Was ist die Verwaltungsschale aus technischer Sicht?

Stand 04/2021

Dieses Dokument wurde von der Task Force „Sichere Verwaltungsschale“ der Plattform Industrie 4.0 erstellt. Die Task Force besteht aus Mitarbeitenden der Arbeitsgruppen „Referenzarchitekturen, Standards und Normung“ und „Sicherheit vernetzter Systeme“ (s.u.). In diesem Dokument wird das gemeinsame Verständnis der Task Force zusammengefasst, was die verschiedenen Ausprägungen der Verwaltungsschale sind. Dieses Dokument beschreibt die technische Sicht und befasst sich z.B. nicht mit Wertversprechen und Potentialen von Verwaltungsschalen oder Geschäftsmodellen.

Welche Dokumente definieren die Verwaltungsschale (engl. Abk.: AAS)?

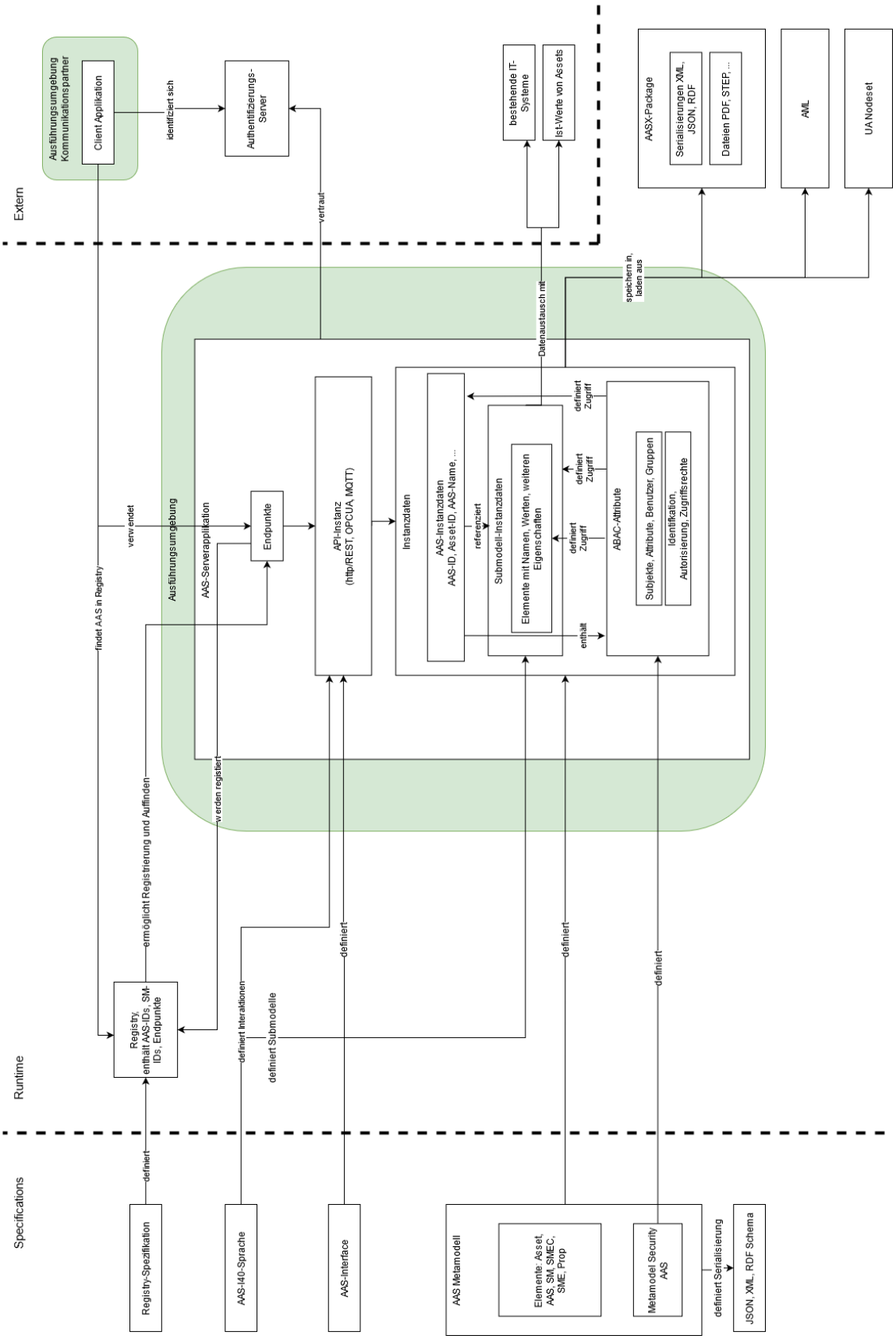
- Details of the Asset Administration Shell (AAS)
  - Teil 1: Metamodell, AASX Package und Serialisierungen nach XML/JSON/RDF/AML
  - Teil 2: http/REST API und weitere APIs, Infrastruktur mit Registry
- Die I40-Sprache
- OPC UA Companion Specification I4AAS
- Usage View of the AAS
- Asset Administration Shell - Reading Guide

Die Verwaltungsschale ist ein Sammelbegriff für folgende Aspekte:

- **AAS-Metamodell** als UML-Klassendiagramm
- **AAS-Metamodell-Serialisierungen** mit Schemata für XML, JSON oder RDF
- **AAS-Instanzdaten** (mit statischen und ggf. dynamischen Daten)
- **AAS-Instanzdaten-Serialisierungen** nach XML, JSON, RDF, AML oder als OPC UA Nodeset
- **AAS-Package-Container** (.AASX)
- **AAS-Serverapplikation**
- **AAS-API** (beispielsweise per http/REST, OPC UA oder MQTT)

Erweiterungen sind:

- AAS-OPC UA Modell I4AAS
- AAS-I40 Sprache
- AAS-Registry


Extern
Runtime
Specifications
Ausführungsumgebung Kommunikationspartner
Client Applikation
identifiziert sich
Authentifizierungs-Server
vertraut
bestehende IT-Systeme
Ist-Werte von Assets
AASX-Package
Serialisierungen XML, JSON, RDF
Dateien PDF, STEP, ...
AML
UA Nodeset
speichern in, laden aus
Datenaustausch mit
Ausführungsumgebung
AAS-Serverapplikation
Endpunkte
API-Instanz (http/REST, OPCUA, MQTT)
Instanzdaten
AAS-Instanzdaten AAS-ID, Asset-ID, AAS-Name, ...
referenziert
Submodell-Instanzdaten
Elemente mit Namen, Werten, weiteren Eigenschaften
enthält
definiert Zugriff
ABAC-Attribute
Subjekte, Attribute, Benutzer, Gruppen
Identifikation, Autorisierung, Zugriffsrechte
verwendet
findet AAS in Registry
ermöglicht Registrierung und Auffinden
werden registriert
Registry, enthält AAS-IDs, SM-IDs, Endpunkte
definiert Interaktionen
definiert Submodelle
definiert
Registry-Spezifikation
AAS-I40-Sprache
AAS-Interface
AAS Metamodell
Elemente: Asset, AAS, SM, SMEC, SME, Prop
Metamodel Security AAS
definiert Serialisierung
JSON, XML, RDF Schema

Die Abbildung stellt die obigen Aspekte und deren Zusammenhänge grafisch dar.

Das **AAS-Metamodell** definiert die möglichen Elemente zur Modellierung der AAS-Metamodell-Instanzen, z.B. Asset, AssetAdminstrationShell (AAS), Submodel (SM), SubmodelElementCollection (SMEC), Property und weitere SubmodelElement(s) (SME). Das technologieneutrale AAS-Metamodell als UML-Klassendiagramm wird für XML, JSON und RDF als Schemadatei serialisiert und abgespeichert.

AAS-Instanzdaten verwenden die im Metamodell beschriebenen Elemente, um Asset-Typen oder Asset-Instanzen zu beschreiben. Bei der Beschreibung von Teilmodellen werden Teilmodell-Templates und Teilmodell-Instanzen unterschieden. Teilmodell-Templates haben dabei typischerweise noch keine Werte. Teilmodell-Instanzen dagegen haben typischerweise Werte für die Teilmodell-Elemente. In AAS-Instanzdaten haben Asset, AAS und Submodel jeweils eine weltweit eindeutige ID. AAS-Instanzdaten beinhalten u.a. auch AAS-Operationen und -Events.

AAS-Instanzdaten können auf verschiedenen Wegen befüllt werden. Einerseits kann dies durch Laden von Serialisierungen erfolgen. Andererseits können Daten bei Bedarf aus bestehenden IT-Systemen abgerufen werden, die dann gemäß des AAS-Metamodells verwendet werden. Für Teilmodelle wird dabei ggf. aus verschiedenen IT-Systemen die Information abgerufen und zusammengestellt. Daten können z.B. als dynamische Werte von den zugehörigen Assets (z.B. Automatisierungsgeräten) abgerufen werden.

AAS-Instanzdaten können in verschiedene **AAS-Instanzdaten-Serialisierungen** formatiert werden, z.B. nach XML, JSON, RDF, AML oder in ein OPC UA Nodeset (gemäß **AAS-OPC UA Modell I4AAS**). Serialisierungen können als Dateien gespeichert und von dort wieder deserialisiert werden. Ein spezielle Serialisierung der AAS-Instanzdaten ist der **AAS-Package-Container** (Datei im AASX-Dateiformat). Ein AAS-Package-Container kann neben dem serialisierten XML oder JSON alle weiteren Dateien der AAS-Instanzdaten, z.B. Files wie PDFs, enthalten.

AAS-Instanzdaten können in eine eigene einzelne **AAS-Serverapplikation** geladen und als Speicherobjekt instanziiert werden. Alternativ besteht die Möglichkeit, dass eine AAS-Serverapplikation mehrere AAS-Instanzen verwaltet. AAS-Serverapplikationen können AAS-Instanzdaten ganzer AAS und/oder nur von Teilmodellen hosten. In einem Industrie-4.0-System interagieren mehrere dezentrale AAS-Serverapplikationen.

Auf AAS-Serverapplikationen kann über das **AAS-API** zugegriffen werden. AAS-Serverapplikationen können neben dem http/REST API ein OPC UA API (gemäß **AAS-OPC UA Modell I4AAS**) anbieten. Eine AAS-Serverapplikation mit http/REST API oder OPC UA API ist eine sogenannte reaktive AAS. Entsprechend kann über weitere Technologien wie z.B. MQTT ein AAS-API bereitgestellt werden.

Die Teilmodell-Instanzdaten sind entweder Teil der AAS-Instanzdaten oder separate, eigene Teilmodell-Instanzdaten. Diese können einzeln serialisiert und auf eigenen AAS-Serverapplikationen analog der AAS geladen werden, und auf diese kann ebenfalls per http/REST API oder OPC UA API zugegriffen werden.

Das AAS-API spezifiziert Services und API-Operationen, die das Lesen und Schreiben von Daten sowie den Aufruf von Funktionen (AAS-Operationen) beinhalten.

Zu jeder AAS-Serverapplikation einer AAS und zu jedem Teilmodell kann ein **Endpunkt** für den API-Zugriff gehören. Dieser Endpunkt ermöglicht unter einer definierten Adresse über die in der AAS-API spezifizierten Protokolle und Operationen den Zugriff auf eine AAS-Serverapplikation. Beispiele für Endpunkte sind „https://admin-shell-io.com:51410“ für http/REST oder „opc.tcp://192.168.1.40:4840“ für OPC UA.

Die **AAS-I40 Sprache** legt Interaktionsmuster und Teilmodelle fest, die proaktiv über die damit festgelegte Schnittstelle per http/REST, OPC UA oder MQTT kommunizieren. Eine AAS-Serverapplikation mit entsprechender Schnittstelle für die I40-Sprache ist gleichzeitig Server und Client und ist eine sogenannte proaktive AAS.

Zusätzlich wird eine Registry-Spezifikation erarbeitet, um Endpunkte und Beschreibungen für AAS-Serverapplikationen zu registrieren und zu finden.

In der oben genannten AAS-Registry werden zu AAS-Serverapplikationen für deren AAS-Instanzdaten und Teilmodell-Instanzdaten die IDs zusammen mit den zugehörigen Endpunkten gespeichert. So können diese per Anfrage an der Registry gefunden werden. Aktuell gibt es noch keine Festlegung, wie die Endpunkte der Registries selbst herstellerübergreifend gefunden werden können.

Für ein real ausführbares System müssen weitere Elemente vorhanden sein.

So ist eine **Ausführungsumgebung** (z.B. Hardware, CPU, Betriebssystem, Speicher, Festplatte) notwendig, in die ein Deployment der **Computerprogramme** erfolgen kann und in der diese ausgeführt werden. Die AAS-Serverapplikation und die zugehörigen Client-Applikationen sind solche Computerprogramme. In der Ausführungsumgebung können auch Serialisierungen und Package-Container gespeichert und aus ihr erzeugt werden.

Mehrere AAS-Serverapplikationen können in einem Computerprogramm implementiert werden. Mehrere Computerprogramme können wiederum in einer Ausführungsumgebung ausgeführt werden.

Security

Welche Dokumente definieren die Security der AAS?

- Security der Verwaltungsschale
- Details of the Asset Administration Shell Part 1, Chapter 6, Attribute Based & Role Based Access
- Access control for Industrie 4.0 components for application by manufacturers, operators and integrators
- Sicherer Downloadservice

Security der AAS

Für die Umsetzung von Industrie 4.0 ist Security in allen Konzepten unverzichtbar. In der Praxis muss sich die Umsetzung an den Sicherheitszielen der Anwendungen orientieren, z.B. dem Schutz von Daten, Safety oder Know-how. Auch wenn Security konzeptionell immer vorgesehen sein muss, kann sie je nach Sicherheitsziel, Anwendung und Einsatzumgebung unterschiedlich stark umgesetzt sein.

Im **AAS-Metamodell** ist die Möglichkeit enthalten, eine **attributbasierte Zugriffssteuerung** (ABAC) zu modellieren. Hierzu gehören die Identifikation von Benutzern und Benutzergruppen sowie weiteren Attributen und zugeordneten Rechten, wobei eine beliebige Granularität bis hin zur Verwaltung einzelner Elemente einer AAS erreicht werden kann. Die Parametrierung der Zugriffssteuerung erfolgt innerhalb des spezifizierten Security-Modells. Weiterhin sind Merkmale für die sichere Identifikation von Benutzern und Client-Applikationen vorgesehen, auf denen **Authentifizierung** und **Autorisierung** basieren. Client Applikationen und AAS Serverapplikationen weisen über solche Merkmale ebenfalls ihre Echtheit nach.

Ein **AAS-Package-Container** (.AASX) ist im Open Packaging Conventions (OPC)-Format umgesetzt. Das OPC-Format beschreibt die enthaltenen Elemente in XML und unterstützt die Security-Elemente zur Authentifizierung entsprechend XMLSIG. Ein AAS-Package-Container kann die **Authentizität** nicht schützen. Die Authentizität kann aber beim Einlesen überprüft werden. Ein AAS-Package-Container kann die **Vertraulichkeit** nicht schützen. Hierzu müssen die zu schützenden Teile oder die ganze Containerdatei verschlüsselt werden. Die Verschlüsselung kann dabei für einen bestimmten Empfänger gedacht sein oder im Fall des Digital Rights Managements (DRM) für eine Gruppe von Empfängern, die zum Zeitpunkt der Verschlüsselung möglicherweise noch gar nicht bekannt ist.

Die **AAS-Serverapplikation** stellt die technische Implementierung der **AAS-API** bereit. Die technisch korrekte und **sichere Implementierung** der AAS-Serverapplikation und der **Ausführungsumgebung** ist die Voraussetzung für den dauerhaft sicheren Betrieb. Für die sichere Implementierung müssen sich AAS-Serverapplikation und Ausführungsumgebung an Standards wie dem sicheren Entwicklungsprozess der IEC 62443-4-1 und der sicheren Betriebsprozesse nach ISO 27000 orientieren. Entsprechend der in der AAS enthaltenen und/oder der für den Anwendungsfall in der AAS-Serverapplikation festgelegten Regeln wird die **Zugriffssteuerung** umgesetzt. Hierzu muss entweder die AAS-Serverapplikation selbst die Authentifizierung des Kommunikationspartners übernehmen oder einen zentralen Dienst anfragen. Danach wenden die in der AAS-Serverapplikation enthaltenen Funktionen zur Zugriffssteuerung die Regeln aus dem Security-Modell an.

Für die Kommunikation mittels **AAS-API** müssen Protokolle eingesetzt werden, die die notwendigen **Sicherheitsmechanismen** bereitstellen. Hierbei werden http/REST über HTTPS und OPC UA berücksichtigt. Je nach Protokoll wird z.B. auf TLS (Transportation Layer Security) auf der Transportschicht zurückgegriffen. Methoden zur Authentifizierung können X.509-Zertifikate nutzen und/oder etablierte Methoden zum **Identity- und Access Management (IAM)** mittels

spezieller Endpunkte oder Token anwenden. Die AAS-Serverapplikation bietet die **AAS-API** so an, dass Authentizität der Kommunikationspartner sowie die Vertraulichkeit und Authentizität der Kommunikation entsprechend des Schutzbedarfs umgesetzt werden. Der Schutzbedarf orientiert sich an den Sicherheitszielen und dem Anwendungsfall, so dass Security-Maßnahmen mehr oder weniger ausgeprägt sein können.

Die obige Zusammenfassung gibt nach Ansicht der Autoren den aktuellen Diskussionsstand der Plattform Industrie 4.0 wieder.

Autoren

Sebastian Bader	Fraunhofer IAIS
Vanessa Bellinghausen	Bundesamt für Sicherheit in der Informationstechnik
Dr. Birgit Boss	Robert Bosch GmbH
André Braunmandl	Bundesamt für Sicherheit in der Informationstechnik
Dr. Gerd Brost	Fraunhofer AISEC
Björn Flubacher	Bundesamt für Sicherheit in der Informationstechnik
Kai Garrels	ABB STOTZ-KONTAKT GmbH
Dr. Michael Hoffmeister	Festo SE & Co. KG
Dr. Lutz Jänicke	PHOENIX CONTACT GmbH & Co. KG
Michael Jochem	Robert Bosch GmbH
Andreas Orzelski	PHOENIX CONTACT GmbH & Co. KG
Jens Vialkowitsch	Robert Bosch GmbH
Thomas Walloschke	Industrie KI GmbH
Jörg Wende	IBM Deutschland GmbH

Kontakt: Geschäftsstelle der Plattform Industrie 4.0, Bülowstraße 78, 10783 Berlin

geschäftsstelle@plattform-i40.de

www.plattform-i40.de