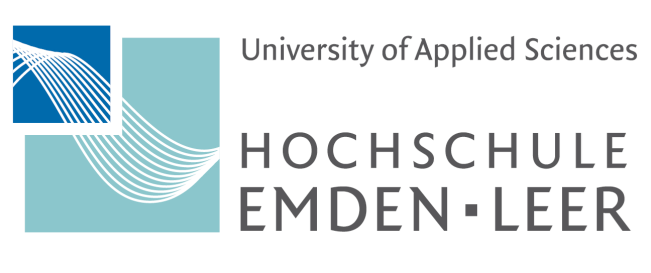
OPC UA, Sicherung des kuka-roboters

*Prof. Dr. Elmar Wings*

*Yahya Fakhet:7012464 Faissal Hammouda:7012301*



**WS/2021**



* ***Kommunikationsprotokoll : OPC UA***
* ***Backup-Manager***

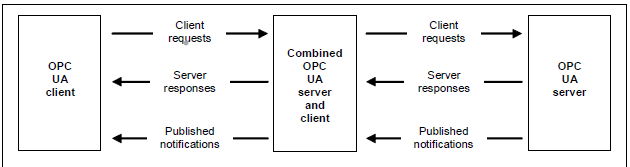


* ***KUKA / OPC UA***
* ***Quellen***

**2**



* Der aktuelle Standard der OPC Spezifikation ist OPC UA
* **OPC UA**: **O**pen **P**latform **C**ommunications **U**nified **A**rchitecture
* Eines der wichtigsten Kommunikationsprotokolle für Industrie 4.0 und IoT



* Datenaustauschstandard für die **industrielle Kommunikation** (Maschine-zu-Maschine oder PC-zu-Maschine-Kommunikation)
* Plattformunabhängigkeit durch Umstellung auf TCP / IP-Kommunikation
* OPC UA unterstützt eine semantische Beschreibung von Daten

**Abbildung 1: OPC UA Systemarchitektur**

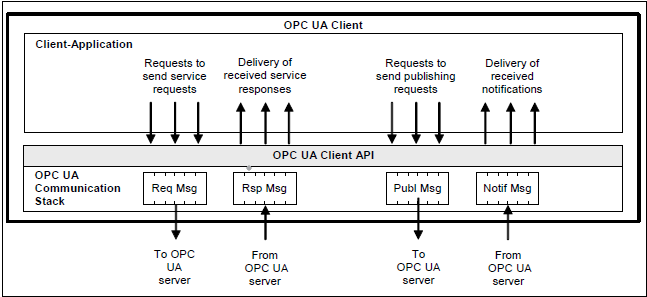
**3**



# *OPC UA Client*

* OPC UA Client Architektur modelliert den Client-Endpunkt von Client/Server Interaktionen

**Abbildung 2: OPC UA Client Architektur**



* ***Client Applikation*** ist der Code, der die Funktion des Clients implementiert
* Sie verwendet die OPC-UA-Client-API zum Senden und Empfangen von OPC-UA- Dienstanforderungen und -Antworten an den OPC-UA-Server

**4**



* ***OPC UA Client API*** ist eine interne Schnittstelle, die den Client-Applikationscode von einem OPC UA Communication Stack isoliert
* ***OPC-UA-Kommunikationsstack*** wandelt *OPC-UA-Client-API*-Aufrufe in Nachrichten um und sendet sie auf Anforderung der Client-Applikation durch die zugrundeliegende Kommunikationseinheit an den Server



* + **OPC-UA-Kommunikationsstack** empfängt auch Antwort- und Benachrichtigungsnachrichten von der zugrundeliegenden Kommunikationseinheit und stellt sie der Client-Applikation über die *OPC-UA-Client-API* zur Verfügung

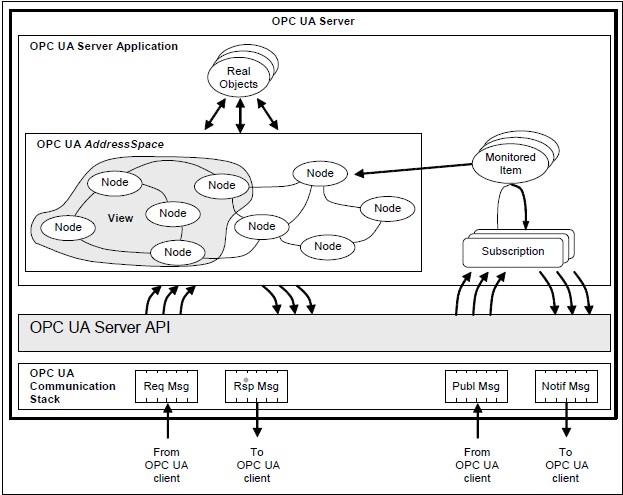
**5**



# *OPC UA Server*

* OPC UA Server-Architektur modelliert den Server-Endpunkt von Client/Server Interaktionen

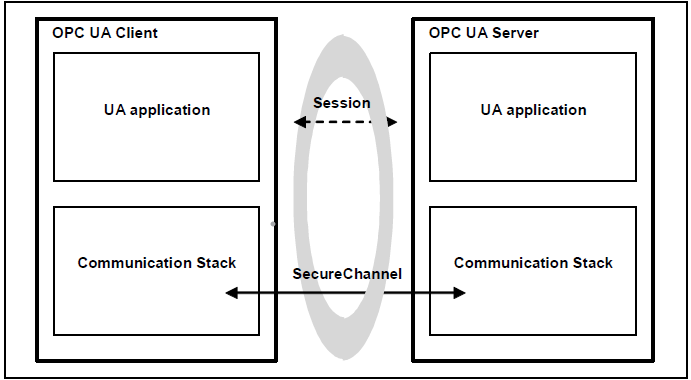
**Abbildung 3: OPC UA Server Architektur**



**6**



* ***Reale Objekte*** sind physikalische oder Software-Objekte, auf die die OPC-UA-Server- Applikation zugreifen kann oder die sie intern verwaltet
* ***OPC UA Server Applikation*** ist der Code, der die Funktion des Servers implementiert



* Er verwendet die OPC-UA-Server-API zum Senden und Empfangen von OPC-UA- Nachrichten von OPC-UA-Clients
* ***OPC UA Server API*** ist eine interne Schnittstelle, die den Server-Applikationscode von

einem OPC UA Communication Stack isoliert

**Abbildung 4: Sicherer Kanal und Sitzungsdienste**

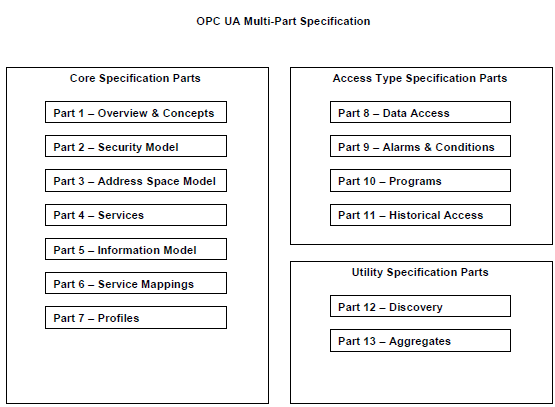
**7**



# *Organisation der Spezifikation*

* Diese Spezifikation ist als mehrteilige Spezifikation aufgebaut

**Abbildung 5: Organisation der OPC UA Spezifikation**



**8**



* Die ersten sieben Teile spezifizieren die Kernfähigkeiten von OPC UA. Diese Kernfähigkeiten definieren die Struktur des OPC-Adressraums und der darauf operierenden Dienste
* Die Teile 8 bis 11 wenden diese Kernfunktionen auf bestimmte Zugriffsarten an, wie z. B. Data Access (DA), Alarms and Events (A&E) und Historical Data Access (HDA)



* Teil 12 beschreibt Discovery-Mechanismen für OPC UA und Teil 13 beschreibt

Möglichkeiten zur Aggregation von Daten

**9**



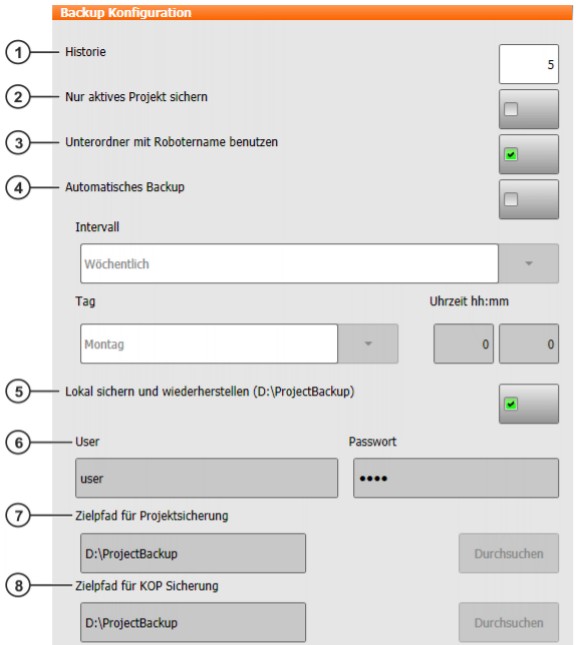
* Der Backup Manager ermöglicht das Sichern und Wiederherstellen von Projekten, Optionspaketen und RDC-Daten

# *Backup-Manager konfigurieren*



* **Voraussetzung:** Benutzergruppe Experte / Betriebsart **T1** oder **T2**
* **Vorgehensweise:** Im Hauptmenü **Datei** > **Backup-Manage**r > **Backup-Konfiguration** wählen
* **Backup-Konfiguration** enthält die allgemeinen Einstellungen. Außerdem kann hier bei Bedarf die automatische Sicherung konfiguriert werden

**10**



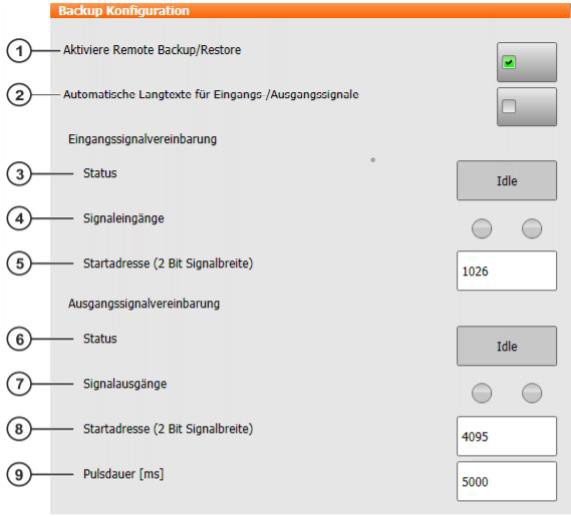
**Abbildung 6: Registerkarte Backup-Konfiguration**

**11**



* Unter **Signalschnittstelle** kann bei Bedarf die E/A-Ansteuerung konfiguriert werden

**Abbildung 7: Registerkarte Signalschnittstelle**



**12**



# *Sicherungsmethode*

* Erforderliche Schritte zur Durchführung eines Backups bei einem Kuka-Roboter mit der Steuerung Kr C4



* USB in den Controller einstecken
* unter Hauptmenü > Konfiguration > Benutzergruppe > Experte wählen
* Unter Hauptmenü > Datei > Archiv>USB(Kabine)>alle/Applikation/Systemdaten/Protokoll-

daten wählen

# *Wiederherstellungsmethode*

* USB-Stick mit den Sicherungsdateien in den Schaltschrank einstecken
* unter Menü>Datei>Wiederherstellen>USB ( Steuerung)>Alle/Anwendung/Systemdaten wählen

**13**



* Grundlage für die Kommunikation ist, dass die KUKA Robotersteuerung **KR C4** als ***Server***

und der ***Rechner*** als ***Client*** funktioniert. Dies ist in Übereinstimmung mit dem ***OPC UA***

# *KUKAVARPROXY*



* Der KUKAVARPROXY ist ein ***Server***, der auf der ***KUKA Robotersteuerung*** ausgeführt werden muss und mit dem internen System kommuniziert

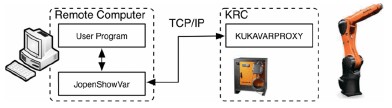
# *OpenShowVar*

* OpenShowVar ist ein ***Client***, der extern ausgeführt werden kann, um mit dem

KUKAVARPROXY über eine ***TCP/IP***-Verbindung zu kommunizieren

* OpenShowVar ist eine für die Plattformkompatibilität entwickelte ***Open-Source- Kommunikationsschnittstelle*** für die ***KUKA KR C4-Robotersteuerungen***
* KUKAVARPROXY ist der TCP/IP-Server, der auf dem Roboter läuft, während OpenShowVar der Client ist, der sich mit dem Server verbindet

**14**



**Abbildung 8: Client-Server-Architektur**

* OpenShowVar und KVP funktionieren als ***Middleware*** zwischen dem Anwenderprogramm

und dem KRL

* sie eröffnen die Möglichkeit einer alternativen Implementierung einer OPC-UA- Kommunikation auf der KR C4-Steuerung
* OpenShowVar bietet die Möglichkeit, globale Variablen von einem Rechner aus in die Robotersteuerung zu lesen und zu schreiben

Rechner

OPC UA

KUKAVARPROXY

KR C4

**Abbildung 9: Kommunikationsarchitektur**

**15**

* Open-Source-Lizenz von KUKAVARPROXY und OpenShowVar sind in Github.com verfügbar
* [KUKAVARPROXY](https://github.com/ImtsSrl/KUKAVARPROXY)



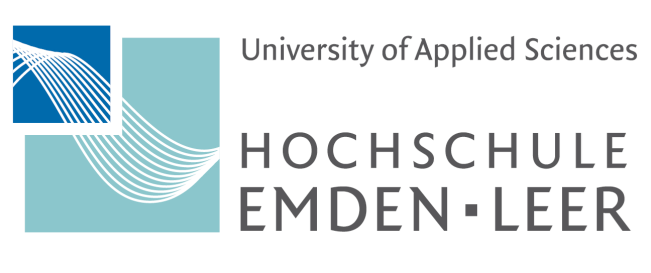
* [OpenShowVar](https://github.com/linuxsand/py_openshowvar)

**16**



# [*OPC Unified Architecture Specification ,Part 1: Overview and Concepts*](https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-1-overview-and-concepts/)

**17**



*Vielen Dank für Ihre*

*Aufmerksamkeit*