



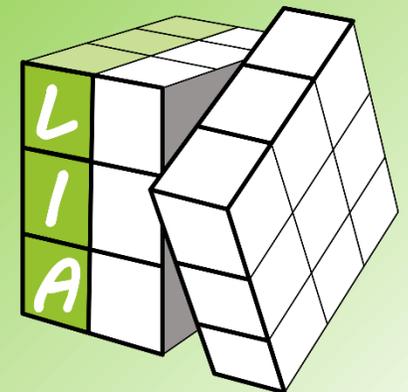
FAKULTÄT FÜR  
ELEKTROTECHNIK UND  
INFORMATIONSTECHNIK

## Interoperabilität und Semantik beim Digitalen Zwilling, bei der Verwaltungsschale – wo stehen wir?

TuLAut 15.10.2021: Der Digitale Zwilling:

Ansichten und Einsichten aus Automatisierungstechnik-Professuren

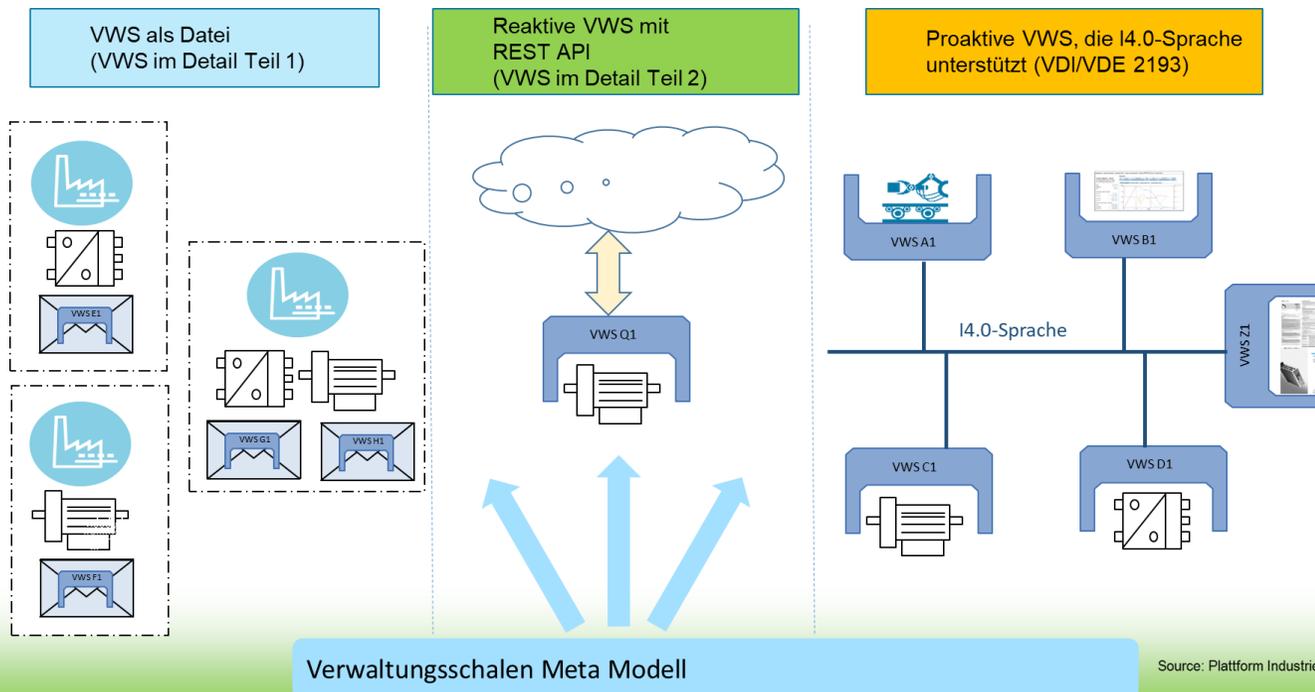
Christian Diedrich, Tizian Schröder, Alexander Belyaev, Chris Urban,  
Harish Kumar Pakala, Thomas Werner



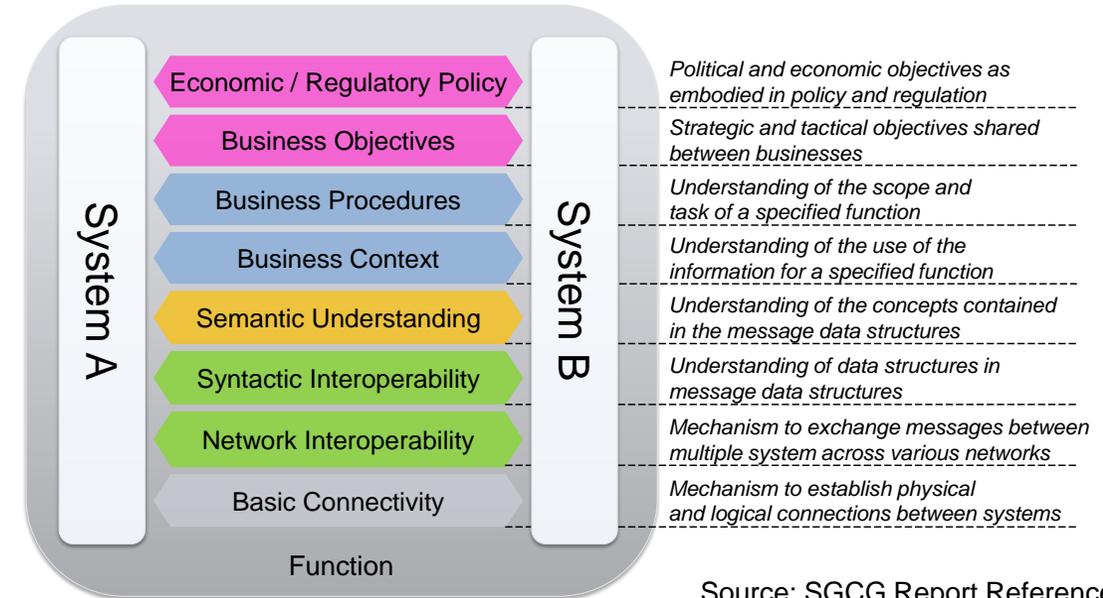
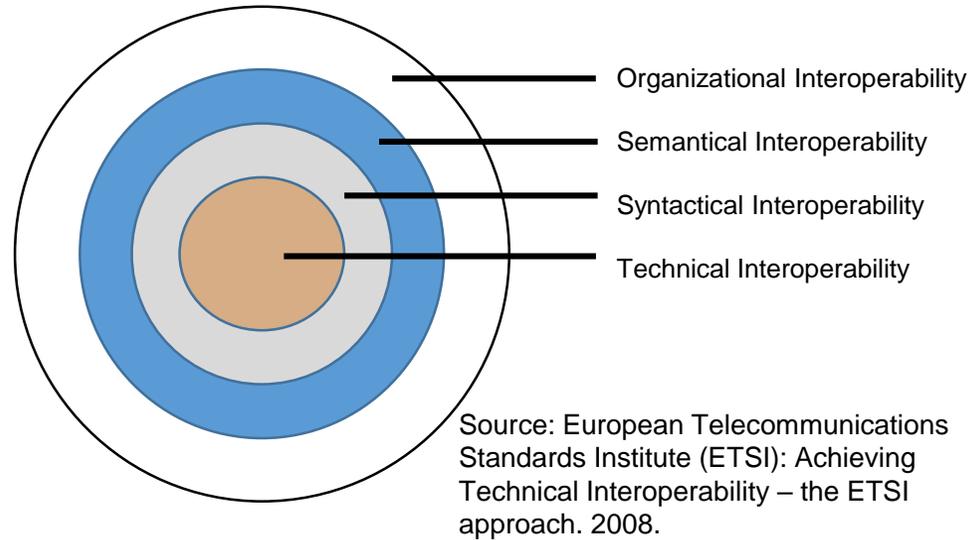
INTEGRIERTE  
AUTOMATION

# Ausgangspunkt

- Die Asset Administration Shell (AAS) ist die Spezifikation eines Digitalen Zwillings der Plattform Industrie 4.0 und der IDTA (Industrial Digital Twin Association)
- Die AAS besteht aus einem Meta-Modell und verschiedenartigen technologischen Umsetzungen
- Die AAS soll Produkt- und Produktions-lebenszyklusübergreifend zum Einsatz kommen
- Die AAS befindet sich im Prototypenstadium



# Interoperabilitätsansätze im industriellen Umfeld



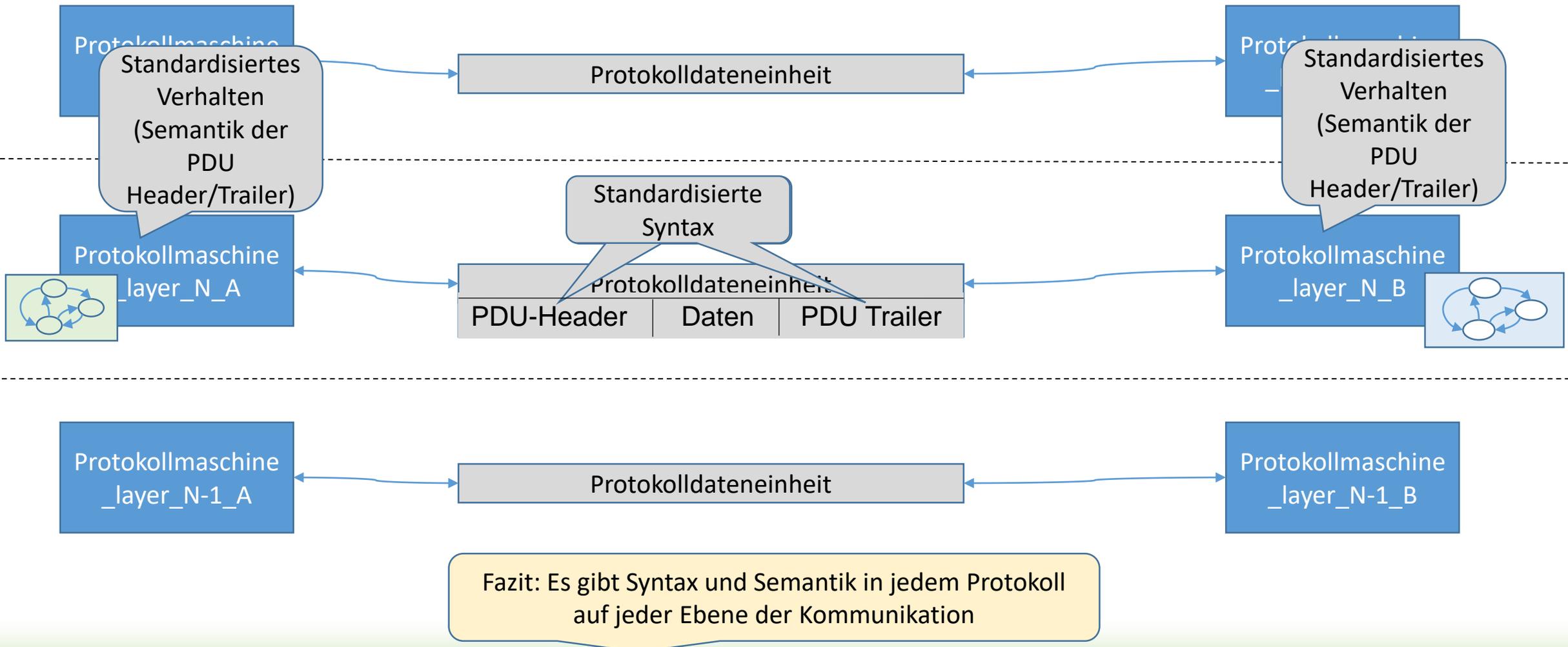
Source: SGCG Report Reference Architecture v0.5 (for SG-CG Sanity Check) 2012-01-25 or CEN-CENELEC-ETSI Smart Grid Coordination Group Smart Grid Reference Architecture. November 2012



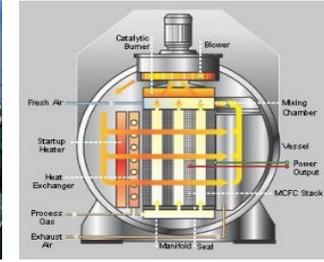
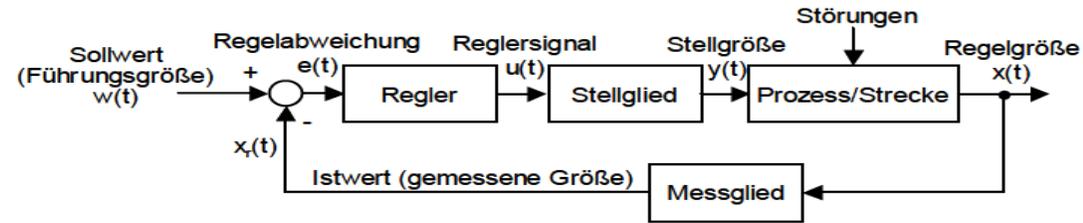
Source: Aamodt, A.; Nygard, M.: Different roles and mutual dependencies of data, information, and knowledge – An All perspective on their integration. Data and Knowledge Engineering, 16 (3), Sept. 1995, S. 191–222, 1995.

Uneinheitliche Ansätze, die durch informale Beschreibungen definiert werden

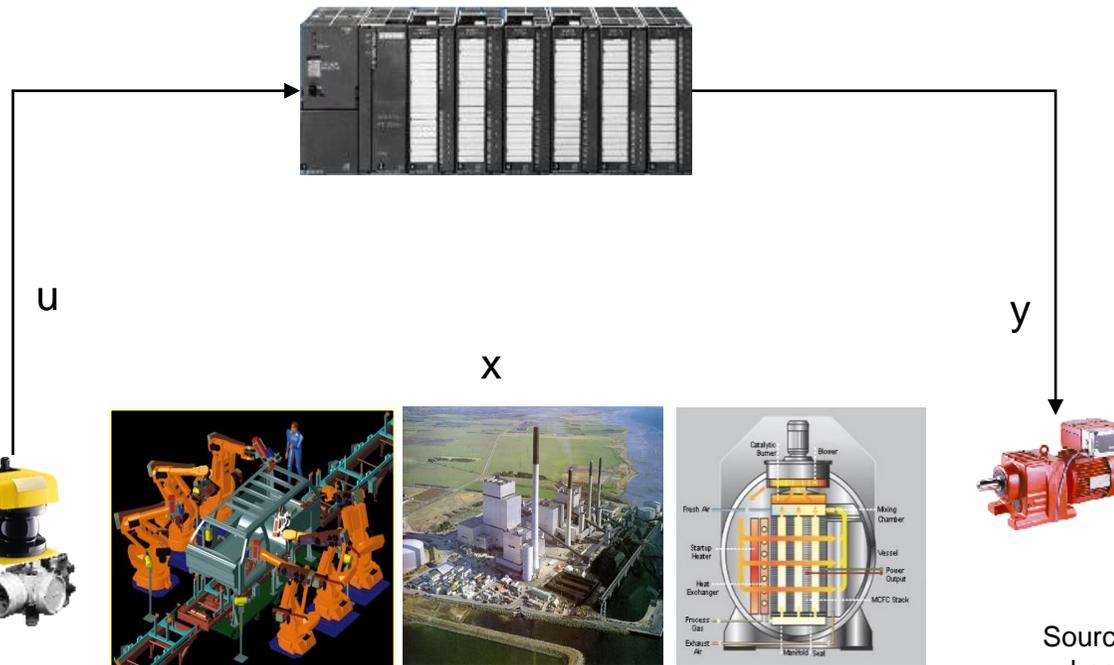
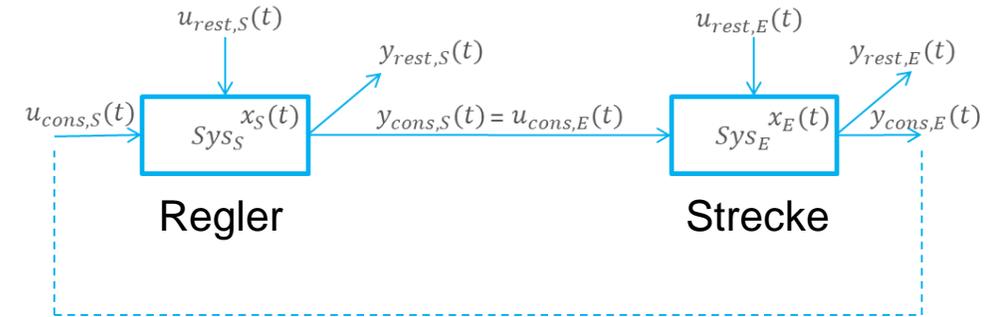
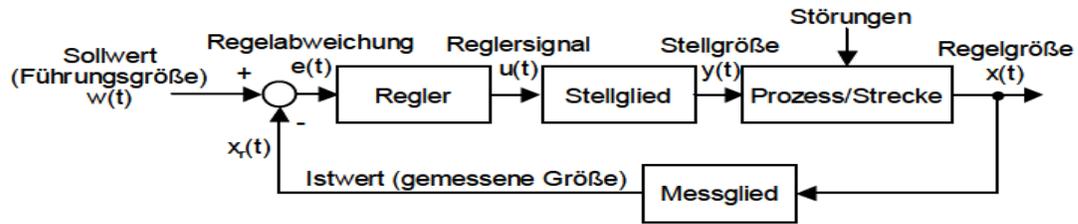
# Ergänzende Erläuterung: Wirkprinzip Kommunikationsprotokolle



# Geräteorientierte Darstellung des Regelkreises



# Systembeschreibung durch die regelungstechnischen Methoden/Systemtheorie



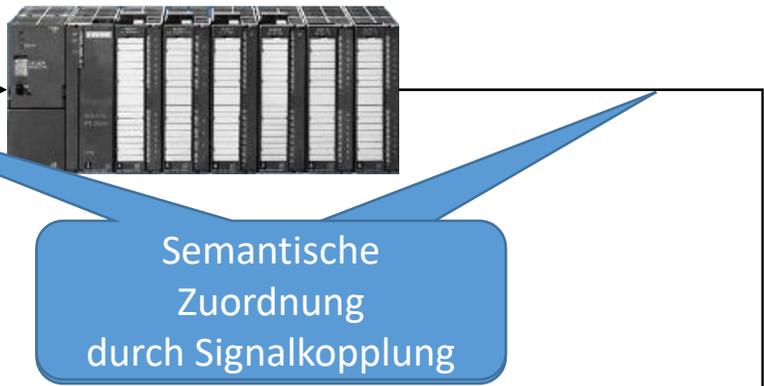
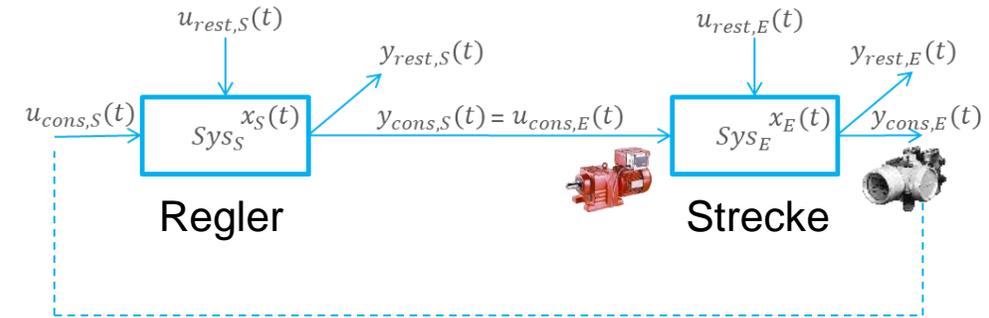
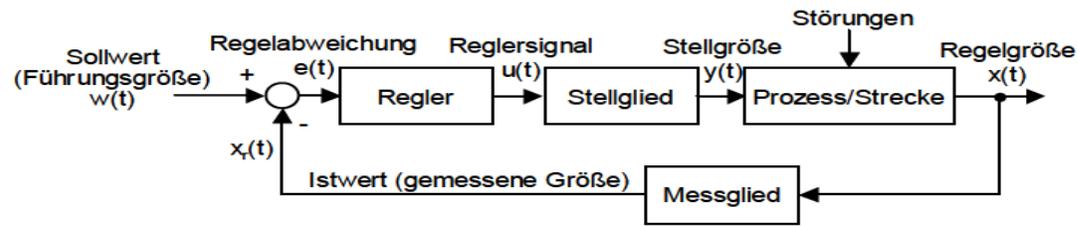
Ein beliebiges System lässt sich beschreiben durch ein Tupel  $Sys = (T, X, U, Y, x_0, f)$  mit:

- Zeitstruktur  $T = (T, succ)$  with Nachfolger-Funktion  $succ: T \rightarrow T, t' = succ(t)$
- Zustandsmenge  $X$
- Eingabealphabet (Inputs)  $U$
- Ausgabealphabet (Outputs)  $Y$
- Initialer Zustand  $x_0$
- Systemfunktion  $f = (f_{int}, f_{ext})$  mit
  - o  $f: X \times U \rightarrow X \times Y$  mit  $(x(t'), y(t')) = f(x(t), u(t))$
  - o  $f_{int}: X \times U \rightarrow X$  mit  $x(t') = f_{int}(x(t), u(t))$
  - o  $f_{ext}: X \times U \rightarrow Y$  mit  $y(t') = f_{ext}(x(t), u(t))$

Systemfunktion

Source: Tizian Schröder, Alexander Willner, Christian Diedrich: Semantische Interoperabilität in cyber-physischen Produktionssystemen. in Börteçin, Ege, Adrian Paschke Hrsg. Semantische Datenintelligenz im Einsatz. Springer Vieweg 2020. ISBN 978-3-658-31937-3.  
Schröder, T.; Diedrich, C.: Formal Definition of the Term "Semantic" as a Foundation for Semantic Interoperability in the Industrial Internet of Things. 21st IFAC World Congress, Berlin, 2020

# Semantik Definition durch systemtheoretischen Ansatz – die Bedeutung wird nicht an ein Wort gebunden, sondern an die Eingang/Funktion-Kopplung Eingang



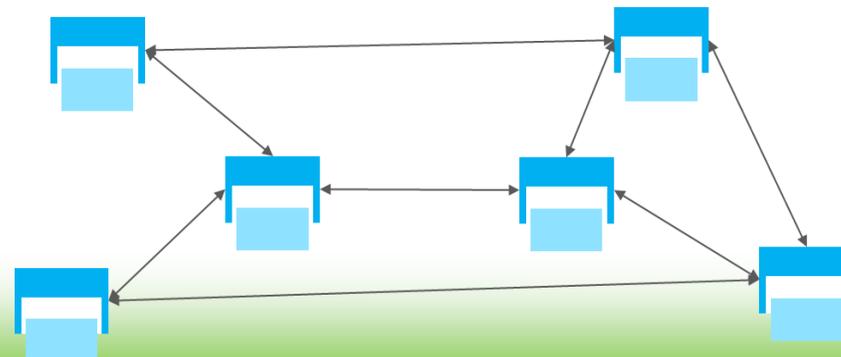
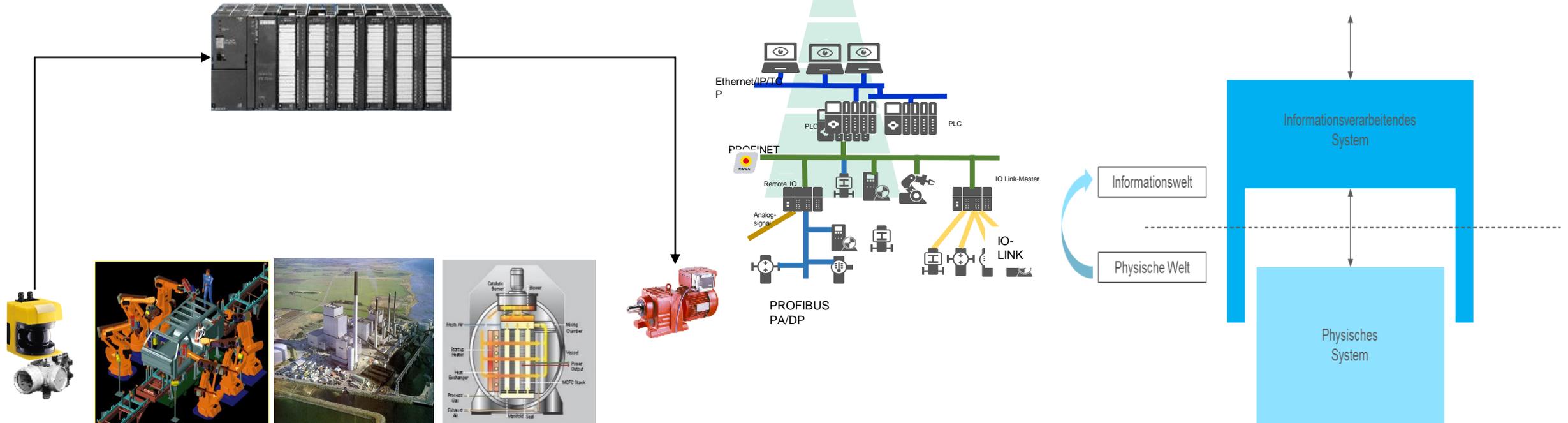
$$Sem_{Sys,x(t)}(u(t)_{cons}) = \left\{ (x(t'), y(t')) \left| \begin{array}{l} x(t') = f_{int}(x(t), (u(t)_{cons}, u(t)_{rest})), \\ y(t') = f_{ext}(x(t), (u(t)_{cons}, u(t)_{rest})), \\ u(t)_{rest} \in U \end{array} \right. \right\}$$

$$Dec_{Sys,x(t)}(u(t)_{rest}) = \left\{ (x(t'), y(t')) \left| \begin{array}{l} x(t') = f_{int}(x(t), (u(t)_{cons}, u(t)_{rest})), \\ y(t') = f_{ext}(x(t), (u(t)_{cons}, u(t)_{rest})), \\ u(t)_{cons} \in U \end{array} \right. \right\}$$

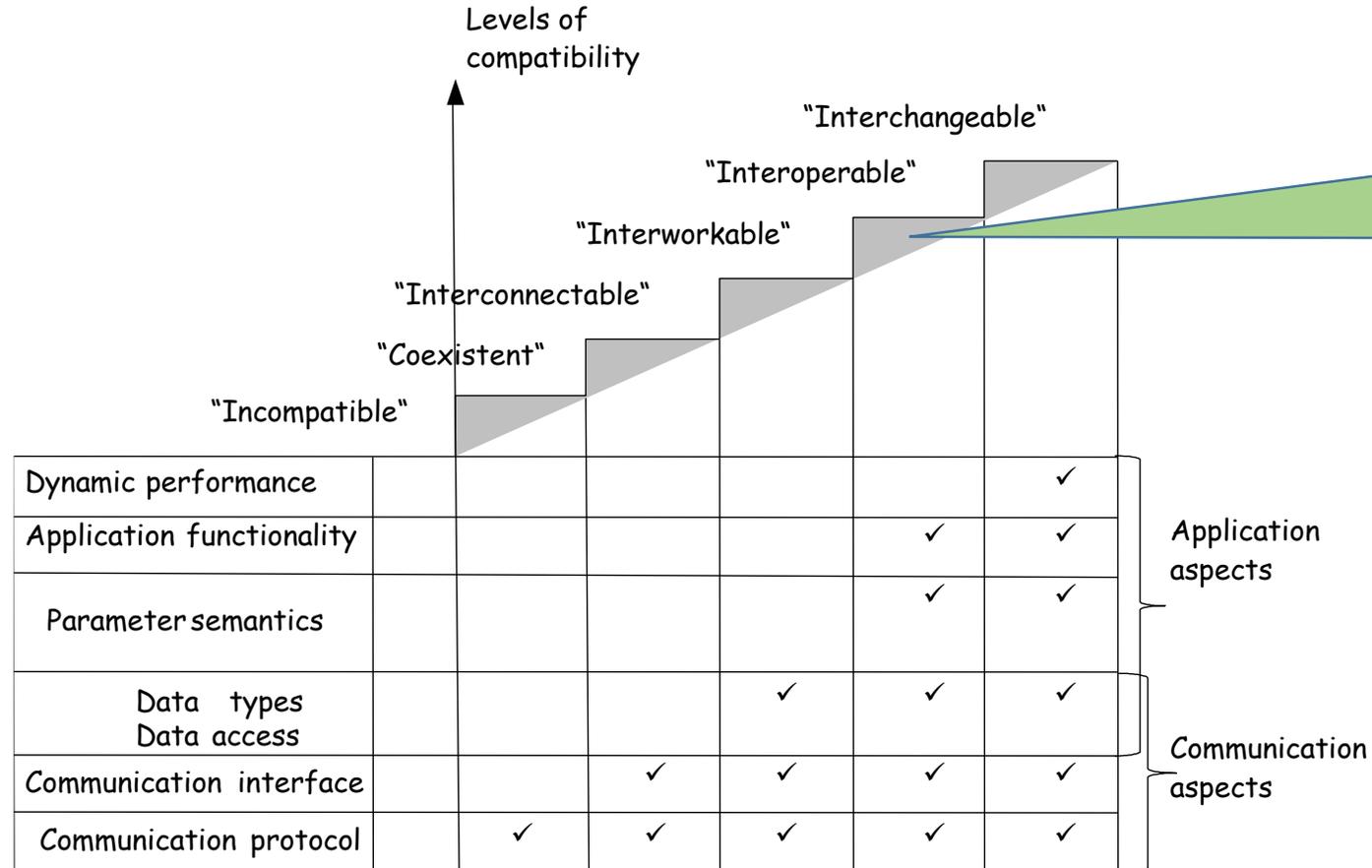


Source: Tizian Schröder, Alexander Willner, Christian Diedrich: Semantische Interoperabilität in cyber-physischen Produktionssystemen. in Börteçin, Ege, Adrian Paschke Hrsg. Semantische Datenintelligenz im Einsatz. Springer Vieweg 2020. ISBN 978-3-658-31937-3.

# Skalierung auf Produktionssysteme mit digitaler Kommunikation

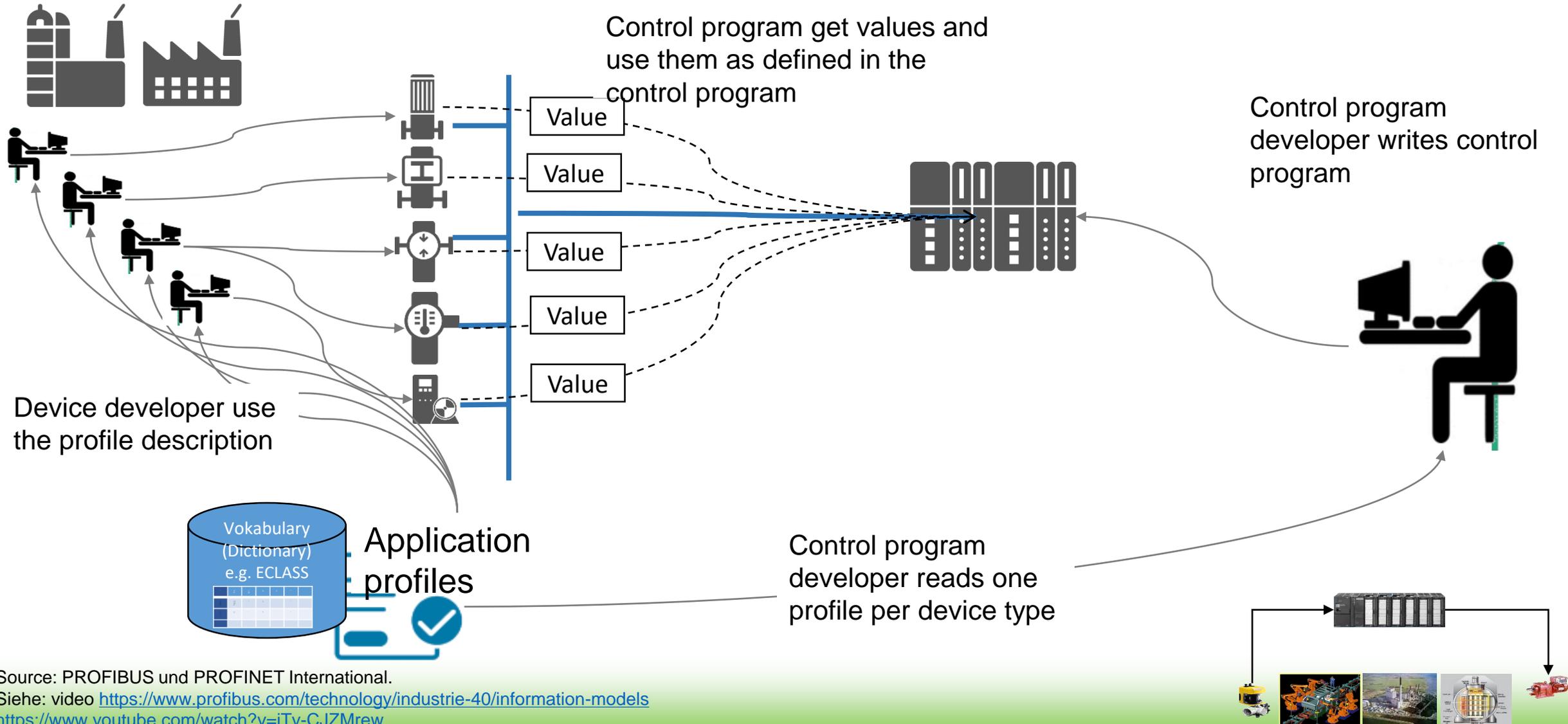


# Weiterer Interoperabilitätsansatz



Anwendungsbezug für die Interoperabilität

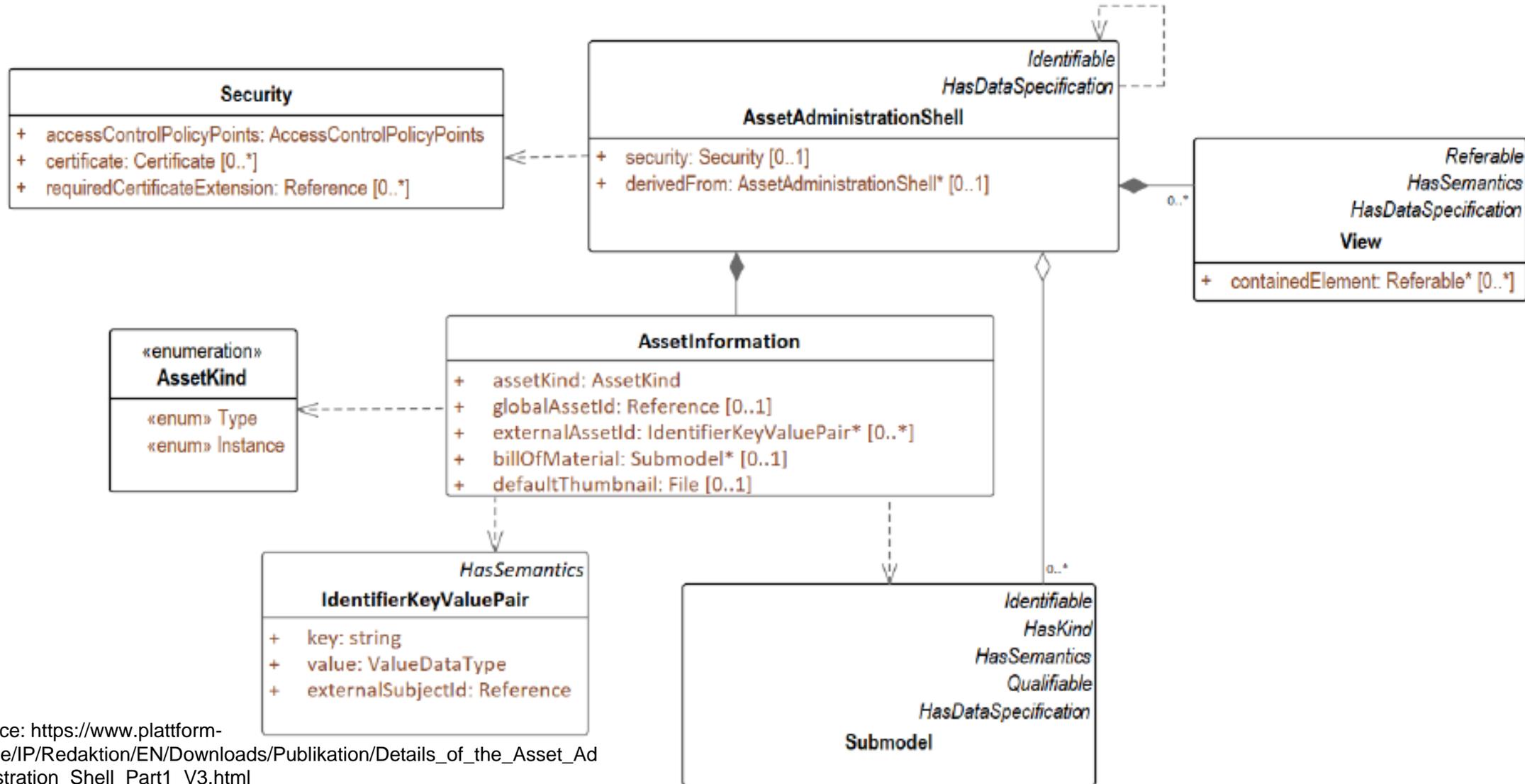
# Die Interoperabilität zwischen Feldgeräten und Steuerung ist geklärt



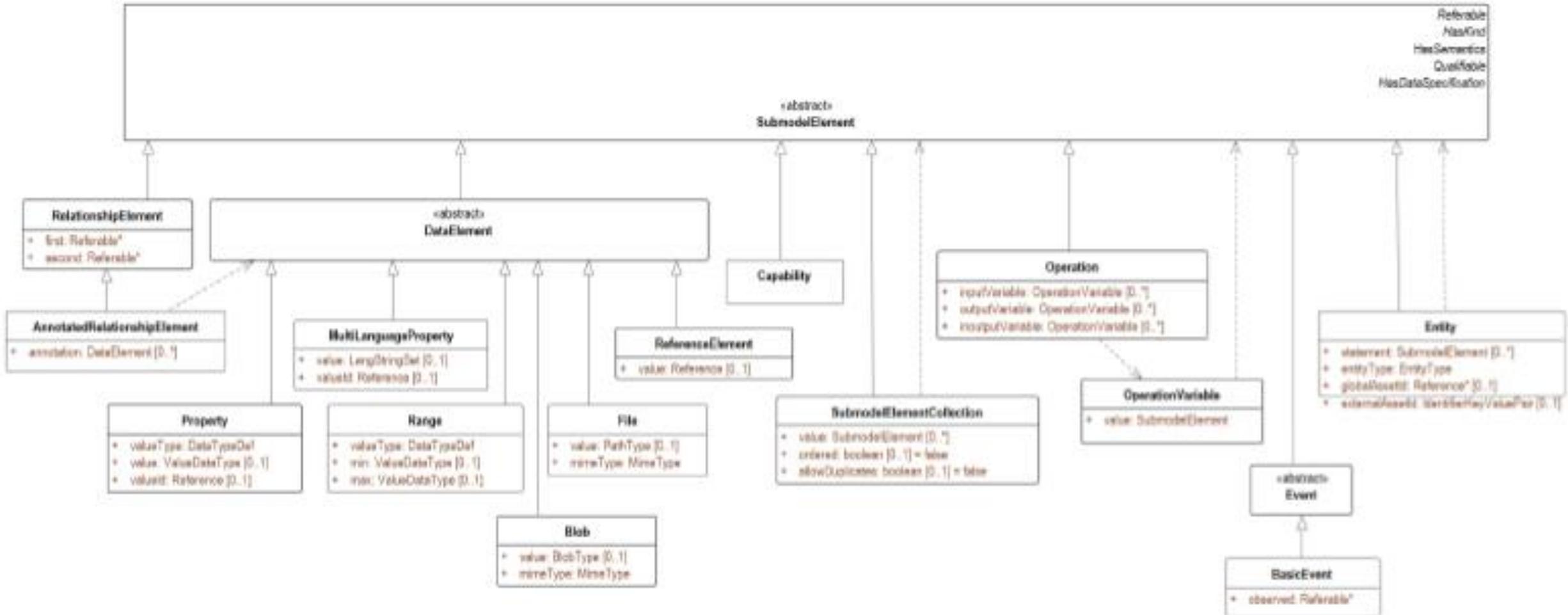
## Zwischenfazit

- Interoperabilität auf der Ebene der Anwendungen basiert auf der Semantikdefinition
  - Wir verstehen darunter, dass die auszutauschenden Daten die Ein- und Ausgangsvariablen der Anwendungsfunktionen sind
  - Das gilt für jede einzelne Funktion, d.h. zwischen zwei oder mehreren Systemen können sowohl semantisch interoperable als auch inkompatible Teile enthalten sein
- Deshalb sprechen wir von der semantischen Interoperabilität
  - D.h. erst wenn die richtigen Variablen an die dafür vorgesehenen Funktionen angebunden werden ist die semantische Interoperabilität gewährleistet, die Einigung auf einen Begriff ist oft nicht ausreichend
- Semantische Interoperabilität in der Feldebene ist bereits vorhanden (wird durch Profile und den Entwicklungsprozesse gelöst)

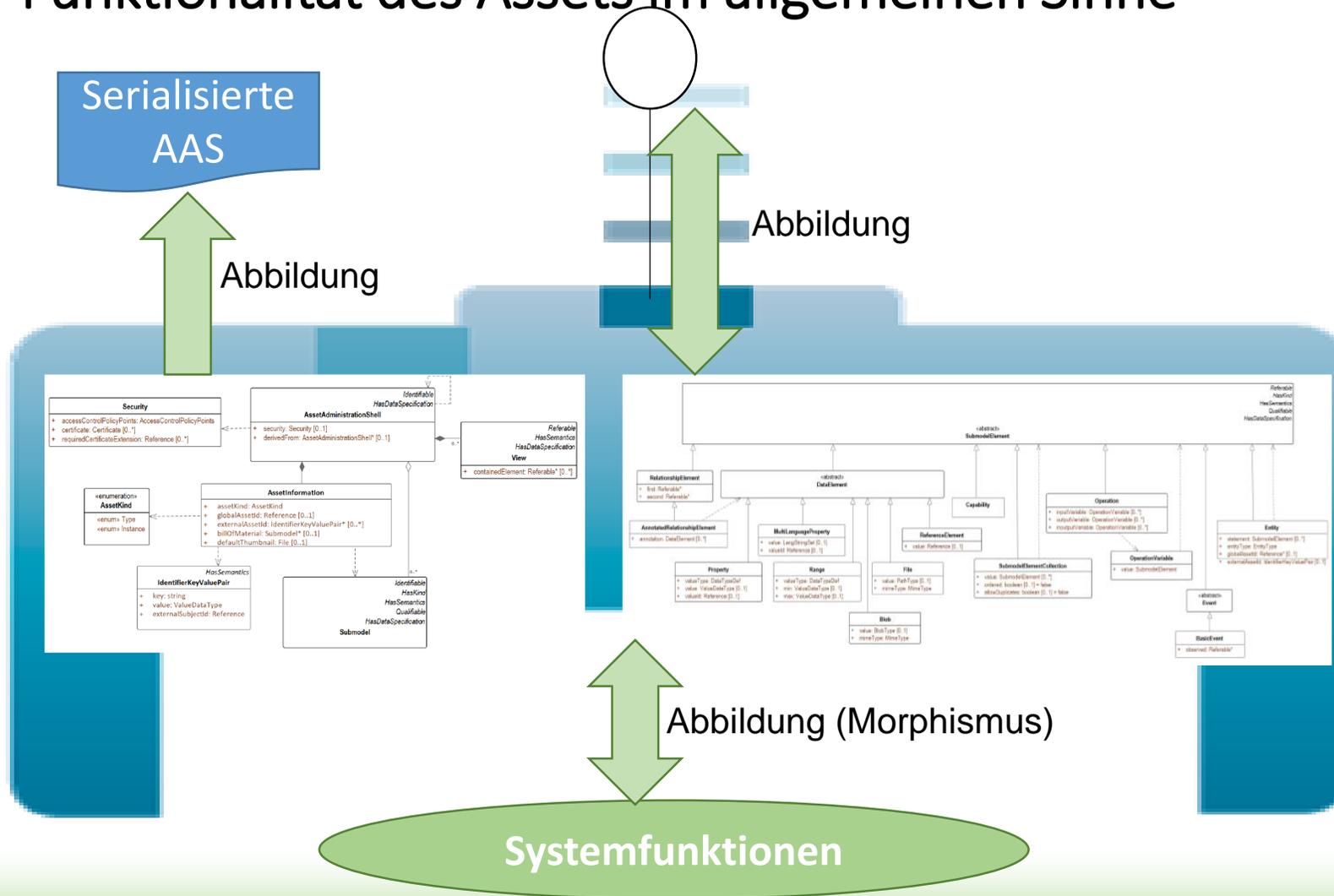
# Meta-Modell des Digitalen Zwillings (Asset Administration Shell - AAS) – Top Abstraktionsebene



# Meta-Modell des Digitalen Zwillings (Asset Administration Shell - AAS) – Elemente der Submodels

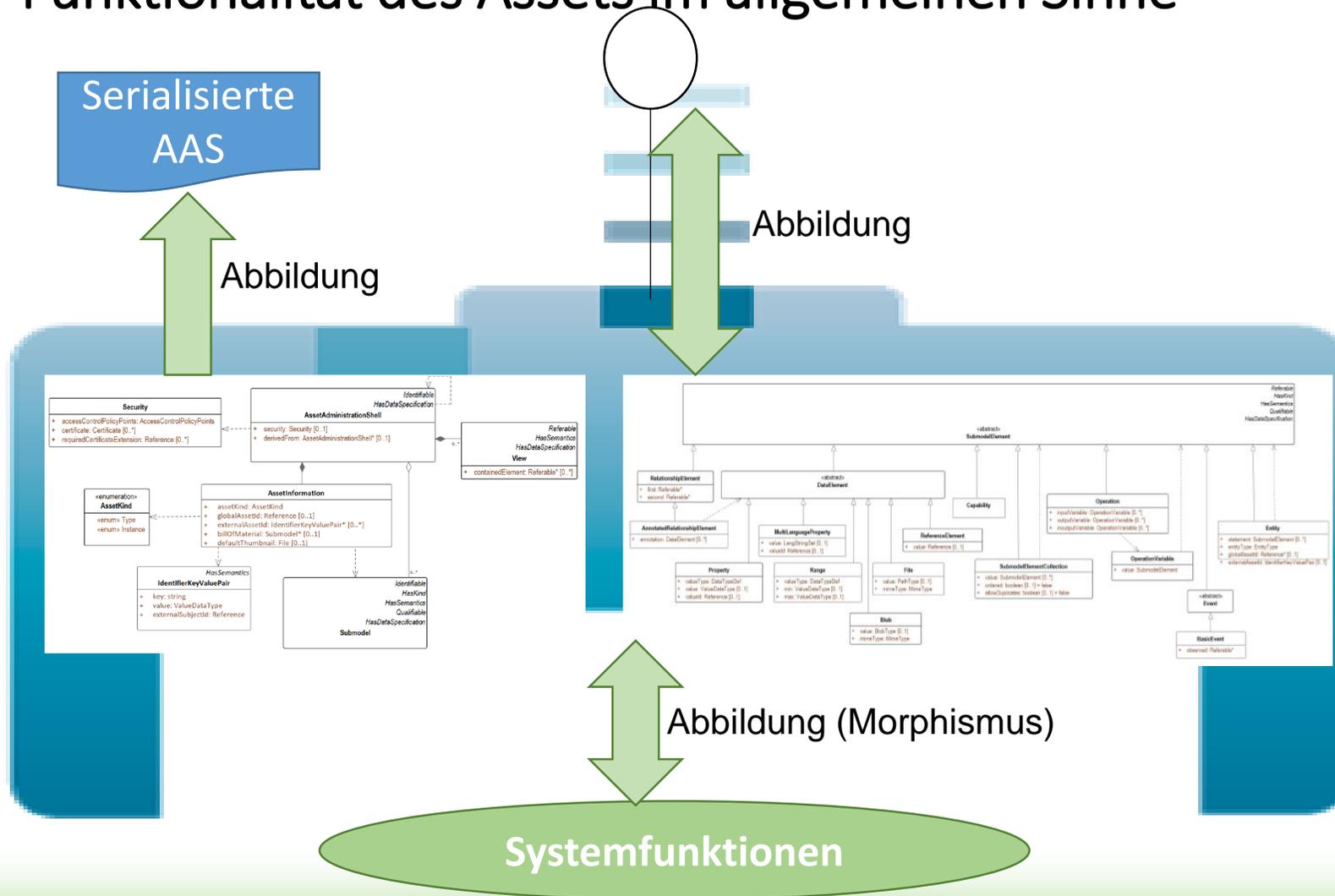


# Das Meta-Modell der Verwaltungsschale ist ein Beschreibungsmodell der Funktionalität des Assets im allgemeinen Sinne



- Der digitale Zwilling/AAS ist eine abstrakte Abbildung des Assets (f,x,y,u → AAS)
- Diese Abbildung ist ein manuell erstellter Morphismus
- Das AAS – Modell wird an Schnittstellen oder als Datei zugreifbar gemacht.
- Aus Sicht der semantischen Interoperabilität:
  - Wie korrekt ist die Abbildung der Systemfunktionen auf das AAS-Modell?
  - Wie korrekt und konsistent sind die Abbildungen des AAS Modells auf die Schnittstellen und die Serialisierung?
  - Sind die Systemfunktionen überhaupt bekannt?

# Das Meta-Modell der Verwaltungsschale ist ein Beschreibungsmodell der Funktionalität des Assets im allgemeinen Sinne



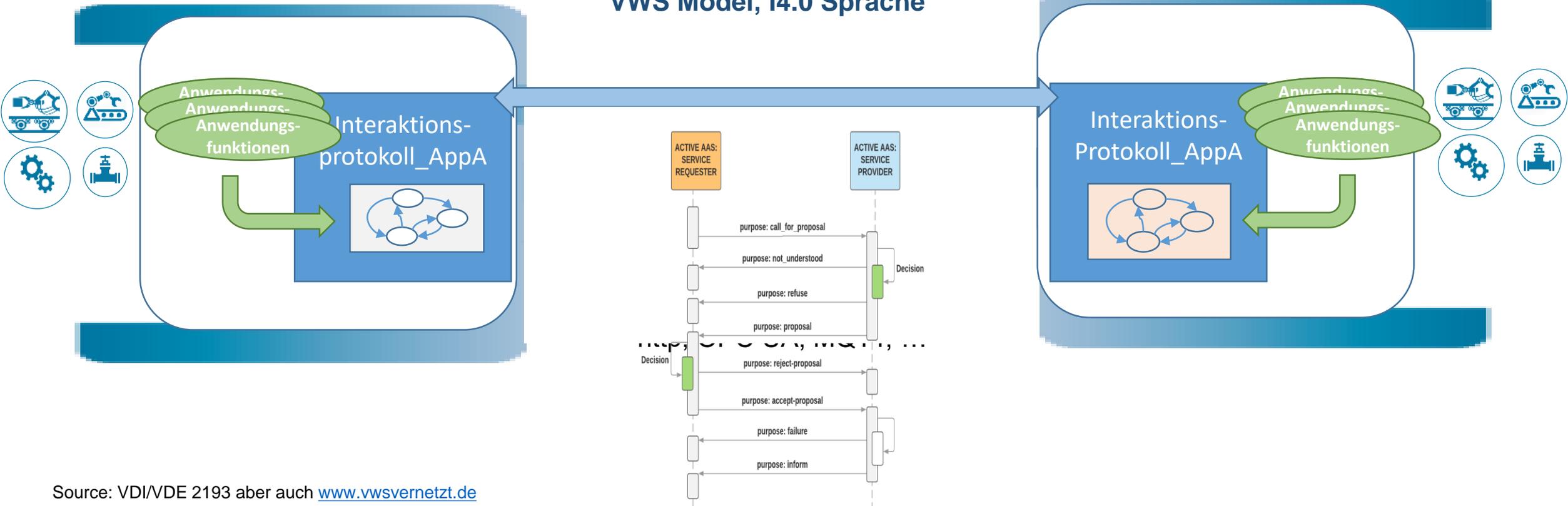
- Leitlinie der Abbildung ist die Beschreibung der Asset/Systemcharakteristika durch Merkmale
- Dahinter steckt die Annahme, dass die Merkmale für die Semantische Interoperabilität der Assets ausreichend sind

# Zu einigen Details

- Meta-Modell
  - Die Modellelemente und ihre Beziehungen einschließlich Kardinalitäten sind semi-formal (Klassendiagramm ++)
- Modellelement: Operation **Aus Sicht des Lehrstuhls fehlt hier das semantische Protokoll**
  - Ist eine Anwendungsfunktion im allgemeinen Sinne (Eingang und Ausgang können alle Submodellelementtypen sein (z.B. BLOB))
- Identifikation der Modellelemente
  - Globale und lokale Identifikatoren
  - Referenzen zu Dictionaries
- Beziehungen zwischen Modellelemente (nicht der Meta-Modellelementen sondern der Elementinstanzen)
  - Referenzen
  - Relationship → können Typen sein, die durch Identifikatoren im Sinne eines Metamodells eindeutig sind (z.B. bestehtAus)
- Serialisierungen
  - AAS-XML
  - AAS-JSON
  - AutomationML
  - rdf
- Schnittstellen
  - http/Rest
  - OPC UA
- Infrastruktur
  - Registry
  - Discovery

# 14.0 Sprache und semantisches Protokoll

## Syntaktische und semantische Interoperabilität VWS Model, I4.0 Sprache



Source: VDI/VDE 2193 aber auch [www.vwsvernetzt.de](http://www.vwsvernetzt.de)

# Zu einigen Details

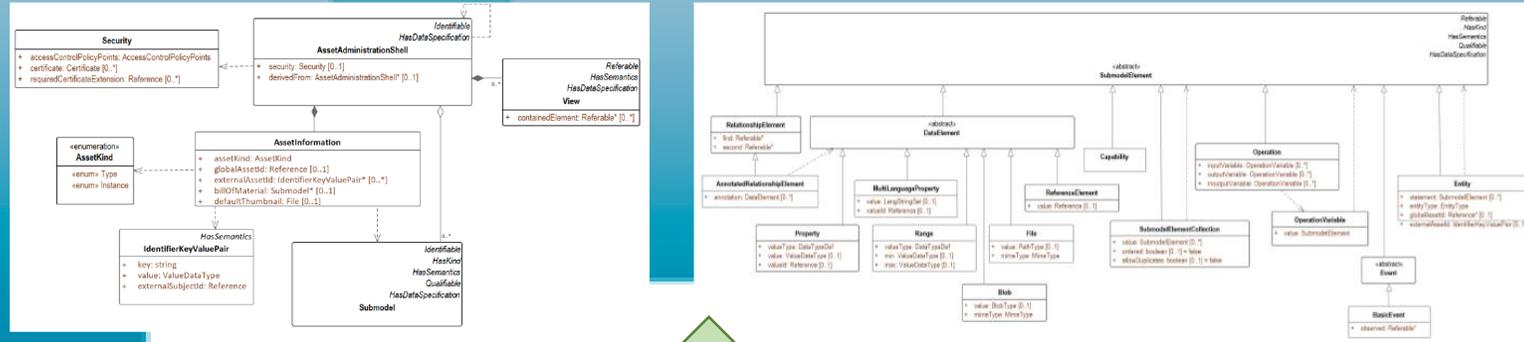
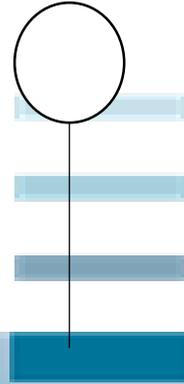
- Meta-Modell
  - Die Modellelemente und ihre Beziehungen einschließlich Kardinalitäten sind semi-formal (Klassendiagramm ++) beschrieben
- Modellelement: Operation
  - Ist eine Anwendungsfunktion im allgemeinen Sinne (Eingang und Ausgang können alle Submodellelementtypen sein (z.B. BLOB))
- Identifikation der Modellelemente
  - Globale und lokale Identifikatoren
  - Referenzen zu Dictionaries
- Beziehungen zwischen Modellelemente (nicht der Meta-Modellelementen sondern der Elementinstanzen)
  - Referenzen
  - Relationship → können Typen sein, die durch Identifikatoren im Sinne eines Metamodells eindeutig sind (z.B. bestehtAus)
- Serialisierungen
  - AAS-XML
  - AAS-JSON
  - AutomationML
  - rdf
- Schnittstellen
  - http/Rest
  - OPC UA
- Infrastruktur
  - Registry
  - Discovery

Projekt SemAnz40 erprobt

OPC UA Erweiterung um Dictionary Reference

# Semantische Referenz auf standardisierte Vokabulare (Ontologien)

Serialisierte  
AAS



Jedes  
Modell-  
element

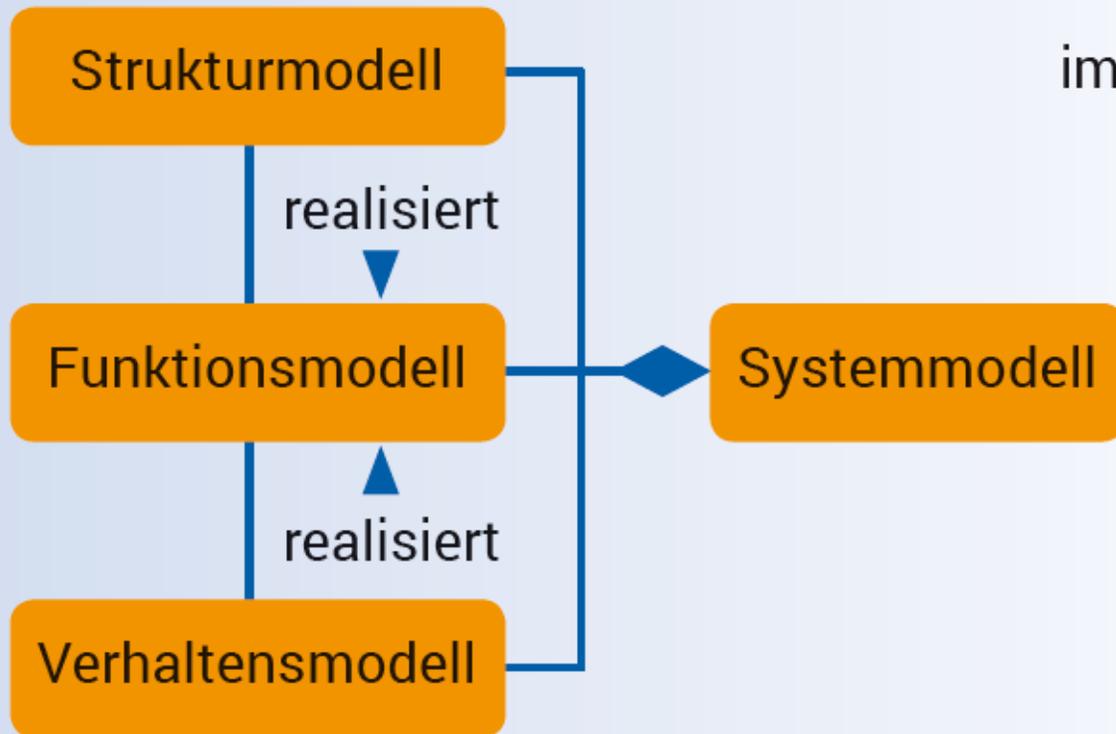
Dictionary  
z.B.  
ECLASS,  
IEC CDD,  
andere  
Vokabulare  
(Feldbusprofile,  
OPC UA CS)

Abbildung (Morphismus)

Systemfunktionen

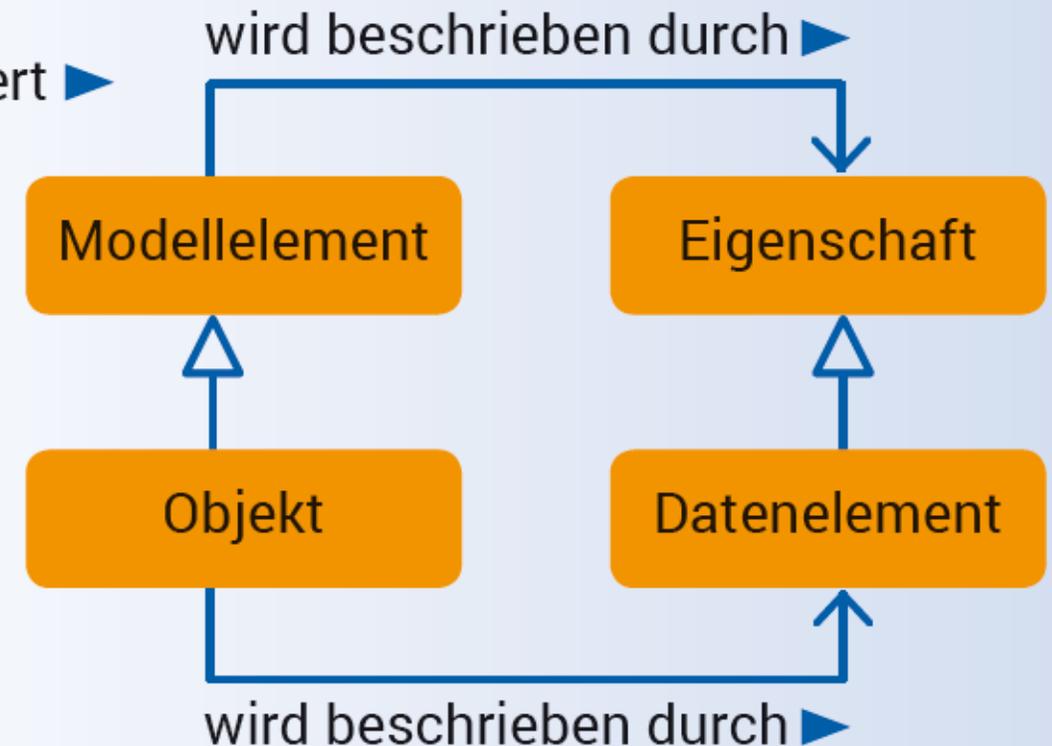


## Class Systemmodell

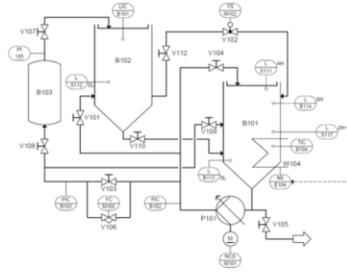


Source: <https://www.eclass.eu/verein/projekt-semanz40.html>

## Class Eigenschaftsmodell



## ► SemAnz4.0 AML-Schnittstelle Mapping



Produktdaten:  
• Stoffbezeichnung

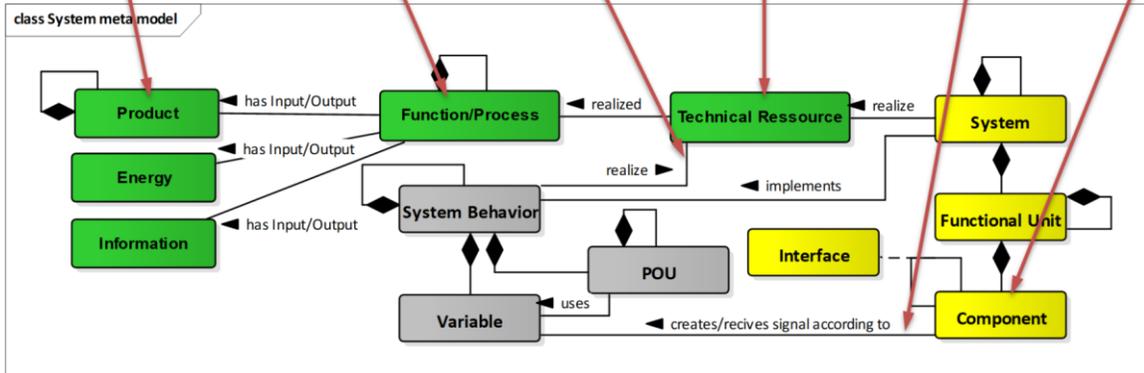
PLT-Stellendaten:  
• PCEDescription

SW-Signaldaten:  
• FBDNumber  
• SWSignalName  
• FBName

Daten der BML:  
• Funktionskennzeichnung

HW-Signaldaten:  
• SafetyRelevant  
• ExplosionProtection  
• Cabinet  
• MainController

Daten der GML:  
• Typ Eingang

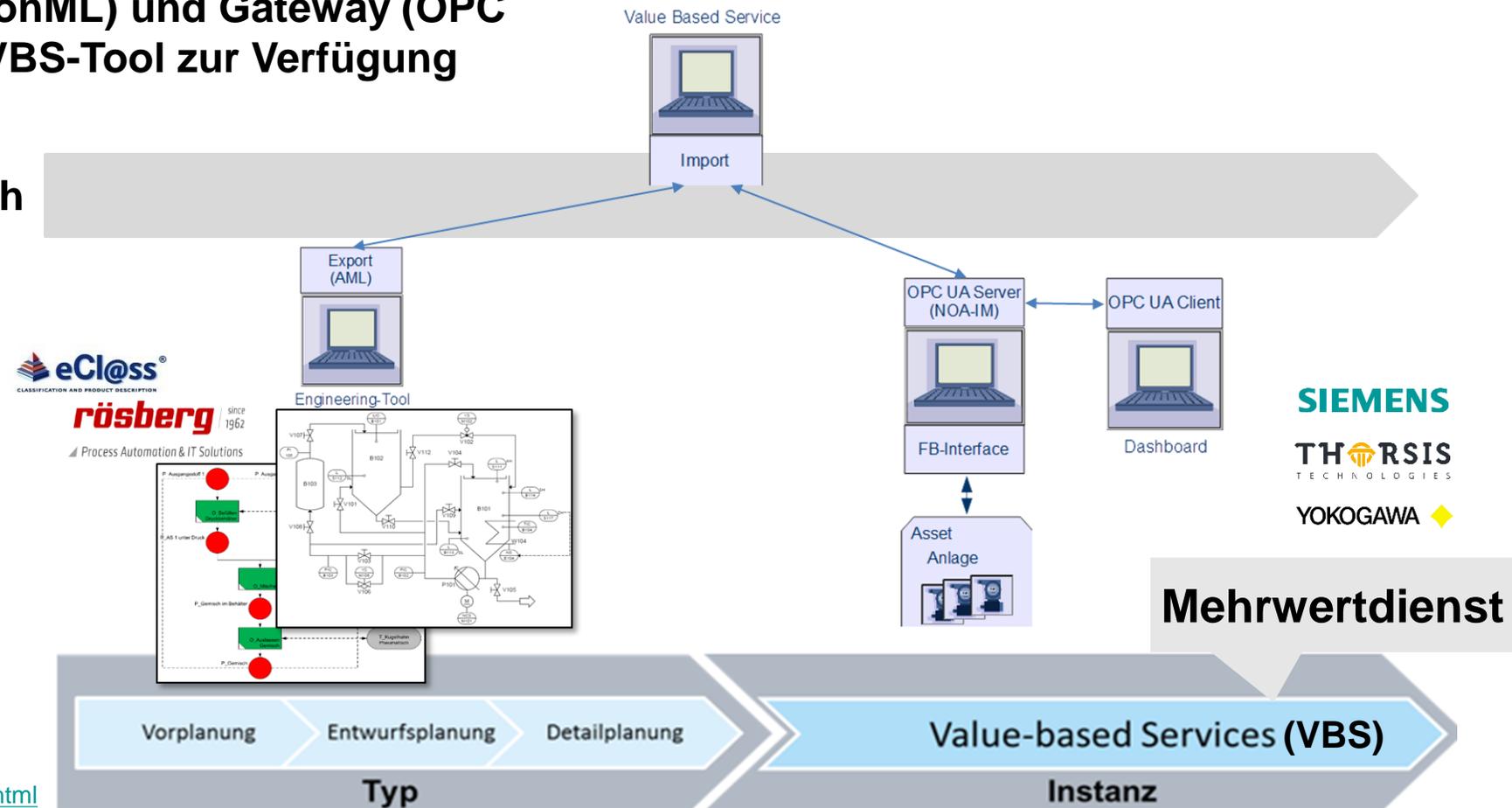


Source: <https://www.eclass.eu/verein/projekt-semanz40.html>

# Value-based Services: Plant Asset Management - Demonstrator

## Use Case: VBS-Tool “as planned/as built – Comparison”

- > Planungsdaten (Typ) und Gerätedaten (Instanz) werden von Engineering-Tool (AutomationML) und Gateway (OPC UA-Informationsmodell) einem VBS-Tool zur Verfügung gestellt
- > Zusammenführung und Vergleich von Planungsdaten und Gerätedaten in VBS-Tool als Value Based Service
  - > Fehlerfreies, effizienteres Engineering
- > Zuordnung durch semantische Annotation mit IDs

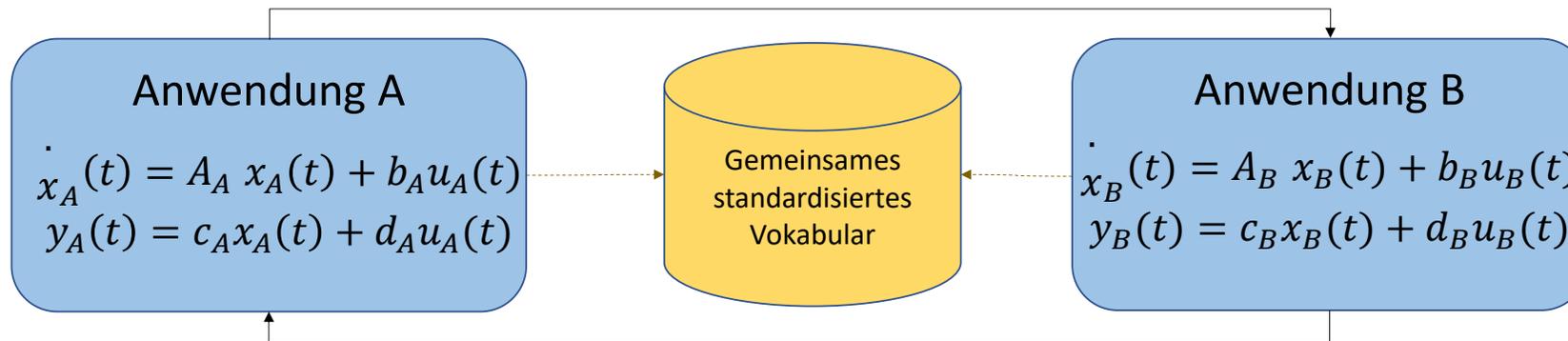


Source: <https://www.eclass.eu/verein/projekt-semanz40.html>

# Standardisierte Vokabular zur Ein- /Ausgangszuordnung der Anwendungsfunktion

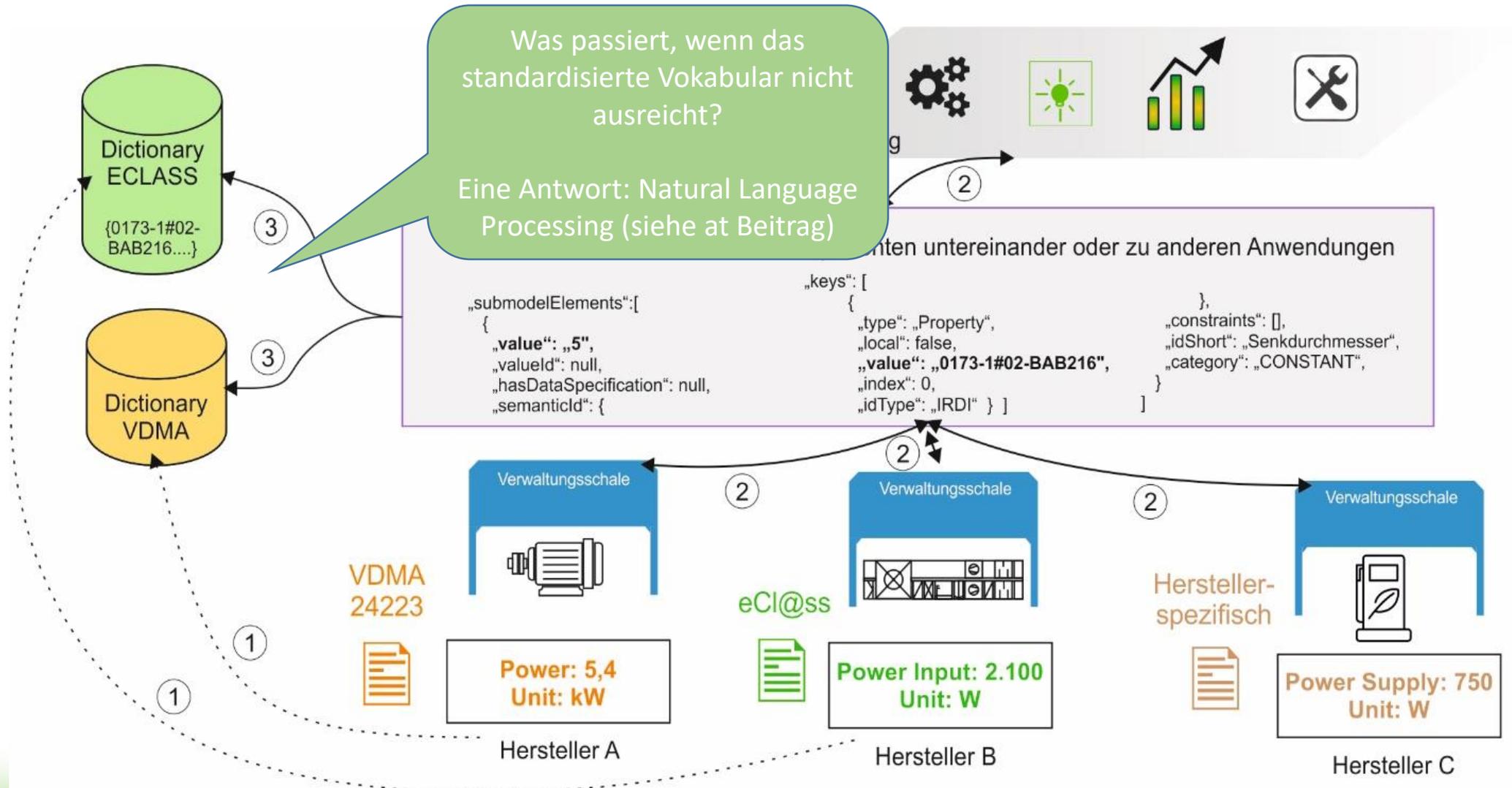
Zuordnung der Transferein- und -ausgänge

$$u_B(t) = y_A(t)$$



$$u_A(t) = y_B(t)$$

# Prinzip des Austausches mit standardisierten Vokabularen



# Zu einigen Details

- Meta-Modell
  - Die Modellelemente und ihre Beziehungen einschließlich Kardinalitäten sind semi-formal (Klassendiagramm ++) beschrieben
- Modellelement: Operation
  - Ist eine Anwendungsfunktion im allgemeinen Sinne (Eingang und Ausgang können alle Submodellelementtypen sein (z.B. BLOB))
- Identifikation der Modellelemente
  - Globale und lokale Identifikatoren
  - Referenzen zu Dictionaries
- Beziehungen zwischen Modellelemente (nicht der Meta-Modellelementen sondern der Elementinstanzen)
  - Referenzen
  - Relationship → können Typen sein, die durch Identifikatoren im Sinne eines Metamodells eindeutig sind (z.B. bestehtAus)
- Serialisierungen
  - AAS-XML
  - AAS-JSON
  - AutomationML
  - rdf
- Schnittstellen
  - http/Rest
  - OPC UA
- Infrastruktur
  - Registry
  - Discovery

# Beispiel semantische ID für ein Submodel „Bill of Material“



The screenshot shows the AASX Package Explorer on the left and the graph display on the right. The Package Explorer shows a tree structure under 'Env "AdministrationShells"' with a submodel 'BOM' selected. The graph display shows a hierarchical structure of entities and relationships.

**Package Explorer:**

- Env "AdministrationShells"
  - AAS "AASPickAndPlaceStation4AssetInstance" [IRI, www.ovgu.de/ifat/lia/ids/VWS, ...]
    - Sub "Nameplate" [IRI, www.company.com/ids/sm/0462\_7060\_8002\_2512]
    - Sub "Document" [IRI, www.company.com/ids/sm/2543\_5072\_7091\_2660]
    - Sub "CompositeAASRelationship" [IRI, www.company.com/ids/sm/3015\_3112\_...
    - Sub "BOM" [IRI, www.company.com/ids/sm/0543\_7062\_9002\_7533]
      - BOM Bill of Material - Graph display ready
        - Ent "PickAndPlaceStationOvGU"
        - Ent "DeltaRobot360"
        - Ent "WorkstationDrilling\_1"
        - Ent "WorkstationDrilling\_2"
        - Ent "Motor\_1"
        - Ent "Motor\_2"
        - Ent "Motor\_3"
        - Ent "MotionDrive\_1"
        - Ent "MotionDrive\_2"
        - Ent "MotionDrive\_3"
        - Ent "DriveMotorKabel\_1"
        - Ent "RobotFrame"

**Graph Display:**

The graph shows a central entity 'DeltaRobot360' (self-managed) connected to 'WorkstationDrilling\_1' and 'WorkstationDrilling\_2' (co-managed). 'DeltaRobot360' is also connected to 'MotionDrive\_1', 'MotionDrive\_2', and 'MotionDrive\_3' (co-managed). 'MotionDrive\_1' is connected to 'Motor\_1', 'Motor\_2', and 'Motor\_3' (co-managed). 'MotionDrive\_2' is connected to 'DriveMotorKabel\_1' (co-managed). 'MotionDrive\_3' is connected to 'Schraube' (asset). 'DeltaRobot360' is also connected to 'RobotFrame' (asset). The legend indicates that blue boxes represent self-managed entities, light blue boxes represent co-managed entities, dark blue boxes represent properties, and grey boxes represent assets. Lines represent relations, and double-headed lines represent property-to-entity relations.

- Semantische ID besagt, dass dieses Submodel eine Stückliste ist
- Die Beziehungen der Elemente in der Stückliste sind Relationships, die auch eine semantische ID haben, z.B. isPartOf
- Wenn es eine Abbildung der semantischen ID des Submodels auf die entsprechende Zeichenfunktion gibt, kann die Datenstruktur interpretiert werden und ein Bild entsteht

# Zu einigen Details

- Meta-Modell
  - Die Modellelemente und ihre Beziehungen einschließlich Kardinalitäten sind semi-formal (Klassendiagramm ++) beschrieben
- Modellelement: Operation
  - Ist eine Anwendungsfunktion im allgemeinen Sinne (Eingang und Ausgang können alle Submodellelementtypen sein (z.B. BLOB))
- Identifikation der Modellelemente
  - Globale und lokale Identifikatoren
  - Referenzen zu Dictionaries
- Beziehungen zwischen Modellelemente (nicht der Meta-Modellelementen sondern der Elementinstanzen)
  - Referenzen
  - Relationship → können Typen sein, die durch Identifikatoren im Sinne eines Metamodells eindeutig sind (z.B. bestehtAus)
- Serialisierungen
  - AAS-XML
  - AAS-JSON
  - AutomationML
  - rdf
- Schnittstellen
  - http/Rest
  - OPC UA
- Infrastruktur
  - Registry
  - Discovery

# Aktueller pragmatischer Ansatz: Matrix für Definition und Abgleich verschiedener Interoperabilitätskriterien

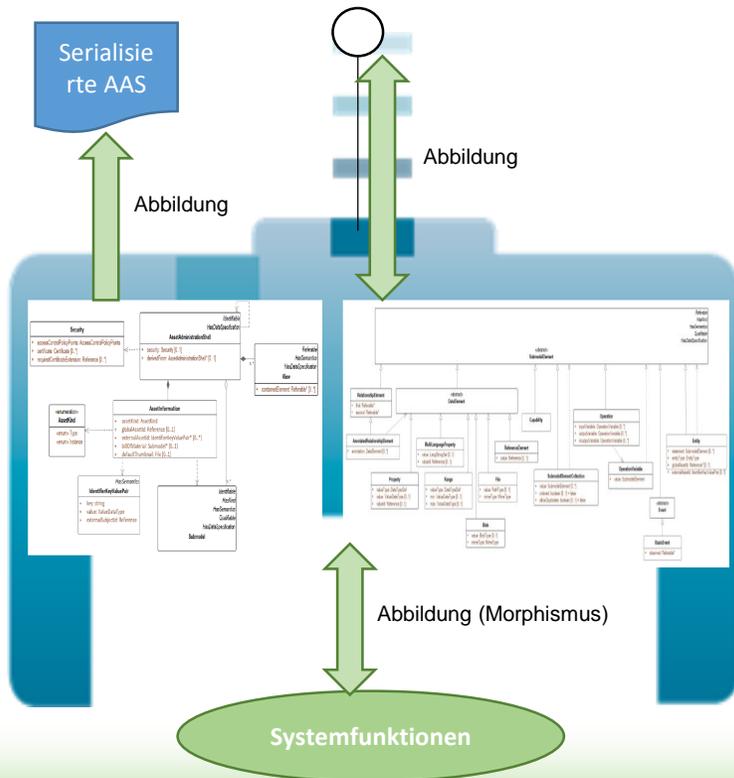
		I4.0-Sprache									
		Technische Interoperabilität			Syntaktische Interoperabilität		Semantische Interoperabilität				Organisatorische Interoperabilität
Type of AAS	Communication protocol	Registry/Discovery	Security (Encryption, Authentication...according AG3?)	Syntax der Nachricht	Syntax der Inhalte der Nachricht	Struktur der Nachrichten	Datenmodell der Merkmale	Semantische Identifikatoren für Merkmale	Teilmodelle Nach AAS-Metamodell	Teilmodelle nach standard. Template	Interaktionsmuster
Spezifikation	...	...	...	VDI/VDE-2193 1 (z.B. JSON-Serialisierung)	AAS-Metamodell Teil 1 (z.B. JSON-Serialisierung)	VDI/VDE-2193-1	IEC 61360, spezifische	AAS-Metamodell Teil 1, ECLASS	AAS-Metamodell Teil 1, ECLASS	Z.B. Nameplate, Technical Data, Template für Capability usw (z.B. von ZVEI, IDTA usw)	VDI/VDE-2193-2 und potentielle weitere Teile
Proaktive AAS	MQTT	Ohne Registry/Discovery Server	Ohne Security								
			Mit Security								
	MQTT	Mit Registry/Discovery Server	Ohne Security								
			Mit Security								

VWS sind interoperabel, wenn sie ein gleiches Interoperabilitätsprofil unterstützen

<-- Relevante Spezifikationen für VWS (beispielhaft)

<-- Beispielafter Interoperabilitätsprofil für VWS

# Wie ist nun der Stand einzuschätzen?



- Der digitale Zwilling/AAS ist eine abstrakte Abbildung des Assets ( $f, x, y, u \rightarrow \text{AAS}$ )
- Diese Abbildung ist ein manuell erstellter Morphismus
- Das AAS – Modell wird an Schnittstellen oder als Datei zugreifbar gemacht.
- Aus Sicht der semantischen Interoperabilität:
  - Wie korrekt ist die Abbildung der Systemfunktionen auf das AAS-Modell?
  - Wie korrekt und konsistent sind die Abbildungen des AAS Modells auf die Schnittstellen und die Serialisierung?
  - Sind die Systemfunktionen überhaupt bekannt?

Vielfältigste Anwendungen  
alternative Abbildungsmöglichkeiten ☹️

Technologische Anbindung gelöst ☹️

Ansatz: Best Practice  
Submodel Tempates ☺️

Modellvalidierung ☹️

In Simulationen ⚠️