



# **i-Twin Middleware**

**Deliverable 3.1b**  
des Forschungsprojekts i-Twin

**Dietmar Glachs**  
**Salzburg Research**

Jänner 2024

## Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>2</b>
<b>1 Zusammenfassung .....</b>	<b>3</b>
<b>2 Plattform für semantische Anwendungsintegration .....</b>	<b>4</b>
2.1 i-Twin Data Integration Layer .....	4
2.2 Asset- und Application Connectors .....	7
<b>3 Ergänzende Projektberichte .....</b>	<b>10</b>
<b>4 Literaturverzeichnis .....</b>	<b>11</b>
<b>Impressum .....</b>	<b>12</b>

# 1 Zusammenfassung

## Publizierbare Version

Der vorliegende Bericht ist ein Begleitdokument zu den im Projekt i-Twin entwickelten Software-Komponenten der Middleware Plattform für die semantische Anwendungsintegration durch Semantic Integration Patterns („i-Twin Middleware“). Der Bericht umfasst eine Momentaufnahme des Entwicklungsstands im Januar 2024 und beschreibt den Zugang zu den Software-Repositories der im System-Design definierten Hauptbausteine sowie der unterstützenden Komponenten der Middleware.

## i-Twin

i-Twin erforscht Interoperabilitätskonzepte für daten-getriebene digitale Zwillinge in der Fertigungsindustrie. Das Projekt propagiert eine Open-Source-Middleware für die Integration von Fertigungs-IT-Systemen und vernetzten Anlagen auf der Grundlage von Semantic Integration Patterns. Das vorrangige Ziel von i-Twin ist es, den Integrationsaufwand zu reduzieren und den Austausch von Stamm- und Betriebsdaten in Fertigungsnetzwerken zu ermöglichen. Die Ergebnisse werden in einem Forschungslabor und in einem industriellen Asset-Management-Szenario validiert.

Das Projektkonsortium unter der Leitung der **Salzburg Research** verbindet die Forschungsinteressen von drei Systemanbietern (**H&H Systems**: CMMS, **COPA-DATA**: OT Software Plattform, **IcoSense**: Edge-Nodes) und eines Industrieunternehmens (**INNIO Jenbacher**: diskrete Fertigung) mit der Expertise der beteiligten Forschungspartner (**Universität Salzburg**: Data Science, **Salzburg Research**: Motion Data Intelligence).

Das Projekt i-Twin wird gefördert vom BMK (Bundesministerium für Klimaschutz, Umwelt, Energie, Mobilität, Innovation und Technologie) und von der FFG (Österreichische Forschungsförderungsgesellschaft mbH) aus Mitteln des Programms IKT der Zukunft.

## 2 Plattform für semantische Anwendungsintegration

Abbildung 1 zeigt die generelle Architektur der i-Asset Plattform, welche den Anwendungs-Layer, den **i-Twin Data Integration Layer** sowie die im Edge-Layer eingebunden Assets und Devices umfasst. Diese Komponenten manifestieren gemeinsam den **Digital Twin** von realen Anlagen für die Zwecke des digitalen Asset Managements.

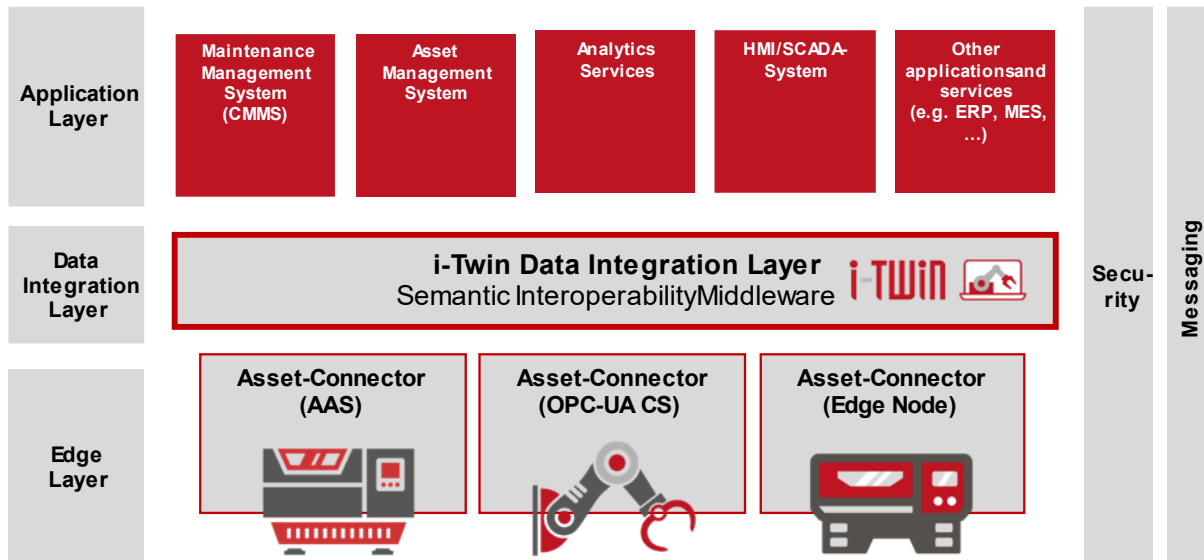


Abbildung 1: Übersicht über die Architektur der i-Asset Plattform

Das Systemdesign der Software wurde im Bericht „*Final System Design*“ (D2.4) ausführlich beschrieben. In einem weiteren Bericht „*Semantic Integration Patterns*“ (D2.2) wurden die Erfordernisse für Semantic Integration Patterns dargelegt

### 2.1 i-Twin Data Integration Layer

Hauptaugenmerk in diesem Bericht liegt auf der Beschreibung Zugangs zu den Repositories der Software-Bausteine, die im Projekt i-Twin auf Basis des konzeptionellen System-Designs entwickelt wurden. Die Summe aller genannten Bausteine repräsentiert den i-Twin Data Integration Layer, wie in Abbildung 2 dargestellt.

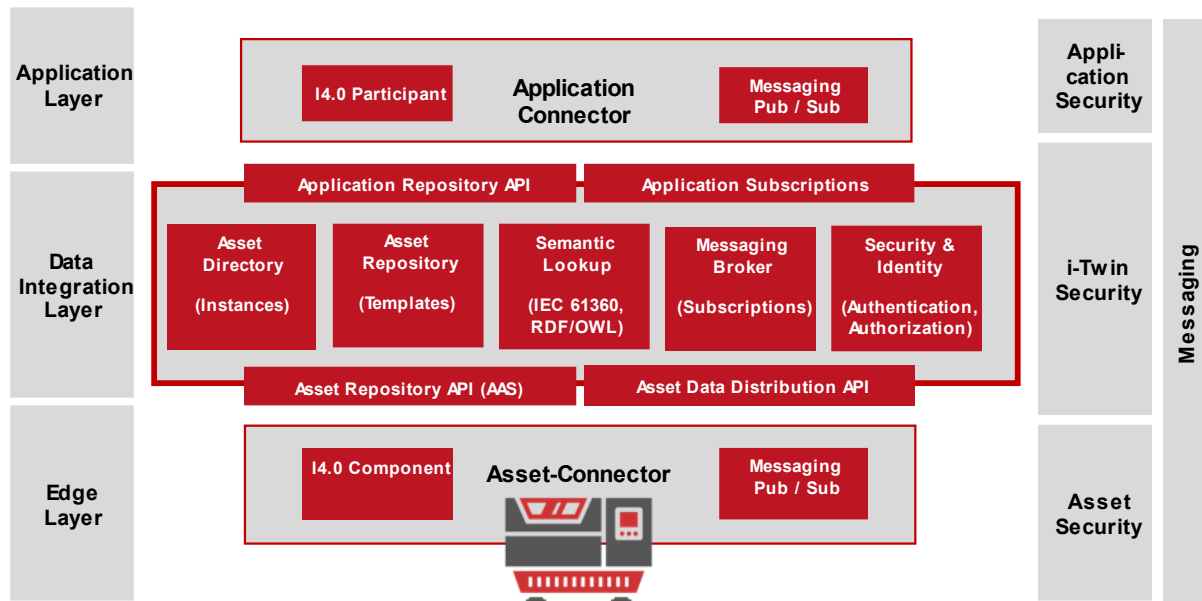


Abbildung 2: i-Twin Data Integration Layer

Für den Data Integration Layer umfasst zum Berichtszeitpunkt folgenden Bausteine:

Komponente	Beschreibung	Sourcecode
<b>Apache Kafka</b>	Messaging Infrastruktur – wird zur Kommunikation innerhalb der i-Asset Plattform verwendet. So werden Ereignisse innerhalb der i-Asset Plattform verteilt bzw. teilnehmende Anwendungen informiert.	<a href="https://kafka.apache.org/">https://kafka.apache.org/</a>
<b>MQTT Broker</b>	Alternative Messaging Infrastruktur – wird zur Kommunikation zwischen Asset Repository und den angeschlossenen Anwendungen verwendet.	<a href="https://mosquitto.org/">https://mosquitto.org/</a>
<b>MongoDB</b>	NoSQL-Datenbank zur Speicherung von AAS-Dokumenten	<a href="https://www.mongodb.com/">https://www.mongodb.com/</a>
<b>Keycloak</b>	Zum Verwalten aller angemeldeten Benutzer, der möglichen Rollen und der Berechtigungen auf der gesamten Plattform. Ein Administrator kann diese Informationen hier verwalten, Nutzer und Rollen editieren, unangemessenes Verhalten sanktionieren oder Zugriffsrechte vergeben. Das Projekt greift intern auf diese Informationen über den sogenannten Identity-Service zu.	<a href="https://www.keycloak.org">https://www.keycloak.org</a>
<b>asset-repository</b>	Entspricht dem i-Twin Semantic Integration Layer und enthält weitere Unterprojekte.	<a href="https://github.com/i-Asset/asset-repository.git">https://github.com/i-Asset/asset-repository.git</a>

<b>asset-repository-service</b>	<p>Micro-Service für das zentrale Management von Asset Administration Shells, Teilmodellen und Concept Descriptions.</p> <p>Diese Service stellt die zentrale Anlaufstelle für Edge-Devices und Anwendungen zur Verfügung, um erforderliche AAS-Informationen aber auch semantische Beschreibungen zu erhalten. Dieser Baustein implementiert das <i>Asset Administration Shell Repository Interface</i> sowie das <i>Submodel Repository Interface</i> gemäß Plattform Industrie 4.0 (AAS Part 2, 2023). In dieser Komponente ist auch das Directory Service enthalten. Dieses erlaubt den Edge-Devices bzw. Anwendungen, sich an zentraler Stelle zu registrieren. Dadurch können die aktiven I4.0 Komponenten im Netzwerk identifiziert werden.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/asset-repository-service">https://github.com/i-Asset/asset-repository/tree/development/asset-repository-service</a>
<b>asset-repository-api</b>	<p>Bibliothek die gemeinsam genutzte Funktionalität bereitstellt. Vor allem die im Standard (AAS Part 2, 2023) definierten REST-Interfaces sind hier definiert.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/asset-repository-api">https://github.com/i-Asset/asset-repository/tree/development/asset-repository-api</a>
<b>asset-repository-mongo</b>	<p>Stellt die Persistenz in Form der NoSQL-Datenbank MongoDB zur Verfügung.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/asset-repository-mongo">https://github.com/i-Asset/asset-repository/tree/development/asset-repository-mongo</a>
<b>semantic-lookup-service</b>	<p>Micro-Service für das zentrale Verwalten von IEC 61360 kompatible Taxonomien.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/semantic-lookup-service">https://github.com/i-Asset/asset-repository/tree/development/semantic-lookup-service</a>
<b>semantic-lookup-api</b>	<p>Gemeinsam genutzte Interfaces und Datenobjekte.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/semantic-lookup-api">https://github.com/i-Asset/asset-repository/tree/development/semantic-lookup-api</a>
<b>asset-connector</b>	<p>Verbindet Anwendungen und Edge-Devices mit der i-Twin Semantic Middleware.</p>	<a href="https://github.com/i-Asset/asset-repository/tree/development/asset-connector">https://github.com/i-Asset/asset-repository/tree/development/asset-connector</a>

<b>aas4j-model-generator</b>	Code Generation Tool for Java. Dieses Repository ist von <a href="https://github.com/eclipse-aas4j/aas-model-generator">https://github.com/eclipse-aas4j/aas-model-generator</a> abgeleitet. Mit diesem Projekt wird das AAS Meta-Modell gemäß Plattform Industrie 4.0 (AAS Part 1, 2023) generiert.	<a href="https://github.com/i-Asset/aas4j-model-generator/tree/feature/update-to-3.0.RC02">https://github.com/i-Asset/aas4j-model-generator/tree/feature/update-to-3.0.RC02</a>
<b>aas4j</b>	Eclipse AAS Model for Java. Dieses Repository ist von <a href="https://github.com/eclipse-aas4j/aas4j">https://github.com/eclipse-aas4j/aas4j</a> abgeleitet.	<a href="https://github.com/i-Asset/aas4j/tree/update-to-rc02">https://github.com/i-Asset/aas4j/tree/update-to-rc02</a>

Tabelle 1: i-Twin Middleware - Software Bausteine

Alle Bausteine sind als Open-Source-Projekt unter dem Github-Organisationsnamen <https://github.com/i-Asset> verfügbar und können hier eingesehen bzw. heruntergeladen werden. Der aktuelle Entwicklungsstand ist unter <https://iasset.salzburgresearch.at> verfügbar.

Die Software ist in unterschiedlichen Code-Repositories organisiert. Für die Entwicklung der i-Asset Plattform werden jeweils verschiedene Branches im Source-Code-Repository verwaltet. Dabei gilt:

- Development: Enthält die jeweils aktuellen Entwicklungsschritte. Dieser Branch ist als nicht stabil anzusehen und kann auch Fehler enthalten.
- Staging: Enthält die jeweils aktuellen, getesteten Entwicklungsschritte. Änderungen, die in diesen Branch eingepflegt werden, lösen einen automatischen Build der Plattform unter <https://iasset.salzburgresearch.at> aus.
- Master: Dieser Branch repräsentiert den aktuellen Release der i-Asset Plattform. Änderungen werden nur durch Zusammenführen des Staging-Banches durchgeführt. Direkte Änderungen in diesem Branch sind nicht erlaubt. Das Projekt hat noch keinen Release durchgeführt!

## 2.2 Asset- und Application Connectors

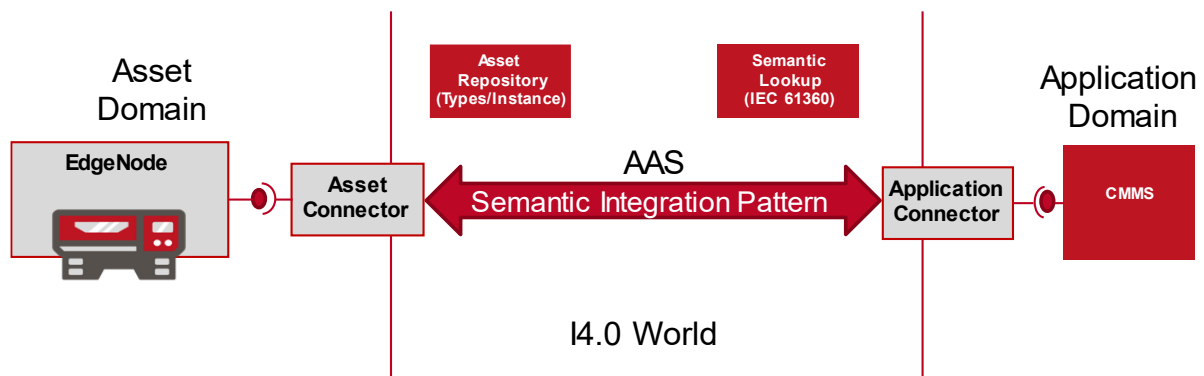
Die Systemarchitektur der semantischen Integrationsplattform sieht als Bindeglied zwischen dem Data Integration Layer auf der einen Seite und den Anwendungen und Assets auf der anderen Seite so genannte Connectors („**Application Connectors**“, „**Asset Connectors**“, s. Abbildung 2) vor. Diese Interface-Elemente haben die Funktion, das jeweilige Asset bzw. eine Anwendung im Fertigungsnetzwerk als Industrie 4.0 (I4.0) Komponente für andere Teilnehmer sichtbar zu machen und zu integrieren. Als reaktive I4.0 Komponente werden Maschinen bezeichnet, die Auskunft über ihren Zustand geben können und auch in der Lage sind, Operationen auszuführen. Proaktive I4.0 Komponenten können mit ihrer Umgebung kommunizieren. Die Application Connectors bilden die Grundlage für die Umsetzung eines zentralen Konzepts von i-Twin, der „**Semantic Integration Patterns**“. Die wesentlichen Merkmale dieser Semantic Integration Patterns sind hier im Überblick dargestellt:

- Repräsentation von vordefinierten Kommunikationskanälen der Teilnehmer (Assets, Anwendungen) eines Fertigungsnetzwerks  
 Beispiel: *Weiterleitung einer Störmeldung („Alarm“) von einer Anlage an ein Instandhaltungsmanagementsystem (CMMS)*
- Systemübergreifende Verbindung verschiedener Anwendungsdomänen

Beispiel: *Anforderung eines Machine Learning Modells zur Ausführung auf einem Edge-Device*

- Semantische Auszeichnung und Spezifikation sowohl der Kommunikations-Endpunkte (synchron, asynchron) der Teilnehmer eines Fertigungsnetzwerks als auch der ausgetauschten Nachrichten („Message-Payload“)  
Beispiel: *Anzeige der Wartungshistorie auf dem Bedienfeld einer Anlage*
- Verwendung existierender Industriestandards  
Beispiel: *IEC 61360, RAMI4.0/AAS (CDD, 2017) (Heidel, Hoffmeister, Hankel, & Döbrich, 2017), (DIN SPEC 91345, 2016)*

Anwendungs- bzw. Asset Connectors stellen somit die Laufzeitumgebung für Semantic Integration Patterns dar und bieten eine standardkonforme I4.0-Schnittstelle, die einen sicheren Zugriff auf Echtzeitdaten ermöglicht, aber auch Anfragen zum Aufruf von Methoden akzeptiert. Die Connectors interagieren direkt mit der Steuerung oder dem Edge Controller der Anlage oder verwenden OPC UA oder ähnliche Protokolle, um Status- und Sensorinformationen von der Anlage zu erhalten (z.B. um den Wert einer Eigenschaft zu erhalten oder zu aktualisieren oder um einen Steuerbefehl aufzurufen). Auf die gleiche Weise stellen Application Connectors eine standardkonforme Schnittstelle für Anwendungen zur Verfügung wie Asset Connectors. Sie transformieren sowohl die Anwendungsmethoden als auch die ausgetauschten Daten in die standardisierte I4.0-Welt. Ziel ist es, Methodenaufrufe wieder in die standardisierte I4.0-Welt zu ermöglichen, natürlich unter Anwendung von Sicherheitseinstellungen.



**Abbildung 3: Integration von Anlagen und Anwendungen mittels Semantic Integration Patterns**

Beide Arten von Connectors erfordern ein AAS-Teilmodell, das die Runtime-API (AAS Part 2, 2023) und, noch wichtiger, die Struktur der ausgetauschten Daten (AAS Part 1, 2023) festlegt. Eine Anlage oder Anwendung, die ihre Informationen auf diese Weise offenlegt, wird als aktive I4.0-Komponente bezeichnet, die einen standardisierten Zugang und standardisierte Daten bietet. Die AAS-Spezifikation liefert die Modellelemente zur Beschreibung der Eigenschaften eines Assets (bzw. einer Anwendung) und seiner Kommunikationsfähigkeiten.

Aktuell sind Connectors zu folgenden Bausteinen prototypisch verfügbar:

Komponente	Beschreibung	Sourcecode
<b>Förderband</b>	Asset-Connector welcher den Zugriff auf Maschinendaten des Förderbands im Labor ermöglicht. Zusätzlich ist auch die Steuerung des Förderbandes mit einfachen Befehlen möglich.	<a href="https://github.com/i-Asset/asset-connector-belt-example/tree/development">https://github.com/i-Asset/asset-connector-belt-example/tree/development</a>



<b>Peak2Pi</b>	Baustein welcher die OEE-Werte von Maschinen AAS konform aufbereitet. Zusätzlich kann dieser Baustein noch Störmeldungen an die i-Twin Middleware publizieren und stellt somit eine erste Realisierung eines Semantic Integration Patterns zur Verfügung	<a href="https://github.com/i-Asset/asset-repository/tree/feature/icosense-p2pi-impl">https://github.com/i-Asset/asset-repository/tree/feature/icosense-p2pi-impl</a>
<b>isproNG</b>	Baustein zum „Auffangen“ von Störmeldungen. Dieser Baustein nimmt Störmeldungen im definierten AAS Format entgegen und repräsentiert somit das funktionale Gegenstück für die Realisierung von Störmeldungen mittels Semantic Integration Pattern.	<a href="https://github.com/i-Asset/asset-repository/tree/huh/IsproNG-Connector">https://github.com/i-Asset/asset-repository/tree/huh/IsproNG-Connector</a>
<b>INNIO KSS Anlage</b>	Stellt einen annotierten Datenstrom mit den aktuellen Werten der Kühl- und Schmierstoff-Anlage bereit. Dieser Datenstrom wird dazu benutzt, um Ereignisse auslösen zu können aber auch um Daten an das existierende Dashboard zu übermitteln.	Derzeit nicht veröffentlicht

### 3 Ergänzende Projektberichte

Der vorliegende Bericht beschreibt die Komponenten der Middleware Plattform für die semantische Anwendungsintegration durch Semantic Integration Patterns („i-Twin Middleware“) mit Stand vom Dezember 2023.

Die Middleware Plattform wird in folgenden Berichten ergänzt und in vertiefender Form beschrieben:

- D2.2 „Semantic Integration Patterns for Manufacturing“
- D2.3 „Semantic Integration Patterns for Artificial Intelligence“
- D2.4 „Final System Design“
- D3.2 „Asset Connectors“
- D3.3 „Application Connectors“

## 4 Literaturverzeichnis

- AAS Part 1. (2023, April). *AAS Part 1: Metamodel*. Retrieved 14. September 2023, from IDTA - Der Standard für den Digitalen Zwilling: [https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01001-3-0\\_SpecificationAssetAdministrationShell\\_Part1\\_Metamodel.pdf](https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf)
- AAS Part 2. (2023, Juni). *AAS Part 2: Application Programming Interfaces*. Retrieved September 2023, from IDTA - Der Standard für den Digitalen Zwilling: [https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01002-3-0\\_SpecificationAssetAdministrationShell\\_Part2\\_API\\_.pdf](https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01002-3-0_SpecificationAssetAdministrationShell_Part2_API_.pdf)
- CDD. (2017). *IEC61360 Common Data Dictionary (CDD)*. Von [https://en.wikipedia.org/wiki/IEC\\_61360](https://en.wikipedia.org/wiki/IEC_61360) abgerufen
- DIN SPEC 91345. (2016). *Reference Architecture Model Industrie 4.0 (RAMI 4.0)*. Abgerufen am 27. Juli 2022 von <https://www.beuth.de/technische-regel/din-spec-91345/250940128>
- Heidel, R., Hoffmeister, M., Hankel, M., & Döbrich, U. (2017). *Industrie 4.0 - Basiswissen RAMI4.0*. VDE Verlag GMBH.

## Impressum

Titel	i-Twin Middleware
Bezeichnung	Deliverable 3.1a (i-Twin)
Autoren	Dietmar Glachs (Salzburg Research Forschungsgesellschaft m.b.H.)
Dateiname	D31b_Middleware_final.docx
Publikationsstatus	Eingeschränkt auf Konsortialpartner
Letzte Änderung	29.01.2024 von Georg Güntner
Kontakt	Salzburg Research Forschungsgesellschaft m.b.H. Herr DI Georg Güntner Jakob Haringer Straße 5/3   5020 Salzburg   Austria T +43-662-2288-401   <a href="mailto:georg.guentner@salzburgresearch.at">georg.guentner@salzburgresearch.at</a>
Copyright	Projektkonsortium i-Twin, Jänner 2024 p.a. Salzburg Research Forschungsgesellschaft m.b.H. Jakob Haringer Straße 5/3   5020 Salzburg   Austria T +43-662-2288-401   <a href="mailto:i-twin-office@salzburgresearch.at">i-twin-office@salzburgresearch.at</a>

Das Projekt i-Twin wird gefördert vom BMK und von der FFG aus Mitteln des Programms IKT der Zukunft.