligen Assets übertragen werden.

Mit einer Erweiterung der Applikation OSCAR-Analysis könnte eine Analyse von empfohlenen Handlungsanweisungen direkt in dieser hinterlegt werden.

Weiterhin ist es denkbar, die optimierten Handlungsempfehlungen direkt an die Datenpunkte in den verschiedenen Assets für die angebundenen Steuerungen der IoT-Connectoren zu verknüpfen, sodass diese automatisiert an den IoT-Connector übertragen werden, ohne dass ein manuelles Eingreifen erforderlich ist.

7 OSCAR BigData

7.1 Funktionsanforderungen

- Datenreihen sicher ablegen
- Datenreihen aus IoT-Geräten in Empfang nehmen

7.2 Big Data

Big Data umfasst in der Regel sehr große Mengen an Daten, die sich aus unterschiedlichen Quellen und in verschiedenen Formaten ergeben. Diese Daten können strukturiert sein, das heißt, sie sind in Tabellenform organisiert und leicht zu verarbeiten, oder unstrukturiert, das heißt, sie sind in unterschiedlichen Formaten vorliegend und schwieriger zu analysieren.

Die Verarbeitung von Big Data erfordert in der Regel spezielle Werkzeuge und Technologien, die in der Lage sind, die enormen Mengen an Daten zu speichern und zu verarbeiten.

Das Ziel von Big Data ist es, aus den enormen Mengen an Daten wichtige Informationen und Erkenntnisse zu gewinnen, die für Unternehmen und Organisationen von Nutzen sein können. Dazu werden die Daten analysiert und mithilfe von speziellen Algorithmen und Tools ausgewertet.

Big Data kann auch dazu genutzt werden, Muster und Trends in den Daten zu erkennen, die für Menschen nicht offensichtlich sind. Diese Erkenntnisse können dann genutzt werden, um neue Geschäftsmöglichkeiten zu erschließen oder Prozesse zu optimieren.

7.3 Konzept

Im Rahmen der Entwicklung des IoT-Connector wurde von Anfang an darauf geachtet, die erfassten Daten in einem zuverlässigen System zu hinterlegen. Dazu gehört im Bereich der MindSphere die TimeSeries-Datenbank.

Bei der MindSphere handelt es sich um eine Cloud-basierte Plattform für das Internet der Dinge (IoT), die von Siemens entwickelt wurde. Mithilfe des Mindsphere IoT-Value Plan kann die Umsetzung des Vorhabens realisiert werden. Der MindSphere IoT Value Plan ist ein Angebot von Siemens, das Unternehmen dabei unterstützt, den Nutzen von IoT-Technologien zu maximieren. Der Plan bietet Zugang zu einer Vielzahl von IoT-Anwendungen und -Diensten, die es Unternehmen ermöglichen, ihre Betriebs- und Produktionsprozesse zu optimieren und die Effizienz und Qualität ihrer Produkte und Dienstleistungen zu verbessern.

Der MindSphere IoT Value Plan beinhaltet auch Tools und Ressourcen, die Unternehmen bei der Integration von IoT-Technologien in ihre Geschäftsprozesse unterstützen, sowie Schulungen und Support, um den Nutzen von IoT-Lösungen zu maximieren. Er ist für Unternehmen jeglicher Größe und Branche geeignet und bietet Zugang zu einer Vielzahl von IoT-Anwendungen und -Diensten, die für die Optimierung von Betriebs- und Produktionsprozessen, die Verbesserung der Kundenzufriedenheit und die Steigerung der Effizienz und Qualität von Produkten und Dienstleistungen entwickelt wurden.

7.4 Umsetzung

Die Umsetzung der Datenhaltung erfolgt auf Basis der MindSphere Timeseries. Diese ist Bestandteil des MindSphere IoT-Value Plans. Mittels des OSCAR IoT-Connector erfolgt die Erfassung der Daten vor Ort und der anschließende Versand der Datenreihen in das MindSphere Cloudsystem. Dort werden die empfangenen Datenreihen mit Datenpunkten aus zuvor definierten Assets verbunden.

Die Datenhaltung erfolgt dann innerhalb der definierten Assets und ist somit unabhängig von der eigentlichen Datenquelle.

Für jeden IoT-Connector kann dazu ein eigener Agent angelegt werden. Eine reibungslose Übertragung der Daten kann dadurch parallel sichergestellt werden.

Die Mindsphere bietet dafür eine skalierbare Lösung an. Dieses Modell der Lizenzierung bietet die Möglichkeit, auf die benötigten Anforderungen angepasst zu werden.

Für eine Analyse der Kapazitäten der Mindsphere stellt diese aber durch die „Usage Transparency“ in einer Übersicht dar, sodass auf benötigte Anpassungen schnell reagiert werden kann.

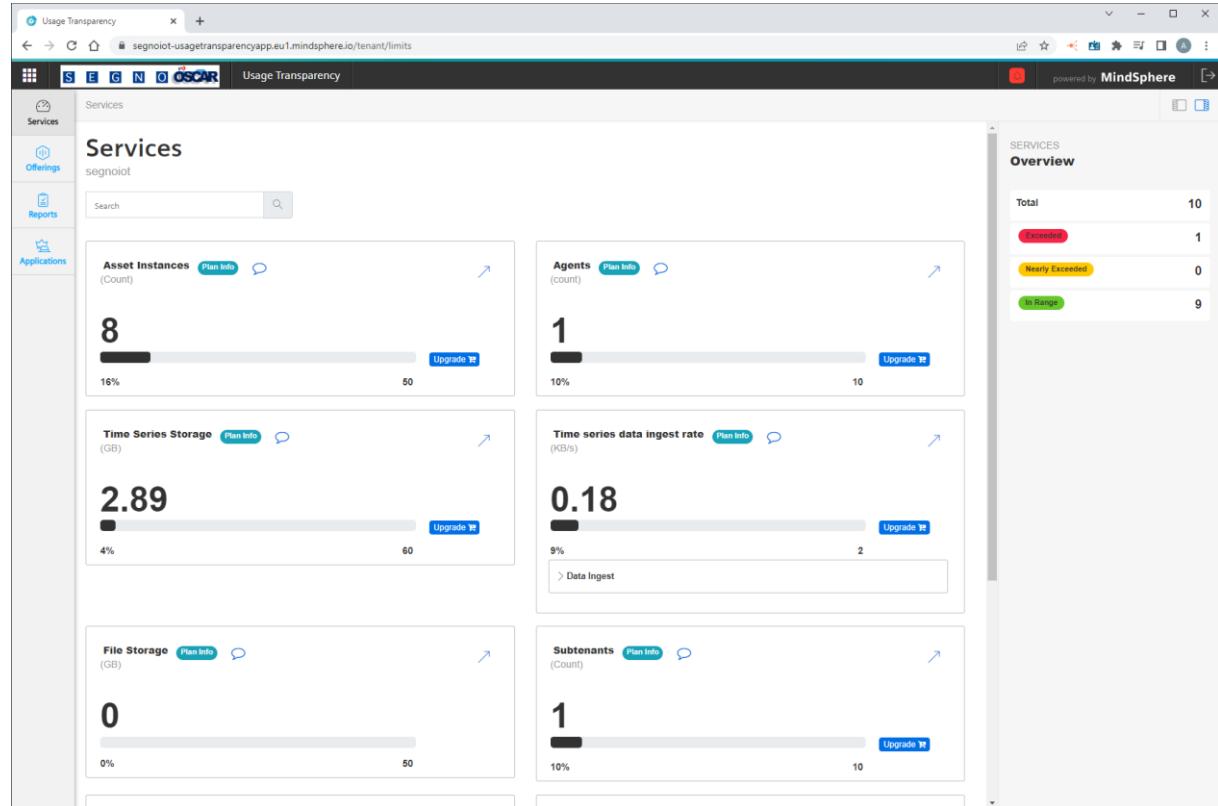


Abbildung 26:Mindsphere Service-Übersicht

8 OSCAR Analysis

Mit OSCAR Analysis soll eine Applikation innerhalb der Mindsphere entstehen, die als zentrale Software für die Datenauswertung sowie die Verwaltung von Sollwerten einzelner Konnektoren agiert.

8.1 Funktionsanforderungen

- Datenreihen in geeigneter Form darstellen
- Kurvendarstellungen definieren
- Kurvenzusammenstellungen speichern
- Dashboards definieren und erstellen
- Bild zur Visualisierung der Datenherkunft implementieren

8.2 Konzept

Es wird eine Applikation entwickelt, welche innerhalb eines Mindsphere Operator-Plans lauffähig und alle Funktionsanforderungen erfüllt.

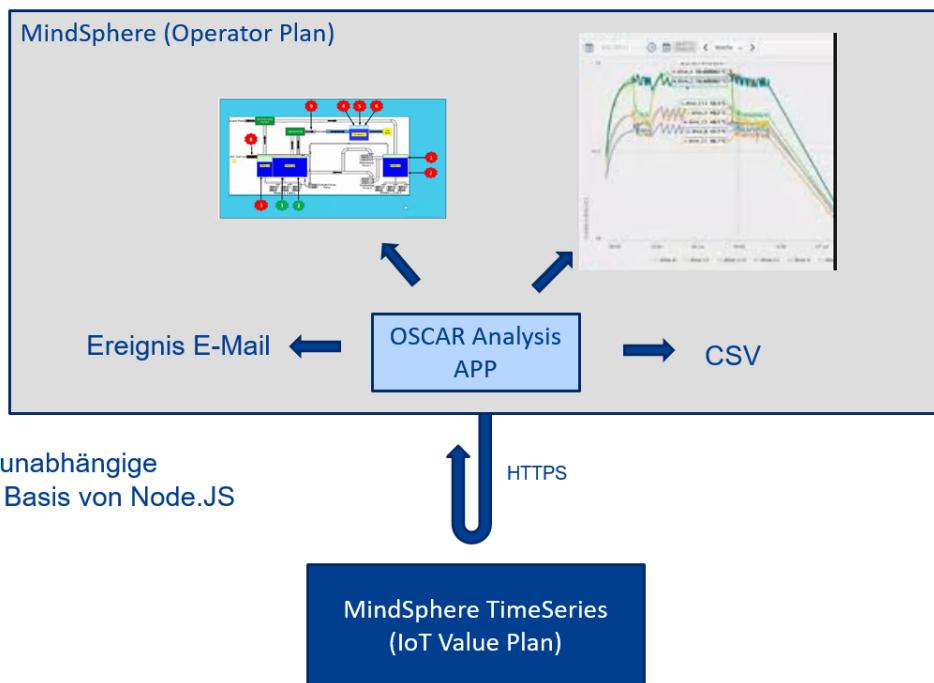


Abbildung 27: Mindsphere Operator - IoT Value Plan

8.3 Aufbau

Für eine übersichtliche Gestaltung von OSCAR-Analysis wurde die Applikation in mehrere Bereiche unterteilt. Zur Veranschaulichung folgt ein Screenshot der Übersicht, in dem die einzelnen Bereiche zur besseren Erläuterung farblich markiert wurden.

8.3.1 Layout

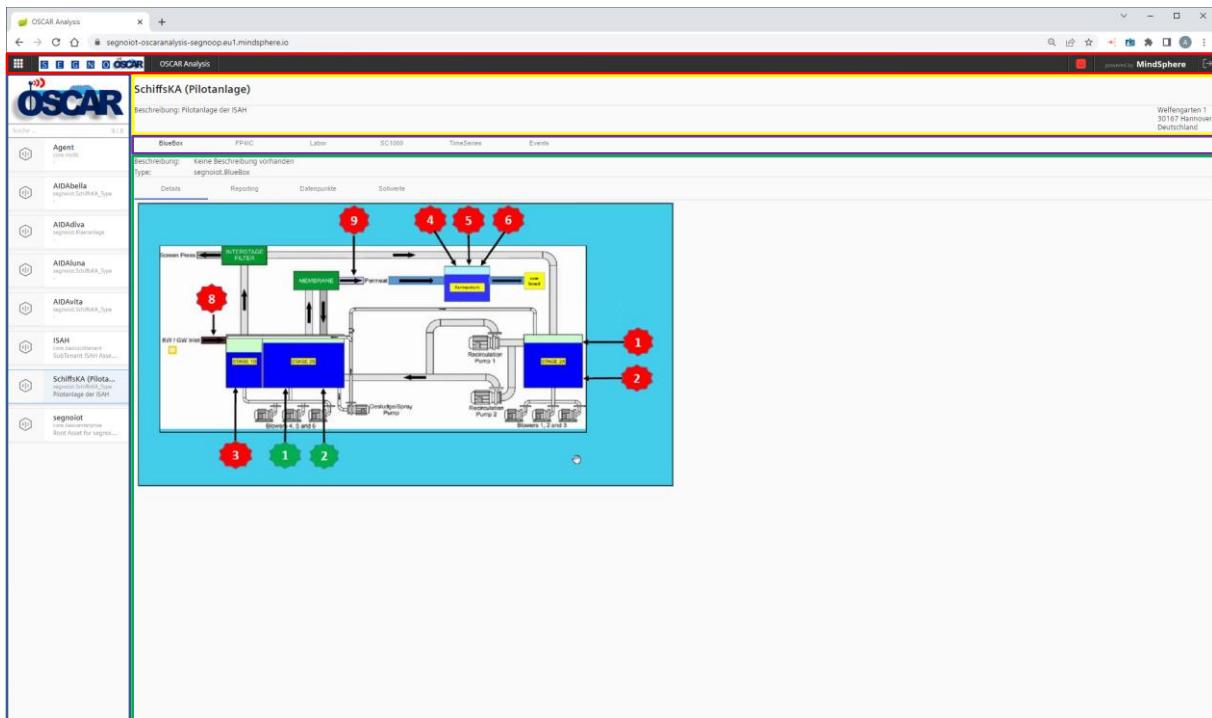


Abbildung 28: OSCAR-Analysis - Layout

8.3.2 Mindsphere OS-Bar

Im oberen roten gekennzeichneten Bereich der Applikation befindet sich die Mindsphere OS-Bar, die zu jeder Applikation in der Mindsphere dazugehört. Neben der Navigation in der oberen Linken Ecke befindet sich dort das Logo sowie der Name der Applikation. Im rechten Bereich befinden sich noch Informationen zur Mindsphere, sowie die Möglichkeit sich von dieser abzumelden.

8.3.3 Asset-Auswahl / Flotten-Management

Auf der Linken Seite, in blau gekennzeichnet, befindet sich die Auswahlliste aller Assets. Über diese kann zwischen jedem angebundenen IoT-Connector gewechselt werden. Für einen schnellen Zugriff gibt im oberen Bereich der Liste ein Suchfeld, über welches die Assets gefiltert werden können.

8.3.4 Asset-Ansicht

Wird ein Asset (IoT-Connector) ausgewählt, so werden die Daten im rechten Bereich der Applikation aktualisiert. Hier gibt es drei verschiedenen Bereiche, die gesondert zu betrachten sind.

Im gelb markierten Bereich wird das ausgewählte Asset angezeigt, inklusive einer Beschreibung und dem eingetragenen Standort.

Darunter, in lila gekennzeichnet, werden die Aspects des ausgewählten Assets in Form von Registerkarten dargestellt. Diese entsprechen den jeweiligen Konnektoren (angebundenen Steuerungen im IoT-Connector).

Zusätzlich sind unabhängig von den Aspects immer die Einträge „TimeSeries“, sowie „Events“ zu finden.

Während die Aspects jeweils nur die Daten einer einzelnen Steuerung darstellen, können über die letzten beiden Reiter die Informationen aller Steuerungen des IoT-Connectors dargestellt werden.

Möchte man die Daten einer einzelnen Steuerung des ausgewählten IoT-Connector sehen, so gibt es für jede Steuerung einen eigenen Reiter mit dem gewählten Namen des Aspects.

Nach Anwahl einer Steuerung wird der Bereich darunter, in grün gekennzeichnet, mit neuem Inhalt gefüllt. Hier gibt es vier weitere Reiter.

- Details
- Reporting
- Datenpunkte
- Events

8.3.4.1 TimeSeries

Über den Punkt „TimeSeries“ wird ein Liniendiagramm angezeigt, über den die Datenpunkte aller Steuerungen dargestellt werden können. Dazu ist im folgenden Screenshot im rechten Bereich eine Liste aus allen Steuerungen und der dazugehörigen Datenpunkte zu sehen. Durch das An- und Abwählen können mehrere Werte aus verschiedenen Steuerungen dargestellt werden. Oberhalb vom Graphen ist ein Web-Control zur Auswahl des Zeitbereichs.



Abbildung 29: OSCAR-Analysis - TimeSeries

8.3.4.2 Events

Im Reiter Events können alle Events vom ausgewählten Asset (IoT-Connector) eingesehen werden. Events innerhalb der Mindsphere können nur einem Asset zugeordnet werden, daher sind diese nicht in der Anzeige der einzelnen Aspects zu finden.

Events werden dazu genutzt, um Ereignisse innerhalb eines IoT-Connectors in der Mindsphere sichtbar zu machen.

Derzeit gibt es zwei verschiedene Ereignisse, die durch den IoT-Connector generiert werden können.

- Sollwert erfolgreich in Steuerung geschrieben (Information)
- Sollwert konnte nicht in die Steuerung geschrieben werden (Error)

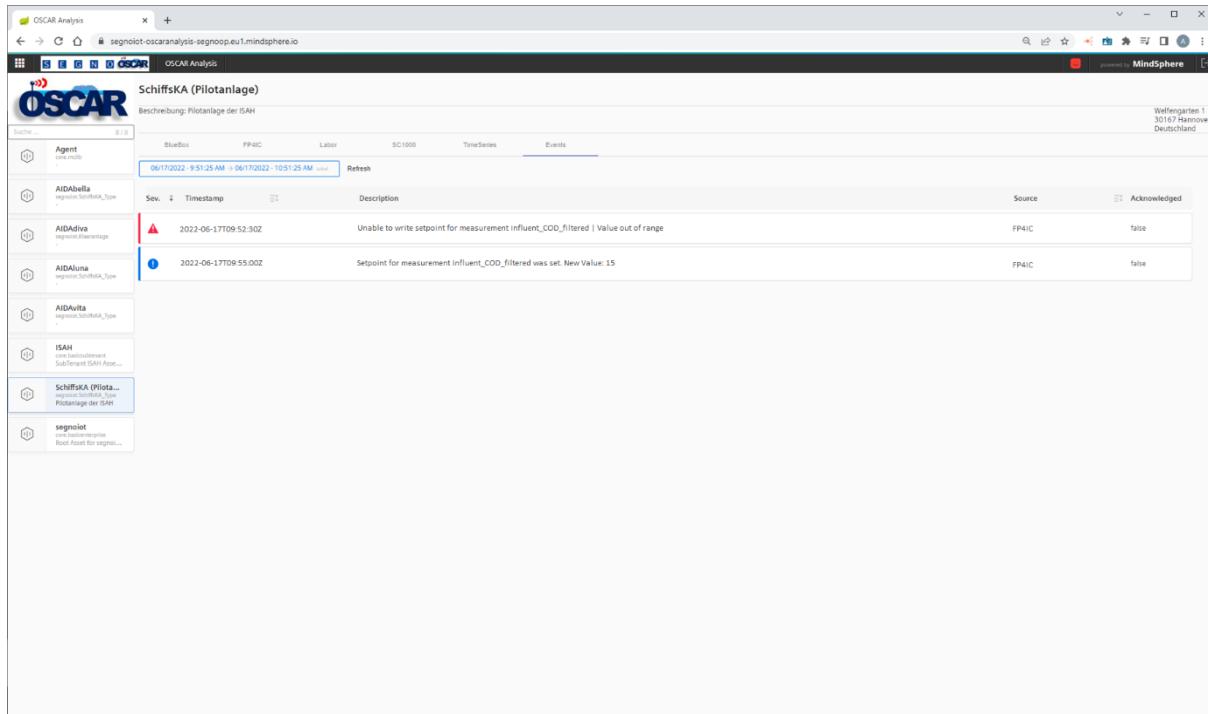


Abbildung 30: OSCAR Analysis - Events

8.3.4.3 Details

Unter dem Reiter „Details“ ist ein statisches Bild der Anlage hinterlegt, wie bereits im ersten Screenshot zu sehen.

8.3.4.4 Reporting

Unter dem Reiter „Reporting“ wird eine tabellarische Ansicht aller Datenpunkte mit dem zuletzt übermittelten Wert, sowie dem dazugehörigen Zeitstempel dargestellt.

Name	aktueller Wert	Zeilentyp	Einheit
infuent_COD_filtered	233.34	17.06.2022 13:11:57	mg/L
infuent_COD_total	787.26	17.06.2022 13:11:57	mg/L
infuent_NH4N_gosys	110.61	17.06.2022 13:11:57	µm
infuent_UV254	500	17.06.2022 13:11:57	µm

8.3.4.5 Datenpunkte

Unter dem Reiter „Datenpunkte“ ist eine Auflistung aller Datenpunkte sowie deren Einheiten zu finden. Hier besteht die Möglichkeit, einen Datenpunkt als Sollwert zu markieren. Anschliessend kann für diesen Datenpunkt im Reiter „Sollwerte“ ein Sollwert gesetzt werden.

Sollwert	Name	Einheit
<input checked="" type="checkbox"/>	infuent_COD_filtered	mg/L
<input type="checkbox"/>	infuent_COD_total	mg/L
<input checked="" type="checkbox"/>	infuent_NH4N_gosys	µm
<input type="checkbox"/>	infuent_UV254	µm

8.3.5 Sollwerte

Unter dem Reiter „Sollwerte“ können Sollwerte für alle Datenpunkte gesetzt werden, die unter dem Reiter „Datenpunkte“ als Sollwerte definiert wurde

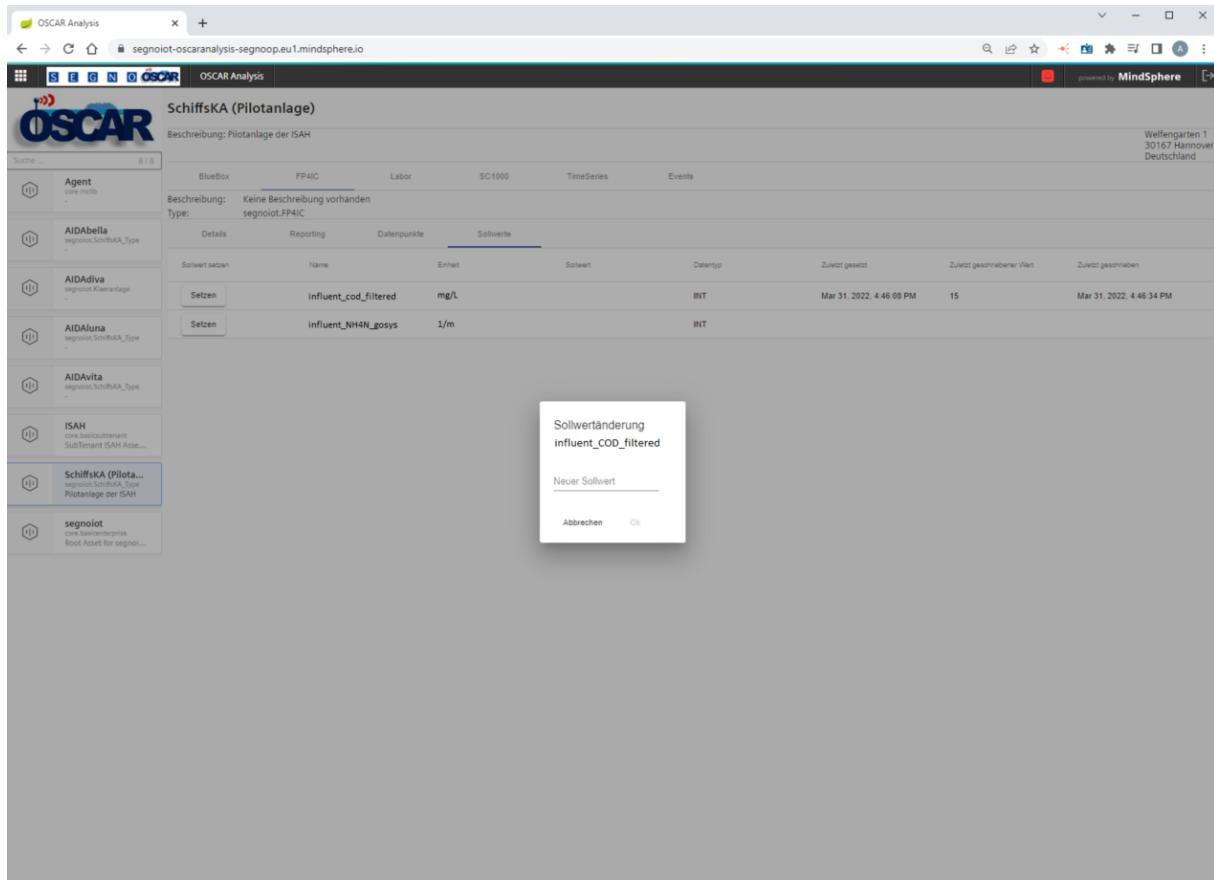
Wird für einen Datenpunkt ein Sollwert hinterlegt, so wird vom IoT-Connector beim nächsten Upload der Prozessdaten der Sollwert ausgelesen aus der Mindsphere ausgelesen und in die Steuerung zurückgeschrieben.

Bei jedem Setzen eines Sollwertes in der Steuerung durch den IoT-Connector wird dabei der Zeitstempel für „Zuletzt gesetzt“ hinterlegt.

Nachdem der IoT-Connector den Sollwert in die Steuerung geschrieben hat, wird dieser innerhalb der Mindsphere wieder zurückgesetzt, und es werden der Zeitstempel des Schreibvorgangs, sowie der gesetzte Wert hinterlegt.

	Name	Einheit	Sollwert	Datentyp	Zuletzt gesetzt	Zuletzt geschriebener Wert	Zuletzt geschrieben
Setzen	influent_COD_filtered	mg/L		INT	Mar 31, 2022, 4:46:09 PM	15	Mar 31, 2022, 4:46:34 PM
Setzen	influent_NH4N_gosys	1/m		INT			

Durch Klicken auf den Button „Setzen“, öffnet sich ein Popup, und der Sollwert kann eingetragen werden. In Absprache mit der LUH kann jeder Datenpunkt als Sollwert hinterlegt werden. Die Plausibilitätsprüfung findet innerhalb der Steuerung statt.



Wird ein Sollwert durch den IoT-Connector in die Steuerung geschrieben, wird ein Event generiert. Konnte ein Sollwert nicht geschrieben werden, wird ebenfalls ein Event generiert.

8.4 Umsetzung

Die Applikation OSCAR-Analysis wird ebenfalls wie der IoT-Connector in Node.JS entwickelt. Als Frontend kommt dabei das Framework „Angular“ zum Einsatz. Die Entwicklung dieser App erfolgt im Bereich des Developer-Plans der Mindsphere. Bei dem Developer-Plan handelt es sich um einen Mindsphere-Tenant, in dem die Entwicklung einer Applikation stattfindet. Neben dem Entwickeln und Testen können hier ebenfalls die Benutzerrechte für einzelne Benutzer oder Gruppen vergeben werden.

Nach Abschluss der Entwicklung kann eine Applikation in den Operator-Tenant verschoben werden, von dem aus diese an verschiedene IoT-Tenants, also den Endanwendern, weitergegeben werden kann. Bevor die Applikation in den Operator-Tenant übertragen werden kann, muss diese erst einer automatisierten Prüfung durch die Mindsphere unterzogen werden.

Im Rahmen der neuen OSCAR-Analysis-App soll u.a. eine bildliche Darstellung der Messpunkte hinterlegt werden, um die Zuordnung der Messwerte besser darzustellen. Die soll zum einen aus einem statischen Bild bestehen und zum anderen aus den jeweiligen letzten gemessenen Werten.

Weiterhin soll eine Funktion implementiert werden, mit der zu definierende Ereignisse ausgewertet werden können und das Ergebnis als Handlungsanweisung an einen E-Mail-Empfänger gesendet werden.

Hierzu ist die Möglichkeit zur Definition von Ereignissen zu erfolgen. Wie diese Ereignisse aussehen können, ist in Zusammenarbeit mit LUH zur definieren. Innerhalb der OSCAR-Analysis-App soll ebenfalls eine graphische Darstellung erstellt werden. Für Die Darstellung bieten sich Kurven-Diagramme an, welche die aufgelaufenen Timeseries-Daten abbilden. Wie schon beim IoT-Connector

wird als Entwicklungsumgebung Visual Studio Code verwendet. Hierbei handelt es sich um eine kostenlose Programmierumgebung von Microsoft, welche allen Anforderungen entspricht.

8.5 Lokale Entwicklung

Die Applikation „OSCAR-Analysis“ wurde lokal auf einer virtuellen Maschine entwickelt. Hierbei stellte sich die Herausforderung, dass die benötigten Daten des Mindsphere-Tenants, wie zum Beispiel Assets, Aspects, Events und Timeseries-Daten nicht verfügbar sind. Diese Informationen werden über die Mindsphere-API geliefert und sind nur verfügbar, wenn der Tenant im Browser angemeldet ist. Bei einer Applikation, die innerhalb der Mindsphere läuft, wird die Authentifizierung gegenüber der API über den Http-Header mitgesendet. Dies entfällt bei einer lokalen Entwicklung.

Um dennoch an die Daten der Mindsphere zu kommen und die Applikation lokal zu entwickeln, gibt es das CLI-Tool „mc dev-proxy“ aus dem „MindConnect-NodeJS“ Paket. Dieses ermöglicht es, mithilfe der Session sowie dem XSRF-Token einen lokalen Proxy Server zu öffnen, der Anfragen an die Mindsphere entgegennimmt und weiterleitet. Diese werden dann um die Authentifizierung gegenüber der API erweitert.

Hierzu sind einige Voraussetzungen notwendig. Zuerst muss eine Applikation innerhalb des Developer-Tenants deployed werden, welche über die entsprechenden Rechte verfügt die gewünschten Daten abzufragen. Bei dieser Applikation kann es sich um eine einfache Frontend-Applikation ohne jegliche Logik handeln. Wird in einer Browser-Session die Applikation geöffnet, so können über die Browser-Erweiterung „MindSphere Authentication Helper Extension“ diese Header-Informationen extrahiert werden. Hinterlegt man diese beiden Informationen als Parameter im mc dev-proxy, so wird jede Anfrage an die Mindsphere um die Credentials erweitert, und die lokale Entwicklung kann gestartet werden. Dieser Schritt ist nach jeder neuen Anmeldung an der Mindsphere zu wiederholen, da die Credentials sich erneuern.

8.6 Erstellung des OSCAR-Analysis Projekt

Nachdem die Entwicklungsumgebung vorbereitet ist, kann mit der Entwicklung der Applikation begonnen werden.

8.6.1 Framework - Angular

OSCAR-Analysis soll als Web-Applikation innerhalb der Mindsphere laufen. Aus diesem Grund wurde sich für das Framework Angular entschieden. Es handelt sich hierbei um ein Frontend-Framework von Google für die Erstellung von Webseiten und unterliegt der MIT-Lizenz. Somit ist eine Verwendung kostenfrei. Zudem ist eine Übertragung von Angular Applikationen

Es bietet sich dabei an, Angular global auf dem Entwicklungssystem zu installieren. Das hat den Vorteil das Angular-Befehle global über die Kommandozeile verfügbar sind.

Die Installation ist über den „node package manager“ (NPM) mit folgendem Befehl möglich:

- npm install -g @angular/cli

Der Parameter -g steht hierbei für global.

8.6.2 Erstellung des Projekts

Ein Grundgerüst einer Angular Applikation kann über folgenden angular-Befehl innerhalb der Kommandozeile erstellt werden.

- ng new {ApplicationName}

Es folgt ein Wizard, über den die Applikation für bestimmte Funktionen und Styling erweitert werden kann.

```
PS C:\mindsphere-apps> ng new oscar-analysis
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? SCSS [ https://sass-lang.com/documentation/syntax#scss ]
```

Abbildung 31: Anlegen eines neuen Angular-Projekts

Die erste Abfrage definiert das Routing, welches für das Aufrufen der einzelnen Unterseiten der Webseite zuständig ist. Wird diese Option aktiviert, existiert ein Standardrouting innerhalb der Applikation, die lediglich für weitere Anzeigen erweitert werden muss.

Als zweiten Parameter kann wahlweise zwischen CSS und SCSS gewählt werden. Damit wird die Stylesheet-Sprache definiert, mit der das Design der Angular Applikation umgesetzt wird. Bei CSS handelt es sich um „Cascading Style Sheets“, während es sich bei SCSS um eine Erweiterung von CSS handelt, welche zusätzliche Funktionen und Vereinfachungen mit sich bringt.

Nach Abschluss des Wizards ist ein Grundgerüst für eine Angular Applikation angelegt.

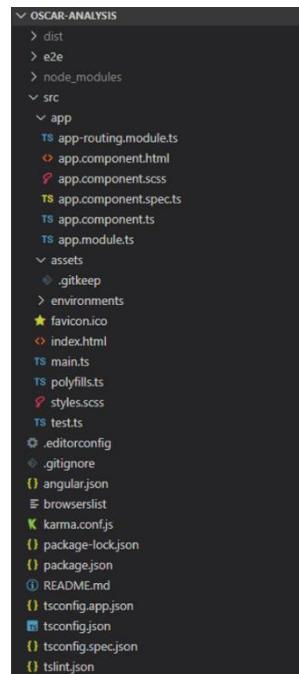


Abbildung 32: Grundstruktur Angular Projekt

8.6.2.1 Hinzufügen der Mindsphere-OS Bar zur Applikation

Applikation die innerhalb der Mindsphere laufen, müssen die Mindsphere Titelzeile, auch OS BAR genannt, implementieren. Die Titelzeile enthält Informationen zum Tenant, in der die Applikation läuft, Navigationselemente der Mindsphere sowie die Möglichkeit, sich von dieser Abzumelden. Informationen zu geplanten Änderungen oder bevorstehenden Wartungen sind ebenfalls dort zu finden.

Im folgenden Screenshot ist die OS-Bar ist rot gekennzeichnet.

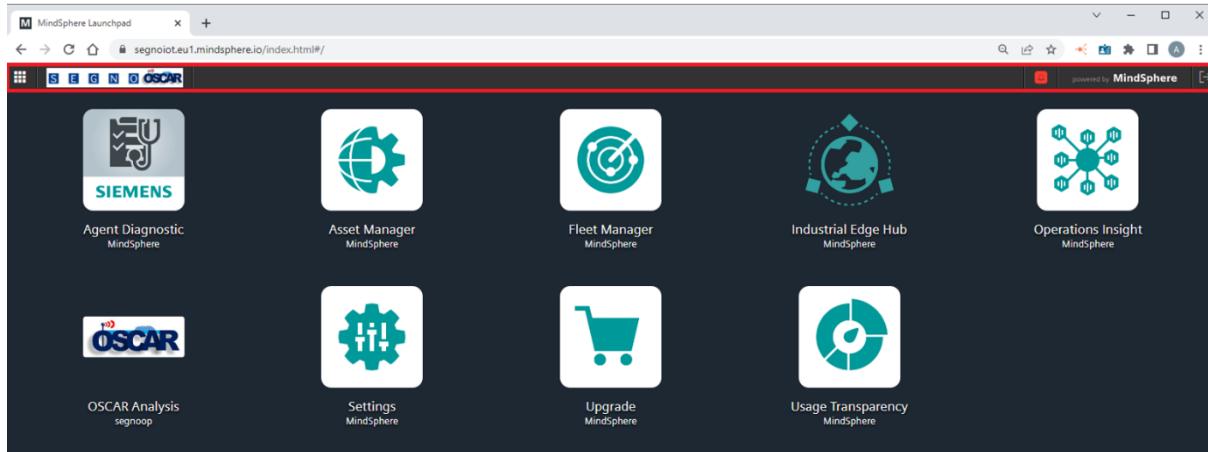


Abbildung 33: Mindsphere OS-Bar

Um die OS Bar einzubinden, wird lediglich ein Skript benötigt, welches in der Hauptansicht der Applikation einzubinden ist. Dieses gibt es als Vorlage in der Mindsphere-Developer Dokumentation zu finden.

(Quelle: <https://design.mindsphere.io/osbar/get-started.html> Stand 2022-05-30)

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>OSCAR Analytics</title>
  <base href="/">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" type="image/x-icon" href="favicon.ico">
  <script src="https://static.eu1.mindsphere.io/osbar/v4/js/main.min.js"></script>
</head>
<body>
  <div id="_mdspcontent">
    <app-root></app-root>
  </div>
  <script>
    _mdsp.init({
      title: 'OSCAR Analytics',
      appId: '_mdspcontent',
      appInfoPath: '/static/app-info.json',
      polyfills: {
        es5adapter: true,
        promise: false,
        webcomponents: true
      }
    })
  </script>
</body>
</html>
```

Abbildung 34: Einbinden der OS Bar in eine Webseite

8.6.3 Applikation lokal starten

Während der Entwicklung kann die Applikation lokal ausgeführt werden. Um Angular Applikationen zu testen, bietet das Angular Framework einen Kommandozeilen-Befehl, durch den die Webseite lokal auf der Entwicklungsmaschine zu starten.

Dazu kann über folgenden Befehl ein Webserver lokal gestartet werden, welcher die Applikation bereitstellt.

- ng serve

Anschließend lässt sich die Webseite über die Adresse „localhost:4200“ im Browser aufrufen. Sofern nicht anders definiert, sind bereitgestellte Webseiten durch den Angular Webserver auf Port 4200 erreichbar.

Beim Erstellen eines neuen Angular-Projektes ist ebenfalls eine vorgefertigte Webseite enthalten, mit dem man den Zugriff testen kann. Dieses Dokument (app.component.html), kann dann für die weitere Verwendung entfernt, und durch eigene Anzeigen erweitert werden.

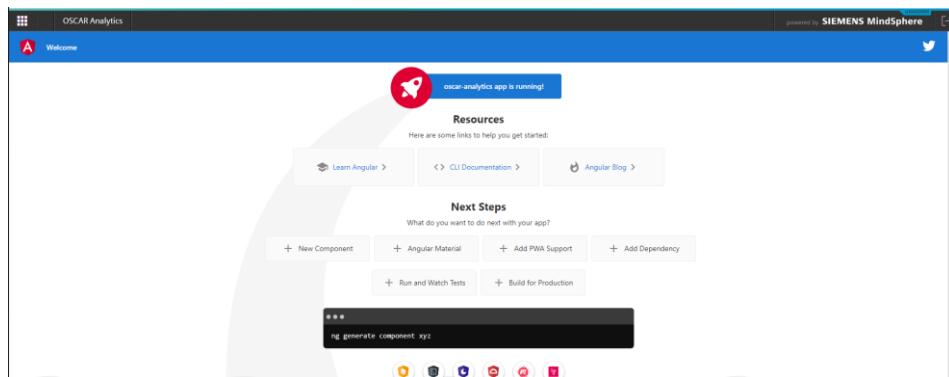


Abbildung 35: Ansicht eines blanken Angular Projekts

8.6.4 Implementierung von Mindsphere Web Komponenten

Bei der Entwicklung von OSCAR-Analysis wurden mehrere „Web Components“ der Mindsphere verwendet. Es handelt sich um bereits vorgefertigte Web-Komponenten, die einen hohen Wiederverwendungsgrad haben und daher in vielen Applikationen zum Einsatz kommen.

Für die Nutzung von Mindsphere Web Komponenten wird vom aufrufenden Tenant eine entsprechende Mindsphere Rolle benötigt.

Im Falle von OSCAR-Analysis wurde sich folgender Web-Components bedient:

Komponente	Benötigte Mindsphere Rolle	Beschreibung
Asset View	msdp:core:assetmanagement.xxx	Navigationselement, um durch Assets innerhalb des Tenants zu navigieren
Aspect Variable Selector	Msdp:core:assetmanagement.xxx	Navigationselement, um durch Aspects des ausgewählten Assets zu navigieren
EventView	Msdp:core:em.eventcreator	Anzeigeelement für Events innerhalb eines Assets
TimeSeries Chart	msdp:core:iot.timUser	Graph für Timeseries Daten von ausgewählten Datenpunkten eines Aspects
Date Time Range Picker	Keine Rolle benötigt	Navigationselement, für die Zeiteingrenzung innerhalb des TimeSeries Chart

Quelle: <https://documentation.mindsphere.io/MindSphere/resources/mindsphere-webcomponents/index.html> (Stand 2022-05-30)

8.7 Zugriffsrechte

Während der Entwicklung sowie nach Veröffentlichung einer Applikation innerhalb der Mindsphere muss der angemeldete Benutzer für jede Aktion die entsprechenden Rechte besitzen. Dafür steht innerhalb der Mindsphere eine Rechteverwaltung zur Verfügung, die jedem Mindsphere-Operator einzeln zugewiesen werden kann. Dies kann von der Anzeige der Assets, den Timeseries-Daten oder den Events sowie vielen weiteren Einstellungen separat pro Nutzer oder Gruppe eingestellt werden.

Innerhalb von OSCAR-Analysis findet eine Unterteilung zwischen Benutzern und Administratoren statt. Benutzer dürfen alle Daten einsehen, aber keine Änderungen tätigen. Administratoren hingegen ist es ebenfalls gestattet, Tags der einzelnen Aspects als Sollwerte zu definieren und diese auch zu setzen.

So kann einem Mindsphere Tenant später die Rolle eines Benutzers oder Admins zugewiesen werden. Erst nachdem einen Mindsphere Tenant eine dieser beiden Rollen zugewiesen wurde, erscheint die OSCAR-Analysis Applikation in der Mindsphere Hauptseite.

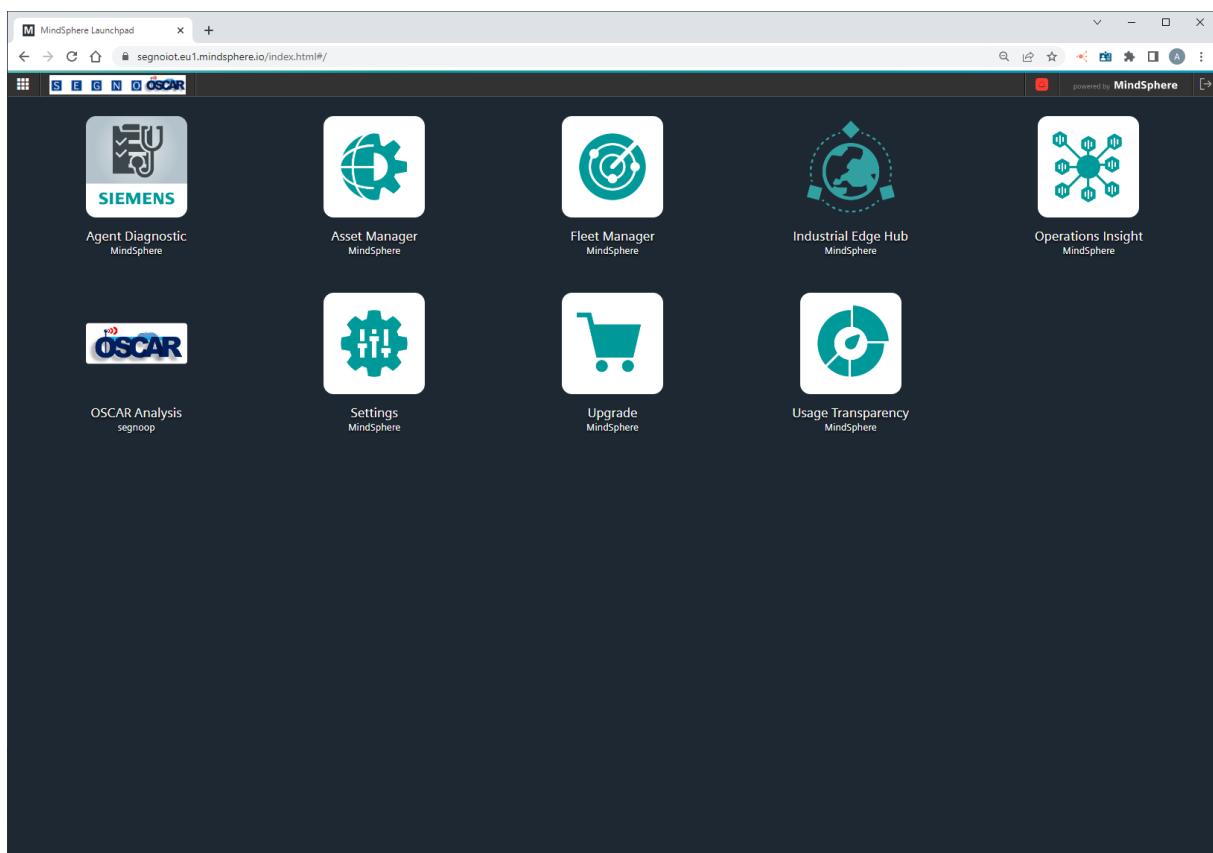


Abbildung 36: OSCAR Analysis auf der Mindsphere Übersichtsseite

Das Erteilen oder Verwehren von Zugriffsrechten einer Applikation ist nur möglich, wenn diese noch nicht an den Operator-Tenant zugewiesen wurde und diese somit noch „in Entwicklung“ ist. Ein nachträgliches Ändern der Zugriffsrechte für eine Applikation ist aus Sicherheitsgründen nicht erlaubt.

8.7.1 Scopes & Roles

Die Zugriffsrechte einer Mindsphere-Applikation können über einzelne „Scopes“ als auch über „Roles“ zusammengefasst werden und können den Tenant-Usern bzw. Gruppen zugeordnet werden. Bei einem Scope handelt es sich um das Recht einer bestimmten Aktion, die erteilt oder verweigert werden kann. Mehrere Scopes können zudem zu einer „Role“ hinzugefügt werden und an Benutzer oder Gruppen zugeordnet werden.

Je nach API-Zugriff werden verschiedene Rechte benötigt. Diese Zuordnung geschieht im Development-Tenant unter „Developer-Cockpit“ – „Authorization Management“ für die jeweilige Applikation.

8.8 Benötigte Rechte

Für die Verwendung von OSCAR-Analysis sind einige Rechte erforderlich, andere wiederum optional. Es folgt eine Auflistung der benötigten Rechte.

Role	admin	user
mdsp:core:assetmanagement.admin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
mdsp:core:assetmanagement.reporter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:em.eventcreator	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:em.eventmanager	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:iot.filAdmin	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:iot.filUser	<input type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:iot.timAdmin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
mdsp:core:iot.timUser	<input type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:iot.tsUser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
mdsp:core:tm.tenantAdmin	<input checked="" type="checkbox"/>	<input type="checkbox"/>
mdsp:core:tm.tenantUser	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abbildung 37: Benötigte Rechte für OSCAR Analysis

Services	Nomenklatur	Beschreibung
Asset Management	mdsp:core:assetmanagement.admin	Mit der Admin-Rolle können Benutzer Assets, Asset-Typen, Aspect-Typen und Bilder im Asset Management Service erstellen, lesen, aktualisieren und löschen.
	mdsp:core:assetmanagement.reporter	Mit der Berichterstatter-Rolle können Benutzer im Asset Management Service Assets, Asset-Typen, Aspect-Typen und Dateien lesen.

Event Management	mdsp:core:em.eventcreator	Diese Rolle gestattet vollständigen Zugriff zum Erzeugen von Ereignissen (Erzeugen, Lesen und Aktualisieren) im Event Management-System.
	mdsp:core:em.eventmanager	Diese Rolle gestattet Zugriff zum Verwalten sämtlicher Inhalte im Event Management-System.
IoT File Service	mdsp:core:iot.filAdmin	Diese Rolle gestattet Lese-, Schreib- und Löschzugriff auf Dateien.
	mdsp:core:iot.filUser	Diese Rolle gestattet Lesezugriff auf Dateien.
IoT Time Series	mdsp:core:iot.timAdmin	Diese Rolle gestattet Lese-, Schreib- und Löschzugriff auf Zeitreihenservices (Time Series, Aggregations, Subscription und Export).
	mdsp:core:iot.timUser	Diese Rolle gestattet Lesezugriff auf Time Series.
IoT Time Series Aggregates	mdsp:core:iot.tsaUser	Diese Rolle gestattet Zugriff auf Time Series Aggregates.
Tenant Management Service	mdsp:core:tm.tenantAdmin	Diese Rolle gestattet vollen administrativen Zugriff auf den jeweiligen Tenant.
	mdsp:core:tm.tenantUser	Diese Rolle gestattet Berechtigungen für Standard-Tenant-Benutzer.

8.8.1 Application Credentials

Mithilfe von Mindsphere Application Credentials kann ein Zugriff auf die Mindsphere von einer Applikation außerhalb der Mindsphere erfolgen. Ohne diesen Zugriff ist es nicht möglich vom IoT-Connector aus auf die im OSCAR-Analysis hinterlegten Sollwerte zuzugreifen. Nach der Generierung der Credentials müssen diese sicher verwahrt werden, da ein nachträgliches Einsehen der Credentials aus Sicherheitsgründen nicht möglich ist. Es ist jedoch möglich diese Application Credentials zurückzusetzen und neu zu generieren.

Es gibt verschiedene Zugriffsberechtigungen über die Applikation Credentials. Für OSCAR-Analysis wurde die Read/Write Variante gewählt, da es neben dem Auslesen der Sollwerte im IoT-Connector auch möglich sein muss, gesetzte Sollwerte in der OSCAR-Analysis Applikation zu markieren, damit diese nicht erneut gesetzt werden.

Zusätzlich wird durch den IoT-Connector ein Zeitstempel in die Mindsphere übertragen, an dem der Sollwert in die Steuerung geschrieben wurde. Gleichzeitig wird ein Event im Asset hinterlegt, wenn ein Sollwert geschrieben wurde.

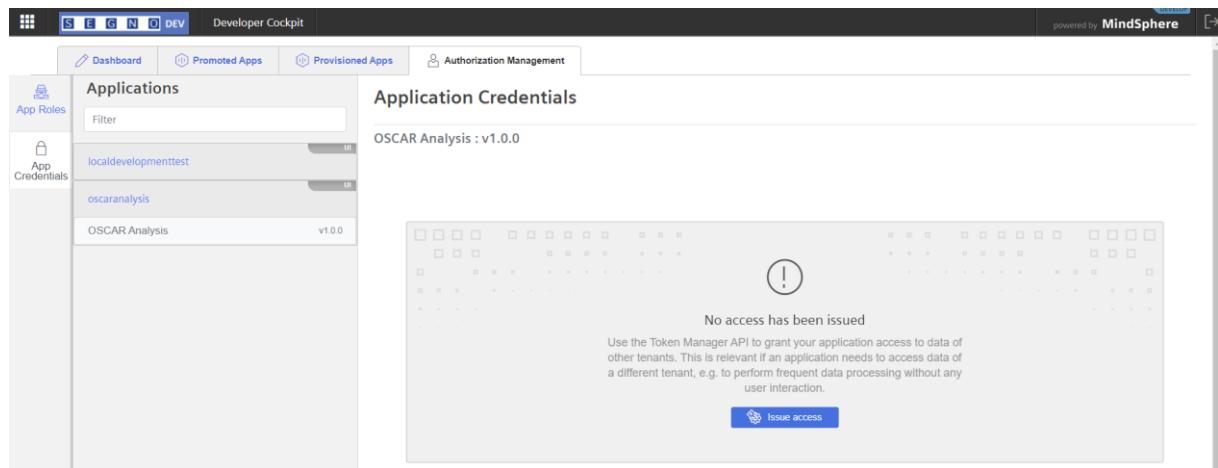


Abbildung 38: Mindsphere – Generieren der Application Credentials

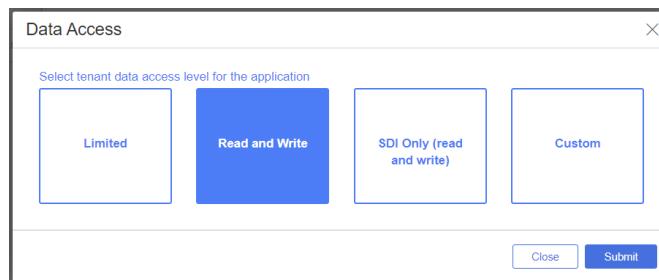


Abbildung 39: Mindsphere - Auswahl der Zugriffsrechte einer Applikation

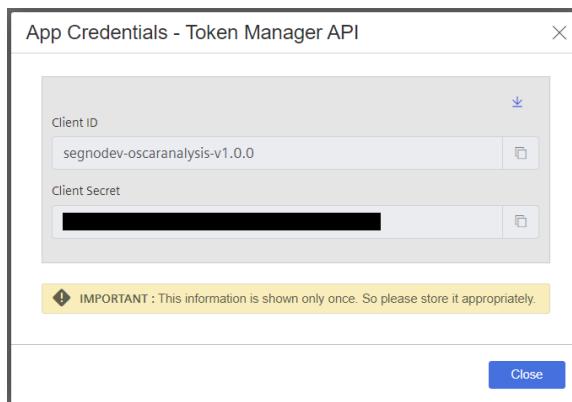


Abbildung 40: Mindsphere - Application Credentials

8.9 Bereitstellen einer Applikation via Cloud-Foundry

Für das Bereitstellen von eigenen Applikationen stellt Siemens mit der Mindsphere einen CloudFoundry-Container zur Verfügung, in den die Software hochgeladen und verwaltet werden kann. Hierzu wird eine vordefinierte „Organisation“ zur Verfügung gestellt, die an den Mindsphere-Tenant gebunden ist. Dabei ist zwischen dem Development-Tenant und dem Operator-Tenant zu unterscheiden. Der Development-Tenant bietet, wie der Name schon sagt, die Möglichkeit Applikation zur Laufzeit im Entwicklungs-Container zu hinterlegen. Nach Installation der

Bei CloudFoundry handelt es sich um eine Platform („PaaS“ – „Platform as a Service“), welche das Ausrollen und die Verwaltung einer Applikation in einer Cloud vereinfacht. Nachdem eine Software in

einem Container hinterlegt wurde, kann diese bei einem Update durch einen einzigen Befehl auf den neuesten Stand gebracht werden, ohne dass die Applikation innerhalb der Mindsphere erneut konfiguriert werden muss.

Für den Zugriff auf den CloudFoundry Container stellt die Mindsphere dem Tenant-Administrator einen Zugang bereit. Die Zugriffsberechtigungen können vom Administrator ebenfalls an andere Tenant-Benutzer übertragen werden. Dieses Kapitel beinhaltet die Installation, Einrichtung und das Übertragen einer Applikation in einen CloudFoundry Container.

8.9.1 Installation

CloudFoundry kann direkt von der offiziellen GitHub-Seite von CloudFoundry heruntergeladen und installiert werden. Nach der Installation ist eine Nutzung über das Kommandozeilen-Interface möglich. Im Folgenden auch CLI Genannt (CLI – Command Line Interface). Neben der Eingabeaufforderung können auch andere Kommandozeilen verwendet werden, unter anderem das Terminal in Visual Studio Code. Dies ermöglicht es, Programme direkt nach dem Erstellen innerhalb von Visual Studio Code über das integrierte Terminal in den CloudFoundry Container zu übertragen. CloudFoundry-Befehle beginnen dabei immer mit „cf“.

Download-Link via GitHub:

- <https://github.com/cloudfoundry/cli#downloads>

Eine Dokumentation ist unter folgendem Link zu finden:

- <http://cli.cloudfoundry.org/en-US/cf/>

Eine deutschsprachige Dokumentation ist ebenfalls verfügbar:

- <http://cli.cloudfoundry.org/de-DE/cf/>

8.9.2 Login

Eine Anmeldung an CloudFoundry erfolgt über die Zugangsdaten aus dem Mindsphere-Konto und findet als Single-Sign-On (SSO) statt. Anmeldungen per Single-Sign-On sind persistent. Dies bedeutet das der Login und die anschließende Einrichtung für CloudFoundry einmalig auf dem System stattfindet, und danach ohne erneute Anmeldung verwendbar ist.

Die Anmeldung kann über folgenden CLI-Befehl ausgeführt werden:

- cf login -a <https://api.cf.eu1.mindsphere.io> --sso

```
PS C:\Projects\segno-omc-app> cf login -a https://api.cf.eu1.mindsphere.io --sso
API-Endpunkt: https://api.cf.eu1.mindsphere.io

Temporary Authentication Code ( Get one at https://login.cf.eu1.mindsphere.io/passcode ):
```

Abbildung 41: CloudFoundry Login per single-sign-on

Nach dem Absenden des Befehls wird ein „Temporary Authentication Code“ (TAC) benötigt.

Einen TAC für die Anmeldung am Mindsphere Cloud-Foundry Container kann man unter folgendem Link anfordern:

- <https://login.cf.eu1.mindsphere.io/passcode>

Nachdem man sich mit seinem Mindsphere-Benutzerdaten angemeldet hat, wird ein „login token“ an die hinterlegte E-Mail-Adresse gesendet.

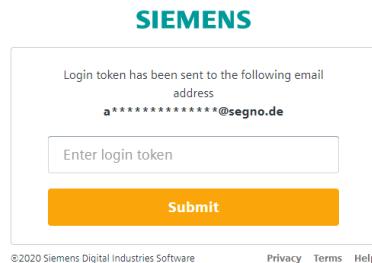


Abbildung 42: Mindsphere Login Token Abfrage

Mithilfe des „login token“ kann man nun das Formular abgeschickt werden und erhält den TAC.

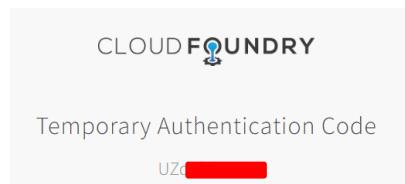


Abbildung 43: CloudFoundry - Temporary Authentication Code

Diesen TAC gibt man in der Kommandozeile an. Nach der Authentifizierung kann der CloudFoundry Container konfiguriert werden.

```
C:\Users\segno\Desktop\Projects\oscar-analysis>cf login -a https://api.cf.eu1.mindsphere.io --sso
API-Endpunkt: https://api.cf.eu1.mindsphere.io

Temporary Authentication Code ( Get one at https://login.cf.eu1.mindsphere.io/passcode ):
Authentifizieren...
OK

Select an org:
1. segnodev
2. segnoop

Organisation (enter to skip):
```

Abbildung 44: CloudFoundry - Organisationauswahl nach Login

Sind die Berechtigungen innerhalb der Mindsphere vorhanden, ist ein Zugriff auf die zugewiesenen Organisationen möglich. Für die Entwicklung wird die Development-Organisation gewählt (segnodev). Anschliessend muss noch ein „Space“ generiert werden. Spaces innerhalb einer Organisation können mit entsprechenden Berechtigungen beliebig generiert werden. Jede Applikation muss dazu in einem eigenen Space hinterlegt werden.

Die Generierung eines neuen Space für eine Applikation erfolgt über folgenden Befehl:

- cf create-space {Applikationsname}

```
C:\Users\segno\Desktop\Projects\oscar-analysis>cf create-space oscar-analysis
Creating space oscar-analysis in org segnodev as andreas.luedtke@segno.de...
OK

Assigning role SpaceManager to user andreas.luedtke@segno.de in org segnodev / space oscar-analysis as andreas.luedtke@segno.de...
OK

Assigning role SpaceDeveloper to user andreas.luedtke@segno.de in org segnodev / space oscar-analysis as andreas.luedtke@segno.de...
OK

TIP: Use 'cf target -o "segnodev" -s "oscar-analysis"' to target new space
C:\Users\segno\Desktop\Projects\oscar-analysis>
```

Abbildung 45: CloudFoundry - Anlegen eines Space

Wurde ein neuer Space angelegt, muss dieser noch als Standard hinterlegt werden, damit folgende Befehle auf diesen Space angewendet. Der Befehl dazu wird nach der Erstellung als Hinweis angezeigt. Somit erfolgen alle weiteren Befehle innerhalb des neuen Space.

- cf target -o {Organisationsname} -s {Applikationsname}

```
C:\Users\segno\Desktop\Projects\oscar-analysis>cf target -o "segnodev" -s "oscar-analysis"
API-Endpunkt: https://api.cf.eu1.mindsphere.io
API version: 3.99.0
Benutzer: andreas.luedtke@segno.de
Organisation: segnodev
Bereich: oscar-analysis
```

Abbildung 46: CloudFoundry - Festlegen eines Target

8.9.3 Benutzer / Benutzerrechte

Innerhalb von CloudFoundry ist es möglich, Benutzer mit verschiedener Rechten zu versehen.

Eine Liste aller Benutzer einer Organisation ist über folgenden Befehl einsehbar:

- „cf org users“

```
PS C:\Projects\segno-omc-app> cf org-users segnodev
Abrufen von Benutzern in Organisation segnodev als thorstenarendt@segno.de...

ORGANISATIONSMANAGER
thorsten.arendt@segno.de
andreas.luedtke@segno.de

ABRECHNUNGSMANAGER
thorsten.arendt@segno.de

ORGANISATIONSAUDITOR
Rolle ORGANISATIONSAUDITOR nicht gefunden
```

Abbildung 47: CloudFoundry - Anzeigen der Nutzer einer Organisation

8.9.3.1 Benutzerrechten innerhalb einer „Organisation“

Innerhalb einer Organisation können Benutzerrechte einzeln vergeben werden. Es folgt ein Auszug aus der offiziellen Dokumentation.

USAGE

```
cf set-org-role USERNAME ORG ROLE [--client]
```

ROLES

OrgManager - Invite and manage users, select and change plans, and set spending limits

BillingManager - Create and manage the billing account and payment info

OrgAuditor - Read-only access to org info and reports

Abbildung 48: CloudFoundry - Vergabe von Benutzerrechten innerhalb einer Organisation

```
PS C:\Projects\segno-omc-app> cf set-org-role andreas.luedtke@segno.de segnodev OrgManager
Zuweisen der Rolle OrgManager zu Benutzer andreas.luedtke@segno.de in der Organisation segnodev als thorstenarendt@segno.de...
OK
```

Abbildung 49: CloudFoundry - Setzen von Benutzerrechten innerhalb einer Organisation

8.9.3.2 Benutzerrechte innerhalb eines „Space“

Innerhalb eines Space können Benutzerrechte einzeln vergeben werden. Es folgt ein Auszug aus der offiziellen Dokumentation.

USAGE

```
cf set-space-role USERNAME ORG SPACE ROLE
```

ROLES

SpaceManager - Invite and manage users, and enable features for a given space

SpaceDeveloper - Create and manage apps and services, and see logs and reports

SpaceAuditor - View logs, reports, and settings on this space

Abbildung 50: CloudFoundry - Dokumentation von Benutzerrechten innerhalb eines Space

```
PS C:\Projects\segno-omc-app> cf set-space-role andreas.luedtke@segno.de segnodev oscaranalytics SpaceDeveloper
Zuweisen der Rolle SpaceDeveloper zu Benutzer andreas.luedtke@segno.de in der Organisation segnodev / im Bereich oscaranalytics als thorstenarendt@segno.de...
OK
```

Abbildung 51: CloudFoundry - Setzen von Benutzerrechten innerhalb eines Space

8.9.4 Deployen einer Applikation in die Mindsphere

Nachdem die Konfiguration abgeschlossen ist, können Applikationen in den Container übertragen werden. Dies ist bereits während der Entwicklung möglich, und Applikationen können fortlaufend aktualisiert werden, ohne das CloudFoundry neu konfiguriert werden muss.

8.9.5 Upload der Applikation in die Siemens Mindsphere

Damit die Applikation innerhalb der Mindsphere gefunden wird, benötigen wird eine Manifest-Datei benötigt. Dazu erstellt man die Datei „manifest.yml“.

Die Einrückung der Zeilen in der manifest.yml muss eingehalten werden und der aus dem unten angegebenem Screenshot entsprechen. Folgende Parameter müssen angepasst werden:

- Name: Applikationsname

- Path: ./dist/Applikationsname (Angular Build-Ordner)
- Memory: Die zu reservierende Größe im Cloudspeicher. Muss bei größeren Applikationen angepasst werden.

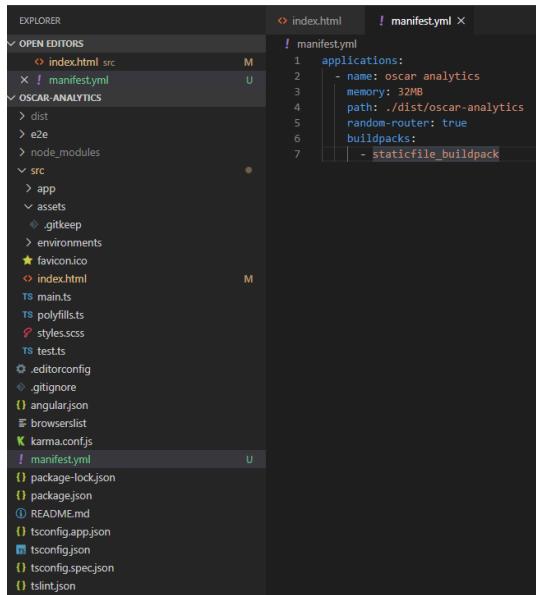


Abbildung 52: Manifestdatei innerhalb eines Angular Projekts

Es müssen Anpassungen an folgendem Parametern getätigt werden:

- Name: Applikationsname
- Path: ./dist/Applikationsname
- Memory: Zu reservierten Größe im Cloudspeicher

8.9.6 Builden der Applikation

Die Applikation kann mit folgendem Befehl gebaut werden:

- ng build—prod (wird ebenfalls durch den nächsten Parameter ausgeführt)
- npm run build—prod (script, welches durch das Angular Projekt angelegt wird und den Befehl ng build—prod aufruft)

Anschließend befindet sich die kompilierte Website im Ordner „dist/Applikationsname“.

8.9.7 Upload in die Mindsphere

Über folgenden Befehl kann die Applikation in die Mindsphere geladen werden:

- cf push APPNAME

Beim ersten „Push“ einer Applikation werden alle benötigten Pakete innerhalb der Cloud geladen und die Umgebung vorbereitet. Der Vorgang dauert einen Moment. Jedes Update einer Applikation benötigt daher weniger Zeit, da nur die Unterschiede gepusht werden.

```
C:\Users\segno\Desktop\Projects\oscar-analysis>cf push
Durchführen einer Push-Operation für App oscar-analysis zur Organisation segnodev / Bereich oscar-analysis als [REDACTED]@segno.de...
Applying manifest file C:\Users\segno\Desktop\Projects\oscar-analysis\manifest.yml...
Manifest applied
Packaging files to upload...
Hochladen der Dateien...
12.43 KiB / 12.43 KiB [=====]

Warten auf den Abschluss der Dateiverarbeitung durch die API...

Staging von App und Tracing von Protokollen...
  Downloading staticfile_buildpack...
  Downloaded staticfile_buildpack
  Cell 4e9fd2aa-28d5-4f26-a903-90cfa7378198 creating container for instance 98cbbe43-cfbc-4b0e-a084-33645a3afca5
  Cell 4e9fd2aa-28d5-4f26-a903-90cfa7378198 successfully created container for instance 98cbbe43-cfbc-4b0e-a084-33645a3afca5
  Downloading app package...
  Downloaded app package (2.1M)
    ----> Staticfile Buildpack version 1.5.19
    ----> Installing nginx
    Using nginx version 1.19.10
    ----> Installing nginx 1.19.10
    Download [https://buildpacks.cloudfoundry.org/dependencies/nginx-static/nginx-static_1.19.10_linux_x64_cflinuxfs3_109f725d.tgz]
    **WARNING** nginx 1.19.x will no longer be available in new buildpacks released after 2021-05-01.
    See: https://nginx.org/
    ----> Root folder /tmp/app
    ----> Copying project files into public
    ----> Configuring nginx
    Exit status 0
  Uploading droplet, build artifacts cache...
  Uploading build artifacts cache...
  Uploading droplet...
  Uploaded build artifacts cache (2.4M)

Waiting for app oscar-analysis to start...

Instances starting...
Instances starting...
Instances starting...

Name:          oscar-analysis
angeforderter Zustand: started
Routen:        oscar-analysis-exhausted-gorilla-xs.apps.eu1.mindsphere.io
Letztes Hochladen: Mon 30 Aug 13:50:55 CEST 2021
Stack:          cflinuxfs3
Buildpacks:
  Name      Version   detect output  buildpack name
  staticfile_buildpack  1.5.19  staticfile  staticfile

Typ:           web
sidecars:
Instanzen:     1/1
Speicherbelegung: 192M
Startbefehl:   $HOME/boot.sh
  Zustand seit      CPU   Speicher      Platte       Details
#0 aktiv  2021-08-30T11:51:07Z  0.0%  12.4M von 192M  17.4M von 256M

C:\Users\segno\Desktop\Projects\oscar-analysis>
```

Abbildung 53: Upload einer Applikation in CloudFoundry

Nachdem die Bearbeitung abgeschlossen ist, sollte der Zustand „aktiv“ sein. Die Route wird später benötigt, um den Pfad zur Applikation innerhalb der Mindsphere anzugeben.

8.10 Konfiguration der Applikation innerhalb der Mindsphere

Die Applikation wurde bereits in den CloudFoundry Container hochgeladen. Als nächstes muss die Anwendung innerhalb der Mindsphere angelegt werden. Dies geschieht über das „Developer Cockpit“ im Development-Tenant der Mindsphere.

Über den Button „Add application“ kann eine Neue Anwendung hinzugefügt werden.

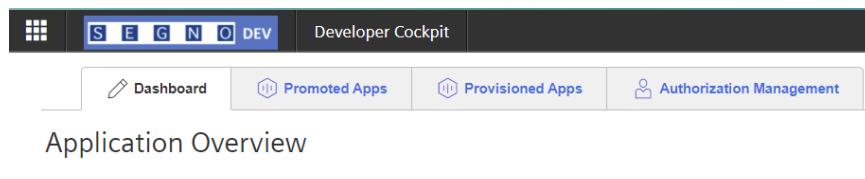


Abbildung 54: Mindsphere - Anlegen einer neuen Applikation

Es öffnet sich das Konfigurationsfenster. Hier muss zuerst der Applikationsname, sowie die Version angegeben werden. Ein Logo kann ebenfalls hinterlegt werden. Dieses ist nachher für Benutzer mit Zugriff auf die Applikation in der Hauptansicht sichtbar.

Abbildung 55: Mindsphere - Konfiguration einer Applikation

Anschliessend muss die Komponente konfiguriert werden. Dazu muss im unteren Bereich „Components“ der Name festgelegt werden und die CloudFoundry Direct URL angegeben werden. Der Name der Komponente sollte hierbei dem Namen innerhalb der Manifest-Datei entsprechen.

Bei der CloudFoundry Direct URL handelt es sich um die „Route“ die nach Abschluss des Befehles „cf push“ zurückgegeben wird.

Abbildung 56: Mindsphere - Eintragen der CloudFoundry Direct URL

Zuletzt muss noch ein „Endpoint“ über den Button „Add Endpoints“ hinzugefügt werden.

Es öffnet sich ein Dialogfenster.

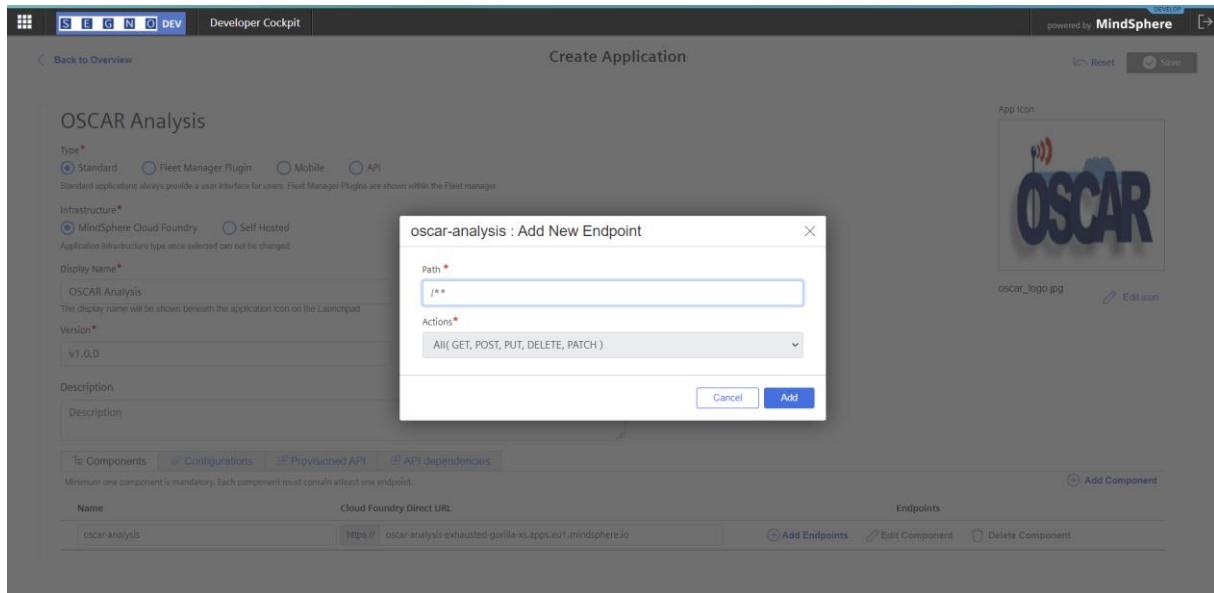


Abbildung 57: Mindsphere - Endpoint hinzufügen

Hier kann der Pfad mit `/**` eingetragen werden. Dadurch werden alle Anfragen an diese URL weitergeleitet.

Über den Button Save kann die Anwendung nun in der Mindsphere hinterlegt werden. Jetzt müssen noch die Zugriffsrechte und Berechtigungen der Applikation über den Button „Roles-Configure“ gesetzt werden.

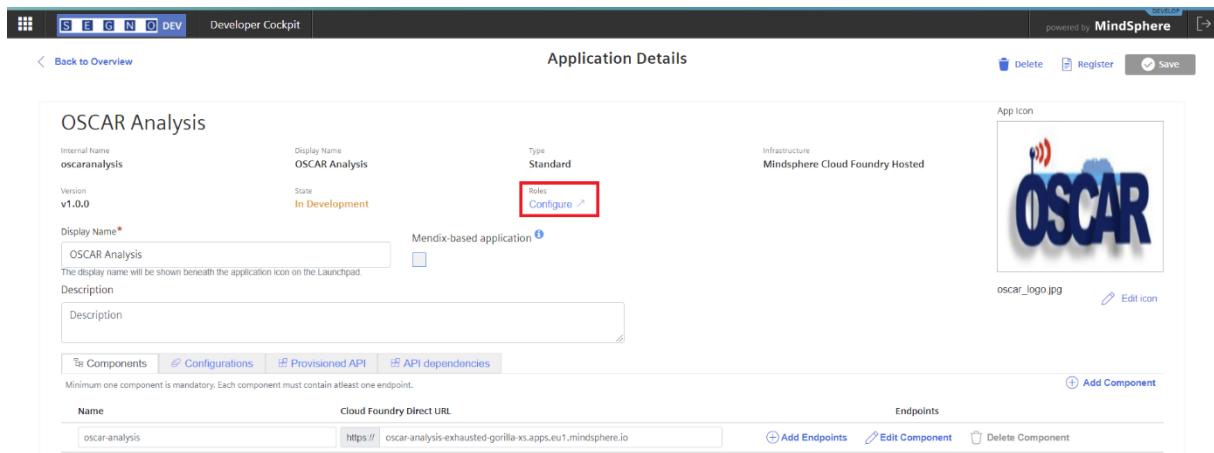


Abbildung 58: Mindsphere - Konfiguration der Rollen

Über den Button „Register“ kann die Anwendung nun registriert und somit zugänglich gemacht werden. Die Applikation kann anschließend über das Mindsphere Launchpad geöffnet werden.

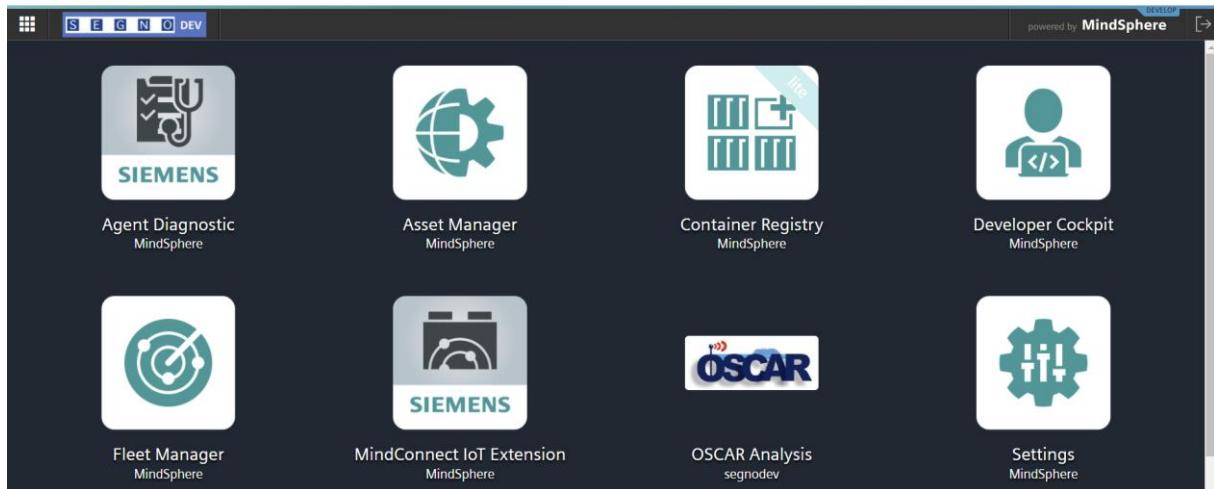


Abbildung 59: Mindsphere - OSCAR Analysis auf der Übersichtsseite

8.10.1 Zuweisung der Applikation für andere Nutzer

Die Zuweisung der Applikationen zu verschiedenen Mindsphere Benutzern erfolgt über „Settings“ im Launchpad. Hierzu wählt man den entsprechenden Benutzer aus und bearbeitet über „Direkte Zuweisungen bearbeiten“ die Rechte des jeweiligen Nutzers.

The screenshot shows the 'User Settings' page in the Mindsphere interface. On the left, a sidebar lists users under 'Benutzer'. On the right, a main panel shows the 'Benutzertyp' (User Type) for a selected user, which is 'Globaler Benutzer' (Global User). It also displays a message stating that the user is currently not assigned to any user groups and provides a button to 'Create User Group'. At the bottom, a section for 'Rollen' (Roles) is shown, indicating the user has the following roles assigned: 'Der Benutzer bekommt Rechte der folgenden Rollen:' (The user gets rights of the following roles:).

Abbildung 60: Vergabe von Rollen einer Applikation

Es öffnet sich ein Popup, in dem die benötigten Rollen vergeben werden können. Nach Auswahl der jeweiligen Rolle muss diese noch bestätigt werden.

Zuweisungen bearbeiten für Baxmann, Ralf

Filter

Alle deselektrieren

Schritt 1 von 3

Filter ▼

<input type="checkbox"/>	localdevelopmenttest (mdsp:segnodev)
<input type="checkbox"/>	admin Administrator role for localdevelopmenttest application
<input type="checkbox"/>	localdevelopmenttest (mdsp:segnodev)
<input type="checkbox"/>	user Standard User role for localdevelopmenttest application
<input checked="" type="checkbox"/>	oscaranalysis (mdsp:segnodev)
<input checked="" type="checkbox"/>	admin * Administrator role for oscaranalysis application
<input type="checkbox"/>	oscaranalysis (mdsp:segnodev)
<input type="checkbox"/>	user Standardbenutzer

* Sie haben eine ungespeicherte Änderung.

Zurücksetzen Abbrechen Weiter

Abbildung 61: Mindsphere - Rollenzuweisung

Bevor die Rollen aktiv sind und die Applikation im Launchpad verfügbar ist, muss der Benutzer sich neu an der Mindsphere anmelden.

8.11 Richtlinien für Mindsphere-Applikationen

Applikationen, die innerhalb der Mindsphere zur Veröffentlichung freigegeben werden, unterliegen einigen Richtlinien. Bevor eine Applikation daher für andere Nutzer zugänglich gemacht werden kann, wird diese einer automatisierten Prüfung seitens Siemens unterzogen.

Siemens hat dafür definierte Richtlinien, die einzuhalten sind.

- Applikationen dürfen nicht als Vertriebsmechanismus agieren. Es ist somit nicht zulässig Mechanismen bereitzustellen, die es ermöglichen Software bereitzustellen
- Es nicht erlaubt, veraltete Software-Komponenten zu verwenden, da hier die Gefahr von Sicherheitslücken ausgehen bestehen kann
- Die Nutzung von Software-Komponenten mit bekannten Sicherheitslücken ist ausdrücklich verboten
- Applikationen dürfen nach einem Neustart keiner Bedienung erfordern, um korrekt zu laufen
- Applikationen müssen unter der Sub-Domain registriert werden, die dem Mindsphere Account zugrundliegt.
- Bei der Veröffentlichung von Applikationen durch CloudFoundry, darf nur eine Applikation pro CloudFoundry-Space existieren.
- Wenn mit Benutzerdaten interagiert wird, muss sichergestellt sein, dass die Verwaltung der Daten den gesetzlichen Anforderungen entsprechen. Weiterhin wird Transparenz gefordert, welche Art von Benutzerdaten erhoben werden, und wie sie geschützt werden. Dabei ist es ebenfalls erforderlich, dass vor einer Nutzung durch den Endanwender dem Erheben solcher Daten zugestimmt wird.

Quelle: <https://documentation.mindsphere.io/resources/html/devops-guide/en-US/101636514571.html> (Stand 2022-05-30)

8.12 Veröffentlichung der Applikation im Operator-Tenant

Nachdem die Applikation entwickelt und getestet wurde, kann diese an den Operator weitergegeben werden, bevor sie schließlich mit anderen Tenants geteilt werden kann. Applikationen, die im Operator-Tenant registriert sind, können ebenfalls im Mindsphere Store bereitgestellt werden.

Da es sich bei diesem Projekt aber um eine Individuallösung in Verbindung mit dem IOT-Connector handelt, ist ein Vertrieb über den Store nicht geplant.

Die Weitergabe der Applikation an den Operator geschieht über die „Application-Details“ im Operator-Cockpit. Über den Button „Manage Uploads“ öffnet sich folgender Popup, in dem wichtige Informationen hinterlegt werden müssen.

The screenshot shows the 'Upload' dialog box. At the top, there is an informational message: 'MindSphere Cloud Foundry requires multiple applications to be described with a single Manifest file. This manifest should be uploaded separately. Manifest uploaded with binaries will be ignored. For additional information, Click here..'. Below this, there are two main sections: 'Binary*' and 'Manifest*'. Under 'Binary*', the file 'oscar-analysis.zip' is selected, and there are 'Browse' and 'Upload' buttons. Under 'Manifest*', the file 'manifest.yml' is selected, and there are 'Browse' and 'Upload' buttons. Next is the 'Service Plan' section, which includes 'Select Service' and 'Select Service Plan' dropdowns, and a field to 'Enter Service name' with a 'Add service' button. Below this is the 'Service List' table:

Name	Instance	Plan

Then comes the 'Description*' field containing 'OSCAR Analysis'. Under 'App Credentials', it says 'Please select the data access level that your application requires:' with a dropdown set to 'Read and Write'. A note below states: 'Your currently configured data access level has been preselected. No credentials are handed over. Only the required data access level is provided to the production system.' At the bottom are 'Close', 'Discard', and 'Submit' buttons.

Abbildung 62: Mindsphere - Veröffentlichen einer Applikation im Operator Tenant

Unter dem Punkt „Binary“ muss die kompilierte Applikation in gezippter Form abgelegt werden. Diese befindet sich zum Beispiel bei Visual Studio Code mit Standardkonfiguration im Ordner „dist/Applikationsname“. Nach Upload der Binary muss ebenfalls das Manifest hochgeladen werden. Es handelt sich hierbei um die Datei „manifest.yml“, die ebenfalls für das Hochladen der Applikation in den CloudFoundry Container benötigt wird.

Zusätzlich müssen noch die „App Credentials“ gesetzt werden. Im Fall der Applikation OSCAR Analysis sind diese auf „Read and Write“ zu setzen damit später der Zugriff auf die Applikations- und Asset-Daten auch ausserhalb der Mindsphere möglich sind. Nachdem die Beschreibung der Applikation gesetzt wurde, kann diese über den „Submit“-Button eingereicht werden.

The screenshot shows the 'Application Details' page for the 'OSCAR Analysis' application. Key details include:

- Internal Name:** oscaranalysis
- Display Name:** OSCAR Analysis
- Type:** Standard
- State:** Ready for Assignment
- Infrastructure:** Mindsphere Cloud Foundry Hosted
- Version:** v1.0.0
- Application URL:** https://segnodev.oscaranalysis-segnodev.eu1.mindsphere.io
- Roles:** Configure ↗

The 'Operator Assignment' tab is selected, showing the assignment to 'segnoop' with the email 'segnoop@segno.de'. A note indicates that only Developer Administrators can assign applications to Operators.

Abbildung 63: Mindsphere - eingetragene Applikation

Nachdem die Applikation in der Mindsphere vorhanden ist , so können die Zugangsdaten in der OSCAR Analysis Applikation für die Auswahl hinterlegt werden.

Die „Application Credentials“ einer CloudFoundry-Applikation können über folgenden CloudFoundry Befehl abgefragt werden :

- „ cf env {Applikationsname}“

```
C:\Users\segno\Desktop\Projects\oscar-analysis>cf env oscaranalysis
Abrufen von Umgebungsvariablen für App oscaranalysis in Organisation segnoop / Bereich oscar-analysis als [REDACTED]@segno.de...
Vom System zur Verfügung gestellt:
VCAP_SERVICES: {}

VCAP_APPLICATION: {
  "application_id": "490f1213-7c57-43c7-b1ee-18cc35ed85b4",
  "application_name": "oscaranalysis",
  "application_uris": [
    "oscaranalysis-thankful-panther-em.apps.eu1.mindsphere.io"
  ],
  "cf_api": "https://api.cf.eu1.mindsphere.io",
  "limits": {
    "fds": 16384
  },
  "name": "oscaranalysis",
  "organization_id": "1961a23b-8d7b-40f4-bbc0-84ee89aba09c",
  "organization_name": "segnoop",
  "space_id": "83c5e9f6-0d63-4b67-a456-192e2bc66f2b",
  "space_name": "oscar-analysis",
  "uris": [
    "oscaranalysis-thankful-panther-em.apps.eu1.mindsphere.io"
  ],
  "users": null
}

Vom Benutzer bereitgestellt
MDSP_HOST_TENANT: segnoop
MDSP_KEY_STORE_CLIENT_ID: segnoop-oscaranalvsis-v1.0.0-89986624
MDSP_KEY_STORE_CLIENT_SECRET: [REDACTED]
MDSP_OS_VM_APP_ID: 79757b71-4b41-406e-8bd5-29ab7ba7d4df
MDSP_OS_VM_APP_NAME: oscaranalysis
MDSP_OS_VM_APP_VERSION: v1.0.0

Es wurden keine aktiven Umgebungsvariablen festgelegt
Keine Stagingumgebungsvariablen festgelegt
```

Abbildung 64: CloudFoundry - Anzeige der Information einer Applikation

8.13 Untersuchung geplanter Cloud-Funktionalitäten

8.13.1 Implementierung eines dynamischen Prozessbildes

Innerhalb der OSCAR-Analysis Applikation sollte ein Prozessbild je Schiff hinterlegt werden, in dem die Zuordnung der Werte aus den verschiedenen Mindsphere-Assets dynamisch eingefügt werden kann. Innerhalb eines Prozessleitsystems ist diese Funktionalität über die Entwicklung mithilfe von Objekten innerhalb von Visualisierungssystemen möglich.

Da es sich bei OSCAR-Analysis aber um eine Web-Applikation handelt, die solche Funktionalitäten von Haus aus nicht unterstützt, hätte ein dynamisches Prozessbild explizit für die Verwendung innerhalb von HTML-Elementen entwickelt werden müssen. Eine Verwendung von verschiedenen Internet-Browsern, Monitor-Auflösungen sowie Zoom-Faktoren innerhalb des Browsers hätten diesen Vorgang nicht zu einem einheitlichen Ergebnis kommen lassen.

Es wurde daher auf ein dynamisches Prozessbild verzichtet, da die benötigte Zeit für die Entwicklung eines solchen Features außerdem den Projektzeitraum zeitlich wesentlich überschritten hätte.

8.13.2 Melde- und Benachrichtigungssystem per E-Mail

Während der Planung des Funktionsumfangs war ein Melde- und Benachrichtigungssystem zur Unterstützung des Schiffspersonals angedacht, welches Benachrichtigungen, Handlungsempfehlungen oder Optimierungsvorschläge per E-Mail bei bestimmten Ereignissen auslösen soll. Dabei kann es sich z.B.: um die Über-/ sowie Unterschreitung von Prozesswerten handeln, wo ggf. eine Maßnahme an Bord erforderlich wird.

Bei der Entwicklung und Umsetzung konnte diese Funktionalität bereits als "Built-In-Feature" innerhalb der Mindsphere-Applikation „Fleet-Manager“ vorgefunden werden. Eine separate Entwicklung und Konzeptionierung waren dahingehend somit nicht mehr erforderlich.

9 Untersuchung OSCAR-Connect für andere Cloud-Systeme

OSCAR-Connect wurde im Rahmen des Projektes für den Betrieb mit der Siemens Cloud - Mindsphere entwickelt. Zusätzlich wurde geprüft und untersucht, ob eine Umsetzung auf andere Cloud-Systeme möglich ist, und welche Anpassungen an OSCAR-Connect notwendig wären, um eine flexible Nutzung mit verschiedenen Cloud-Anbieter zu ermöglichen.

Da es sich bei OSCAR-Connect um eine Kombination aus dem IOT-Connector, sowie der Mindsphere-Applikation OSCAR-Analysis handelt, werden diese gesondert betrachtet.

9.1 Untersuchung Übertragbarkeit IOT-Connector

Der IOT-Connector ist für die Kommunikation erstens mit dem Steuerungsnetz und zweitens mit der angebundenen Cloud-Umgebung zuständig.

Die Kommunikation mit dem Steuerungsnetz ist unabhängig von der angebundenen Cloud. Hierbei handelt es sich um das Auslesen von Werten aus den konfigurierten Steuerungen, sowie dem Zurückschreiben von Verfahrensanweisungen in Form von Sollwerten. Es ist daher nur das Protokoll entscheidend, mit dem die Steuerungen angesprochen werden. Somit würde dieser Teil auch ohne Anpassung an verschiedene Cloud-Anbieter funktionsfähig bleiben. Eine Übertragbarkeit wäre hier sicherlich gegeben.

Anders sieht es bei der Kommunikation mit der Cloud aus. Der Datenaustausch zwischen dem IOT-Connector und der Cloud umfasst folgenden Punkte:

- Übermitteln der Konfiguration in die Cloud
- Übermitteln von TimeSeries-Daten in die Cloud
- Generieren von Ereignissen innerhalb der TimeSeries-Datenbank
- Auslesen von Verfahrensanweisungen (Sollwerten) aus der Cloud

Für alle aufgelisteten Operationen ist eine Anmeldung an der Cloud erforderlich. Im Falle der Mindsphere sind dafür Credentials notwendig, die innerhalb der Mindsphere erstellt werden können, und im IOT-Connector hinterlegt werden müssen. Mittels dieser Credentials ist ein „onboarding“, also eine Anmeldung an der Mindsphere möglich. Diese Informationen werden im IOT-Connector unter dem Punkt „Mindsphere-Konnektor“ hinterlegt.

Für die Nutzung mit einer weiteren Cloud muss dafür ein neuer Konnektor implementiert werden, der die benötigten Informationen zur Anmeldung an der jeweiligen Cloud beinhaltet. Eine Übertragbarkeit ohne Anpassungen ist so nicht möglich.

Neben dem Onboarding über die Credentials des Mindsphere Agenten besteht noch eine weitere Art der Anmeldung an der Mindsphere. Während der Mindsphere Agent „mindconnectlib“ nur für die Kommunikation vom IOT-Connector zur Mindsphere agiert, werden noch die „Application Credentials“ von der OSCAR-Analysis Applikation benötigt, um die dort hinterlegten Sollwerte auszulesen.

Für diese Art der Anmeldung sind ebenfalls Zugangsdaten im IOT-Connector zu hinterlegen. Diese unterliegen dem Punkt „Sollwert“. Bei jeder Übernahme von Sollwerten in die Steuerung wird ebenfalls ein Event innerhalb der Mindsphere-TimeSeries-Datenbank generiert. Je nach Umsetzung und Verfügbarkeit von Events in anderen Cloud-Systemen muss auch hier die Implementierung auf andere Cloud-Systeme angepasst werden.

Eine genaue Implementation der Kommunikation in oder aus verschiedenen Cloud-Anbietern konnte im Rahmen des Projekts jedoch nicht genau untersucht werden. Je nach Cloud-Anbieter muss aber sichergestellt werden, dass eine Anmeldung für einen Upload der Prozesswerte in die entsprechende TimeSeries-Datenbank erfolgen kann, sowie Verfahrensanweisung (Sollwerte) auch aus der Cloud ausgelesen werden können.

Der IOT-Connector muss entsprechend die Konfigurationsmöglichkeiten auf der internen Webseite des ermöglichen. Zusätzlich wird eine Anpassung der darunterliegenden Datenbank für die Anmelde-Informationen erfolgen müssen, da die Parameter für die Anmeldung sich unterscheiden können.

Ausblick:

Im Rahmen eines möglichen Anschlussvorhabens könnte eine genauere Untersuchung der oben beschrieben Punkte erfolgen und erprobt werden.

9.2 Untersuchung Übertragbarkeit OSCAR-Analysis

Bei der OSCAR-Analysis Applikation innerhalb der Mindsphere handelt es sich um eine Webseite auf Basis von Angular und NodeJS, die in einem CloudFoundry-Container liegt. Obwohl eine Verwendung von Angular-Webseiten in CloudFoundry-Containern auch in anderen Cloud-Systemen möglich ist, ist eine Nutzung von OSCAR-Analysis in anderen Cloud-Systemen dennoch nicht möglich.

OSCAR-Analysis wurde speziell auf die Mindsphere zugeschnitten. Innerhalb der Applikation befinden sich zudem einige Mindsphere Web-Elemente, die nur innerhalb der Mindsphere Verwendung finden (siehe Kapitel 8.6.4 - Implementierung von Mindsphere Web Komponenten). Eine Übertragbarkeit von WEB Elementen aus dem Mindsphere System ist für andere Cloud Umgebungen nicht gegeben.

Hierunter fallen:

- Mindsphere Navigationsleiste
- Mindsphere Web-Komponenten
 - o Asset-Selector (entspricht dem angebundenen IOT-Connector)
 - o Aspect-Selector (entspricht den Steuerungen innerhalb eines IOT-Connector)
 - o Event-Viewer
 - o TimeSeries Chart
 - o Date Time Range Picker

Weiterhin können innerhalb der OSCAR-Analysis Applikation Sollwerte hinterlegt werden. Die hinterlegten Sollwerte werden innerhalb der Mindsphere in den Assets hinterlegt. Während das Hochladen von TimeSeries-Daten zu den gängigen Operationen einer Cloud gehört, ist das Herunterladen von Daten in anderen Plattformen anders umzusetzen, oder eventuell gar nicht

möglich. Eine Portierung der OSCAR-Analysis Applikation auf andere Cloud-Systeme kann somit nicht stattfinden und müsste für die jeweilige Cloud neu konzeptioniert und entwickelt werden.

Ausblick:

Im Rahmen eines möglichen Anschlussvorhabens könnte eine genauere Untersuchung der oben beschriebenen Punkte (WEB-Elemente und steuern aus der Cloud heraus) erfolgen und erprobt werden.

9.3 Untersuchung Übertragbarkeit des digitalen Zwillings in verschiedene Cloudumgebungen

Eine Implementierung des digitalen Zwillings in die Mindsphere wurde konzeptionell untersucht und für möglich erachtet. Eine Portierung für andere Cloudumgebungen wurde bisher nicht betrachtet.

Ob die verwendete Simulationssoftware SIMBA von ifakFAST auch in anderen Cloudumgebungen lauffähig wäre, hängt davon ab, ob diese die Programmiersprache C# (.NET Core) unterstützen.

SIMBA kann derzeit nur über eine .NET Core Implementation verwendet werden, soll aber in Zukunft auch für JAVA verfügbar sein. Ein Veröffentlichungstermin ist allerdings noch nicht bekannt. Je nach Cloudumgebung könnte dann ebenfalls eine Implementierung in JAVA in Betracht gezogen werden.

Ausblick:

Im Rahmen eines möglichen Anschlussvorhabens könnte eine genauere Untersuchung der oben beschriebenen Punkte der Übertragbarkeit des digitalen Zwillings in eine andere Cloudumgebung erfolgen und erprobt werden. Untersuchungen und Erprobungen könnten diesbezüglich mittels Erweiterung auf die NoSQL Datenbank „InfluxDB“ und die Ankopplung an die Cloudumgebung von Microsoft „MS Azure“ inkl. Insight erfolgen.

10 Literaturangaben

Simulationssystem SIMBA

<https://www.ifak.eu/de/produkte/simba>

Siemens MindSphere – Industrielles IoT-as-aService

<https://siemens.mindsphere.io/de>

<https://design.mindsphere.io/osbar/get-started.html>

<https://documentation.mindsphere.io/MindSphere/resources/mindsphere-webcomponents/index.html>

Cloud Foundry Foundation - an independent non-profit open source organization

<https://www.cloudfoundry.org/>

<https://github.com/cloudfoundry/cli#downloads>

<http://cli.cloudfoundry.org/en-US/cf/>

<http://cli.cloudfoundry.org/de-DE/cf/>

Node.js als asynchrone, Event-basierte Laufzeitumgebung

<https://nodejs.org/de>

OPC Unified Architecture (UA) - OPC Classic specifications and framework

<https://opcfoundation.org/about/opc-technologies/opc-ua/>

unipi - Raspberry Pi computer

<https://www.unipi.technology/>

PhoenixContact - Lüfterloser Industrie-Box-PC

<https://www.phoenixcontact.com/de-de/produkte/box-pc-bl2-bpc-1501e-64-w10-1158241>

Wago – Edge Computer

<https://www.wago.com/de/sps/edge-computer/p/752-9401>

Siemens – IoT 2050 smartes Gateway für industrial Edge

<https://new.siemens.com/de/de/produkte/automatisierung/pc-based/iot-gateways/simatic-iot2050.html>

Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Veröffentlichung (Publikation) Schlußbericht		
3. Titel Verbundprojekt: OSCAR - Online Modeling, Simulation and Remote Control System for environmental technologies on-board (Cruise) Ships; Vorhaben: Übertragung, Sicherung, Analyse und Verarbeitung von Umweltdaten von Kreuzfahrtschiffen in einer industriellen Cloud-Umgebung in Verbindung mit einem digitalen Zwilling			
4. Autor(en) [Name(n), Vorname(n)] Arendt, Thorsten; Niclas, Christian; Lüdtke Andreas; Baxmann, Ralf;		5. Abschlussdatum des Vorhabens 31.12.2022	
		6. Veröffentlichungsdatum	
		7. Form der Publikation Document Control Sheet	
8. Durchführende Institution(en) (Name, Adresse) SEGNO Industrie Automation GmbH Admiralstr. 54 28215 Bremen Deutschland		9. Ber. Nr. Durchführende Institution	
		10. Förderkennzeichen 03SX486B OSCAR	
		11. Seitenzahl 79 Seiten	
12. Fördernde Institution (Name, Adresse) Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn		13. Literaturangaben Siehe Kapitel 10	
		14. Tabellen 4 Tabellen	
		15. Abbildungen 64 Abbildungen	
16. Zusätzliche Angaben			
17. Vorgelegt bei (Titel, Ort, Datum) Mehrere Veröffentlichungen (vgl. Liste im Bericht)			
18. Kurzfassung Mit der Konzipierung und Entwicklung einer standardisierten Daten-Schnittstelle (OSCAR-Connect / IoT) für Umweltanlagen auf (Kreuzfahrt)-Schiffen haben wir mit diesem Vorhaben eine Grundlage für umwelttechnischen Anlagen auf (Kreuzfahrt)-Schiffen sowie für die Übertragbarkeit auf andere Betriebsanlagen (Biogasanlagen, Abgasreinigungsanlagen, unterschiedliche verfahrenstechnische Abwasserreinigungsanlagen, industrielle Produktionsanlagen, etc..) gelegt. Die technologische Kombination der Datenerfassung, der sicheren und langfristigen Datenzwischenspeicherung sowie des gesicherten und robusten Transfers der Daten über das Internet in verschiedene Cloud-Umgebungen in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe ist uns in dieser Form unbekannt am Markt. Mit der Einarbeitung und Umsetzung von speziellen umweltbezogenen Cloud-Lösungen in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe haben wir das Know-how sowie das Portfolio der SEGNO erweitert. Moderne WEB-Entwicklungen und Cloud-Apps von speziellen branchenbezogenen Umwelt-Daten (OSCAR-Analysis) in Bezug auf ein Flottenmanagement für (Kreuzfahrt)-Schiffe sind der Anfang einer ganzen Produktgruppe von Umwelt-Apps in Verbindung mit Cloud-Lösungen. Für Simulationen, verfahrenstechnische Prognosen und daraus resultierende Empfehlungen für die Optimierung von Prozessparametern oder Anweisungen für Fahrbefehle von Umweltanlagen wurde eine Kommunikationsschnittstelle (OSCAR-Connect / digitaler Zwilling) entwickelt und erprobt für einen digitalen Zwilling, der außerhalb der Cloud-Umgebung operiert. Konzeptionell wurde untersucht, ob der digitale Zwilling in einer standardisierten Form übertragbar als Softwaredienst in der Cloud-Umgebung lauffähig ist. Für die Sicherstellung des reibungslosen Datenverkehrs ist eine Verbindungsüberwachung der Kommunikationswege realisiert sowie die Messwertüberwachung der Prozesswerte. Eine Fernalarmierung und Fernparametrierung der prozessrelevanten Daten kann aus der Clouddomäne erfolgen. Damit lässt sich ein individuelles Flottenmanagement für Umweltdaten für Kreuzfahrt-/Schiffe umsetzen.			
19. Schlagwörter OSCAR, Cloud, digitaler Zwilling, Kreuzfahrt, Schiffe, Flotte, IoT Connector, Umweltanlagen, Siemens, Mindsphere			
20. Verlag		21. Preis	

Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final Report	
3. title Joint project: OSCAR - Online Modeling, Simulation and Remote Control System for environmental technologies on-board (Cruise) Ships; Project: Transmission, backup, analysis and processing of environmental data from cruise ships in an industrial cloud environment in connection with a digital twin.		
4. author(s) (family name, first name(s)) Arendt, Thorsten; Niclas, Christian; Lüdtke Andreas; Baxmann, Ralf;	5. end of project 31.12.2022	
	6. publication date	
	7. form of publication	
8. performing organization(s) (name, address) SEGNO Industrie Automation GmbH Admiralstr. 54 28215 Bremen Deutschland	9. originator's report no.	
	10. reference no. 03SX486B OSCAR	
	11. no. of pages 79 pages	
12. sponsoring agency (name, address) Bundesministerium für Wirtschaft und Klimaschutz (BMWK) 53107 Bonn	13. no. of references see chapter 10	
	14. no. of tables 4 tables	
	15. no. of figures 64 figures	
16. supplementary notes		
17. presented at (title, place, date) Several publications (see list in report)		
18. abstract With the conception and development of a standardised data interface (OSCAR-Connect / IoT) for environmental plants on (cruise) ships, we have laid a foundation for environmental plants on (cruise) ships as well as for the transferability to other operating plants (biogas plants, waste gas purification plants, different process waste water treatment plants, industrial production plants, etc..) with this project. The technological combination of data acquisition, secure and long-term intermediate data storage as well as the secure and robust transfer of data via the internet into different cloud environments in relation to a fleet management for (cruise) ships is unknown to us in this form on the market. With the incorporation and implementation of special environment-related cloud solutions in relation to a fleet management for (cruise) ships, we have expanded the know-how as well as the portfolio of SEGNO. Modern WEB developments and cloud apps of special industry-related environmental data (OSCAR-Analysis) in relation to a fleet management for (cruise) ships are the beginning of a whole product group of environmental apps in connection with cloud solutions. For simulations, procedural forecasts and resulting recommendations for the optimisation of process parameters or instructions for driving commands of environmental systems, a communication interface (OSCAR-Connect / digital twin) was developed and tested for a digital twin operating outside the cloud environment. Conceptually, it was investigated whether the digital twin can be transferred in a standardised form and run as a software service in the cloud environment. To ensure smooth data traffic, connection monitoring of the communication paths has been implemented, as well as monitoring of the measured process values. Remote alarming and remote parameterisation of the process-relevant data can be carried out from the cloud environment. This makes it possible to implement individual fleet management for environmental data for cruise ships/vessels.		
19. keywords OSCAR, Cloud, digital twin, cruise, ships, fleet, IoT Connector, environmental systems, Siemens, Mindsphere		
20. publisher	21. price	