

# Schlussbericht

zu dem IGF-Vorhaben

**Architektur und Schnittstellen für ein weborientiertes Automatisierungssystem (WOAS)**

der Forschungsstelle(n)

FH Düsseldorf - Düsseldorfer Telelabor

Das IGF-Vorhaben 16753 N/1 der Forschungsvereinigung Elektrotechnik beim ZVEI e.V. - FE  
wurde über die



im Rahmen des Programms zur Förderung der Industriellen Gemeinschaftsforschung (IGF) vom



Bundesministerium  
für Wirtschaft  
und Energie

aufgrund eines Beschlusses des Deutschen Bundestages gefördert.

Düsseldorf, den 16.06.14

A handwritten signature in black ink, appearing to read "R. Langmann".

Prof. Dr.-Ing. Reinhard Langmann

Ort, Datum

Name und Unterschrift des/der Projektleiter(s)  
an der/den Forschungsstelle(n)



## Inhalt:

1	Abkürzungen und Begriffserläuterungen .....	4
2	WOAS-Ergebnisse auf einem Blick .....	5
3	Problemstellung und Forschungsziel.....	6
3.1	Ausgangssituation .....	6
3.2	Problemstellung und Stand der Technik .....	7
3.3	Forschungsziel .....	8
4	Wissenschaftlich-technische Ergebnisse.....	9
4.1	Patentrecherche .....	9
4.1.1	Recherche: Geräteintegration.....	9
4.1.2	Recherche: Dienste .....	10
4.2	Umfrage bei den PA-Mitgliedern .....	11
4.3	Prozessdatenzugang zu Automatisierungsgeräten .....	12
4.3.1	Virtual Devices im WOAS-System .....	12
4.3.2	Allgemeine Struktur eines VD .....	14
4.3.3	Datenmodell eines VD .....	15
4.3.4	Aufrufchnittstelle des VD-Basisobjekts .....	19
4.3.4	Testumgebung für ein VD .....	21
4.3.5	VD-Beispiele für WOAS .....	22
4.3.6	WOAS Device Protocol.....	23
4.3.7	Zeitverhalten .....	24
4.4	Automatisierungsfunktionen als Dienste .....	29
4.4.1	Dienste im WOAS-System.....	29
4.4.2	Allgemeine Struktur eines WOAD.....	31
4.4.3	Datenmodell eines WOAD .....	32
3.4.4	Aufrufchnittstelle des WOAD-Basisobjekts .....	36
4.5	WOAS-Demonstratoren .....	38
4.5.1	WOAS-Demonstrator I .....	38
4.5.2	WOAS-Demonstrator II .....	43
4.5.3	Dienste .....	44
4.6	WOAS-Integrationsportal .....	47
4.6.1	Aufbau der Benutzeroberfläche .....	47
4.6.2	Verwendung und Betrieb .....	49
4.7	Mieten von Automatisierungsdiensten .....	53
4.7.1	Das Projekt WOAS als CPS-Integrationssystem.....	54
4.7.2	Wohldefinierte Schnittstellen als Basis eines smarten Geschäftsmodells .....	55
4.7.3	Das WOAS-Clearing-System.....	57
4.7.4	Geschäftsmodelle auf Basis eines Mietszenarios .....	58
5	Projektergebnisse nach Arbeitspaketen .....	60
5.1	Zusammenfassende Ergebnisse nach Arbeitspaketen .....	60
5.2	Erfüllungsgrad der Arbeitspakete .....	62
6	Veröffentlichungen und Transfermaßnahmen .....	63
6.1	Veröffentlichungen in Fachzeitschriften .....	63
6.2	Veröffentlichungen auf Konferenzen .....	64
6.3	Presseinformationen über WOAS .....	64
6.4	Workshops und Seminare .....	64
6.5	Messen .....	65
6.6	Transfermaßnahmen per Internet, Flyer u.a. ....	65
6.7	Arbeitskreise, Fachausschüsse .....	65
7	Nutzen für KMUs .....	65
8	Transferkonzept .....	66
8.1	Einzelmaßnahmen .....	66
8.2	Strategische Maßnahmen: .....	67

# 1 Abkürzungen und Begriffserläuterungen

AS	<b>AutomatisierungsSystem</b>
AG:	<b>AutomatisierungsGerät</b> , welches an ein nutzerspezifisches WOAS über ein IP-Netzwerk angeschlossen ist.
AF	<b>AutomatisierungsFunktion</b> . Ein WOAS muss unterschiedliche AF realisieren. Dazu gehören z.B. HMI- und SCADA-Funktionen, Prozessdaten-Verarbeitungs- und Steuerungsfunktionen, Datenarchivierung und -auswertung, aber auch die Verbindung von Prozessdatenereignissen mit Email- und SMS-Systemen für Alarmierung und Diagnose.
PD:	<b>ProzessDaten</b> . Prozessdaten sind Signale von Sensoren und Aktoren sowie Prozessparameter komplexerer Geräte wie z.B. Steuerungen, Messsysteme. WOAD erhalten die Prozessdaten grundsätzlich über die Channels von VDs. Dabei gilt, dass Sensordaten (Prozessausgangssignale) über einen ereignisbasierten VD-Output-Channel und Aktordaten (Prozesseingangssignale) über einen VD-Input-Channel asynchron gelesen bzw. asynchron geschrieben werden.
VD:	Ein <b>Virtual Device</b> ist ein Software-Objekt, welches die Eigenschaften eines AGs für ein nutzerspezifisches WOAS abbildet. Es kann direkt im WOAS-Kern oder als zulinkbare Dateien realisiert sein. Jedes VD besitzt zur Verbindung mit dem WOAS-Kern ein einheitliches Objektinterface sowie zur Verbindung mit dem AG eine WOAS-Device-Protokollschnittstelle oder eine integrierte Industrieschnittstelle. Eine WOAS View (Webseite) kann n Instanzen von VDs beinhalten
WDG	Als <b>WOAS Device Gateway</b> wird eine externe Hardware bezeichnet, die ein oder mehrere WPGs realisiert. Kommt dabei ein IPC zum Einsatz können auch weitere erforderliche Geräte-/Protokolltreiber (z.B. OPC DA-Server) auf dem WDG installiert sein.
WOAD	<b>Web-Orientierter AutomatisierungsDienst</b> . Ein WOAD bildet eine Automatisierungsfunktion als Dienst innerhalb des WOAS-Systems in Form eines Software-Objekts ab und interagiert i.d.R. mit VDs.
WOAS:	<b>WebOrientiertes AutomatisierungsSystem</b> . Ein WOAS wird nutzerspezifisch in einem WOAS-Portal mittels eines WOAS-Creators erzeugt und besteht aus n WOAS Views (n Webseiten)
WOAS-DP	Das <b>WOAS Device Protocol</b> dient zur Kommunikation zwischen einem VD und dem WPG bzw. zwischen einem VD und einem AG, wenn das AG das WOAS-DP versteht.
WPG:	Das <b>WOAS Protocol Gateway</b> realisiert die Umsetzung des WOAS Device Protocol in das erforderliche Industrieprotokoll zur Kommunikation mit dem realen AG. Das WPG kann direkt als JS-Protokollumsetzer Bestandteil des VD seins oder es ist auf einer externen Hardware (IPC, embedded Server) realisiert.

## 2 WOAS-Ergebnisse auf einem Blick

Mit dem Prototyp des WOAS-Portals steht erstmalig eine Konfigurations- und Runtime-Umgebung online und webbasiert zur Verfügung mit der weltweit verteilten Geräte (Automatisierungsgeräte) mit beliebigen verteilten Diensten (Automatisierungsdiensten) zu nutzerspezifischen Funktionsystemen integriert und für einen Testbetrieb genutzt werden können. Als weiteres Alleinstellungsmerkmal gegenüber allen anderen Cloud-basierten Integrationsystemen für die Automatisierung kann das WOAS-Portal durch beliebige Dienste Dritter und weitere Gerätezugänge über konsistente und offengelegte Schnittstellen erweitert werden.

Das WOAS-Portal ist unter den folgenden Webadressen öffentlich für Testzwecke erreichbar:

<http://woas.ccad.eu>

<http://woastest.ccad.eu>

Die Zugangsinformationen für einen Testbereich finden sich in den WOAS-Nutzerhinweisen, die von der Home-Page des WOAS-Portals heruntergeladen werden können.

Zusammengefasst wurden im Projekt folgende wissen.-technische Ergebnisse erzielt:

- Modellierung eines Web-basierten Gerätezugangs und Definition von Schnittstellen (Virtual Devices).
- Modellierung Web-basierter Automatisierungsdienste und Definition von Schnittstellen (WOAD).
- Implementierung von fünf Virtual Devices für unterschiedliche Zugänge zu Industrie-schnittstellen (OPC DA, Modbus TCP, SPS direkt) zur Evaluierung der WOAS-VD-Methodik.
- Implementierung von 8 unterschiedlichen Automatisierungsdiensten zur Evaluierung der WOAS-WOAD-Methodik.
- Durchführung von Zeitmessungen zur Web-basierten Übertragung von Prozessdaten über ein TCP-Netz und Bewertung.
- Modellierung einer mandantenfähigen und rollenbasierten Integrationsplattform (WOAS-Portal) zur nutzerspezifischen Konfiguration eines Automatisierungssystems.
- Implementierung eines Prototyps für ein WOAS-Portal zum Nachweis und Evaluierung der WOAS-Methodik.
- Ausrüstung von zwei Automatisierungsanlagen der Forschungsstelle (Montagestation für Modellautos, Bearbeitungs- und Teststation) mit Komponenten der Mitglieder des PA als WOAS-Demonstratoren und öffentliche Bereitstellung des Zugangs über das WOAS-Portal für Testzwecke.
- Entwicklung eines Clearing-Konzepts zur Vermietung von Automatisierungsdiensten als Basis für neue Geschäftsmodelle.

Bezogen auf die Ziele im Projektantrag wurde das Forschungsziel erreicht.

### 3 Problemstellung und Forschungsziel

#### 3.1 Ausgangssituation

Im Gegensatz zur klassischen Verwendung des Begriffs „Automatisierungssystems“ wird im WOAS-Projekt das Gesamtsystem zur Automatisierung eines technischen Prozesses in die beiden Teile

- Automatisierungssystem (AS) und
- Feldsystem (FS)

geteilt (Bild 1).

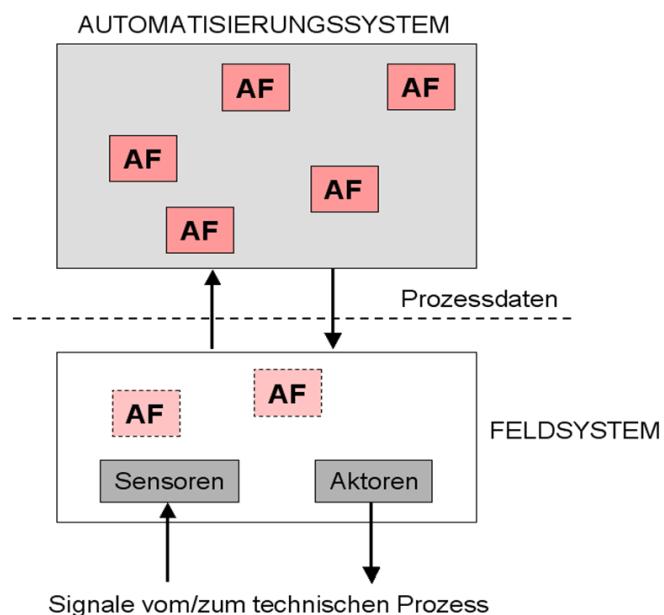


Bild 1: Trennung in Automatisierungs- und Feldsystem

Das *Automatisierungssystem* besteht im wesentlichen aus Software-Komponenten, die auf Rechnern unterschiedlicher Ausprägung verteilt sein kann. Es lässt sich in eine definierte Anzahl von Automatisierungsfunktionen (AF) aufteilen. Gegenüber dem Automatisierungssystem präsentiert sich der technische Prozess ausschließlich über die vom Feldsystem bereitgestellten Prozessdaten. Diese

Prozessdaten werden im Automatisierungssystem innerhalb verschiedener AF verarbeitet. Für das Automatisierungssystem ist ein „weiches“ Echtzeitverhalten charakteristisch. Typische Beispiele für Automatisierungssysteme sind Prozessvisualisierungs-, HMI- und Leitsysteme.

Das *Feldsystem* arbeitet mit Prozesssignalen und verarbeitet diese zu Prozessdaten. Es beinhaltet alle erforderlichen Sensoren und Aktoren und kann darüber hinaus auch Automatisierungsfunktionen wie z.B. Steuerung und Regelung beinhalten. Das Feldsystem muss üblicherweise ein „hartes“ Echtzeitverhalten gewährleisten. Typische Feldsysteme sind SPS- und CNC-Steuerungssysteme.

Im Sinne der hier verwendeten Begriffsdefinition könnte ein Automatisierungssystem aber auch alle erforderlichen Automatisierungsfunktionen beinhalten, so dass das Feldsystem dann nur noch aus Sensoren/Aktoren mit Signalaufbereitung besteht. Dies hängt von der jeweiligen maximalen Reaktionszeit des zu automatisierenden technischen Prozesses sowie vom konkret verwendeten Kommunikationsnetz (z.B. Echtzeit-Ethernet, Standard-TCP/IP) ab.

## 3.2 Problemstellung und Stand der Technik

Aktuelle Automatisierungssysteme wie z.B. Prozessvisualisierungssysteme, SCADA<sup>1</sup>/HMI<sup>2</sup>-Systeme und Prozessleitsysteme sind proprietäre und herstellerspezifische Systeme, die mit klassischen Programmiersprachen erstellt sind und meist herstellerspezifische interne Schnittstellen und Datenformate aufweisen (Beispiele: WinCC, Genesis, InTouch). Eine für Engineering-Zwecke geeignete konsistente und offene informationstechnische Modellierung eines AS liegt nicht vor, da die Nutzung der proprietären Formate, Schnittstellen und Methoden dieses kaum zulassen.

Unter dem Stichwort Internettechnologie für die Automatisierung werden seit einigen Jahren verstärkt bekannte Standardtechniken aus der Welt der Informationstechnologie für AS verfügbar gemacht und zunehmend eingesetzt. Dazu gehören insbesondere

- die Nutzung des Webbrowsers als universelle Bedienschnittstelle für automatisierte Geräte, Maschinen und Anlagen,
- der Einsatz von Ethernet und TCP/IP als Kommunikationsstruktur zwischen Elementen des AS sowie zwischen AS und FS,
- die Nutzung von XML als Beschreibungssprache für Protokolle, Dienste und Komponenten.

Die Zielstellung der bekannten F/E-Arbeiten ist

- eine wesentliche Verbesserung der horizontalen und vertikalen Kommunikation und Datenintegration in Automatisierungssystemen,
- eine Reduzierung der Schnittstellenprobleme durch Vereinheitlichung und Nutzung von informationstechnischen Standard sowie damit verbunden
- eine Erhöhung von Flexibilität und Effizienz.

Die meisten bekannten Arbeiten setzen dazu die Web-Technologien partiell, herstellerspezifisch und/oder begrenzt offen ein. Selbst Systeme, die durchgängig Webtechnologien für die Funktionsrealisierung einsetzen, verbleiben in der klassischen Server-Client-Struktur und lassen weder eine verteilte Strukturierung des AS noch die Nutzung verteilter Dienste zu. Web 2.0-Technologien werden bisher nur wenig berücksichtigt, neuere Technologien (wie z.B. MashUp, REST, Ruby on Rails, Google Native Client) überhaupt nicht. Eine auf Basis von Web-Technologien mögliche Modellierung eines AS sowie eine entsprechende Engineering-Umgebung steht nicht zur Verfügung. Damit ergeben sich auch keine prinzipiellen Änderungen in der Struktur der Automatisierungssysteme und auch die Zielstellungen sind nur unvollständig realisierbar.

Ein ausführlicher Bericht zum Stand der Technik findet sich im Projektantrag. Während der Laufzeit des Projekts hat sich bezogen auf die Projektzielstellung trotz international verstärkter Bemühungen zum Einsatz der Internettechnologie in der Automation daran auch kaum etwas geändert. Die Ergänzungen zum im Projektantrag aufgeführten Stand der Technik finden sich in der ausführlichen Patentrecherche, die während der Projektdurchführung realisiert wurde (s. Kapitel 4.1).

Der Forschungsbedarf für das WOAS-Projekt Antrag ergibt sich damit zusammengefasst aus den folgenden Nachteilen des Standes der FuE:

---

<sup>1</sup> SCADA – Supervisor, Control and Data Aquisition

<sup>2</sup> HMI - Human Machine Interface

- Es fehlt eine durchgängige, offene und konsistente Modellierung eines AS auf Basis von webbasierten-Standardtechnologien.
- Webtechnologien werden zwar eingesetzt, die flexible Verteilung von Struktur und Funktion eines Automatisierungssystems werden aber nicht genutzt.
- Neue Web 2.0-Technologien wie MashUps und REST werden für AS bisher nicht eingesetzt.
- Weborientierte Architekturen (WOA) über IP-Netze für AS als Zukunft einer webbasierten Automatisierungsdienste-Industrie stehen nicht zur Verfügung.
- Verfügbare Standardtechnologien aus der Welt der IP-Netze für die Erhöhung von Flexibilität und Effizienz sowie durchgängige und konsistente Kommunikation für AS werden nur unzureichend genutzt.

### 3.3 Forschungsziel

Zielstellung des Forschungsvorhabens war es, eine neue Architektur für Automatisierungssysteme auf Basis von Web-Technologien zu erforschen. Diese Architektur wird in Anlehnung an den aus dem IT-Bereich bekannten technologischen Ansatz einer weborientierten Architektur (WOA) als *weborientiertes Automatisierungssystem (WOAS)* bezeichnet. Ein WOAS besteht aus einem WOAS-Kernel sowie einer konfigurierbaren Anzahl von *weborientierten Automatisierungsdiensten (WOAD)*, die die erforderlichen Automatisierungsfunktionen wie z.B. Prozessdatenvisualisierung, Datenerfassung, Bedienung realisieren.

Die im IP-Netz verteilten WOAD sowie ein WOAS-Kernel werden durch verschiedene Instanzen (Server bzw. Cloud) im Netz verwaltet. Sie werden erst zur Betriebszeit des Automatisierungssystems in einen WOAS-Client (= Standard-Webbrowser) geladen und dort ausgeführt. Ein WOAS kann mit beliebig vielen Feldsystemen und beliebig vielen Diensten gleichzeitig zusammenarbeiten. Die Bereitstellung der Prozessdaten des Feldsystems für das WOAS erfolgt unter Nutzung geeigneter Web-Technologien (z.B. mittels XMLHttpRequest, Webanwendungen) über einen Webserver des Feldsystems.

Als weitere Zielstellung sollten neue Geschäfts- bzw. Betreibermodelle erforscht werden, die aufbauend auf WOAS/WOAD eine hocheffiziente Nutzung von in IP-Netzen verteilten Automatisierungsfunktionen ermöglichen.

Mit der WOAS/WOAD-Methodik soll ein konsistentes Architekturmodell für weborientierte Automatisierungslösungen geschaffen werden, welches die Grundlage für die Generierung konkreter in einem IP-Netz verteilter AS bildet. Auch wenn als Beispieldaten in der Projektbeschreibung verschiedentlich zur Illustration einer praktischen Anwendung auf Prozessvisualisierungs-/HMI-Systeme Bezug genommen wird, kann ein WOAS prinzipiell jede mögliche AF oberhalb der Feldebene realisieren (s. Bild 1). Abhängig ist dies nur von den jeweiligen konkreten Anforderungen an die AF. Aufgrund der zu entwickelnden Modellnotation und darauf aufzubauender Engineering-Werkzeuge behandelt ein WOAS den gesamten Lebenszyklus eines AS, von der werkzeuggestützten Generierung über den Betrieb bis hin zu Pflege und Wartung. Der wissenschaftliche Anspruch des Projekts bestand in der Erforschung dieser neuen weborientierten verteilten Architektur für AS.

Das Forschungsvorhaben setzte auf den bekannten Arbeiten des Antragstellers zur „Lean Web Automation“ auf und setzte den dort eingeschlagenen Weg methodisch fundiert fort bis hin zu einem weborientierten Architekturkonzept für Automatisierungssysteme.

Die Voraussetzungen für das WOAS-Projekt bildeten ursprünglich (2010) folgende Aussagen nach der ZVEI-Roadmap 2015+:

- Die Bedeutung von IT-Standard-Technologien und dabei insbesondere der Technologien aus dem TCP/IP- und Web-Umfeld für die industrielle Automation wird weiterhin stark ansteigen. Zukünftig sind nur noch Standard-Technologien für die zunehmend komplexen Automatisierungssysteme wirtschaftlich effizient einsetzbar.
- IP-basierte Kommunikationsnetze werden in verschiedenen Ausprägungen einen sehr hohen Datendurchsatz erreichen, der auch für Echtzeitkommunikation geeignet sein kann.
- Elemente des Feldsystems (z.B. Sensoren, Aktoren, Steuerungen) besitzen zunehmend IP-basierte Kommunikationsanschlüsse sowie integrierte Webserver und lassen sich über Webbrowser konfigurieren, ansprechen und z.T. auch betreiben.

Durch die 2012 veröffentlichten Umsetzungsempfehlungen der Forschungsunion für das Zukunftsvorhaben Industrie 4.0 wurden die WOAS-Zielstellungen in vollem Umfang in ihrer Bedeutung bestätigt. Das Projekt WOAS kann aus heutiger Sicht deshalb auch als ein frühes Industrie 4.0-Projekt eingeschätzt werden.

## 4 Wissenschaftlich-technische Ergebnisse

### 4.1 Patentrecherche

#### 4.1.1 Recherche: Geräteintegration

##### Rechercheumfang

Eine Patentrecherche zur Geräteintegration wurde in drei Phasen durchgeführt:

- Recherche in vorher ausgewählten IPC-Klassen,
- Recherche nach Stichwörtern,
- Recherche in neu ermittelten IPC-Klassen.

Die Patentschriften wurden zuerst nach Bezeichnungen sortiert, dann wurde der Abstract bewertet und schließlich die Patentbeschreibung ausgewertet, um die endgültige Relevanz der Patentschrift zu bestimmen. Die Patente in den Klassen wurden zurück bis in das Jahr 2010 durchsucht. Kleinere oder für WOAS besonders relevante Klassen wurden vollständig bzw. zumindest zurück bis zum Jahr 2005 durchsucht.

Durchsuchte Begriffe:

*Device Integration, Gerätintegration, Device configuration, Field device, Automation Web, Cloud Computing, Web services, Multi Agent System, Rockwell, Beckhoff, Siemens, Schneider Electric, Honeywell*

Das Ergebnis der Recherche besteht aus 17 gefundenen Patenten, davon beziehen sich sechs auf das Thema der Geräteintegration und 11 andere auf den Einsatz von Webtechno-

logien. Besonders relevant sind drei Patente, die im Projekt zu berücksichtigen waren. Ausführliche Angaben und Bewertungen zu den gefundenen Patenten befinden sich in [1].

Zusätzlich zur eigentlichen Patentrecherche wurden Klassen und Gruppen zusammengestellt und deren Relevanz bestimmt.

### Rechercheergebnisse

**Tabelle 1:** Wichtige relevante Patente zur Geräteintegration

IPC	Nr.	Jahr	Titel	Inhaber
G05B 19/04	DE 102010029952A1	2010	Verfahren zum Integrieren von zumindest einem Feldgerät in ein Netzwerk der Automatisierungstechnik	Endress + Hauser
G05B 19/042	EP 1906274A3	2010	Webbasiert Konfigurationsserver für Automatisierungssysteme	Rockwell
G05B 19/418	WO002010043629A1	2009	VERFAHREN ZUR ENTWICKLUNG EINES MULTI-AGENTEN-SYSTEMS SOWIE MULTI-AGENTEN-SYSTEM	Schneider Electric Automation

### **4.1.2 Recherche: Dienste**

#### Rechercheumfang

In der Regel wurden die Klassen bis zum Jahr 2009 durchsucht, bei kleinen oder für das Projekt besonders interessanten Klassen auch bis zum Jahr 2000. Insgesamt wurden 8502 Patenttitel und zum größten Teil auch die Kurzbeschreibungen ausgewertet. Dabei wurden 38 relevante Patentschriften gefunden, wobei sich aufgrund der Nebenklasseneintragungen sich die Zahl auf 21 reduziert. Der vollständige Bericht dazu beinhaltet [2].

Es wurden folgende Klassen untersucht:

G05B 19/042, G05B 19/05, G05B 19/418, G05B 23/02, G06F 03/01, G06F 03/048, G06F 9/46, G06F 9/50, G06F 13/00 S, G06F 13/00 I, G06F 13/14, G06F 13/368, G06F 15/16, G06F 15/173, G06F 15/76, G06F 17/00, G06F 17/30, G06F 21/00, H04L 12/56, H04L 29/06 H04L 29/08

### Rechercheergebnisse

Keines der gefundenen Patente ähnelt WOAS so stark, dass dessen Umsetzung aus patentrechtlichen Gründen nicht möglich wäre, allerdings gibt es einige Patente, die bestimmte Techniken und Ideen mit WOAS teilen. Dazu gehören insbesondere die in Tabelle 2 aufgelisteten Patentschriften.

**Tabelle 2:** Wichtige relevante Patente zu Automatisierungsdiensten

IPC	Nr.	Jahr	Titel	Inhaber
G05B 19/042	WO0020100998 02A1	2009	VERFAHREN UND VORRICHTUNG ZUR BE-REITSTELLUNG VON DATEN IN EINEM AUTOMATISIERUNGSSYSTEM	Siemens
G05B 19/418	WO0020100436 29A1	2009	VERFAHREN ZUR ENTWICKLUNG EINES MULTI-AGENTENSYSTEMS SOWIE MULTI-AGENTEN-SYSTEM	Schneider Electric Automation

G06F 15/16	US00000797083 0B2	2009	Cloud computing for an industrial automation and manufacturing system	Honeywel
G06F 17/30	US02011015359 0A1	2010	APPARATUS AND METHOD FOR SEARCHING FOR OPEN API AND GENERATING MASHUP BLOCK SKELETON CODE	Korea Electronics Telekom

Alle vier Patente aus Tabelle 2 beschreiben eine Art Mash-Up-System, welches (meist) Automatisierungsfunktionen als Software Orientierte Architektur darstellt. Es gibt allerdings kein Patent, welches dabei auch von *Software as a Service* spricht. Diese Idee scheint als einzige bisher nicht (für industrielle Zwecke) patentiert worden zu sein. Ungeachtet dessen, sollten aber bei einer Produktumsetzung von WOAS auch die anderen in Tabelle 2 aufgeführten Patente berücksichtigt werden, da diese meist wichtige Teilfunktionen des WOAS zumindest teilweise behandeln.

#### Berichte und Dokumente:

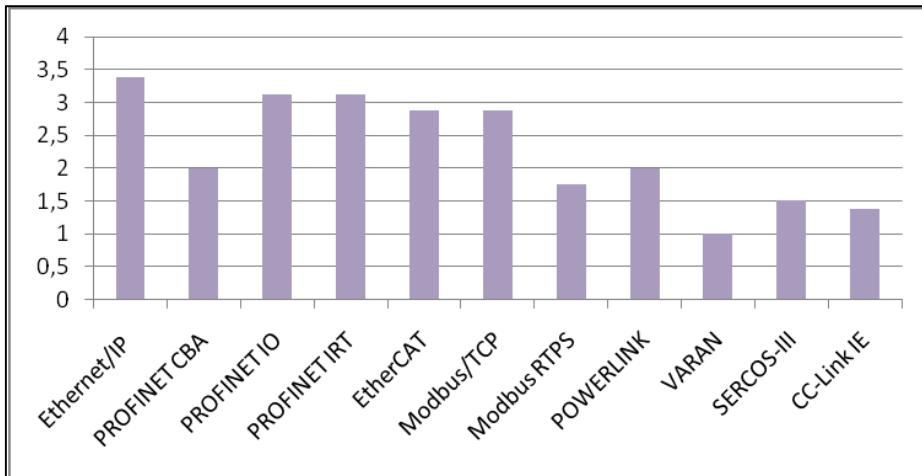
- [1] Makarov O.: Patentrecherche: Web-Orientiertes AutomatisierungsSysteme. – Feb. 2012
- [2] Meyer, L.: Patentrecherche WOAS-Projekt. – 14.02.12

## 4.2 Umfrage bei den PA-Mitgliedern

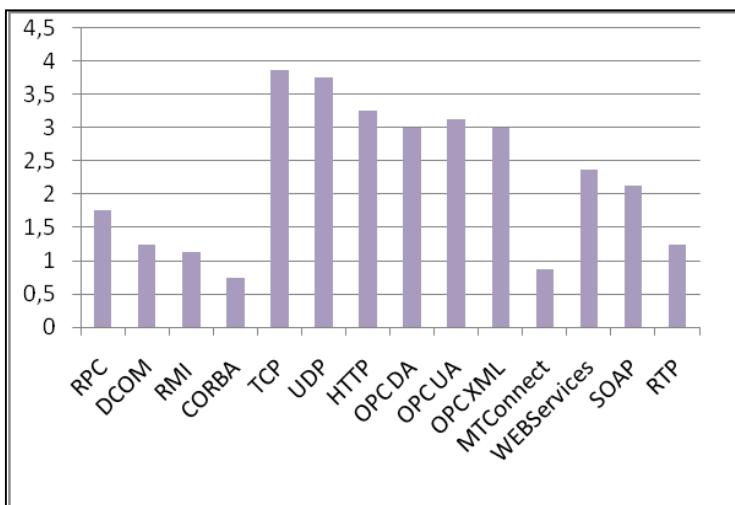
Nach Projektstart 2011 erfolgte zur Berücksichtigung der Anwenderinteressen im Projekt ein Umfrage bei den PA-Mitgliedern mit folgenden Fragestellungen:

1. Welche Wichtigkeit auf einer Skala von 1 ...4 besitzen die folgenden Automatisierungsfunktionen für Sie?
2. Welche Wichtigkeit besitzen die folgenden Nutzungseigenschaften eines Automatisierungssystems nach WOAS-Definition für Sie?
3. Welche Wichtigkeit besitzen die folgenden Feldbussysteme für Sie?
4. Welche Wichtigkeit besitzen die folgenden Echtzeit-Ethernetsysteme für Sie?
5. Welche Wichtigkeit besitzen die folgenden Geräteschnittstellen für Sie?
6. Welche Wichtigkeit besitzen die folgenden wireless Schnittstellen für Sie?
7. Welche Wichtigkeit besitzen die folgenden Protokollschnittstellen für Sie?
8. Welche der genannten Web-Technologien betrachten sie als besonders perspektivisch für das WOAS-Projekt (soweit Ihnen die entsprechende Technologie bekannt ist)?
9. Welche Wichtigkeit besitzen die folgenden Kommunikationsstandards für Sie?
10. Welche der aufgeführten Web-Server (soweit bekannt) würden Sie gern im Projekt einsetzen?
11. Welche weiteren Empfehlungen haben Sie aus Sicht eines Technologieeinsatzes für das Projekt?

Bild 2 und Bild 3 zeigen als Beispiel die Auswertung für die Fragen 4 und 7.



**Bild 2:** Wichtigkeit von Echtzeit-Ethernet-Systemen für die Mitglieder des PA



**Bild 3:** Wichtigkeit von Protocolschnittstellen für den PA

Aufgrund der Einschätzungen nach Bild 2 und 3 wurden z.B. als Gerätezugänge OPC DA und Modbus TCP im Projekt realisiert.

Die vollständige Auswertung findet sich in den ausführlichen Projektdokumenten [3] und [4].

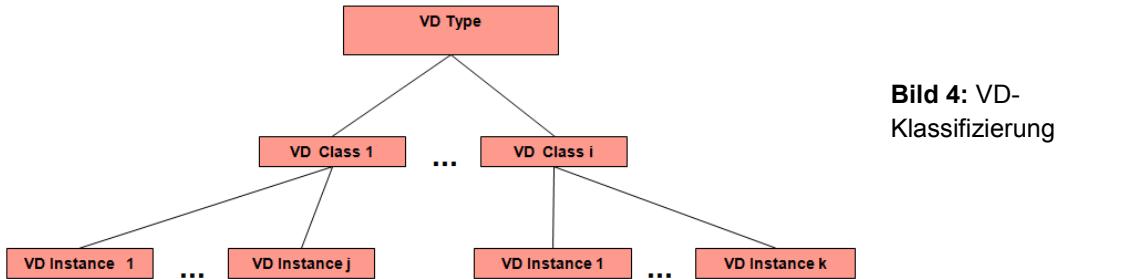
#### Berichte und Dokumente:

- [3] CCAD: Fragebogen WOAS. - 20.11.11
- [4] CCAD: Auswertung der Fragen (Excel-Auswertung). – 28.03.12

### 4.3 Prozessdatenzugang zu Automatisierungsgeräten

#### 4.3.1 Virtual Devices im WOAS-System

Zur Anpassung an unterschiedliche AGs sowie Infrastrukturbedingungen werden in WOAS verschiedene Arten von VDs eingeführt. Es ergibt sich damit die in Bild 3 dargestellte VD-Klassifizierung.



Die VD-Elemente der Klassifizierung nach Bild 4 sind entsprechend Tabelle 3 zu verstehen.

**Tabelle 3:** Elemente der VD-Klassifizierung

Klassifizierungselement	Beschreibung
VD Type	Charakterisiert die prinzipielle Struktur eines WOAS-VD.
VD Class	Charakterisiert die jeweilige allgemeine implementierungstechnische Umsetzung.
VD Instance	Ein WOAS-Nutzer mit der Rolle <i>Admin</i> kann im WOAS-System verfügbare VD Classes parametrieren und erhält dann konkrete VD Instances.

Aktuell werden drei VD-Typen unterschieden (Bild 5), die sich insbesondere durch die Art und Weise ihrer Integration in ein WOAS-System unterscheiden:

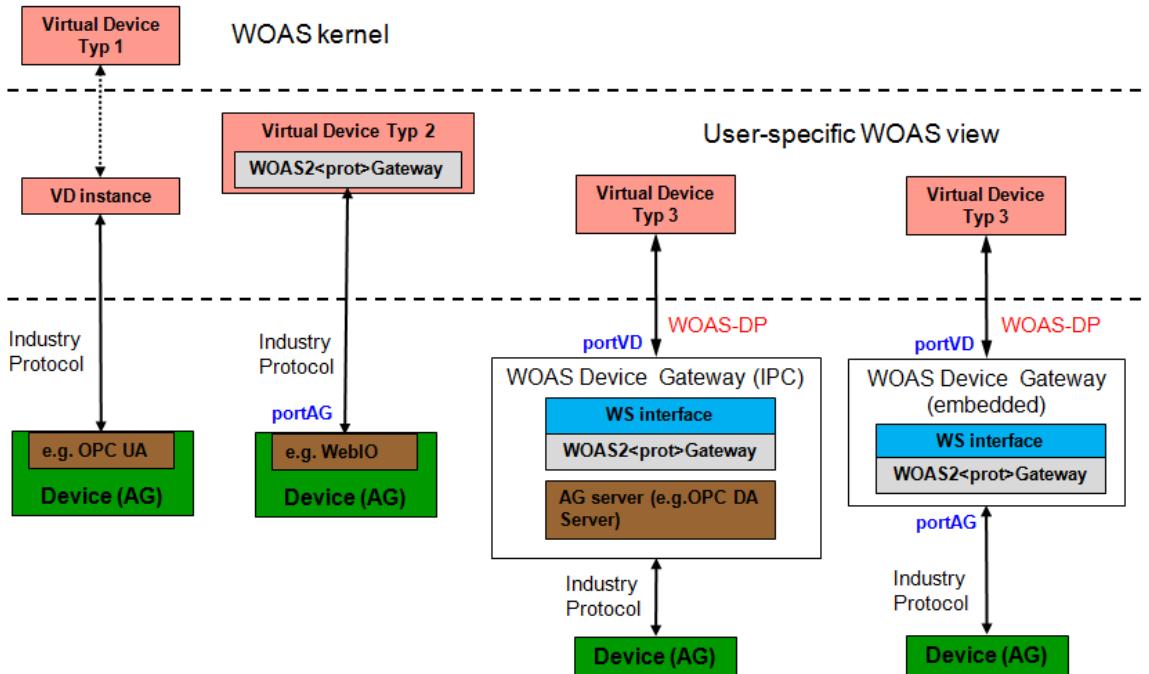
- **VD Type 1:** Diese VDs sind bereits als Plugin im WOAS-Kern realisiert (internal VD) und setzen die Kernaufrufe direkt in die erforderlichen Geräte-Protokollinformationen um. VDs vom Typ 1 können nur durch den WOAS-Provider in das System integriert werden.
- **VD Type 2:** Das VD beinhaltet ein separates WPG in JavaScript (JS), welches die Kernaufrufe in das erforderliche Industrieprotokoll/Geräteschnittstelle umsetzt. Das WOAS-DP kann hierbei zur internen Kommunikation auf Softwareebene genutzt werden. VDs vom Typ 2 können durch Drittanbieter (Manufacturer) in das System integriert werden.
- **VD Type 3:** Das VD kommuniziert mit einem externen WOAS Device Gateway, welches i.d.R. über WebSockets (WS)/TCP mit dem VD verbunden ist und über das WOAS-DP Nachrichten austauscht. Für die Kommunikation mit dem externen WDG können aber auch andere Protokolle genutzt werden. VDs vom Typ 3 können durch Drittanbieter (Manufacturer) in das System integriert werden.

Im WOAS-Demoportal sind bereits die in Tabelle 4 aufgeführten VD Classes integriert.

**Tabelle 4:** VD Classes im WOAS-Demoportal

Name der Class	Beschreibung	Type
VD4Vathauer	Server push über Websockets auf ein proprietäres TCP-Protokoll von MSF Vathauer (Ansteuerung von Frequenzumrichtern).	1
VD4PHP2OPCDA	Client push mittels PHP auf OPC DA. Das VD befindet sich auf dem lokalen WOAS-Server oder auf einem entfernten Server.	2
VD4PLC	Server push über Websockets direkt in eine SPS. Das VD greift direkt auf einen IEC 61131-Funktionsbaustein in der SPS zu.	2
VD4MODBUS	Server push über Websockets auf Modbus TCP. Das VD ist in einem embedded Device realisiert.	3
VD4WPG	Server push über Websockets auf OPC DA. Das VD kommuniziert mit	3

	einem externen WDG über das WOAS-DP.	
VD4WAK	Server push über einen konfigurierbaren TCP-Kanal auf OPC DA. Das VD kommuniziert mit einem externen WDG über ein proprietäres Protokoll (WAK).	3



**Bild 5:** Die drei VD-Typen im WOAS-System

#### 4.3.2 Allgemeine Struktur eines VD

Ein VD besteht aus einer JS-Datei, einer Bilddatei sowie optional einem Dateiverzeichnis, in dem speziellen Dateien, die für die Realisierung eines Prozessdatenzugriffs für das jeweilige VD erforderlich sind, abgelegt sind. Tabelle 5 zeigt dazu eine Übersicht.

**Tabelle 5:** Bestandteile eines Virtual Device in WOAS

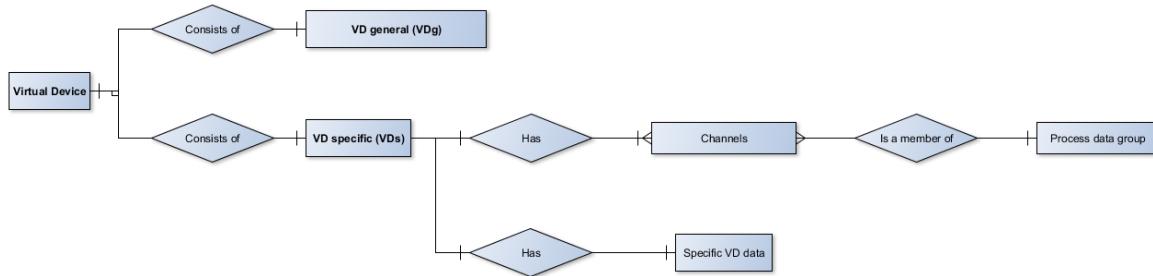
Dateiname	Erforderlich	Beschreibung
<class>.js	ja	Die JS-Datei enthält die Implementierung des entsprechenden VD-Objekts (VD class)
icon_140.png	ja	Icon für dieses VD mit Abmessungen 140 x 140
<name_1>/ <name_2>/ ... <name_N>/	optional	Je nach Realisierungsart kann ein VD 1 ..n Ordner mit spezifischen Dateien (Applets, php-Dateien usw.) benötigen

Als VD Class-Name sollte ein aussagefähiger Kurzname gewählt werden z.B. VD4PHP2OPCDA oder VD4WPG. Für die Implementierung eines VD wird der Einsatz des JS-Frameworks *jQuery* empfohlen. *jQuery* steht im WOAS-Kern bereits zur Verfügung. Sollen andere JS-Frameworks genutzt werden, müssen diese in das VD über den optionalen Ordner mit eingebunden werden.

Nach dem Einloggen in das WOAS-Portal kann ein User mit der Rolle *Manufacturer* die VD-Dateien in das Portal hochladen oder eine IP-Adresse angeben, über die die VD-Dateien erreichbar sind.

### 4.3.3 Datenmodell eines VD

VD Classes sowie die zugehörigen durch einen Nutzer parametrierten unterschiedlichen VD Instances werden im WOAS-System in einer Datenbank gespeichert und verwaltet. Die Datenstruktur eines VD ist entsprechend Bild 6 aufgebaut.



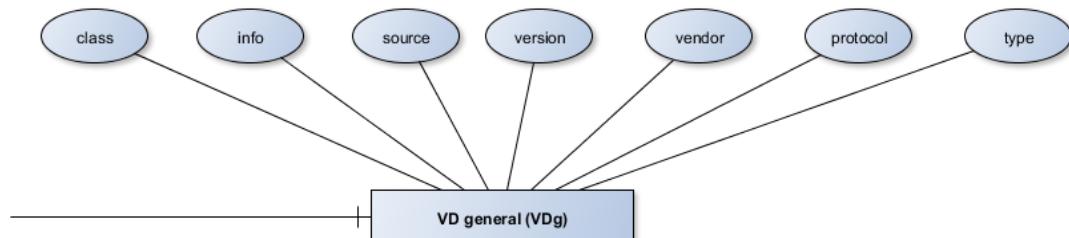
## Bild 6: Datenstruktur eines VD

Jedes VD besteht aus allgemeinen Daten (General Data) sowie spezifischen VD-Daten (Specific Data). Die General Data werden durch den Hersteller des VD (Manufacturer) vorgegeben und können auch nur durch einen WOAS-Nutzer mit der Rolle *Manufacturer* verändert werden. Die General Data charakterisieren die VD Class.

Die spezifischen VD-Daten charakterisieren die VD Instances und werden durch den WOAS-Nutzer mit der Rolle *Admin* erzeugt und parametriert.

## Allgemeine VD-Daten

Die allgemeinen VD-Daten (Bild 7) werden im Wesentlichen nur im Edit-Mode benötigt. Sie dienen der Beschreibung des jeweiligen Gerätezugangs. Sie werden in der Datenbank abgelegt und im VD-Edit-Mode aufgerufen, angezeigt und können nur durch den *Manufacturer* editiert werden.



**Bild 7:** General VD Data

Die einzelnen Daten sind in Tabelle 6 beschrieben.

**Tabelle 6:** General VD Date

Name	Beschreibung	Beispiel
class	Name der VD Class = Bezeichnung der JS-Datei zur Implementierung des VD	VD4PHP2OPCDA
info	Beschreibung der VD Class	Server-based PHP access to OPC DA
source	IP-Adresse des Hauptordners des VD	/WOAS/VD/VD4PHP2OPCDA/
version	Versionsnummer der Class	1.0
vendor	Hersteller des VD	CCAD
protocol	Protokollname zum Zugriff auf das AG	HTTP
type	Nummer des VD Type	2

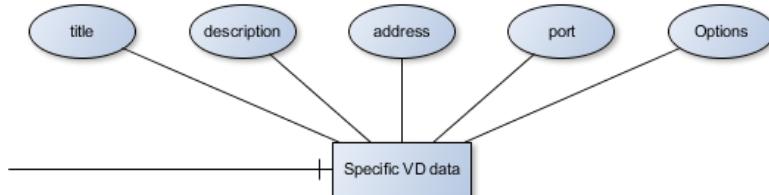
### Spezifische VD-Daten

Die spezifischen Daten eines VD teilen sich in

- *Specific VD Data*: Zugriffscharakteristik für das AG.
- *Specific VD Process Data Groups*: Diese Daten legen die Art und Weise der Prozessdateneignisse im Subscribe-Betrieb fest.
- *Channels*: Diese Daten beinhalten die zur Verfügung stehenden Prozessdaten des AG für die jeweilige VD Instance

In Bild 8 und Tabelle 7 sind die spezifischen VD-Daten im Detail erläutert.

#### *Specific VD Data:*



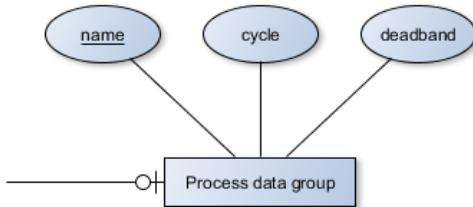
**Bild 8:** Specific VD Data

**Tabelle 7:** Specific VD Data

Name	Beschreibung	Beispiel
title	Name der VD Instance	SimAGI
description	Kurzbeschreibung der VD Instance	Softing OPC Simulation Server
address	URL zum Zugriff auf das AG bzw. auf ein externes WDG oder interne Webanwendungen	php/
port	Port, der zum Zugriff auf das AG bzw WDG verwendet wird	80
options	Zusätzliche nutzerspezifische variable Anzahl von weiteren Specific VD Data	opcname: Softing.OPCToolboxDemo_ServerDA.1

### Specific VD Process Data Groups:

Prozessdatengruppen (Bild 9, Tabelle 8) bestimmen wie im Subscribe-Betrieb (Server Push) die einer VD Instance zugeordneten Prozessdaten an das VD übertragen werden. Allen Prozessdaten einer VD Instance kann immer nur eine Prozessdatengruppe zugeordnet werden. Die Verfügbarkeit von Gruppen hängt von der Implementierung der jeweiligen VD Class ab.



**Bild 9:** Specific VD Process Data Groups

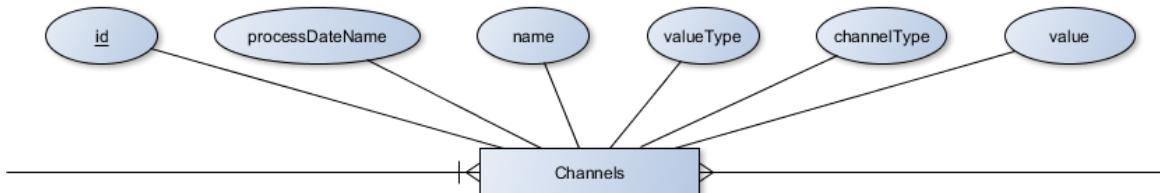
**Tabelle 8:** Specific VD Process Data Group

Name	Beschreibung	Beispiel
name	Name der Gruppe, wenn die VD Class Gruppen unterstützt	No groups
cycle	Zykluszeit in ms für die Benachrichtigung bei Prozessdatenänderungen	100
deadband	Erforderliche Prozessdatenänderung in %, um ein Ereignis zur Benachrichtigung auszulösen, wenn die VD Class dies unterstützt	0

### Channels:

Diese Daten (Bild 10) beinhalten die der VD Instance zugeordneten Prozessdaten. Im subscribe-Betrieb werden immer alle Prozessdaten dieser VD Instance abonniert und bei Änderungen auch client-seitig verarbeitet. Je nach WOAS View sollten deshalb angepasste VD Instances erstellt werden, damit nur die Prozessdaten verarbeitet werden, die auch tatsächlich für die jeweilige View benötigt werden.

Jeder VD Instance lassen sich beliebig viele Prozessdaten des jeweiligen AGs zuordnen.



**Bild 10:** Channels

Die Channels beinhalten die in Tabelle 9 aufgeführten Daten. Ein VD kann beliebig viele Channels beinhalten.

**Tabelle 9:** Daten eines Channels

Name	Beschreibung	Beispiel
id	Eindeutige numerische Identifikation des Channels	1
processDateName	Prozessdatenname entsprechend der Notation des zugehörigen realen Gerätes	maths.sin

name	Beschreibender Kurzname des Prozessdatums (Alias)	Sine
value	Wert des Prozessdatums	0.2079
valueType	Prozessdatentyp-Beschreibung entsprechend der OPC-Notation als Typnummer (s. Anlage 1)	17
channelType	0 – Prozessdatum ist nur lesbar 1 – Prozessdatum ist les- und schreibbar 2 – Prozessdatum ist nur schreibbar	0

### Editierbarkeit der VD Daten

Je nach VD Type können VD Data von Nutzern mit unterschiedlichen Rollen editiert werden oder sind über Profildateien festgelegt. Die folgende Tabelle zeigt dazu eine Übersicht.

**Tabelle 10:** Editierbarkeit von VD Data

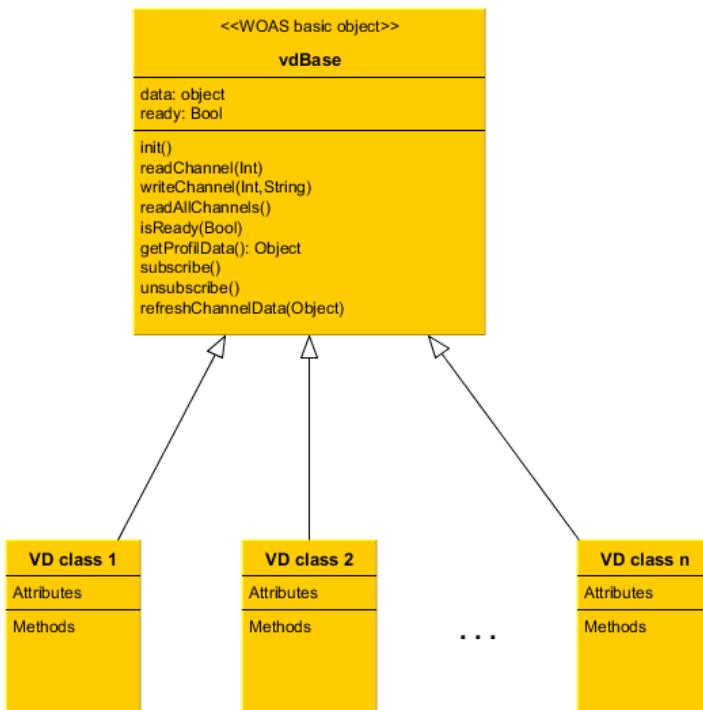
		VD Data	VD Type 1	VD Type 2	VD Type 3
General	VD Data	class	Manufacturer	Manufacturer	Manufacturer
		info	Manufacturer	Manufacturer	Manufacturer
		source	x	Admin	Admin
		version	Manufacturer	Manufacturer	Manufacturer
		Vendor	Manufacturer	Manufacturer	Manufacturer
		protocol	Manufacturer	Manufacturer	Manufacturer
		type	Manufacturer	Manufacturer	Manufacturer
Specific	VD Data	id	Admin	Admin	Admin
		title			
		description			
		address			
		port			
		options	Admin (falls vorhanden)	Admin (falls vorhanden)	Admin (falls vorhanden)
	VD Process Data Group	name	Admin	Admin	x
		cycle	Admin	Admin	x
		deadband	Admin	Admin	x
		id	Admin	Admin	Admin
Channels	Channels	processDateName	Admin	Admin	x
		name	Admin	Admin	x
		value	x	x	x
		valueType	Admin	Admin	Admin
		channelType	Admin	Admin	Admin

In Tabelle 10 bedeuten:

- x: Wert ist nicht editierbar, sondern muss aus der WOAS-Profil-Datei ausgelesen werden bzw. ändert sich dynamisch.
- Manufacturer: Wert kann nur durch einen Nutzer mit der Rolle *Manufacturer* editiert bzw. festgelegt werden.
- Admin: Wert kann nur durch einen Nutzer mit der Rolle *Customer-Admin* editiert werden.

#### 4.3.4 Aufrufsschnittstelle des VD-Basisobjekts

Die JavaScript-Datei <class>.js beinhaltet die Implementierung der jeweiligen VD Class. Diese VD Class erweitert ein bereits im WOAS-Kernel verfügbares Meta-Objekt vdBase und belegt dessen Attribute bzw. überschreibt die zugehörigen Methoden. Durch die Nutzung eines Meta-Objekts für alle WOAS-VD wird ein einheitliches VD-Interface als JS-Aufrufsschnittstelle geschaffen. Das WOAS-Portal kann deshalb jederzeit mit weiteren VD für andere Industrieprotokolle durch einen Dritten (Manufacturer) in konsistenter Art und Weise erweitert werden. Bild 11 zeigt das zugehörige Klassendiagramm für ein VD.



**Bild 11:** VD-Klassendiagramm

#### Attribute

Das Basisobjekt besitzt die folgenden in Tabelle 11 aufgeführten Attribute.

**Tabelle 11:** Attribute des vdBase –Objekts

Name	Beschreibung
data	Beinhaltet das VD-Datenobjekt
ready	Zeigt an, ob das VD für eine Prozessdatenübertragung zum realen Gerät bereit ist.
this.hasSubscriber	Wird im WPOAS-Portal benötigt und über das Basis-Objekt vdBase verwendet.

#### Methoden

Über die Methoden des Basisobjekts können Prozessdaten vom AG gelesen und geschrieben werden. Die wesentlichen Methoden des Basisobjekts, die für eine Testumgebung benötigt werden, sind in Tabelle 12 aufgelistet und beschrieben.

**Tabelle 12:** Methoden des vdBase –Objekts

Name	Parameter	Beschreibung
init	-	Initialisiert die VD Instance auf der Webseite, liest alle Prozessdaten dieses VD und startet die Subscribe-Methode
isReady	status: BOOL	Diese Funktion wird durch das VD aufgerufen, wenn das VD zur Prozessdatenübertragung zum realen Gerät bereit ist.
readChannel	channel:INT Identifier des zu lesenden Prozessdatums	Asynchrones Lesen eines Prozessdatums
writeChannel	channel, value channel:INT Identifier des zu lesenden Prozessdatums value: STRING Wert des zu schreibenden Prozessdatums	Asynchrones Schreiben eines Prozessdatums
readAllChannels	-	Asynchrones Lesen aller diesem VD zugeordneten Prozessdaten
subscribe	-	Abonnieren der VD-Prozessdaten für die ereignisbasierte Übertragung
doSubscribe	Abonnieren der Prozessdaten, die das Basisobjekt vdBase realisiert. Das VD muss diese Methode nutzen, um im WOAS-Portal funktionsfähig zu sein. Beinhaltet in der Testumgebung eine spezielle Implementierung und ruft die Methode subscribe auf	
unsubscribe	-	Stoppen der ereignisbasierten Übertragung von Prozessdaten
doUnsubscribe	Stoppen des Polling-Zyklus der Prozessdaten, die das Basisobjekt vdBase realisiert. Das VD muss diese Methode nutzen, um im WOAS-Portal funktionsfähig zu sein. Beinhaltet in der Testumgebung eine spezielle Implementierung und ruft die Methode unsubscribe auf	
refreshChannelData	data data:Object Objekt mit den Komponenten id und value	Listener-Funktion; wird durch das VD-Objekt aufgerufen, wenn sich der Wert eines abonnierten Prozessdatums ändert
getProfilData	-	Auslesen der WOAS-Profildatei (falls eine solche vorhanden ist)  Rückgabewert: -1 = keine Profildatei vorhanden JSON = Daten aus der Profildatei
logError logWarning log	errorMsg  Verschiedene Meldungen als String	Wird für die Ausgabe von Meldungen in die Browser-Konsole genutzt

Der folgende Quellcode zeigt das WOAS-VD-Basisobjekt (Meta-Objekt):

```
function vdBase() {
    this.data;
    this.ready;

    this.init = function(){}
    this.readChannel = function(channel){}
    this.readAllChannels = function(){}
    this.writeChannel = function(channel,value){}
    this.isReady = function(status){}
    this.subscribe = function(){}
    this.unsubscribe = function(){}
    this.refreshChannelData = function(data){}
    this.getProfilData = function(){}
    this.LogError = function(errorMsg){}
    this.LogWarning = function(){warningMsg}
    this.log = function(msg){}
}
```

Für Testzwecke kann das vdBase-Objekt mit weiteren Methoden erweitert sein, die aber nur für einen Einsatz außerhalb eines WOAS-Portals sinnvoll sind.

#### 4.3.4 Testumgebung für ein VD

Für die Implementierung einer neuen VD Class wird eine Testumgebung benötigt, um das VD zu erproben, bevor es in das WOAS-Portal integriert wird. Die Testumgebung muss so ausgelegt sein, dass ein vollständiger Funktionstest des VD außerhalb der WOAS-Umgebung erfolgen kann. Das VD Class-Objekt muss deshalb in einem Anwendungsrahmen getestet werden, der implementierungstechnisch identisch zur WOAS-Umgebung ist.

Der Anwendungsrahmen für den VD-Test besteht aus einer Verzeichnisstruktur im Pfad eines lokalen Webservers. Tabelle 13 erläutert diese Verzeichnisstruktur.

**Tabelle 13:** Verzeichnisstruktur des Anwendungsrahmens für den VD-Test

Name	Name	Beschreibung	Bemerkungen
	/VD	Ordner für alle zu testenden VD Classes	Nur für Testzwecke
	<class>	Ordner für die jeweilige VD Class	
	/img	Ordner für die VD-Icons	Upload in das WOAS-Portal
	/name	Optionaler Ordner für spezielle Dateien, die die jeweilige VD Class benötigt	Upload in das WOAS-Portal
	<class>.js	JS-Implementierung der VD Class	Upload in das WOAS-Portal
	VD_template.html	HTML-Template zur Visualisierung der VD Class in der Testumgebung	Nur für Testzwecke
	/woas	Ordner zur Simulation der	

		WOAS-Umgebung	
	jquery-1.9.1.min.js	jQuery- JavaScript-Framework	
	json2.js	JSON-Library	
	vdBase.js	WOAS-Meta-Objekt für Virtual Devices	
	<class>_test.html	Testseite für die jeweilige VD Class	

Nur die markierten Dateien bzw. Ordner werden letztendlich in das WOAS-Portal als VD hochgeladen. Alle anderen Dateien und Ordner gehören zur Testumgebung und können auch individuell eingerichtet werden.

Als Webserver zum Test kann prinzipiell jeder beliebiger Webserver fungieren. Falls Webanwendungen für die VD Class genutzt werden, muss der Testserver allerdings diese auch realisieren können.

Das WOAS-Portal bietet in der Demoversion die Möglichkeit PHP-Anwendungen als Webanwendungen auszuführen. Es wird deshalb empfohlen den Apache-Webserver für die Testumgebung einzusetzen.

Im WOAS-Portal selbst steht folgende Systemumgebung zur Verfügung:

- Webserver: XAMPP
- JS-Frameworks: jquery-1.9.1.min.js, jquery.mobile-1.3.2.min.js, json2.js
- CSS-Frameworks: jquery.mobile-1.3.1.min.css

Für die Testumgebung muss weiterhin das Basisobjekt vdBase modifiziert werden, so dass in der Testumgebung die Visualisierung der VD Instances mit einer Testseite möglich ist. Es wird deshalb für die Methode refreshChannelData eine Implementierung genutzt, die die Prozessdaten in der Test-Tabellenstruktur visualisiert bzw. spezielle Test-Listener auf der Testseite aufruft. Der folgende Quellcode zeigt die Implementierung von refreshChannelData im Basisobjekt für den Testbetrieb:

```
this.refreshChannelData = function(data){
    //ONLY FOR TESTING
    for(i=0;i<data.length;i++){
        var channel = this.getChannelByID(data[i].id);
        if(this.visible){
            this.refreshPD(data[i].id,data[i].value);// if the VD is visible
        }
        else{
            var listener =
                this.data.id+_onDataChanged(""+data[i].id+","+data[i].value+"");
            eval(listener) //Call the instance-specific listener on the test page
        }
    }
}
```

#### 4.3.5 VD-Beispiele für WOAS

In [5] werden VD-Implementierungen beschrieben, die im WOAS-Portal bereits integriert sind und durch Dritte genutzt werden können. Dazu gehören u.a. die folgenden VD classes:

- VD2PHPOPCDA: Zugriff auf Prozessdaten beliebiger OPC DA-Server per PHP
- VD2WAK: Zugriff auf Prozessdaten über ein externes WPG per TCP/IP und WAK-Protokoll aus dem Web Access Kit (WAK) des CCAD [1]. Aktuell steht der Zugriff über WAK 1.3 auf beliebige OPC DA-Server zur Verfügung
- VD2WPG: Zugriff auf Prozessdaten per WebSocket und WOAS Device Protocol. Aktuell steht der Zugriff über das WPG 1.5 auf beliebige OPC DA-Server zur Verfügung

Alle VD Classes nutzen als Meta-Objekt das `vdbase`-Objekt und implementieren die entsprechenden Methoden zum Teil aber unterschiedlich. Darüber hinaus beinhalten die VD Classes einige Methoden, die für alle VD Classes zwar identisch, aber teils unterschiedlich, entsprechend der Art des jeweiligen Prozesszugriffs, implementiert sind

#### **4.3.6 WOAS Device Protocol**

##### Allgemeiner Protokollaufbau

Das WOAS-DP orientiert sich am Telegrammaufbau für das Web Access Kit (WAK) zum Zugriff auf OPC-DA-Daten und optimiert dieses basierend auf den mit WAK gewonnenen Erfahrungen (WAK – früheres FuE-Projekt im CCAD). Es ist ein ASCII-basiertes Telegrammprotokoll, welches bei Übertragung über ein IP-Netz TCP nutzt und folgenden Aufbau besitzt:

```
<command> <parameter 1>;<parameter 2>;...<parameter N>
```

- Jedes Telegramm ist ein abgeschlossener String, d.h. er wird immer mit der Escape-Sequenz `\r\n` beendet (Carriage Return und Zeilenvorschub).
- Boolesche Werte werden in allen Telegrammen als `,1‘` und `,0‘` und nicht als `,true‘` und `,false‘` notiert.

##### WOAS Device Profile

Zu jedem WPG gehört eine WOAS-Profildatei (Gerätebeschreibung), in der die WOAS-spezifischen Eigenschaften des jeweiligen AGs mittels XML beschrieben sind (ähnlich der WAK-Konfigurationsdatei). Für ein VD Typ 2 ist die Profildatei auf dem WOAS-Server oder einem Server des Geräteanbieters gespeichert. Für ein VD Typ 3 ist die Profildatei im WDG gespeichert und fungiert gleichzeitig als Konfigurationsdatei für das WDG.

Die Profildatei muss durch den jeweiligen Nutzer z.Z. noch extern, außerhalb des WOAS-Engineering-Prozesses, erstellt werden.

Die Struktur des WOAS-Geräteprofils orientiert sich an der WAK-Konfigurations-Datei und beinhaltet

- Verbindungsinformationen zum AG,
- Prozessdatenzuordnungen,
- Parameter für den ereignisbasierten Lese-Zugriff auf Prozessdaten,
- Zugriffs- und Benutzerrechte.

#### 4.3.7 Zeitverhalten

Um Automatisierungsfunktionen als Dienste aus einer Cloud mit einem Automatisierungsgerät realisieren zu können, spielt die Zeitcharakteristik des Interfaces (CPS<sup>3</sup>-Interface) eine wesentliche Rolle. Das Zeitverhalten der Prozessdatenübertragung über ein solches CPS-Interface bestimmt im Wesentlichen, für welche Anwendungsklassen (Echtzeitklassen) eine CPS-basierte bzw. dienste-basierte Automation überhaupt eingesetzt werden kann.

Aus dem Stand der Technik liegen bisher kaum Messungen zur Prozessdatenübertragung über das Internet/Web vor. Die Angaben schwanken und werden (wenn überhaupt) je nach Übertragungsprinzip im zweistelligen ms-Bereich bis Sekunden angegeben. Belastbare und systematische Messungen zur Prozessdatenübertragung zwischen einer Webseite und dem realen physischen Ausgang/Eingang eines AGs (z.B. einer SPS) fehlen bisher vollständig. Web-basierte SCADA- bzw. Prozessleitsysteme geben deshalb nur sehr vage die erzielbaren Aktualisierungsraten für die Prozessdatenübertragung beim Webzugriff an. Eine wie auch immer festgelegte Garantie für eine definierte Laufzeit eines Prozessdatums von der Webseite bis zum Ausgang eines physischen AGs wird nicht übernommen. Da im WOAS-Projekt die VDs gemeinsam mit den AGs als CPS-Komponenten betrachtet werden können, wurden im Projekt umfangreiche Zeitmessungen durchgeführt, um das mögliche Zeitverhalten Dienst – Gerät einschätzen zu können.

##### Serverseitige Ermittlung de Zeitverhaltens

Für WOAS sollte ein Messverfahren gefunden werden, über welches die unterschiedlichen Laufzeiten eines Prozessdatums von einer Webseite zum physischen Ausgang eines AGs möglichst genau ermittelt werden können. Des Weiteren sollte der Messaufbau eine realistische, anwendungstypische Umgebung darstellen. Bild 12 zeigt den Aufbau des gewählten Messaufbaus.

Nach verschiedenen Versuchen wurde eine Messmethode angewandt, mit der die gesamte Laufzeit eines digitalen Prozessdatums für Lesen und Schreiben zwischen einem VD auf einer Webseite und einem AG (SPS) ermittelt wurde. Dabei wird ein Ausgang der SPS mit einem Eingang softwaremäßig durch einen Client nach dem OPC-Server und auf der Webseite (nach dem VD) „kurzgeschlossen“ und die beiden Zeitdifferenzen  $\Delta T_1$  und  $\Delta T_2$  an den jeweiligen SPS-Aus- bzw. Eingang mit einem Oszilloskop vermessen.

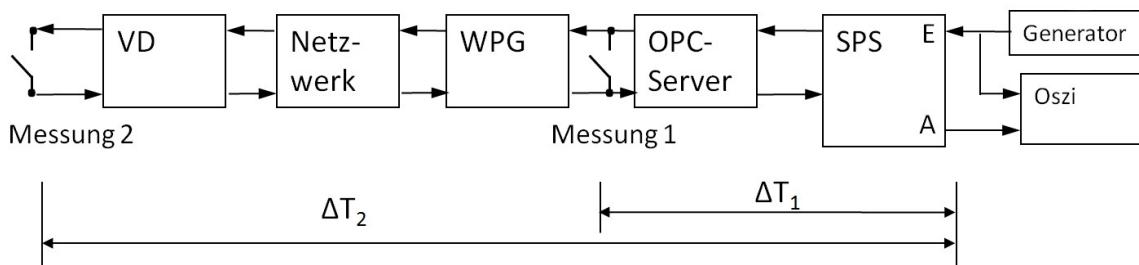


Bild 12: Messaufbau zur serverseitigen Messung des Zeitverhaltens

Es bedeuten:

$\Delta T_1$  – Laufzeit des Signals vom SPS-Eingang über den OPC-Server zurück auf einen SPS-Ausgang.

<sup>3</sup> CPS – Cyber Physical System

$\Delta T_2$  – Laufzeit des Signals vom SPS-Eingang über das VD auf der Webseite zurück auf einen SPS-Ausgang.

Die Laufzeit eines Signals von der Webseite bis zum realen Ausgang des AGs beträgt damit bei einer one-way-Betrachtung:

$$\Delta T_{\text{lauf}} = 0,5 \times \Delta T_2$$

Die durch den Webzugriff verursachte Verzögerungszeit beträgt:

$$\Delta T_{\text{web}} = 0,5 \times \Delta T_2 - 0,5 \times \Delta T_1$$

Als SPS wurde eine Phoenix Contact-Steuerung ILC 150ETH mit Zugriff über einen OPC-Server „AX OPC-Server 3.0“, installiert auf einem Spectra-IPC, eingesetzt. Am Eingang der SPS ist ein Funktionsgenerator angeschlossen, dessen Rechteckspannung auf Kanal A eines Oszilloskops aufgezeichnet wird. Mit Kanal B des Oszilloskop wird die verzögerte Ausgangsspannung am Ausgang der SPS aufgezeichnet. Als Oszilloskop kam das „PicoScope 5204“ zum Einsatz. Das SPS-Programm realisiert nur die Verbindung zwischen den physischen Ein-/Ausgängen und den dazu erforderlichen OPC-Variablen. Für die Messung mit 10 Prozessdaten gleichzeitig werden im SPS-Programm 10 OPC-Eingangsvariablen generiert, die durch einen Java-OPC-Client auf 10 Ausgangsvariablen an die SPS zurückgeschrieben werden. Erst nach Eintreffen aller Variablen wird der Ausgang der SPS auf „1“ gesetzt.

Es wurden jeweils 200 Messungen durchgeführt und der Mittelwert ermittelt. Die Messungen erfolgten mit unterschiedlichen Frequenzen des Rechteckgenerators (0,5 Hz, 2 Hz und 5 Hz) Tabelle 14 zeigt die gemessene Übertragungszeit zwischen SPS-Ein-/Ausgang und dem Ein-/Ausgang des OPC-Servers.

**Tabelle 14:** Laufzeiten zwischen OPC-Server und SPS

Anzahl der OPC-Variablen	Laufzeit $\Delta T_1$ [ms]
1	32
10	139

Aus Tabelle 1 ist ersichtlich, dass bereits die Laufzeit über den OPC-Server sowie das SPS-Programm mit  $\Delta T_1 = 32$  ms eine erhebliche Verzögerung besitzt.

Für die Prozessdatenübertragung über das IP-Netz wurden drei unterschiedliche VDs verwendet, die jeweils auch verschiedene WOAS Protocol Gateways (WPG) nutzen. Die WPGs sind gleichfalls auf dem IPC installiert:

- VD 1: Das VD nutzt PHP, um die Prozessdaten mittels eines HTTP-Request/Response zu einem PHP-basierten WPG per Client-Pull zu übertragen.
- VD 2: Das VD nutzt ein Java-Applet, um die Prozessdaten mittels eines proprietären TCP-Protokolls zu einem java-basierten WPG per Server-Push zu übertragen. Diese Methode nutzt das Web Access Kit (WAK) aus einer Vorstudie des Autors [5].
- VD 3: Das VD nutzt nur JavaScript, um die Prozessdaten mittels des WOAS-DP über Websockets zu einem java-basierten WPG per Server-Push zu übertragen.

VD 1 und 3 können Port 80 auch für die Prozessdatenübertragung nutzen, VD 2 benötigt einen weiteren Port als Prozessdatenkanal.

In Tabelle 15 sind die bisher gemessenen Werte für die Laufzeiten  $\Delta T_2$ , bedingt durch das WPG, das Netzwerk selbst sowie das VD auf der Webseite ausgeführt. Die Messungen erfolgten lokal auf dem IPC, im Intranet sowie über das Internet (bisher konnten leider noch nicht alle Messungen abgeschlossen werden).

**Tabelle 15:** Laufzeiten  $\Delta T_2$ , bedingt durch WPG, Netzwerk und VD

	Laufzeit $\Delta T_2$ [ms] Lokal	Laufzeit $\Delta T_2$ [ms] Intranet	Laufzeit $\Delta T_2$ [ms] Internet
<b>1 Prozessdatum</b>			
<b>VD 1</b>	160	165	206 (Solingen – Düsseldorf)
<b>VD 2</b>	44	45	103 (Solingen – Düsseldorf) 112 (Odessa – Düsseldorf)
<b>VD 3</b>	46	47	95 (Solingen – Düsseldorf)
<b>10 Prozessdaten</b>			
<b>VD 1</b>	> 300		
<b>VD 2</b>	146		
<b>VD 3</b>	139		

Für die Variante VD 2 ergibt sich z.B. eine Laufzeit für ein Prozessdatum im Internet (Solingen – Düsseldorf) von einer Webseite bis zum Ausgang einer SPS von **51,5 ms**. Abzüglich der Verzögerung durch den OPC-Server/SPS-Programm ergibt sich eine Laufzeit bedingt durch den IP-Anschluss und der IP-Übertragung von **35,5 ms**. Ein ähnliches Verhalten zeigt auch VD 3 (Übertragung über Websockets). Die PHP-Variante ist erheblich langsamer, da diese mit einem Client-Pull arbeitet und die serverseitige PHP-Verarbeitung auf dem IPC die CPU stark belastet.

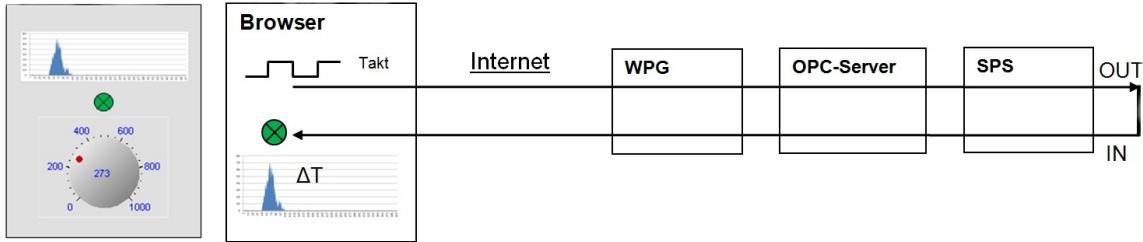
Aus Anwendersicht lassen die Messergebnisse die Interpretation zu, dass nach Aktivierung eines Buttons auf der Webseite nach ca. 50 ms das zugehörige Signal über das Internet am SPS-Ausgang bereits anliegt.

Problematisch erweist sich allerdings die Streubreite der gemessenen Laufzeiten, d.h. die Durchschnittswerte sind nur sehr bedingt für Aussagen über das Zeitverhalten anwendbar. Es wurde deshalb eine zweite, clientseitige Messmethode eingesetzt, um die Zeitdeterministik der Prozessdatenübertragung über das CPS-Interface besser darstellen zu können.

#### Clientseitige Ermittlung des Zeitverhaltens

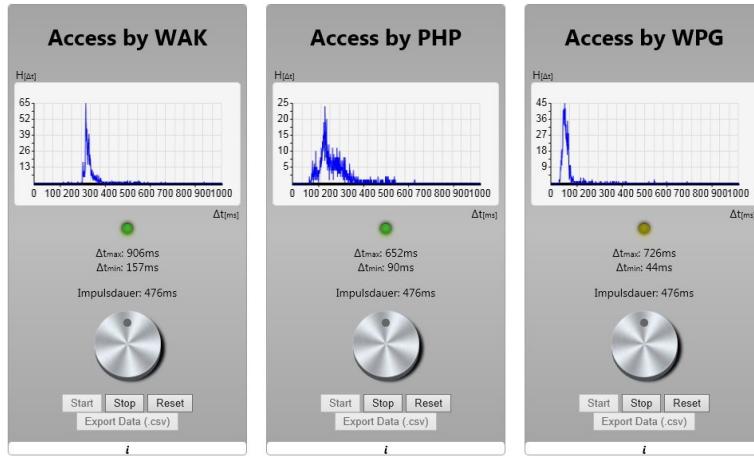
Für die clientseitige Ermittlung des Zeitverhaltens erzeugt auf einer Test-HTML-Seite ein JS-Generator eine Pulsfrequenz, die an einen SPS-Ausgang geschickt wird. Dieser Ausgang ist hardwaremäßig mit einem SPS-Eingang verbunden, der seinen Wert wieder zurück auf die HTML-Seite liefert. Dort erfolgt dann eine Zeitdifferenzmessung und Aufzeichnung in einem Histogramm. Bild 13 verdeutlicht das Messprinzip.

Mit dieser Messmethode erhält man die Zeitcharakteristik der Prozessdatenübertragung über einen längeren Zeitraum und erfasst auch die Streubreite der Messungen über eine Wahrscheinlichkeitskurve.



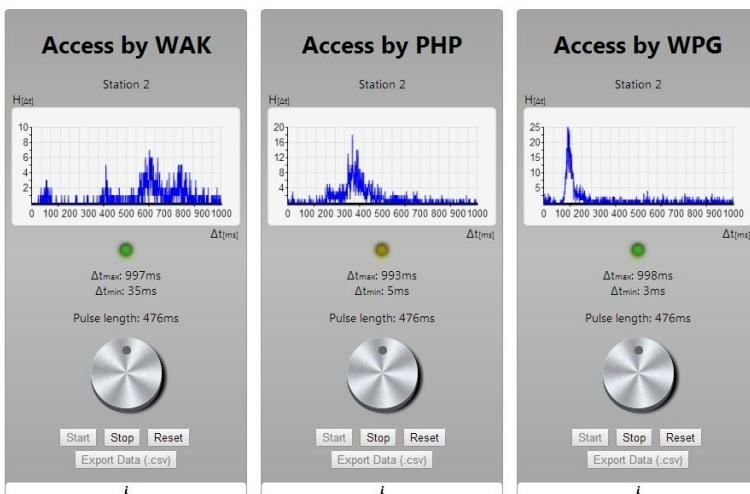
**Bild 13:** Messmethode zur clientseitige Ermittlung des Zeitverhalten

Bild 14 zeigt als Beispiel die Messungen über einen Zeitraum von 30 min für einen Standort in Solingen mit Kabel-DSL-Anschluss sowie Bild 15 für einen Standort in Porto (Portugal) mit WLAN-Anschluss im Hotel.



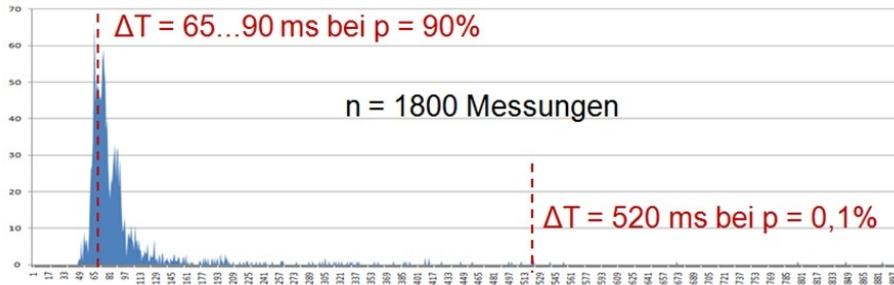
**Bild 14:** Zeitcharakteristik für die Übertragung eines Prozessdatums über das Internet (Solingen – Düsseldorf) während einer Zeitdauer von 30 min

Bild 15 zeigt deutlich eine verlängerte Laufzeit und ein erheblich größeres Rauschen (höhere Wahrscheinlichkeit einer Streuung der Laufzeiten) offensichtlich verursacht durch den qualitativ schlechteren WLAN-Zugang.



**Bild 15:** Zeitcharakteristik für die Übertragung eines Prozessdatums über das Internet (Porto – Düsseldorf) während einer Zeitdauer von 30 min

Bei Aufruf der Testseite im Chrom-Browser, können die Zeit-Histogramme als ExcelSheet exportiert und analysiert werden. Bild 16 zeigt dazu die Zeitverhältnisse einer Prozessdatenübertragung über das Internet mit Websockets auf OPC DA (VD 3).



**Bild 16:** Zeitdeterministik einer Prozessdatenübertragung über das Internet mit Websockets auf OPC DA

Als eine wichtige Schlussfolgerung ergibt sich, dass für das Echtzeitverhalten (Rechtzeitigkeit) über ein IP-Netz die Wahrscheinlichkeit der Reaktionszeit berücksichtigt werden muss.

Die in Bild 14 dargestellte Testseite kann für eine Phoenix Contact-Steuerung (unter <http://testvd1.ccad.eu>) sowie für eine Siemens-Steuerung (unter <http://testvd2.ccad.eu>), die beide im WOAS-Demonstrator verbaut sind, aufgerufen und getestet werden.

Die ausführlichen Messergebnisse zum Zeitverhalten der VDs finden sich in [6].

#### Zusammenfassung zum Zeitverhalten

Mit den WOAS-VDs lässt sich über das Web eine Prozessdatenübertragung unter Nutzung von Websockets mit Zugriff auf OPC DA-Prozessdaten mit einer Reaktionszeit von 32 ... 45 ms (one-way-Betrachtung) mit einer Wahrscheinlichkeit von 90% über das Internet erreichen.

Die beiden ausgewählten Messmethoden liefern grundsätzlich für das PHP-basierte VD und für das WebSocket-basierte VD vergleichbare Ergebnisse. Für das Java-basierte VD (VD 2) ergeben sich für die clientbasierte Messmethode über das Internet erheblich höhere Laufzeiten (ca. 140 ms) gegenüber der serverbasierten Methode. Die Ursachen dafür konnten noch nicht sicher ermittelt werden. Es wird vermutet, dass die Ursachen im Routing des im VD 2 genutzten proprietären TCP-Protokolls liegen.

Die PHP-Variante (VD 1) ist für relativ langsame technische Prozesse (> 500 ms) nutzbar. Die Streubreite der Laufzeiten ist aber relativ hoch (s. Bild 13) und die Performance wird durch die serverseitigen Rechnerressourcen stark beeinflusst.

Die besten und stabilsten Ergebnisse wird mit einer Prozessdatenübertragung über Websockets erzielt. Die WebSocket-Ausführung der VDs (VD 3) wird deshalb zukünftig weiter ausgebaut und untersucht. Weitere Arbeiten sollen sich in diesem Zusammenhang zukünftig konzentrieren auf:

- Ermittlung der detaillierten Einflüsse auf die Prozessdaten-Laufzeiten im Netzwerk sowie im clientseitigen VD. Insbesondere sind hier die Ursachen für die Streubreite der Messungen zu ermitteln.

- OPC-Server verursachen eine relativ hohe Zeitverzögerung. Zukünftig sollen deshalb mittels Zugriff auf das AG über andere Industrieschnittstellen (z.B. Modbus TCP) die Laufzeiten reduziert werden.
- Überprüfung der Zeitcharakteristik bei Nutzung einer OPC-UA-Schnittstelle im AG.
- Durchführung weltweiter, globaler Tests aus verschiedenen Orten und mit unterschiedlichen Internet-Zugängen.

Berichte und Dokumente:

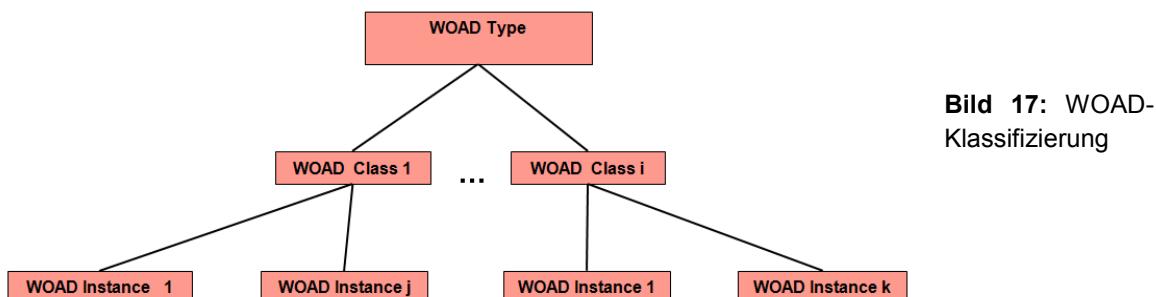
[5] CCAD: Virtual Devices – Beschreibung. – 13.03.13

[6] Frank, B.: Testbericht: Web-basierte Bereitstellung von OPC-Daten. – 24.07.13

## 4.4 Automatisierungsfunktionen als Dienste

### 4.4.1 Dienste im WOAS-System

Zur Realisierung unterschiedlicher AF werden die äquivalenten Dienste (WOAD) in fünf unterschiedliche WOAD Typen eingeteilt, die sich insbesondere durch die Art ihrer Kommunikation mit den Virtual Devices (VD) unterscheiden. Die fünf Typen bieten weiterhin eine grobe Richtlinie für die softwaretechnische Implementierung von WOAD. Alle WOAD Typen basieren softwaretechnisch auf einer Basisklasse (ähnlich dem Prinzip für die Erstellung von VDs [5]), sind damit logisch miteinander verbunden und teilen ihre Grundfunktionalität. Bild 17 zeigt die WOAD-Klassifizierung.



Die WOAD-Elemente der Klassifizierung nach Bild 16 sind entsprechend Tabelle 16 zu verstehen:

**Tabelle 16:** Elemente der WOAD-Klassifizierung

Klassifizierungselement	Beschreibung
WOAD Type	Charakterisiert die prinzipielle Struktur eines WOAD
WOAD Class	Charakterisiert die jeweilige allgemeine implementierungstechnische Umsetzung
WOAD Instance	Ein WOAS-Nutzer mit der Rolle <i>Admin</i> kann im WOAS-System verfügbare WOAD Classes parametrieren und erhält dann konkrete WOAD Instances

Aktuell werden in WOAS fünf WOAD-Typen unterschieden:

- **WOAD Type 1:** Der Dienst ist auf der Webseite nicht sichtbar. Damit können z.B. auf der Webseite PD Input/Output-Algorithmen (Steuerung, Datenvorverarbeitung) abgearbeitet oder Input-Daten können in einer DB abgespeichert werden. Type 1 benötigt Prozessdaten von VD-Output-Channels oder von VD (Input & Output)-Channels.

- **WOAD Type 2:** Der Dienst ist auf der Webseite sichtbar. Dieser Typ verwendet nur Output-Channels eines VDs und ist damit ein typischer Vertreter eines Online-Prozessvisualisierungsdienstes.
- **WOAD Type 3:** Der Dienst ist auf der Webseite sichtbar. Dieser Typ verwendet i.d.R. nur VD-Output-Channeldaten und ist damit ein typischer Vertreter einer Bedienelemente-Schnittstelle. Falls erforderlich können aber auch VD-Input-Channels für z.B. Rückmeldungen über Bedienaktionen erforderlich sein.
- **WOAD Type 4:** Der Dienst ist auf der Webseite sichtbar. Dieser Typ verwendet Prozessdaten von Output- und Input-Channels eines VD und ist damit ein typischer Vertreter einer HMI-Schnittstelle mit Beeinflussung des Prozessgeschehens durch Bedienelemente und Online-Visualisierung von Prozessdaten.
- **WOAD Type 5:** Der Dienst ist auf der Webseite sichtbar. Dieser Typ benötigt aber keine Prozessdaten von einem VD. Anwendungen sind z.B. die Realisierung von historischen Trendanalysen, basierend auf bereits abgespeicherten Prozessdaten in einer Datenbank.

Bild 18 veranschaulicht Beispiele der fünf WOAD Typen.

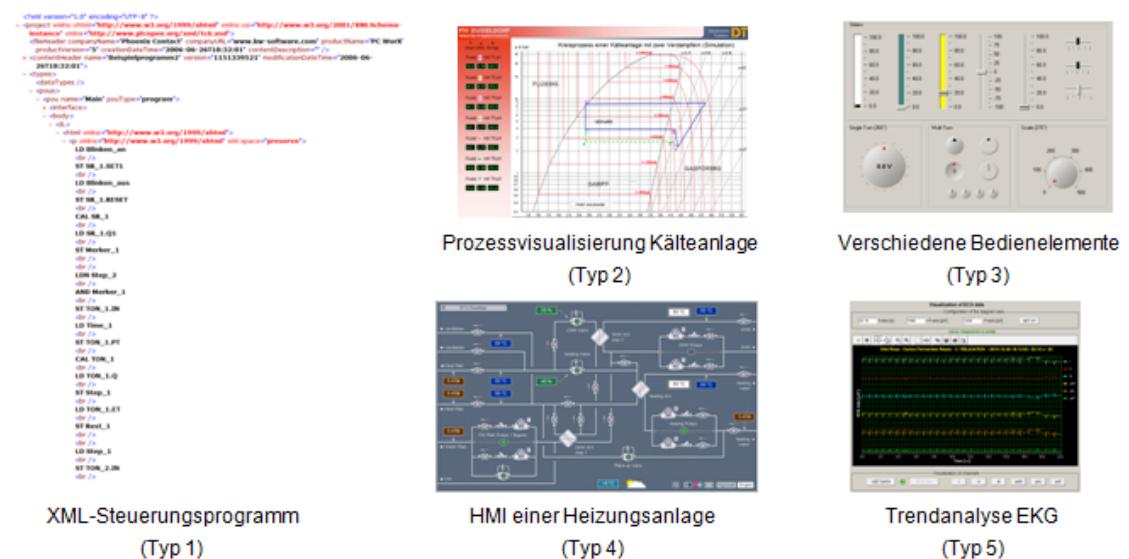


Bild 18: Beispiele der fünf WOAD-Typen im WOAS-System

Im WOAS-Demoportal sind die in Tabelle 17 aufgeführten WOAD Classes (Dienste) bereits integriert.

Tabelle 17: WOAD Classes im WOAS-Demoportal

Dienstname	Dienstklasse	Beschreibung
Simple LED	service4sled	Einfacher HTML5-Canvas-Kreis als LED-Anzeigeelement
Simple Button	service4button	Einfacher HTML5-Canvas-Kreis als Click-Button
Plotter	service4fplotter	Plotter mit zwei Kanälen zur Echtzeit-Anzeige von analogen Prozessdaten
LED Faceplate	service4ledfaceplate	Faceplate mit bis zu 6 Anzeige-LEDs, angeordnet in parametrierbaren Zeilen und Spalten.

Image Switch	service4iswitch	Schalter, der parametrierbare Images zur Darstellung der Schalterstellungen verwendet
Image LED	service4led	Digitales Anzeigeelement, welches parametrierbare Images zur Darstellung der 1/0 verwendet
Station Visualization	service4visustation	Grafisch-dynamische Anzeige für die Stationen des WOAS-Demonstrators.
Webcam	service4webcam32	Anzeige eines Streaming-Videos basierend auf dem Webcam32-Videoserver
Analog Display	service4txtanalog	Dynamische Textanzeige analoger Prozessdaten
Table Display	service4txtdisplay	Dynamische Textanzeige von Prozessdaten in Tabellenform mit interaktiver Eingabe für schreibbare Prozessdaten
Input Field	Service4inputfield	Interaktives Eingabefeld für analoge Werte

WOAD können prinzipiell mittels clientseitiger oder serverseitiger Programmierung bzw. auch gemischt realisiert werden.

Durch die Nutzung eines einheitlichen Channel-Konzepts für die WOAD und die VDs ist es möglich, dass Dienste auch untereinander – ohne Einbeziehung von VDs - über ihre Channels - verbunden werden können (z.B. kann ein Schalter als Dienst eine LED als Dienst eins-/ausschalten)

#### 4.4.2 Allgemeine Struktur eines WOAD

Ein WOAD besteht aus einer JS-Datei, einer Bilddatei sowie optional einem Dateiverzeichnis, in dem spezielle Dateien, die für die Implementation des Dienstes erforderlich sind, abgelegt werden. Tabelle 18 zeigt dazu eine Übersicht.

Als WOAD Class-Name sollte ein aussagefähiger Kurzname gewählt werden z.B. service4leds oder service4plot. Für die Implementierung eines WOAD wird der Einsatz des JS-Frameworks *jQuery* empfohlen. *jQuery* steht im WOAS-Kern bereits zur Verfügung. Sollen andere JS-Frameworks genutzt werden, müssen diese in den WOAD über den optionalen Ordner mit eingebunden werden.

**Tabelle 18:** Bestandteile eines WOAD in WOAS

Dateiname	Erforderlich	Beschreibung
<class>.js	ja	Die Javascript-Datei, die die Implementierung eines Dienstes als WOAD enthält.
icon_140.png	ja	Icon für den WOAD mit Abmessungen 140 x 140
<name_1>/ <name_2>/ ... <name_N>/	optional	Abhängig von der Implementierung des Dienstes sind zusätzliche Dateien notwendig, z.B. php-, weitere Javascript-, oder andere Dateien. Diese werden in separaten Ordnern abgelegt, die flexibel benannt werden können.

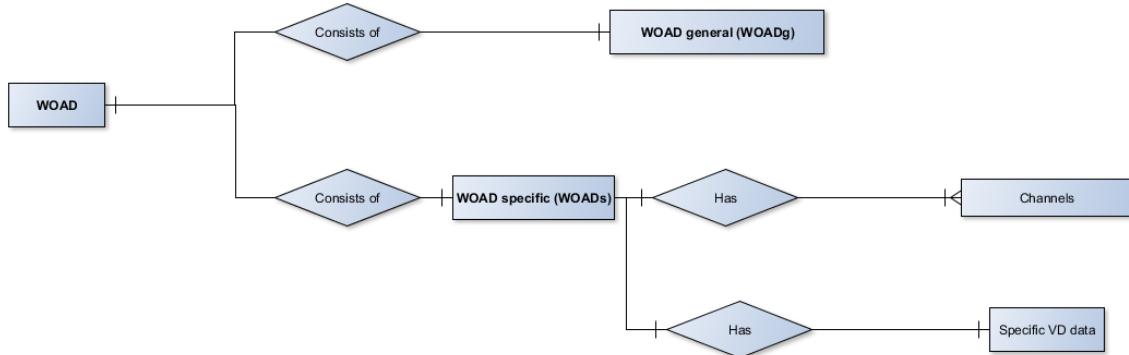
Nach dem Einloggen in das WOAS-Portal kann ein User mit der Rolle *Publisher* die WOAD-Dateien in das Portal hochladen oder eine IP-Adresse angeben, über die die WOAD-Dateien erreichbar sind.

Die Rahmenrichtlinien zur Implementierung einer WOAD Class finden sich in [7].

#### 4.4.3 Datenmodell eines WOAD

WOAD Classes sowie die zugehörigen durch einen Nutzer parametrierten unterschiedlichen WOAD Instances werden im WOAS-System in einer Datenbank gespeichert und verwaltet. Die Datenstruktur eines WOAD ist entsprechend Bild 19 aufgebaut.

Jeder WOAD besteht aus allgemeinen Daten (General Data) sowie spezifischen WOAD-Daten (Specific Data). Die General Data werden durch den Anbieter des WOAD (Publisher) vorgegeben und können auch nur durch einen WOAS-Nutzer mit der Rolle *Publisher* verändert werden. Die General Data charakterisieren die WOAD Class.

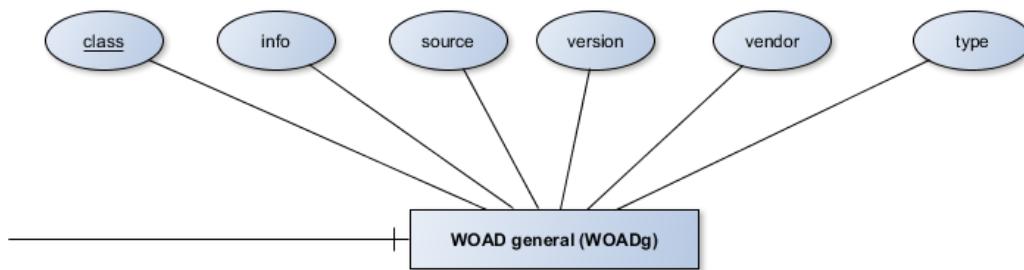


**Bild 19:** Datenstruktur eines WOAD

Die spezifischen WOAD-Daten charakterisieren die WOAD Instances und werden durch den WOAS-Nutzer mit der Rolle *Admin* erzeugt und parametriert.

##### Allgemeine WOAD-Daten

Die allgemeinen WOAD-Daten (Bild 20) werden im Wesentlichen nur im Edit-Mode benötigt. Sie dienen der Beschreibung des jeweiligen Dienstes. Sie werden in der Datenbank abgelegt und im WOAD-Edit-Mode aufgerufen, angezeigt und können nur durch den *Publisher* editiert werden.



**Bild 20:** General WOAD Data

Die einzelnen Daten sind in Tabelle 19 beschrieben.

**Tabelle 19:** General WOAD Date

Name	Beschreibung	Beispiel
class	Name der WOAD Class = Bezeichnung der JS-Datei zur Implementierung des VD	service4led
info	Beschreibung der WOAD Class	Einfache Anzeige-LED
source	IP-Adresse des Hauptordners des WOAD	/WOAS/WOAD/SERVICE4LED/
version	Versionsnummer der Class	1.0
vendor	Hersteller des WOAD	CCAD
type	Nummer des WOAD Typs	2

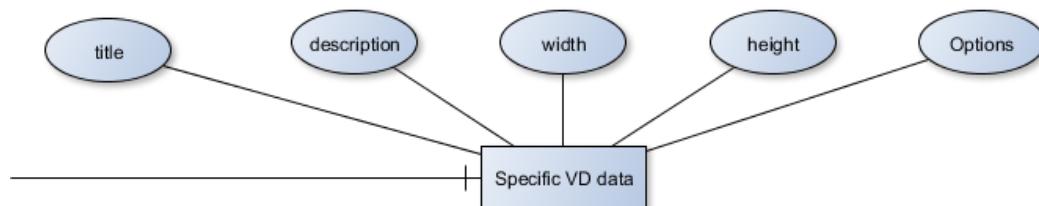
### Spezifische WOAD-Daten

Die spezifischen Daten eines VD teilen sich in

- *Specific WOAD Data* (Bild 21, Tabelle 20): Parametrierung für Funktion und Darstellung des Dienstes.
- *Channels* (Bild 22): Diese Daten beinhalten die editierbaren Parameter sowie die für den Dienst zur Verfügung stehenden Prozessdaten.

Im Folgenden werden die spezifischen WOAD-Daten im Detail erläutert.

*Specific WOAD Data:*



**Bild 21:** Specific WOAD Data

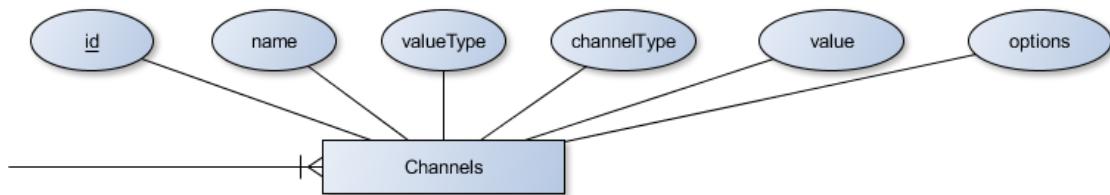
**Tabelle 20:** Specific WOAD Data

Name	Beschreibung	Beispiel
title	Name der WOAD Instance	SimpleLED
description	Kurzbeschreibung der WOAD Instance	LED zur Visualisierung
width	Breite des sichtbaren Teils des WOAD auf der Webseite in Pixel oder %	80
height	Höhe des sichtbaren Teils des WOAD auf der Webseite in Pixel	20
options	Zusätzliche WOAD-spezifische variable Anzahl von weiteren Parametern, die für alle Channels gelten	bgcolor: Hintergrundfarbe für den Dienst

### *Channels:*

Diese Daten beinhalten die durch den Dienst bereitgestellten Kanäle, die mit Prozessdaten der VDs verbunden werden können. Die Eingangskanäle eines WOAD arbeiten ereignisbasiert, d.h. jede Änderung eines Datums am Eingangskanal wird automatisch in den Dienst übernommen und dort verarbeitet.

Im WOAS-Edit-Modus (WOAS-Creator) werden beim Editieren einer Verbindung zwischen einem VD-Channel und einem WOAD-Channel die Typgleichheit der Channel über die Parameter valueType und channelType geprüft.



**Bild 22:** WOAD Channels

Die Channels beinhalten die in Tabelle 21 aufgeführten Daten.

**Tabelle 21:** Daten eines Channels

Name	Beschreibung	Beispiel
id	Eindeutige numerische Identifikation des Channels	1
name	Beschreibender Kurzname des Channels (Alias)	Limit Switch
valueType	Channeldatentyp-Beschreibung entsprechend der WOAS-Notation als Typnummer (s. Anlage 1)	17
channelType	0 – Channel ist nur lesbar 1 – Channel ist les- und schreibbar 2 – Channel ist nur schreibbar	0
value	Wert des Channels als String [oder BOOL oder INT]	TRUE
options	Zusätzliche Channel-spezifische variable Anzahl von weiteren Parametern	LEDOn: Image für die Anzeige von LEDOn

Innerhalb eines WOAS Portals werden Channels als Instanzen eines serviceChannels-Objekts realisiert. Dieses hat im wesentlichen die Aufgabe alle Änderungen des Kanals (durch den Benutzer und VDs) über verschiedene Clients und Seiten hinweg zu synchronisieren. Diese Synchronisation erfolgt Eventbasiert, weshalb das Objekt serviceChannel Methoden zum registrieren eines Eventlisteners bereitstellt. Die Nachbildung dieses Verhaltens außerhalb eines WOAS-Portals in einer Testumgebung erfolgt durch die zwei JS-Dateien eventTarget.js und serviceChannel.js.

### Editierbarkeit der WOAD Daten

WOAD Data können im WOAS-Portal von Nutzern mit unterschiedlichen Rollen editiert werden. Die Tabelle 22 zeigt dazu eine Übersicht.

**Tabelle 22:** Editierbarkeit von WOAD Data

		WOAD Data	Editor
General	WOAD Data	class	Publisher
		info	
		source	
		version	
		vendor	
		protocol	
		type	
Specific	WOAD Data	title	Admin
		description	
		xPos	
		yPos	
		width	
		height	
	Channels	options	Admin (falls vorhanden)
		id	Publisher
		name	Publisher
		value	x
		valueType	Publisher
		channelType	Publisher
		options	Admin (falls vorhanden)

In Tabelle 22 bedeuten:

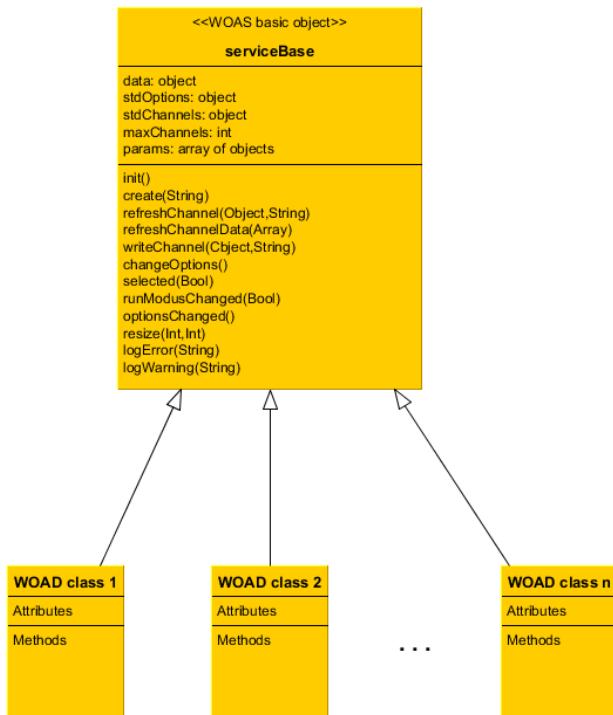
- x: Wert ist nicht editierbar
- Publisher: Wert kann nur durch einen Nutzer mit der Rolle *Publisher* editiert bzw. festgelegt werden.
- Admin: Wert kann nur durch einen Nutzer mit der Rolle *Customer-Admin* editiert werden.

Die WOAD-Daten, die für die Darstellung des Dienstes im WOAS-Portal (oder einer Testumgebung) verantwortlich sind (Parameter) werden durch den Dienst selbst implementiert. Es gibt dann pro Dienst jeweils ein Parameterobjekt mit den für alle Channel des Dienstes geltenden Options (`data.options`) und zusätzlich Parameterobjekte, die die Parameterdarstellung der Channel des Dienstes (`data.channels[0].options`, `data.channels[1].options`, ... ) beinhalten.

Die Art der Darstellung im Edit-Mode für alle anderen WOAD-Daten (z.B. für die geometrischen Abmessungen) legt das WOAS-Portal selbst bzw. die Testumgebung fest. Diese Darstellungen (HTML-Code) sind nicht Bestandteil des Dienstes.

### 3.4.4 Aufrufschnittstelle des WOAD-Basisobjekts

Die JavaScript-Datei <class>.js beinhaltet die Implementierung der jeweiligen WOAD Class. Diese WOAD Class erweitert ein bereits im WOAS-Kernel verfügbares Meta-Objekt serviceBase und belegt dessen Attribute bzw. überschreibt die zugehörigen Methoden. Durch die Nutzung eines Meta-Objekts für alle WOAS-Dienste wird ein einheitliches WOAD-Interface als JS-Aufrufschnittstelle geschaffen. Das WOAS-Portal kann deshalb jederzeit mit weiteren Diensten für andere Funktionalitäten durch einen Dritten (Publisher) in konsistenter Art und Weise erweitert werden. Bild 23 zeigt das zugehörige Klassendiagramm für einen WOAD.



**Bild 23:** WOAD-Klassendiagramm

#### Attribute

Das Basisobjekt besitzt die folgenden in Tabelle 23 aufgeführten Attribute.

**Tabelle 23:** Attribute des serviceBase –Objekts

Name	Beschreibung
data	Beinhaltet das WOAD-Datenobjekt.
stdOptions	Beinhaltet die default-Werte für die spezifischen optionalen WOAD-Daten. Diese werden beim erstmaligen Aufruf einer Instanz benötigt.
stdChannels	Beinhaltet die default-Werte für alle spezifischen optionalen WOAD Channel-Daten. Diese werden beim erstmaligen Aufruf einer Instanz benötigt.
maxChannels	Anzahl der maximalen Ein-/Ausgänge (Channels) des Dienstes
params	Beinhaltet die HTML-Darstellung aller optionalen WOAD-Daten und WOAD Channel-Daten in einem Array mit Strings. Jeder String hat die Form <Parametername>, <HTML-Code für die Darstellung des Parameters>

## Methoden

Über die Methoden des Basisobjekts können Daten zum Dienst geschrieben bzw. vom Dienst gelesen sowie die WOAD-Daten verändert werden. Die wesentlichen Methoden des Basisobjekts, die für eine Testumgebung als Simulation des WOAS-Portals benötigt werden, sind in Tabelle 24 aufgelistet und beschrieben.

**Tabelle 24:** Methoden des serviceBase –Objekts

Name	Parameter	Beschreibung
init	-	Initialisiert die WOAD Instance auf der Webseite und setzt den Parameterdialog für den Dienst. Diese Methode muss nach der Instanziierung aufgerufen werden.
create	htmlDIV: jquery-Selector	Stellt den Dienst in der View innerhalb eines DIV-Tags dar (zeichnet den Dienst). I.d.R. erfolgt in dieser Methode ebenfalls die Zuordnung von Events und Listeners an die channels.
writeChannel	channel, value  channel: OBJECT Channel-Objekt aus dem WOAD-Datenobjekt  value: STRING Wert des zu schreibenden WOAD-Channels	Asynchrones Schreiben eines WOAD-Channels
refreshChannel	channel, value  channel: OBJECT Channel-Objekt aus dem WOAD-Datenobjekt  value: STRING oder BOOL oder INT  Wert des zu schreibenden WOAD-Channels	Listener-Funktion; wird durch den Dienst aufgerufen, wenn sich der Wert eines Output-Channels des Dienstes ändert (nur für WOAD des Typs 1, 3 und 4)
refreshChannelData	data  data: ARRAY  Das Array hat die Form: [[channelObject1,"newValue1"],[channelObject2,"newValue2"]]	Listener-Funktion; wird durch den Dienst aufgerufen, wenn sich der Wert mehrerer Output-Channels des Dienstes ändern. (nur für WOAD des Typs 1, 3 und 4)
changeOptions	-	Muss durch den Dienst immer dann aufgerufen werden, wenn sich ein editierbarer Parameter des Dienstes geändert hat. Die Methode wird zur Mitteilung an externe Programme (z.B. das WOAS-Portal) benötigt.
optionsChanged	-	Wird extern (z.B. durch das WOAS-Portal) aufgerufen, wenn sich ein optionaler Parameter des Dienstes geändert hat

selected	isSelected isSelected: BOOL Zeigt mit TRUE an ob der Dienst im Edit-Mode ausgewählt ist.	Muss aufgerufen werden, wenn der Dienst in den Edit-Mode geschalten wird, damit Event-Handler die geänderten Parameter in das data-Objekt schreiben können.
runModusChanged	state state: BOOL Zeigt mit TRUE an, dass der Edit-Mode aufgerufen wurde.	Wird extern (z.B. durch das WOAS-Portal) aufgerufen, wenn sich die Betriebsart (Run/Edit) ändert.
resize	width, height Werte in Pixel	Wird extern (z.B. durch das WOAS-Portal) aufgerufen, wenn sich im Edit-Mode die geometrischen Parameter des Dienstes geändert haben.
logError logWarning	errorMsg Verschiedene Meldungen als String	Wird für die Ausgabe von Meldungen in die Browser-Konsole genutzt

Der folgende Quellcode zeigt das WOAD-Basisobjekt (Meta-Objekt):

```
function serviceBase() {
  this.data;
  this.stdOptions;
  this.stdChannels;
  this.maxChannels = 1;
  this.params = [];

  this.create = function(htmlDIV) {}
  this.init = function(){}
  this.writeChannel = function(channel,value){}
  this.refreshChannel = function(channel,value){}
  this.refreshChannelData = function(data){}
  this.changeOptions = function(){}
  this.selected = function(isSelected){}
  this.runModusChanged = function(state){}
  this.optionsChanged = function(){}
  this.resize = function(){}
  this.logError = function(errorMsg){}
  this.logWarning = function(errorMsg){}
}
```

Für Testzwecke kann das serviceBase-Objekt mit weiteren Methoden erweitert sein, die aber nur für einen Einsatz außerhalb eines WOAS-Portals sinnvoll sind. Je nach Art des Dienstes werden nicht immer alle Methoden des Basis-Objekts benötigt.

#### Berichte und Dokumente

- [7] CCAD: WOAD – Interfacebeschreibung. – 09.03.13
- [8] Schroeder, R.: WOAD-Test-Environment. – 07.01.14

## 4.5 WOAS-Demonstratoren

### 4.5.1 WOAS-Demonstrator I

Zielstellung des WOAS-Demonstrators I (“großer” WOAS-Demonstrator) ist der Nachweis der grundsätzlich neuen Funktionen eines WOAS an einer relativ komplexen technologi-

schen Fertigungsanlage. Dazu gehören insbesondere das Web-basierte WOAS-Engineering mit unterschiedlichen Automatisierungsdiensten und verschiedenen Zugängen zu den Prozessdaten der Automatisierungsgeräte sowie der Betrieb nutzerspezifischer WOAS mit Abrechnung von Diensten nach Laufzeitdauer.

### Gerätetechnischer Aufbau

Als WOAS-Demonstrator I (WD I) wird eine vorhandene Anlage im DT zur Montage von Modellautos (Sprinter) genutzt. Die Anlage besteht aus fünf Montage- bzw. Demontagestationen (Bild 24 – [9]). Dazu gehören:

- Station 1: Prüfung der Vollständigkeit der Montageteile auf dem Werkstückträger
- Station 2: Fügen der Achsbaugruppen
- Station 3: Fügen Karosse auf Achsbaugruppe
- Station 4: Demontieren der Karosse
- Station 5: Demontieren der Achsbaugruppe

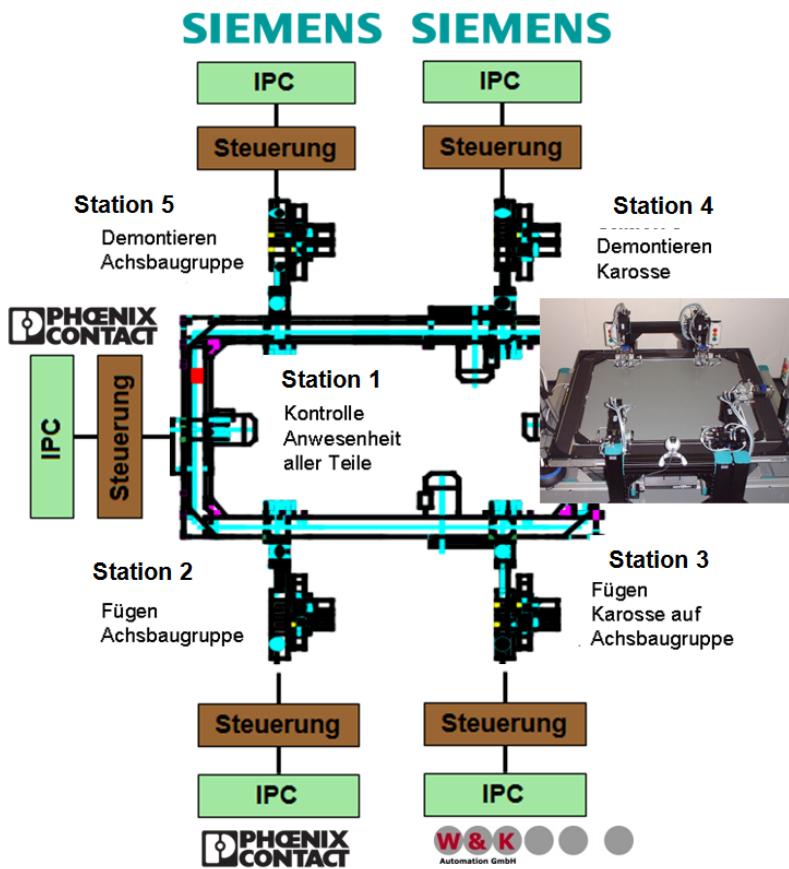


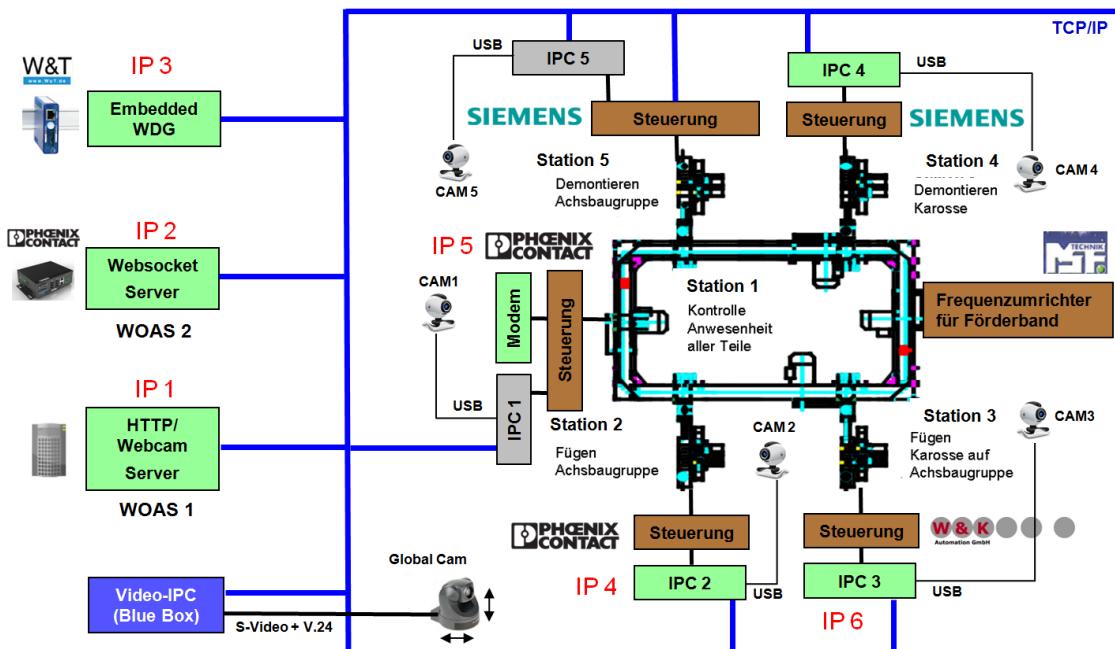
Bild 24: Struktur des WOAS-Demonstrator I

Die Montageteile befinden sich auf einem Werkstückträger, der mittels eines Transportbandes zwischen den einzelnen Stationen bewegt wird. Durch den gewählten Montage/Demontageprozess, der keine Teilezu- und -abführung benötigt, kann die Anlage für den Testbetrieb über das Internet ohne Aufsicht dauerhaft genutzt werden. Die Anlage wurde ursprünglich als Remote Lab im Rahmen eines BMBF-Projekts genutzt und nur von einer Steuerung betrieben. Für das WOAS-Projekt wurde die Anlage steuerungsmäßig komplett umgebaut, so dass jede Station von einer eigenen Steuerung betrieben wird. Es kommen Steuerungen folgender Projektpartner zum Einsatz:

- 2 x Siemens S7-315-PN (Siemens)
- 2 x Phoenix Contact ILC 150-ETH (Phoenix Contact)
- 1 x Horner XL4 (W&K Automation)

Für den geschwindigkeitsgesteuerten Betrieb des Transportbandes wurden zwei Frequenzumrichter des PA-Mitglieds MSF Vathauer eingesetzt.

Für die Remote-Beobachtung des WD I mittels Webbrowser sind fünf Webcams eingesetzt, die den jeweiligen Stationen fest zugeordnet sind. Eine zentrale und steuerbare Webcam (Global Cam) soll zukünftig auch noch einen Gesamtüberblick der Anlage vermitteln. Eine Gesamtübersicht zum gerätetechnischen Aufbau zeigt Bild 25.



**Bild 25:** Gerätetechnischer Komponentenaufbau für den WOAS-Demonstrator I

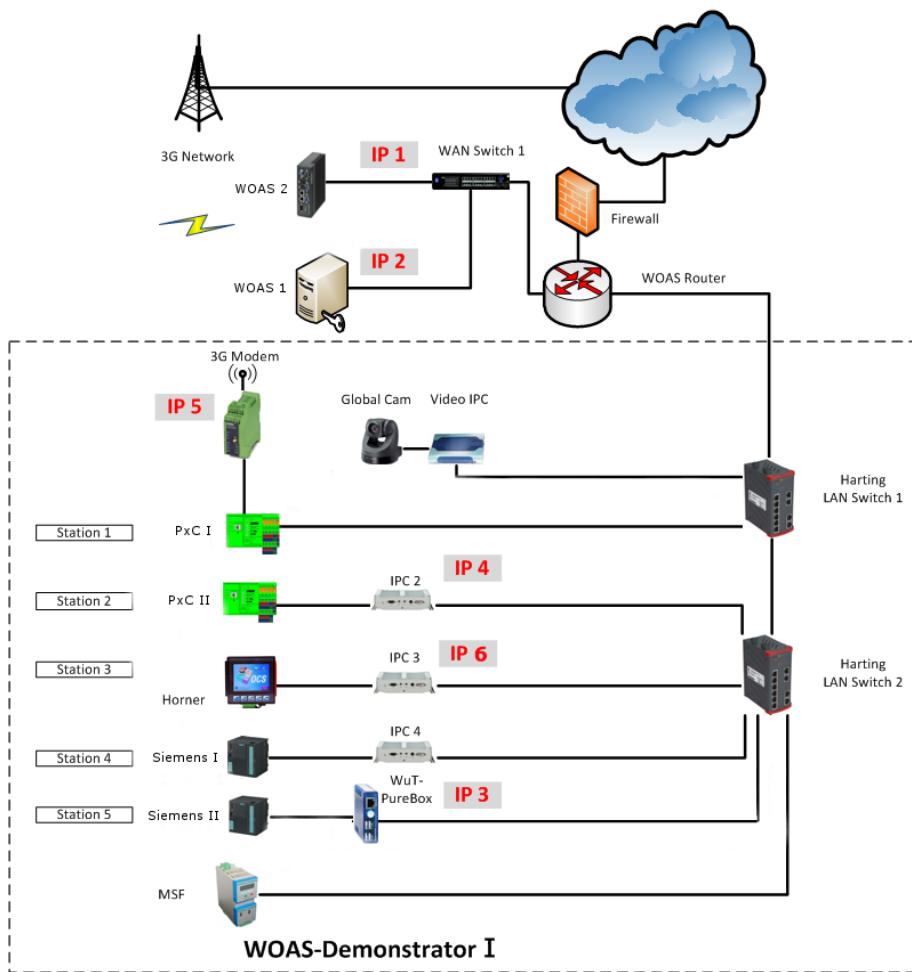
Die erforderliche Anzahl von insgesamt sechs öffentlichen IP-Adressen wird über ein Routing auf entsprechende Ports gelöst.

#### Netzeinbindung

Zentrale Komponenten des Netzwerks sind zwei Switches, die über einen WOAS-Router vom Internet abgeschottet werden (Bild 26). Die beiden LAN-Switches konzentrieren alle lokalen Verbindungen des WOAS-Demonstrators. Es handelt sich um Industrial-Ethernet-Switches des Projektpartners Harting.

Die Netzwerkteilnehmer sind :

- 2 x ILC 150 ETH von Phoenix-Contact
- 1 x IX40 von Horner
- 2 x S7 315 PN von Siemens
- 1 x pure.box von Wiesemann & Theis
- 1 x Umrichter von MSF Vathauer
- 1 x SOB Video-Server von Spectra
- 3 x IPC von Spectra
- 2 x Ha-VIS mCon 3100 von Harting



**Bild 26:** Netzwerkaufbau für den WOAS-Demonstrator I

Die Steuerung der Station 1 kann alternativ über ein 3G-Modem von Phoenix Contact oder den LAN-Switch 1 von Harting per Kabel erreicht werden.

Der WAN-Switch (auch von Harting) ist über den WOAS-Router sowohl mit dem Internet, als auch mit den beiden WOAS-Servern verbunden. Der WOAS-Router realisiert die Sicherheitsinfrastruktur für das gesamte System.

#### Softwaretechnischer Aufbau

Der WOAS-Server wurde aus Performance-Gründen auf zwei Server aufgeteilt (WOAS 1 und WOAS 2). Als Serverrechner sind eingesetzt:

- WOAS 1: Desktop-Rechner von Dell
- WOAS 2: Phoenix Contact IPC

Eine Übersicht zu den Softwarekomponenten des WD I veranschaulicht Bild 27.

WOAS 1 betreibt auch gleichzeitig den Videostreaming-Server für die fünf Webcams der Stationen. Durch die Aufteilung auf zwei Server sind alle WOAS-Webelemente über Port 80 zugänglich. Für die Hauptelemente eines WOAS wird deshalb kein weiterer Port benötigt.

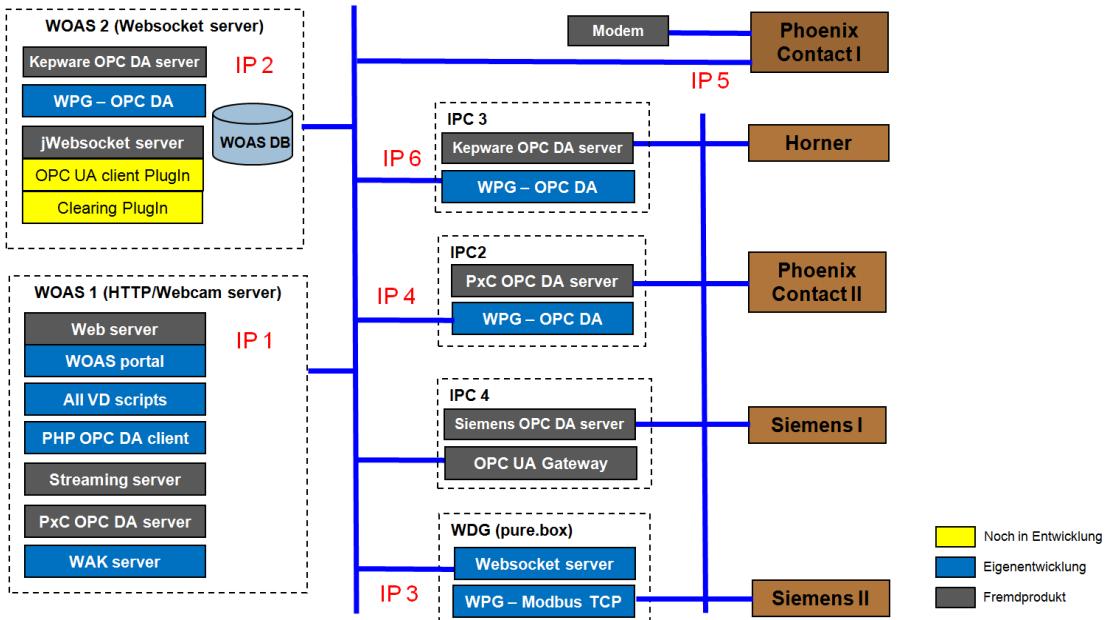


Bild 27: Softwarekomponenten des WOAS-Demonstrators I

Ein Schwerpunkt in WOAS sind u.a. unterschiedliche industriegerechte Zugänge zu den Prozessdaten der Automatisierungsgeräte. Tabelle 25 listet die in WOAS entwickelten Gerätezugänge und deren Anwendung in den beiden WOAS-Demonstratoren auf.

Tabelle 25: WOAS-Prozessdatenzugänge

Zugangstyp	Art des Zugangs	Refreshrate pro PD [ms]	Einsatz
PHP/OPC DA	Client-Push	> 100	alle Steuerungen des WD I
WAK/OPC DA	Server-Push mit Applet	> 40	alle Steuerungen der beiden WDs
WPG/OPC DA	Server-Push mit Websockets	> 40	alle Steuerungen des WD I
WPG/TCP	Client-Push	noch nicht ermittelt	Frequenzumrichter im WD I
WPG/Modbus TCP	Server-Push mit Websockets	noch nicht ermittelt	Servoantrieb im WD II
WPG/FBS	Server-Push mit Websockets	> 20	extern mit Phoenix Contact-SPS getestet

In Tabelle 25 bedeuten die Zugangstypen:

- **PHP/OPC DA:** Zugang zu beliebigen OPC-DA-Servern mittels Ajax und PHP im Client-Push-Verfahren (nur Port 80)
- **WAK/OPC DA<sup>4</sup>:** Zugang zu beliebigen OPC-DA-Servern mittels JavaScript und Java-Applet über einen zweiten TCP/IP-Kanal
- **WPG<sup>5</sup>/OPC DA:** Zugang zu beliebigen OPC-DA-Servern mittels JavaScript unter HTML5 über eine WebSocket-Verbindung (WS und Port 80)

<sup>4</sup> WAK – Web Access Kit

<sup>5</sup> WPG – WOAS Protocol Gateway

Mit den realisierten Prozessdatenzugängen lassen sich unterschiedliche Prinzipien für den Internet-Zugang zu Automatisierungsgeräten in WOAS testen. Dazu gehören server-basierte Prozesszugänge sowie client-basierte Prozesszugänge.

Vorteile der *server-basierten Prozessdatenzugänge* (z.B. PHP/OPC DA) sind die bessere Absicherung der Prozessdatenzugänge gegenüber möglichen Angriffen über die Client-Seite. Nachteilig ist allerdings die höhere Serverbelastung durch viele Automatisierungsgeräte und damit auch ein schlechteres Zeitverhalten. Außerdem müssen die Gerätetreiber (OPC DA-Server) vom WOAS-Provider verwaltet werden. Dies entspricht nicht dem eigentlichen Ziel von WOAS.

Bei den *client-basierten Prozessdatenzugänge* greifen die WOAS-Webseiten im Runtime direkt auf die jeweiligen Steuerungen bzw. auf die zugeordneten Gateways zu. Nachteilig ist dies hinsichtlich des sicheren Zugriffs auf die Steuerungen, da die Steuerungen eine öffentliche IP-Adresse besitzen müssen. Für eine Produktivanwendung werden deshalb weitere Sicherheitsmaßnahmen im Übertragungskanal Client – Steuerung benötigt. Geeignete Autorisierung/Verschlüsselungstechniken sind im WDP<sup>6</sup> vorgesehen, konnten aber im WOAS-Projekt wegen fehlender Ressourcen nicht mehr realisiert werden. Vorteile eines client-basierten Zugriffs sind u.a.

- Schnelle Prozessdatenübertragung.
- Der Datenverkehr zu/von den Automatisierungsgeräten belastet nicht die WOAS-Server.
- Die Gerätetreiber einschließlich erforderlicher Gateways werden durch den Nutzer selbst verwaltet.

Ein spezieller Prozessdatenzugang ist mit dem Typ WDG/FBS entstanden. Hierbei realisiert ein IEC-61131-Funktionsbaustein (FBS) einen WebSocket-Server mit flexiblem Zugriff auf die Prozessvariablen eines SPS-Programms. Es ergibt sich ein sehr schneller Prozessdatenzugang (Refresh-Zeit ca. 20 ms) zu den Prozessdaten, ohne dass ein spezielles Gateway benötigt wird. Getestet wurde dieser Zugang mit einer separaten Phoenix Contact-Steuerung ILC 150. Ausführliche Testergebnisse finden sich in [10]

#### 4.5.2 WOAS-Demonstrator II

Der WOAS-Demonstrator II („kleiner“ WOAS-Demonstrator) ergänzt den WOAS-Demonstrator I hinsichtlich bestimmter Techniken und Geräte (Servoantrieb). Er dient gleichzeitig als mobiles Testsystem für die Präsentation auf Ausstellungen, Messen etc.

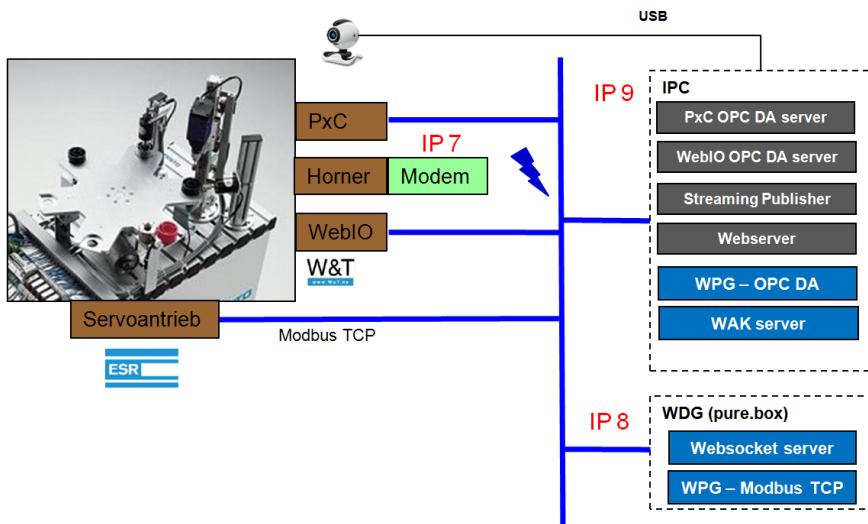
##### Gerätetechnischer Aufbau

Als WD II kommt eine Bearbeitungsstation von Festo Didactic zum Einsatz (Bild 28). In der Station werden Werkstücke auf einem Rundschalttisch geprüft und bearbeitet. Der Rundschalttisch hat 6 halboffene Werkstückaufnahmen und wird originär von einem DC Motor angetrieben. Auf dem Rundschalttisch werden die Werkstücke in 2 parallelen Abläufen geprüft und bearbeitet. Eine Spule mit Tauchanker überprüft, ob die Werkstücke lagerichtig eingelegt sind. Beim Bearbeiten wird das Werkstück durch eine elektrische Spannvorrichtung fixiert. Fertige Werkstücke werden mit der elektrischen Weiche weitergeleitet.

Der ursprünglich vorhandene DC-Motor mit Positionier-Relaischaltung wurde durch einen lagegeregelten Servoantrieb ersetzt.

---

<sup>6</sup> WDP – WOAS Device Protocol



**Bild 28:** Komponentenaufbau für den WOAS-Demonstrator II

Es kommen Automatisierungsgeräte (AG) folgender Projektpartner zum Einsatz:

- 1 x Phoenix Contact Steuerung ILC 150-ETH (Phoenix Contact)
- 1 x Horner-Steuerung mit integriertem Modem XLE (W&K Automation)
- 1 x WebIO (Wiesemann&Theis)
- 1 x Servoantrieb (ESP Pollmeier)

Die Prozessdaten (Steuerung) der Station sind auf die drei AGs aufgeteilt. Für die Remote-Beobachtung des WD II mittels Webbrower besitzt dieser ein fest installierte Webcam.

Im CCAD ist der WOAS-Demonstrator II genau wie der WOAS-Demonstrator I hinter dem WOAS-Router/Firewall angeschlossen (s. Bild 25). Für einen mobilen Betrieb (z.B. auf Ausstellungen oder Messen) des WOAS-Demonstrators II kann dieser über den zentralen LAN-Switch aber auch über andere Netzinfrastrukturen an das Internet angebunden werden.

#### Softwaretechnischer Aufbau

Der WOAS-Demonstrator II nutzt wie der WD I die beiden WOAS-Server WOAS1 und 2. Gleichfalls genutzt werden die für den WD I entwickelten client- und server-basierten Prozessdatenzugänge zu OPC-DA mit entsprechender Parametrierung.

Ein besonderer Schwerpunkt liegt auf der neuen Stationskomponenten Servoantrieb. Nach Auswahl eines geeigneten Servoantriebs musste die Struktur der Station bzw. der technologische Ablauf entsprechend angepasst werden. Außerdem wurde festgelegt, welche Prozessdaten für den Servoantrieb relevant für eine Nutzung durch ein WOAS sind.

Alle erforderlichen Softwarekomponenten im WD II sind auf einem IPC von Spectra installiert.

#### 4.5.3 Dienste

Zum Nachweis des Konzepts der Automatisierungsdienste (WOAD) werden für den WOAS-Demonstrator typische Dienste für den Betrieb eines Automatisierungssystems benötigt. In Anlehnung an die charakteristischen WOAD-Objekttypen aus dem WOAS-Konzept (Bild 18) wurden für die Diensttypen repräsentative Beispiele realisiert.

## Typ 1-Dienst

Typ 1-Dienste sind auf den WOAS-Views nicht sichtbar. Beispiele dafür sind Algorithmen für Messwertverarbeitung, Datenarchivierung und Ablaufsteuerung.

Im Projekt wurde folgender Dienst realisiert:

Bezeichnung:	Input:	Parameter:
Prozessdatenarchivierung	ein beliebiges Prozessdatum	<ul style="list-style-type: none"> <li>• Prozessdatenwert / WOAS-Kanal</li> <li>• Zeitpunkt der Abspeicherung</li> <li>• Prozessdatenname</li> <li>• Quelle (Automatisierungsgerät)</li> </ul>

Die Abspeicherung erfolgt in einer MySQL-Datenbank. Die Datenbank ist nicht identisch mit der WOAS-Datenbank und ist auf einem anderen (Dienste-) Server installiert. Die Autorisierung für diese Datenbank erfolgt mit den WOAS-Nutzerdaten. Es handelt sich um einen serverseitigen WOAD, der clientseitig nur einen WOAD-Stub (JS-Dienstinterface) besitzt.

## Typ 2-Dienst

Typ 2-Dienste dienen der Visualisierung von Prozessdaten. Sie besitzen deshalb auch nur Input-Daten

Im Projekt wurden folgende Dienste realisiert:

Bezeichnung:	Input:	Parameter:
<i>LED-Faceplate</i>	Max. 6 digitale Prozessdaten	<ul style="list-style-type: none"> <li>• Anzahl der LEDs</li> <li>• Anordnung der LEDs</li> <li>• LED-Typ, Farbschema etc.</li> <li>• WOAS-Kanäle</li> <li>• LED-Texte</li> <li>• Faceplate-Text</li> </ul>

Die Erstellung des Faceplates erfolgte mit HTML5.

Bezeichnung:	Input:	Parameter:
<i>Plotter-Faceplate</i>	zwei analoge Prozessdaten	<ul style="list-style-type: none"> <li>• 12 unterschiedliche Parameter (s. Plotterbeschreibung)</li> </ul>

Die Prozessdaten werden in Echtzeit im Plotter angezeigt. Der Plotter nutzt die Plotting Library for jQuery.

Bezeichnung:	Input:	Parameter:
<i>2D-Objekt-Faceplate</i>	Max. vier digitale Prozessdaten	<ul style="list-style-type: none"> <li>• WOAS-Kanäle</li> <li>• Objektanordnung/-anzahl</li> <li>• Farbschema</li> <li>• Faceplate-Text</li> </ul>

Die Erstellung des Faceplates erfolgte mit SVG. Das Faceplate dient zur dynamischen grafischen 2D-Darstellung der jeweiligen Stationen im WD I.

<b>Bezeichnung:</b>	<b>Input:</b>	<b>Parameter:</b>
<i>Image LED</i>	ein digitales Prozessdatum	<ul style="list-style-type: none"> <li>• WOAS-Kanal</li> <li>• Image zur Darstellung von 0/1</li> <li>• Farbschema</li> <li>• Bezeichnung</li> </ul>

<b>Bezeichnung:</b>	<b>Input:</b>	<b>Parameter:</b>
<i>Analog Display</i>	ein analoges Prozessdatum	<ul style="list-style-type: none"> <li>• WOAS-Kanal</li> <li>• Farbschema</li> <li>• Darstellungsformat</li> <li>• Bezeichnung</li> </ul>

Die Erstellung des Ausgabedisplays erfolgte mit HTML4.

### Typ 3-Dienst

Typ 3-Dienste dienen der Beeinflussung von Prozessdaten. Sie besitzen deshalb i.d.R. nur Output-Daten, für Rückmeldequittungen evtl. auch Input-Daten.

Im Projekt wurden folgende Dienste realisiert:

<b>Bezeichnung:</b>	<b>Output:</b>	<b>Parameter:</b>
<i>Input-Feld</i>	ein analoges Prozessdatum	<ul style="list-style-type: none"> <li>• WOAS-Kanal</li> <li>• Farbschema</li> <li>• Darstellungsformat</li> <li>• Bezeichnung</li> </ul>

Die Erstellung des Input-Feldes erfolgte mit HTML4.

<b>Bezeichnung:</b>	<b>Output:</b>	<b>Parameter:</b>
<i>Leuchtdruckschalter-Faceplate</i>	ein digitales Prozessdatum	<ul style="list-style-type: none"> <li>• WOAS-Kanal</li> <li>• Auflösung, Anordnung etc.</li> <li>• Farbschema</li> <li>• Bezeichnung</li> </ul>

Das Faceplate dient zum Einschalten eines Aktors mit Rücklesen des Schaltsignals.

### Typ 4-Dienst

Typ 4-Dienste sind komplexere Dienste und eigentlich eine Mischung aus Typ 2 und Typ 3-Dienste.

Im Projekt wurde folgender Dienst realisiert:

Bezeichnung:	Input:	Output:	Parameter:
Table-HMI	max. 16 digitale/analoge Prozessdaten	max. 16 digitale/analoge Prozessdaten	<ul style="list-style-type: none"> <li>WOAS-Kanäle</li> <li>Anzahl der Elemente</li> <li>Anordnung der Elemente, Farbschema, Texte</li> </ul>

Die Erstellung des Tabellen-Dienstes erfolgte mit HTML5. Damit können digitale und analoge Prozessdaten in Tabellenform gelesen und geschrieben werden.

Der Umfang der Dienste wird permanent durch weitere studentische Projektarbeiten erweitert. Aktuell befinden sich noch folgende Dienste in der Implementierung:

*Slider-Dienst, Knob-Dienst, Bar-Dienst, Big Data-Store-Dienst, Big Data-Analyse-Dienst*

#### Berichte und Dokumente

[9] CCAD: Konzept: WOAS-Demonstratoren. – 26.07.13

[10] Rojas-Pena, L.F.: Implementation and testing of a Web Socket Access to a Phoenix Contact PLC. – 05.05.14

## 4.6 WOAS-Integrationsportal

Das WOAS-Integrationsportal (kurz: WOAS-Portal) ist der Prototyp einer Integrationsplattform, um Automatisierungsgeräte mit Automatisierungsdiensten zu ausführbaren Automatisierungssystemen zu verbinden. Es dient gleichzeitig als Runtime-Umgebung und Engineering-Werkzeug. Der Prototyp bietet eine Basis-Funktionalität, um die Eigenschaften eines WOAS testen zu können. Es ist nicht für einen Produktivbetrieb vorgesehen.

Das Portal kann über die folgenden beiden IP-Adressen erreicht werden:

<http://woas.ccad.eu>

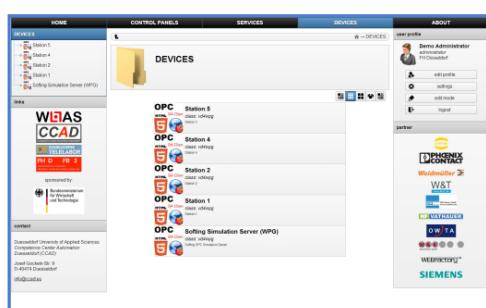
<http://woastest.ccad.eu>

### 4.6.1 Aufbau der Benutzeroberfläche

Das WOAS-Portal ist unter Berücksichtigung eines responsive Design aufgebaut und kann prinzipiell sowohl auf Desktop-PCs wie auch auf mobilen Geräten mit beliebigen Webbrowsern genutzt werden.

#### Darstellung als Webseite

Auf großen Desktop-Monitoren wird das Portal, wie in Bild 29, dargestellt.



**Bild 29:** Darstellung des WOAS-Portals als Webseite bei einer hohen Auflösung.

Am oberen Rand befindet sich die Hauptnavigation in Form einer Menüleiste. Auf der linken und rechten Seite befinden sich Widgets zur Navigation, Verbindungsstatus und Einstellungen.

Bild 30 zeigt wie das Portal auf kleineren Desktop-Monitoren (Auflösung  $y > 1280$  Pixel) aussieht. Die Inhalte bleiben erhalten, allerdings werden Breiten, Abstände und Ränder an die Bildschirmauflösung angepasst.

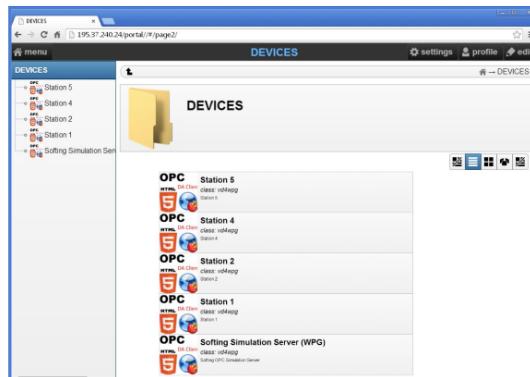


**Bild 30:** Darstellung des WOAS-Portals als Webseite bei einer niedrigen Auflösung.

### Darstellung als Anwendung (App)

Neben der Darstellung als "klassische Webseite" kann die Weboberfläche auch als Anwendung dargestellt werden. Bild 31 zeigt, wie die Oberfläche auf größeren mobilen Bildschirmen, wie sie Beispielsweise bei Tablets und Netbooks verbaut sind, dargestellt wird (Auflösung  $1024 < y < 1280$ ).

Die Hauptnavigation verbirgt sich hier hinter einer Schaltfläche und wird dem Benutzer bei Bedarf als Popup angezeigt. Wichtige Widgets wie Login, Einstellungen und Verbindungsstatus werden wie auch das Menü als Popups dargestellt. Andere Widgets, die für den Benutzer nicht direkt von Interesse sind werden ausgeblendet, um den zur Verfügung stehenden Bildschirmplatz optimal nutzen zu können. Die Subnavigation ist wie auch bei der Darstellung als Webseite am linken Bildschirmrand angeordnet.



**Bild 31:** Darstellung des WOAS-Portals als Anwendung bei einer hohen Auflösung (z.B. Google Nexus 7)

Bild 32 zeigt, dass auf kleineren mobilen Bildschirmen, wie sie bei Smartphones verbaut werden, die Texte einiger Schaltflächen ausgeblendet und die Subnavigation nur bei Bedarf eingeblendet wird (Auflösung  $< 1024$ ).



**Bild 32:** Darstellung des WOAS-Portals als Anwendung bei einer niedrigen Auflösung (z.B. iPhone 5).

#### 4.6.2 Verwendung und Betrieb

##### Benutzerrollen

Das WOAS-Portal ist ein multimandantenfähiges und rollenbasiertes Portal. Benutzer können die in Tabelle 26 dargestellten Rechte besitzen.

**Tabelle 26:** Nutzerrollen im WOAS-Portal

Name	Beschreibung	Rollenfestlegung durch
Provider (Host)	Betreibt das WOAS-Portal	-
Customer-Admin	Der Nutzer des Kunden, der ein nutzerspezifisches WOAS projektiert.	Provider
Customer-User	Der Nutzer des Kunden, der ein nutzerspezifisches WOAS bedient und betreibt.	Customer-Admin
Publisher	Kann neue Automatisierungsdienste (WOAD) im Portal anbieten.	Provider
Manufacturer	Kann neue Gerätezugänge im Portal anbieten.	Provider
Guest	Hat Zugang zu ausgewählten Test-WOAS	Provider, Customer-Admin

Unabhängig von der Rolle muss sich jeder Nutzer zuerst im Portal registrieren. Danach erfolgt entsprechend Tabelle 26 die Zuordnung der Rollen durch die jeweiligen Verantwortlichen.

Für den Test des Portals mit Beispielen des CCAD steht folgender Zugang als Customer-User frei zur Verfügung:

Nutzername: user@ccad.eu  
Passwort: woasTest

Nach dem Einloggen kann dieser Nutzer die vom CCAD bereitgestellten Control Panels nutzen. Aktuell verfügbar sind dabei der Bereich „Demo“ mit zwei Control Panels und der WOAS-Demonstrator (Montageanlage für Modellautos im DT) mit sechs Control Panels. An den Test-Panels wird weiterhin gearbeitet, so dass sich diese verändern können.

Daüber hinaus steht folgender Customer-Admin und Customer-User Zugang für allgemeine und öffentliche Testzwecke zur Verfügung:

#### Customer-Admin

Nutzername: admin1@ccad.eu  
Passwort: SeXqGG8g

#### Customer-User

Nutzername: user1@ccad.eu  
Passwort: NFfkuKKs

In diesem Testzugang steht ein Control Panel („DEMO“) mit zwei Views zur Verfügung, auf denen eigene Operator Panels (WOAS-Views) projektiert werden können. Zur Geräteanbindung steht ein VD für die Anbindung eines Softing Demo OPC Servers mit sechs digitalen bzw. analogen simulierten Prozessdaten sowie die Station 2 aus dem WOAS-Demonstrator mit ausgewählten Prozessdaten zur Verfügung.

#### Seiten und Bereiche des Portals

Nach dem Einloggen als Customer-Admin stehen die folgenden Bereiche zur Verfügung.

- |                 |   |
|-----------------|---|
| HOME:           | Informiert über das Projekt und Partner. Stellt WOAS-Dokumente zur Verfügung.                                       |
| CONTROL PANELS: | Dient zur Projektierung und Nutzung von Control Panels eines WOAS (Bedien-/Visualisierungspansels, HMI, DCS, SCADA) |
| SERVICES:       | Auflistung der verfügbaren und verwendeten WOAD (Services).<br>Ermöglicht die Parametrierung der WOAD               |
| DEVICES:        | Verwalten der verfügbaren Virtual Devices (Geräte). Ermöglicht die Erzeugung/Lösung und Parametrierung von VD       |

Loggt man sich als Customer-User ein, ist lediglich die Seite CONTROL PANELS verfügbar. Die Projektierung eines WOAS (Editieren) ist mit dieser Rolle nicht möglich.

#### Devices

Für den Customer-Admin steht die Seite DEVICES zur Verfügung. Hier findet sich eine Übersicht aller verwendeten VD (Geräte). Im Prototyp des Portals stehen die inn Tabelle 27 aufgeführten VDs zur Verfügung:

**Tabelle 27:** Verfügbare VDs im Prototyp des WOAS-Portals

Name	Beschreibung	Anzahl der analogen Prozessdaten	Anzahl der digitalen Prozessdaten
Softing Simulation Server	Softing OPC Demo Server als Automatisierungsgerät. Kommunikation zum Client über Websockets.	4	2
Station 1	Phoenix Contact OPC Server zur Steuerung der Station 1 des WD I. Kommunikation zum Client über Websockets.	1	23
Station 2	Phoenix Contact OPC Server zur Steuerung der Station 2 des WD I. Kommunikation zum Client über Websockets.	-	32
Station 4	Siemens OPC Server zur Steuerung der Station 4 des WD I. Kommunikation zum Client über Websockets.	-	25
Station 5	Siemens OPC Server zur Steuerung der Station 5 des WD I. Kommunikation zum Client über Websockets.	-	29

Wählt man ein Gerät über die Haupt bzw. Seitennavigation aus, wird neben der Parametrierung auch die bereits eingebundenen Kanäle und deren aktueller Zustand angezeigt (Bild 33). Alle schreibbaren Kanäle können in dieser Ansicht auch beschrieben werden.

Über eine Schaltfläche am rechten Bildschirmrand lässt sich das System in den „Editiermodus“ schalten. Hier können die spezifischen VD-Daten sowie die Channel-Daten parametriert werden. Über die Schaltflächen „add“ und „delete“, lassen sich auch neue Geräte hinzufügen und entfernen.

**Bild 33:** Vorkonfiguriertes DEVICE

## Services

Die Seite SERVICES listet die in den nutzerspezifischen Control Panels des Customer verwendeten WOAD strukturiert auf und ermöglicht ihre Parametrierung. Außerdem werden hier auch die Abrechnungsinformationen angezeigt.



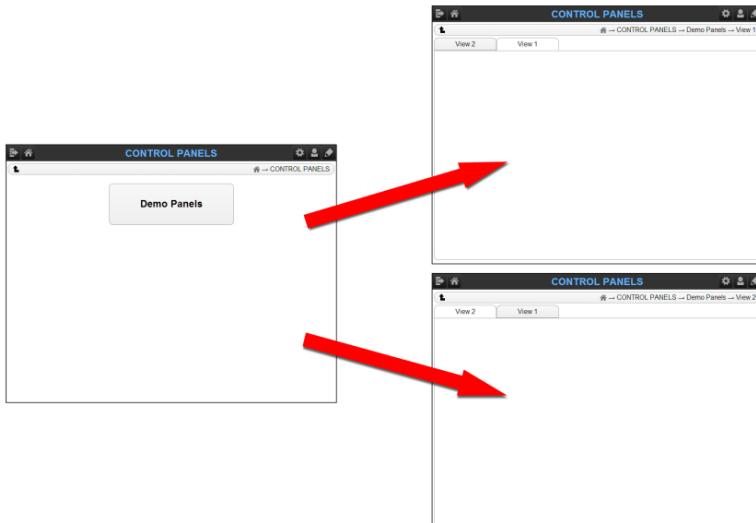
**Bild 34:** Auflistung und Parametrierung der SERVICES

Aktuell steht im Prototyp des Portals die Editier-Funktion für die WOAD sowie die Abrechnungsinformation noch nicht zur Verfügung. Die verwendeten Dienste werden nur strukturiert aufgelistet (Bild 34).

## Control Panels

Über die Seite CONTROL PANELS (auch „Views“) lassen sich diverse Dienst-Instanzen (WOAD) erzeugen/löschen, parametrieren und mit Virtual Devices verbinden.

Die Seite gliedert sich in verschiedene Arbeitsbereiche (workspaces) und Ansichten (views). Bild 35 veranschaulicht diese Gliederung.



**Bild 35:** Arbeitsbereiche und Ansichten in CONTROL PANELS

Aktuell lassen sich im Prototyp des Portals noch keine Arbeitsbereiche erzeugen. Neue Control Panels und Views können jedoch erzeugt und gelöscht werden. Für die Testzugänge

sind jeweils ein Control Panel mit je zwei leeren Views bereits eingerichtet. Diese können für die Projektierung eines Test-WOAS verwendet werden.

#### Channel Connection Groups

Im Gegensatz zu klassischen SCADA/HMI-Systemen, bei denen zur Projektierung den einzelnen Funktionalitäten/Elementen eindimensional bestimmte Prozessdaten zugeordnet werden, erfolgt in WOAS während der Projektierung der Aufbau von *Channel Connection Groups* (CCG). Eine CCG beinhaltet alle Service-Channels eines Panels, die mit einem bestimmten VD-Channel verbunden sind. Damit können ereignisbasiert Änderungen in einem bestimmten Channel an alle mit diesem Channel verbundenen Dienste bzw. VDs übermittelt werden.

[11] beschreibt ausführlich die Nutzung des WOAS-Portals. Das Dokument kann von der Homepage des Portals heruntergeladen werden.

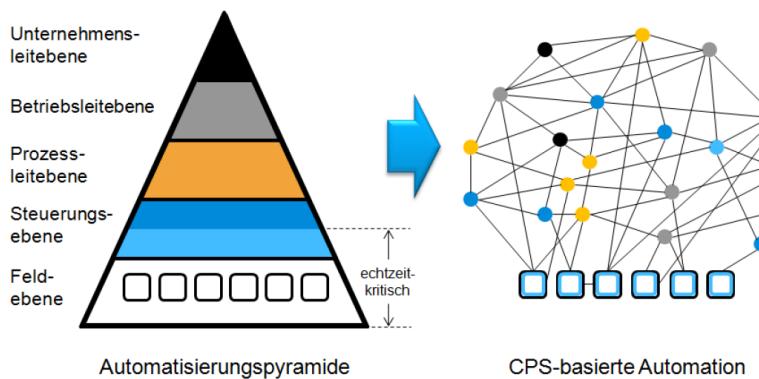
#### Berichte und Dokumente

[11] CCAD: WOAS-Integrationsportal – Nutzerhinweise. – 12.05.14

## **4.7 Mieten von Automatisierungsdiensten**

Klassische Automatisierungssysteme wie z.B. PLS, SCADA nutzen bisher ein lizenziertes Geschäftsmodell mit weiteren kostenpflichtigen Updates während der Betriebszeit der Systeme. Die tatsächliche Nutzung dieser Systeme beim Kunden spielt bei diesem Geschäftsmodell keine Rolle. Es sind proprietäre und herstellerspezifische Systeme, die eine Integration von Automatisierungsfunktionen dritter Anbieter nicht zulassen. Eine Reihe neuerer Arbeiten beschäftigen sich zwar mit der Entwicklung von Dienststrukturen für Automatisierungssysteme auf Basis einer SOA-Struktur. Zielstellung ist dabei aber insbesondere die Ausrüstung von Geräten mit embedded Diensten. Auch die Nutzung von Cloud-Technologien für den Betrieb von Automatisierungsdiensten wird zunehmend thematisiert und im Bereich der Gebäudeautomatisierung, aber auch für Maschinenoptimierung bereits als Geschäftsmodell im Sinne einer *Platform as a Service* (PaaS) angeboten. Diese Systeme sind aber geschlossen und Dritte können keine eigenen Dienste anbieten. Die Geschäftsmodelle basieren meist auf einer pauschalierten Nutzung der zugehörigen Plattform und Abrechnung nach Anzahl der betriebenen Geräte bzw. des generierten Datenstroms. Eine Abrechnung nach Automatisierungsdiensten ist nicht vorgesehen.

Nach "Frost & Sullivan" wird in der Fabrik der Zukunft die Betonung auf das Produktpotfolio der Industrieautomatisierung voraussichtlich in den nächsten Jahren abnehmen. Der Bedarf an globalisiertem Service & Support dürfte hingegen, zusammen mit den Kostenfaktoren, an Bedeutung gewinnen. Die Einführung von sicheren Cloud-Computing-Technologien ermöglicht dabei den Zugang zu relevanten strategischen Daten über das Internet, mit deren Hilfe Entscheidungen in Echtzeit gefällt werden können sowie die Betriebseffizienz gesteigert werden kann. Im Zuge dieser Entwicklung wandelt sich die klassische Automatisierungshierarchie zunehmend zu einer flachen Automatisierungs"wolke", in der neue Technologien/Paradigmen wie z.B. ServiceOrientierte Architekturen (SOA) eingesetzt werden (Bild 36).



**Bild 36:** CPS-basierte Automation<sup>7</sup>

Die Zielstellungen für eine solche „Fabrik der Zukunft“ sind u.a. in den Umsetzungsempfehlungen Industrie 4.0 der Forschungsunion formuliert. Danach ist es eine wichtige Aufgabe, die neuen Technologien auch zu nutzen, um neu entstehende Wertschöpfungsnetze zu gestalten. Dies beinhaltet u.a. die Entwicklung neuer Geschäftsmodelle für die Produktionsautomatisierung auf der Basis von service-orientierten Strukturen unter Nutzung von Cyber Physical Production Systems (CPPS) als Plattform, um damit insbesondere Produkte mit geeigneten Dienstleistungen zu verknüpfen.

#### 4.7.1 Das Projekt WOAS als CPS-Integrationssystem

Betrachtet man das ursprüngliche Forschungsziel von WOAS (formuliert ca. 2010) zur Schaffung einer offenen und flexiblen Webtechnologie-basierten Systemarchitektur, die über systematische und wohldefinierte Schnittstellen verfügt, um Hersteller, Anbieter und Anwender von Automatisierungsdiensten auf einer Plattform zusammenzuführen, so wird klar, dass das WOAS-Portal unter der neuen Begriffswelt von Industrie 4.0 und CPS (seit 2012) im Prinzip eine CPS-integrationsplattform bildet. Damit sind auch die in Industrie 4.0 geforderten bzw. angesprochenen neuen und smarten Geschäftsmodelle über die Nutzung von in IP-Netzen verteilten Automatisierungsfunktionen als Dienste möglich.

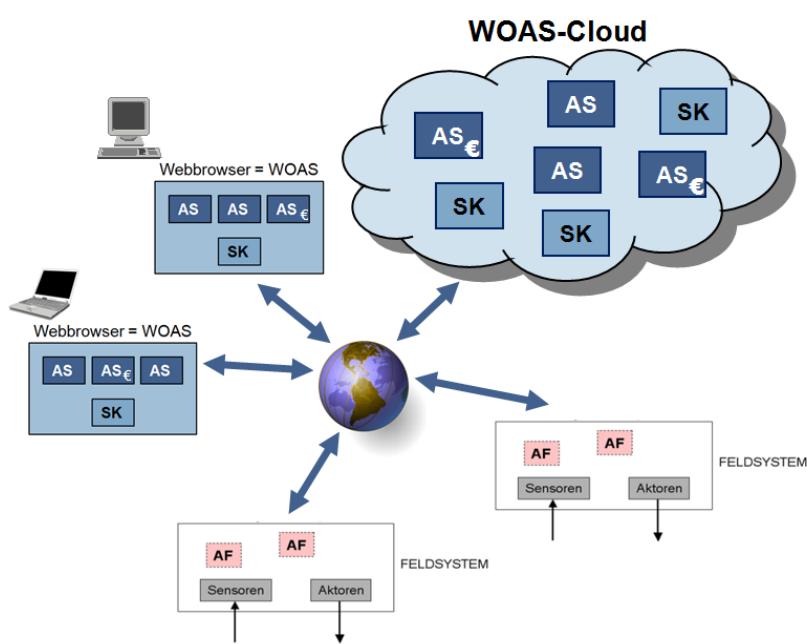


Bild 37 veranschaulicht nochmals die Prinzipstruktur eines WOAS, so wie diese bereits im Projektantrag formuliert wurde.

**Bild 37:** Prinzipstruktur von WOAS

<sup>7</sup> nach VDI/VDE-GMA: Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation, April 2013

Das Automatisierungssystem wird erst zur Betriebszeit aus einer Bauvorschrift (SK = Systemkern) im Webbrowser eines Nutzers erzeugt und dort ausgeführt. Die erforderlichen Automatisierungsfunktionen (AF) werden als Automatisierungsdienste (AS) über eine WOAS-Cloud genutzt. Der Anschluss der Feldsysteme (Automatisierungsgeräte) erfolgt über Standard-Industrieschnittstellen mit Webanbindung. Alle Elemente eines WOAS einschließlich des Engineering-Systems (WOAS-Creator) sind über das Web zugreifbar. Vergleicht man Bild 36 mit der CPS-basierten Automation in Bild 35, so bilden die Feldsysteme in WOAS die CPS-Komponenten, die nun flexibel mit den AS als CPS-Dienste verbunden werden können. Das zugehörige WOAS-Webportal bildet damit eine CPS-Integrationsplattform, in der beliebige CPS mit beliebigen Diensten verbunden werden können.

#### 4.7.2 Wohldefinierte Schnittstellen als Basis eines smarten Geschäftsmodells

Neue Geschäftsmodelle wie sie in Industrie 4.0 vorgesehen sind, sollen auf Smart Services basieren. Ein Smart Service besitzt u.a. folgende Kennzeichen<sup>8</sup>:

*„Smart Services sind technische Systeme die bewusst in Produkten oder Dienstleistungen implementiert werden zum Zwecke der Erfüllung spezifischer Aufgaben speziell im Umfeld von IT basierten Dienstleistungen. Sie sind gekennzeichnet durch eine Verbindmöglichkeit zu einer technischen geeigneten Serviceinfrastruktur, ...“*

*„Sie können als technisches Dienstleistungssystem in einer Ausprägungsbandbreite bis hin zum intelligenten autonomen System entweder **beim Endkunden direkt oder indirekt mit Hilfe des Dienstleistungsanbieters** wirksam werden.“*

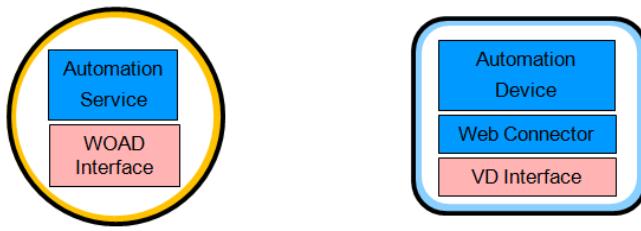
*„Entwickler von Smart Services suchen mit Hilfe eines ständigen Diskurses mit den Smart Service Anspruchsgruppen diese angeführten Entwicklungs- und Anwendungsprinzipien zu verbessern und **transparent offen zu legen**.“*

Will man Automatisierungsfunktionen als Smart Services im Rahmen eines Geschäftsmodells anbieten, muss man also geeignete wohldefinierte Schnittstellen schaffen, diese transparent und offen legen, um Drittanbietern auch die Möglichkeit zu geben, ihre Geschäftsmodelle an diese Schnittstellen anzupassen zu können. Im Bereich der IT-Industrie ist dies seit einigen Jahren bereits Standard und insbesondere mit dem Aufkommen der Cloud-Technologien werden verschiedene Dienstleistungsszenarios (Software as a Service, Platform as a Service, Infrastructure as a Service) mit stark steigender Tendenz genutzt. In der klassischen Automatisierungsindustrie sind diese Dienstleistungsmodelle bisher weitgehend unbekannt bzw. aus verschiedenen Gründen schwierig zu realisieren oder auch nicht gewollt. Nur in der Gebäudeautomation gibt es seit einiger Zeit Bestrebungen, auch Funktionen als Web-basierte Dienste aus einer Cloud anzubieten (z.B. DigiCloud, Linemetrics Analyse Platform), dies aber i.d.R. im Rahmen eines geschlossenen Systems. Für Dritte ist eine Integration eigener Dienste nicht möglich.

Da ein Hauptziel von WOAS die Schaffung der Grundlagen für ein neues und dienstorientiertes Betreibermodell ist, wurden wohldefinierte und transparente Schnittstellen sowohl für die Dienste (WOAD) wie auch für die Feldsysteme (CPS-Komponenten) geschaffen (Bild 38).

---

<sup>8</sup> nach Aschbacher, H: Smart Services. - Fachhochschule CAMPUS 02,  
[http://www.serviceengineering.at/blog/?page\\_id=274](http://www.serviceengineering.at/blog/?page_id=274)

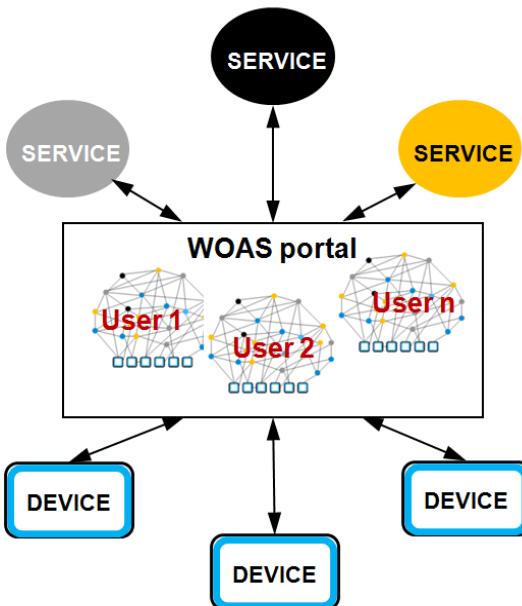


a) Automatisierungsdienst    b) Feldsystem (CPS-Komponente)

**Bild 38:** Automatisierungsdienst (a) und Feldsystem (b) als unabhängige Komponenten mit wohldefinierten Schnittstellen

Beide Schnittstellen sind ähnlich aufgebaut, basieren im Wesentlichen auf der Übertragung von Prozessdaten und nutzen ein Channel-Konzept. Virtual Devices (VD) bilden Feldsysteme (Automatisierungsgeräte) in der WOAS-Cloud ab und können in einheitlicher Art und Weise über ihre Channels mit Automatisierungsdiensten über das WOAD-Interface verbunden werden.

Durch die Kapselung der Dienste und der Geräte in unabhängige und vernetzbare Komponenten mit definierten Web-basierten Schnittstellen können nun in einer Integrationsplattform (WOAS-Portal) diese Komponenten miteinander zu einem Funktionssystem (z.B. SCAD-System, Operator-Panel, Prozessvisualisierungssystem) verbunden werden. Bild 39 verdeutlicht das Gesamtsystem.



**Bild 39:** Integration von Services und Geräten zu nutzer-spezifischen Lösungen im WOAS-Portal

Das Projekt WOAS separiert damit die Automatisierungsfunktionen auf der Anwendungsebene als Dienste vom Kern eines Automatisierungssystems und lässt über offengelegte Schnittstellen die Integration von weiteren Diensten, aber auch von weiteren Geräte-Zugängen durch Drittanbieter zu.

Damit daraus tatsächlich ein neues Geschäftsmodell entsteht, muss aber die entsprechende Integrationsplattform noch drei weitere Bedingungen erfüllen:

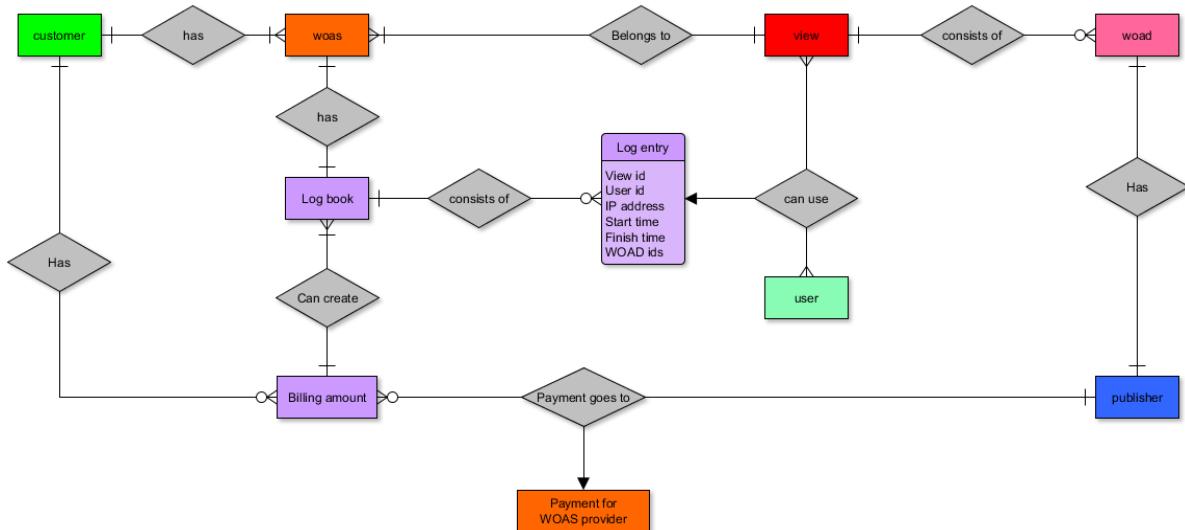
1. Sie muss zur Bedienung vieler Kunden Multi-Mandanten-fähig sein.
2. Sie muss rollen-basiert sein, um den unterschiedlichen Rollen der Nutzer gerecht zu werden (Systementwickler, Dienstanbieter, Operator usw.).
3. Sie muss ein Abrechnungssystem (Clearing-System) beinhalten, um die Dienstnutzung geeignet abrechnen zu können.

#### 4.7.3 Das WOAS-Clearing-System

Bei der Entwicklung von WOAS wurde das Clearing-System (CS) als integrale Komponente des Gesamtsystems betrachtet und auf Basis eines CS-Modells (Bild 40) in seinen wesentlichen Teilen im Prototyp strukturell bereits berücksichtigt. Das CS benötigt zwingend die unterschiedlichen Rollen eines Nutzers, um die Abrechnungsdaten geeignet zuordnen zu können. WOAS kennt deshalb gemäß Tabelle 26 unterschiedliche Nutzerrollen.

Dienste (WOAD) sind immer Bestandteil einer View (HTML-Seite), so dass die Laufdauer der Views im Client-Rechner die Abrechungszeit bestimmt. Zu jedem nutzerspezifischen WOAS gehört deshalb ein Logbuch, in dem die Laufzeiten der Views mit den darin integrierten WOAD aufgezeichnet werden. Dabei werden folgende Daten im Logbuch eingetragen:

View id	-	Identifikation der View
User id	-	Identifikation des Nutzer
IP address	-	IP-Adresse des Client-Rechners
Start time	-	Startzeit der Anzeige der View im Client-Browser
Finish time	-	Zeit des Verlassens der View im Browser
WOAD id	-	Identifikation der Dienste auf der View (WOAD id 1 ... WOAD id n)



**Bild 40:** Datenmodell des Clearing-Systems

Die Eintragungen im Logbuch erfolgen nur im Runtime-Mode. Entwicklungszeiten durch den Customer-Admin werden nicht berücksichtigt. Das Logbuch dient gleichzeitig als Abrechnungsquittung gegenüber dem Customer und dem Publisher.

Nach jedem Ausloggen eines Nutzers erfolgt eine kumulierte Grobberechnung der zeitabhängigen Kosten für den Customer, so dass er diese Information beim nächsten Login einsehen kann. Dazu führt eine Webanwendung folgende Verarbeitung durch:

Für jedes nutzerspezifische WOAS des Customers werden auf Basis des zugehörigen Logbuches die darin enthaltenen WOAD analysiert (Kostenpflichtigkeit, Höhe der Kosten pro Zeiteinheit) und die Beträge für alle WOAD dieses WOAS aufaddiert. Die ermittelten Beträge werden für alle nutzerspezifischen WOAS des jeweiligen Customers summiert und kumuliert als Attribut des Customers (noch zu zahlender Betrag) abgespeichert.

Zu definierten Zeiten (Abrechnungszeitpunkte) kann aus den WOAS-Logbüchern in Verbindung mit den anderen Clearing-Daten eine detaillierte Abrechnung gegenüber Customer und Publisher erfolgen. Die Logbücher können dazu vom WOAS-Portal heruntergeladen werden, damit die Kostenanalyse und -abrechnung mittels eines externen Programms erfolgen kann.

Das Clearing-System basiert auf der Ermittlung der Laufdauer eines WOAD. Dazu wird die Verbindung zwischen Client und WOAS-Server überwacht. Da die WOAD immer mit einer View verbunden sind und die Laufzeit der View identisch mit der Laufzeit der darin integrierten WOAD ist, reicht es aus, wenn die Laufdauer der View im Browser des Nutzers überwacht wird. Prinzipiell gibt es dazu verschiedene technische Möglichkeiten, die im Einzelnen in [12, 13] erläutert sind. Für das WOAS-Portal ist die Realisierung folgender Variante konzipiert und in Grundzügen umgesetzt:

Für den im WOAS-Server verwendeten jWebSocket-Server wird ein *Clearing-Plugin* erstellt, welches einen spezialisierten Websocket-Server mit einem nur für das Clearing-System genutzten Port realisiert. Auf jeder View wird vom WOAS-Portal automatisch ein Websocket-Client installiert, welcher in regelmäßigen Zeitabständen Weckrufe an den Server versendet (ping). Der Server muss diese Aufrufe quittieren (pong). Ohne erfolgreiche Quittung wird der Dienst im Client gestoppt (Server nicht erreichbar etc.). Die Taktzeit für den ping-pong könnte bei ca. 1/min liegen (hängt vom gewünschten Abrechnungstakt des WOAS-Providers ab).

Das Clearing-System ist strukturell im Prototyp des WOAS-Portals vorbereitet und soll noch 2014 umgesetzt werden.

#### 4.7.4 Geschäftsmodelle auf Basis eines Mietszenarios

Auf der Grundlage des Clearing-Systems können nun unterschiedliche Geschäftsmodelle basierend auf dem Prinzip *SaaS* auch für Automatisierungsdienste realisiert werden.

Prinzipiell kann ein WOAS dabei nach verschiedenen Kriterien gegenüber einem Customer abgerechnet werden. Die Erlöse aus der Dienste- bzw. Gerätenutzung könnten sich WOAS-Provider, Publisher und Manufaturer teilen (Tabelle 28).

**Tabelle 28:** Abrechnungsvarianten für eine WOAS-Nutzung

Art der Abrechnung	Erlöse für WOAS-Provider	Erlöse für Publisher	Erlöse für Manufacturer
Nach Laufzeit der WOAD	X	X	-
Nach statischer Nutzung der WOAD	X	X	-
Nach Laufzeit der Geräte	X	-	X
Nach der Anzahl der integrierten Geräte	X	-	X
Portalnutzung als Pauschale	X	X	X

Zu einem Kostenvergleich für ein WOAS-Geschäftsmodell werden die Kosten für ein klassisches Prozessleit- und –visualisierungssystem zugrundegelegt:

- Lizenzkosten für ein klassisches Prozessleit- und –visualisierungssystem mit Webanbindung (z.B. WinCC/WebNavigator) bei gleichzeitiger Nutzung von 10 Webclients: 10.000 €

Bezogen auf einen geschätzten Nutzungszeitraum von 5 Jahren und Update-Kosten von ca. 500 €/Jahr ergeben sich damit pro Jahr geschätzte Kosten von 2500 €/Jahr. Da es sich um Lizenzkosten handelt, spielt die tatsächliche Nutzungsdauer keine Rolle.

Die Kosten für einen WOAS-Customer sollten im vergleichbaren Fall erheblich darunter liegen z.B. bei 1000 €/Jahr für ein nutzerspezifisches WOAS. Als Problem stellt sich hier die Vergleichbarkeit dar, da zur tatsächlichen Nutzungsdauer von Prozessleit- und –visualisierungssystem bisher keine Untersuchungen veröffentlicht sind. Dies ist stark von den jeweiligen Anlagen und deren Betriebszeit abhängig.

Der Einsatz von WOAS wird sich vorerst zum Betrieb von verteilten Anlagen eignen, die im Regelfall autonom arbeiten und die nicht einer ständigen Prozesskontrolle/-bedienung unterliegen. Abhängig von der täglichen Laufzeit könnten sich z.B. die in Tabelle 29 aufgeführten Stundensätze ergeben. Für die Berechnung werden 250 Betriebstage/Jahr angesetzt. In der dritten Spalte ist der notwendige Abrechungstakt aufgeführt, der für eine Cent-genaue Abrechnung erforderlich wäre.

Für die testweise Abrechnung im Prototyp des WOAS-Portals ist eine Abrechnung ausschließlich über die Zeitdauer der Nutzung der WOAD-Dienste geplant. Berücksichtigt man die Laufdauer mit 8h/Tag für ein nutzerspezifisches WOAS z.B. eines Operator-Panels und nimmt man weiterhin an, dass pro Operator-Panel ca. 10 WOAD-Dienste verwendet werden (Echtzeit-Trend, Knöpfe, Buttons, Eingabefelder etc.) so ergeben sich pro Dienst Abrechnungskosten von ca. 5 Cent/h Laufzeit bzw. Nutzungzeit.

**Tabelle 29:** Stundensätze für den Betrieb eines nutzerspezifischen WOAS bei jährlichen Maximalkosten von 1000 €

Tägliche Laufdauer [h]	Stundensatz [€]	Notwendiger Cent-genauer Abrechungstakt [s]
2	2,00	18
4	1,00	36
8	<b>0,50</b>	<b>72</b>
16	0,125	288

Zur Refinanzierung eines kommerziellen WOAS-Portals könnte später auch ein pauschalierter Monatsbetrag für den Customer gestaffelt nach der Anzahl der Views (bzw. Control Panels) vorgesehen werden z.B.:

< 5 Views	-	kostenfrei
5 ... 10 Views	-	5 €/Monat
pro weitere 10 Views	-	5 €/Monat

In einer erweiterten Version des Clearing-Systems könnte auch für die zeitabhängige Nutzung von Virtual Devices (Geräteanbindung) ein Entgelt erhoben werden.

Für die Umsetzung des Geschäftsmodells zur Vermietung von Automatisierungsfunktionen sind weitere wirtschaftlich-technische Untersuchungen notwendig. Insbesondere ist aktuell die Nutzungscharakteristik von Automatisierungssystemen im Bereich der oberen, nicht-zeitkritischen Automatisierungsebenen (Prozessleitebene, Betriebsleitebene) unbekannt. Erst nach genauer Kenntnis der Kundenanforderungen (Wie oft und wie lange wird ein HMI oder ein SCADA-System eigentlich für eine bestimmte automatisierte Anlage/Maschine benötigt?) lässt sich ein marktkonformes Geschäftsmodell auf Mietbasis aufstellen und in der Praxis erproben.

Für WOAS sind dazu noch folgende zukünftige Schritte vorgesehen:

1. Der Prototyp des WOAS-Portals wird 2014 ausgebaut und evtl. als *Open Community Automation Platform (OCAP)* für die öffentliche Test-Nutzung freigegeben. Die WOAD/VD-Schnittstellen werden gleichfalls veröffentlicht und so aufbereitet, dass Dritte eigene Dienste bzw. auch Geräte-Zugänge zu eigenen Anlagen in das Portal integrieren können. Damit sollen Erfahrungen zur Akzeptanz und zu Problemen gewonnen werden (s. auch Kapitel 7).
2. Gegenwärtig wird der Prototyp des WOAS-Portals durch das CCAD der FH Düsseldorf betrieben. Der Zugang erfolgt über das öffentliche Internet. Zur Berücksichtigung der Sicherheitsproblematik soll das WOAS-Portal als Plug-and-Work-Paket (ähnlich z.B. dem Lernmanagementsystem Moodle) aufbereitet werden, so dass es auch in einem Intranet eines Kunden durch diesen selbst betrieben werden kann.

#### Berichte und Dokumente

[12] CCAD: Sessions in PHP. – 20.01.13

[13] CCAD: Clearing-System für WOAS (WOAS-CS). – 21.01.13

## 5 Projektergebnisse nach Arbeitspaketen

### 5.1 Zusammenfassende Ergebnisse nach Arbeitspaketen

**AP 1:** Untersuchung und Bewertung verfügbarer Web-Technologien und Architekturen hinsichtlich ihrer Eignung für die industrielle Automation

Ergebnisse	Machbarkeitsstudie, Vor-Recherchebericht zum Problemkreis WOAS-Dienste, Vor-Recherchebericht zum Problemkreis WOAS-Devices, Kurzdokumentationen zu einzelnen Fragestellungen, erste Ergebnisse aus Technologietests (z.B. Einsatz von WebSockets und HTML 5), Patentrecherche WOAS- Services, Patentrecherche WOAS Devices, Ergänzung des Berichts zum Stand der Technik.
Pesonaleinsatz	HPA A: 2,5 PM HPA B: 3 PM

**AP 2:** Modellierung eines WOAS

Ergebnisse	Spezifikation des WOAS Device Protocol, Modellnotation eines WOAS einschließlich der einzelnen WOAS-Komponenten
------------	---

Pesonaleinsatz	HPA A: 2 PM HPA B: 2 PM
----------------	----------------------------

#### **AP 3:** Modellierung von WOAD

Ergebnisse	Spezifikation für die WOAD-Typen, Modellnotation für das WOAD-Interface, WOAD-Implementierungsschnittstelle einschl. einer Nutzungsanleitung, Konzept WOAS-Demonstrator
Pesonaleinsatz	HPA A: 1 PM HPA B: 2 PM

#### **AP 4:** WOAS-Editor und Systemgenerator

Ergebnisse	Modellierung und Implementierung eines Demonstrationsbeispiels eines WOAS-Portals einschl. WOAS-Creator für den öffentlichen Test ( <a href="http://woas.ccad.eu">woas.ccad.eu</a> ), Implementierung von acht client-basierten Automatisierungsdiensten wie Echtzeitrend und Bedienelementen,
Pesonaleinsatz	HPA A: 5 PM HPA B: 10 PM

#### **AP 5:** Bewertung von Industrieschnittstellen und Beispieldatenbindung an WOAS

Ergebnisse	Analyse von Industrieschnittstellen entsprechend den Anforderungen des PA, 5 x Schnittstellenmodule als WOAS Virtual Device, verschiedene Test-Webseiten für die unterschiedlichen VDs, Testbericht: „Webbasierte Bereitstellung von OPC-Daten“, Bericht „Zeitmessungen“
Pesonaleinsatz	HPA A: 2 PM HPA B: 3 PM

#### **AP 6:** Funktionsbeispiel und Evaluierung

Ergebnisse	WOAS-Demonstratoren I und II, WOAS-Beispieldienste (WOAD), WOAS-Funktionsbeispiele am WOAS-Demonstrator I (Stationsbedienung, Prozessvisualisierung).
Pesonaleinsatz	HPA A: 1 PM HPA B: 4 PM

#### **AP 7:** Dienstestruktur und Betreibermodelle

Ergebnisse	Clearing-Konzept als Bericht Übersicht zum Session-Konzept in PHP, Integration der wesentlichen Clearing-Komponenten in die WOAS-Datenbank, in die WOAD-Dienste sowie in die Menüstruktur des WOAS-Portals.
Pesonaleinsatz	HPA A: 1 PM

### **AP 8: Management von WOAD**

Ergebnisse	Unterschiedliche öffentliche Zugänge für die Erzeugung/Projektierung und Verwaltung von WOAD im WOAS-Testportal, Dokumentation „Rent an Automation Service - Smarte Geschäftsmodelle für Industrie 4.0“ mit ersten Wirtschaftlichkeitsbetrachtungen
Pesonaleinsatz	HPA B: 1 PM

### **AP 9: Know-How-Transfer und Vorbereitung der wirtschaftlichen Verwertung der Forschungsergebnisse.**

Ergebnisse	verschiedene Pressemitteilungen, 10 Veröffentlichungen in Fachpublikationen und Konferenzen, Aussteller auf der SPS IPC DRIVES vom 27. – 29.11.12 in Nürnberg, Mitarbeit im FA „Cyber Physical Systems“ der VDI/VDE-GMA, 3 x Workshops bei Mitgliedsunternehmen des PA sowie ein internationales Workshop
Pesonaleinsatz	HPA A: 2,5 PM HPA B: 6,5 PM

Es wurden insgesamt 17 PM der Gruppe HPA A und 25,5 PM der Gruppe HPA B eingesetzt. Dies entspricht einer Summe von 42,5 PM. Die Projektvorgabe der Summe von HPA A+B waren 39 PM. Die Abweichungen von ca. 9% resultieren aus der Komplexität der Entwicklungsleistungen und den dazu erforderlichen Erfahrungen insbesondere der Programmierung für komplexe Web-Systeme, die bei anderem Fachpersonal nicht vorhanden war.

Die durchgeführten Arbeiten entsprachen im Wesentlichen den vorgesehenen Arbeitsschritten aus dem Arbeitsplan (Lösungsweg zur Erreichung des Forschungsziels) und waren deshalb, wie geplant, erforderlich und angemessen.

## **5.2 Erfüllungsgrad der Arbeitspakete**

Die folgende Tabelle 30 gibt eine Übersicht über den Erfüllungsgrad der einzelnen Arbeitspakete bezogen auf die Planungen im Projektantrag.

**Tabelle 30:** Erfüllungsgrad der Arbeitspakete

Arbeits-paket	Erfüllungs-grad [%]	Bemerkungen
AP 1	100	-
AP 2	100	-
AP 3	100	-
AP 4	100	Die Implementierung des WOAS-Portals als zentrale Engineering- und Runtime-Umgebung für ein WOAS erwies sich als sehr aufwendig, da die erforderliche objektorientierte JS/Java-Programmierung einschl. der Datenbankanbindung tiefgehende

		Kenntnisse und Fertigkeiten erforderte. Die permanente Synchronisierung aller Systemkomponenten erforderte darüber hinaus zusätzlichen Personalaufwand
AP 5	> 100	Zusätzlich zu den Planungen wurden umfangreiche Zeitmessungen zur Übertragung der Prozessdaten über ein IP-Netz mit der WOAS-Methodik durchgeführt. Außerdem wurden zusätzliche Schnittstellenmodule für die Prozessdatenzugänge entwickelt (insgesamt sechs gegenüber zwei ... drei geplanten).
AP 6	100	Der Ressourcenaufwand für den Aufbau der beiden Demonstratoren war erheblich. Es wurden sechs mechatronische Teilstationen komplett neu elektrisch und softwaretechnisch mit SPS-Steuerungen und IPC aufgebaut (einschl. des Einbaus eines Servoantriebs und neuen Frequenzumrichters). Dies konnte nur durch zusätzliche kostenneutrale studentische Ressourcen im Rahmen von Projektarbeiten und Bachelorarbeiten erfolgen.
AP 7	80%	Es erfolgte die Erarbeitung eines Konzepts für die Bereitstellung und Nutzung von WOAD in einem IP-Netz (Betreibermodell). Die Rahmenstrukturen für ein Clearing-System wurden implementierungstechnisch im WOAS-Portal umgesetzt. Eine Erprobung konnte wegen nicht ausreichenden Personalressourcen im Projekt noch nicht erfolgen, da im AP 4 mehr Ressourcen als geplant eingesetzt werden mussten. Die Erprobung ist aber noch für 2014 vorgesehen.
AP 8	70%	Eine Untersuchung der Wirtschaftlichkeit von WOAS/WOAD gegenüber klassischen sowie serverorientierten Automatisierungssystemen konnte wegen des Fehlens eines Mitarbeiters mit BWL-Kenntnissen noch nicht durchgeführt werden. Die Umsetzung dieser Teilaufgabe ist aber weiterhin im Rahmen einer Projektarbeit mit Studierenden des Studiengangs Wirtschaftsingenieur geplant. Dazu sollen noch geeignete Studierende in 2014 gefunden werden.
AP 9	> 100	Die Transferaktivitäten übersteigen die ursprünglichen Planungen erheblich.

## 6 Veröffentlichungen und Transfermaßnahmen

### 6.1 Veröffentlichungen in Fachzeitschriften

Pressemeldung in der Zeitschrift Open automation, 6/2011, S.18

Nachgehakt bei Prof. Dr.-Ing. Reinhard Langmann: Das Projekt WOAS. - Computer & Automation, 12/2011, S. 66

Langmann, R.: Automation aus der Cloud. - Computer & Automation, 11/2012, S. 66 -72

In der Neuauflage des „Taschenbuch der Automatisierung“ (Hrsg.: Reinhard Langmann, Carl Hanser Verlag), geplant für 2015, wird die WOAS-Methodik im Kapitel „Strukturen“ mit aufgenommen.

## 6.2 Veröffentlichungen auf Konferenzen

*Langmann, R.; u.a.: The WOAS project: Web-oriented Automation System. - REV 2012 – Intern. Conference on Remote Engineering & Virtual Instrumentation, 03.07.12 – 06.07.12, Bilbao/Spanien*

*Langmann, R.: Automatisierungsdienste aus der Cloud - Neue Trends in der Prozessleittechnik. - Kongress zur SPS/IPC/DRIVES 2012, 27. – 29.11.12, Nürnberg*

*Langmann, R.: Effiziente Nutzung verteilter Automatisierungsfunktionen –Mit dem WOAS-Konzept zu Industrie 4.0. - Forum Industrial IT und Integrated Industry, Hannover-Messe, 10.04.13*

*Langmann, R.; Meyer, L.: Architecture of a Web-oriented Automation System. - ETFA 2013 – IEEE International Conference on Emerging Technology & Factory Automation, 10.09.13 – 13.09.13, Cagliari/Italien*

*Langmann, R.: Die Rolle der IT in einer Industrie 4.0 Referenzarchitektur. - EUROFORUM Conference „Industrie 4.0“, 18. – 19.11.13, Stuttgart*

*Langmann, R.: Ein Interface für CPS-basierte Automatisierungsgeräte. – Konferenz AALE 2014, 08. – 09.15.14, Regensburg, Tagungsband S. 133 - 142*

*Langmann, R.: Rent an Automation Service - Smarte Geschäftsmodelle für Industrie 4.0. GMA-Kongress Automnation 2014, Beitrag angenommen*

## 6.3 Presseinformationen über WOAS

Mitteilung über WOAS in den VDI nachrichten vom 30.11.12 und 15.12.12

Information zu WOAS in der Zeitschrift „Messtechnik“ 12/2013 , S. 57

## 6.4 Workshops und Seminare

0,5-tägiges Workshop zur Vorstellung des WOAS-Konzepts am 29.01.13 bei Phoenix Contact in Bad Pyrmont für Entwickler

1-tägiges Workshop zu den Prozessdatenzugängen in WOAS am 27.09.13 bei Phoenix Contact in Bad Pyrmont für das Management

0,5-tägiges WOAS-Workshop am 07.10.13 bei Weidmüller in Detmold für das Entwicklungs-Management

Durchführung eines WOAS-Workshops (3 h) „The WOAS project–A possible CPS Integration Platform“ zur Int. Konferenz REV 2014 in Porto

Seit Start des Projekts wurden fünf ganztägige Projektmeetings mit den Mitgliedern des PA durchgeführt (davon vier in der Projektlaufzeit).

Die WOAS-Methodik wird seit SS2014 im Masterstudiengang “Elektro- und Informations-technik” im Pflichtfach “Modellierung & Simulation” gelehrt und im Rahmen von studentischen Projekten werden Automatisierungsdienste modelliert, implementiert und im WOAS-Portal getestet.

## **6.5 Messen**

Vorstellung des WOAS-Konzepts sowie erster Ergebnisse auf einem eigenen Messestand zur SPS/IPC/DRIVES vom 27. – 29.11.12 in Nürnberg. Mitaussteller war das Mitglied des PA: Wiesemann&Theis, Wuppertal.

## **6.6 Transfermaßnahmen per Internet, Flyer u.a.**

Zum öffentlichen und internen Zugang zu den WOAS-Dokumenten wurden im Webportal der Forschungsstelle [www.telelabor.de](http://www.telelabor.de) zwei Bereiche (Gastbereich und internes Dokumentenportal) erstellt. Im internen WOAS-Dokumentenportal „WOAS intern“ befinden sich alle WOAS-Entwicklungsdocumente für die Mitglieder des PA.

Entwicklung, Druck und Verteilung eines WOAS-Info-Flyer (bisher ca. 300 Stück)

## **6.7 Arbeitskreise, Fachausschüsse**

Mitarbeit des Projektleiters seit 2011 im VDI/VDE-GMA Fachausschuss „Cyber Physical Systems“ und Vorstellung des Projekts im Fachausschuss

## **7 Nutzen für KMUs**

Die Forschungsergebnisse können prinzipiell in allen Wirtschaftszweigen und Branchen für die Automatisierung technischer Prozesse genutzt werden. Wegen der Möglichkeit zur Schaffung hochflexibler und „schlanker“ Automatisierungslösungen werden als Einsatzbereich aber eher typische KMU-Lösungen im Sektor des kundenspezifischen Geräte-/Maschinen- und Anlagenbaus gesehen.

Viele KMUs, und dazu gehören auch die KMUs des PA, beschäftigen sich mit Entwicklung und Vertrieb automatisierungstechnischer Gerätetechnik auf der MSR<sup>9</sup>-Ebene und müssen für kundenspezifische Gesamtlösungen Systemkomponenten für übergeordnete Automatisierungsfunktionen dazukaufen, da für entsprechende eigene FuE-Arbeiten die Personalkapazität in den KMUs meist nicht ausreicht.

Die Forschungsergebnisse ermöglichen es diesen KMUs durch industriennahe Folgeprojekte eigene angepasste und „schlanke“ Automatisierungsprodukte und –leistungen für die Prozess- und Betriebsleitebene mit tragbarem Aufwand zu entwickeln und damit ihr Portfolio zu erweitern, so dass sie moderne webbasierte Automatisierungslösungen aus einer Hand anbieten können. Dies steigert den Innovationsgrad und die Produktverantwortung der KMUs und damit ihre Leistungsfähigkeit. Beispiele dazu sind Web-basierte Operator-Panels und Prozessdaten-Visualisierung auf Basis der WOAS-Methodik.

Im globalen Wettbewerb stehen die KMUs in Konkurrenz mit Firmen aus Fernost (z.B. Moxa) die gleichfalls sehr kostengünstig einzelne Baugruppen und MSR-Module (E/A-Module, Kleinststeuerungen, Bedienterminals) liefern können. Die zusätzliche Nutzung von Komponenten eines weborientierten Automatisierungssystems und dessen kundenspezifische Anpassung gibt den KMUs einen wesentlichen Wettbewerbsvorteil gegenüber diesen Firmen und bietet neue Marktchancen bei der Lieferung schlüsselfertiger Lösungen.

---

<sup>9</sup> MSR – Messen, Steuern, Regeln

Darüber hinaus ermöglichen die Projektergebnisse neue Geschäftsmodelle durch das Vermieten von Automatisierungsfunktionen (SaaS-Modell), die sich durch die erforderliche Flexibilität und Agilität insbesondere für KMUs eignen und hier den KMUs neue Märkte mit hohem Wachstumspotential erschließen können. Basierend auf WOAS/WOAD kann eine Vermietung von AF prinzipiell für jede AF oberhalb der Feldebene erfolgen. Vorzugsweise werden dazu aber zuerst AF aus dem Prozessführungs-, Prozesskoordinierungs- und Leitbereich zum Einsatz kommen. Dazu gehören z.B. Visualisierungsdienste für Prozessdaten, interaktive Bedienelemente, Datenarchivierung und –auswertung, Messdatenverarbeitung, Alarmbehandlung. Für zeitunkritische Steuerungsanwendungen ist auch für einzelne spezifische Einsatzfälle (z.B. Beleuchtungs- und Klimasteuerung) die Vermietung von Steuerungsfunktionalitäten denkbar.

Die Nutzung eines SaaS-Modells stößt z.Z. bei den KMUs noch auf Akzeptanzprobleme, da diese noch das Risiko scheuen, ein vollständig neues Geschäftsmodell für Automatisierungslösungen für ihre Kunden anzubieten. Mit WOAS steht aber eine Methodik und ein Testportal zur Verfügung, welches in der nächsten Zeit sicher auch von einzelnen KMUs des PA eingesetzt werden wird (wenn auch evtl. nur probeweise).

Durch die Zusammenarbeit der KMUs mit führenden Global-Playern der Automatisierungstechnik (Nicht-KMU) im projektbegleitenden Ausschuss erhielten die KMUs des PAs weiterhin Zugang zu entsprechenden Netzwerken und es wurde eine Bewusstseinssteigerung für den Einsatz neuer ITK-Technologien bei den beteiligten KMUs erzeugt. Über das WOAS-Projekt waren die KMUs gleichzeitig an Lösungen zu Industrie 4.0 beteiligt und kommunizierten dies auch gegenüber ihren Kunden (Beispiel: Vorstellung WOAS durch MSF Vathauer auf einer AUDI-Industrie 4.0-Veranstaltung)

## 8 Transferkonzept

Die im Projekt bereits durchgeführten Transfermaßnahmen sind in Kapitel 5 im Detail beschrieben. Im folgenden werden die nach Projektende beabsichtigten Maßnahmen aufgeführt, die WOAS bzw. Teile davon in eine nachhaltige Nutzung überführen sollen.

### 8.1 Einzelmaßnahmen:

Mit einzelnen Unternehmen des PA werden z.Z. Diskussionen und Gespräche geführt, um je nach Bedarf der Unternehmen Teilergebnisse von WOAS in Kundenprodukten zu nutzen. Dazu gehören z.B. die browser-basierte Bedienung von Servoantrieben und Frequenzumrichtern, die Überwachung und Diagnose von Heizkesselanlagen sowie die Bedienung & Beobachtung von Modbus TCP-gesteuerter E/A-Baugruppen.

Alle PA-Mitglieder besitzen einen Testzugang zum WOAS-Portal und können WOAS-Lösungen gegenüber eigenen Kunden demonstrieren, um diese z.B. über den Einsatz von Industrie 4.0-Lösungen zu informieren und für eine Nutzung zu motivieren.

Mit weiteren PA-Mitgliedern (Nicht-KMUs) wird über eine zukünftige strategische Forschungszusammenarbeit auf dem Gebiet der Anwendung der Internettechnologie für die Automation diskutiert.

Die WOAS-Ergebnisse bilden die Grundlage, um darauf aufbauend Industrie 4.0-FuE-Projekte zu entwickeln und zu beantragen. Dazu wurde bisher bereits ein neuer IGF-Projektantrag im April 2014 mit dem Titel „Potential, Struktur und Schnittstellen von Cloud-basierten industriellen Steuerungsdiensten (Cloud-based Industrial Control Services –

CICS)“ eingereicht. Projektkoordinator sind die Forschungsstelle und die Uni Augsburg, Lehrstuhl Kommunikationssysteme gemeinsam.

Im CICS-Projekt soll die WOAS-Methodik auf die unteren Ebenen der klassischen Automatisierungspyramide (Steuerungseben, Feldebene) übertragen werden, um eine CPS-basierte Automation nach Bild 35 realisieren zu können. Im PA von CICS sind auch drei Unternehmen mit beteiligt, die bereits an WOAS teilgenommen haben.

Ein weiterer Antrag im Rahmen des bayrischen FuE-Förderprogramms BY mit dem Titel „Interaktives Videostreaming für industrielle Anwendungen (INVIA)“ wird aktuell vorbereitet. Dabei ist u.a. vorgesehen, dass das WOAS-Portal für die Bedienung&Beobachtung von Testszenarios genutzt werden soll. Projektkoordinator ist das Fraunhofer EKS (München). Die Forschungsstelle wird im Projekt INVIA als wissen. Projektpartner mitwirken.

Für Oktober/November 2014 wird ein öffentliches Workshop „Web-basierte Projekte & Lösungen im Umfeld von Industrie 4.0“ vorbereitet, in dem neben einigen Vorträgen der PA-Mitglieder die WOAS-Ergebnisse ausführlich mit praktischer Vorführung vorgestellt werden.

## 8.2 Strategische Maßnahmen:

Die strategischen Maßnahmen für eine Überführung der WOAS-Methodik in eine praktische Nutzung hängt von den zur Verfügung stehenden Ressourcen der Forschungsstelle ab. Als Fachhochschule besitzt die Forschungsstelle grundsätzlich keine grundfinanzierten Mitarbeiter, die für Forschungszwecke bzw. für Transfermaßnahmen eingesetzt werden können. Dies ist in einer FH bisher so nicht vorgesehen. Für jede weitere WOAS-Transfermaßnahme nach Projektende müssen die erforderlichen Personalressourcen entsprechend eingeworben werden, d.h. es müssen Industriepartner gefunden werden, die WOAS auch weiterhin mit finanziellen Mitteln unterstützen. Grundsätzlich wurden auf einem zusätzlichen WOAS-Meeting mit dem PA am 22.05.14 (nach Projektende) mögliche strategische WOAS-Umsetzungskonzepte vorgestellt und für eine Beteiligung der PA-Mitglieder geworben. Dazu gehören:

- *Technologiepartnerschaft*: Im Rahmen einer Vereinbarung beteiligen sich die Industriepartner des PA (+ evtl. externe Partner) an einer Technologiepartnerschaft mit der Forschungsstelle, um die WOAS-Thematik im Rahmen von Industrie 4.0-Projekten weiterzuführen. Bei Beteiligung mehrerer Unternehmen würde sich der finanzielle Aufwand pro Unternehmen in Grenzen halten.
- *StartUp*: Gründung eines StartUp zur Umsetzung von WOAS in ein Produkt. Auf Basis von WOAS wird ein Produkt „Process Control System from the Cloud“ entwickelt. Entwicklungszeit: ca. 2 Jahre mit 2 ...3 professionellen Java-und Webentwicklern. Vermarktet werden könnten dabei Migrationswerkzeuge zu Industrie 4.0-Architekturen, Mietdiensste aus einem WOAS-Portal, Vertrieb des WOAS-Portals selbst für lokale Cloud-Lösungen. Geschätzter Finanzierungsbedarf für eine WOAS-Produktentwicklung: ca. 0,8 ...1 Mio. EUR.
- *WOAS Open Community*: Das WOAS-Integrationsportal wird durch die Forschungsstelle einschl. der Beschreibung in Englisch zur Erstellung von VDs und Diensten in das öffentliche Netz gestellt und über verschiedene Aktivitäten beworben. Dritte können eigene Geräte in das Portal einbinden und auch eigene Dienste entwickeln und integrieren. Der Vorteil dabei wäre, dass bei einer funktionierenden Community nach einiger Zeit eine Vielzahl von Automatisierungsdiensten entstehen, die durch alle Nutzer kostenfrei einge-

setzt werden können. Damit könnte erstmalig eine Plattform für z.B. eine offene und kostenfreie Prozessvisualisierung im Web für einfache und nicht kritische Automatisierungs-lösungen entstehen. Die Finanzierung für Betrieb und Wartung eines offenen WOAS-Portals könnte durch Werbung und Spenden der Nutzer erfolgen.

Aktuell werden die o.a. Strategien mit verschiedenen Partnern diskutiert, um bis Ende 2014 festzulegen, welche Strategie tatsächlich umgesetzt werden kann.