abat consult

Freiraum für das Business von morgen.

WHITEpaper

Digitales Produktgedächtnis

Transparenz entlang des Produktlebenszyklus

©unsplash

Dieses Whitepaper beschreibt, wie durch Sensoren physische Objekte mit digitalen Identitäten versehen werden können. Welche Konzepte bereits angewandt werden, wie die technische Umsetzung im Detail aussieht und wie die Anbindung an SAP-Systeme abläuft.

AUTOR
CARSTEN MANSKE

AUTOR
RENÉ KESSLER

rene.kessler@abat.de

Was ist ein digitales Objektgedächtnis?

Wird ein Objekt, das eine digitale Identität besitzt mit einer Gedächtnisfunktion (Speicher) ausgestattet, dann spricht man von einem digitalen Objektgedächtnis. Die Grundlage, um Objekte mit einer digitalen Identität zu versehen, entstammt aus dem Themenbereich Industrie 4.0.

Ein zentraler Aspekt dabei ist die Anbindung von Produkten, Anlagen oder weiteren Sensoren an ein Netzwerk, um diese Objekte an weitere Systeme koppeln zu können. Die Kombination aus Anlage, Sensoren und angebundenem System kann dann in der Folge als cyberphysisches System verstanden werden. Neben der Netzwerkkomponente ist in diesem Zuge insbesondere die Nutzung der Internet-of-Things-Devices (IoT-Devices) von großer Bedeutung, da durch das IoT physikalische Objekte (Sensoren) mit dem Internet verbunden werden können und so eine Kommunikation zwischen Mensch und Objekt sowie Objekt und Objekt realisiert wird. Durch die Nutzung von IoT-Devices können auch zuvor nicht-digitale Anlagen oder Objekte eine digitale Identität erhalten und dadurch Mehrwerte realisiert werden:

Kommunikation und Kooperation: Objekte können mit Ressourcen im Netzwerk verbunden werden. Es können Verknüpfungen zwischen Objekten und Systemen realisiert werden, um gezielt Daten auszutauschen und Dienste zu nutzen. Dabei kommen vor allem Funktechnologien, wie WiFi, GSM/UMTS oder WPAN zum Einsatz.

Adressierbarkeit: Durch die Anbindung an das Netzwerk können Objekte aus der Ferne adressiert und Daten oder Services angefragt werden.

Identifikation: Objekte sind durch eine definierte ID über das Netzwerk eindeutig identifizierbar. Darüber hinaus können Technologien wie RFID, NFC oder Barcodes genutzt werden, um auch passive Objekte zu identifizieren. Dadurch können sowohl aktive als auch passive Gedächtnisse realisiert werden.

Sensorik: Objekte können mithilfe von Sensoren Informationen über sich und ihre Umwelt sammeln, aufzeichnen und über die Netzwerkanbindung an Systeme kommunizieren.

Eingebettete Informationsverarbeitung: Objekte besitzen Prozessoren, Mikrocontroller und eigene Speicherkapazität. Dadurch können erhobene Daten direkt verarbeitet und in Diensten operationalisiert werden.

Lokalisierung: Objekte kennen ihren physischen Standort oder sind lokalisierbar. Hierbei können Technologien wie GPS, das Mobilfunknetz oder Triangulationsverfahren zum Einsatz kommen.

Benutzungsschnittstellen: Smart Objekte können mit Menschen auf direkter oder indirekter Weise kommunizieren (z. B. über eine Applikation).

Unterschieden werden kann weiterhin zwischen passiven und aktiven Gedächtnissen. Während passive Gedächtnisse auf einfacher Identifikation eines Objektes und der Verarbeitung in der Systemumgebung basieren, werden bei aktiven Gedächtnissen die Daten direkt am Objekt erhoben und verarbeitet. Bei zu produzierenden Produkten wird das Objektgedächtnis oftmals auch als Produktgedächtnis bezeichnet und begleitet den kompletten Lebenszyklus eines Produktes, angefangen bei der Produktdefinition und -konzeption über die Herstellung bis hin zur Entsorgung und zum Recycling des Produktes.

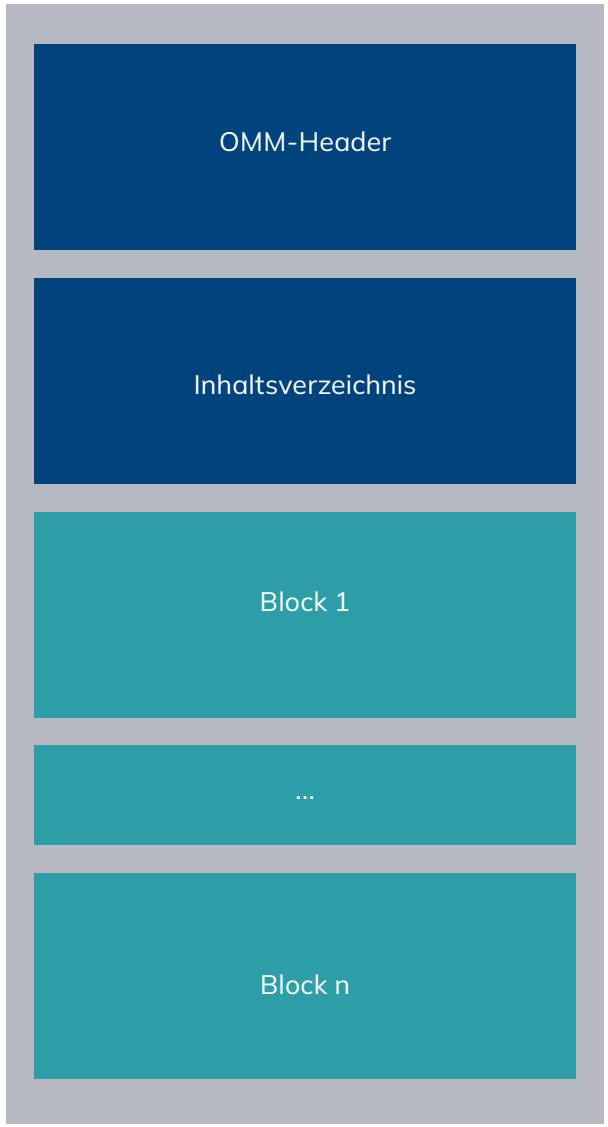


Abbildung 1: Struktur eines OMM-Produktgedächtnisses

Welche Konzepte und Ideen existieren?

Zur Umsetzung eines Digitalen Produktgedächtnises (DPGs) wurden in Forschung und Praxis verschiedene Konzepte erarbeitet. Am weitesten verbreitet ist dabei das Object Memory Model (OMM), welches von einer W3C-Inkubator-Gruppe entwickelt wurde (u. A. DFKI, SAP, Siemens). Mithilfe dieses Modells können Ereignisse und Informationen für oder von physischen Objekten modelliert werden. Die Grundlage des OMM ist die Auszeichnungssprache XML, mit der es möglich ist, Daten in strukturierter und hierarchischer Art und Weise in Form einer Textdatei darzustellen. In der Architektur des OMM werden alle Informationen in Blöcken gespeichert. Dadurch sind die Informationen aufgeteilt und somit enthält ein einzelner Block nur Fragmente des gesamten Datenpools. Jeder Block stellt darüber hinaus Metadaten bereit, die wiederum dazu genutzt werden können, die Informationen in der Liste an Blöcken zu lokalisieren. Diese Liste enthält neben den Blöcken zusätzlich einen Header und ein optionales Inhaltsverzeichnis.

OMM-Header

Jedes OMM beginnt mit dem OMM-Header. Dieser enthält zum einen Meta-Informationen zur verwendeten Version des OMM-Standards und zum anderen die primäre ID des Gedächtnisses. Die ID eines Gedächtnisses wird dabei beim OMM als optional betrachtet, ist aber sofern sie einmal definiert wurde, nicht mehr veränderbar. Ist eine ID definiert, kann ein Gedächtnis mit anderen Gedächtnissen, Blöcken oder Strukturen in Relation gesetzt werden. Darüber hinaus kann im OMM-Header ein Verweis auf Blöcke hergestellt werden, die außerhalb des jeweiligen Gedächtnisses gespeichert werden, beispielsweise um selten genutzte Blöcke auszulagern oder Blöcke teilen zu können. Solche Verweise und Relationen lassen sich dabei in drei Arten herstellen:

1. **Verweis auf ein anderes Gedächtnis:**
   können externe Blöcke (oder ganze Gedächtnisse) in ein anderes Gedächtnis integriert werden. So lassen sich Gedächtnisse auf verschiedene Speicherorte verteilen, was insbesondere bei mehreren Gedächtnissen, mit großen Schnittmengen, einen Vorteil darstellen kann.

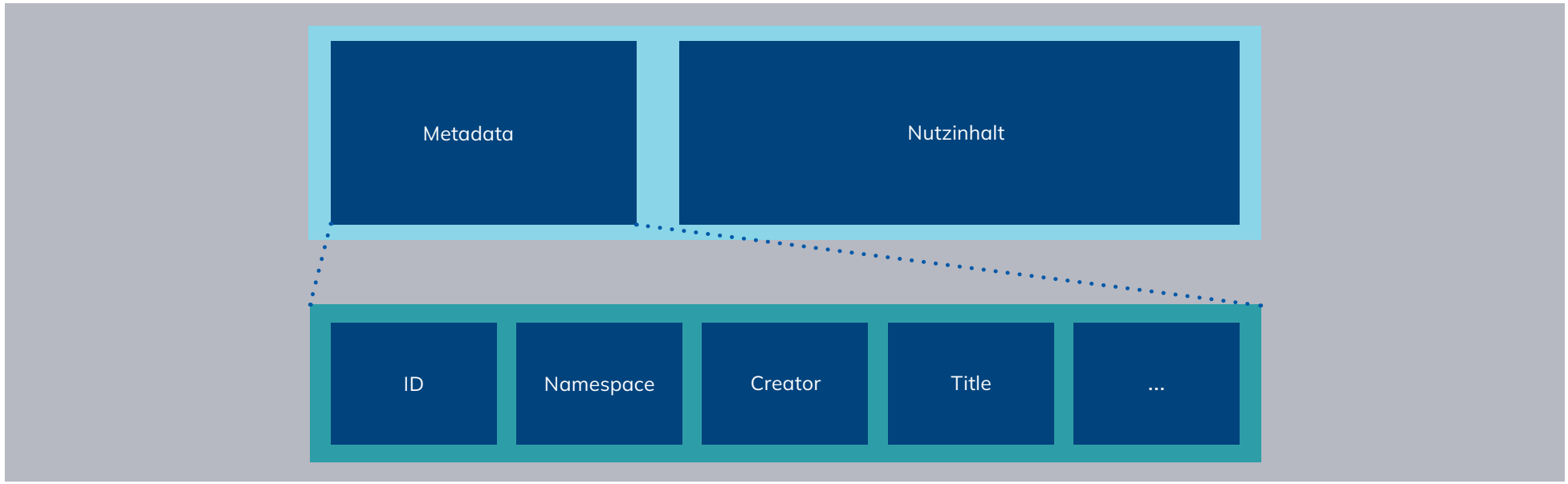
2. **Verweis auf einen dedizierten Block eines anderen Gedächtnisses**:
   In diesem Fall können einzelne Blöcke eines externen Gedächtnisses integriert werden, womit die Redundanz von Blöcken reduziert und die Erreichbarkeit erhöht werden kann.

3. **Verweis auf einen externen Speicherplatz für OMM-Blöcke**:
   Gerade bei größeren oder nur sehr selten genutzten Blöcken kann es von Vorteil sein, diese auf einem externen Speicherplatz zu verwalten.

OMM-Block

Jeder Block eines OMM-Gedächtnisses enthält Informationen zu einem spezifischen Themengebiet. Alle Blöcke eines Gedächtnisses sind gleichberechtigt, d.h. eine Gewichtung oder Priorisierung von Blöcken ist, abgesehen von der OMM-Struktur, im Standard nicht vorgesehen. Gleichermaßen stellt die Reihenfolge der Blöcke nicht zwingend eine logische Ordnung dar. Jegliche Identifizierung eines Blocks kann über seine enthaltenen Metadaten erfolgen. Diese Metadaten können dabei verschiedene Parameter enthalten[1] :

- **ID**: eindeutige Identifikation eines Blockes im Gedächtnis
- **Namespace** (**optional, sofern Format angegeben wird**): definiert den Inhalt und das Format eindeutig
- **Format** (**optional, sofern Namespace angegeben wird**): definiert das Format und die Kodierung des Nutzinhalts eines Blocks als MIME-Typ
- **Creator**: enthält das Erstellungsdatum des Blocks sowie die Identität des Erstellers
- **Contributor** (**optional**): enthält das Veränderungsdatum des Blocks und die Identität des Editors
- **Title**: stellt einen s, gegebenenfalls mehrsprachigen Titel für den Block zur Verfügung
- **Description** (**optional**): bietet einen, eventuell mehrsprachigen Text zur Beschreibung des Blocks
- **Type** (**optional**): gibt den Typ eines Blocks an
- **Subject** (**optional**): Beschreibung des Nutzinhalts eines Blocks (in RDF oder OWL-Format), Hierarchien können als Strings genutzt werden
- **Link** (**optional**): ein Link kann die Position des externen Speicherorts definieren, sofern die Nutzdaten eines Blocks nicht direkt in der XML-Struktur des Blocks eingebettet sind
- **Payload** (**optional, sofern Link vorhanden ist**): stellt die Struktur zur Aufnahme von Nutzinhalten dar, die direkt im Gedächtnis abgelegt werden



1 Haupert et al. (2011): Object Memory Modeling: W3C Incubator Group Report. 26.09.2011.

Abbildung 2: Aufbau eines OMM-Blocks

OMM-Inhaltsverzeichnis

Die dritte Datenstruktur eines OMM stellt das optionale Inhaltsverzeichnis dar. Immer dann, wenn es unerwünscht ist, das gesamte Gedächtnis einzulesen oder es dabei zu Performanceproblemen kommen könnte (z. B. bei sehr großen Gedächtnissen), kann das Inhaltsverzeichnis einen Mehrwert darstellen und den Kommunikationsaufwand reduzieren. Das Inhaltsverzeichnis benötigt dabei zwingend die ID des Blocks, um den Block und den Eintrag im Inhaltsverzeichnis miteinander zu verknüpfen. Darüber hinaus müssen zwingend der Namespace oder das Format sowie das Attribut, der Creator und ein Title vorliegen. Alle weiteren Metadaten sind optional.

OMM-Blöcke

Blöcke stellen das Kernelement eines Produktgedächtnisses dar. Im OMM-Standard sind dabei verschiedene Vorlagen zur Darstellung des Nutzinhaltes eines Blocks definiert. Nach der Erweiterung des Standards zum OMM+ sind insgesamt sechs Vorlagen für Blöcke vorhanden:

- **Struktur-Block**: Der Struktur-Block ermöglicht es Nutzern und Anwendungen, Relationen zwischen Objekten zu erzeugen. Beispielsweise werden so in der Produktion hierarchische Beziehungen verdeutlicht (Objekt A ist Teil von Objekt B). Darüber hinaus enthält jede Relation eine Zeit, ein Datum sowie eine Gültigkeit der Relation.

- **IDs-Block**: Die Aufgabe eines IDs-Blocks ist es, verschiedene IDs eines Objektes an einer zentralen Stelle zu bündeln, bzw. zu vereinheitlichen, damit ein Objekt mit mehreren unterschiedlichen IDs klar identifiziert werden kann.

- **Schlüssel-Wert-Paar-Blockvorlage**: Diese Vorlage erweitert den Nutzinhalt um eine Schlüssel-Wert-Struktur, zur Ablage beliebiger Daten. So können z.B. in Produktionsprozessen bestimmte Rohstoffe in ihrer spezifischen Menge gespeichert werden.

- **OMM+ Semantik-Block**: Ein Semantik-Block erweitert den Strukturblock um einen offenen und flexibleren Ansatz. Im Gegensatz zum Strukturblock, der eine fixe Liste an Relationen besitzt, kann im Semantik-Block der Typ der Relation frei gewählt werden und entweder für das ausgewählte Gedächtnis frei definiert oder von einer bestehenden Ontologie (RDF/OWL) importiert werden.

- **OMM+ Embedded-Block**: Über einen Embedded-Block kann ein komplettes Gedächtnis in einem Block gespeichert werden. Das ist insbesondere dann nützlich, wenn das Gedächtnis zuvor auf einem externen Speichermedium – z.B. einem RFID-Tag – gespeichert wurde und dieses in Zukunft nicht mehr erreichbar ist.

- **PiVis – Plugin Block**: Dieser Block stellt eine Besonderheit dar, da er nur für die Visualisierung eines Objektes im Visualisierungsframework PiVis benötigt wird. Mithilfe dieses Blocks lassen sich außerdem weitere Plugins im Gedächtnis abzulegen.

© Connect-world, stock.adobe.com

Technische Umsetzungsmöglichkeiten

Das OMM basiert im Wesentlichen auf einer XML-Repräsentation der gespeicherten Daten, kann aber auch im RDFa- oder Microdata-Format dargestellt werden. Damit ist es möglich, eine einfach lesbare, erweiterbare Darstellung zu erreichen und semantische Informationen maschinenlesbar zu persistieren.

Wenn der physische Speicher eines Objektgedächtnisses nicht ausreicht, um alle Daten eines Objektes zu speichern oder wenn mehrere Gedächtnisse verwaltet werden sollen, dann kann ein Object Memory Server (OMS) genutzt werden, der erweiterte Kapazitäten und Ressourcen zur Verfügung stellt. Zum Schutz vor Manipulation und zum Herkunftsnachweis wird dabei mit Hashwerten und Zertifikaten gearbeitet. Durch die Auslagerung der Gedächtnisse auf einen Server ergeben sich zudem neue Funktionen, wie eine Versionsverwaltung oder ein Rechte- und Rollenmanagement, über das der Zugriff auf die verschiedenen Gedächtnisse verwaltet werden kann. Auch die Nutzung und Anbindung der Daten an weitere Systeme, z.B. ein ERP-System, kann durch die Implementierung von geeigneten Schnittstellen realisiert werden. Wird nun nach Standard vorgegangen, sind die wichtigsten Funktionen eines OMS folgende:

- **Ablage von Gedächtnisblöcken**: Die grundlegende Aufgabe des OMS ist, die Ablage und Bereitstellung von Objektgedächtnissen zu gewährleisten. Durch den Einsatz des OMM-Formats können Daten entweder direkt im jeweiligen Block abgelegt werden oder der OMS wird als Index verwendet und die Daten werden auf einem externen Datenspeicher ausgelagert.
- **Server für multiple Gedächtnisse**: Durch einen OMS können beliebig viele Gedächtnisse in einem System verwaltet werden.
- **Versionsverwaltung**: Stand-Alone wird das OMM auf einem RFID-Tag oder einem internen Speicher als XML-Datei oder Binärblock abgelegt. Wird das OMM dagegen über einen OMS verwaltet, kann eine Versionsverwaltung realisiert werden. Dadurch lassen sich über Metadaten gezielt ältere Gedächtniszustände abrufen, um bspw. Änderungen nachverfolgen zu können.
- Datenintegrität und Herkunftsnachweis: Zum Schutz vor Datenmanipulation bietet das OMM die Möglichkeit, Hashwerte als Metadaten zu speichern. Der OMS bietet sämtliche Datenübertragung nur über signierte XML-Zertifikate an.
- **Rollenbasierter Zugriff**: Durch den Einsatz der erwähnten Zertifikate lässt sich der Zugriff auf das Gedächtnis und einzelne Blöcke steuern. Das geschieht im OMS mithilfe von Black- und White-Lists.
- **Integriertes Aktivitätsmodul**: Dieses Modul erlaubt externen Anwendern, gedächtnis-relevante Operationen direkt im Gedächtnis ablaufen zu lassen.

Zusammenfassend stellt das OMM (bzw. OMM+/OMM++) und das darauf aufbauende OMS-Konzept einen Blueprint dar, um ein digitales Produktgedächtnis zu implementieren. Es handelt sich dabei nicht um ein fertiges Produkt, sondern lediglich um einen Standard, der genutzt werden kann, um ein entsprechendes System daran zu orientieren, bzw. den OMM-Standard in einem System zu verankern.

Implementierung eines Objektgedächtnisses auf Blockchain-Basis

Bei der Planung der Implementierung stand in unserem Vorhaben der Gedanke der Manipulationssicherheit und Nachvollziehbarkeit von Zustandsänderungen im Vordergrund, weshalb auch die Blockchain-Technologie immer mehr in den Fokus rückte. Die Idee war dabei klar: Ebenso wie die Blockchain ist auch das OMM in Blöcken strukturiert, weshalb es naheliegt, die Funktionalitäten eines OMS über einen dezentralen Ledger abzubilden.

Allerdings muss eine geeignete Blockchain-Technologie gleichzeitig bestimmte Kriterien erfüllen, um die OMM- und OMS-Struktur überhaupt abbilden zu können:

- Zugriffskontrolle
- Versionskontrolle
- Erweiterbarkeit
- Anpassbarkeit

Eigene Anforderungen wurden ebenfalls an die zu implementierende Lösung gestellt. So muss diese nicht nur effizient in Entwicklung und Betrieb sein, sondern auch zuverlässig funktionieren. Da die Kommunikation des Gedächtnisses mit einem SAP-System sichergestellt werden sollte, muss auch die Anbindung der Lösung an ein solches System sichergestellt werden. Durch die Anforderung der Zugriffskontrolle kommen bei der Vielfalt der verfügbaren Blockchain-Implementierungen nur solche in Betracht, die private und konsortiale Blockchains ermöglichen.

Im Rahmen der digitalen Innovationsplattform „Leonardo“ bietet die SAP aktuell verschiedene Blockchain-Frameworks, wie z. B. Hyperledger Fabric, Multichain und Quorum an. Aufgrund der weiten Verbreitung und Einbindung von weiteren Projekten, wurde Hyperledger Fabric als Blockchainkonzept ausgewählt. Grundsätzlich hätten aber auch die anderen genannten Frameworks genutzt werden können, da diese ebenso die zugrundeliegenden Anforderungen erfüllen können.

Anbindung an SAP-Systeme

Um die Anbindung an ein SAP-System zu realisieren, wurde das OMM mithilfe des Hyperledger Fabric-Frameworks anhand eines Szenarios aus dem Automotive-Bereich umgesetzt. Auf Basis des Anwendungsszenarios wurde die Annahme getroffen, dass die Identifizierung des Produkts über ein RFID-Tag erfolgt. Dieser RFID-Tag enthält eine eindeutige ID und einen Link zum Gedächtnisspeicher, in diesem Fall eine Blockchain.

Akteure aus der Umwelt des Produktgedächtnisses (z.B. Sensoren oder Lesegeräte) lesen die ID des Gedächtnisses aus. Mithilfe der ID können dann Daten aus oder in das Blockchainnetzwerk übertragen werden. Dies geschieht mittels einer API und Smart Contracts, wobei sich die API als Schnittstelle zwischen den Anwendungen der Akteure und dem Netzwerk mit seinen Smart Contracts darstellt und so die Kommunikation zwischen allen beteiligten Systemkomponenten ermöglicht. Innerhalb des Netzwerks erlauben die Smart Contracts eine Verarbeitung von Anfragen. Die Ledger im Netzwerk enthalten die Daten des Produktgedächtnisses. Zwingend erforderlich ist dabei das Hauptledger, in dem alle digitalen Produktgedächtnisse mit einer ID angelegt werden müssen.

In der praktischen Anwendung kommt es oftmals vor, dass nicht alle Prozessbeteiligten auch einen vollen Zugriff auf alle Daten bekommen sollten, sondern nur bestimmte sehen und verändern dürfen. Dies kann über das Implementieren von verschiedenen Channels realisiert werden. Jeder Channel verfügt wiederum über ein eigenes, definierbares Rechte- und Regelmanagement. Durch den hierarchischen Aufbau dieser Channel kann ein vollumfängliches Rollenmanagement implementiert werden.

Anwendungsszenario

Das vorgestellte Konzept eignet sich immer besonders dann, wenn viele verschiedene Teilnehmer am Lebenszyklus eines Produktes beteiligt sind, das Produkt eventuell den Besitzer wechselt und die dabei und während der Nutzung erzeugten Daten miteinander geteilt werden sollen, unter der Einschränkung, dass nicht jeder Teilnehmer alle Daten einsehen oder verändern darf. Um den Nutzen eines solchen Produktgedächtnisses noch eindeutiger demonstrieren zu können, haben wir ein stark vereinfachtes Szenario aus dem Automotive-Bereich definiert.

Die Automobilindustrie basiert oftmals auf einer großen und komplexen Supply Chain, in der dementsprechend verschiedene Lieferantenbeziehungen vorliegen. Wobei an vielen Stellen Daten erzeugt, verarbeitet und verändert werden – dies über den Produktlebenszyklus hinweg: Angefangen bei der Produktion der Rohstoffe über deren Weiterverarbeitung zu Bauteilen, der Fertigung des Automobils bis hin zu dessen Nutzung durch den Verbraucher sowie der Entsorgung am Ende des Lebenszyklus. Unser gewähltes Szenario beginnt beim Anlegen der Bestellung eines Fahrzeugs, bei der eine bestimmte Konfiguration des Fahrzeugs gespeichert wird. Der nächste Schritt innerhalb des Szenarios ist die Produktion des Automobils, bei der im Vorfeld zwischen verschiedenen Lieferanten ausgewählt werden muss. Ebenso werden für den Verkauf des gefertigten Produktes verschiedene Akteure, z.B. Vertriebsabteilungen, für verschiedene Kontinente angelegt. Selbiges gilt für den Transport der gefertigten Produkte. Darüber hinaus verfügt in unserem Szenario jedes Fahrzeug über einen Board-Computer, der das Fahrzeug lokalisieren und Betriebsdaten (z. B. Fahrtberichte) sammeln kann.

Als Ergebnis ist eine prototypische UI5-App entstanden, die von den am Produktlebenszyklus beteiligten Akteuren genutzt werden konnte, um Daten zu lesen oder einzugeben. Die Zugriffs-verwaltung erfolgte dabei über eine Authentifizierung in der App, womit der Nutzer und damit seine Rolle innerhalb des Blockchain-Netzwerks erkannt wurde. Dadurch wurde auch entschieden, welche Daten dieser Nutzer sehen durfte und ob dieser Nutzer über Schreibrechte verfügt. Sämtliche Produktaktivitäten (z.B. Produktion oder Entsorgung) wurden dabei simuliert, es konnte allerdings gezeigt werden, dass das Konzept des digitalen Produktgedächtnis auf die Blockchain übertragen werden konnte. Ferner konnten verschiedene Rollen mithilfe eines Rechtemanagements realisiert werden. Produktbezogene Daten konnten sicher persistiert werden. Ebenso konnte demonstriert werden, dass auch die Anbindung an ein SAP-System keine technische Hürde darstellt.

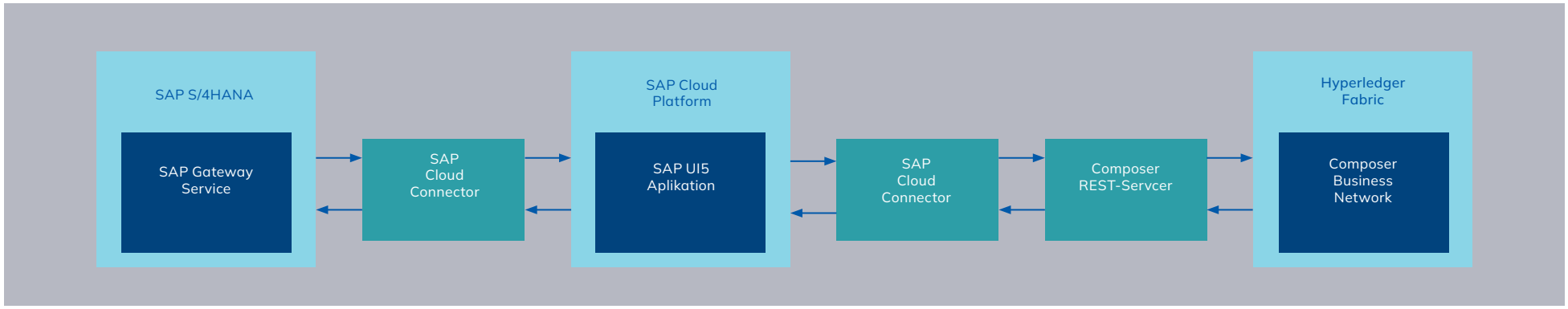


Abbildung 3: Überblick über die Systemumgebung des Prototyps

Wann lohnt sich der Einsatz eines digitalen Produktgedächtnisses mit einer Blockchain?

Der Einsatz eines digitalen Produktgedächtnisses, das Daten in einer Blockchain speichert, kann sich immer genau dann lohnen, wenn mehrere Interessensgruppen am Lebenszyklus eines Produktes beteiligt sind, Produktdaten konsumieren oder generieren und wenn zwischen diesen Akteuren ein gewisses Misstrauen (z.B. Manipulationsverdacht) besteht. Mit einer Blockchain kann erreicht werden, dass durch Sicherheitsmechanismen und dezentrale Datenhaltung die vorliegenden Daten zu jeder Zeit korrekt und konsistent sind.

Für einzelne Unternehmen würde sich der Einsatz einer Blockchain dagegen weniger eignen, zumal ein grundsätzliches Vertrauen zwischen den Abteilungen herrscht. Da das OMM-Konzept aber auch rein serverbasiert implementiert werden kann, stellt dies allerdings kein Argument gegen ein digitales Produktgedächtnis dar. Es kann nämlich dennoch ein sinnvoller Ansatz sein, um produktbezogene Daten über den ganzen Lebenszyklus hinweg zu vereinheitlichen. Auch zwischen beteiligten Akteuren, über eine Lieferkette hinweg, kann ein serverbasierter Ansatz vorteilhaft sein. Insbesondere dann, wenn ein Vertrauen zwischen allen Beteiligten herrscht oder zumindest ein Intermediär von allen beteiligten Parteien respektiert wird.

Gründe für ein digitales Produktgedächtnis

- Vereinheitlichung von Produktdaten
- Nachvollziehbarkeit über den gesamten Lebenszyklus hinweg
- Manipulationssicherheit
- Volle Kontrolle
- Transparenzsteigerung

ANSPRECHPARTNER

RENÉ KESSLER
rene.kessler@abat.de

ÜBER UNS

Die abat Gruppe, 1998 gegründet, ist SAP-Dienstleister, innovativer Softwareentwickler und Anbieter von Komplettlösungen für die softwaregestützte Prozessoptimierung – vor allem in den Kernbranchen Automotive und Diskrete Fertigung sowie in logistischen Prozessen und der Fertigungssteuerung. Mit unseren sechs Leistungsbereichen verschaffen wir Unternehmen den Freiraum, den sie für neue Ideen, effiziente Prozesse und zukunftsweisende Lösungen benötigen.

Im Leistungsbereich **consult** beraten und begleiten wir Sie in allen Phasen eines SAP-Projektes – von der Konzeption über die Implementierung bis zum Betrieb Ihres SAP-Systems. Dank abat **manufacture** erhalten Sie digitale Hochverfügbarkeitslösungen zur Produktionssteuerung im Bereich der komplexen Fertigungsindustrie. Mit abat **transform** bieten wir innovative und einzigartige Lösungen, die Sie besonders machen: von AI über Cloud bis zu X-Reality. Der Bereich **plm** hält übergreifende Prozessberatung bereit, mit dem Ziel, einen durchgängigen Datenfluss über PLM, ERP und MES zu erreichen. Angebote aus dem Bereich **protect** helfen Kunden dabei, Informationen zu schützen sowie Vertraulichkeit, Verfügbarkeit und Integrität in Geschäftsbeziehungen zu bewahren. Und schlussendlich beraten unsere Expert*innen des Bereichs **sustain** u.a. dazu, wie sich Nachhaltigkeit sowie die Berichterstattung darüber strategisch und strukturell im Unternehmen verankern lassen.

abat | An der Reeperbahn 10 | 28217 Bremen | +49 421 43 04 60 | www.abat.de