

Inhaltsverzeichnis

Vorwort	5
Die Herausgeberin	6
1 Keine Industrie 4.0 ohne OPC UA	15
(STEFAN HOPPE)	
1.1 OPC UA ist kein Protokoll – sondern viel mehr	15
1.2 Migrationsschritte zur Einführung von OPC UA	16
1.3 Das war's? Daten- und Interface-Standards für Branchen	16
1.4 Differenzierung!	17
1.5 Ausblick	17
1.5.1 Trend: SoA und PubSub	17
1.5.2 Trend: «Service to Service»	18
1.5.3 Trend: OPC UA im Chip	18
1.5.4 Zukunft: OPC UA via TSN	18
1.5.5 Cross-Domain-Harmonisierung	18
2 Grundlagen	21
2.1 Architekturen im Einsatz	21
(Dr. MIRIAM SCHLEIPEN)	
2.1.1 Fall 1: 1 OPC UA Server je Asset	23
2.1.2 Fall 2: 1 externer Software OPC UA Server für alle Assets	24
2.1.3 Fall 3: 1 externer Middleware OPC UA Server / externes Gateway für alle Assets	24
2.1.4 Fall 4: Externer aggregierender Software OPC UA Server, der die OPC UA Server der einzelnen Assets aggregiert	26
2.1.5 Fall 5: Externer aggregierender Software OPC UA Server (in anderem Netz), der die OPC UA Server der einzelnen Assets aggregiert	27
2.1.6 Fall 6: Externer aggregierender Software OPC UA Server (in externem Netz), der mehrere externe aggregierende Software OPC UA Server (in anderem Netz) aggregiert	27
2.1.7 Fall 7: Externer aggregierender Software OPC UA Server (in externem Netz), der die OPC UA Server der einzelnen Assets aggregiert	28
2.2 OPC UA Companion Specifications	29
(Dr. MIRIAM SCHLEIPEN)	
2.3 Best practices OPC UA Security	32
(Dr. CHRISTIAN HAAS / Dr. SÖREN FINSTER)	
2.3.1 Grundlagen IT-Sicherheit	32
2.3.2 Sicherheitsmechanismen in OPC UA	33
2.3.3 Best practices und bekannte Herausforderungen	35
<i>Management der Vertrauensbeziehungen</i>	35
<i>Schutz des privaten Schlüssels eines digitalen Zertifikats</i>	37
2.4 Profile, Konformitätstest, Zertifizierung	38
(REINHOLD DIX)	
2.4.1 Profile	38

2.4.2	Werkzeug zum Konformitätstest (CTT – Unified Architecture Compliance Test Tool)	43
	<i>Servertest</i>	44
	<i>Clienttest</i>	44
2.4.3	Zertifizierung	45
2.4.4	Aktueller Stand, Erfahrungen, Erkenntnisse	46
	<i>Profile</i>	46
	<i>CTT</i>	46
	<i>Zertifizierung</i>	48
2.5	OPC-UA-Informationsmodelle und deren Modellierung	49
	(Dr. MIRIAM SCHLEIPEN / ROBERT HENBEN)	
2.5.1	Knoten	50
2.5.2	Knotentypen	51
2.5.3	Typendefinitionen	52
2.5.4	Referenztypen	53
2.5.5	Beispiel	54
2.5.6	Vorgehensweise	55
2.6	Vorgehensmodell zur Einführung von OPC UA in Produktionsanlagen	57
	(CHRISTOPH BERGER)	
2.6.1	Ausgangssituation für produzierende Unternehmen	57
2.6.2	Definition unternehmensinterner Randbedingungen	58
	<i>Informationssystem</i>	58
	<i>Informationsaustausch</i>	59
	<i>Datenerfassung</i>	60
2.6.3	Einführungsprozess	61
	<i>Projektteam</i>	61
	<i>Aktueller Status</i>	61
	<i>Zielstellung</i>	62
	<i>Bewertung</i>	63
3	Nachrüstooptionen, Produkte	65
3.1	Bedeutung und Konzepte von OPC UA für die Steuerungsebene	65
	(Dr. HENNING MERSCH)	
3.1.1	Echtzeit	66
3.1.2	PLCopen: Mapping und Funktionsbausteine für die Steuerungsebene ..	67
3.1.3	OPC UA Publisher-Subscriber für die Steuerungsebene	68
3.1.4	Unterstützung von Domänen-spezifischen Modellen in der Steuerung (UA NodeSets für Companion Specifications)	69
3.1.5	Das TwinCAT-Konzept der SoA-PLC	71
3.1.6	OPC UA in der Steuerung – heute und morgen	72
3.2	OPC UA in der Steuerung – Siemens	73
	(JAN BAJORAT / ROBERT WINTER)	
3.2.1	Siemens und OPC UA	73
3.2.2	OPC UA und PROFINET	74
3.2.3	OPC UA in der SIMATIC S7-1500	74
	<i>Übersicht der Funktionalität</i>	75
	<i>Methoden</i>	75
	<i>Konfiguration</i>	77

	<i>Performance und Details</i>	78
	<i>SiOME (Siemens OPC UA Modeling Editor)</i>	81
	<i>Client</i>	83
3.2.4	Weitere OPC-UA-Implementierungen in der SIMATIC	83
	<i>SIMATIC 57-1200</i>	84
	<i>SIMATIC NET OPC UA Server</i>	84
	<i>SIMATIC CP 443-OPC UA</i>	85
	<i>SIMATIC WinCC Advanced & Professional</i>	85
	<i>TeleControl</i>	85
3.3	Wie Altgeräte und neue Applikationen und Services von den Vorzügen von OPC UA profitieren	85
	(CHRIS MÜNCH / JOHN TRAYNOR)	
3.3.1	Was für eine umfassende OPC-UA-Implementierung spricht	86
3.3.2	Ansatz 1: Schnelle Massenmigration	87
3.3.3	Ansatz 2: Langsame Migration mit Support mehrerer Legacy-Protokolle	87
3.3.4	Ansatz 3: Migration über OPC UA Gateways	88
3.3.5	OPC UA für Geräte ohne Softwareschnittstelle	90
3.3.6	Der Einsatz von OPC UA auch jenseits des industriellen Umfeldes	91
3.4	OPC UA in eingebetteten Systemen	93
	(CHRIS PAUL IATROU / Prof. Dr. LEON URBAS)	
3.4.1	Kategorien eingebetteter Mikrorechensysteme	95
3.4.2	Einschränkungen eingebetteter Systeme	96
3.4.3	Anforderungen an OPC UA Stacks in eingebetteten Systemen	99
3.4.4	Umsetzungsstrategien für eingebettete OPC UA Stacks	100
3.4.5	Hierarchische Codierung	101
3.4.6	Timing-Determinismus und Scheduling	101
3.4.7	Entkopplung der Kommunikationsschnittstelle	102
3.4.8	Speicherallokation	102
3.4.9	Entkopplung der Applikation von OPC UA	103
3.4.10	Verarbeitung von Nachrichten	104
3.4.11	Hinterlegung von Adressrauminhalten	105
3.4.12	Methodenaufrufe und Interrupts	107
4	Implementierungshilfen	109
4.1	Entwicklerbibliotheken und Tools von Unified Automation	109
	(UWE STEINKRAUSS)	
4.1.1	Open Source oder kommerziell	109
4.1.2	Sprache	110
4.1.3	Betriebssystem	111
4.1.4	OPC-UA-Funktionalität	111
4.1.5	Nutzbarkeit	113
4.1.6	Interface-Design	114
4.1.7	Erweiterungen	115
4.1.8	Performance und Ressourcen	115
4.1.9	Migration von Bestandsanlagen	116
4.1.10	Testtool und Performancevergleich	116

4.2	Entwicklerbibliotheken und Tools von Prosys OPC	117
	(JOUNI ARO / HEIKKI TAHVANAINEN)	
4.2.1	Prosys OPC UA Java-SDK – Zertifizierte Multiplattform OPC UA	118
	<i>Entwicklung mit Java</i>	118
	<i>Allgemeine Funktionen</i>	118
	<i>Zertifizierte OPC-UA-Konformität</i>	119
	<i>Funktionen des Client SDK</i>	119
	<i>Funktionen des Server SDK</i>	122
	<i>Codegenerator</i>	124
4.2.2	Kostenfreie Test-Tools	126
	<i>Prosys OPC UA Simulation Server</i>	126
	<i>Prosys OPC UA Client</i>	127
	<i>Prosys OPC UA Client für Android</i>	127
4.2.3	Prosys OPC UA Historian	128
	<i>Überblick über die Funktionen des Historian</i>	129
	<i>Ausblick</i>	130
4.2.4	Prosys OPC UA Modbus-Server	130
	<i>Sichere Datenkommunikation</i>	131
	<i>Neue Integrationsmöglichkeiten</i>	132
	<i>Allgemeine Funktionen</i>	132
	<i>Ausblick</i>	132
4.3	Open-Source-Implementierungen von OPC UA am Beispiel von open62541 ...	133
	(JULIUS PFROMMER)	
4.3.1	Vorteile von Open-Source-Implementierungen	133
4.3.2	Nachteile einer Open-Source-Implementierung	135
4.3.3	Ziele des open62541-Projektes	136
5	Anwenderberichte	139
5.1	Candy Hoover Use case	139
	(NADIA SCANDELLI / MIRCO MASA)	
5.1.1	Produktionsszenario	140
5.1.2	Qualitätsszenario	142
5.1.3	Beschreibung der Applikation	142
5.1.4	OPC-UA-Architektur	144
5.1.5	Der Einsatz von OPC UA: Nutzen und Betrachtungen	145
5.2	Voith – OPC UA aus Anwendersicht	146
	(DANIEL PAGNOZZI)	
5.2.1	Einleitung	146
5.2.2	Herausforderung für den Aufbau eines Industrie-4.0-Projektes im Konzern	148
	<i>Die Voith Composites GmbH & Co. KG</i>	148
	<i>Die Projektidee</i>	148
	<i>Der Standort</i>	149
	<i>Die Auswahl der Partner</i>	149
	<i>Die Projektmethode</i>	150
5.2.3	Anbindung der Gewerke mittels OPC UA	150
	<i>Auswahl OPC UA als M2M-Maschinenkommunikationsprotokoll</i>	150
	<i>Automatisierungskonzept mittels OPC UA</i>	151

	<i>Standardisierung der Konnektivität</i>	151
	<i>Datenmanagement und Semantik</i>	153
5.3	OPC-UA-Funktionalität für bestehende Steuerungen von Festo mittels Firmware-Update	155
	(MARTIN PLANK / Dr. ANDREAS Graf GATTERBURG)	
5.3.1	Motivation und technische Voraussetzungen	155
5.3.2	Beispiel zur Einrichtung eines OPC UA Servers durch ein Firmware- Update	156
	<i>Vorbereitende Maßnahmen</i>	157
	<i>Firmware-Update</i>	158
	<i>Nutzen von OPC UA in Codesys</i>	159
	<i>Aufbau einer Verbindung und Testen des Servers</i>	160
5.3.3	Beispiel der kombinierten Maschinen- und Energiedatenerfassung	162
5.4	PLUGandWORK mit OPC UA und AutomationML	164
	(Dr. OLAF SAUER / Dr. MIRIAM SCHLEIPEN / Dr. LJILJANA STOJANOVIC)	
5.4.1	PLUGandWORK und Industrie 4.0	164
5.4.2	OPC UA und AutomationML – Technologien für die Praxis	165
5.4.3	Umsetzung	167
5.5	Agiles Service Engineering unter Beachtung von Plattformfähigkeiten	170
	(Dr. THOMAS USLÄNDER)	
5.5.1	Motivation	170
5.5.2	Stand der Technik	171
	<i>Service-orientierte Analyse und Design</i>	171
5.5.3	Standpunkte und Sichten auf eine Systemarchitektur	171
5.5.4	Die SERVUS-Methodik	174
	<i>Übersicht</i>	174
	<i>Ablauf bei der Erfassung von Anforderungen</i>	176
	Sponsored Content	181
	Management-Statement: OPC UA wird fester Baustein innerhalb der HIMA Smart Safety Platform	183
	Abkürzungen	187
	Lebensläufe	191
	Literaturverzeichnis	197
	Quellenverzeichnis der Bilder	203
	Stichwortverzeichnis	205

1 Keine Industrie 4.0 ohne OPC UA

Eine zentrale Herausforderung von Industrie 4.0 und dem IIoT (Industrial Internet of Things) ist der sichere, standardisierte Daten- und Informationsaustausch zwischen Geräten, Maschinen und Diensten auch aus verschiedenen Branchen. Bereits im April 2015 hat das RAMI 4.0 (Reference Architecture Model for Industry 4.0) [RAMI4.0] den IEC-62541-Standard OPC Unified Architecture (OPC UA) [IEC62541] als einzige Empfehlung für die Umsetzung des Kommunikationslayers gelistet. Im November 2016 hat die Plattform Industrie 4.0 einen Leitfaden veröffentlicht: «Welche Kriterien müssen Industrie-4.0-Produkte erfüllen?» [ZVEI4.0]. Produkthersteller können anhand einer Checkliste prüfen, ob ihre Produkte die Kriterien der Kategorien Industrie 4.0 «Basic», «Ready» oder «Full» erfüllen: Bereits die niedrigste Stufe listet bei dem Kriterium «I4.0-Kommunikation» die Anforderung, dass das Produkt im Netzwerk online per TCP / UDP oder IP mit mindestens dem Informationsmodell von OPC UA ansprechbar sein muss. Wer also mit dem Produktstempel «Industrie 4.0 enabled» werben will, muss OPC-UA-fähig sein (integriert oder per Gateway). Explizit wird auch die Eigenschaft der Informationsmodellierung von OPC UA hervorgehoben.

1.1 OPC UA ist kein Protokoll – sondern viel mehr

Informationsmodellierung? Viele kleinere und mittlere Unternehmen schalten hier bereits ab – schnell wird dann OPC UA mit anderen Protokollen wie MQTT verglichen, und vermeintliche Einschränkungen in Einsatzszenarien werden festgestellt: «OPC UA direkt in die Cloud geht nicht – oder?» Was also ist OPC UA?

OPC UA ist ein Framework für industrielle Interoperabilität: Geräte- und Maschinenbauer beschreiben die objektorientierten Informationen ihres Systems und definieren auch die Zugriffsrechte mit integrierter IT Security dazu. Das deutsche BSI hat die Ergebnisse der OPC-UA-Sicherheitsanalyse [BSI] auf seinem Web veröffentlicht und sich dabei sehr positiv über OPC UA geäußert. Der Maschinenbauer bleibt damit Herr seiner Daten bzw. er kann sie gezielt und kontrolliert verteilen und somit auch an Big Data und der Analyse seiner Daten monetär teilhaben.

Um diese Daten nun auszutauschen, vereinigt OPC UA zwei Mechanismen für die Umsetzung unterschiedlicher Szenarien:

1. ein Client-Server-Modell, in dem OPC UA Clients die dedizierten Dienste des OPC UA Servers nutzen (siehe Kapitel 2). Dieser Peer-to-peer-Kontext wird genutzt für den sicheren, bestätigten Informationsaustausch – aber mit Einschränkungen in der Anzahl der Verbindungen;
2. ein Publisher-Subscriber-Modell, bei dem ein OPC UA Server konfigurierbare Untermengen von Informationen an eine beliebige Anzahl Zuhörer verfügbar macht (siehe Abschnitt 3.1). Diese

Broadcast-Verteilung an viele agiert unter dem Aspekt «Fire and forget» als unbestätigter Informationsaustausch.

OPC UA bietet beide Mechanismen – wichtig aber: Sie sind losgelöst vom eigentlichen Protokoll umgesetzt! So stehen TCP und HTTPS für die Client Server und UDP und AMQP und MQTT für das Subscriber Model zur Verfügung.

Die Frage «OPC UA oder MQTT» stellt sich somit aus OPC-Foundation-Sicht nicht.

Der eigentliche Schlüssel für Industrie 4.0 liegt jedoch auf der Bedeutung und der Beschreibung der Daten – der sogenannten Informationsmodellierung.

Unbewusst liefert jeder Geräte- und Maschinenbauer bereits heute ein Informationsmodell: Daten und Schnittstellen sind bereits (über diverse Protokolle) verfügbar. Wir Menschen haben uns an die Denkweise der Computer angepasst und haben in Dokumentationen festgehalten, welche Bedeutung sich hinter den Bits & Bytes & Hexcodes verbergen. Die neue Welt der SoA-fähigen Geräte unterstützt uns Menschen nun schneller und einfacher, die «Things» zu verstehen, da diese «Dienste», aber noch wichtiger, deren Bedeutung anbieten. Gar nicht neu ist das Thema SoA (**S**ervice-**o**rientierte **A**rchitektur) in der IT-Welt – nun wandert aber diese SoA bis in die «Things» selber vor.

1.2 Migrationsschritte zur Einführung von OPC UA

Die Einführung von SoA in den Maschinen und Geräten, basierend auf OPC UA, kann schrittweise erfolgen (siehe dazu auch Abschnitt 2.6). Erste notwendige Voraussetzung ist ein auf dem Internet-Protokoll (IP) basierendes Netzwerk. Bestehende Geräte mit einer seriellen Schnittstelle benötigen somit ggf. ein Gateway, um im Netz als IP-fähiges Gerät erkannt zu werden. OPC UA kann dann bereits als einheitliche Schnittstelle für den Maschinen-übergreifenden Informationszugriff genutzt werden. Jedes Gerät und jede Maschine bietet dann seine / ihre Daten und Dienste per OPC UA an – bereits dieser erste Schritt bedeutet einen großen Vorteil: Viele unterschiedliche Kommunikationslösungen sind somit auf einen Mechanismus vereinigt – dieser eine Standard OPC UA liefert auch noch deutlich bessere Security (siehe Abschnitt 2.3).

1.3 Das war's? Daten- und Interface-Standards für Branchen

In einem nächsten Schritt wäre es doch angenehm, wenn unterschiedliche Hersteller, die aber gleiche Geräte liefern, sich auch auf die gleichen Daten und Dienste einigten. Die Integration dieser Geräte wäre viel einfacher und würde ein Plug & Play liefern. Genau das liefert die OPC Foundation mit vielen Partnern bei der gemeinsamen Erstellung von standardisierten Informationsmodellen (Companion Specifications). Bevor also ein Geräte- und Maschinenbauer startet, sollte geprüft werden, ob es nicht schon ein standardisiertes Informationsmodell gibt. Diese Interoperabilität wird von mehr und mehr Betreibern aus verschiedenen Bereichen der Factory- und Prozess-Industrie, aber auch der Energie- oder Logistikbranche gefordert (siehe Abschnitt 2.2). Sogar in industriellen Großküchengeräten (Verband HKI) ist OPC UA inzwischen in industriellen Kaffeemaschinen angekommen.

Verbände (siehe Bild 1.1) wie z.B. PLCopen, AutomationML, AIM für die AutoID-Industrie (RFID Reader, Scanner usw.), VDMA-Fachgruppen wie Spritzgussmaschinen, Robotik oder Machine Vision (und 35 weitere VDMA-Branchen) definieren bereits ihre Informationen in OPC UA Servern – einer sogenannten OPC-UA-Companion-Spezifikation.



Bild 1.1 Kooperationspartner der OPC Foundation

1.4 Differenzierung!

Als Anbieter einen Branchenstandard zu erfüllen, bedeutet nicht direkt austauschbar zu sein: Jeder Hersteller kann seine eigenen speziellen Dienste weiterhin parallel zum Standard anbieten. Dafür bietet OPC UA die Erweiterung mit sogenannten Hersteller-spezifischen Informationsmodellen. Intelligente Geräte sollten also unbedingt parallel mehrere Informationsmodelle unterstützen können: neben der dedizierten Funktionalität z.B. der Spritzgussmaschine oder des Roboters eben auch die Modelle für Energiedaten oder MES-Schnittstellen usw. Zur Reduzierung des Engineerings wird die Bedeutung dieser Branchen- und Cross-Branchen- Informationsmodelle in Zukunft stark steigen: Mit OPC UA wird man gegebenenfalls nicht mehr Geräte verkaufen – aber Hersteller, die hier nicht den OPC-UA-Standard unterstützen, werden in Zukunft deutlich weniger verkaufen können. Die Wünsche nach einem Daten- und Schnittstellenstandard (der dann über verschiedene Protokolle ausgetauscht werden kann) haben letztendlich gleiche Motivationen zur Reduzierung von Aufwänden: Betreiber und Integratoren wünschen ein Plug&Produce für die Reduzierung von Engineering für Systeme unterschiedlicher Hersteller – Maschinenanbieter wollen sich auf ihre Kernkompetenz konzentrieren und nicht in verschiedene Konnektoren für verschiedene Eco-Systeme der Endkunden oder ERP & Cloud investieren.

1.5 Ausblick

1.5.1 Trend: SoA und PubSub

Die Welten ticken unterschiedlich: Während die IT schon immer service-orientiert (SoA) betrieben hat, ist in der Automatisierung auf unterster IO-Ebene die zyklische PubSub-Verarbeitung gefordert. OPC UA bietet beides – die OPC-UA-Architektur ist erweiterbar angelegt. Die bisher entstandenen branchenspezifischen Informationsmodelle fokussieren auf die Anbindung zur IT und bieten ihre Daten meistens nicht mehr per Bit/Byte-Property-Austausch, sondern per SoA-Diensten mit komplexen Parametern an. Ein OPC UA Client, der keine Methoden oder komplexe Parameter unterstützt, wird mit OPC UA Servern

zunehmend wenig kommunizieren können. Service-basierte Ansätze mit OPC UA beleuchtet Abschnitt 5.5.

1.5.2 Trend: «Service to Service»

OPC UA bietet eine durchgängige Skalierung vom Sensor bis in die IT-Enterprise-Ebene – mit erheblichen Auswirkungen auf die Automatisierungspyramide: Diese Pyramide bleibt sicherlich weiterhin bestehen für die organisatorische Struktur einer Fabrik – die Kommunikationspyramide ist mit OPC UA aber komplett aufgehoben: Geräte können Daten direkt oder parallel an die SPS, das MES, die ERP oder die Cloud-Ebene liefern. Hier wittern viele Anbieter neue Geschäftsmodelle – die gesicherte Verteilung von Informationen macht's möglich. Mit möglichen Architekturen befasst sich Abschnitt 2.1.

1.5.3 Trend: OPC UA im Chip

OPC UA wird in immer kleinere Geräte und Sensoren «wachsen». Kleinste OPC-UA-Softwarelösungen benötigen heute mit limitierter (aber auslesbarer) Funktionalität 35 kb RAM und 240 kb Flash. Die ersten Chips mit integriertem OPC UA sind am Markt verfügbar, und so wächst OPC UA weiter in die Sensorwelt hinein. Schon jetzt wachsen die Anwendungen von OPC UA aus dem Kernbereich der Automatisierung in andere Branchen wie z.B. industriellen Großküchengeräten. Um OPC in eingebetteten Systemen geht es in Abschnitt 3.4.

1.5.4 Zukunft: OPC UA via TSN

In Zukunft wird mit der Kombination OPC UA via TSN (**T**ime-**S**ensitive **N**etworking) [TSN] auch die Echtzeitfähigkeit garantiert werden. OPC UA liefert dabei weiterhin die sichere Interoperabilität – TSN und 5G liefern den echtzeitfähigen Layer 2, um OPC UA zu tunneln. Die OPC Foundation hat dazu im November 2018 die Initiative «Field Level Communications» gegründet. Die Vision der Initiative ist es, eine offene einheitliche, sichere und standardbasierte IIoT-Kommunikationslösung zwischen Sensoren, Aktoren, Controllern und Cloud anzustreben, die alle Anforderungen der industriellen Automatisierung erfüllt. Explizit sind damit auch deterministische Kommunikation, funktionale Safety und Motion als optionale Eigenschaft adressiert – OPC UA würde so die Möglichkeit schaffen, «die Option TSN einzuschalten», wenn eine deterministische Übertragung notwendig ist, z.B. zur horizontalen Kommunikation zwischen Controllern oder auch vertikal in die Feldebene. Aktuell müssen allerdings zunächst die TSN-Spezifikationen unter dem Dach IEEE weiter entwickelt werden, um ein Netzwerk von mehreren TSN-Knoten von verschiedenen Herstellern konfigurieren zu können.

1.5.5 Cross-Domain-Harmonisierung

Die Standardisierung der Daten und Interfaces inklusive der Bedeutung ist ein großes Bestreben von Industrie 4.0 – aktuell definieren 50+-Gruppen weltweit OPC UA Companion Specifications für ihre Branchen im Bereich Fabrik- und Prozessautomation, Energie, Öl & Gas, Pharmazie, industrielle Großküchengeräte usw. Die wirkliche Herausforderung der OPC Foundation liegt indes in der Cross-Domain-Harmonisierung der Informationsmodelle, um dort keine «Konflikte

der Informationsmodelle» entstehen zu lassen und mehrere Definitionen für OEE-Daten, MES- und Powermanagement-Schnittstellen usw. zu vermeiden oder wenigstens zu minimieren. Die OPC Foundation hat das erkannt und bereits eine «Cross-Domain Harmonization»-Arbeitsgruppe gestartet.

In Kürze

OPC UA ist bereits als De-facto-Standard im Automatisierungsbereich und für Industrie 4.0 gesetzt. OPC UA deckt immer weitere Szenarien der Kommunikation ab und es wird für Anbieter schwieriger, eigene Lösungen zu erklären. Eine Differenzierung wird zunehmend nicht mehr an der Schnittstelle, sondern durch Faktoren im Gerät oder Diensten außerhalb erfolgen – nicht aber an der Schnittstelle selbst. Für die Zukunft ist mit einem rasanten Anwachsen der weiteren Informationsmodelle in anderen Branchen zu rechnen: OPC UA ist die Plattform des weltweit größten Eco-Systems für Interoperabilität.

2 Grundlagen

2.1 Architekturen im Einsatz

Die OPC Unified Architecture bietet Mechanismen zur standardisierten, synchronen oder asynchronen, verteilten Kommunikation. OPC UA ermöglicht den Zugriff auf Daten unterschiedlichster Art in vertikaler als auch horizontaler Richtung. Dies ermöglicht ein breites Anwendungsspektrum, schließt aber auch unterschiedliche Architekturvarianten und die Betrachtung der zugrunde liegenden Netzinfrastruktur ein. OPC-UA-Komponenten können in unterschiedlichsten Varianten, Plattformen, von verschiedenen Herstellern kombiniert werden. Das Spektrum reicht von OPC-UA-Komponenten direkt integriert auf den Geräten und Steuerungen bzw. Maschinen und Anlagen über sogenannte Gateways bis hin zu aggregierenden Servern. Daher sollen nachfolgend mögliche Architekturvarianten mit einigen Beispielen vorgestellt werden, an denen sich zukünftige Anwender orientieren können.

DEFINITION

Unter dem Begriff Architektur versteht man die «Kombination von Elementen eines Modells, aufbauend auf Prinzipien und Regeln zum Zweck seiner Konstruktion, Weiterentwicklung und Nutzung». [I40Statusreport]



Im Fall von OPC UA handelt es sich um die IT-Architektur (System- und teilweise Software-Architektur), die diverse Einflussfaktoren zur Ermittlung möglicher Realisierungsvarianten einbeziehen muss. Dies können beispielsweise sein: Produktionskomponenten und deren Schnittstellen sowie weitere vorhandene Infrastruktur, Hardware, Software, Schnittstellen, Netzwerke oder Standorte.

Die Definition einer Architektur ermöglicht die ganzheitliche Sicht auf ein System mit OPC-UA-Komponenten und sollte vor Beginn der Realisierung in jedem Fall (wenn auch nur rudimentär) erstellt und diskutiert worden sein.

OPC-UA-Server- und auch -Client-Funktionalität wird normalerweise aufgrund der Komplexität nicht von Grund auf implementiert, sondern auf Basis sogenannter Software Development Kits (SDKs) erstellt. Hier gibt es sowohl kommerzielle Varianten (siehe Abschnitte 4.1 und 4.2) als auch Open-Source-Lösungen (siehe Abschnitt 4.3), die abhängig von der Prozessorarchitektur, vom Betriebssystem, aber auch von der Programmiersprache einzusetzen sind.

Prinzipiell sollten sich Anwender vor der Nutzung von OPC UA über den Einsatz und Nutzen von OPC UA sowie die vorhandene Infrastruktur und Umgebung im Klaren sein.

- Ist OPC UA als Brücke zwischen verschiedenen Komponenten und Stationen auf der Feldebene geplant, können alle Komponenten innerhalb des Fabriknetzes betrieben werden (siehe Abschnitt 2.3).

- Ist ein überwachender und steuernder Eingriff auf Basis von berechneten Kennzahlen, zentralen Arbeitszeitmodellen oder anderen MES-Funktionalitäten angedacht, wird die MES-Funktionalität sicherlich von der zentralen IT bzw. im Rechenzentrum betrieben. Ein Übergang zwischen Fabriknetz und Office-IT, beispielsweise unter Nutzung einer DMZ (**demilitarisierte Zone**) sollte diskutiert werden (siehe Abschnitt 2.3).
- Soll das Ganze dem Zweck der Fernwartung dienen, ist sicherlich eine Verbindung zu beispielsweise Clients in externen Netzen über verschiedene Netzgrenzen hinweg nötig. Auch hier sind nochmals Diskussionen über eine DMZ zwischen zentraler IT und dem Internet bzw. externen Firmennetzen sinnvoll (siehe Abschnitt 2.3).
- Für die Überwachung einer bestimmten Gerätegruppe ist vielleicht kein schreibender Zugriff nötig, sondern alle Werte werden nur lesend bereitgestellt. Dies senkt das Risiko der Beeinflussung des Produktionsprozesses und reduziert die zu betrachtenden Bedrohungen.
- Aber auch die Anzahl an zu erwartenden Kommunikationspartnern sollte vorab eingegrenzt werden. Es muss geklärt werden, welche der möglichen Realisierungsvarianten das Mengengerüst (nicht nur hinsichtlich der zu übertragenden bzw. verwalteten Informationen) überhaupt abdecken können.
- Sofern selbst implementiert und kein fertiges Produkt gekauft werden soll, ist die einzusetzende Programmiersprache ebenfalls ein zu berücksichtigender Faktor, der die möglichen Programmierhilfen (SDKs, siehe Abschnitte 4.1, 4.2 und Abschnitt 4.3) einschränkt.
- Eventuell gibt es verschiedene Realisierungsvarianten, beispielsweise mit einem in die Steuerung integrierten OPC UA Server, der Werte des Geräts für den lesenden Zugriff bereitstellt, oder einem OPC UA Client, der aktiv diese Werte in einen zentralen OPC UA Server schreibt. Diese müssen mit all ihren Vor- und Nachteilen bezüglich Kosten, Wartungsaufwand, Erweiterbarkeit und vielen anderen Zielkriterien gegenübergestellt werden, bevor strategische Entscheidungen getroffen werden.
- Außerdem muss geklärt werden, wo die einzelnen OPC-UA-Komponenten lokalisiert sind. Sind diese im eigenen Produktionsnetz, direkt an der Maschine, laufen sie im Rechenzentrum des Unternehmens oder sind sie gar beim Hersteller der Komponenten (außerhalb des Firmennetzes) zu finden? Der Einsatzort beeinflusst die Realisierbarkeit ebenso wie die vorhandene Infrastruktur und Implementierungsumgebung. Hiermit hängt eng das Konzept zur IT-Sicherheit (siehe Abschnitt 2.3) zusammen.
- Schließlich beeinflusst der Einsatzzweck auch das zugrunde liegende Informationsmodell (siehe Abschnitt 2.5), die zu realisierende Funktionalität (siehe Abschnitt 2.4), die Nutzung möglicher Companion Specifications (siehe Abschnitt 2.2) und damit auch die Realisierungsvarianten, die diese Rahmenbedingungen umsetzen können.

Zusammenfassend lässt sich die Zielarchitektur nicht planen, ohne unter anderem die genannten Einflussfaktoren zu kennen und zu berücksichtigen. Eine mögliche Strategie für den Einführungsprozess ist in Abschnitt 2.6 beschrieben. Zunächst müssen also die Anforderungen definiert und dann mit den möglichen Realisierungsvarianten abgeglichen werden (siehe Abschnitt 5.5).

OPC UA bietet mit den Möglichkeiten zur Konformitätsüberprüfung praktische Unterstützung, um die geforderte Funktionalität zu gewährleisten bzw. zu prüfen. Diese werden in Abschnitt 2.4 näher erläutert.

Die meisten OPC-UA-Anwendungen basieren auf dem Client-Server-Modell für verteilte Anwendungen. Der Server stellt Daten und Dienste bereit, die der Client anfordert und konsumieren kann. Die bidirektionale Kommunikation zwischen Client und Server folgt dem Request-Response-Modell. Der Client fordert, der Server antwortet. Dies bedeutet im Umkehrschluss, dass der Server in letzter Instanz die Hoheit über die Daten besitzt.

Anmerkung: Zur Unterstützung von verbindungsloser Kommunikation, beispielsweise in Anwendungsfällen des Industrial Internet of Things (IIoT), unterstützt OPC UA mit der neuesten Pub-Sub-Spezifikation [pubsub] ebenfalls das verbindungslose Publish-Subscribe-Modell. Dieses Kapitel fokussiert allerdings die zuvor genannten Client-Server-Anwendungen.

2.1.1 Fall 1: 1 OPC UA Server je Asset

Jede SPS, jedes Gerät, jede Maschine und Anlage erhält in diesem Fall einen eigenen OPC UA Server mit individuellem (vom Hersteller des Servers vorgegebenem) Informationsmodell.

Hierbei sind unterschiedliche Ausprägungen für die einzelnen Assets möglich.

1. Die naheliegende Variante ist ein OPC UA Server des Geräte-, Maschinen- oder Steuerungsherstellers selbst, der auf dem Gerät / der SPS / der Maschine integriert ist (Bild 2.1, links unten). Beispiele finden sich in den Abschnitten 3.1, 3.2 und 3.4.
2. Diese OPC UA Server müssen nicht zwingend auf dem Gerät / der SPS / der Maschine direkt lokalisiert sein, sondern können auch in Software auf einem anderen Rechner ablaufen (Bild 2.1, Mitte unten).
3. Aber auch Adapter (Bild 2.1, rechts unten), z.B. für bestimmte Steuerungstypen, sind heute in der Praxis häufig im Einsatz. In Tabelle 2.1 sind einige aufgeführt, allerdings ohne Wertung oder Anspruch auf Vollständigkeit.

Tabelle 2.1 Adapter zur Nachrüstung von OPC UA

Hersteller	Produkt
Bosch Rexroth	IoT Gateway
CERTEC EDV	atvise scada + box
Contec	Comprosys
eWON (HMS)	eWON Flexy
Harting	MICA Base
HB-Softsolution e.U.	HB-DataHub
Hilscher	netIoT Edge
HMS	OPC UaGateway
IBHsofttec	IBH Link UA
softing	echocollect
SOFTING Industrial	dataFEED uaGate
SOTEC	CloudPlug
Spectra	Spectra PowerBox 100-IOT

In einigen Fällen kann die OPC-UA-Funktionalität auch nachträglich auf bestehenden Geräten nachgerüstet werden (siehe Abschnitt 5.3).

Hierbei besteht die Herausforderung beim Anwender häufig darin, anhand definierter Anforderungen und Rahmenbedingungen die richtigen Produkte auf dem Markt zu finden (siehe Abschnitt 5.2) oder aber die geeigneten Entwicklungsarbeiten anzustoßen.

Diese Architekturvariante wird häufig für den Zugriff auf aktuelle Prozesswerte per OPC UA genutzt.

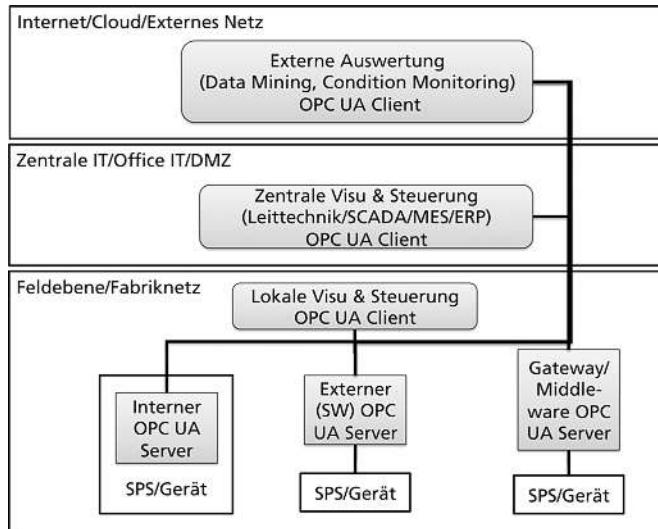


Bild 2.1 Fall 1

OPC UA Clients können in diesem Fall sowohl lokale OPC UA Clients beispielsweise in einer anlagennahen Visualisierung sein, aber auch zentrale SCADA- oder MES-Systeme, die von der zentralen IT betrieben werden, oder sogar externe Anwendungen, die beispielsweise zum Condition Monitoring genutzt werden.

2.1.2 Fall 2: 1 externer Software OPC UA Server für alle Assets

Ein externer Software OPC UA Server (Bild 2.2) bündelt in diesem Fall die Informationen mehrerer Assets, z.B. ein OPC UA Server für einen bestimmten SPS-Typ, der für mehrere SPSen des gleichen Typs zuständig ist. Meist sind dies OPC UA Server des Steuerungsherstellers selbst oder OPC UA Server, die von den Herstellern der SDKs angeboten werden. Dies kann aber auch ein in Software realisierter OPC UA Server sein, der eine bestimmte Funktionalität realisiert, wie z.B. das Qualitätsmanagement (Candy Hoover Use case). Dadurch, dass meist ein SPS-Typ oder eine bestimmte Anwendung unterstützt wird, ist auch die Grundstruktur des Informationsmodells vorgegeben.

Mögliche OPC UA Clients sind in diesem Fall identisch zu Fall 1.

2.1.3 Fall 3: 1 externer Middleware OPC UA Server / externes Gateway für alle Assets

Ein externes Gateway bzw. ein externer Middleware OPC UA Server (Bild 2.3) integriert in diesem Fall die Informationen aller Assets. Dies kann sowohl ein Gateway für einen bestimmten Steuerungstyp sein, aber auch eine Middleware, die neben OPC UA als Kommunikationskanal noch einen oder mehrere weitere Kommunikationsprotokolle unterstützt, z.B. CANopen oder ModbusTCP, und deren Variablen auf OPC-UA-Variablen «umsetzt». Voraussetzung hier ist die Einigung auf eine einheitliche Grundstruktur des Informationsmodells, die meist vom Hersteller des Gateways / der Middleware vorgegeben ist.

Mögliche OPC UA Clients sind in diesem Fall identisch zu Fall 1.

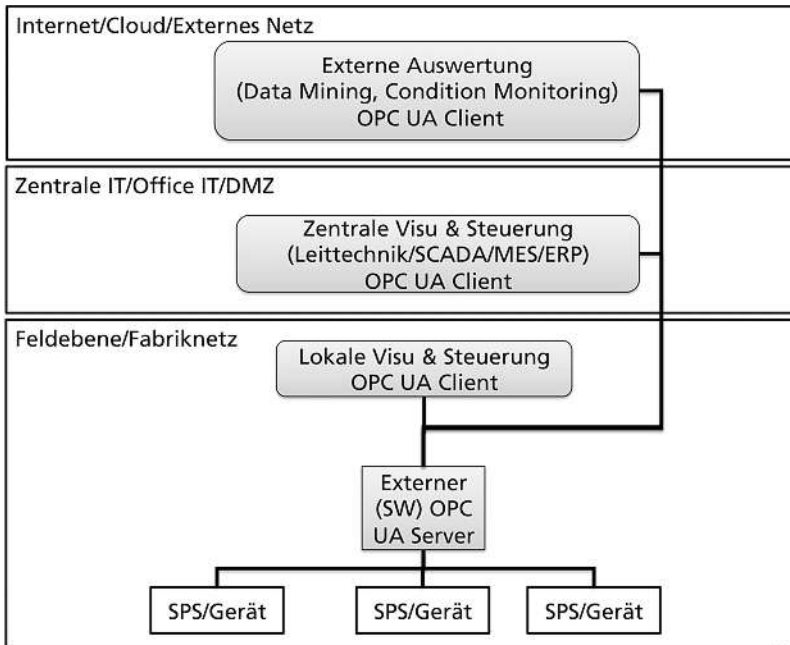


Bild 2.2 Fall 2

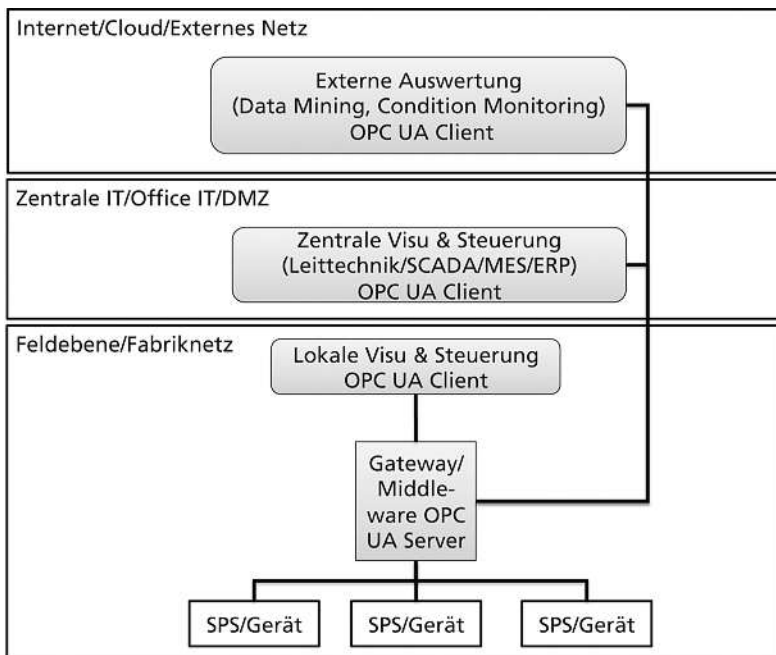


Bild 2.3 Fall 3